

# Handlungsbeobachtung zur Instruierung von Robotersystemen

Zur Erlangung des akademischen Grades eines  
Doktors der Natur-/Ingenieurwissenschaften  
der Fakultät für Informatik  
der Universität Karlsruhe (Technische Hochschule)

vorgelegte

## Dissertation

von

Markus Ehrenmann

aus Sigmaringen

Tag der mündlichen Prüfung:	25.6.2003
Erster Gutachter:	Prof. Dr.-Ing. R. Dillmann
Zweiter Gutachter:	Prof. Dr. rer. nat. A. Waibel

HIRSCHSTRASSE 144 • 76137 KARLSRUHE  
TELEFON 0721/608-4243 • PRIVAT 0179/6674728  
E-MAIL [EHRENMAN@IRA.UKA.DE](mailto:EHRENMAN@IRA.UKA.DE)

MARKUS EHRENMANN

## PERSÖNLICHE INFORMATION

---

- Familienstand: ledig
- Staatsangehörigkeit: deutsch
- Geburtstag: 30.11.1970
- Geburtsort: Überlingen/Bodenseekreis



## AUSBILDUNG

---

### Schulausbildung

Aug. 1977 – Jun. 1981	Bilharz-Grundschule	Sigmaringen
Aug. 1981 – Mai 1990	Gymnasium Hohenzollerngymnasium	Sigmaringen

*Abschluss: Abitur*

- Note: Sehr gut (1,5)
- Leistungskurse: Altgriechisch, Bildende Kunst
- 1988 Teilnahme an einem 2-monatigen Austauschprogramm mit Banff, Kanada

### Wehrdienst

Jul. 1990 – Jun. 1992	Nachschubkompanie 290	Horb, Stetten a.k.M., Amberg, München
-----------------------	-----------------------	---------------------------------------

*Abschluss als Stabsunteroffizier im Sanitätsdienst und Krankenpflegehelfer*

- 1991 Bestehen der zweiten Stufe des Informatik-Wettbewerbs der Gesellschaft für Mathematik und Datenverarbeitung

### Studium

Okt. 1992 – Jan. 1998	Universität Karlsruhe (TH)	Karlsruhe
-----------------------	----------------------------	-----------

*Studiengang: Informatik*

- Abschluss: Diplom in Informatik
- Note: Sehr gut (1,2)
- Schwerpunktfächer: Logik und Grundlagen, Kognitive Systeme
- Ergänzungsfach: Philosophie

Apr. 1997 – Dez. 1997	Universidad de Málaga	Málaga, Spanien
-----------------------	-----------------------	-----------------

*Auslandsstudium und Diplomarbeit*

- Thema der Diplomarbeit: „Objekterkennung in Kamerabildern in einem fusionierten Ansatz von Maschinensehen und Greifwinkelbetrachtung“ (Note: 1,0)

### Promotion

Feb. 1998 – heute	Universität Karlsruhe (TH)	Karlsruhe
-------------------	----------------------------	-----------

*Doktorand an der Fakultät für Informatik*

- Promotion als Doktor der Ingenieurwissenschaften (Dr.-Ing.)
- Prüfungstermin: 25.6.2003
- Thema der Dissertation: „Beobachtung von Benutzerhandlungen zur Instruierung von Robotersystemen“

---

# Inhaltsverzeichnis

Inhaltsverzeichnis	i
<b>1 Einführung</b>	<b>1</b>
1.1 Herausforderung Roboterassistenz	1
1.2 Roboterprogrammierung	4
1.2.1 Aufgabenorientierte textuelle Programmierung	6
1.2.2 Symbolisches Programmieren	7
1.2.3 Analyse manueller Programmierung	7
1.2.4 Programmieren durch Vormachen	8
1.3 Mensch-Roboter-Interaktion	9
1.4 Beitrag der Arbeit	9
1.5 Aufbau der Arbeit	11
<b>2 Stand der Forschung</b>	<b>13</b>
2.1 Skopus und Randbedingungen	13
2.2 Sensorentwicklung	14
2.2.1 Bildgebende Sensoren	14
2.2.2 Magnetfeldbasierte Positionssensoren	15
2.2.3 Datenhandschuhe	16
2.2.4 Datenanzüge	16
2.2.5 Exoskelette	17
2.3 Spezielle Aspekte der Handlungsbeobachtung	18
2.3.1 Szenenanalyse	19
2.3.2 Objektverfolgung	20
2.3.3 Gestenerkennung	24
2.3.4 Griffenerkennung	26
2.4 Qualitative Handlungserkennung	27
2.4.1 Programmieren durch Vormachen	29
2.4.2 Interaktive Ansätze	33
2.5 Vergleichende Bewertung der Ansätze	35
2.6 Zusammenfassung	39
<b>3 Beobachtung menschlicher Handlungen</b>	<b>41</b>
3.1 Instruierung von Menschen	41
3.2 Typen beobachtbarer Handlungen	43

3.3	Zusammenfassung . . . . .	45
<b>4</b>	<b>Sensorik und Modellierung</b>	<b>47</b>
4.1	Die Interpretation komplexer performativer Handlungen . . . . .	47
4.2	Randbedingungen . . . . .	49
4.3	Sensorik . . . . .	50
4.3.1	Vorführungsumgebung für performative Handlungen . . . . .	50
4.3.2	Ausführungsumgebung zur Interaktion . . . . .	55
4.4	Systemarchitektur . . . . .	56
4.5	Modellierung . . . . .	57
4.5.1	Weltmodell . . . . .	58
4.5.2	Handmodell . . . . .	58
4.5.3	Handlungsmodell . . . . .	59
4.6	Beobachtung elementarer Handlungen . . . . .	60
4.6.1	Handverfolgung . . . . .	61
4.6.2	Grifferkennung . . . . .	63
4.6.3	Gestenerkennung . . . . .	65
4.7	Zusammenfassung . . . . .	67
<b>5</b>	<b>Elementare kognitive Operatoren</b>	<b>69</b>
5.1	Szenenanalyse . . . . .	70
5.1.1	Konturbasierte Objektdetektion . . . . .	71
5.1.2	Musteranpassung . . . . .	75
5.1.3	Farbbasierte Objektdetektion . . . . .	76
5.1.4	Diskussion . . . . .	79
5.1.5	Positionsbestimmung . . . . .	81
5.1.6	Operator zur Objektdetektion . . . . .	86
5.2	Handverfolgung . . . . .	87
5.2.1	Handverfolgung in der Vorführungsumgebung . . . . .	87
5.2.2	Handverfolgung in der Ausführungsumgebung . . . . .	99
5.2.3	Operator zur Bewegungsverfolgung . . . . .	102
5.3	Grifferkennung . . . . .	103
5.3.1	Kalibrierung des Datenhandschuhs . . . . .	104
5.3.2	Schichtklassifikator mit Verwendung objektspezifischer Griffinformation	105
5.3.3	Operator zur Griffdetektion . . . . .	107
5.4	Gestenerkennung . . . . .	107
5.4.1	Erkennung statischer Gesten in der Vorführungsumgebung . . . . .	107
5.4.2	Erkennung statischer Gesten in der Ausführungsumgebung . . . . .	109
5.4.3	Erkennung dynamischer Gesten in der Ausführungsumgebung . . . . .	114
5.4.4	Operator zur Gestendetektion . . . . .	122
5.5	Registrierung beobachteter Ereignisse . . . . .	122
5.6	Zusammenfassung . . . . .	126



<b>6 Experimentelle Validierung</b>	<b>127</b>
6.1 Validierung der kognitiven Operatoren	127
6.1.1 Szenenanalyse	127
6.1.2 Bewegungsverfolgung	132
6.1.3 Grifferkennung mit Hilfe des Datenhandschuhs	140
6.1.4 Gestenerkennung	142
6.2 Validierung der Handlungsbeobachtung	161
6.2.1 Handlungsbeobachtung in der Vorführungsumgebung	161
6.2.2 Handlungsbeobachtung in der Ausführungsumgebung	162
6.3 Zusammenfassung	162
<b>7 Schlußbetrachtung</b>	<b>171</b>
7.1 Zusammenfassung der erzielten Ergebnisse und Erkenntnisse	171
7.2 Diskussion	172
7.3 Ausblick	174
<b>A Definitionen</b>	<b>177</b>
<b>B Technische Daten der verwendeten Sensorik</b>	<b>179</b>
<b>C Farbräume</b>	<b>181</b>
<b>D Farbkonstanz</b>	<b>185</b>
<b>Literatur</b>	<b>194</b>
<b>Stichwortverzeichnis</b>	<b>212</b>

---

# Kapitel 1

## Einführung

Das Interesse an androiden Maschinen fand bereits im 18. Jahrhundert einen Höhepunkt in den mechanischen Apparaten Vaucansons, Drosz' oder Leschotts. Es zeichnete sich jedoch bald ab, dass die Steuerungen dieser schreibenden, zeichnenden oder klavierspielenden Automaten, die durch Feder- und Uhrwerke realisiert waren, nicht flexibel auf ihre Bewunderer in den Salons reagieren konnten. Die Anziehungskraft der damaligen Exponate schwand zwar bald, nicht jedoch die Faszination der Thematik. Die Autoren, die sich damit beschäftigen, sind Legion. In E. T. A. Hoffmanns Erzählung „Die Automate“ erledigen Roboter beispielsweise nicht nur die Arbeit. Er bezeichnet sie als wunderliche, lebendig-tote Figuren, die sogar herzergreifende Melodien singen können [Speidel 02].

Das Verlangen nach Apparaten, die dem Menschen in einer Mischung aus Arbeitshilfe und Unterhaltungsinstrument dienen, auf ihn reagieren und wie er kommunizieren, ist bis heute ungebrochen. Es findet außer in der Literatur und im Film nicht zuletzt seinen Ausdruck in den künstlichen Haustieren, die bereits kommerziell vertrieben werden. In Japan werden diese als „digitale Lebensgefährten“ apostrophierten Maschinen bereits als Vorstufe zum Humanoiden gefeiert [Frankfurter Allgemeine Zeitung 01b]. Flankiert von Utopien und spielerischen Anwendungen wird die Robotik heute mit glänzenden Zukunftsaussichten beschieden; euphorische Forscher prognostizieren sie als bald „größte Industrie des Planeten“ [Moravec 00].

### 1.1 Herausforderung Roboterassistenz

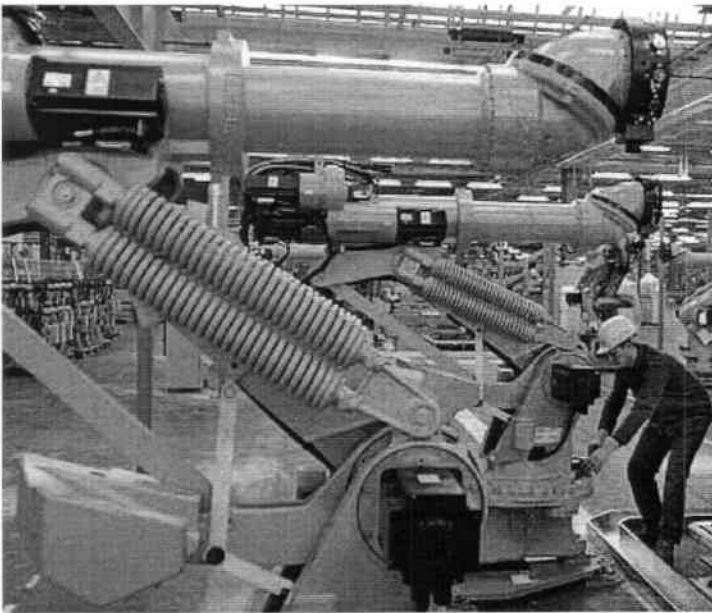
Die wichtigste Abnehmerbranche für Roboter ist in Deutschland derzeit nach wie vor die Automobilindustrie<sup>1</sup>, gefolgt vom Einsatz in der Schweißtechnik vor allem im Maschinenbau. Obwohl in diesen Sparten immer noch Rationalisierungsmöglichkeiten bestehen [SPIEGEL ONLINE 01], verzeichnen sie inzwischen keine nennenswerten Zuwächse mehr durch Neuinstallationen. Neue Einsatzfelder bieten sich gegenwärtig in der Lebensmittelindustrie, in der Chemie- und Kunststoffindustrie, in Warenverteilzentren und in

---

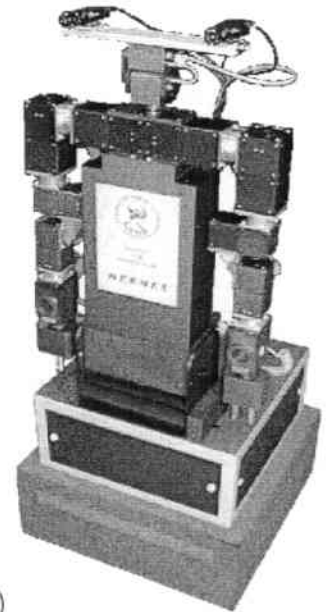
<sup>1</sup>1999 verzeichnete die Automobilindustrie 50 Prozent aller in Deutschland installierten Roboter [Frankfurter Allgemeine Zeitung 00]

der pharmazeutischen Industrie. Obwohl seitens der Industrie großes Interesse besteht, Produktionsabläufe zu automatisieren, ist hier Voraussetzung, dass Roboter leichter programmierbar und dadurch billiger werden. Deutsche Roboterhersteller verdienen in jüngster Zeit fast soviel Geld mit der Schulung ihrer Kunden wie mit dem Verkauf ihrer Maschinen [Frankfurter Allgemeine Zeitung 01a]. Erst das Senken des Bedarfs an teuren Programmierexperten kann hier zur Kostenreduktion führen. Ursache des heute erforderlichen immensen Aufwands ist die Komplexität heutiger Systeme. Eine breite Marktakzeptanz von Robotern kann aber nur dann eintreten, wenn sie ohne besondere Einarbeitung auch in Krankenhäusern, Werkstätten oder gar zuhause benutzt werden können.

Die Vertrautheit mit der inzwischen alltäglichen Anwendung von *Servicerobotern* wie Reinigungs-, Kletter- oder Rohrinspektionsmaschinen lässt auch erwarten, dass psychologische Barrieren für maschinelle Assistenten in Haushalt und Pflege in Zukunft überwunden werden [Frankfurter Allgemeine Zeitung 99a]. Multimodale Schnittstellen zwischen Mensch und Maschine, welche Spracherkennung, Gesten- und Blickwinkelverfolgung sowie graphische Eingaben integrieren, sollen solche Systeme einfach bedienbar machen. Dies stellt wiederum hohe Anforderungen bezüglich der Anpassungsfähigkeit und kognitiven Leistung bei solchen Maschinen. Die Nachfrage danach und das Interesse daran belegt jedoch nicht zuletzt das Wachstum der Umsätze bei Sensorherstellern [Frankfurter Allgemeine Zeitung 99b].



(a)



(b)

Abbildung 1.1: Industrieroboter im Automobilbau (a) und Roboterassistent *HERMES* der TU München (b)

Die Klasse dieser sogenannten *Roboterassistenten* oder *Serviceroboter* steht damit hinsichtlich ihrer Flexibilität im Gegensatz zu den „klassischen“ Industrierobotern, die in einer festen Umgebung feste Programmschleifen zyklisch wiederholen (siehe Abbildung 1.1). Der Begriff Industrieroboter ist nach ISO genormt als „automatisch gesteuertes, wiederprogrammierbares, vielfach einsetzbares Handhabungsgerät mit mehreren Freiheitsgraden, das

entweder ortsfest oder beweglich in automatisierten Fertigungssystemen eingesetzt werden kann“ [ISO 00]. Der des *Serviceroboters* grenzt sich davon vor allem durch die Eigenschaft ab, mit dem Menschen auch physisch zu interagieren oder sich in unbekanntem Terrain zurechtzufinden. Mit *Assistenzsystemen* verbindet sich die Erwartung, Aufgaben sogar kooperativ zu lösen. Diese sind damit gekennzeichnet durch zwei wesentliche Eigenschaften:

- Es besteht eine direkte Interaktion zwischen Mensch und Maschine. Aufgaben werden kooperativ gelöst. Die Kommunikation spielt dabei eine wesentliche Rolle.
- Es sind wechselnde, nicht getaktete Aufgaben in schwach strukturierten Umgebungen auszuführen. Der Roboter muss sich an seine Umgebung anpassen und über die Flexibilität verfügen, an neue Aufgaben schnell angepasst werden zu können.

Die Verwendung des Begriffs der Roboterassistenz hat sich erst in den letzten Jahren etabliert. Eine ähnliche, graphische Definition wird in Abbildung 1.2 widergegeben (zitiert nach [Hägele 01]). Auch hier sind die trennenden Merkmale der Grad der Interaktion, der Autonomie und der Umweltstrukturierung.

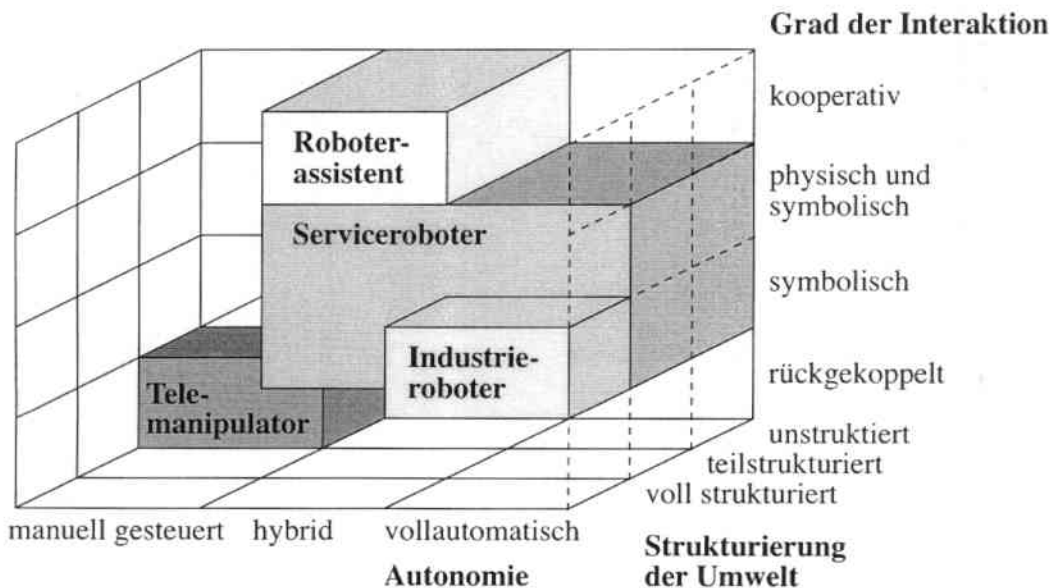


Abbildung 1.2: Abgrenzung von Roboterassistenten zu Servicerobotern und anderen Robotersystemen nach [Hägele 01]

Es sollte jedoch bei der Diskussion dieser Roboter nicht nur der ökonomische Nutzen Beachtung finden, sondern auch der erkenntnistheoretische Fortschritt. Dieser ist mit der Möglichkeit gegeben, bei der Konstruktion solcher Maschinen Einblick in das Funktionieren des menschlichen Verstandes zu gewinnen bzw. schon vorhandene Erkenntnisse zu formalisieren und zu testen. Es ist der Verdienst von Brooks, darauf hinzuweisen, dass Intelligenz

immer an den jeweiligen Körper gebunden ist, obwohl diese „Körperlichkeit“<sup>2</sup> und die Wechselwirkung zwischen Körper und ihn umgebender Welt in der Erforschung künstlicher Intelligenz bis Ende der achtziger Jahre überhaupt keine Rolle gespielt hatten [Brooks 86].

Erst die Untersuchung dieses Problems macht den flexiblen Einsatz von Robotern möglich. Gerade die Entwicklung der Intelligenz sowie der Wissensrepräsentation bei Kognitions- und Schlüsselaufgaben derartiger Maschinen stellt aber bis heute die grösste Herausforderung dar [Singer 00]. Die Triade aus Mobilität, Beobachtungsgabe und der Fähigkeit, in dynamischen Umgebungen zu überleben, zeichnen heute nach Meinung vieler Experten die notwendige Basis für die Entwicklung echter Intelligenz aus.

Dabei stellt sich nicht zuletzt die Frage, wie die Abbildung intelligenten Verhaltens auf eine Maschine erfolgen soll. Das Wissen zu einer Problemlösung ist nach aktuellen Vorstellungen des Wissensmanagements immer direkt an Personen gebunden [Probst 99]. Der folgende Abschnitt behandelt daher Techniken, die Menschen die Übertragung ihres Wissens auf Robotersysteme gestatten.

## 1.2 Roboterprogrammierung

Die Robotern einzuhauchende Intelligenz, die genauso notwendig für Perzeptionsaufgaben und das Lösen von Manipulationsproblemen ist wie auch für die Interaktion mit einem menschlichen Kommunikationspartner, erfordert Wissen über die Welt. Der hier verwendete Wissensbegriff ist derjenige der Informationsverarbeitung ([Devlin 99], siehe Tabelle 1.1).

Daten	:=	Zeichen + Syntax
Information	:=	Daten + Bedeutung
Wissen	:=	Internalisierte Informationen + Fähigkeit, sie zu nutzen

Tabelle 1.1: Zusammenhang von Daten, Information und Wissen

Wenn für die Robotik in diesen Kurzformeln Daten mit Sensormesswerten gleichgesetzt werden, können sie für die Programmierung von Roboterassistenten übernommen werden. Wodurch sich die Fähigkeit zur Nutzung internalisierter Information konstituiert, ist jedoch nicht klar. Deutlich ist allein, dass hier kognitive und motorische Fähigkeiten zusammen mit Problemlösungsmethoden integriert sein müssen. Mithin die grössten Forschungsbemühungen widmen sich heute immer noch der Perzeption, der Erkennung der eigenen Raum-Zeit-Position, des Bewegungsfreiraums und der manipulierbaren Umwelt sowie dem Erlernen dort stattfindender komplexer, zielgerichteter Eingriffe.

Das Wissen um die Behandlung von Manipulationsaufgaben wird in der industriellen Robotik bislang durch roboterorientierte textuelle Programmierverfahren oder manuelles Program-

<sup>2</sup>engl.: Embodiment

mieren, d.h. direktes Anfahren von Positionen mit dem Roboter selbst, von einem Experten an einen Automaten vermittelt:

**Textuelle Programmierung:** Hierbei handelt es sich zum Einen um den Einsatz höherer Programmiersprachen wie *C* oder *Pascal*, die um spezielle Roboterbefehle erweitert sind. Hier ergibt sich ein Vorteil durch die Nutzung von klassischen Konstrukten wie Schleifen oder Verzweigungen. Damit lassen sich flexible Roboterprogramme erzeugen.

**Manuelle Programmierung:** Abbildung 1.3 a zeigt eine Bedieneinheit für Mehrachsroboter<sup>3</sup>, mit Hilfe derer man Bewegungsbahnen auf Gelenkebene programmieren kann. Dazu werden Stützpunkte auf der gewünschten Trajektorie angefahren und abgespeichert<sup>4</sup>. Zur Ausführungszeit wird zwischen diesen Punkten eine Bahn interpoliert, die beliebig oft abgefahren werden kann. Dasselbe Prinzip liegt auch der haptischen Programmierung zugrunde: hier wird der Endeffektor mit Hilfe eines Kraft-Momenten-Sensors quasi auf Manipulatorebene durch direkte Verschiebung zu den Zielpunkten bewegt (siehe Abbildung 1.3 b).

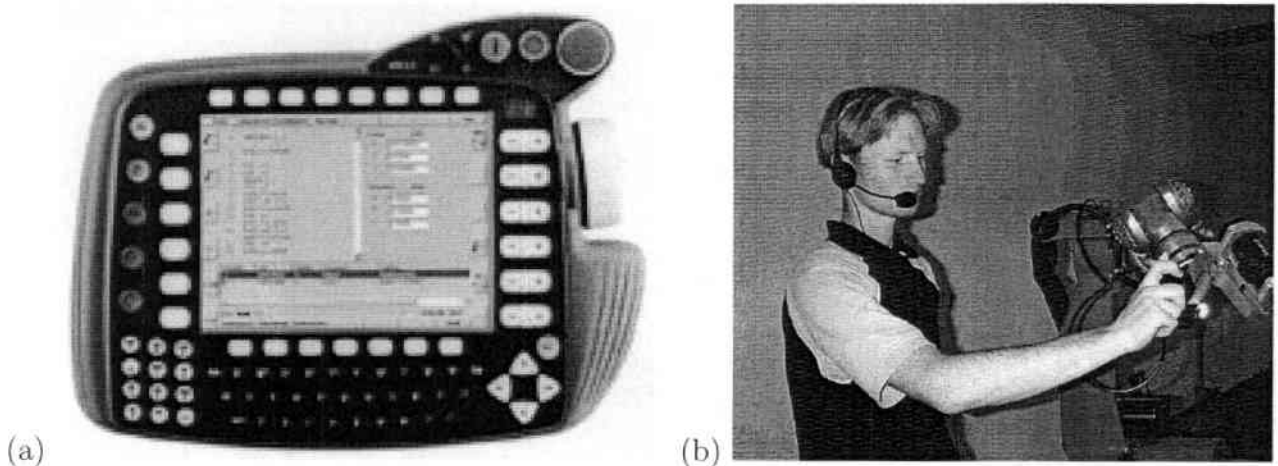


Abbildung 1.3: Bedienschnittstelle für Mehrachsroboter der Firma Kuka (a) und haptische Schnittstelle am Endeffektor des Roboterherstellers Reis (b)

Monkman gibt in [Monkman 91] an, daß weithin vier Ebenen bei der Programmierung von Robotern unterschieden werden können (siehe Tabelle 1.2). Jede Ebene repräsentiert hierbei einen bestimmten Abstraktionsgrad, der bei der Fusionierung von Sensormessungen auf dieser Ebene nicht verlassen werden sollte, um Modellierungsfehler zu vermeiden. So entsprechen beispielsweise der Gelenkebene direkte Sensormessungen wie Pixel- oder Encoderwerte, der Manipulatorebene Bildarrays oder Geschwindigkeitsdaten, während die Sensorfusionierung auf Objektebene als boolesche Formeln aus den Einzelmessungen erscheinen. Auf Aufgabenebene stehen dann aus den Messungen geschlossene Beschreibungen wie „kein Werkzeug vorhanden“.

<sup>3</sup>engl.: Teachpanel

<sup>4</sup>engl.: Teach-In



Ebene	Problemlösungswissen	Programmierung	Abstraktionsgrad
Aufgabenebene	Vollständige Aufgabenroutine	„Baue Produkt“	geschlossene Beschreibungen
Objektebene	Programmsegmente	„Greife Teil A“	Sensorfusion, Objekte, Klassen
Manipulatorebene	Robotersprachenkommandos	„Fahre Greifer an Position von A“	Merkmale, Bildarrays, Kartesisches Bezugssystem
Gelenkebene	Programmierprimitive	„Gelenk A auf Position B“	Gelenkwerte, Bildpunkte

Tabelle 1.2: Ebenen der Roboterprogrammierung

Das Vokabular heutiger Programmiersprachen entspricht dem Abstraktionsgrad der ersten beiden Ebenen. In diese ordnen sich die Ansätze zur textuellen und manuellen Programmierung ein. Eine für Menschen interessante Kommunikationsebene stellt jedoch erst die Objekt- oder Aufgabenebene dar. Der qualitative Sprung auf diese Ebene entspricht jedoch dem von Information zu Wissen und hat sich als schwer modellierbar gezeigt. Für die Abbildung von Problemlösungswissen nach unten sind Experten notwendig. Hier liegt der Grund für die hohen Kosten bei der Roboterprogrammierung.

Es ist aber auch auf Aufgabenebene nicht einsichtig, wie ein Roboter auf unkomplizierte Art und Weise lernen soll, wie eine bestimmte Aufgabe abzuarbeiten ist. Mehrere Ansätze verstehen sich als Lösungsvorschläge zu diesem Problem. Sie werden im Folgenden kurz vorgestellt.

### 1.2.1 Aufgabenorientierte textuelle Programmierung

Bei dieser Methode werden einem Benutzer abstrakte Operatoren textuell zur Verfügung gestellt, die vollständige Lösungen für Teilaufgaben repräsentieren. Die Programmierung einer Lösung im Rahmen einer Aufgabe erfolgt durch die Verkettung der zur Verfügung stehenden Teillösungen. Diese Sequenz wird in ein roboterorientiertes Programm übersetzt. Der dafür eingesetzte Aufgabenplaner benötigt dazu eine Reihe von Informationen: ein Umweltmodell sowie das Wissen um die Zerlegung einer Aufgabe und um deren Teillösungen.

Der Bedarf an umfangreichem Hintergrundwissen und die fehlende individuelle Spezifikation von Verfahrensbahnen schränken den Nutzen dieses Verfahrens ein. Zwar können Programme auf Aufgabenebene erstellt werden, es können aber im Bedarfsfall keine fehlenden Bewegungsfolgen nachträglich integriert werden.

## 1.2.2 Symbolisches Programmieren

Wie bei der aufgabenorientierten textuellen Programmierung wird hier dem Benutzer gestattet, sich Roboterprogramme auf Basis von Teillösungen zusammenzusetzen, die symbolisch auf einem Bildschirm repräsentiert werden. Die Problematik liegt bei diesem Ansatz in der Bestimmung der Objekte, die manipuliert werden sollen, und der Parametrierung von Aktionen. Diese Aufgabe ist in der Regel umfangreich und schwierig. Zwei Systeme sollen beispielhaft für diese Herangehensweise genannt werden:

- Das System *SKORP*<sup>5</sup> stellt eine Menge aktorischer und kognitiver Elementaroperatoren wie „aufnehmen“, „fahre zu“ oder „identifiziere Objekt“ sowie Schleifen- und Verzweigungskonstrukte in einer symbolischen Programmieroberfläche zur Verfügung [Archibald 93]. Durch graphische Aneinanderreihung kann der Benutzer ein Programm erzeugen; die Parametrierung einzelner Kommandos erfolgt anschließend per Hand über Dialogmenüs.

Die Aufgabenprogrammierung wird hier von der roboterspezifischen Implementierung der Elementaroperatoren getrennt.

- Die visuelle Programmierung von  $2\frac{1}{2}$ D Transferaufgaben wird in dem System von Shepherd durchgeführt [Shepherd 93]. Die Programmierumgebung erlaubt die direkte Manipulation von Objekten und Bildbereichen mittels eines Zeigegerätes. Die dargestellten 2D-Szenen sind entweder konstruierte graphische Szenen oder Kamerabilder. Operationen zum Greifen und Ablegen werden durch Verschieben von Objektmodellen der Szenen spezifiziert und dann durch Situations-Aktions-Regeln repräsentiert. Die Regeln enthalten die Situation mit einer Auswahl relevanter Objekte als Vorbedingung für ihre Ausführung. Die Aktionen der Regeln werden auf dem Bildschirm nach der Spezifikation durch Verschiebungen der zu transportierenden Objekte dargestellt.

## 1.2.3 Analyse manueller Programmierung

Als Erweiterung manueller Programmierung wurden mehrere Methoden vorgeschlagen, die nicht nur die angefahrenen Punkte abspeichern, sondern zur Erzeugung flexibler Programme nutzbar machen:

- Beim *ETAR*-System<sup>6</sup> soll der Benutzer mehrfach den Roboterarm direkt Führen und Greifvorgänge ausführen lassen [Heise 92]. Die Lage des Greifers wird dazu ständig aufgezeichnet und mit Schwellwerten zur Datenreduktion gefiltert. Konfigurationen, bei denen der Greifer in der Nähe eines Objektes liegt, werden gespeichert, die anderen verworfen. Aus diesen verbleibenden Lagen werden symbolische Aktionen abgeleitet, die aus vier verschiedenen Handlungen bestehen: dem translatorischen und rotatorischen Verfahren, Greifen und Loslassen. Aktionen, die in der Nähe ein und desselben Objektes ausgeführt werden, lassen sich zu Gruppen zusammenfassen. Innerhalb der so gruppierten Aktionssequenz können Schleifen und Verzweigungen aus der Verschmelzung mehrerer Beispielprogramme gewonnen werden. Die Bedingungen hierzu werden

<sup>5</sup>engl.: Skill Oriented Robot Programming

<sup>6</sup>engl.: Example-based Task Acquisition in Robotics



automatisch generiert. Die Beispiele müssen für diesen Zweck jedoch hinlänglich ähnlich sein, d.h. die Positionen beim Verfahren dürfen nicht stark voneinander abweichen. Das System dient zur Programmierung von Transportaufgaben in einer Blockwelt.

Die Forderung nach mehreren Beispielen zur Lösung einer einzelnen Aufgabe schränkt jedoch die Anwendbarkeit ein. Außerdem ermöglicht das System dem Benutzer keine Überprüfung der Hypothesen zur Intention der einzelnen Aktionen.

- Die direkte Steuerung eines Roboterarms durch den Benutzer benutzt auch Onda, um die aufgenommenen Daten anschließend in einem Simulationssystem zu analysieren [Onda 97]. Programme werden hier als Zustandsübergänge zwischen den Kontaktrelationen der in einem geometrischen Umweltmodell gespeicherten Objekte interpretiert. Diese Übergänge werden bei der Simulation der Aufnahme berechnet und durch Verkettung flexibler Elementaroperationen realisiert. Liegt eine bis dato unbekannte Kontaktkonfiguration vor, so wird die programmierte Lösung als auf Elementaroperationen basierendes Programm gespeichert.

#### 1.2.4 Programmieren durch Vormachen

Ein vielversprechendes Programmierparadigma ist das *Programmieren durch Vormachen*<sup>7</sup>. Hier geht man von der Annahme aus, dass es Menschen zwar schwer fällt, verbal zu formulieren, welche Einzelschritte und Parameter zur Lösung einer Aufgabe notwendig sind, sie diese selbst aber leicht vorführen können. Ziel ist, eine Maschine eine solche menschliche Vorführung durch Sensoren als Musterbewegung registrieren und anschließend auf ihre eigene Kinematik übertragen zu lassen.

Hier gibt es einerseits Arbeiten zum Erwerb elementarer Fähigkeiten [Kaiser 96, Asada 91], andererseits Verfahren zur Programmierung umfangreicher Manipulationsaufgaben wie bei der Montage von einfachen mechanischen Produkten [Ikeuchi 94, Kuniyoshi 94, Friedrich 98]. Erstere befassen sich hauptsächlich mit der Generierung von Regelungsstrategien zur Ausführung kraftrückgekoppelter Bewegungen oder der Rekonstruktion von Trajektorien aus Sensordaten [Ude 96]. Oftmals werden zu diesem Zweck subsymbolische Repräsentationen und verschiedene Verfahren zur Funktionsapproximation wie z.B. neuronale Netze eingesetzt. Auf der Ebene der interaktiven Programmierung von Manipulationsaufgaben werden Elementarfähigkeiten zumeist vorausgesetzt. Der Schwerpunkt liegt dann in der Ableitung und Parametrierung einer Sequenz von Elementarfähigkeiten, die den durch den Benutzer ausgeführten Aktionen entspricht.

Es existieren Verfahren zum Programmieren durch Vormachen, die Vorführungen in der Virtuellen Realität nutzen, ebenso wie solche für physische Vorführungen. Gegenwärtig befassen sich einige Arbeiten mit der Frage, wie aus mehreren Vorführungen Schleifen oder Verzweigungen abgeleitet werden oder wie mehrere Roboter bzw. Mensch und Roboter sich zusammen eine Aufgabe kooperativ teilen können [Rybski 99, Zhang 99]. Eine umfassende Klassifizierung der Ansätze wird in [Dillmann 99] gegeben.

---

<sup>7</sup>engl.: Programming by Demonstration

## 1.3 Mensch-Roboter-Interaktion

Roboter müssen aus Sicherheitsgründen über Schnittstellen verfügen, über die programmierte Verhalten situationsbedingt aufgerufen werden können und über die deren Ausführung ausgelöst und überwacht werden kann. Insbesondere sollte bei Auftreten unerwarteter Ereignisse während des Programmablaufs bei Mangel von Verhaltensalternativen eine Rückfrage beim Benutzer möglich sein. Deshalb sind nicht nur Sensoren zur Überwachung des Ablaufs, sondern auch für die Mensch-Maschine-Interaktion zur Verfügung zu stellen. Zusätzlich müssen für die Mensch-Roboter-Kommunikation geeignete Interaktionsmechanismen vorhanden sein. Um teure Schulungen zu verkürzen, sollten diese einfach und schnell erlernbar sein.

In dem BMBF-Leitprojekt *Morpha* werden multimodale Schnittstellen entwickelt und erprobt, die dem Menschen einen natürlichen und intuitiven Zugriff auf die Fähigkeiten des Roboters gestatten [Bundesministerium für Bildung und Forschung 01]. Ziel ist hier die Identifikation eines Menschen als autorisierten Benutzer, die Erkennung dessen Intention und Aufmerksamkeit, also z.B. dessen Blickrichtung und Gestikulierung, sowie die Spracheingabe. Der Verbindung von Gesten und Sprache widmen sich spezielle Forschungsbemühungen, die erkannten gesprochenen Text und Handbewegungen zu fusionieren versuchen [Perzanowski 01].

Zu diesem Problemkomplex tragen nicht nur Sicherheitsaspekte, sondern auch Gestaltungsmerkmale des Roboters selbst bei. Menschen fällt die Interaktion mit technischen Systemen beispielsweise leichter, wenn Roboter über eine anthropomorphe Gestalt verfügen und sich auch menschenähnlich verhalten. Dies spielt bereits beim Entwurf solcher Systeme neben ergonomischen Betrachtungen eine große Rolle [Bischoff 98]. Der Roboterassistent soll auch multimodal antworten. Anthropomorph gestaltete Robotersysteme implizieren bzw. wecken bei dem Menschen Erwartungshaltungen, gleichberechtigten Systemen mit vergleichbaren Fähigkeiten gegenüberzustehen. An menschliche Gesichter angelehnte Roboterköpfe, mit visuellen und akustischen Gesichtssinnen ausgestattet, ermöglichen es, über deren Blickrichtung gleich den Aufmerksamkeitsfokus der Maschine abzuschätzen. Mehrere Arbeiten widmen sich dem Thema, Zustände des Roboters durch Gesichtsmimik direkt sichtbar zu machen [Breazeal 00].

## 1.4 Beitrag der Arbeit

Die Instruierung von Robotern besteht im Spezifizieren von Aufgaben und der sich anschließenden Interaktion zur Parametrierung und zur Auslösung von Aufträgen. Der Schlüssel zu einer einfachen und praxisgerechten Instruierung von Robotern wird in einer möglichst einfachen und intuitiven Schnittstelle zwischen Mensch und Maschine gesehen.

Unter den genannten Programmierverfahren erscheint das Programmieren durch Vormachen als eine vielversprechende Methode hinsichtlich ihrer Einfachheit und Intuitivität für den Benutzer. Die direkte Demonstration einer Aufgabe durch den Benutzer macht dieses Verfahren zu einem hervorstechenden Ansatz. Unter den genannten Möglichkeiten ist es jedoch

gleichzeitig das ambitionierteste, denn die Anforderungen an solch ein System zeigen sich an mehreren Stellen als sehr hoch:

- Die Vorführung von Handlungen muss sensorisch erfasst und in sinnvolle Abschnitte unterteilt werden. Dabei müssen einzelne Aktionen erkannt und Bewegungen verfolgt werden. Wichtig ist dabei eine genaue Lokalisierung der Hände.
- Aus der registrierten Vorführung müssen Schlüsse auf das Ziel einzelner Handlungssegmente gezogen werden, damit das Programm flexibel an unterschiedliche Situationen angepasst werden kann.
- Ein an eine bestimmte Szenensituation angepasstes Programm muss zur Ausführung adäquat auf die Kinematik des verwendeten Roboterarms und dessen Hand abgebildet werden.

Die vorliegende Arbeit leistet einen Beitrag auf dem Gebiet der Beobachtung menschlicher Handlungen im Rahmen der Instruierung von Robotern. Handlungen können dabei sowohl der Interaktion mit dem Gerät wie auch der Durchführung komplexer Manipulationen zum Zweck der Vorführung dienen. Die zur Verfolgung von Handlungen aufgebauten Aufnahmesysteme gestatten es, solche Handlungen unter sehr wenigen Einschränkungen vorzunehmen. Es handelt sich hierbei einerseits um ein fest installiertes Sensorsystem zur genauen Verfolgung manueller Manipulationen und andererseits um die auf einem Roboterassistenten angebrachte Sensorik zur Beobachtung von Handlungen im Rahmen der Mensch-Roboter-Kommunikation.

Die Arbeit ist zwischen der Registrierung einer menschlichen Handlung und deren Analyse durch ein System zur Interaktion oder zum Programmieren durch Vormachen angesiedelt. Ansätze zur Lösung der Probleme bei der Interpretation und Abbildung einer Vorführung finden sich an anderer Stelle [Friedrich 98, Rogalla 00]. Die Hauptthesen der Arbeit stellen sich wie folgt dar:

- Die Instruierung von Servicerobotern oder Assistenzsystemen ist auf Basis der Beobachtung von Benutzerhandlungen möglich. Es zeigt sich, dass diese zur Anweisung, Kommentierung und zum Vormachen einer Handlungsfolge dienen können. Die sensorielle Detektion aller drei Handlungen ist möglich.
- Die Anforderungen an ein Beobachtungssystem stellen sich je nach Zweck des Systems unterschiedlich dar. Insbesondere unterscheiden sich die zu verfolgenden Handlungsmerkmale bei Aktionen, die ausschließlich zur Interaktion beziehungsweise zur Programmierung komplexer Manipulationen gedacht sind. Es ist daher sinnvoll, dedizierte Sensorik in unterschiedlichen Umgebungen einzusetzen. In der vorliegenden Arbeit sind dies eine Vorführungsumgebung und eine Ausführungsumgebung.
- Die Anweisung eines solchen Systems kann auf einfache Weise und ohne aufwändigen Lernaufwand durch Gesten erfolgen.
- Die Erfassung einer komplexen Handlungsfolge durch das vorgeschlagene System ist als Basis für die weitere Verarbeitung im Rahmen des Programmierens durch Vormachen

geeignet; es lassen sich aus einer Benutzerdemonstration Lösungsstrategien ableiten, die auf einem realen Roboter ausführbar sind. Die Phase der Erfassung durchläuft dabei die Schritte:

- Aufbau des Weltmodells, d.h. Detektion der manipulierbaren Objekte.
- Erfassung einer Demonstration durch Fusion der Messungen mehrerer Sensortypen. Berücksichtigt werden dabei neben Verfahrbahnen Griffe und Transferbewegungen von Objekten.
- Möglichkeit zur Kommentierung.

Im Unterschied zu vergleichbaren Systemen erlaubt die gemeinsame Verwendung aktiver Sichtsysteme und Datenhandschuhe, Vorführungen bei wesentlich größerer Bewegungsfreiheit des Benutzers durchzuführen. Die Verarbeitungsgeschwindigkeit des Systems ist durch vielfache Optimierung ausreichend für die Bearbeitung natürlicher, fließender Bewegungen.

Zudem gestattet es die bereits während einer Demonstration ablaufende Segmentierung und Klassifikation elementarer Handlungen dem System, Rückfragen noch in der Vorführungsphase zu stellen. Das Aufbrechen des starren Programmierprozesses Vorführung–Interpretation–Korrektur–Ausführung mit sofortiger Korrekturmöglichkeit kann so mit die Programmierung erheblich beschleunigen.

Es soll dabei darauf hingewiesen werden, dass bestimmte Kontexte angenommen werden, in deren Rahmen die Handlungsbeobachtung abläuft. Ein solcher Kontext kann zum Beispiel im Haushaltsszenario das Einräumen einer Geschirrspülmaschine sein, das Decken eines Tisches oder Einräumen eines Kühlschranks. Derartige Kontextfestlegungen sind für eine robuste Verarbeitung notwendig.

## 1.5 Aufbau der Arbeit

Die vorliegende Arbeit ist in sieben Kapiteln organisiert. Ein kurzer Überblick über den Inhalt der einzelnen, sich anschließenden Abschnitte ist nachfolgend aufgeführt:

2. Der gegenwärtige Stand der Forschung auf diesem und relevanten ähnlichen Gebieten findet sich im zweiten Kapitel und wird dort auch diskutiert. Da die zu leistende Beobachtung von Benutzerhandlungen zum einen durch die verwendete Sensorik, zum anderen durch die angewandte Methodik konstituiert wird, wird zunächst auf die Entwicklungen auf dem Gebiet der Sensortechnik eingegangen. Anschließend finden sich grundlegende Algorithmen zur Behandlung von Sensormessungen im Hinblick auf Benutzervorführungen. Systeme zur qualitativen Handlungserkennung integrieren diese meist in speziell zugeschnittenen Varianten. Die Diskussion der Stärken und Schwächen dieser Systeme bildet den Abschluß des Kapitels.

3. Die vorgestellten Systeme stellen oft gewisse Randbedingungen an die Vorführung des Benutzers. Dadurch wird die Natürlichkeit der Interaktion mit einem System eingeschränkt. Auf die Vorstellung der benutzerbeobachtenden Systeme folgen daher in Kapitel drei grundlegende Gedanken zur Beobachtung menschlicher Vorführungen. Ausgehend von Demonstrationen zwischen zwei Personen werden typische Handlungselemente identifiziert.
4. Das vierte Kapitel stellt dann den gesamten Vorgang der Interpretation beobachteter Handlungsfolgen vor. Anschließend wird die Sensorik ausgewählt, mit Hilfe derer das vorgeschlagene System zur Handlungsbeobachtung Aktionen erkennt und klassifiziert. Dabei ist die Erkennung von Aktionen in zwei separaten Umgebungen möglich, einmal in einer speziellen Vorführungsumgebung zur hochgenauen Verfolgung von Verfahrbahnen und Griffklassifikation und einmal in der Ausführungsumgebung vor dem Roboterassistenten *Albert* [Ehrenmann 01b]. Neben der Sensorik wird auch die kinematische und geometrische Modellierung des Benutzers sowie seiner manipulierbaren Umwelt erläutert, die in beiden Umgebungen zum Einsatz kommt. Einen wichtigen Aspekt bildet hier die Modellierung der erkennbaren Aktionen, die gesondert behandelt wird.
5. Die Vorstellung der einzelnen Methoden zur Erkennung und Registrierung beobachteter Aktionen, sogenannter *elementarer kognitiver Operatoren*, ist Gegenstand des fünften Kapitels. Hier werden die Methoden zur Szenenanalyse, Handverfolgung, Griff- und Gestenerkennung im Kontext von Serviceroboteranwendungen vorgestellt, die in den genannten Umgebungen zum Einsatz kommen.
6. Die Validierung der im vorhergegangenen fünften Kapitel vorgeschlagenen *elementaren kognitiven Operatoren* erfolgt durch umfassende Analysen und Testläufe. Dafür dient das sechste Kapitel. Es wird hier ebenfalls ein Beispiel für die Beobachtung von komplexen Handlungsfolgen gegeben, die mit Hilfe dieser Operatoren Handhabungen auf eine Repräsentation im Weltmodell abbildet. Diese wiederum wird zur Reproduktion der Ausführung auf dem Robotersystem *Albert* genutzt.
7. Das letzte Kapitel dient der Zusammenfassung der Arbeit und einem Ausblick auf zukünftige Entwicklungen.



# Kapitel 2

## Stand der Forschung

Eine intuitive Schnittstelle zur Programmierung oder Anweisung eines Roboters auf Aufgabenebene ist Gegenstand intensiver Forschungsbemühungen. Viele Veröffentlichungen werden mit einer Kritik an dem bis heute vorherrschenden textbasierten Programmierparadigma eingeleitet. Im Folgenden soll aufgezeigt werden, welche Fortschritte bisher gemacht wurden, um Menschen einen direkteren und intuitiveren Zugang zur Nutzung von Servicerobotern zu geben.

### 2.1 Skopus und Randbedingungen

Ein Blick über das Spektrum der Arbeiten zu der Mensch-Roboter-Interaktion zeigt, dass zum größten Teil die Implementierung einzelner Aspekte der Beobachtungsfähigkeiten von Robotern bezüglich der Interaktion des Menschen untersucht wurde. Dies betrifft vor allem die Erkennung von Befehlen im Rahmen eines vorgegebenen Konzeptes oder Systems. Multimodale Schnittstellen zu einem Assistenzroboter gewinnen erst seit jüngster Zeit mehr Aufmerksamkeit. Von ergonomischen Untersuchungen oder den Bestrebungen, Kommunikationsmodalitäten quasi zu normieren wie dies z.B. bei fenstergestützten Betriebssystemen der Fall ist, ist jedoch noch nicht die Rede. Die Kenntnis der Bedienung eines Systems läßt sich deshalb nicht auf die Bedienführung eines anderen übertragen.

In diesem Kapitel wird ein Überblick bezüglich des gegenwärtigen Standes der Technik bei der Handlungserkennung gegeben, soweit sie zur Programmierung oder Dialogführung im Rahmen von Handhabungsaufgaben Verwendung findet. Da sich die folgenden Kapitel in dem genannten Kontext auf Transportaufgaben konzentrieren, stehen diese im Vordergrund. Dabei sollen ausschließlich graphische<sup>1</sup> oder physische Manipulationsvorführungen eines Benutzers betrachtet werden, allerdings keine symbolischen<sup>2</sup>. Zur Roboterprogrammierung wurden zahlreiche Verfahren des manuellen Programmierens erweitert, indem spezielle Vorführgeräte (z.B. ein mit Markierungen überzogenes Objekt in [Tso 95]) oder unmittelbar

---

<sup>1</sup>als graphische Vorführung gelten Demonstrationen, bei denen der Benutzer eine Manipulationsaufgabe durch direkte Manipulation von 3D-Objektmodellen in einer graphischen Simulation löst

<sup>2</sup>als symbolische Vorführung werden Demonstrationen bezeichnet, bei denen der Benutzer eine Aufgabe durch Aneinanderreihung von Operatorsymbolen spezifiziert

der Roboter manipulator direkt durch den Benutzer (siehe Abschnitt 1.2.3) geführt werden. Aus charakteristischen Messergebnissen z.B. von Kraftmessdosen und der internen Sensorik der Roboter werden daraufhin Elementaraktionen erkannt und entsprechende Operatoren in das zu erstellende Programm eingefügt. Solche indirekt ausgerichtete Vorführungen werden im Rahmen dieser Arbeit nicht behandelt.

## 2.2 Sensorentwicklung

Da die Wahl und die Anwendung der zur Handlungsbeobachtung eingesetzten Sensorik für die zu beschreibenden Verfahren einen wesentlichen Aspekt bildet, wird zunächst ein kurzer Überblick über die wichtigsten Sensoren gegeben. Danach werden die einzelnen Aspekte der visuellen Handlungsbeobachtung wie Szenenanalyse, Objektverfolgung und Gestenerkennung genauer betrachtet.

### 2.2.1 Bildgebende Sensoren

Bildgebende Sensoren werden in der Regel zur Beobachtung von Objekten, Szenen oder Handlungen verwendet. Der Einsatz von Kameras hat den Vorteil, dass neben der eigentlichen Aufnahme und Analyse einer Handlungsfolge auch die Modellierung der Umwelt auf der Basis von Verfahren für die Objektmodellierung, -erkennung und für die Bestimmung von Objektlagen möglich ist.

Viele Verfahren zur Bewegungsverfolgung basieren auf stationären Kamerasystemen. Dies schränkt zwar den Beobachtungsbereich ein, vereinfacht aber die Kamerakalibrierung und erlaubt lokal eine hohe Pixelauflösung. Um den Blickwinkel von Kamerasystemen flexibler auf örtlich verteilte Objekte einstellen zu können, wurden bereits in den siebziger Jahren monokulare Bildsensoren am Effektor eines Manipulatorarms befestigt, um die Kameras mit Hilfe des Roboters günstig zu den zu beobachtenden Objekten positionieren zu können [Shirai 73]. Aktuelle Arbeiten finden sich auch in der jüngeren Literatur, z.B. in [Marchand 98].

Die Regelung zur visuellen Echtzeitverfolgung<sup>3</sup> eines Objektes hat erst in jüngerer Zeit große Fortschritte gemacht [Corke 00]. Vor zehn Jahren war als Zykluszeit zwischen Objekt-lagenschätzung und Fokussierungsbewegung fast eine Sekunde erforderlich [Miura 92], heute kann die Verfolgung von Objekten seitens der Hardware mit fließenden Bewegungen durch hohe Abstraten in Echtzeit erfolgen [Braten 99]. Probleme bei Echtzeitanforderungen bereiten nach wie vor die relativ hohen Totzeiten, die durch Bildverarbeitung, inverse Kinematik und Servoansteuerung entstehen [Vincze 00].

Die meisten heutigen Bildverarbeitungssysteme verwenden zwei Grauwert- oder Farbkameras, die mit zwei Freiheitsgraden bewegt werden können (kippen- und rotierbar<sup>4</sup>).

---

<sup>3</sup>engl.: Visual Servoing

<sup>4</sup>engl.: Turn and Tilt Units

[Bernardino 98] schlägt ein System zur Handverfolgung vor, das zwei biologisch motivierte Verhalten verwendet, nämlich Vergenz- und Fokussierungsbewegungen. Die Bilder eines binokularen Kamerakopfes werden zur Verarbeitung zunächst in logarithmische Polardarstellung<sup>5</sup> transformiert. In dieser wird die Hand des Menschen mit Disparitätsberechnungen lokalisiert. Die Vergenzmotorsteuerung hält die Hand im Zentrum beider Kameras. Solche Fokussierbewegungen eines Kamerakopfes auf Basis der Berechnung des optischen Flusses in Logmapbildern finden sich auch bei [Baratoff 99], eine weitere disparitätsbasierte Verfolgung mit schnellen Blickwechselln (Sakkadenbewegungen) bei einer Verzögerung von 150ms in [Uhlin 95].

[Peixoto 00] benutzt zur Beobachtung und Verfolgung von Personen in einem Raum eine Kombination aus einer statischen Deckenkamera und einem aktiven, dreh- und schwenkbaren Kamerakopf zur genaueren Benutzerverfolgung. Die Erkennung und Verfolgung von Personen geschieht über Differenzbildansätze auf der statischen Kamera. Der Kamerakopf kann eine Einzelperson fixieren und durch Sakkaden verfolgen. Menschliche Handlungen durch ein System von fixierten Deckenkameras zu verfolgen ist auch das Ziel von [Mori 97]. Mit Differenzbildansätzen und Schablonenanpassung wird hier die Intention von Bewegungen abgeschätzt: Umkehr, Annäherung an bestimmte Einrichtungen im Raum oder zielgerichtetes Laufen. Ein System, das in ähnlicher Weise die Messungen von Deckenkameras und Sensormessungen von mobilen Robotern fusioniert, um Bewegungen von Menschen im Kontext einer Kollisionsvermeidung für mobile Roboter als Hindernisse in der Roboternavigation vorherzusagen, wird in [Steinhaus 99] vorgestellt.

Allgemein lässt sich feststellen, dass die Verwendung bildgebender Sensoren zur Beobachtung des Bewegungsverhaltens von Menschen einen hohen Berechnungsaufwand erfordert. Bei der Entwicklung geeigneter Bildverarbeitungsalgorithmen stellen sich noch eine Vielzahl ungelöster Probleme z.B. im Bereich der Objekterkennung, der Bildfolgenanalyse und der Szeneninterpretation. Außerdem stellen Verdeckungen, Reflexionen und Beleuchtungsvarianzen große Schwierigkeiten dar, die bislang nicht zufriedenstellend gelöst werden konnten.

### 2.2.2 Magnetfeldbasierte Positionssensoren

Bei magnetfeldbasierten Positionssensoren tritt im Unterschied zu bildverarbeitenden Verfahren das Problem der Verdeckung oder variierender Lichtverhältnisse nicht auf. Lage und Position der Benutzerhand lassen sich direkt aus Sensormessungen ableiten. Die quadratische Abnahme der Magnetfeldstärke mit zunehmender Entfernung vom Magnetfeldemitter wie auch Störungen im Empfangsbereich durch metallische Gegenstände, Monitore, Netzgeräte und weitere Erzeuger elektrischer Felder schränken Arbeitsbereich und Genauigkeit jedoch stark ein [Stasch 97]. Ein weiteres Negativum dieser Meßtechnik ist das mitzuführende Kabel, das den Sensor mit einer Schnittstelle verbindet.

---

<sup>5</sup>engl.: Logmap



### 2.2.3 Datenhandschuhe

Ein Eingabemedium, das ursprünglich für Benutzer von Systemen im Bereich virtueller Realitäten entwickelt wurde, ist der Datenhandschuh (siehe Abbildung 2.1). Hier wird die Beugung und Spreizung einzelner Fingergelenke durch Lichtleiter oder Dehnmessstreifen gemessen, die in einen Handschuh eingearbeitet sind [Virtex 00, Mindflux 00]. Unabhängig von der Lage der Hand kann direkt Auskunft über die Fingerstellung gegeben werden. Deshalb haben Datenhandschuhe eine weite Verbreitung gefunden. Eine Zusammenstellung von Geräten und Anwendungsfeldern findet sich bei [Sturman 94].

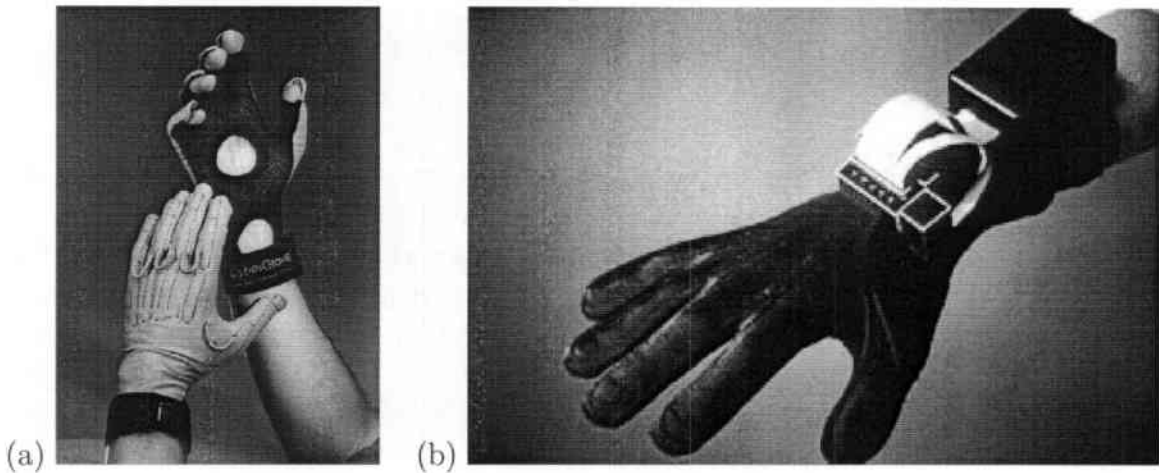


Abbildung 2.1: Kommerzielle Datenhandschuhe: der *Cyberglove* der Firma Virtual Technologies aus [Virtex 00] (a) und der *Data glove* von Mindflux aus [Mindflux 00] (b)

Probleme ergeben sich bei diesen Systemen aus den anatomischen Abweichungen der Benutzer. Aufgrund unterschiedlicher Finger- und Handabmessungen liegen die Sensoren sogar bei jedem Anziehen des Handschuhs an unterschiedlichen Positionen und liefern abweichende Messwerte. Dies schränkt die Genauigkeit bei der Verwendung dieses Sensortyps ein. Bei Griffdetektionen durch Kontaktberechnungen mit geometrischen Modellen müssen die Messungen daher aufwändig vorkalibriert werden. Wie die magnetfeldbasierten Positionssensoren sind diese Geräte üblicherweise durch Kabel mit einer Schnittstelle verbunden.

Meist werden Datenhandschuhe in Kombination mit magnetfeldbasierten Positionssensoren oder Gyroskopen verwendet, um außer der Fingerkonfiguration auch die Lage der Hand nutzen zu können.

### 2.2.4 Datenanzüge

In den letzten Jahren sind aufbauend auf Datenhandschuhen eine Vielzahl kommerzieller Systeme vorgestellt worden, die versuchen, menschliche Bewegungen zu verfolgen, um sie auf die kinematische Struktur von Avataren zu übertragen. Diese agieren dann als künstliche Dialogpartner oder in Filmen.

Die Sensorik kann hier auf Magnettrackern basieren (siehe [SimGraphics 01, Ascension 01, Polhemus 02] und Abbildung 2.2), die einzeln an allen artikulierten Extremitäten fixiert

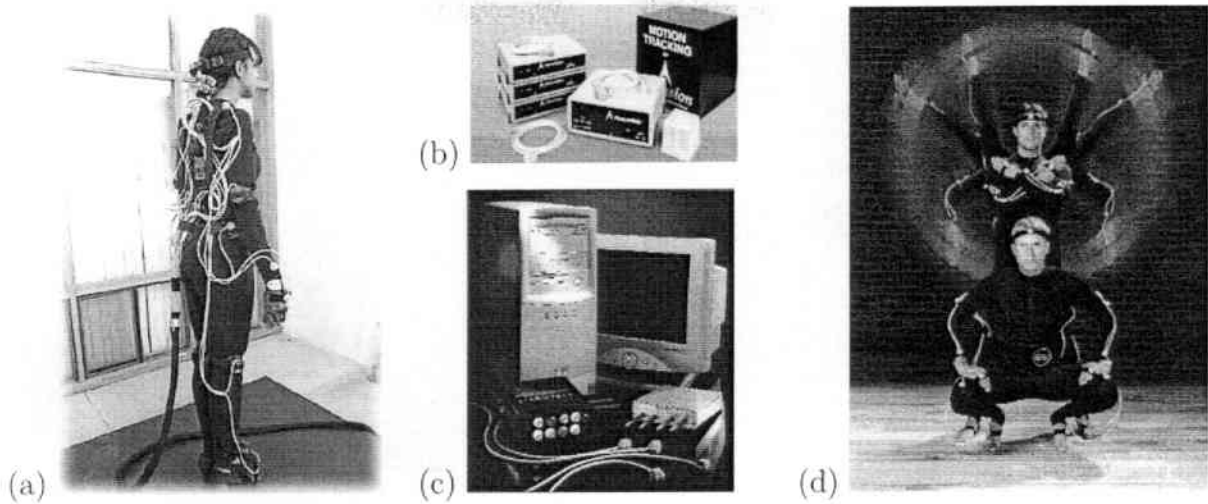


Abbildung 2.2: Magnetische Trackingsysteme für den ganzen Körper von SimGraphics (a, [SimGraphics 01] entnommen), Ascension (b, aus [Ascension 01]) und Polhemus (c und d, aus [Polhemus 02])

werden. Für größere Szenen werden vorkalibrierte Multikamerasysteme angeboten, die ebenfalls fixierte optische Marker bzw. Leuchtdioden auf dem Körper verfolgen (siehe hierzu [MOTEK 01] bzw. [Phoenix Technologies 01] und Abbildung 2.3).

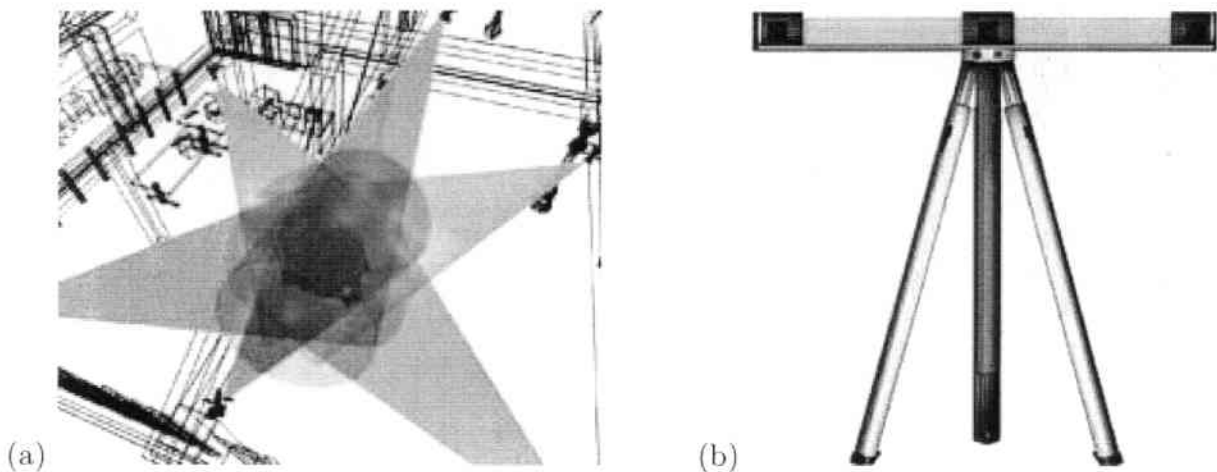


Abbildung 2.3: Bildbasierte Trackingsysteme für den ganzen Körper von Motek (a, aus [MOTEK 01]) und von Phoenix (b, Quelle: [Phoenix Technologies 01])

Alle kommerziellen Systeme sind passive Beobachtungssysteme. Ihre Verwendung ist jedoch durch hohe Kosten (150–400 T Euro) und den hohen Aufwand beim Anlegen und Kalibrieren des Anzugs eingeschränkt.

### 2.2.5 Exoskelette

Um trotz Magnetfeldstörungen durch metallische Gegenstände in grossen Bewegungsräumen präzise messen zu können, verwenden manche Anzüge zur Messung der Beugung der Körpergelenke Dehnungsmessstreifen. Mit Hilfe eines kinematischen Modells können dann Aussagen

über die Positionen einzelner Extremitäten getroffen werden (siehe [MetaMotion 01] sowie Abbildung 2.4 a).

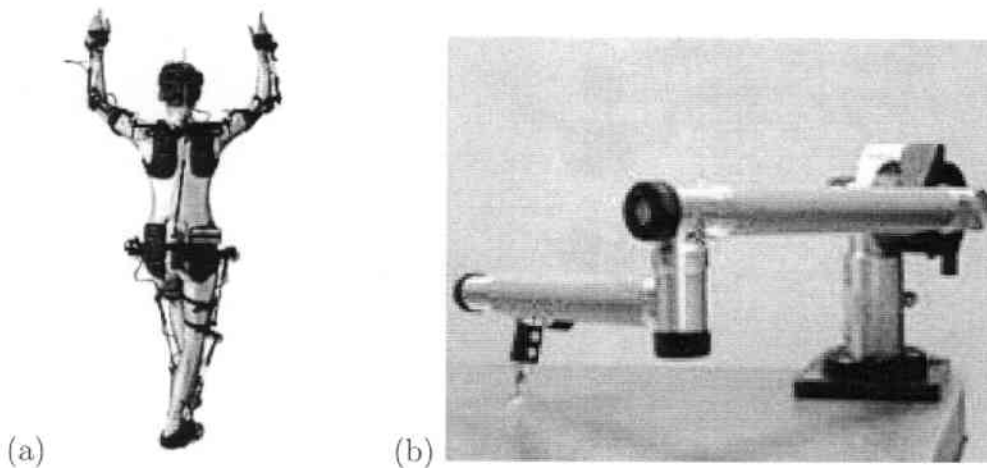


Abbildung 2.4: Datenanzug mit Beugungswinkelmessung von Meta Motion (a, Quelle: [MetaMotion 01]) und Faro-Arm zur präzisen Lagebestimmung (b, aus [Faro 01])

Zur hochpräzisen Lagebestimmung werden auch Systeme eingesetzt, die Manipulatoren gleichen. Ein Beispiel hierfür ist der Faro-Arm (siehe Abbildung 2.4 b). Interne Temperaturkompensation und optische Drehwinkelencoder gestatten Positionsmessung im Zehntelmillimeterbereich [Faro 01]. Neben der Positionsbestimmung erlaubt das *Phantom* die gleichzeitige Messung ausgeübter Kräfte auf den Endsensor sowie Krafrückkopplung [Sensable 01].

## 2.3 Spezielle Aspekte der Handlungsbeobachtung

Mehrere sensorische Teilaspekte spielen bei der Handlungsbeobachtung eine Rolle. Da die unten in Abschnitt 2.4 vorgestellten Verfahren diese Aspekte zum großen Teil mitbehandeln, sie aber nicht immer optimal lösen, wird an dieser Stelle zunächst eine umfassende und genaue Betrachtung vorangestellt. Drei Themenkomplexe lassen sich hier unterscheiden:

**Szenenanalyse:** Vor beabsichtigten Eingriffen in die Umgebung bzw. vor der Beobachtung einer solchen muss sich ein System Information über seine Umwelt verschaffen. Meist wird hierzu ausschließlich der manipulierbare Teil der Umgebung untersucht. Dies schafft einen Kontext, in dem viele Handlungen des beobachteten Agenten erst interpretierbar werden. Im Folgenden werden nur kamerabasierte Ansätze zur Lösung dieses Problems behandelt.

**Objektverfolgung:** Eine weitere grundlegende Voraussetzung für die Handlungserkennung ist die Fähigkeit, Objekte zu verfolgen und während ihrer Bewegung zu lokalisieren.

**Gestenerkennung:** Ein Modal zur Programmierung oder Dialogführung sind Gesten. Griffe werden als Spezialfall von Gesten betrachtet.

In den folgenden Abschnitten wird ein Einblick in Algorithmen gegeben, die diesen Themengebieten gewidmet sind.

### 2.3.1 Szenenanalyse

Systeme, die in der Robotik Szenen analysieren, integrieren Verfahren zur Kamerakalibrierung, zur Bildaufnahme und zur Detektion relevanter Information. Letzteres dient der Objekterkennung und Objektlokalisierung für die Erstellung eines Weltmodells bzw. zur Greifplanung [Schmidt 98]. Während die Bildaufnahme hier nicht gesondert betrachtet werden soll, widmet sich der folgende Abschnitt den anderen beiden Aspekten:

**Kamerakalibrierung:** Die Kenntnis äußerer und innerer optischer Kameraparameter bildet die Grundlage für die Rekonstruktion der räumlichen Lage von Bildpunkten. Zur Bestimmung der Objektlage werden meist schwach oder stark kalibrierte bi- oder trinokulare Sichtsysteme eingesetzt [Beyer 96, Dornaika 97]. Eine lineare Projektion von  $3D$ -Raumkoordinaten auf die  $2D$ -Bildkoordinaten, wie sie beispielsweise in [Faugeras 93] verwendet wird, ist aufgrund von Linsenverzerrungen bei kurzen Brennweiten nicht zur genauen Lokalisierung verwendbar. Deshalb wurden Kalibriertechniken vorgeschlagen, welche radiale Verzerrungen durch Kameralinsen [Tsai 87, Wilson 94] oder einen Bias für Kontrollpunkte [Heikkilä 00] mit berücksichtigen. Die Kalibrierung selbst kann grundsätzlich nach einem photogrammetrischen Verfahren oder einem Selbstkalibrierungsverfahren erfolgen [Zhang 00]:

**Referenzobjektkalibrierung:** Hier wird das Kamerabild eines Kalibrierobjekts mit dessen genau vermessener Raungeometrie in Beziehung gesetzt [Weckesser 97]. Meist werden hierzu planare oder orthogonal zueinander stehende planare Objekte verwendet.

**Selbstkalibrierung:** Diese Kategorie kommt ohne Kalibrierobjekt aus. Genutzt werden mehrere, hintereinander aufgenommene Bilder einer statischen Szene. Werden die internen Kameraparameter wie die Brennweite nicht verändert, reichen drei Bilder eines Referenzobjektes aus [Luong 97].

Zur Verbesserung der Genauigkeit werden dazu meist Merkmale oder Marker verwendet, deren Position subpixelgenau bestimmbar ist.

**Objekterkennung:** Objekterkennungsverfahren dienen zum einen zur Initialisierung eines Weltmodells, in dem ein menschlicher Benutzer oder Roboter agiert oder zum anderen zur Initialisierung von Objektverfolgungsalgorithmen.

Hierzu werden mit Erfolg ansichtsbasierte Verfahren unter Verwendung von Schablonen-Techniken, Silhouetten- bzw. Kontur- oder Eigenfenstermethoden [Brunelli 95, Gonzalez-Linares 99, Bandlow 98, Pauli 98, Steinhaus 97] eingesetzt. Letztere sind noch immer Gegenstand der Forschungsbemühungen. Erweiterungen sollen vor allem ermöglichen, teilverdeckte Objekte zu klassifizieren [D. Huttenlocher 99]. Robustheit gegenüber Verdeckungen oder Rauschen lässt sich bei konturbasierten Klassifikatoren wie der allgemeinen Hough-Transformation [Hough 62, Duda 72, Ballard 81, Gonzalez-Linares 99] oder elastischen Graphen (aktiven Konturen, Schlangenmodellen) verzeichnen [Kefalea 97, Terzopoulos 88]. Hier wird lediglich in der Umgebung von in einem Modell gegebenen Stützpunkten

nach wählbaren Bildmerkmalen gesucht. Statt Ansichtsmodellen werden in manchen Ansätzen auch nur Teilansichten verwendet, wenn sie einen ausreichenden Informationsgehalt tragen [Büker 99, Theis 01]. Andere Arbeiten analysieren im Bild erkannte einzelne Merkmale, die zu komplexeren Strukturen zusammengesetzt und dann mit Modellen, beispielsweise mit aus CAD-Modellen generierten Ansichten, verglichen werden [Ude 94, Kim 99].

Mehr und mehr werden die genannten Methoden gemeinsam oder von hierarchischen Klassifikatoren genutzt [Kestler 99, Ehrenmann 00]. Gleichzeitig sind jedoch häufig mit Markern versehene künstliche Manipulationsobjekte in Benutzung [Tanaka 00]. Aufgrund der immer noch nicht zufriedenstellenden Lösungen der Forschungsbemühungen über die letzten Jahrzehnte insbesondere bezüglich Verkippungen im Raum, Vergrößerungen und des Rechenaufwands orientieren sich manche Arbeiten an der neurobiologischen Forschung und experimentieren mit dem Einsatz künstlicher neuronaler Netze<sup>6</sup> [Poggio 93, Elsen 98].

### 2.3.2 Objektverfolgung

Nach dem Scheitern der von Marr konzipierten *Theorie des rechnenden Sehens*<sup>7</sup>, in der Sehen als rein informationsverarbeitender Rekonstruktionsprozeß aufgefaßt wird [Marr 82], etablieren sich seit etwa einer Dekade alternative Paradigmen des Maschinensehens.

Aspekte unterschiedlicher Beleuchtung, unvollständigen Modellwissens und der Aufmerksamkeit sollen hier mitmodelliert werden. Aloimonos hat dazu den aktiven Beobachter eingeführt, der eigenständig die Parameter seines okulomotorischen Systems zu ändern in der Lage ist und damit den Begriff des *aktiven Sehens*<sup>8</sup> geprägt [Aloimonos 93]. Diese Ansicht wird vor allem durch neurobiologische und kognitionspsychologische Untersuchungen gestützt [Brockmann 99]. Von Bajcsy wurde die Betonung auf eine aufgabenspezifische Modellierung und Steuerung gelegt [Bajcsy 92], während Ballard mit dem Terminus des *anregenden Sehens*<sup>9</sup> auf das Zusammenspiel von Sehen und Handeln aufmerksam machte [Ballard 92].

Unter dem Begriff des aktiven Sehens sind heute all diese Aspekte bei der Detektion und Verfolgung von Objekten subsumiert. Eine Vielzahl von Forschungsarbeiten zur Unterstützung von Sakkaden und Fixationen interessanter Regionen oder Merkmale ist auf der Seite der Hardware geleistet worden. Auf Seite der Software wurden neue Algorithmen zur Regelung der Echtzeitverfolgung eines Objektes und Segmentierung entwickelt. Ein Spezialgebiet ist dabei die Verfolgung menschlicher Arme und Hände. Die einzelnen Themen werden im Folgenden diskutiert:

---

<sup>6</sup>engl.: Artificial Neural Networks

<sup>7</sup>engl.: Computational Theory of Vision

<sup>8</sup>engl.: Active Vision

<sup>9</sup>engl.: Animate Vision



**Regelung zur Echtzeitverfolgung:** Zur Regelung der Sensormotorik sind viele verschiedene Ansätze bekannt. Meist wird hierbei auf Prädiktionen der Objektbewegungen z.B. auf der Basis von Kalman-Filtern zurückgegriffen [Rao 96, Yeasin 00]. Modellwissen kann auch genutzt werden, um hochgenaue Posenschätzungen aus Kamerabildern abzuleiten. In [Ruf 97, Tonko 97] wird ein Schätzverfahren zur Verfolgung eines Endeffektors vorgestellt.

Ziel der motorischen Regelung ist die Fixierung bestimmter Merkmale in Kamerabildern. In den letzten Jahren wurde hierzu vor allem die Modellierung von Hautfarbe oder Konturmodellen als Interessensgebiet untersucht. Die Verfolgung eines Menschen auf Basis von Hautfarbsegmentierung wird in [Sidenbladh 99] beschrieben. Ein dreh- und schwenkbarer Kamerakopf sucht dazu in farbgefilterten Kamerabildern nach Regionen von Hautfarbpixeln einer gewissen Grösse und regelt den Kamerakopf zur Zentrierung dieser Regionen nach. Die Hautfarbsegmentierung selbst wie auch die Verfolgung von Konturmerkmalen hat sich als eigenes Wissensgebiet entwickelt.

**Hautfarbsegmentierung:** Ansätze zur Hautfarbsegmentierung werden seit dem Beginn der neunziger Jahre verfolgt. In diesem Zeitraum standen die ersten Framegrabber für Farbsignalverarbeitung kostengünstig zur Verfügung. Es ist festzustellen, dass Hautfarbe im Farbraum sehr eng verteilt und deshalb gut zu segmentieren ist [Garcia 99]<sup>10</sup>. Allerdings ändert sich die Spektralzusammensetzung sehr stark mit Änderungen des Lichteinfalls. Einfache Algorithmen modellieren die Hautfarbe als gaussverteilte Größe im *RGB*-Farbraum [Xu 98]. Leistungsfähigere Methoden lassen sich einteilen in adaptive Methoden und Methoden zur Farbkonstanzerstellung:

**Adaptive Hautfarbsegmentierung:** Einer theoretisch fundierteren Modellierung von Hautfarbe und ihrer Adaption an schwankende Lichtverhältnisse widmen sich die Arbeiten von [Yang 98, Sigal 00]. Hier werden Algorithmen vorgestellt, die Schwellwerte im *HSY*-Farbraum an die Gegebenheiten anpassen. In [Avrithis 00, Park 00] werden darauf basierende spezielle Segmentierungs- oder Region-Growing-Methoden vorgestellt, die an die Gesichtsdetektion mit Hilfe zusätzlichen Hintergrundwissens angepasst sind. In [Terrillon 98] wird für denselben Zweck ein etwas abweichender Farbraum definiert und Fourier-Mellin-Momente zum Verfolgen der Gesichtskontur eingesetzt.

In [Shiga 00] wird zusätzlich zur Farbinformation Bewegung aus einer Differenzbildberechnung in die Segmentierung einbezogen. Ohne den Einsatz von Hintergrundwissen oder höherer Verarbeitungsstufen kann auch das Verfolgen von Händen nicht erfolgen, da sie oft mit Gesichtsregionen in den Kamerabildern verschmelzen [Imagawa 98].

**Farbkonstanzalgorithmen:** Nach dem Vorbild der menschlichen Wahrnehmung wird hier versucht, nicht nur eine bestimmte Farbe aufgrund einer Kalibrierung zu verfolgen, sondern die direkte Abbildung von Farben zu modellieren. Dabei können gleichzeitig viele Farben berücksichtigt werden. Interessante Ansätze sind

<sup>10</sup>Zur Repräsentation von Farben in Farbräumen siehe Anhang C

hier das Retinex-Verfahren, das die Existenz von Flächen mit bekanntem Reflexionsverhalten zur Prämisse nimmt sowie das Gamut-Mapping, bei dem von einer konstanten konvexen Hülle im Farbraum einer Szene ausgegangen wird. Bei ersterem wird ein globales Reflexionsmaximum für jede interessante Farbe gesucht, auf das sich im Folgenden Deskriptoren zur Farbqualitätsbeschreibung beziehen. Diese werden durch Verhältnisse betrachteter Bildpunkte zu ihrer Umgebung lokal berechnet [Land 86, Jobson 95]. Beim Gamut Mapping wird versucht, die Abbildung zu rekonstruieren, die zu einer Farbveränderung geführt hat. Voraussetzung ist hier die Bekanntheit der Farbigekeit eines Gegenstandes der Szene und das Vorhandensein von nicht mehr als einer Lichtquelle. Für die Suche nach geeigneten Abbildungen werden die Mengen der möglichen Abbildungen jedes Hüllpunktes des aktuellen Bildes in diese kanonische Hülle ermittelt und geschnitten [Finlayson 96, Barnard 95].

**Konturmodelle:** Die Objektverfolgung mit Hilfe von konturbasierten Methoden hat in den neunziger Jahren vor allem aufgrund der Reduktion von Rechenzeiten viel Resonanz erfahren. In der dafür grundlegenden Arbeit wurden Konturen als Splines („Schlangenmodelle“<sup>11</sup>) repräsentiert, deren Anpassung an das Bild durch Energieminimierung erreicht werden soll [Kass 88]. Die Energiefunktion beschreibt einerseits die Anpassung an geeignete Bildmerkmale, andererseits die Stärke der Konturverformung. Mit solchen Verfahren werden auch dreidimensionale Kantenmodelle in die Ebene projiziert und direkt mit den Bildgradienten verglichen [Kollnig 95]. In einem anderen Ansatz werden feste Kantenmodelle mit Hilfe linearer Transformationen an das Kamerabild angepasst [Blake 98a]. Hier wird die optimale Transformation gesucht, welche das Modell möglichst passend auf die in Konturnähe gefundenen Merkmale einpasst. Erweiterungen arbeiten mit stochastisch modellierten Konturen oder Punktmodellen, welche in der Lage sind, auch Objekte mit Translations- oder Rotationsgelenken zu verfolgen [Heap 97]. In [Wachter 97] werden spezielle Ellipsen verwendet, um Menschen mit bewegten Extremitäten zu verfolgen. Aufgrund des hohen Aufwands bei der Modellierung und Initialisierung von konturbasierten Methoden gibt es zu diesem speziellen Thema ebenso Forschungsarbeiten [Lai 94].

Viele Algorithmen beruhen auf dedizierten Modellen und Ansätzen. Für die vorliegende Arbeit sind spezielle Verfahren zum Verfolgen von Armen oder Händen besonders relevant. Sie sollen separat behandelt werden:

**Handverfolgung:** Ein spezielles und sehr herausforderndes Problem ist das bildbasierte Verfolgen der menschlichen Hand bzw. des menschlichen Arms. Durch die Vielzahl von Gelenken und die komplexe Struktur ist nicht nur ein hoher Grad an Artikulationsmöglichkeiten gegeben, sondern auch von Verdeckungen. Dadurch wird eine genaue Zuordnung der einzelnen Finger erschwert. Ansätze zum Handtracking basieren auf Projektionen der Silhouette [Rehg 95, Gavril 95, Leibe 01], Bewegungsinformationen [Yamamoto 91] oder Merkmalen wie Kanten und Farbmarkern [Yeasin 00].

---

<sup>11</sup>engl.: Snakes

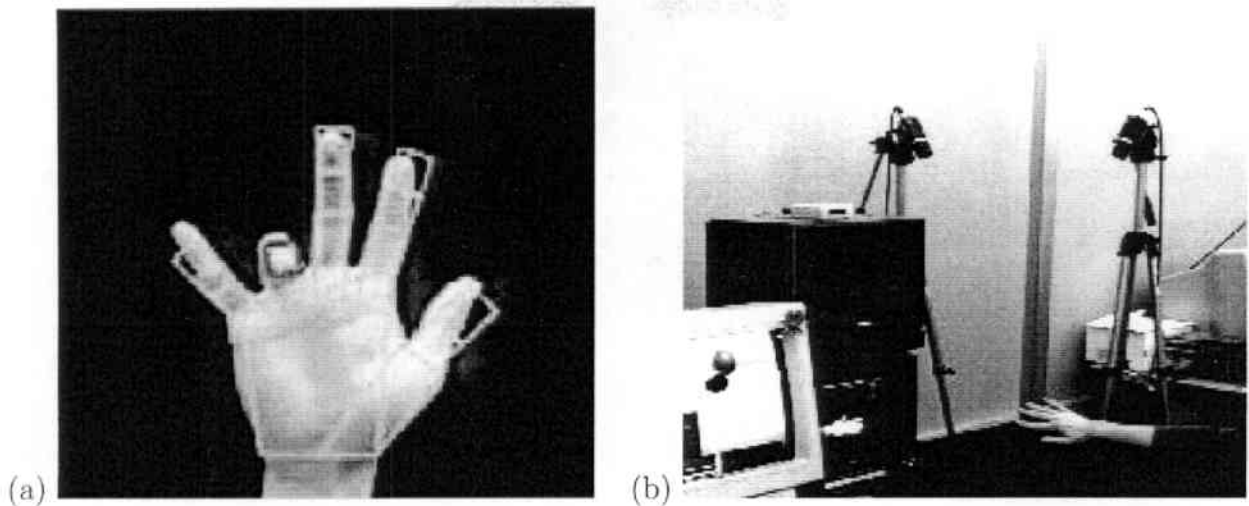


Abbildung 2.5: Handverfolgung nach Wu (a, aus [Wu 01]) und Aufbau des Systems *DigitEyes* mit Stereoverfolgung nach Rehg (b, aus [Rehg 94])

In einigen Applikationen wird eine vollständige 3D-Rekonstruktion mit aufwändigen Handmodellen versucht. An diesen Modellen lassen sich Beugungswinkel direkt ablesen. In [Rehg 93, Rehg 94] wird die Hand so durch Zylinder mit insgesamt 27 Freiheitsgraden approximiert, deren Lagen geschätzt werden. Ein ähnliches Vorgehen mit 21 Freiheitsgraden erfolgt bei Wu ([Wu 01], Ausschnitte beider Ansätze finden sich in Abbildung 2.5). Die Winkelschätzungen für die geometrischen Handmodelle erfolgen entweder über erkannte Bildmerkmale oder durch Anpassung des projizierten Modells auf das Kamerabild selbst. Dabei werden Bewegungseinschränkungen der Handkinematik genutzt, um Mehrdeutigkeiten auszuschließen [Lin 00]. Aufgrund der Selbstverdeckungen der Hand und Schattenwürfen ist dieser Ansatz trotzdem nur als beschränkt einsatzfähig zu bewerten. Dies gilt auch für Handverfolgungsalgorithmen, die direkt auf Tiefenbildern operieren. So versucht der 3D-Schablonenalgorithmus<sup>12</sup> eine Musteranpassung von Oberflächenmodellen direkt in 3D-Bildern [Jiar 96a]. Aufgrund der dabei anfallenden hohen Problemkomplexität sind diese Methoden bislang nicht unter Echtzeitbedingungen eingesetzt worden. Selbst wenn aufgezeichnete Bildfolgen verwendet werden, wird bei fest gewählten Beobachtungsbedingungen die Betrachtung der Finger zur Erhöhung der Robustheit oft auf einen kleinen Ausschnitt reduziert [Jiar 96b]. Tabelle 2.1 gibt einen kurzen Überblick über die prominentesten Arbeiten mit dreidimensionalen Modellen. Neben dem Autor sind die verfolgten Extremitäten, deren Freiheitsgrade im Modell, der Modelltyp und die verwendete Anpassungsmethode des Modells auf das Bild aufgeführt.

Oft wird die Fingergelenkstellung gar nicht betrachtet und zur Verfolgung einer Hand die Berechnung des optischen Flusses angestellt [Nordlund 96], die Korrelation von Stereofarbbildern [Arsenio 97] oder Farbsegmentierungen [Sidenbladh 99]. Es existieren auch immer mehr multimodale Methoden, die Tiefenbildauswertung, Farbbetrachtungen und Bewegungsdetektion kombinieren [Feyrer 99, Fritsch 00].

<sup>12</sup>engl.: 3D Template Matching Algorithm



Autor	Objekt	<i>DoF</i>	Modell	Anpassungsbasis
Goncalves [Goncalves 95]	Arm	4	Konus	Merkmale
Kakadiaris [Kakadiaris 96]	Arm	3	Ell. Zylinder	Merkmale
Rehg/Kanade [Rehg 94]	Hand	27	Zylinder	Projektion
Rehg/Kanade [Rehg 95]	Hand	9	Zylinder	Projektion
Yamamoto [Yamamoto 91]	Arm	3	—	Merkmale
Yeasin [Yeasin 00]	Hand	4	Zylinder	Merkmale
Wheeler [Wheeler 95]	Hand	7	Polygone	Projektion
Wu [Wu 01]	Hand	21	Kästen	Projektion
Jiar [Jiar 96b]	Hand	12	Oberflächenmodelle	Tiefenbildanpassung

Tabelle 2.1: Überblick über Hand- und Armverfolgungsansätze (*DoF*: Anzahl der Freiheitsgrade des Modells)

### 2.3.3 Gestenerkennung

Gesten können als Oberbegriff für durch Handstellungen ausgedrückte Symbole zur Kommandierung, Instruierung oder Dialogführung angesehen werden. Bei Handzeichen lässt sich eine Unterscheidung in statische und dynamische Gesten treffen, bei denen sich der bedeutungstragende Teil aus der Fingerstellung bzw. aus der Handbewegung ergibt.

**Statische Gesten:** Kestler benutzt Kamerabilder, um statische Handgesten zu erkennen und zu klassifizieren [Kestler 96]. Die Klassifikation geschieht nach einer Vorverarbeitung durch Vergleiche von Tensoren der Gestenmuster mit dem aktuellen Bild wie bei einer Eigenraummethode.

Wird die Handverfolgung durch Konturmodelle realisiert, lassen sich direkt aus der aktuellen Merkmalsverteilung Parameter extrahieren, mit denen eine Geste beschreibbar wird [Blake 98a, Heap 95]. Dieser Ansatz entspricht im Wesentlichen dem Anpassen elastischer Graphen an Bildmerkmale in [Triesch 01]. Grundlage ist hier Bewegungs- und Farbinformation, die mit 8Hz bei  $96 \times 71$  Pixeln Auflösung erreicht wird. Damit sind sechs Gesten klassifizierbar, die zur Greifsteuerung eines Roboters dienen (siehe Abbildung 2.6).

Bei monochromem Hintergrund können vorsegmentierte Bereiche auch von neuronalen Netzwerken als Gesten klassifiziert werden [Banarse 96]. Die Eingabeneuronen erhalten dazu ihre Eingabe direkt von den Bildelementen. In einer ersten Schicht werden die Eingabedaten zunächst mit Gaborfiltern behandelt. Zwei nachgeschaltete *RBF*-Schichten dienen zur Erkennung von Merkmalen in den vorgefilterten Werten. In einer nachgeschalteten Klassifikationsschicht wird dann ein Gewinner ausgewählt.

Ein ähnlicher Ansatz wird von Lamar verfolgt [Lamar 99]: Hier klassifiziert ein neuronales Netz einen 20-dimensionalen Vektor, auf den nach einer Hauptkomponentenana-

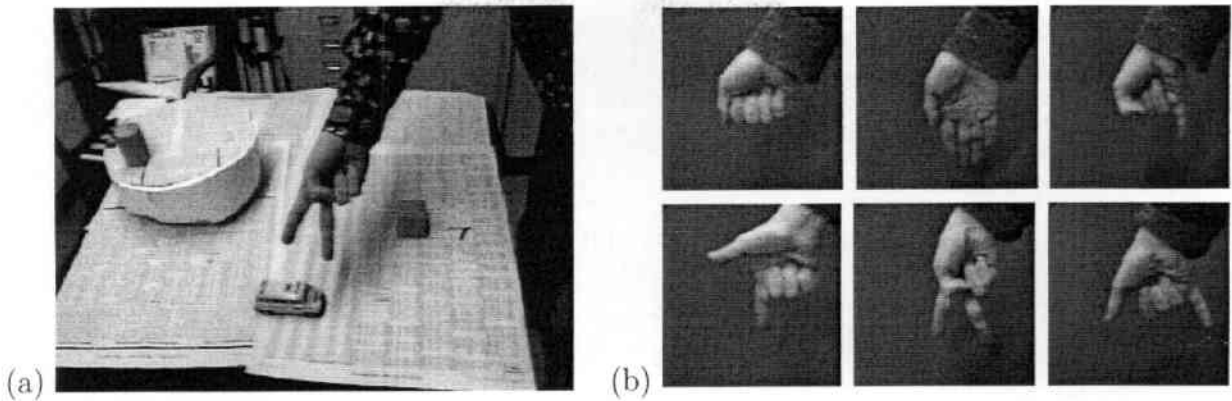


Abbildung 2.6: Kamerabild vor der Skalierung (a) und klassifizierbare Gesten (b) nach Triesch (Quelle: [Triesch 01])

lyse eines vorher segmentierten Hautfarbbereichs die Benutzerhand abgebildet wurde. Die Erkennungsraten sind in beiden Verfahren sehr gut; dabei ist jedoch anzumerken, dass Handrotationen nicht berücksichtigt wurden.

All die genannten Verfahren untersuchen Gesten auf Basis von zweidimensionaler Information, die in einer bestimmten Ausrichtung zur Kamera ausgeführt werden müssen. Eine Arbeit zur Bestimmung der Zielrichtung einer Zeigegeste in 3D findet sich in [Jojic 00]. Er verwendet Disparitätsbilder und Hintergrundsubtraktion zur Detektion eines ausgestreckten Arms. Dazu werden die Punkte des Tiefenbildes mit statistischen Methoden den Klassen „Arm“ und „Körper“ zugeordnet und die Hauptachse der Armpunkte bestimmt.

Die Silhouette der Hand in Projektionen verschiedener Winkel wird in [Leibe 01] genutzt, um Tiefeninformation über die Zeigerichtung zu erhalten. Dies ist jedoch nur möglich, wenn Handansichten aus verschiedenen Blickwinkeln gleichzeitig verfügbar sind.

**Dynamische Gesten:** Auf der Basis eines Hidden Markov Modells (HMM) wurde ein System zur Erkennung von 14 verschiedenen Gesten vorgeschlagen, von denen jede online aus nur zwei bis drei Beispielen gelernt wurde [Lee 96]. Als Eingabegerät wird hier ein Datenhandschuh mit Magnetfeldsensor verwendet, dessen Positionsänderungen mit Hilfe der Modelle klassifiziert werden. Ein System zur Klassifizierung der koreanischen Zeichensprache, bei der Fingerstellungen und Handbewegungen beider Hände Bedeutungsträger sind, ist mit Hilfe von zwei Datenhandschuhen und magnetfeldbasierten Positionssensoren unter Verwendung von fuzzy min-max neuronalen Netzwerken realisiert worden [Kim 96].

Es ist jedoch zu beobachten, dass aufgrund der mit neuronalen Netzen schwierig zu erreichenden Erkennung bedeutungsloser Verfahrbahnen zunehmend HMM zur Klassifikation eingesetzt werden [Starner 95, Rybski 99]. Meist werden sie mit Standard-Methoden trainiert und getestet (Einführungen in diese Modelle und Methoden finden sich in [Forney 73, Rabiner 89, Ryan 93a, Ryan 93b]). Zur Verfolgung der Handbewegungen dienen beispielsweise Differenzbildverfahren. Auf dieser Basis können

zweihändige Bewegungsmuster mit vormodellierten HMMs erkannt werden [Rigoll 97]. Dynamische Gesten können so zur Steuerung einer Vortragspräsentation eingesetzt werden, wobei zur Vermeidung von Fehlerkennungen ein Schwellwertmodell eingeführt worden ist [Lee 99].

Da das Problem der Gestenerkennung bei dynamischen Handzeichen der Erkennung von Handschrift ähnlich ist, werden Methoden auf beiden Seiten in gleicher Weise verwendet [Plamnondon 00]. Neuerdings werden Kameras auch am Kopf des Benutzer befestigt, um Gesten aus dessen Perspektive zu erkennen [Pentland 00].

Einen Überblick über die technische Realisierung verschiedener gestenerkennender Systeme hat Kohler zusammengestellt [Kohler 00]. Hier werden jedoch nur bildverarbeitende Systeme berücksichtigt.

### 2.3.4 Grifferkennung

Neben den beschriebenen Handmodellen und Ansätzen zur Handverfolgung werden Datenhandschuhe und Kamerasysteme eingesetzt, um Griffe zu erkennen und zu klassifizieren.

Die meisten Grifferkennung sind sehr einfach gehalten und erkennen lediglich einen Griffotyp. Bei Datenhandschuhen werden beispielsweise Schwellwerte für das Beugeverhalten der vier Finger und des Daumens betrachtet [Takahashi 92, Yuan 97]. Bei der bildbasierten Handverfolgung werden oft nur zwei Finger verfolgt und bei einer bestimmten Nähe zueinander ein Greifzeitpunkt detektiert [Kuniyoshi 94]. Manchmal erfolgt die Detektion auch heuristisch anhand der Nähe zu Objekten bzw. deren Verschiebung [Leibe 01] oder allein anhand der Bewegungsgeschwindigkeit der Benutzerhand. Studien belegen, dass die Trajektorie zwischen Griffen in kartesischen Koordinaten nahezu eine gerade Linie mit glockenförmigem Geschwindigkeitsprofil ist [Hauck 98].

Um verschiedene Griffotypen klassifizieren zu können, werden neben Kontaktbetrachtungen oft Grifffhierarchien benutzt. Der erste Versuch, eine solche Hierarchie aufzustellen, ist eine Untersuchung von Schlesinger und sollte die Konstruktion von Prothesen verbessern [Schlesinger 19]. Grundlage für die Beobachtung und Analyse von Griffen ist heute weithin das Klassifikationsschema nach Cutkosky [Cutkosky 89], das auf Schlesingers und Napiers Arbeiten aufbaut [Napier 56].

Für die Greifplanung und die Verschmelzung von prophetischen Händen und Greifsystemen werden Griffe oft nach einem Gegenraummodell<sup>13</sup> klassifiziert [Iberall 94]. Dabei werden Finger betrachtet, die einander gegenüber stehen und die eine Kraft auf den Schwerpunkt der Finger zu auswirken. Finger und Handfläche, die mit ähnlichen Kräften und Momenten auf ein Objekt wirken, werden zu virtuellen Fingern zusammengefasst [Arbib 85].

Aktuelle Grifferkennung basieren überwiegend auf den Sensormessungen von Datenhandschuhen. Zur Erkennung der Greif- und Ablagezeitpunkte werden lokale Geschwindigkeitsminima

---

<sup>13</sup>engl.: Opposition Space Model

in der Demonstration betrachtet, in deren Nähe Sensormuster liegen, bei denen die Polygonfläche zwischen Fingerspitzen des Benutzers lokal maximal ist. Dieses Verfahren beruht auf Studien, gemäß derer der Mensch kurz vor dem Greifen eines Objektes die Finger weitet [Jeanneroud 84]. Zum Feststellen des genauen Griffzeitpunkts werden meist Kontaktpunkte zwischen Hand und Objekten in Geometriemodellen berechnet [Kang 97]. Die Kontaktpunktberechnung ist jedoch aus mehreren Gründen schwer exakt zu lösen, denn dazu sind exakte Handmodelle notwendig. Da menschliche Hände jedoch stark in Größe und Form variieren, müsste jedes Modell benutzerspezifisch ausfallen. Des Weiteren liegen die Dehnmessstreifen, die als Sensoren die Fingergelenkstellungen messen, meist unterschiedlich an und müssen aufwändig kalibriert werden (siehe Abschnitt 2.2.3).

Sämtliche Griffe des Cutkosky-Schemas können auch als statische Gesten mit geringem Rechenaufwand von einem hierarchischen neuronalen Netz zuverlässig erkannt und unterschieden werden [Ehrenmann 98, Friedrich 99].

Zusammenfassend lässt sich sagen, dass bei der bildgestützten Gesten- und Grifferkennung von sehr vielen Einschränkungen hinsichtlich der Beleuchtungsqualität und der Art der Vorführung Gebrauch gemacht wird. Viele Gestenerkennungssysteme erwarten Handartikulationen in einem Abstand von 40–60cm vor der Kamera, die in einer Ebene parallel zum CCD-Sensor ausgeführt werden.

## 2.4 Qualitative Handlungserkennung

Die Erkennung einzelner bedeutungstragender Aktionen oder solcher innerhalb einer komplexen Aufgabe wie beim Zusammensetzen von Bauteilen zu einem komplexen Konstrukt ist Voraussetzung für das Verstehen bzw. Lernen aus Beobachtung. Zwei Systeme stehen beispielhaft für dieses Problemfeld:

- Die gezielte Beobachtung von Handlungen zum Lernen von Zusammenhängen wird von Fritsch behandelt [Fritsch 00]. Er stellt einen bildbasierten Ansatz zur Analyse von Konstruktionshandlungen vor. Die mit den Händen durchgeführten Objektmanipulationen sind das Nehmen und Ablegen von Bauteilen, das Schrauben sowie die Herstellung von Verbindungen. Eine Vorverarbeitung ermittelt durch Einsatz von Farb- und Bewegungsinformation Handhypothesen im Bild, die mit Kalmanfiltern verfolgt werden (siehe Abbildung 2.7 b). Die folgende Analyse der Trajektorien realisiert die Bestimmung der Handregionen, deren Bewegungsmuster für die Handlungsklassifikation mit dem *Condensation*-Algorithmus verwendet werden. Dabei können mehrere Aktionshypothesen gleichzeitig verfolgt sowie probabilistische Informationen über Aktionsfolgen in den Klassifikationsprozeß integriert werden. Angestrebt wird, Bildfolgenerkennung und Sprache zu fusionieren, um sprachliche Referenzierung durch Zeigegesten zu disambiguieren (z.B.: „greife das grüne Objekt“). Aus den Vorführungen werden schließlich Baupläne für Modelle aus Bauspielzeug in Form von Graphen erzeugt.
- Einen bezüglich der qualitativen Handlungsanalyse ähnlichen Ansatz verfolgt der *ak-*

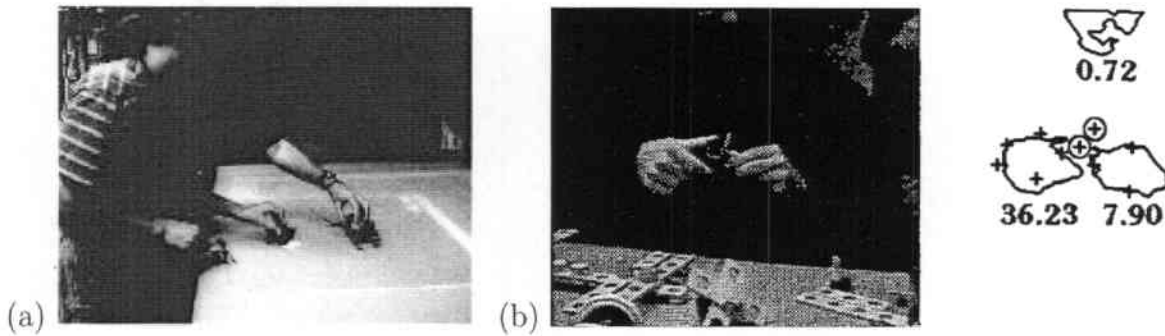


Abbildung 2.7: Der *aktive Arbeitsplatz* mit der Tischfläche als Projektionsleinwand (a, aus [Starner 00]) und die Verfolgung zweier Hände nach Sagerer (b links, aus [Fritsch 00]) mit dem Segmentierungsergebnis (b rechts, Quelle dito)

*tive Arbeitsplatz* [Leibe 01, Starner 00],<sup>14</sup> der bildbasiert Gesten sowie das Aufnehmen und Bewegen von Objekten auf einem Tisch verfolgen kann (siehe Abbildung 2.7 a). Beobachtet wird dazu von mehreren fixierten Kameras unter und neben dem Tisch der infrarote Schatten des Arms. Mit einer rotatorischen Genauigkeit von  $10^\circ$  können Zeigegesten erkannt werden, die eine Objektidentifizierung anstoßen. Die Objekte können dazu einfach auf den Tisch gelegt und bewegt werden. Der Arbeitsplatz überlagert durch Projektion echte mit virtuellen Objekten, mit denen der Benutzer interagieren kann. Hier soll die von Computerschnittstellen bekannte Schreibtischmetapher in die reale Welt übertragen werden.

Da für Roboter jedoch spezielle Handlungsparameter wie Trajektorien, Kraftwerte oder objektspezifische Informationen beim Greifen wie Greifpunkte oder Griffarten zu berücksichtigen sind und die obigen Beispiele kein Handhabungswissen speichern oder zu generieren in der Lage sind, können die Ergebnisse aus diesen Arbeiten nicht direkt auf Robotersysteme übertragen werden. Beide Systeme greifen außer durch die Visualisierung nicht in ihre Umwelt ein.

Hier ist eine Betrachtung dedizierter Ansätze notwendig. Diese roboterspezifischen Ansätze werden im Folgenden in zwei Kategorien unterteilt:

**Ansätze des Programmierens durch Vormachen:** Beim Programmieren durch Vormachen soll ein System eine Beschreibung von beobachteten Aktionen erstellen, aus der ein ausführbares Roboterprogramm generiert werden kann.

**Interaktive Ansätze:** Hier interagieren Benutzer und Roboter während der Vorführung. Der Mensch benennt und bewegt Gegenstände so, dass sie sofort von beiden gemeinsam benutzt werden können.

<sup>14</sup>engl.: Perceptive Workbench



### 2.4.1 Programmieren durch Vormachen

Die Erkennung von Manipulationen in komplexen Handlungsfolgen dient dazu, Robotern das Lernen von Fähigkeiten durch Beobachtung zu ermöglichen<sup>15</sup>. Dieses sogenannte „Programmieren durch Vormachen“ (PdV)<sup>16</sup> hat bereits eine fast zehnjährige Tradition. Im Folgenden sollen die wichtigsten Vertreter der dazu vorgeschlagenen Systeme vorgestellt werden.

**APO:** Von Ikeuchi, Kang, Jiar et. al. wurde erstmals ein solches, *APO*<sup>17</sup> genanntes, System entwickelt [Ikeuchi 94, Paul 95].

Grundlegend für die Interpretation einer Manipulationsfolge ist die Unterteilung der gesamten Handlung in sinnvolle Abschnitte. Die Segmentierungspunkte sind Griffe und Ablegezeitpunkte [Kang 93]. Zu diesen Zeitpunkten werden die Kontaktzustände zwischen den Objekten, die in einem geometrischen Weltmodell vertreten sind, neu bestimmt. Die ausschließliche Verwendung einfacher polyedrischer Körper beschleunigt diesen Prozess. Die Lage dieser Objekte wird zunächst grob von Hand vorgegeben. Diese initiale Schätzung wird iterativ verfeinert. Erst danach kann die Manipulation beginnen.

Die gesamte Manipulation wird als Folge von Kontaktzuständen repräsentiert. Operationen, die diese Zustandsübergänge durch Aktionen eines Roboters überführen, werden Task-Modelle genannt und bilden die elementaren Aktionen des Systems, die erkannt werden müssen. Kang verwendet entweder Tiefenbilder eines statischen Kamerasystems oder einen Datenhandschuh, um verschiedene Grifftypen in Handlungsfolgen unterscheiden zu können [Kang 94]. Erfolgt die Vorführung mit dem Datenhandschuh, werden die Ungenauigkeiten des verwendeten Positionssensors durch eine lokale geometriebasierte Optimierung am Greifpunkt ausgeglichen. Die Griffe selbst werden auf Basis des Geometriemodells durch Kontaktpunkte detektiert, die sich zwischen den Handsegmenten und den manipulierten Objekten ergeben. Sie werden nach einer auf Arbib's virtuellem Fingerkonzept aufbauenden Hierarchie [Arbib 85] klassifiziert und auf Roboterhände übertragen. Die drei Handlungsphasen bestehen aus der Vorbereitung des Griiffs, dem Griff selbst und der Manipulationsphase. Fingerbewegungen während dieser letzteren Phase werden mitverfolgt.

Bei der Ausführung eines Montageplans werden nach der Identifizierung der zu manipulierenden Objekte die a-priori programmierten roboterspezifischen Aufgabenmodelle für die benötigten Kontaktzustandsübergänge initialisiert. Hierbei werden auch die ebenfalls a-priori erstellten Unterprogramme für die auszuführenden Griffe spezifisch für den verwendeten Greifer ausgewählt. Die Aufgabenmodelle und Greiferprogramme führen dann die erforderlichen Zustandsübergänge aus. Alle Berechnungsmethoden basieren dabei auf den Geometriemodellen der in der Umwelt enthaltenen Objekte.

<sup>15</sup>In diesem Abschnitt soll nicht der Erwerb einzelner, elementarer Fähigkeiten bzw. Regler betrachtet werden. Die vorgestellten Arbeiten gehen auf dieses Thema nicht ein. Für diese Aufgabe werden meist Bildverarbeitungssysteme in Verbindung mit Kraftmessdosen verwendet. Dem interessierten Leser sei hierzu beispielsweise [Kaiser 96] empfohlen.

<sup>16</sup>engl.: Programming by Demonstration (PbD)

<sup>17</sup>engl.: Assembly Plan from Observation

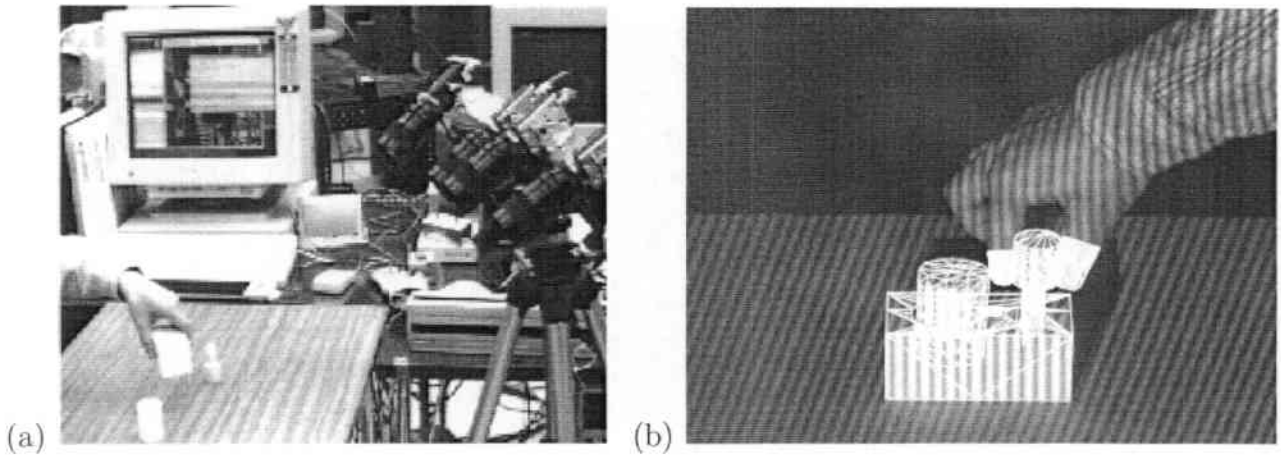


Abbildung 2.8: Handverfolgung in *APO*: Experimentierumgebung (a) und Schablonenanpassung (b, beide Bilder [Jiar 96a] entnommen)

Um dieses System zu erweitern, schlug [Jiar 96a] den ausschließlichen Einsatz von Tiefenbildfolgen vor, um auf die Verwendung des Datenhandschuhs gänzlich zu verzichten. Dazu verwendet er zwei fest installierte Kameraköpfe mit jeweils vier Kameras. Nach der Aufnahme einer Handlungssequenz mit den verwendeten Holzelementen werden Tiefenbildsequenzen berechnet und über den 3D-Schablonenalgorithmus<sup>18</sup> (siehe Abbildung 2.8) die initiale Position der Objekte sowie der Hand festgestellt. Dabei werden lediglich zwei Finger, der Daumen und Zeigefinger verfolgt. Aufgrund von Überdeckungen werden die restlichen nicht betrachtet. Die Handlung wird in vier Teilhandlungsphasen segmentiert: Annäherung, Berührung, Manipulation und Abrücken. Als Manipulationsaktionen stehen das Verschieben ebenso wie das Schrauben und Greifen zur Auswahl.

Der gesamte Prozeß der Handlungsbeobachtung gliedert sich dabei nach der Kalibrierung des Bildverarbeitungssystems und der Modellierung von Hand und Objekten wie folgt:

1. Aufzeichnung einer Bildsequenz während der Vorführung.
2. Berechnung einer Tiefenbildfolge aus der Aufzeichnung.
3. Bestimmung der initialen Konfiguration des Arbeitsraumes basierend auf dem ersten Tiefenbild. Hierzu werden manuell grobe Hinweise über die Objektpositionen gegeben, die dann iterativ verbessert werden.
4. Verfolgung der Hand sowie der Objekte.
5. Segmentierung der aufgezeichneten Datenströme.
6. Interpretation der Handlung.
7. Berechnung der Beziehungen zwischen den Objekten im Arbeitsraum.
8. Generierung einer Operatorenliste zur Realisierung der erkannten Kontaktzustandsübergangsfolge.

<sup>18</sup>siehe auch Abschnitt 2.3.2

9. Übersetzung der Operatorenliste in Roboterkommandos zur Reproduktion der Vorführung.

**LFO:** Eine qualitative Handlungserkennung wurde ebenfalls sehr früh von Inoue, Kuniyoshi et. al. unter dem Titel *Lernen aus Beobachtung*<sup>19</sup> vorgeschlagen [Kuniyoshi 93, Kuniyoshi 94]. Hier wurde erstmals das Problem der Aufmerksamkeitssteuerung im Rahmen des *PdV* betrachtet. Als Eingabe dienen Stereokamerabilder von einer Manipulationsaufgabe, die in Echtzeit verarbeitet werden. Auch hier finden ausschließlich einfache polyedrische Körper Verwendung, deren Position in der Initialphase bestimmt wird. Das implementierte System erkennt als Einzeloperationen das Greifen, Ablegen und genaue Platzieren von Objekten in der Szene<sup>20</sup> und erstellt aus der erkannten Aktionsfolge einen hierarchischen, symbolischen Ablaufplan für einen Roboter. Ein Planer generiert daraus ein lauffähiges Roboterprogramm.

Die Beobachtungsphase gliedert Kuniyoshi in folgende vier Abschnitte:

1. Bestimmung des Startzustandes und Aufbau des Umweltmodells
2. Finden und Verfolgen der Benutzerhand
3. Visuelle Suche nach dem Ziel der aktuellen Handlung
4. Erkennung bedeutungstragender Umweltveränderungen in der Nähe des Ziels

Griffe der Objekte müssen dabei immer so ausgeführt werden, dass das Bildverarbeitungssystem Daumen, Handrücken und Zeigefinger von links nach rechts isoliert voneinander detektieren kann. Die Hand wird dazu in ihrer zweidimensionalen Projektion betrachtet (siehe Abbildung 2.9 a). Wird der Abstand zwischen Zeigefinger und Daumen (linkes bzw. rechtes lokales Fenster) in der Nähe einer Objektposition kleiner als ein Schwellwert, gilt ein Griff als detektiert. Das Ziel einer Operation kann das Greifen oder Ablegen nach unten sein (siehe Abbildung 2.9 b). Die Modellierung der Manipulationsreihenfolge geschieht durch einen endlichen Automaten, der die Detektionsmodule in der Reihenfolge *Bewegen–Greifen–Bewegen–Ablegen* aufruft.

**TLT:** Ogata und Takahashi benutzen eine Umgebung in virtueller Realität als Demonstrationsschnittstelle [Ogata 94, Takahashi 92] im *aufgabenorientierten Vorführen*<sup>21</sup>. Der Benutzer trägt dazu eine Stereobrille und einen Datenhandschuh. Die einzige Operation in der simulierten Umgebung ist das Greifen eines Objekts, das anhand einer festen Fingerkonfiguration erkannt wird. Es werden ausschließlich Daten der drei translatorischen Freiheitsgrade der Benutzerhand betrachtet. In einem weiteren Experiment wird ein *Faro-Arm* (siehe Abschnitt 2.2.5) zur Positions- und Lageaufnahme genutzt [Tsuda 99].

Über einen vorcodierten endlichen Automaten werden die Handbewegungen nach Geschwindigkeit und Beschleunigung segmentiert und auf eine Folge von Operatorsymbolen abgebildet. Das resultierende Programm kann in einer äquivalenten Umwelt aus-

<sup>19</sup>engl.: Learning from Observation

<sup>20</sup>engl.: Pick and Place

<sup>21</sup>engl.: Task-Level Teaching



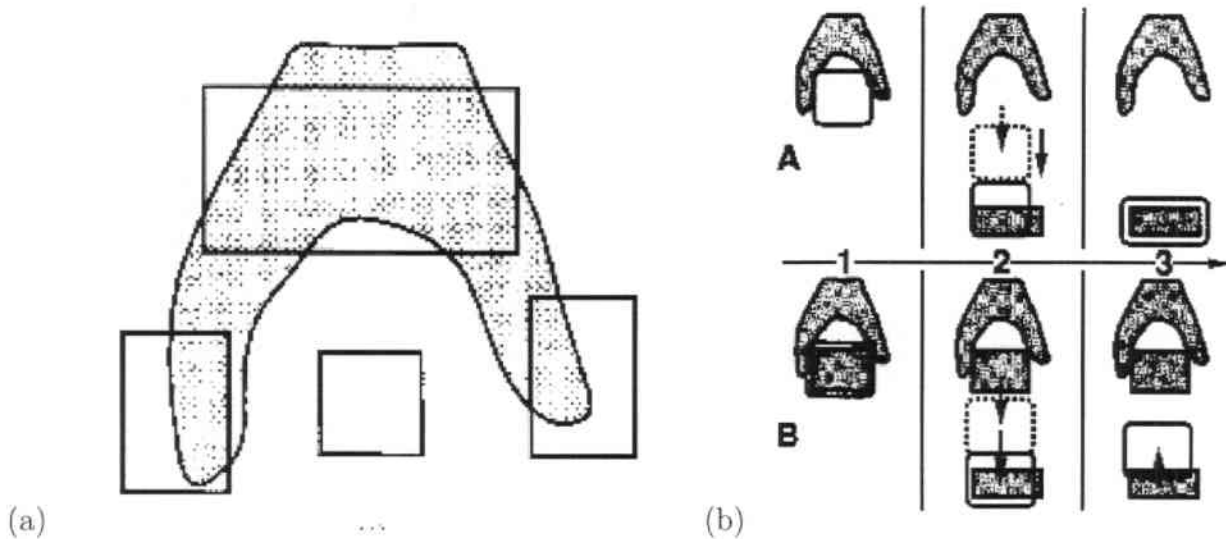


Abbildung 2.9: Handverfolgung in *LFO*: Lokales Verfolgen mit drei Fenstern (a) und Suche nach Handlungsziel bei leerer (b/A) und voller Hand (b/B) in zeitlicher Reihenfolge (1,2,3). Quelle: [Kuniyoshi 93]

geführt werden, wobei Anzahl und Art der Objekte der Vorführungsumgebung gleichen müssen, die Objektlagen aber variiert werden können.

Analog zu den *APO*- und *LFO*-Systemen werden hier lediglich die Auswirkungen einer Demonstration erfasst und repräsentiert. Der Trajektorienverlauf und dessen Charakteristika werden neu geplant. Es existiert auch hier keine Schnittstelle für Kommentierungen oder zur Erfassung der Intention des Benutzers.

Eine Erweiterung von Takahashi [Takahashi 96] behandelt die Verarbeitung mehrerer Demonstrationen durch Datennormalisierung und Waveletanalyse. Ogata et. al. [Ogata 97] erweitern den Ansatz auf die vollen sechs Freiheitsgrade der Lage der Benutzerhand. Sie erfassen und repräsentieren weiterhin zusätzliche Objektkonfigurationen während des Transports. Die Ausrichtung der Eckpunkte, Kanten und Flächen des transportierten Objekts wird hier mitrepräsentiert.

**ALAT:** Von Tung wird ein System vorgestellt zum *automatischen Lernen von Montageaufgaben*<sup>22</sup>, das ausschliesslich einen Datenhandschuh und einen magnetfeldbasierten Positionssensor verwendet [Tung 95]. Zunächst muss dazu ein Weltmodell mit Oberflächenmodellen der manipulierbaren Objekte manuell initialisiert werden. Das Handmodell besteht aus Quadern für die Handfläche und die Fingersegmente. Griffe werden hier durch Kontakte zwischen den Fingerspitzen des Handmodells und die modellierten Objekte einhüllende Kugeln erkannt. Die Kontaktflächen werden daraufhin genau bestimmt. Damit lassen sich alle Objekte des Weltmodells in drei Klassen rubrizieren: liegendes Objekt, gegriffenes Objekt und Benutzerhand. Die Kontakte zwischen solchen Objekten sind ebenfalls dreifach: liegendes Objekt/Hand, liegendes Objekt/gegriffenes Objekt und gegriffenes Objekt/Hand. Drei Phasen werden dementsprechend unter-

<sup>22</sup>engl.: Automatic Learning of Assembly Tasks

sucht: das Aufnehmen eines Objektes, das Verbinden zweier Objekte und das Ablegen bzw. Bewegen eines Objekts. Eine Sequenz dieser Phasen wird zusammen mit den entsprechenden Kontaktflächen auf ein Roboterprogramm abgebildet.

**ARP:** Yeasin [Yeasin 00, Yeasin 97] präsentiert ein System zur *automatischen Roboterprogrammierung*<sup>23</sup>, bei dem die Vorführung von Handlungen mit einer Hand vor einem Farbkamerapaar erfolgt. Die Handfläche, der Daumen und die Finger tragen farbige Marker. Der kleine Finger wird dabei außer Acht gelassen. Über einen Kalmanfilter und Clusteringalgorithmen werden Trajektorien gewonnen. Die Bewegungsvorführung wird mehrmals vorgenommen, um ein ganzes Bündel an Trajektorien zu erhalten. Aus diesem Bündel kann eine glatte Bewegungsbahn mit Alternativen bei Kollisionen für jeden Roboterfinger gewonnen werden. Anhand des Geschwindigkeitsprofils wird die Vorführung in freie, Anrück- und Feinbewegungssegmente unterteilt. Griffe werden jedoch nicht gesondert behandelt. Vielmehr werden die beobachteten Bahnsegmente in Roboterprogrammen als Teillösungen verwendet.

**IPOR:** Roboterprogrammierung durch Demonstration wird in dem von Friedrich vorgestellten System *Interaktives Programmieren von Robotern*<sup>24</sup> [Friedrich 98] realisiert. In dieser Arbeit werden über einen Datenhandschuh und über ein magnetfeldbasiertes Positioniersystem aufgezeichnete Benutzerdemonstrationen auf fünf elementare symbolische Operatoren abgebildet. Das An- und Abfahren, Greifen, Loslassen und Transferbewegungen werden automatisch mittels neuronaler Netze und Geschwindigkeitscharakteristika segmentiert. Der Benutzer kann fehlerhaft segmentierte Abschnitte mittels einer graphischen Schnittstellen korrigieren. Zu den Elementaroperationen werden Vor- und Nachbedingungen über den Geometriemodellen des Umweltmodells berechnet. Nach der Berechnung einer Auswahlbedingung für jedes direkt manipulierte Objekt wird die Intention des Benutzers festgelegt. Darauf basierend kann die Operatorsequenz optimiert und generalisiert werden. Die Auswahlbedingungen und Benutzerintention sind ebenso wie die Operatorsequenz im Anschluss an die Vorführung graphisch korrigier- und kommentierbar.

## 2.4.2 Interaktive Ansätze

Anders als beim Programmieren durch Vormachen, wo sich eine umfassende Analyse- und Interpretationsphase an die Vorführung anschließt, reagieren die folgenden Systeme bereits nach der Beobachtung einer bestimmten Situation oder nach der Detektion einer bestimmten Teilhandlung. Das Ziel dieser Ansätze ist aber ebenfalls die Erzeugung eines Roboterprogramms oder das Implementieren eines gewünschten Verhaltens.

**MA:** Bei Yuan findet sich ein dem aufgabenorientierten Vormachen von Takahashi ähnliches Vorgehen, das Programmieren von Montageaufgaben in einer virtuellen Umwelt (*mechanische Montage*<sup>25</sup>, [Yuan 97]). Die Aufgaben werden hier nicht direkt ausgeführt,

<sup>23</sup>engl.: Automatic Robot Programming

<sup>24</sup>engl.: Interactive Programming of Robots

<sup>25</sup>engl.: Mechanical Assembly

sondern durch Gesten parametrisiert. Erkannt werden mit einem Datenhandschuh als Sensor vier Gesten: die Faust, die Faust bei ausgestrecktem Zeigefinger bzw. Daumen und die geöffnete Hand. Diese werden direkt als vier diskrete Steuerbefehle interpretiert und lösen Zustandswechsel des Systems aus. Vier Zustände kennt die Schnittstelle: den Menüaufruf, das Halten eines Gegenstandes, Freibewegen und das Beenden zum Abbruch der Demonstration. In den Zuständen zur Menüauswahl und zum freien Bewegen werden die Handbewegungen aufgezeichnet. Über das Menü werden spezielle Roboterfähigkeiten wie Greifen, Schrauben oder Schieben aufgerufen. Diese beziehen sich immer auf das zuletzt manipulierte Objekt. Im Zustand Halten wird dieses nach der Auswahl eines Greifbefehls zusammen mit der Hand in der Simulation bewegt, um dem Benutzer eine visuelle Rückmeldung zu geben.

**GBP:** Die alleinige Verwendung von Gesten zur Programmierung von Robotern wird auch direkt in der Roboterumgebung untersucht. Hier wird eine Folge von Gesten verwendet, um bestimmte Roboterprimitive sukzessive einzugeben. Eine Liste von Schlüsselwörtern, die grundlegende Primitive intuitiv beschreibt, findet sich in [Tatsuno 96]. Voyles, der darauf aufbauend den Begriff des gestenbasierten Programmierens<sup>26</sup> prägte, unterscheidet symbolische Gesten zur Auslösung einer Elementaroperation, Bewegungsgesten zum Transport, taktile Gesten für die Modellierung von Kontaktzustandsübergängen und artikulatorische Gesten (d.h. gesprochene Sprache) zur Festlegung von Zielobjekten. Gesten werden hier verstanden als „unpräzise kontextabhängige Ereignisse, die die Intention eines Benutzers übermitteln“ (siehe [Voyles 95]). Zusammen mit Zustandsinformationen aus einem Umweltmodell lassen sich dazu Roboterprimitive parametrisiert hintereinanderreihen [Voyles 99b]. Eine semantische Analyse der Vorführung wird hier jedoch nicht unternommen; auch sind die unterschiedenen Gestentypen nicht in einem System zusammen implementiert:

- Die Erkennung der symbolischen Gesten und Bewegungsgesten geschieht auf Basis von Hidden Markov Modellen [Rybski 99]. Die Eingabesymbole dafür sind in ein diskretes Alphabet abgebildete Abstandsvektoren, die über eine auf einem mobilen Roboter installierte Farbkamera gewonnen werden. Das Bildverarbeitungssystem kann zwei farblich markierte Objekte und die Hand des Lehrers verfolgen. Der Abstandsvektor repräsentiert die relative Nähe der Benutzerhand zu einem der beiden Objekte. Die erkannten Gesten sind: Bewegung hin zu und weg von einem Objekt, Greifen und Ablegen eines Objektes. Für diese Gesten ist eine Bibliothek mit Elementarfähigkeiten für einen Manipulator entworfen worden, aus der entsprechende Aktionen ausgewählt werden, um die erkannte Gestensequenz zu reproduzieren.
- Taktile Gesten werden mit Hilfe eines Kraft/Momentensensors erkannt, der entweder direkt am Handgelenk eines Manipulatorarms fixiert ist oder an einer 3D-Maus [Voyles 95]. Mehrere Agenten detektieren Ecken oder Kanten beim Verfahren und gewinnen daraus Information über die Objektform, die gegriffen werden soll.

<sup>26</sup>engl.: Gesture Based Programming

- Ein Experiment mit Datenhandschuh, Trackingsensor und Kraftsensoren auf den Fingerspitzen ist in [Voyles 99a] beschrieben. Hier wird ein Stift in ein Loch gefügt. Die erkannten Gesten sind das kraftgeregelte Greifen und das Bewegen entlang beliebiger Achsen. Die Funktionsweise der Detektion und Klassifikation der Gesten ist nicht beschrieben.

**CORA:** Die Roboter *CORA*<sup>27</sup> und *Arnold* sind Experimentiergeräte für die Modellierung der Verhaltensorganisation nach neurobiologischen Vorbildern. Von Seelen, Steinhage et. al. verwenden dazu die Theorie dynamischer Systeme. Spezielle neuronale Netze, sogenannte Aramifelder [Steinhage 00b] sollen in einem geschlossenen Regelkreis vor allem zwei Verhaltensaspekte berücksichtigen: das Verändern der Umwelt seitens des Robotersystems und das Festlegen des 'Aufmerksamkeitsfokus' für Gehör und Sichtwahrnehmung.

Der Roboter *CORA* ist gegenüber dem Benutzer an eine Tischplatte montiert und manipuliert gemeinsam mit ihm darauffliegende Objekte (siehe Abbildung 2.10). Die Sensorik von *CORA* nimmt abhängig vom Situationskontext gesprochene Einzelworte auf, erkennt über Hautfarbsegmentierung statische Zeigegesten und die auf dem Tisch befindlichen Objekte über ansichtsbasierte Verfahren. Der Situationskontext wird hier von einem stabilen Zustand des Aramifeldes gegeben. Der Mensch kann nun Objekte benennen, bewegen oder den Roboter zum Greifen auffordern. Ein Ziel dabei ist es, auch bei unvollständiger Sensorinformation Verhaltenssequenzen zu trainieren, die den Roboter Programme ausführen lassen.

## 2.5 Vergleichende Bewertung der Ansätze

Im folgenden werden die oben beschriebenen Ansätze hinsichtlich ihrer Stärken und Schwächen diskutiert. Dabei werden zunächst die Ansätze besprochen, die nicht physische Vorführungen betreffen, sondern im virtuellen Raum arbeiten.

Handlungen im virtuellen Raum zu tätigen und zu beobachten hat einen anderen Umgang mit der Sensorik zur Folge als dies bei physischen Demonstrationen der Fall ist. Hier wird dem Benutzer zunächst die Übertragung des Szenarios in das System überlassen. Während der Demonstration sieht er hier über graphische Simulationen sofort die Effekte seiner Aktionen (*TLT*, *MA*). Fehlmessungen der Positionssensoren oder der Gestenerkennung gleicht er deshalb selbst aus, indem er seine Bewegungen dem Messverhalten anpasst oder Aktionen solange wiederholt, bis sie erkannt werden. Zu dieser gewöhnungsbedürftigen Anpassung des Verhaltens kommt das Problem der Navigation in der virtuellen Umgebung. Trotz verwendeter multimodaler Schnittstellen erreicht die Simulation für die sensomotorischen Fähigkeiten des Menschen keinen genügenden Realitätsgrad. Viele Benutzer zeigen deshalb bei längerem Arbeiten in der virtuellen Realität Symptome sogenannter Simulationsübelkeit, die sich in Wahrnehmungsstörungen, Gleichgewichtsstörungen und Orientierungsschwierigkeiten äußert [Kennedy 93]. Diese tritt besonders dann auf, wenn die Signale des Gleichgewichts-

<sup>27</sup>engl.: Cooperative Robot Assistant



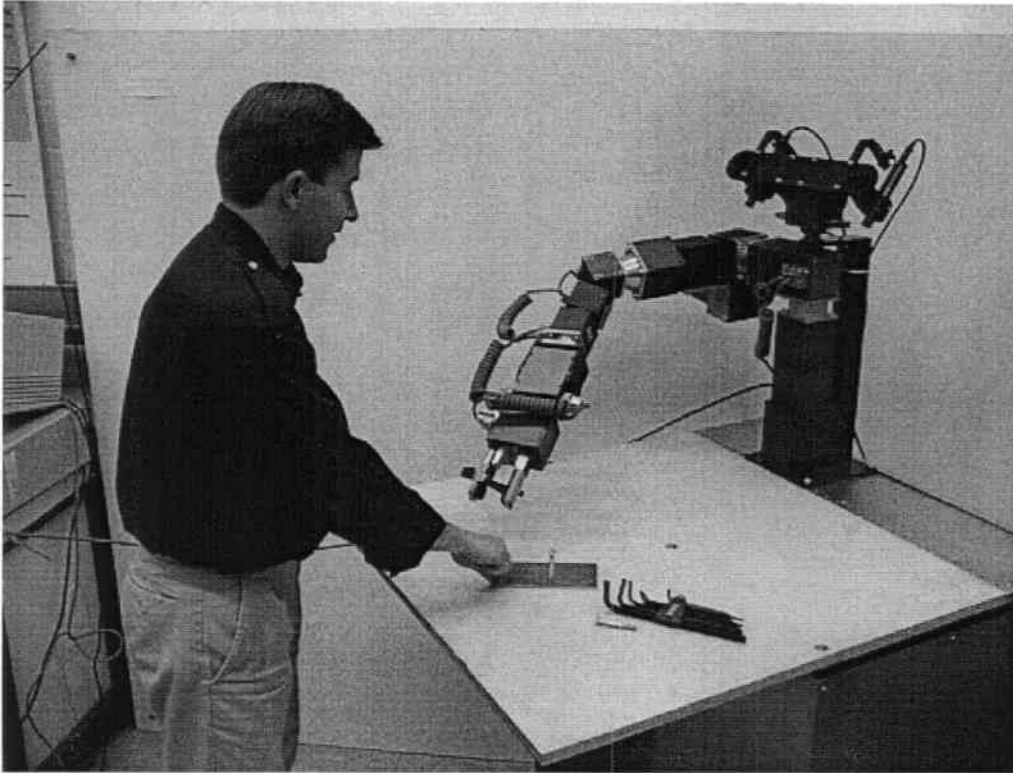


Abbildung 2.10: Handlungsbeobachtung bei interaktiver Belehrung des Roboters *CORA*, Bild aus [Bundesministerium für Bildung und Forschung 01]

und Orientierungssinns denen des Seh- und des Gehörsinns widersprechen [Nemire 94]. Das wiederum hängt vom Immersionsgrad ab. Während die Arbeit in voller Immersion bei Tragen eines Datenhelms oder in einer CAVE-Umgebung problematisch ist, verspricht vor allem die sogenannte halbe Immersion, also die Verwendung von 3D-Bildschirmen oder von Projektionsbrillen zur erweiterten Realität Arbeitserleichterungen. Es erscheint allerdings natürlicher, die Vorführungen physisch durchzuführen.

Den meisten Ansätzen gemeinsam ist jedoch auch bei physischen Vorführungen die räumliche Trennung von Vorführungs- und Ausführungsumgebung. Die Ausnahme ist der Ansatz beim Roboter *CORA*: die zur Beobachtung eingesetzte Sensorik ist hier bereits in das ausführende Robotersystem selbst integriert. Tabelle 2.2 gibt einen Überblick über den Sensoreinsatz und die Modellverwendung der betrachteten Systeme.

Mit Ausnahme des *CORA*-Systems lässt sich bei keinem dieser Ansätze der Einsatz aktiver Sensorik feststellen. Dies ist bei den bildverarbeitenden Systemen wie *APO* und *LFO* jedoch hinderlich für die Vorführung. Die Hand darf sich hier nur in ca. 50cm Entfernung zu den fest montierten Kamerasystemen in einem kleinen Sichtkorridor bewegen. Abgesehen von der Errechnung der Tiefenbilder bei diesen Systemen werden mit Ausnahme des *CORA*-Systems keine Sensordaten fusioniert. Dies ist deshalb überraschend, weil dadurch Mängel einzelner Verfahren gut kompensiert werden könnten.



System	Sensorik	Einsatz	Modelle	Umgebungen
APO [Kang 94]	Datenhandschuh, Polhemus	passiv	geometrisch, polyedrisch	separat
APO [Jiar 96b]	Zweimal vier Kameras	passiv	geometrisch, polyedrisch	separat
LFO [Kuniyoshi 94]	Stereokamera	passiv	geometrisch, polyedrisch	separat
TLT [Ogata 97]	Datenhandschuh, Polhemus	passiv	geometrisch, polyedrisch	separat
ALAT [Tung 95]	Datenhandschuh, Polhemus	passiv	geometrisch, beliebig	separat
ARP [Yeasin 00]	Stereofarbkamera	passiv	keine	separat
IPOR [Friedrich 98]	Datenhandschuh, Polhemus	passiv	geometrisch, beliebig	separat
MA [Yuan 97]	Datenhandschuh, Polhemus	passiv	geometrisch, beliebig	separat
GBP [Rybski 99]	Stereofarbkamera	passiv	Farbmerkmale	identisch
GBP [Voyles 95]	Kraft/Momenten- sensor	passiv	keine	identisch
GBP [Voyles 99a]	Datenhandschuh, Polhemus, Kraftsensoren	passiv	keine	identisch
CORA [Steinhage 00a]	Stereofarbkamera, Kraftsensorik, Richtmikrofon	aktiv	inhärent (neuronale Netze)	identisch

Tabelle 2.2: Sensoreinsatz bei den Ansätzen zur roboterspezifischen Handlungsbeobachtung. Betrachtet werden die Wahl der Sensorik, der Sensoreinsatz, die Objektmodellierung und die Trennung der Vorführ- und Ausführungsumgebung für ein Robotersystem

Die Wahl der Handlungsrepräsentation als Operatoresequenz, die Kontaktzustandsübergänge zwischen den Objekten bzw. der Hand realisiert, limitiert die Systeme *APO*, *LFO*, *ALAT* und *TLT* auf polyedrische Objektmodelle, bei denen Kontaktflächen leicht bestimmbar sind. Sie erfordern zudem hochgenaue Handmodelle zur präzisen Berechnung der Kontaktpunkte. Bei dem System *ALAT* ist das Handmodell aus diesem Grund sogar an einen speziellen Mitarbeiter angepasst worden. Der Vorteil dieser Systeme liegt in der Flexibilität der erzeugten Programme: diese sind in verschiedenen Szenarien nutzbar,

System	Trajektorien	Griff-		Kommen- tierung	Komman- dierung
		detektion	klassifikation		
APO [Kang 94]	—	ja	nach Arbib	—	—
APO [Jiar 96b]	—	ja	—	—	—
LFO [Kuniyoshi 94]	—	ja	—	—	—
TLT [Ogata 97]	ja	ja	—	—	—
ALAT [Tung 95]	ja	ja	—	—	—
ARP [Yeasin 00]	mehrfach	—	—	—	—
IPOR [Friedrich 98]	ja	ja	nach Cutkosky	graphisch	—
MA [Yuan 97]	ja	ja	—	gestenbasiert	—
GBP [Rybski 99]	ja	ja	—	—	—
GBP [Voyles 95]	ja	—	—	—	—
GBP [Voyles 99a]	ja	ja	—	—	—
CORA [Steinhage 00a]	—	ja	—	sprachlich, gestenbasiert	sprachlich

Tabelle 2.3: Beobachtete Merkmale der vorgestellten Systeme

solange dieselben Objekte vorliegen. Allerdings verwerfen die genannten Systeme bei der Planung von Kontaktzustandsübergängen die beobachteten Trajektorien und damit einen wichtigen Bestandteil der in der Vorführung enthaltenen Information. Deutlich wird dies in Tabelle 2.3, wo die von den Systemen beobachteten Merkmale zusammengestellt sind.

Hier wird auch gleich sichtbar, dass der Rückzug vom Datenhandschuh auf ein rein bildverfolgendes System bei Jiar einen Rückschritt hinsichtlich der Mächtigkeit der Griffklassifikation bedeutet: unterschiedliche Griffe können bildbasiert nicht stabil unterschieden werden. Die genaue Betrachtung der verwendeten Greifart ist aber notwendig zur adäquaten Abbildung von Griffen auf komplexe Mehrfingergrifer.

Als ein interessanter Ansatz zur Integration von Lernen von Verhaltenssequenzen und

multimodaler Interaktion stellt sich das System *CORA* dar. Leider ist hier der Lernerfolg nicht explizit repräsentiert, sondern liegt nur in Gewichten und Strukturen der neuronalen Felder vor. Deshalb ist auch eine manuelle Adaption oder ein direkter Einblick in den Lernerfolg nicht möglich. Auch ist nicht klar, wie gelernte Verhaltensmuster in ähnlichen Szenarien ablaufen oder aus einzelnen Mustern komplexere zusammengesetzt werden können. Hier fehlen die Kontrollstrukturen, die Programmiersprachen zur Verfügung stellen. Die sprachliche Kommandierung ist jedoch ein wichtiger Schritt hin zu kooperativ zu lösenden Aufgaben. Außerdem ist das System in der Lage, bei unsicheren Messungen Rückfragen zu stellen sowie Kommentierungen während Benutzerhandlungen entgegenzunehmen.

Rückfragen werden im Anschluss an die Demonstration auch vom System *IPOR* gestellt. Hier ist eine sinnvolle Kombination von physischer und graphischer Vorführung gegeben: die Vorführung geschieht mit realen Objekten, wird aber in der Simulation zur Kontrolle der Sensormessungen nachgespielt und interaktiv über Menüs korrigiert. Der Benutzer erhält auch eine graphische Repräsentation des erzeugten Programmes, das er erweitern und anpassen kann. Der vorherige, manuell zu leistende Aufbau des Umweltmodells und die ungenaue Messung des Positionssensors stellt aber hohe Anforderungen an den Benutzer.

Wie ohne die interne Verwendung von Weltmodellen und ohne Analyse die Intention einzelner Aktionen des Benutzers erfasst werden sollen, ist unklar. Eine semantische Analyse der Demonstration wird auch weder im *GBP*- noch im *ARP*-System beschrieben. Damit sind die erkannten Aktionen aber auch nicht an unterschiedliche Situationen für die Roboter Ausführung flexibel anpassbar. Deshalb erscheinen diese Ansätze als schwächer hinsichtlich der Zielstellung der Roboterinstruktion.

## 2.6 Zusammenfassung

Im obigen Kapitel wurden die wichtigsten Systeme zur Beobachtung von Handhabungsaufgaben vorgestellt und diskutiert. Obwohl in den letzten Jahren keine Veröffentlichungen zu den Systemen *APO* und *LFO* publiziert wurden, sind diese beiden immer noch die erfolgreichsten und meistzitierten auf dem Gebiet des Programmierens durch Vormachen. Keines der beiden Systeme verfügt jedoch über interaktive Mechanismen zur Kontrolle der Handlungsinterpretation, und an die Vorführung selbst werden sehr hohe Anforderungen gestellt. Trotzdem wurde mit diesen Systemen der beste Beleg dafür geliefert, dass beobachtete Demonstrationen von komplexen Aufgaben zur Roboterprogrammierung ausreichend sind: die auf deren Basis generierten Roboterprogramme reproduzierten die Vorführung jeweils korrekt. Einfachere Systeme, die virtuelle Räume zur Ausführung der Manipulationshandlungen nutzen, können wegen des wesentlich einfacheren Sensoreinsatzes als Vorstudien zu diesem Komplex betrachtet werden. Die Verwendung von kommentierenden Handlungen als Hilfe zur Interpretation wie im *GBP* und der Einsatz von aktiven Sensoren wie in *CORA* sollte die Leistungsfähigkeit und Flexibilität bei der Demonstration steigern helfen.

Für die Nutzung im Rahmen der Programmierung von Robotern erweisen sich die bislang präsentierten Systeme insgesamt als beschränkt. Dies betrifft sowohl ihre Fähigkeit,

wesentliche Handlungsparameter zu erfassen, als auch die hohen Anforderungen an den handelnden Benutzer. Die Erweiterung bestehender Handlungsverfolgungssysteme durch die kombinierte Nutzung verschiedener Sensorsysteme und eine umfassendere Aktionsverfolgung erweist sich als notwendig. Für eine robuste Grifferkennung erscheint dabei der Einsatz von Datenhandschuhen unabdingbar.

Für die vorliegende Arbeit leitet sich aus dem Gesagten der Anspruch ab, Vorführungen im Rahmen der Problemlösungsvorführung wie auch im Rahmen der Interaktion zu erfassen. Dabei werden entsprechende Sensordaten aus unterschiedlichen Beobachtungen fusioniert, um ohne Einschränkungen der Geschwindigkeit oder des Bewegungsspielraums Verfahrbahnen, Griffe und Gesten zu erfassen und zu klassifizieren.

# Kapitel 3

## Beobachtung menschlicher Handlungen

Es ist fraglich, ob Maschinen in der Art und Weise lernen können wie Menschen bzw. ob die Adaption menschlichen Beobachtungsverhaltens eine für die Maschine effiziente Art des Lernens ist. Die Programmierung von Robotern soll sich jedoch an der Anweisung von Menschen ausrichten und möglichst wenig Spezialwissen erfordern. Deshalb bereitet dieses Kapitel anhand der Fragestellung, wie Menschen Vorführungen betrachten bzw. wie Vorführungen für Menschen ablaufen, das nachfolgende Kapitel zur Modellierung und Repräsentation beobachtbarer Aktionen vor. Dabei stehen prototypische Handhabungsaufgaben aus dem Alltag wie Haushalts- oder Werkstattbereiche im Vordergrund.

### 3.1 Instruierung von Menschen

Drei Merkmale beschreibt die kognitive Psychologie [Anderson 89] als kennzeichnend für das praktische Problemlösungsverhalten:

**Zielgerichtetheit:** Das Verhalten ist eindeutig auf ein bestimmtes Ziel hin organisiert (zum Beispiel die Nahrungssuche).

**Zerlegung in Teilziele:** Das eigentliche Ziel wird in Teilziele zerlegt.

**Operatorenauswahl:** Für die Teilziele sind oft Handlungen bekannt. Der Begriff Operator bezeichnet eine Handlung, durch die ein Ziel direkt erreichbar wird. Die Lösung eines Gesamtproblems ist eine Folge aus solchen bekannten Operatoren.

Interessanterweise ist auch die Kognition nach heutiger Lehrmeinung zielgerichtet: die Wahrnehmung ordnet sich der Verhaltensorganisation unter. Die Zielgerichtetheit soll in den folgenden Betrachtungen während der ganzen Vorführung erhalten bleiben. Insbesondere werden Kontextwechsel nicht mit einbezogen. Kontextwechsel sind Unterbrechungen oder Beendigungen einer Vorführung durch den Demonstrierenden bei gleichzeitiger Aufnahme von Handlungen, die in anderem Zusammenhang mit einer eigenen Zielsetzung stehen. Zu Kontextwechseln kann es aufgrund von Ereignissen außerhalb des Manipulationsszenarios



oder aufgrund von Ablenkungen kommen. Da diese schwer zu erfassen sind, würde die Interpretation des Handlungsablaufs wesentlich schwieriger.

Die Betrachtung einiger Handhabungsaufgaben im Haushalt soll hierüber Aufschluß geben. Untersucht werden Handlungen, die täglich verrichtet werden. Beispiele hierfür sind das Öffnen von Schubladen oder Türen, Sammeln von Besteck und Geschirr auf einem Tablett, Drücken von Knöpfen, Putzen von Oberflächen oder Füllen einer Gießkanne. Hier ist festzustellen, dass die Hände in höchst unterschiedlichen Stellungen benutzt werden. Meist wird die Handlung nur von der dominanten Hand ausgeführt, während die andere in einer Ruheposition wartet (siehe Abbildungen 3.1).

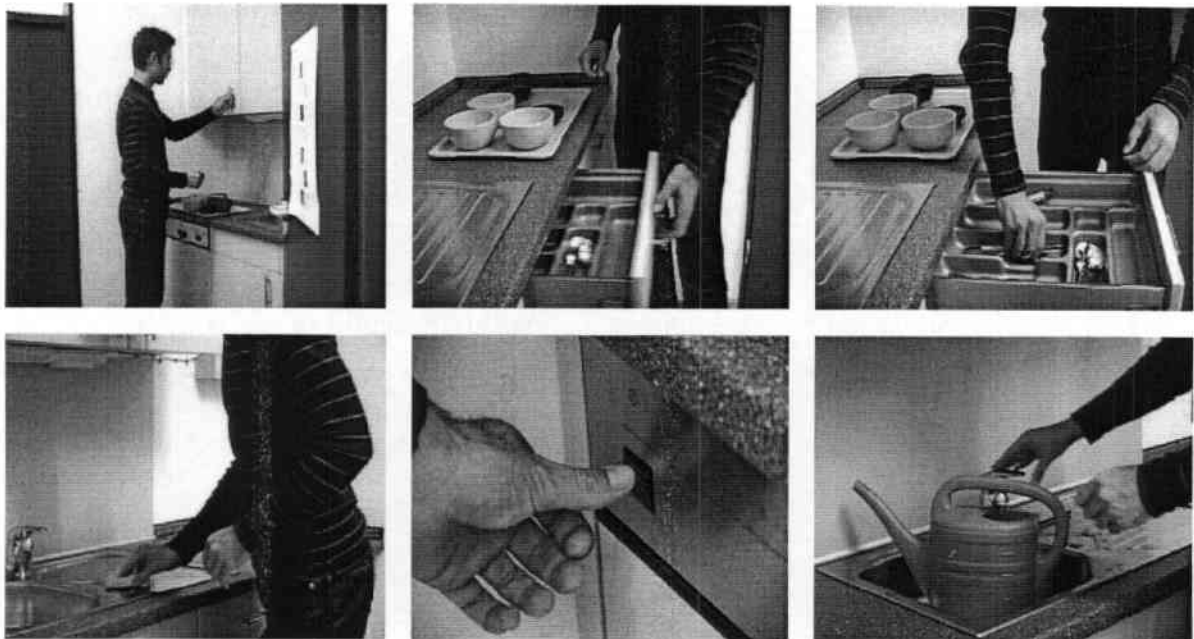


Abbildung 3.1: Beispiele von Handhabungsaufgaben im Haushalt

Werden Aufgaben mit zwei Händen oder durch mehrere Personen ausgeführt, müssen meist Gegenstände übergeben oder Teilaufgaben genauer spezifiziert werden, wenn einer der beiden Interaktionspartner nicht über ein Teilziel informiert ist. Dazu kommen ausgesprochene Kommandos ebenso wie Handzeichen zum Einsatz (Abbildung 3.2). Unter den Handgesten sind besonders Zeigegesten oder Stoppgesten häufig vertreten. Zeigegesten eignen sich oft besser als verbale Umschreibungen zur Referenzierung von Objekten oder zum Angeben von Richtungen.

Damit Aufgaben eigenständig ausgeführt werden, können diese wie in Abbildung 3.3 durch einen Demonstrierenden vorgeführt werden. Im Fall des Tischdeckens werden dazu hauptsächlich verschiedene Griffe und Gesten zusammen mit den jeweiligen Handbewegungen verwendet. Beide Handlungsweisen werden während der Ausführung der entsprechenden Teilhandlung üblicherweise sprachlich erklärend begleitet (siehe Abbildung 3.3).

Eine für die Handlungserkennung grundlegende Erkenntnis ist die, dass für Menschen eine Aktionsfolge als Sequenz klar voneinander geteilter Einzelhandlungen wahrgenommen wird



Abbildung 3.2: Beispiel für Kooperation bei Handhabungsaufgaben

und dass die brauchbarste Information zur Interpretation einer Handlung im Zustand des Wechsels zwischen zwei solchen Einzelaktionen vorliegt [Newton 77]. Die Segmentierung in einzelne Abschnitte wird also nicht nur wie eingangs erwähnt zur Planung einer Problemlösung vorgenommen, sondern auch zur Interpretation beobachteter Problemlösungen.

Der Vergleich der Demonstration direkt ausgeführter Handlungsfolgen mit zu Lehrzwecken ausgeführten Handlungsfolgen zeigt einen Unterschied in den ausgeführten Trajektorien. Um bestimmte Teilhandlungen deutlich zu zeigen, werden diese besonders betont (z.B. durch Führen eines Gegenstandes in die Nähe der Augen des Beobachters) oder zur besseren Wahrnehmung an das Sichtfeld des Beobachters angepasst. [Jiar 96b] führt eine analoge Beobachtung an: Der Demonstrierende greift eine Schraube, positioniert sie in einer Fügestelle und rückt ab, damit der Lernende dies sieht. Danach wird sie erneut gegriffen und eingedreht. Ein solches Verhalten wird besonders deutlich, wenn Gegenstände einzeln durch Gesten referenziert und ihre Lage sprachlich kommentiert wird (Bild rechts unten in Abbildung 3.3, „das Messer sollte rechts neben dem Teller liegen“). Hier wird die Vorführung sogar unterbrochen, um die Ziele von Einzelhandlungen zu definieren.

## 3.2 Typen beobachtbarer Handlungen

Menschliche Handlungen sollen im Rahmen dieser Arbeit mit dem Ziel erkannt werden, einem Roboter die Interpretation auf Basis eines gegebenen Kontextes und damit auch die Auswahl einer entsprechenden Reaktion zu ermöglichen. Lässt man den Sonderfall der Beobachtung menschlicher Aktionen während der Ausführung kooperativ gelöster Aufgaben beiseite, lassen sich im Rahmen der Interaktion zwischen zwei Menschen drei Typen beobachtbarer Handlungen identifizieren:

**Kommandierungen:** Das Geben eines Auftrages. Dies können beispielsweise Anweisungen zum Bewegen, Halten, Übergeben oder Holen eines Gegenstandes sein. Kommandierungen werden meist sprachlich oder durch symbolische Gesten vorgenommen.

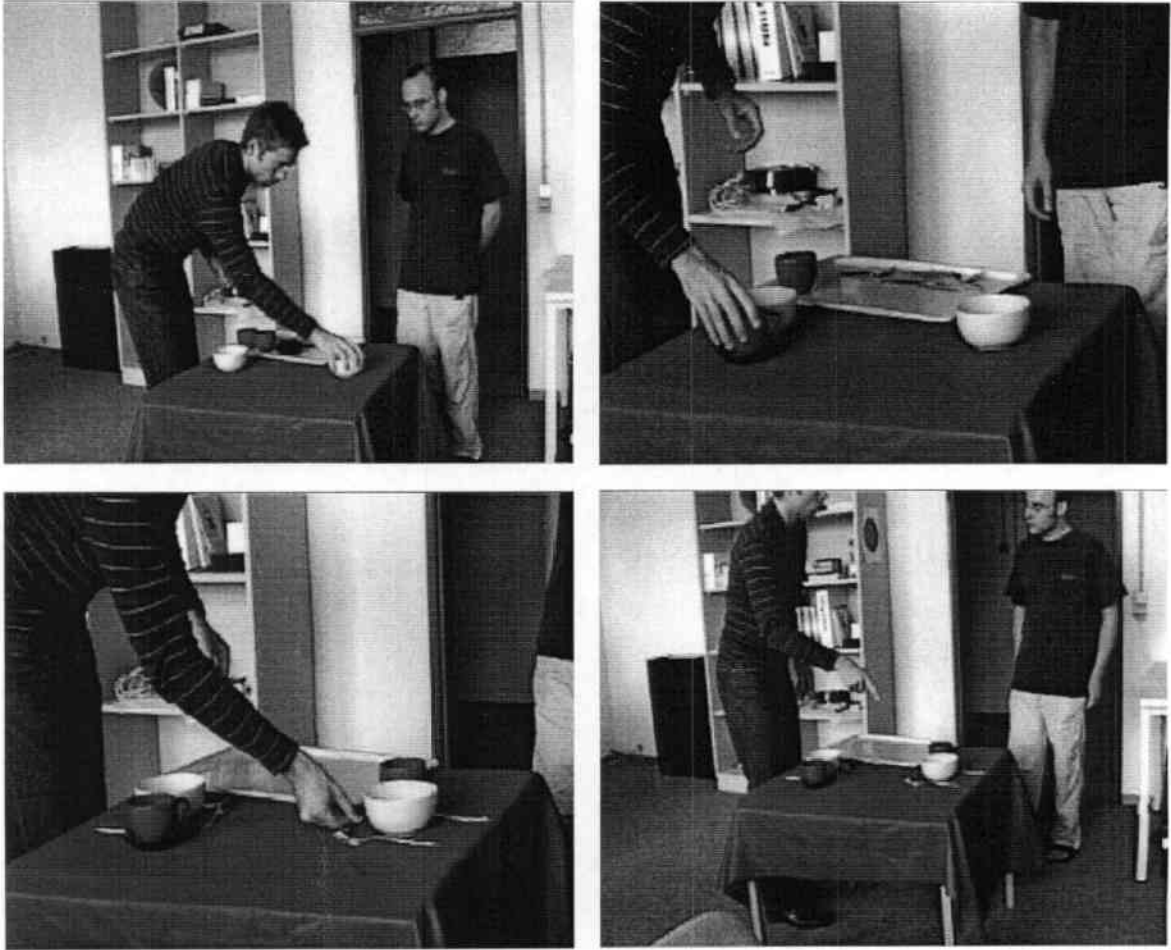


Abbildung 3.3: Beispiel für eine Lösungsvorführung

**Performative Handlungen:** Der Mensch führt die Lösung einer Aufgabe selbst vor. Der Interaktionspartner beobachtet die Lehrdemonstration und analysiert die ausgeführte Handlung. Er merkt sich die einzelnen Handlungsabschnitte und interpretiert sie soweit möglich.

**Kommentierungen:** Darunter sollen Erklärungen, die Benennung von Gegenständen oder Prozessen sowie Antworten auf Fragen im Rahmen eines Dialogs verstanden werden. Neben sprachlichen Referenzierungen kommen hier vor allem beschreibende und zeigende Gesten zum Tragen. Dieser Typ der Äußerungen dient vor allem zur Komplexitätsreduktion und Interpretationshilfe nach performativen Handlungen.

Während Kommandierungen und Kommentierungen zur Interpretation meist lediglich der Information über den aktuellen Kontext bedürfen, ergibt sich der Sinn einzelner performativer Akte oft erst nach mehreren Schritten oder liegt sogar außerhalb des Vorführszenarios. Tabelle 3.1 illustriert dies anhand eines Beispiels. Das Platzieren eines Messers rechts neben einem Teller beim Tischdecken erhält erst dann Sinn, wenn es von einer Person zum Essen dort erwartet wird. Auf Grundlage erfasster Objektlagen und deren Beziehungen allein kann nicht auf dieses Ziel geschlossen werden. Lediglich zwischen den Objekten hergestellte Relationen können detektiert werden. Es ist daher nicht zu erwarten, dass die Interpretation

Syntax	„Bewege Objekt <i>A</i> auf Position <i>B</i> “
Semantik	„Lege das Messer <i>C</i> neben den Teller <i>D</i> “
Pragmatik	„Lege das Messer für Person <i>E</i> zum Essen bereit“

Tabelle 3.1: Beispiel von Syntax, Semantik und Pragmatik beim Tischdecken

einer Vorführung ohne Hintergrundwissen oder ohne begleitende Erklärungen vollständig sein kann.

### 3.3 Zusammenfassung

Das vorliegende Kapitel dient zur Vorbereitung der Modellierung von Handlungen: diese sollen sich an Vorführungen für Menschen orientieren. Wichtig dabei ist die festgestellte klare Trennung komplexer Vorführungen in einzelne Handlungselemente und die Identifikation dreier unterschiedlicher Handlungstypen, nämlich kommandierender, kommentierender und performativer. Deren Interpretation bedarf jeweils eigener Methoden, wobei der höchste Aufwand für die Analyse performativer Aktionen zu leisten ist.

# Kapitel 4

## Sensorik und Modellierung

Im vorliegenden Kapitel wird eine Wahl für die Sensorik und Repräsentation der aufgezeichneten Sensordaten aus den Vorführungen getroffen und erläutert. Vor der Vorstellung der konkreten Sensorkomponenten wird zunächst der Prozess zur Interpretation von Vorführungen in seinen einzelnen Schritten präsentiert und Randbedingungen festgelegt, die für die Demonstrationen gelten sollen.

### 4.1 Die Interpretation komplexer performativer Handlungen

Performative Handlungen setzen sich aus einer Folge von partiellen Aktionen zusammen. Als Handlung wird im Folgenden die Bewegung der Hände betrachtet. Sprache, Blickkontakt, Kopfgesten oder Beinbewegungen werden nicht mit einbezogen. Für die Interpretation menschlicher Handlungen im Kontext von Objektmanipulationen ist damit zunächst die Erkennung der einzelnen manuellen Teilhandlungen notwendig. Abbildung 4.1 gibt einen Überblick über den gesamten Prozess. Der Programmierprozess beginnt mit der Benutzerdemonstration einer bestimmten Aufgabe, die von einem Sensorsystem beobachtet<sup>1</sup> wird. Die folgenden Phasen stellen die grundlegenden Komponenten für eine erfolgreiche Umsetzung der Vorführung dar [Ehrenmann 01b]:

**Beobachtung:** Das Sensorsystem wird zur visuellen Verfolgung der Benutzerbewegungen und zur Beobachtung der Benutzeraktionen benutzt. Wichtige Änderungen wie Objektlagen<sup>2</sup> werden erkannt.

**Segmentierung:** In der nächsten Phase werden die relevanten Operationen und Umweltzustände basierend auf der aufgezeichneten Sensorinformation extrahiert. Ziel ist die Segmentierung in bedeutsame semantisch zusammenhängende Teilabschnitte. Dies kann während der Vorführung in Echtzeit oder offline anhand aufgezeichneter Vorführungen erfolgen. Wenn der Benutzer bereits während der Vorführung über die Hypothesen des Systems bezüglich der beobachteten Ereignisse informiert wird, kann der Segmentierungsprozess stark beschleunigt werden.

---

<sup>1</sup>Zur formalen Definition des Begriffs „Beobachtung“ siehe Anhang A

<sup>2</sup>Zur formalen Definition des Begriffs „Objektlage“ siehe Anhang A



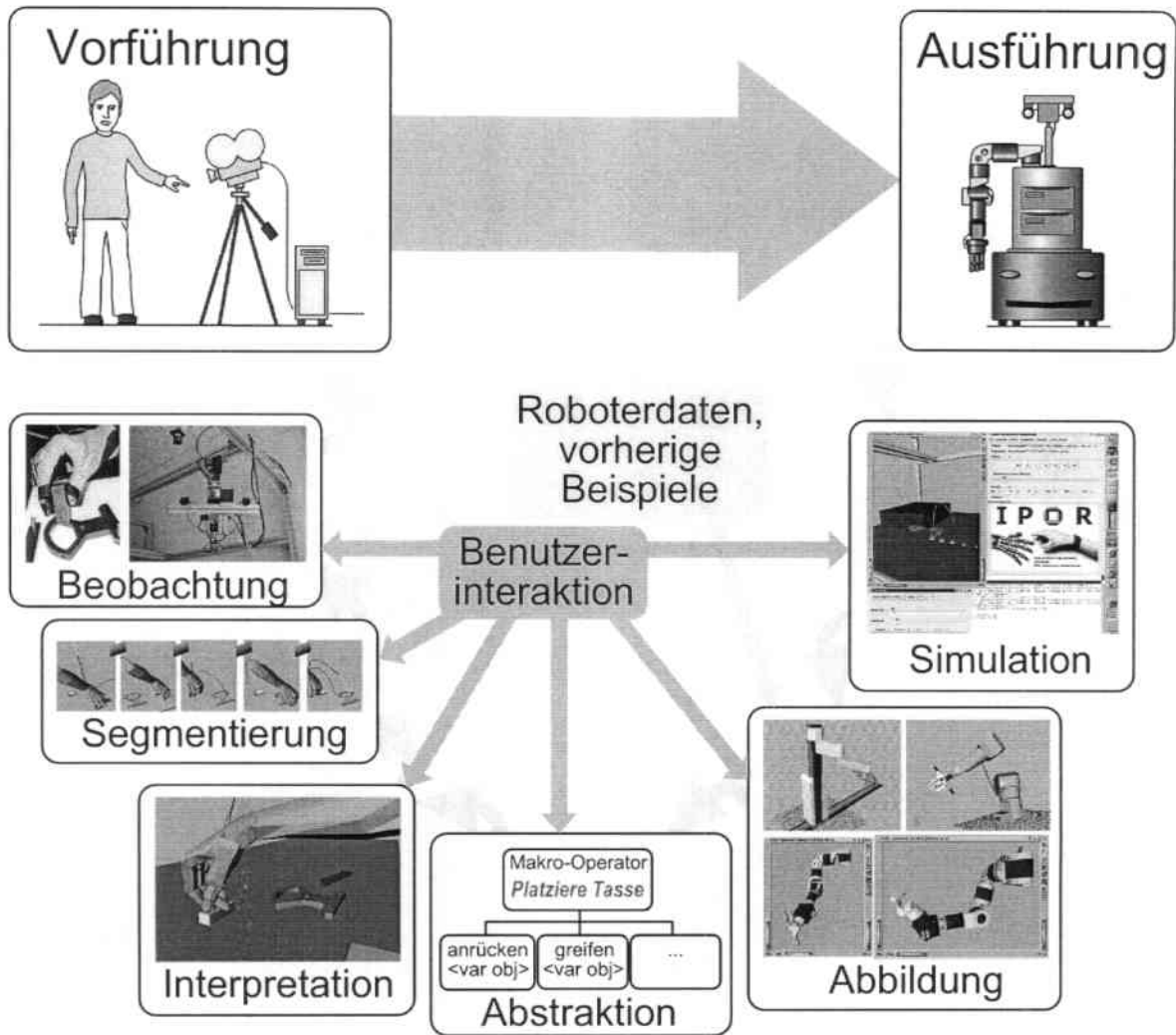


Abbildung 4.1: Interpretationsprozess für komplexe performative Handlungen

**Interpretation:** In der Interpretationsphase wird die segmentierte Benutzerdemonstration auf eine Symbolsequenz abgebildet. Die Symbole tragen Information über den verwendeten Handlungstyp (z.B. Geste, Griffstyp, Trajektorientyp) und entsprechende Parameter, die aus der Sensorik gewonnen werden (z.B. die genaue Stellung der Hand, das gegriffene Objekt).

**Abstraktion:** Das nächste Ziel ist die Bildung einer abstrakten Beschreibung der Vorführung auf Basis der erfassten Benutzerdemonstration. Die Generalisierung dient dazu, die relevanten Aktionen zu detektieren und in eine allgemeingültige, auch auf ähnliche Problemklassen anwendbare, Lösungsbeschreibung zu überführen.

**Abbildung:** Das gelernte Problemlösungswissen kann nun in einem weiteren Schritt zur Lösung der spezifizierten Aufgabe genutzt werden. Dazu ist eine explizite Umsetzung in Bewegungsoperationen des gewählten Robotersystems notwendig.

**Simulation:** Vor der Ausführung wird das generierte Roboterprogramm hinsichtlich seiner konkreten Anwendbarkeit auf ein definiertes Zielsystem in der geplanten Ausführungs-

umgebung simulativ getestet. Der Benutzer sollte dabei korrigierend eingreifen können, um gefährliche Situationen während der realen Ausführung zu vermeiden.

**Ausführung:** Während der physikalischen Ausführung der beobachteten Handlung auf dem Roboter können Erfolg und Misserfolg der gelernten Handlung zur Modifikation des Abbildungsverfahrens genutzt werden. Wird beispielsweise ein Objekt im Greifer instabil, können die Greifkräfte erhöht werden.

Während jedem Prozessschritt ist es sinnvoll, dem Benutzer die Möglichkeit korrigierender Eingriffe zu geben. Dadurch können Systemhypothesen überprüft oder verrauschte Sensormessungen verbessert werden. Sinnvollerweise geschieht dies direkt in der Vorführungsumgebung durch sprachliche Kommentare und ergänzende Gesten. Zur Überprüfung der Verarbeitung der Vorführungsdaten schließt sich also eine Phase interaktiver Kommentierung an, bis das Programm auf dem Roboter ausführbar ist. Die vorgeschlagenen Prozessphasen sind prinzipiell auf beliebige Handhabungsfolgen anwendbar.

## 4.2 Randbedingungen

Die in den Abschnitten 3.1 und 3.2 gemachten Annahmen sollen für die Aufzeichnung der Benutzeraktionen berücksichtigt werden. Für die Vorführungen sind also folgende grundlegende Bedingungen gültig:

**Ein-Hand-Modell:** Analog zum Effektor des programmierten Roboters soll auch beim Menschen zunächst nur die Bewegung einer Hand beobachtet werden. Sämtliche Manipulationen in dem Szenario sollen mit dieser erfolgen.

**Objektgröße:** Die manipulierten Objekte sollten der Größe der Hand entsprechen. Insbesondere müssen sie von dieser Hand manipulierbar sein.

**Geschlossene Welt:**<sup>3</sup> Außer der Benutzerhand greift kein anderer Agent in die Szene ein. Objekte, die an einer Position abgestellt wurden, bleiben stationär, bis sie wieder gegriffen und bewegt werden.

Erkannte Objektpositionen lassen sich also nicht ohne Zutun des Demonstrierenden verändern. Außerdem lassen sich die beobachteten Objekte in der Regel ansichtsbasiert erkennen. Zusätzlich zu diesen Bedingungen wird für die Handlungen des Benutzers angenommen:

**Kontextstabilität:** Der Benutzer wechselt nicht ohne Ankündigung den Kontext seiner Vorführung. Insbesondere sollten während einer Vorführung performativer Handlungen keine Spontanhandlungen auftreten. Beispiele sind etwa das Kratzen am Kopf oder das Begrüßen einer Person.

Derartige Kontextwechsel können nur mit Zusatzwissen erkannt und entsprechend zugeordnet werden. Von Spontanhandlungen gestörte Trajektorien führen bei der Generierung eines Roboterprogramms zu Fehlern und müssen vor während der Interpretationsphase gefiltert werden.

---

<sup>3</sup>engl.: Closed World Assumption

## 4.3 Sensorik

Wie in den Abschnitten 2.3.2 und 2.4.1 gezeigt wurde, ist die rein bildbasierte Verfolgung einer Handbewegung inklusive der Fingerstellungen nur in einem eingeschränkten Blickfeld einer stationären Kamera möglich. Aufgrund von Verdeckungen ist eine klare Aussage über die Stellung aller Finger der beobachteten Hand nicht möglich. Besonders hinderlich ist dies bei der Griffklassifikation. Die Möglichkeit, Vorführungen intuitiv und natürlich gestalten zu können und gleichzeitig verschiedene Grifftypen erkennen und klassifizieren zu können, lassen die Verwendung eines Datenhandschuhs für die Beobachtung performativer Handlungen sinnvoll erscheinen. Die Verwendung eines Datenhandschuhs ist allerdings bei der Interaktion mit einem Roboter eher hinderlich. In einem solchen Fall sollte ein Benutzer möglichst ohne Verwendung von Datenhandschuhen oder gar einem Exoskelett auskommen.

Die Anforderungen an die Handlungsbeobachtung zeigen sich im Kontext einer Programmierung komplexer Manipulationen als grundlegend unterschiedlich zu denjenigen einer online Mensch-Roboter Interaktion. Deshalb werden zwei unterschiedlich ausgeprägte Beobachtungsumgebungen vorgeschlagen:

**Vorführungsumgebung:** Zur Aufnahme und Aufzeichnung performativer Handlungen wird eine sogenannte Trainingsumgebung verwendet. Hier findet die Programmierung von Handhabungen durch Vormachen statt. Um Verfahrbahnen präzise verfolgen zu können und um verschiedene Griffarten robust zu klassifizieren, werden fusionierte Daten unterschiedlicher Sensoren wie einem Kamerasystem und einem Datenhandschuh angewandt. Neben performativen Handlungen werden Kommandos und Kommentierungen eingegeben, über die eine Vorführung korrigiert und bearbeitet werden kann.

**Ausführungsumgebung:** Die direkte Interaktionsschnittstelle zu dem Roboter unterstützt neben der direkten Kommandierung des Roboters zur Aktivierung und Parametrierung vorher gelernter Programme. Für die Ausführungsumgebung wird in der Regel nur das visuelle System des Zielroboters verwendet. Zur Erkennung kommentierender oder kommandierender Gesten ist dieser Sensor ausreichend. Ein nicht zu vernachlässigender Aspekt aktiver Kameras auf Robotersystemen ist deren für den Menschen wahrnehmbare Blickrichtung des Systems: der Benutzer kann somit unmittelbar auf den Aufmerksamkeitsfokus des Roboters schließen. Dies ist z.B. bei geplanten Bewegungen des Robotersystems zur Antizipation eines Verfahrziels durch den Benutzer sinnvoll.

Die Arbeitsweise dieser beiden Umgebungen wird im Folgenden erläutert.

### 4.3.1 Vorführungsumgebung für performative Handlungen

Zur Beobachtung performativer Vorführungen wird eine Trainingsumgebung für die Aufnahme prototypischer Handlungen vorgeschlagen. Sie besteht aus einem Tisch mit den Handhabungsobjekten als Szene sowie einem aktiven Sichtsystem, einem magnetfeldbasierten Positionssensor und einem Datenhandschuh zur Beobachtung und Aufzeichnung der Handlungen (siehe Abbildung 4.2 a). Die einzelnen Sensoren werden im Folgenden beschrieben.

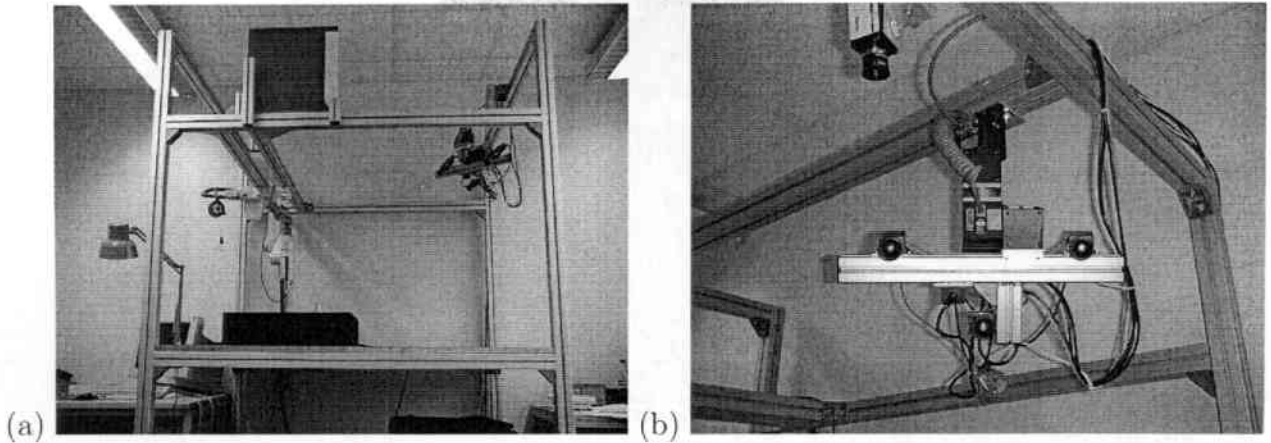


Abbildung 4.2: Die Vorführungsumgebung (a) mit dem integrierten trinokularen aktiven Sichtsystem (Detailaufnahme, b)

### Aktives Sichtsystem

Über das trinokulare Sichtsystem (siehe Abbildung 4.2 b) werden die auf der Tischplatte manipulierten Objekte klassifiziert und ihre Lage bestimmt. Außerdem wird die Position der Benutzerhand mit hoher Genauigkeit bestimmt. Drehachsen der Firma Amtec [Amtec 00] werden zum Neigen und Schwenken des Kopfes genutzt<sup>4</sup>. Die Bewegungsmodule zeichnen sich durch sehr hohe Positions- und Wiederholgenauigkeit aus. Die Bildsensorik besteht aus drei Pulnix TM765i Graubildkameras<sup>5</sup>. Die drei Bildsignale über RGB-Kanäle in einen Framegrabber [Matrox 98a] eingelesen und digitisiert.

### Datenhandschuh

Zur Erfassung der Handkonfiguration bzw. der Fingerposen wird ein Datenhandschuh mit 22 Sensoren der Firma Virtual Technologies (siehe Abbildung 4.3 a) eingesetzt.

Die Sensoren bestehen aus Dehnungsmessstreifen, die am Ort der Fingergelenke angebracht sind. Die Beugung bzw. Spreizung der Fingergelenke hat eine Längen- und Querschnittsänderung der Dehnungsmessstreifen zur Folge, deren resultierende Widerstandsänderung messtechnisch erfasst werden kann. Daraus werden Rückschlüsse über die Beuge- und Spreizwinkel der Finger gewonnen [Virtual 95]. Die von den Sensoren in dieser Art gelieferten Messwerte werden an die Systemeinheit übertragen und dort in die entsprechenden Fingerkonfigurationen umgewandelt. Über eine serielle Schnittstelle werden die Fingerkonfigurationen  $\vec{w} = (w_1, \dots, w_{22})^T$  an einen Rechner weitergeleitet und können dort graphisch dargestellt werden.

<sup>4</sup>technische Daten siehe Tabelle B.2 im Anhang B. Die Auflösung eines Moduls liegt bei 2000 Impulsen pro Umdrehung. Der Arbeitsraum wurde so gewählt, dass die gesamte Szene visuell erfasst werden kann. Softwaretechnisch wurden dazu die Drehwinkel des Neigemoduls auf einen Bereich zwischen  $0^\circ$  und  $90^\circ$ , die des Schwenkmoduls auf einen Bereich zwischen  $-10^\circ$  und  $110^\circ$  beschränkt.

<sup>5</sup>technische Daten siehe Tabelle B.1 im Anhang B. Für die Objekterkennung und Handverfolgung wird eine Auflösung von  $640 \times 512$  Pixeln bei 8 Bit Auflösung verwendet. Bei der Handverfolgung werden aus Geschwindigkeitsgründen die Halbbilder einzeln betrachtet.

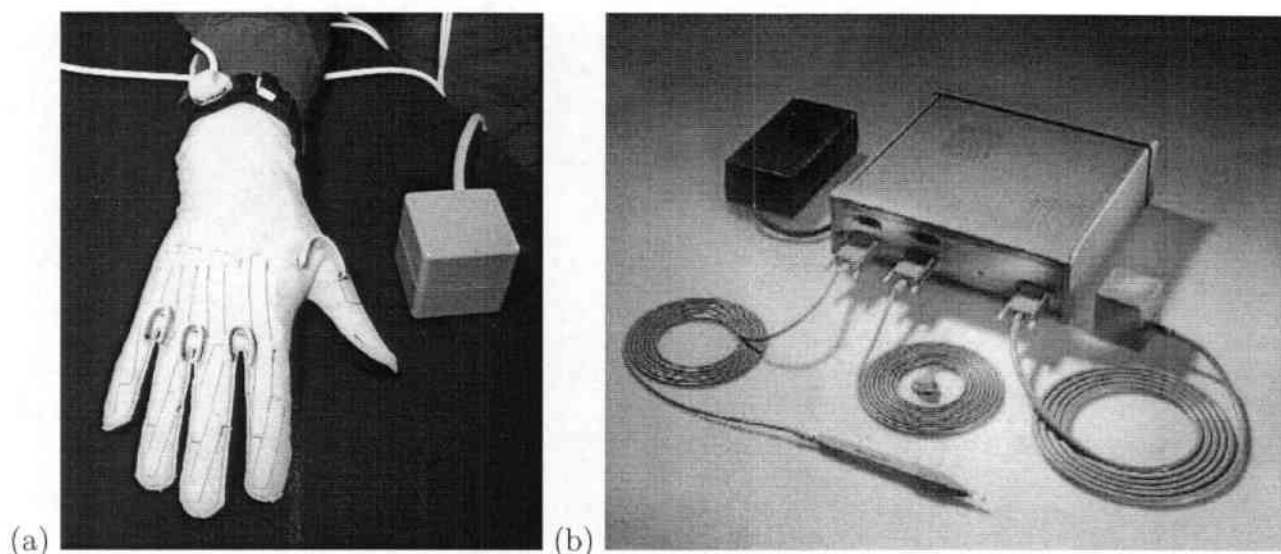


Abbildung 4.3: Ein Datenhandschuh (a) mit magnetfeldbasiertem Positionssystem (b)

### Magnetbasiertes Trackingsystem

Zur Lage- und Orientierungsbestimmung der Hand im 3D-Raum wird ein magnetfeldbasierter Sensor der Firma Polhemus [Polhemus 93] eingesetzt. Er erfasst Lagen und Orientierungen mit sechs Freiheitsgraden, je drei Translations- und drei Rotationsanteile. Der Sensor besteht aus drei Komponenten, einem Sender, einem Empfänger und einer Systemeinheit (siehe Abbildung 4.3 b).

Im Zentrum des Senders befinden sich drei als Sendeantennen wirkende Spulen. Ihre Achsen stehen senkrecht zueinander und bilden das voreingestellte Bezugskoodinatensystem des Sensors. Jede Spule erzeugt bei Betrieb des Senders ein elektromagnetisches Wechselfeld. Diese Felder werden vom Empfänger zur Positionsbestimmung gemessen.

Der Empfänger ist oberhalb des Handgelenks am Datenhandschuh befestigt. Sender und Empfänger sind mit der für die Stromversorgung zuständigen Systemeinheit verbunden. Die von dem Empfänger aufgenommenen Signale werden von dieser Einheit in die entsprechenden Lage- und Orientierungswerte umgewandelt, die über eine serielle Schnittstelle an einen Rechner gesendet werden. Die Abtastrate seitens des Rechners kann mit einer Frequenz von bis zu 120Hz erfolgen.

### Softwarearchitektur

Die Softwaremodule, mit deren Hilfe die in Abschnitt 4.1 vorgestellten Verarbeitungsphasen einer Handlungsvorführung interpretiert und auf einen Roboter abgebildet werden, sind in einem Rechnernetz verteilt. Abbildung 4.4 gibt die Struktur des Gesamtsystems zur Handlungsabbildung wieder<sup>6</sup>. Es besteht aus fünf Komponenten:

**Beobachtung und Segmentierung:** Für die Aufzeichnung, die Analyse und Vorsegmentierung verschiedener sensorischer Eingabekanäle ist ein Sensormodul verantwortlich. Seine Ausgaben sind Vektoren, die den Zustand der beobachteten Objekte in der

<sup>6</sup>Eine genaue Beschreibung findet sich in [Ehrenmann 01b].



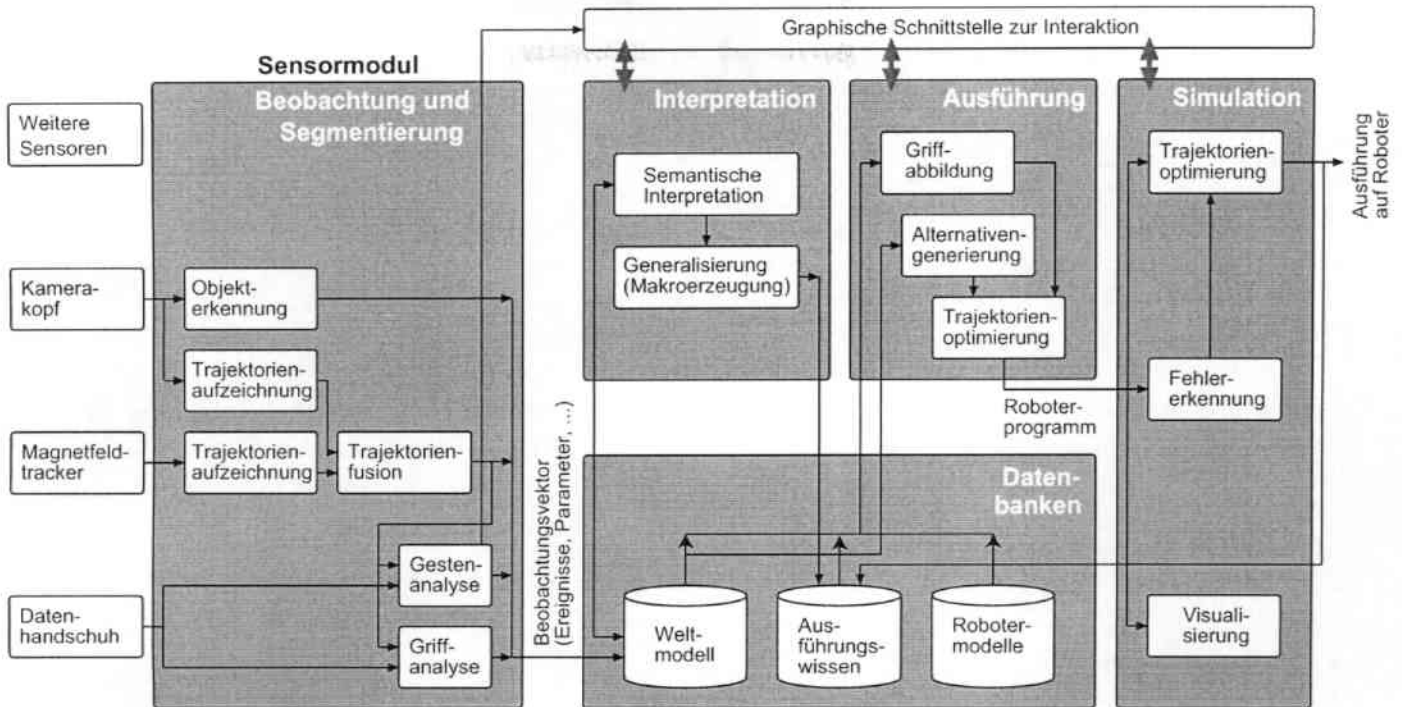


Abbildung 4.4: Softwarearchitektur des Gesamtsystems zum Programmieren durch Vormachen

Vorführungsumgebung und die Benutzerhandlungen parametrisiert beschreiben. Das Sensormodul wurde im Rahmen der vorliegenden Arbeit entwickelt.

**Interpretation:** Das zweite Modul sammelt diese Beobachtungsvektoren und assoziiert eine Menge vordefinierter Symbole zu bestimmten Beobachtungsvektorsequenzen. Diese parametrierbaren Symbole repräsentieren elementare Handlungsteile<sup>7</sup>. Während der Interpretationsphase werden diese Symbole zu hierarchisch gegliederten Makrooperatoren zusammengefasst, nachdem spezifische aufgabenabhängige Parameter durch Variablen ersetzt wurden. Das Ergebnis wird als generalisiertes Ausführungswissen bzw. als prototypische Handlung in einer Datenbank gespeichert.<sup>8</sup>

**Ausführung:** Das Ausführungswissen kann von dem Ausführungsmodul zusammen mit spezifischen Informationen über Robotertypen für die weitere Verarbeitung genutzt werden. Für das jeweilige Zielsystem werden die Bewegungsbahnen an das aktuelle Umweltmodell der Ausführungsumgebung angepasst.

**Simulation:** Bevor das generierte Programm an das Zielsystem zur Ausführung übergeben wird, findet eine offline Validierung auf der Basis einer Simulation statt. Im Falle detektierter Fehler bzw. Kollisionen werden die Bewegungen des Roboters korrigiert.

<sup>7</sup>engl.: Skills

<sup>8</sup>Für eine ausführliche Darstellung dieses und des folgenden Abschnittes sei auf die Arbeit [Friedrich 98] verwiesen.

**Datenbanken:** Bis auf die Datenbanken kommunizieren alle Komponenten über eine interaktive graphische Schnittstelle mit dem Benutzer. Sie dienen zum Speichern des Weltmodells, des gelernten Ausführungswissens sowie der Robotermodelle.

Der Instruierungsprozess erfolgt mit Hilfe dieser Module wie folgt ab (die Beschreibung entspricht derjenigen in Abschnitt 4.1):

**Beobachtung:** Während der Vorführung manipuliert der Benutzer Objekte in der Vorführungsumgebung.

**Segmentierung:** Die Segmentierung der Vorführung erfolgt durch eine Heuristik, die Greif- und Loslassoperationen als spezielle Kontextwechsel interpretiert<sup>9</sup>. Die Größen der gewonnenen Segmente können direkt durch den Benutzer graphisch modifiziert werden. Zusätzlich wird die Bewegungstrajektorie aufgrund der Bewegungsgeschwindigkeiten in Segmente linearer, freier oder spline-förmiger Bewegungen eingeteilt. Diese Aufgaben werden von dem Modul zur Beobachtung und Segmentierung wahrgenommen.

**Interpretation:** Die Zuordnung der Bedeutung der beobachteten Handlungssegmente bzw. der symbolischen Operatoren erfolgt auf Basis der aufgezeichneten Segmente. Jedem Segment wird ein Symbol und ein Parametersatz zugeordnet. So stellt sich beispielsweise ein Greifoperator aus der Beschreibung des Griffes und der Fingerwinkel dar. Eine Linearbewegung enthält Anfangs- und Endpunkt der Bewegung in objektrelativen Koordinaten. Verantwortlich für diese Zuordnung ist das Interpretationsmodul.

**Abstraktion:** In der Generalisierungsphase werden die abgeleiteten symbolischen Operatoren zu semantischen Hierarchien bzw. zu Makrooperatoren zusammengefasst. So stellt sich die Repräsentation eines Greifvorgangs aus einer Anrück-, Greif- und Abrückphase dar. Diesen Phasen werden die jeweils zugehörigen Operationen zugeordnet. Die Bildung der Hierarchie erfolgt, indem die Zusammenfassung semantischer Gruppen rekursiv fortgesetzt wird. Außerdem erfolgt eine Analyse der Nützlichkeit der erfassten Operatoren. Diese Analyse beinhaltet die Betrachtung der Relationen der im Umweltmodell enthaltenen Objekte. Die gewünschten Relationen werden durch Benutzerinteraktion von dem Interpretationsmodul generiert.

**Abbildung:** Das akquirierte Problemlösungswissen dient nun zur Generierung eines speziell für ein Robotersystem zugeschnittenen Programms. Dazu wurden Algorithmen zur Abbildung von Greif- und Bewegungsoperatoren auf das Zielsystem entwickelt. Berücksichtigt werden dabei der jeweilige Aufgabenkontext zum gezielten Einsatz geeigneter Kraftregelungen und die kinematischen Eigenschaften des Robotersystems. Es werden hierzu generische Roboter- und Greifermodelle verwendet [Rogalla 00]. Das resultierende Roboterprogramm besteht aus einer Menge elementarer Bewegungsprimitive für den jeweiligen Roboter, die von dem Ausführungsmodul erzeugt werden.

**Simulation und Ausführung:** Das generierte Roboterprogramm kann nun in einer Simulationsumgebung getestet und bewertet werden. Auch hier besteht die Möglichkeit, interaktiv Positionen, Greifpunkte und Bewegungsbahnen zu korrigieren. Die letzte Phase

---

<sup>9</sup>Siehe dazu wieder [Friedrich 98].

besteht in der Übertragung des getesteten Programmes auf den realen Roboter in der Ausführungsumgebung.

Die Architektur ist für Erweiterungen offen gehalten. Zur Verwendung weiterer Sensoren wie z.B. haptischer Sensoren für die Beobachtung oder weiterer Robotertypen für die Ausführung erfordert keine Änderung der Struktur.

### 4.3.2 Ausführungsumgebung zur Interaktion

Die Vorführungsumgebung dient der hochgenauen Verfolgung und Aufzeichnung von Benutzerhandlungen. In der Ausführungsumgebung dagegen muss der Roboter kommandiert und über Kommentierungen belehrt werden. Hierzu sollen keine Landmarken bzw. Referenzkoordinatensysteme oder invasive<sup>10</sup> Sensoren Verwendung finden. Zur online Beobachtung wird ein Farbsichtsystem eingesetzt.

#### Mobiles aktives Sichtsystem

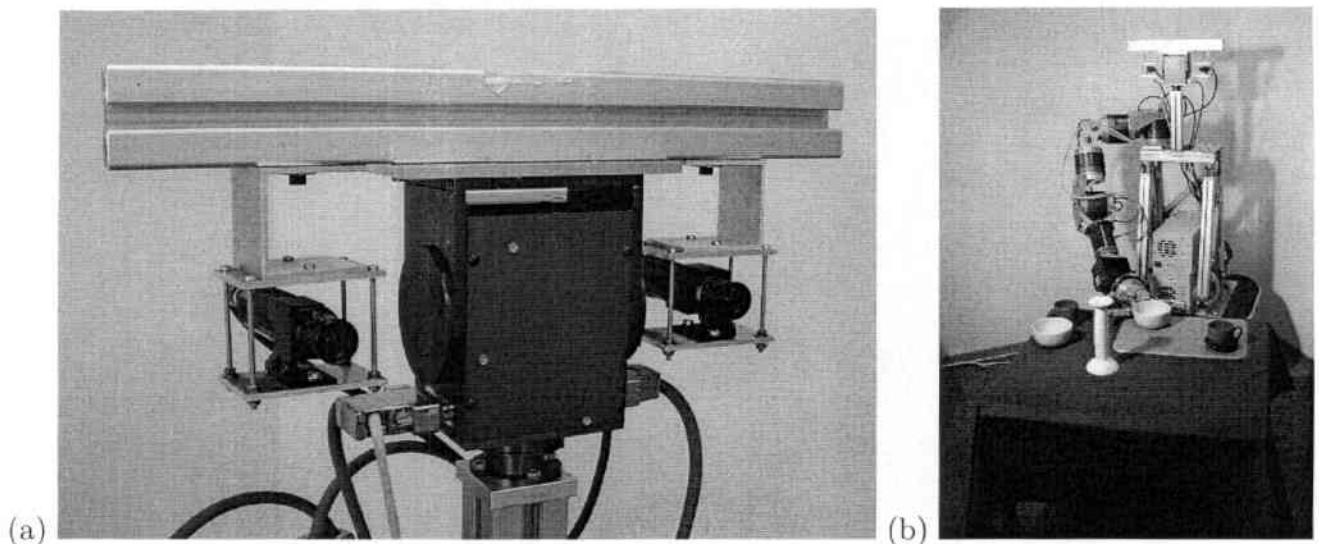


Abbildung 4.5: Binokulares aktives Sichtsystem (a Detailaufnahme) des Robotersystems *Albert* (b)

Der Roboter muss in der Lage sein, die vor ihm auf einer Tischplatte liegenden Objekte zu klassifizieren und ihre Lage zu bestimmen. Nur so ist die Überprüfung der Anwendungsbedingungen eines vorher generierten Makrooperators möglich. Außerdem muss der Ort der Benutzerhand festgestellt und ausgeführte Gesten erkannt werden. Zur Bewegung des Kamerakopfes mit zwei Bewegungsfreiheitsgraden wird ein schwenk- und neigbares Gelenk der Firma Amtec [Amtec 00]<sup>11</sup> genutzt. Die Bildsensorik besteht aus zwei Sony AP777 Farb-

<sup>10</sup> „invasiv“ im Sinne von: „vom Benutzer anzulegen“

<sup>11</sup> technische Daten siehe Tabelle B.2 im Anhang B

kameras<sup>12</sup>. Zur Digitalisierung wird jeweils ein MeteorII-Framegrabber von Matrox verwendet [Matrox 98c]. Abbildung 4.5 zeigt den Aufbau des realisierten Systems.

### Softwarearchitektur

Wie im Fall der Vorführungumgebung erfolgt die Informationsverarbeitung des Roboters verteilt in einem Rechnernetz. Abbildung 4.6 gibt einen Überblick über die verwendete Struktur des Gesamtsystems. Es enthält vier grundlegende Module, wobei das Datenbankmodul dieselbe Struktur und denselben Inhalt wie in der Vorführungumgebung besitzt:

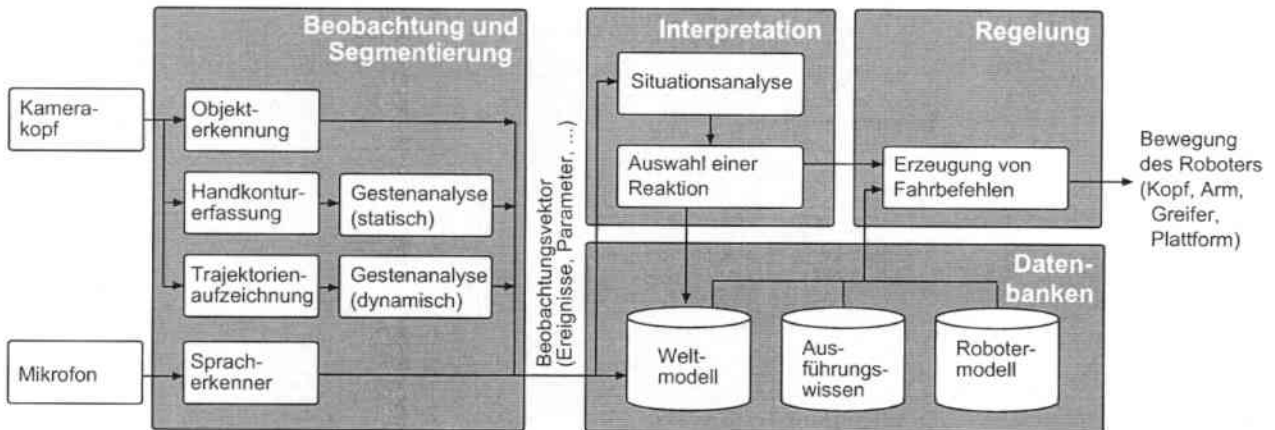


Abbildung 4.6: Softwarearchitektur des Gesamtsystems in der Ausführungsumgebung

**Beobachtung:** Das Sensormodul ist hier ebenso wie in der Vorführungumgebung verantwortlich für die Verfolgung der Benutzeraktionen sowie für die Analyse der Szene.

**Interpretation:** Die Interpretationskomponente analysiert die Situation anhand der beobachteten Szene, die im Weltmodell gespeichert ist. Sie wählt eine Reaktion auf eine beobachtete Situation oder Handlung hin aus. Das kann beispielsweise die Ausführung einer durch Vormachen programmierten Befehlssequenz sein.

**Regelung:** Die Regelung der gesamten Roboteraktuatorik unterliegt dem Regelungsmodul.

## 4.4 Systemarchitektur

Handlungsrelevante Beobachtungen wie manipulierbare Objekte, Griffe, Gesten oder Handtrajektorien werden zur Interpretation der Vorführung im Weltmodell gespeichert. Da die Methoden zur Detektion dieser Ereignisse nicht von einem einzigen Rechner bewältigt werden können, wird das Softwaremodul zur Beobachtung und Segmentierung auf mehrere Recheneinheiten verteilt. Abbildung 4.7 zeigt den Aufbau des physischen Systems mit der Vorführungumgebung und dem Roboter *Albert*.

<sup>12</sup>technische Daten siehe Tabelle B.1 im Anhang B. Verwendet wird hier die halbe PAL-Auflösung, d.h. 320 × 240 Pixel bei 3 × 8 Bit Farbtiefe

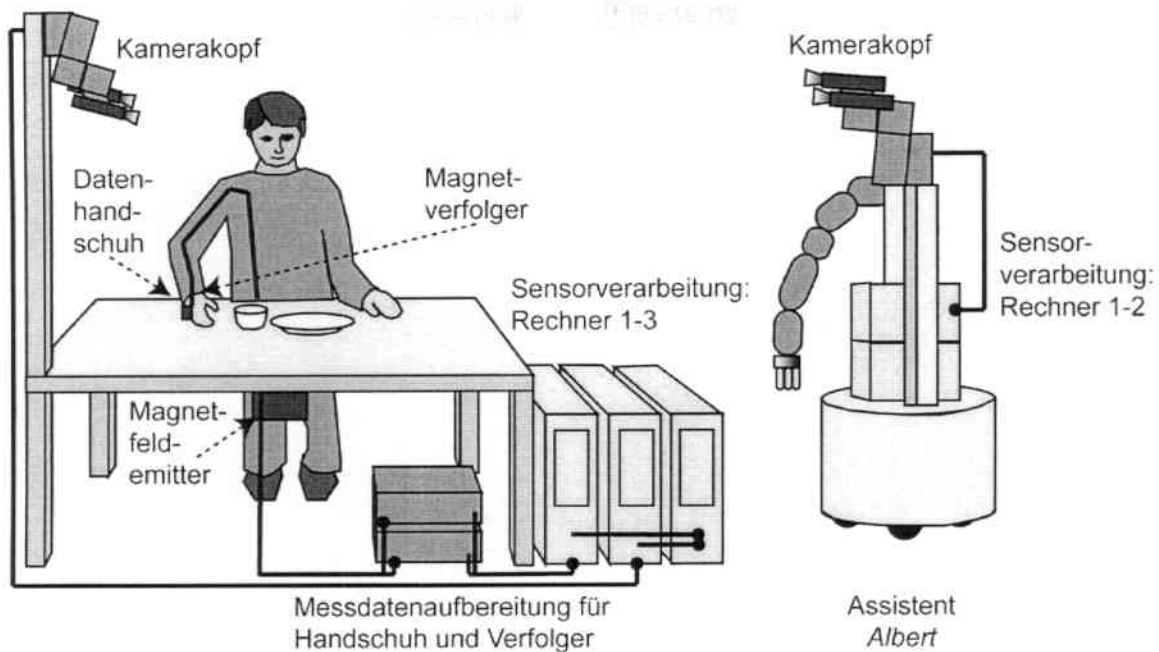


Abbildung 4.7: Aufbau des Beobachtungssystems in der Vorführungsumgebung (links) und in der Ausführungsumgebung (rechts)

**Vorführungsumgebung:** Zur Sensordatenaufnahme und Fusionierung der Sensordaten sind drei Rechner notwendig:

- Ein PC regelt die Pose des Kamerakopfes und verfolgt mit diesem die Benutzerhand auf der Basis eines lokalen Trackingverfahrens. Er leistet auch die Erkennung manipulierbarer Objekte in der Szene.
- Ein weiterer PC nimmt die vom Datenhandschuh gelieferten Messwerte für die Benutzerhand auf und dient der Gesten- und Grifferkennung. Zudem werden hier die Handposen aus dem magnetfeldbasierten Tracker mit den vom ersten PC verschickten Positionsschätzungen fusioniert.
- Das Weltmodell wird durch einen dritten PC verwaltet. Hier werden die gemeldeten Ereignisse entgegengenommen und gespeichert.

**Ausführungsumgebung:** Zur Regelung des Kamerakopfes des Roboters dient ebenfalls ein PC. Dieser ist auch für die Verfolgung der Benutzerhand und die Gestenerkennung verantwortlich. Das Weltmodell und Ausführungswissen liegen auf einem zweiten Rechner, an den die gemachten Beobachtungen gemeldet werden müssen.

## 4.5 Modellierung

Der folgende Abschnitt ist der Vorstellung der Modellierung der Szene und ihrer zeitlichen Veränderung gewidmet. Die entsprechenden Daten werden in der Datenbank des „Weltmodells“<sup>13</sup> gespeichert und für den Interpretationsprozess bereitgestellt. Dabei wird auf die

<sup>13</sup>Zur formalen Definition des Begriffs „Weltmodell“ siehe Anhang A



Modellierung des Benutzers („Handmodell“) und dessen Verhalten in der Szene („Handlungsmodell“) eingegangen, das ebenfalls dort gespeichert wird. Entsprechend dieser Modellierung werden Vorführungen von den einzelnen Modulen bearbeitet. Dies gilt in gleicher Weise für die Vorführungsumgebung wie auch für die Ausführungsumgebung.

### 4.5.1 Weltmodell

Die Modellierung von Objekten, Relationen und Ereignissen<sup>14</sup> im Rahmen einer Roboteranwendung erfordert als Ausgangspunkt die Modellierung ihrer Geometrie sowie ihrer technischen und physikalischen Eigenschaften.

Geometrische Objektmodelle, die zur Repräsentation des Weltmodells im vorgeschlagenen Beobachtungssystem genutzt werden, können sowohl durch oberflächenbasierte als auch durch auf parametrisierten Volumenprimitiven basierende Modelle realisiert werden. Die Objekte einer Umwelt werden dazu mit einem CAD-Systemen modelliert und dann in das für diese Arbeit verwendete *Open Inventor* [Open Inventor Architecture Group 94] Dateiformat übertragen. Die genutzte Visualisierungs- und Simulationsumgebung KAVIS<sup>15</sup> stellt auf der Basis der geometrischen Objektmodelle Funktionen für alle zur interaktiven Programmierung und Simulation erforderlichen geometrischen Berechnungen, wie etwa Kollisionsbetrachtungen, Transformationen u. Ä. zur Verfügung.

Eigenschaften von Objekten wie etwa Gewicht, Oberflächenrauheit oder Flexibilität können sinnvoll durch Attribut-/Wertpaare repräsentiert werden. Daher sind im vorgeschlagenen System alle ein Objekt bzw. eine Objektklasse charakterisierenden Eigenschaften durch eine Liste solcher Attribut-/Wertpaare modelliert. Im Rahmen der vorliegenden Arbeit sind dabei vor allem zwei Attributtypen von Bedeutung:

- Die Erkennung der manipulierbaren Objekte in der Vorführungs- oder Ausführungsumgebung ist bei bildbasierten Verfahren über die Kameras nur mit Hilfe spezieller Modelle (Objektansichten, Konturmerkmale, ...) zu leisten. Die spezifische Eignung einer Methode zur Detektion eines Objekts wird zusammen mit dem entsprechenden Modell in einem Attributtyp gespeichert.
- Zur Stabilisierung der Griffklassifikation lässt sich a-priori Wissen über mögliche Griff-typen nutzen. Beispielsweise kann ein kugelförmiges Objekt nicht mit einer für die Aufnahme eines Stiftes passenden Handkonfiguration gegriffen werden. Für ein Objekt sinnvolle Griffe werden deshalb in dessen Attributliste aufgenommen, um sensorische Messungen auf ihre Plausibilität hin überprüfen zu können.

Abbildung 4.8 zeigt eine Auswahl der betrachteten Objekte und einen Ausschnitt aus deren Attributlisten. Die Nutzung dieser Information wird in den entsprechenden Abschnitten 5.1 und 5.3 zur Objektdetektion und Griffenerkennung erläutert. Auf die Modellierung der Greifklassen für ein Objekt wird in Abschnitt 4.6.2 näher eingegangen.

<sup>14</sup>Zur formalen Definition des Begriffs „Ereignis“ siehe Anhang A

<sup>15</sup>KARlsruher VISualisierer, siehe [Schaude 96].

"Schüssel"  
 Konturbasierte Erkennung  
 Greifbar mit Grifftyp 1, 6, 7, 8, 11

"Kerzenständer"  
 Konturbasierte Erkennung  
 Greifbar mit Grifftyp 2

"Pinsel"  
 Ansichtsbasierte Erkennung  
 Greifbar mit Grifftyp 6, 7, 8, 9, 16

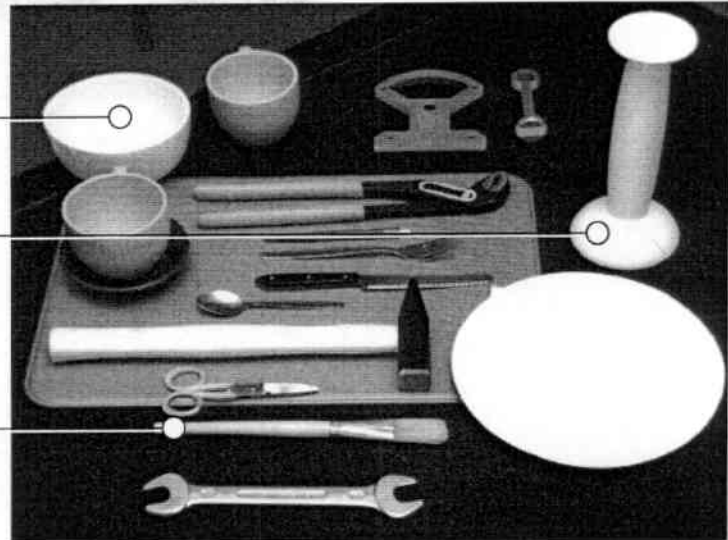


Abbildung 4.8: Objekte und Attribute

## 4.5.2 Handmodell

Entsprechend den Randbedingungen aus Abschnitt 4.2 sollen alle Eingriffe des Benutzers in der Szene per Hand erfolgen. Es ist daher ausreichend, zur Erfassung von Demonstrationen eine Hand zu modellieren. Diese ist ebenso wie die Modelle der manipulierbaren Objekte als geometrisches Oberflächenmodell in *Open Inventor* definiert. In Abbildung 4.10 sind dessen Elemente Unterarm, Handrücken und die Fingerelemente dargestellt. Mit Hilfe dieses Modells lassen sich die Messwerte des Datenhandschuhs visualisieren und Abstands- bzw. Kollisionsberechnungen zu Objekten durchführen. Dies ist für eine genauere Betrachtung von Griffen notwendig, beispielsweise zur Bestimmung von Berührungspunkten auf Objekten.

## 4.5.3 Handlungsmodell

Flexible Roboterprogramme lassen sich aus einer Vorführung nur dann erzeugen, wenn sie nicht ausschließlich quantitativ erfasst werden und auch qualitativ beschreibbar sind. Ziel des vorgestellten Systems ist es, eine parametrisierte symbolische Beschreibung eines Manipulationsvorgangs zu erfassen, die diesen so beschreibt, dass sie für die Generierung eines Roboterprogramms nutzbar ist. Dazu sind die relevanten Phasen der Demonstration zu segmentieren und zu identifizieren. Spezielle Rahmenbedingungen erleichtern die Verfolgung einzelner Aktionen. Beispielsweise braucht nicht nach greifbaren Objekten gesucht werden, wenn der Benutzer bereits ein Objekt in der Hand hält und kein weiteres greifen kann.

Eine allgemeine Ablaufbeschreibung für Benutzerhandlungen lässt sich in Form eines endlichen Automaten notieren. Abbildung 4.9 zeigt das zur Erfassung von Vorführungen verwendete Modell. Zu Beginn müssen die in der Szene befindlichen Objekte identifiziert und lokalisiert werden. Danach kann die Benutzerhand gesucht und verfolgt werden. Greift der Benutzer ein Objekt, muss die Bewegung der Hand bis zum Ablegen weiterverfolgt werden. Der Ablageort wird bestimmt und danach die Hand weiterverfolgt. Erkannte Gesten lösen

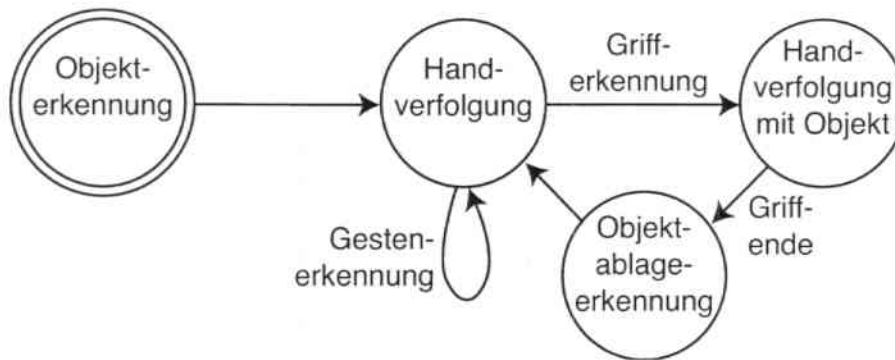


Abbildung 4.9: Handlungsmodell in der Formulierung eines endlichen Automaten

währenddessen arbiträre Funktionen aus, können beispielsweise den Kamerakopf dirigieren oder zur Auslösung der Handschuhkalibrierung dienen.

Die Vorführung selbst wird durch das Handlungsmodell überlappungsfrei und lückenlos in seine wesentliche Bestandteile untergliedert. Die in Abschnitt 2.4.1 vorgestellten Systeme zum Programmieren durch Vormachen gehen von einer Beobachtungsphase mit einer a posteriori Interpretation aus. Im Gegensatz dazu soll eine vorläufige Segmentierung und damit die Erkennung der Handlungsfolge nun in vivo erfolgen. Dafür lassen sich mehrere Argumente angeben:

- Bei der Verwendung bildverarbeitender Methoden ist die Beobachtung selbst sehr rechenaufwändig—nur wenige Methoden sind zur Verarbeitung der Sensorinformation gleichzeitig nutzbar. Es kann beispielsweise nicht zur selben Zeit neben der Objekterkennung eine Methode zur Handverfolgung ablaufen, da der Kamerakopf hierfür unterschiedlich positioniert sein müsste. Ein Aufmerksamkeitsfokus sollte der Situation entsprechend eine Auswahl der im Moment notwendigen Methodenaufrufe bestimmen. Dafür ist eine Vorinterpretation notwendig.
- Da ein Teil der Sensorik ständig aktiv ist, muss für diese der relevante Szenenausschnitt zur Beobachtung festgelegt werden. Dies geschieht in Abhängigkeit des Handlungsmodells. Beispielsweise richtet sich der Kamerakopf zu Beginn einer Vorführung auf die Szene, um die manipulierbaren Objekte zu registrieren<sup>16</sup>. Danach wird begonnen, die Benutzerhand zu verfolgen.
- Durch die Vorinterpretation werden Rückfragen während der Vorführung möglich. Dadurch kann der aufwändige Schritt der Korrektur der Messung stark beschleunigt werden, ohne dass der Benutzer den Kontext wechseln müsste.

## 4.6 Beobachtung elementarer Handlungen

Die dem Handlungsmodell zu Grunde liegenden Benutzeraktionen müssen erkannt und klassifiziert werden. Die Detektion und Parameterschätzung dieser Handlungen wird von elemen-

<sup>16</sup>Zur formalen Definition des Begriffs „Registrierung“ siehe Anhang A

taren kognitiven Operatoren<sup>17</sup> geleistet. Diese bestimmen gleichzeitig den Kontext für die Beobachtung der Folgehandlung entsprechend der Festlegung im Handlungsmodell. Der folgende Abschnitt beschreibt die Aufgabe dieser Operatoren und die Klassifikationsschemata für die entsprechenden Handlungstypen.

### 4.6.1 Handverfolgung

Die Verfolgung von Bewegungen der Benutzerhand hat in der Vorführungsumgebung das Ziel, die entsprechenden Trajektorien möglichst genau im Weltmodell zu registrieren und später für eine Ausführung auf einem Roboter nutzbar zu machen. Interessant ist also die Repräsentation der Verfahrbahn mit Bezug auf das Endeffektorkoordinatensystem<sup>18</sup>, das üblicherweise mitten zwischen den Greiferfingern angenommen wird. Dieses wird hier entsprechend [Friedrich 98] in das Zentrum der Handfläche gelegt (siehe Abbildung 4.10 b). Unterstützt wird die Genauigkeitsschätzung durch das Setzen des Positionsmarkers und des Magnetfeldempfängers auf den Handrücken<sup>19</sup>.

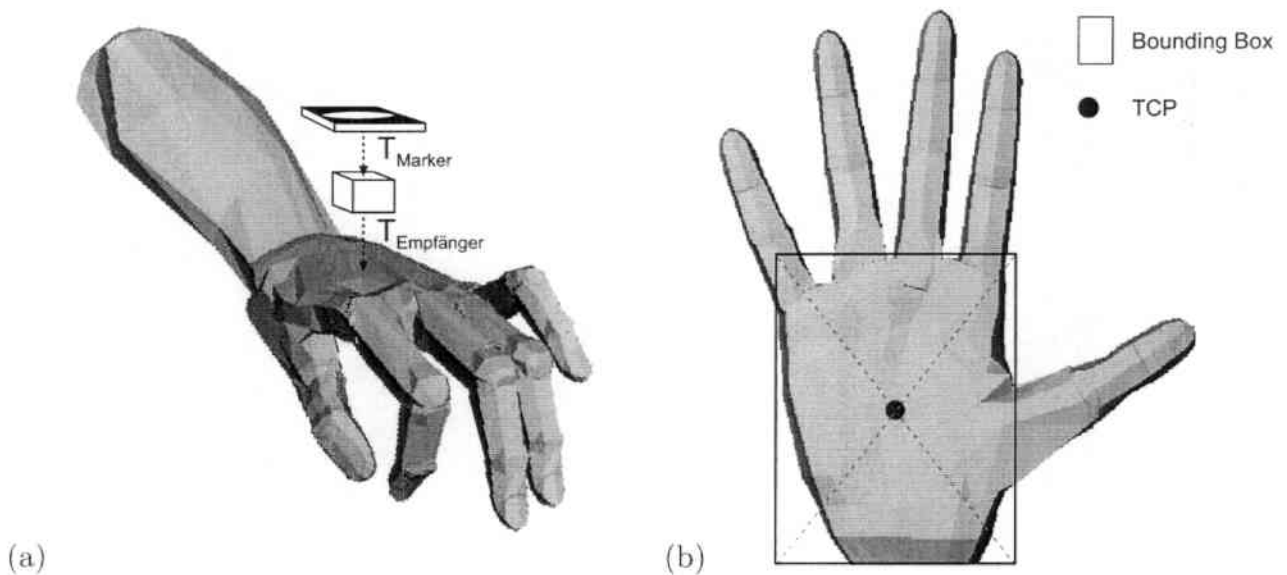


Abbildung 4.10: Transformationen der Messkoordinatensysteme im Handmodell (a) und Ort des Endeffektors (*TCP*, b)

Die Bewegungstrajektorie  $T$  des Unterarms und der Hand ist nach der Aufnahme durch eine Folge von Transformationen  $T = \{T^1, \dots, T^n\}$  gegeben. Diese Daten werden den Gleichungen 4.1, 4.2 entsprechend in das Weltkoordinatensystem des Umweltmodells transformiert. Dabei ist je nach Sensortyp die Transformationsgleichung  $T'$  für die bildbasierte oder  $T''$  für die magnetfeldbasierte Verfolgung anzuwenden:

$$T'_{\text{Welt, TCP}} = T_{\text{Welt, Kamerakopf}} \cdot TR_{\text{Kamerakopf, Kamasas}}$$

<sup>17</sup>Zur formalen Definition des Begriffs „elementarer kognitiver Operator“ siehe Anhang A

<sup>18</sup>engl.: Tool Center Point (TCP)

<sup>19</sup>Der Hersteller sieht beim Datenhandschuh Positionsmessungen am Unterarm vor. Da die gemessenen Winkelstellungen zwischen Handrücken und Unterarm jedoch um mehrere Grad differieren, wurden hier Fehler bei der Positionierung der Fingerkuppen von bis zu zwei Zentimetern gemessen.

$$T_{\text{Kameras, Marker}} \cdot T_{\text{Marker, TCP}} \quad (4.1)$$

$$T''_{\text{Welt, TCP}} = T_{\text{Welt, Sender}} \cdot T_{\text{Sender, Empfänger}} \cdot T_{\text{Empfänger, TCP}} \quad (4.2)$$

Vom Ursprung des Weltkoordinatensystems wird hier zunächst die Verschiebung zum jeweiligen Posensensorkoordinatensystem berücksichtigt, um dann weiter zu verfahren:

**Kamerakopf ( $T'$ ):** Bei dem Kamerakopf müssen beim Übergang zu den Bezugskoordinatensystemen der einzelnen Kameras die rotatorischen Achszustände der Drehgelenke berücksichtigt werden. Diese Koordinatensysteme liegen im Zentrum der CCD-Bildaufnehmer. Die Transformation von dort aus zur Basis des Markerkoordinatensystems wird durch Triangulation rekonstruiert. Die Basis des TCP-Koordinatensystems wird dann durch eine Translation erreicht.

**Magnetfeldtracker ( $T''$ ):** Die Transformation von der Basis des Bezugskoordinatensystems des Magnetfeldsenders zum Empfänger wird durch den Sensor geliefert. Durch eine Translation um einen festen Verschiebungsvektor gelangt man zum Endeffektorkoordinatensystem.

Abbildung 4.11 stellt die Bezugskoordinatensysteme der Transformationsketten für die Vorführungsumgebung aus Gleichung 4.1 und 4.2 grafisch dar.

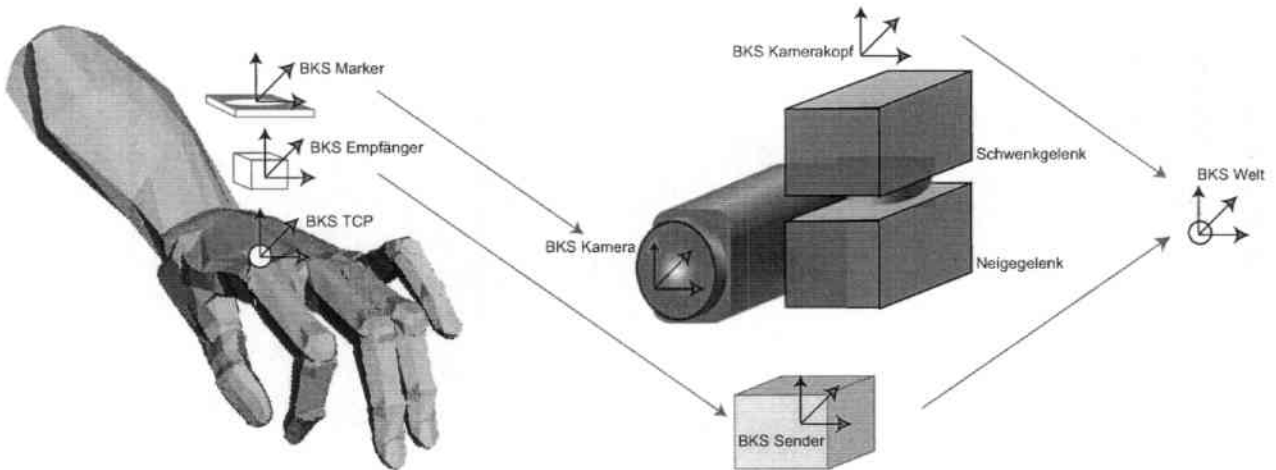


Abbildung 4.11: Bezugskoordinatensysteme in der Vorführungsumgebung

Zu jedem Zeitpunkt  $t$  einer Posenmessung wird damit die Lage des Endeffektorkoordinatensystems durch Verkettung mehrerer Koordinatentransformationen bestimmt. Das Ergebnis dieses Verarbeitungsschrittes ist nach der Sensordatenfusion<sup>20</sup> die im Verlauf einer Demonstration im Endeffektorkoordinatensystem beschriebene Trajektorie

$$T_{\text{TCP}} = \{T_{\text{Welt, TCP}}^1, \dots, T_{\text{Welt, TCP}}^n\} \quad (4.3)$$

Für die Ausführungsumgebung gilt das für die bildbasierte Positionsrekonstruktion Gesagte entsprechend. Anstelle eines Markers wird hier der Schwerpunkt<sup>21</sup> von als zur Benutzerhand

<sup>20</sup>Die Fusion von magnetfeldbasierten und bildbasierten Positionsschätzungen wird in Abschnitt 5.2.1 erläutert.

<sup>21</sup>engl.: Center of Gravity



gehörend identifizierten Bildregionen zu Grunde gelegt, Gleichung 4.1 wird daher abgewandelt zu:

$$T'_{\text{Welt, TCP}} = T_{\text{Welt, Kamerakopf}} \cdot TR_{\text{Kamerakopf, Kameras}} \cdot T_{\text{Kameras, CoG}} \cdot T_{\text{CoG, TCP}} \quad (4.4)$$

Grafisch stellen sich die entsprechenden Bezugskoordinatensysteme wie in Abbildung 4.12 dar. Auch hier müssen die Stellungen der Drehgelenke im Übergang von dem Bezugskoordinatensystem des Roboterkamerakopfes zu den einzelnen Kameras berücksichtigt werden. Der Ursprung des Endeffektorkoordinatensystems wird im rekonstruierten Raumpunkt des Handregionenschwerpunktes angenommen.

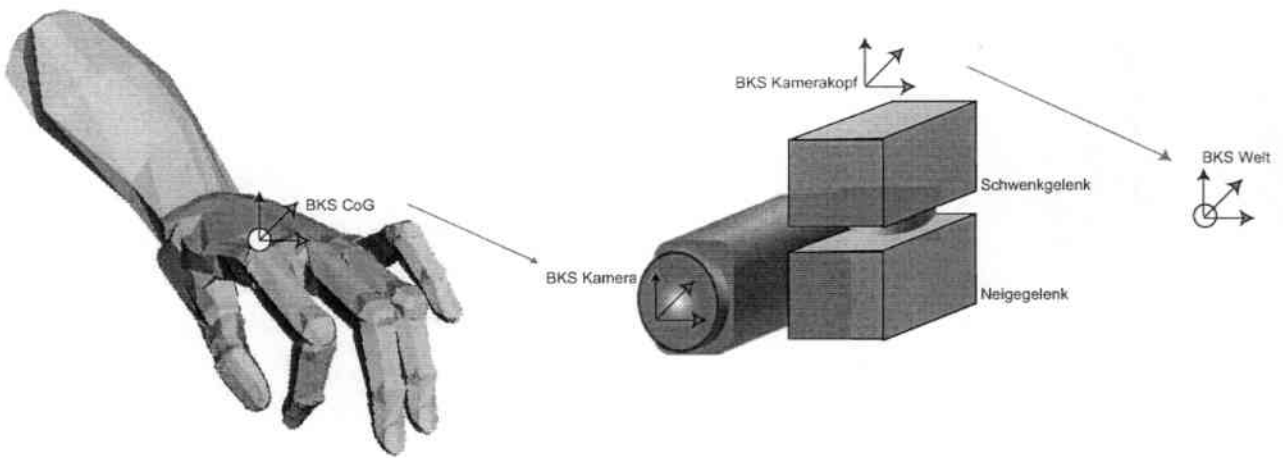


Abbildung 4.12: Bezugskoordinatensysteme in der Ausführungsumgebung

### 4.6.2 Grifferkennung

Bei der Manipulation von Objekten wenden Menschen je nach deren Material, Gewicht und Form sowie dem Ziel ihrer Manipulationen eine Vielzahl verschiedener Griffe an. Es gibt eine Reihe verschiedener Griff-taxonomien auf der Basis unterschiedlicher Klassifikationshierarchien [Kang 94, MacKenzie 94]. Die meisten dieser Ansätze gehen auf die Arbeiten von Schlesinger [Schlesinger 19] und die weitergehenden Analysen von Napier [Napier 56] zurück.

Da die Anwendung des geeignetsten Griffs zur beabsichtigten Manipulation ein wichtiger Faktor ist, kommt der Wahl der verwendeten Taxonomie große Bedeutung zu. Diejenige von Cutkosky [Cutkosky 89] basiert ebenfalls auf Schlesingers und Napiers Arbeiten. Sie ist in Abbildung 4.13 dargestellt. Die grundlegende Unterscheidung bei der Betrachtung eines Griffes ist die zwischen Kraftgriffen (zur Manipulation schwerer Objekte oder dem schnellen Verfahren) und Präzisionsgriffen (zur geschickten Manipulation bzw. der Handhabung kleiner Objekte). Cutkoskys Griff-taxonomie orientiert sich an Griffen, die bei handwerklichen Tätigkeiten typischerweise auftreten. Deshalb erscheint sie für den Bereich von Serviceanwendungen im Haushalt geeignet. Sie wurde als Grundlage zur Griffmodellierung

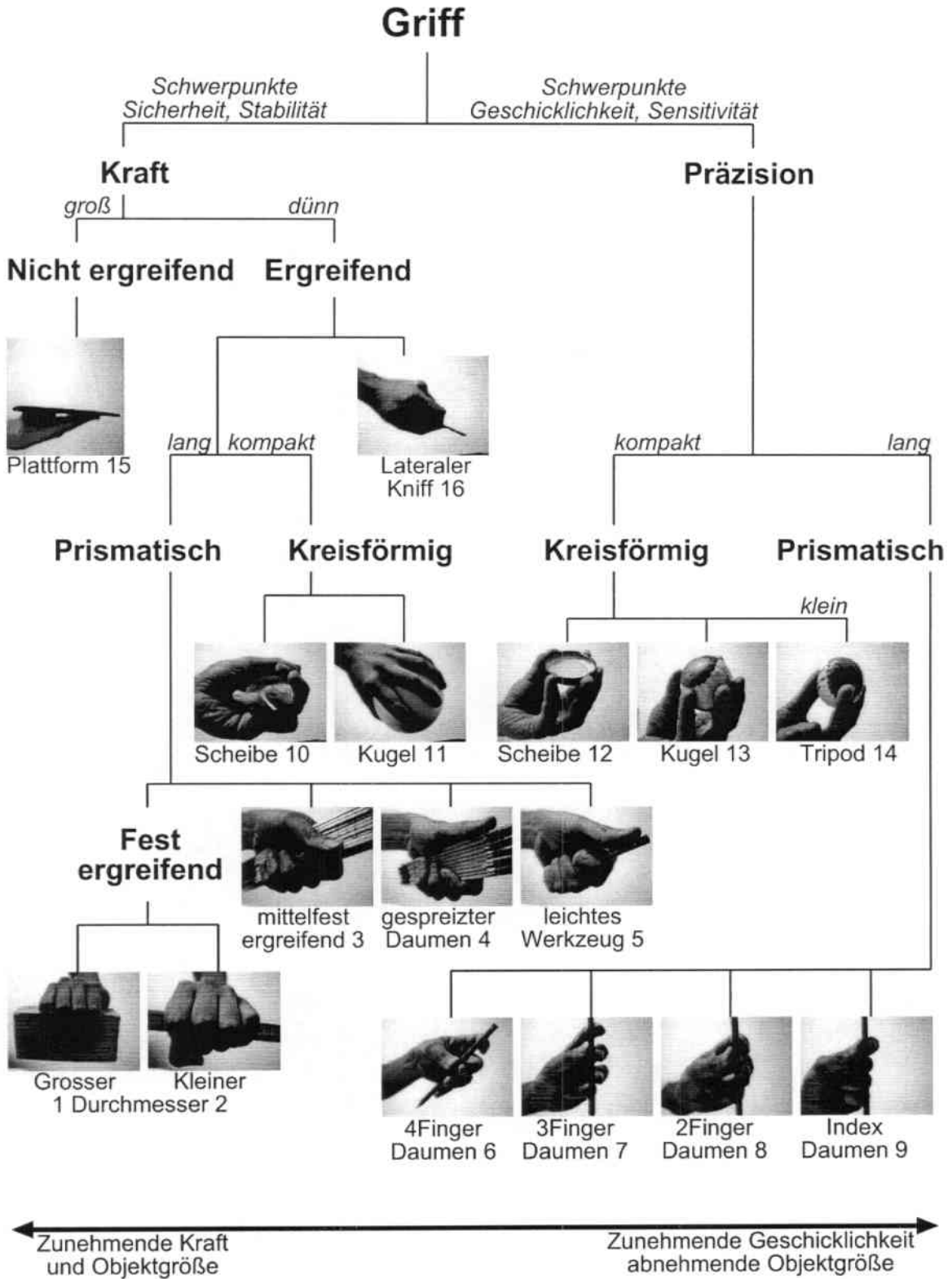


Abbildung 4.13: Griff-taxonomie nach Cutkosky zur Klassifikation von Griffen, die bei Handhabungen auftreten (nach [Cutkosky 89])

bereits in der Arbeit von Friedrich [Ehrenmann 98, Friedrich 98] ausgewählt.

Es ist hier zu bemerken, dass die Klassifikation von Griffen nach dieser Taxonomie nicht allein aufgrund der Handkonfiguration erfolgt. Stattdessen werden geometrische Eigenschaften der manipulierten Objekte in die Rubriken aufgenommen, beispielsweise bei den Griffotypen 12 und 13, präzisen Griffen zum Handhaben von scheiben- und kugelförmigen Gegenständen. In Tabelle 4.1 wird der Zusammenhang zwischen sechs in der Cutkosky-Hierarchie isolierbaren Objektgeometrien und deren Griffotypen festgestellt. Es ist jedoch festzustellen, dass Objekte komplexer Struktur oft mehrere der genannten geometrischen Merkmale aufweisen und daher in mehrere dieser Klassen fallen.

Objektgeometrie	Anwendbare Griffe
großes Objekt mit glatter Oberfläche	15
längliches Objekt geringen Durchmessers	5, 6, 7, 8, 9, 16
scheibenförmiges Objekt	10, 12, 14
kugelförmiges Objekt	11, 13, 14
längliches Objekt großen Durchmessers	1
längliches Objekt mittelgroßen Durchmessers	2, 3, 4

Tabelle 4.1: Korrespondenzen zwischen Objektgeometrie und Griffotypen

Im Hinblick auf eine robustere Grifferkennung werden in der Modellierung daher die manipulierbaren Objekte mit einem Attribut ausgestattet, das in einer Liste die für dieses Objekt möglichen Griffotypen aufzählt. Dieses Attribut beschreibt implizit Eigenschaften der Objektgeometrie und dient dazu, die Zahl anwendbarer Griffotypen für ein spezifisches Objekt einzuschränken.

### 4.6.3 Gestenerkennung

Wie für Griffe existieren eine Reihe von Klassifikationsschemata für Handzeichen und Gestensprache [Mandel 77, Rime 91, Badler 00]. Eine Aufzählung, die den Anspruch erhebt, einzelne Handzeichen vollständig aufzuzählen, wie das bei der Cutkosky-Hierarchie der Fall ist, gibt es nicht. Diejenigen Handbewegungen, die zur Kommunikation zwischen Menschen dienen können, hat McNeil jedoch hinsichtlich ihres zeichenhaften Gebrauchs umfassend kategorisiert [McNeil 92]:

**Ikonisch:** Hier werden Eigenschaften wie die Form oder Größe des Kommunikationsgegenstandes repräsentiert. Dies geschieht durch eine bildhafte Beschreibung, beispielsweise durch Abfahren der gedachten Objektkontur.

**Metaphorisch:** Dieser Gestentyp dient zur Repräsentierung abstrakter Eigenschaften wie Austausch oder Gebrauch.

**Deiktisch:** Die Referenzierung von Objekten im Raum oder die Angabe von Richtungen bedient sich der Zeigegesten.

**Takt:** Zur Unterstützung der gesprochenen Sprache werden manche Wörter durch Handbewegungen unterstrichen. Dies geschieht vor allem kurz vor dem Sprecherwechsel.

**Embleme:** Stereotypen mit bekannter Semantik wie das Zeichen für „OK“ oder der nach oben gestreckte Daumen.

In jeder Kategorie gibt es ein- und zweihändig ausgeführte Gesten. Für den Einsatz in Mensch-Maschine-Schnittstellen sind bislang vor allem Embleme und deiktische Gesten studiert worden. Da die Erkennung kontinuierlicher Sprache noch nicht zufriedenstellend für akustisch gestörte Szenarien gelöst ist, finden Taktgesten bislang kaum Beachtung. Im Rahmen der Instruktion von Robotern ist ein eingeschränktes Gestenvokabular ausreichend. Da ikonische Gesten implizit durch Bewegungsbahnen im Rahmen einer Vorführung gegeben werden, sollen sie nicht explizit klassifiziert werden. Weiter können die metaphorischen und emblematischen im selben Situationskontext ausgeführt werden und deshalb wie in [Marsh 98] fortan als symbolische Gesten zusammengefasst werden. Es fällt auf, dass in den einzelnen Kategorien zum Teil die Handstellung, zum Teil die Bewegung für einen Gestentyp charakteristisch sein kann. Für die automatische Klassifikation von Gesten erscheint daher die Unterscheidung statischer und dynamischer Gesten sinnvoll:

**Statische Gesten:** Bedeutungstragend ist hier die Stellung der Handkonfiguration, die sich während der Interpretationsphase nicht ändert. Dies ist beispielsweise bei manchen symbolischen Gesten wie dem nach oben gerichteten Daumen der Fall. Genutzt werden statische Gesten vor allem bei kleiner Distanz zwischen den Interaktionspartnern. Bei der Ausführung lässt sich beobachten, dass Menschen zur Hervorhebung die gestenzeigende Hand einen kurzen Moment an einer festen Position verharren lassen.

**Dynamische Gesten:** Hier ist die Trajektorie der Handbewegung bedeutungstragend. Diese wird bei den meisten ikonischen, silhouettenbedeutenden Gesten genutzt. Dynamische Gesten werden zur Überbrückung grösserer Entfernungen eingesetzt, z. B. in der Schiff- und Luftfahrt oder im Militärwesen.

Dementsprechend werden für die in der obigen Liste aufgeführten Typen Methoden zur Klassifikation statischer oder dynamischer Ausprägung benötigt. Verwendet werden sollen diese vorrangig in Dialogen bei Rückfragen, zur Kommandierung des Assistenten oder Kommentierung während einer Demonstration sowie zur Referenzierung von Objekten. Dabei kommen zur Kommandierung in grösserer Entfernung zum Roboter dynamische Gesten zum Einsatz, wenn die Handkonfiguration aus Kamerabildern nicht mehr erschlossen werden kann.

Die Nutzung der Gesten liegt im Rahmen von Dialogen. Sie werden außerdem für Objektreferenzierungen eingesetzt oder dienen zur Kommandierung. Für eine intuitive Bedienung mit möglichst kurzer Einlernzeit wurden die in Tabelle 4.2 aufgeführten statischen Gesten ausgewählt. Sie zeichnen sich durch eine geringe Ähnlichkeit in ihrer Gestalt aus. In Dialogen

dienen einfache symbolische Gesten zum Antworten mit „Ja“ oder „Nein“ (Faust mit nach oben bzw. unten ausgestrecktem Daumen). Zum Anhalten von in der Ausführung befindlichen Prozessen wird die Stopp-Geste (flache, nach oben gerichtete Hand) verwendet. Zur Bezugnahme auf Objekte oder zum Anzeigen von Richtungen dient die Zeigegeste (Faust mit ausgestrecktem Zeigefinger). Zum Auslösen arbiträrer Roboterfunktionen können mehrere Kommandogesten (Faust, Spreizung aller Finger, das „Victory“-Symbol und das Zeigen mit zwei Fingern) zum Einsatz kommen.









Dialogführung			Referenz	Kommandierung			
1	2	3	4	5	6	7	8
							

Tabelle 4.2: Verwendete statische Gesten

Bei der Untersuchung dynamischer Gesten ist das Verhalten eines Klassifikators bei unterschiedlichen Bewegungsbahnen interessant. Da Menschen einfache geometrische Muster selten präzise beschreiben, sind mehrere Merkmale zu betrachten. Interessante Merkmale sind Geradensegmente, Kreisbahnen und Kurven sowie unterschiedliche Bahnlängen. Die Erkennung sollte unabhängig von der Geschwindigkeit und Beschleunigungen während der Ausführung geschehen. Eine Auswahl dynamischer Gesten stellen die in Tabelle 4.3 gezeigten Beispiele dar. Die Reaktion einer Maschine auf solche Bewegungsmuster kann arbiträr belegt werden.

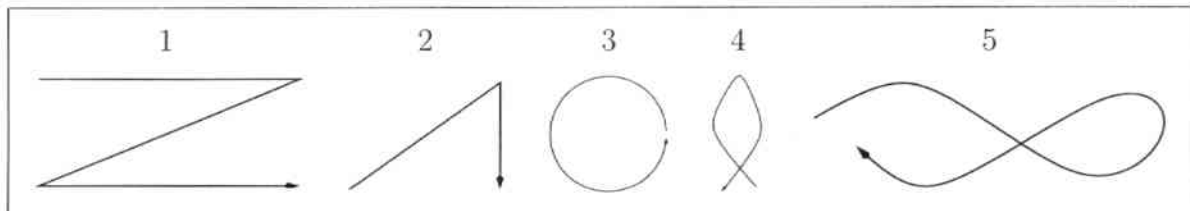


Tabelle 4.3: Verwendete dynamische Gesten

## 4.7 Zusammenfassung

Die Erfassung von Handlungsfolgen zur Programmierung und zur Interaktion stellt unterschiedliche Anforderungen an die Sensorik. Aufgrund dieser Beobachtung wurden in diesem Kapitel zwei separate Umgebungen eingeführt, die bei der Handlungsverfolgung entsprechende Merkmale und Sachverhalte fokussieren. Die Sensorik der Vorführungsumgebung besteht neben einem Datenhandschuh und einem magnetfeldbasierten Verfolgungssystem aus einem aktiven Kamerakopf. Bei der Ausführungsumgebung wird ausschließlich ein aktiver Kamerakopf verwendet, der variabel durch den Roboter positioniert werden kann. Der Unterschied im



Gebrauch der Sensorik schlägt sich auch in der vorgestellten Softwarearchitektur nieder.

Das Modellwissen bezüglich der betrachteten Benutzerhand, der Objekte und des gesamten Weltmodells ist in beiden Umgebungen gleich. Es wird zusammen mit den betrachteten Griffen und Gesten im folgenden Kapitel vorgestellt.

# Kapitel 5

## Elementare kognitive Operatoren

Im Folgenden werden kognitive Operatoren vorgestellt<sup>1</sup>, die aus beobachteten Sensordatenaufzeichnungen relevante Handlungsmerkmale detektieren, klassifizieren und im Ergebnis als parametrisierte Ereignisse im Weltmodell registrieren. Abbildung 5.1 a zeigt das Aussehen eines solchen Operators, der durch seinen Namen, die verarbeiteten Sensordatentypen, die registrierten Beobachtungsereignisse und ein ausführbares Programm gekennzeichnet ist. Das ausführbare Programm wird in Abhängigkeit des Handlungsmodells aktiviert und trägt bei Beobachtung eines Ereignisses dieses in das Weltmodell ein. Aus dem Weltmodell wiederum können für den Operator wichtige Parameter entnommen werden. Dieses Zusammenspiel ist in Abbildung 5.1 b wiedergegeben.

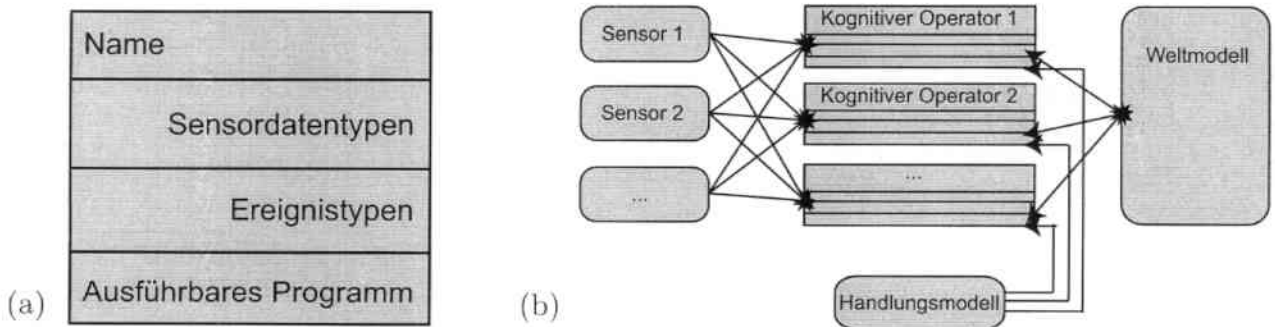


Abbildung 5.1: Kognitiver Operator (a) und Zusammenspiel der kognitiven Operatoren mit dem Beobachtungssystem (b)

In Abschnitt 4.6 wurden drei Handlungselemente festgelegt, zu deren Behandlung kognitive Operatoren im Rahmen einer Benutzervorführung dienen. Zu diesen kommt die Erfassung der aktuellen Szenensituation hinzu:

**Szenenanalyse:** Im Vorfeld der Beobachtung von Manipulationen müssen Orte und Typen der greifbaren Objekte bekannt sein. Auch nach dem Ablegen von Gegenständen müssen deren genaue Positionen bekannt sein. Der erste Operator wird daher zur Szenenanalyse eingesetzt.

<sup>1</sup>Eine formale Definition des Begriffs „kognitiver Operator“ ist im Anhang A gegeben

**Bewegungsverfolgung:** Die Verfolgung der Bewegung der menschlichen Hand dient der Gesten- und Grifferkennung. Die beobachteten Trajektorien sind im Kontext des Programmieren durch Vormachen nutzbar für die Generierung eines Roboterprogramms und müssen deshalb genau rekonstruiert werden<sup>2</sup>.

**Grifferkennung:** Die Basis für Manipulationen bilden das Greifen und das Bewegen von Objekten. Die Erkennung und die Klassifikation eines Griffes ist daher ebenfalls Aufgabe eines Operators.

**Gestenerkennung:** Während der Vorführung ausgeführte Gesten müssen auf Basis der Handkonfiguration oder der Handbewegung detektiert werden.

Die einzelnen Operatoren werden in dieser Reihenfolge behandelt.

## 5.1 Szenenanalyse

Aufgabe der Szenenanalyse ist die Objektdetektion, -klassifikation und Lokalisierung. Detektiert und klassifiziert werden im Kontext der Roboterinstruierung handlungsrelevante Objekte. In den typischen Anwendungsszenarien wie etwa Haushalten können dies beispielsweise Geschirr, Besteck, Stifte, Werkzeug oder Bücher sein. An die Objekterkennung werden die folgenden Anforderungen gestellt:

1. Die Objekterkennung muss robust und in Echtzeit ablaufen.
2. Die Objekterkennung muss gegenüber
  - gemustertem Hintergrund,
  - Objektrotation<sup>3</sup>,
  - Skalierungsänderungen der Objektansicht,
  - partiellen Objektverdeckungen

stabil sein.

3. Die Genauigkeit der Positions- und Lageschätzung muss den Anforderungen zur Roboterprogrammierung durch Vormachen bzw. zur Greifplanung genügen.

Zur Objekterkennung existiert eine sehr große Anzahl von Verfahren und Methoden, die in der Literatur [Kestler 99, Nagel 95, Lai 95, Klaus 87, Hesse 88] und in Bibliotheken [Matrox 98b, Intel 03] zu finden sind. Die Bibliographien [Gonzales 93, Jähne 97, Haberäcker 95, Zamperoni 89, Jain 95, McInerney 96] geben einen guten Überblick hierzu.

Im Folgenden werden vier der vielversprechendsten Ansätze zur Objekterkennung verglichen, die implementiert und auf ihren Einsatz in diesem Rahmen hin getestet. Eine Diskussion der

<sup>2</sup>engl.: Motion catch

<sup>3</sup>Dies wird zur Reduktion des Rechenaufwands als zweidimensionales Problem betrachtet.

Ergebnisse folgt in Abschnitt 5.1.4. Die dort festgestellten jeweiligen Stärken der Verfahren werden in einem integrierten Verfahren genutzt<sup>4</sup>. Da die Objekterkennung schnell sein muss, wurden vorrangig Methoden ausgewählt, die zweidimensionale Modelle verwenden:

1. Graphendetektion
2. Allgemeine Houghtransformation
3. Musteranpassung
4. Farbhistogrammbetrachtung

Als Modelle dienen für die Erkennung geeignete Ansichtsmuster (3), Konturmodelle (1, 2) oder Farbcharakteristika (4). Zur vollständigen Objekterkennung muss nach der Detektion auch die Lage des Objektes erfasst werden. Daher wird nach der Vorstellung der Detektionsverfahren ein Abschnitt der Lösung des Korrespondenzproblems und der Tiefenrekonstruktion gewidmet, bevor die Operatoren zur Szenenanalyse vorgestellt werden.

### 5.1.1 Konturbasierte Objektdetektion

Die im Folgenden erläuterten Methoden zur Objekterkennung stützen sich auf Bildkanten als zugrundeliegende Merkmale zur. Diese können stabil gegen Veränderungen der Beleuchtungsverhältnisse mit Kantendetektoren wie dem *Canny*-, *Sobel*-, *Laplace*-Filter oder einem Kantenmodell gefunden werden. Die damit erhaltenen Kantenbilder unterstützen in Verbindung mit Konturmodellen die Lagebestimmung von Objekten. Der Konturzug eines 3D-

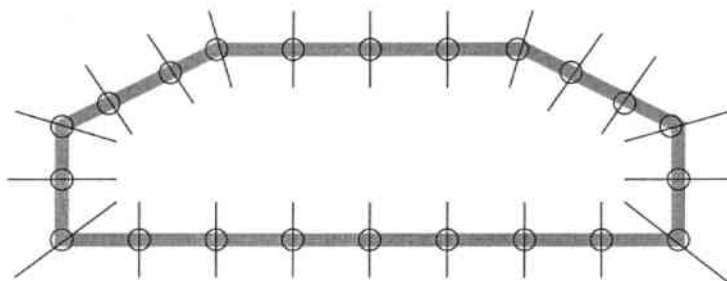


Abbildung 5.2: Konturmodell  $\vec{r}$  einer 2D-Objektansicht mit Stützpunkten und Orthogonalen  
Objektmodells lässt sich für die Objekterkennung mit Hilfe von B-Splines beschreiben:

$$\vec{r}(s) = (x(s), y(s)) \quad (5.1)$$

wobei  $x(s)$  und  $y(s)$  Splinefunktionen des Kurvenparameters  $s$  sind ( $0 \leq s \leq L$ , wobei  $L$  die Länge der Kurve bestimmt). Der Umriss  $\vec{r}$  lässt sich als gewichtete Summe von Kontrollpunkten  $\vec{q}_n = (q_n^x, q_n^y)^T$  über  $N_B$  Basisfunktionen  $B_n$  darstellen:

<sup>4</sup>Dieser Ansatz wurde erstmals in [Ehrenmann 00] vorgestellt.

$$\vec{r}(s) = \sum_{n=0}^{N_B-1} B_n(s) \vec{q}_n \quad (5.2)$$

Die Basisfunktionen sind derart konstruiert, dass gilt:

$$\sum_{n=0}^{N_B-1} B_n(s) = 1 \quad \text{für alle } s \quad (5.3)$$

und

$$\forall k \in \{0, 1, \dots, N_B - 1\} \quad \exists s \in [0, L] : \quad \vec{r}(s) = \vec{q}_k \quad (5.4)$$

Es ist einfacher, zur Zusammenfassung der Stützpunkte und Basisfunktionen eine Vektornotation zu verwenden. Mit

$$\vec{Q} := \begin{pmatrix} \vec{Q}^x \\ \vec{Q}^y \end{pmatrix} \quad \text{wobei} \quad \vec{Q}^x := \begin{pmatrix} q_0^x \\ \dots \\ \dots \\ q_{N_B-1}^x \end{pmatrix} \quad \text{und} \quad \vec{Q}^y := \begin{pmatrix} q_0^y \\ \dots \\ \dots \\ q_{N_B-1}^y \end{pmatrix} \quad (5.5)$$

und einem Vektor linearer Basisfunktionen  $\vec{B}(s)$  lässt sich Gleichung 5.2 schreiben als:

$$\vec{r}(s) = \left( \vec{B}(s)^T \vec{Q}^x, \vec{B}(s)^T \vec{Q}^y \right) \quad (5.6)$$

Auf Basis dieser Modellierung des Konturzuges eines Objektes erzeugen die folgenden beiden Algorithmen Kandidaten für dessen Lagepositionen in einem Bild.

## Graphendetektion

Die Methode der Graphendetektion folgt der Idee, entlang von Orthogonalen zum Konturmodell bestimmte Bildmerkmale zu überprüfen. Die Orthogonalen passieren als Abtastpunkte die Stützpunkte  $\vec{q}_n$  des Konturzuges  $\vec{r}$  analog Abbildung 5.2. Um Lageveränderungen der Kontur berücksichtigen zu können, ist es notwendig, den Stützpunktvektor  $\vec{Q}$ , der das Aussehen der Kontur beschreibt, durch eine Transformation  $\vec{X}$  anzupassen. Dies kann linear folgendermaßen ausgedrückt werden:

$$\vec{Q} = W \vec{X} + \vec{Q}_0 \quad (5.7)$$

Dabei ist  $W$  eine konstante  $N_Q \times N_X$ -Matrix und  $\vec{Q}_0$  das konstante Ausgangskonturmodell. Der Vektor  $\vec{X}$  beschreibt also die Transformation des Konturmodells  $\vec{Q}_0$  auf eine Position im Kamerabild. Die Konstruktion von  $W$  hängt von der Wahl der möglichen Transformationen in  $\vec{X}$  ab. Zur Objektdetektion sollen alle Rotationen um den Schwerpunkt sowie Translationen zugelassen werden. Allgemein lassen sich alle planaren affinen Transformationen in folgender Form darstellen:

$$\vec{r}(s) = \vec{u} + M \vec{r}_0(s) \quad (5.8)$$



mit einer beliebigen Translation  $\vec{u}$  und einer beliebigen  $2 \times 2$ -Matrix  $M$ . Entsprechend den sechs Dimensionen dieser Transformationen besitzt  $\vec{X}$  sechs Elemente und  $W$  hat die Gestalt

$$W = \begin{pmatrix} \vec{1} & \vec{0} & \vec{Q}_0^x & \vec{0} & \vec{0} & \vec{Q}_0^y \\ \vec{0} & \vec{1} & \vec{0} & \vec{Q}_0^y & \vec{Q}_0^x & \vec{0} \end{pmatrix} \quad (5.9)$$

wobei  $\vec{Q}_0^x$  und  $\vec{Q}_0^y$  wieder entsprechend Gleichung 5.5 die Modellkontur beschreiben, deren Ursprung im Schwerpunkt des Umrisses liegt. Die ersten beiden Spalten von  $W$  dienen dann offensichtlich der horizontalen und vertikalen Translation, während die anderen vier Linearkombinationen aus Gleichung 5.8 entsprechen:  $\vec{X} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})$ . Durch einen Vektor  $\vec{X} = (m, n, \cos(\theta - 1), \cos(\theta - 1), \sin \theta, \sin \theta)^T$  lassen sich somit  $(m, n)$ -Verschiebungen und Drehungen um  $\theta$  ausdrücken, aber auch Skalierungen. Beispielsweise repräsentiert  $\vec{X} = (0, 0, 1, 1, 0, 0)^T$  das um Faktor 2 skalierte Originalmuster  $\vec{Q}_0$ . Für die Objekterkennung im vorliegenden System sei  $\vec{X}$  jedoch auf Translationen und Rotationen limitiert.

Eine gegebene Kontur  $\vec{r}$  muss durch die Transformation  $\vec{X}$  so an das Kamerabild angepasst werden, dass das Modell auf korrespondierenden Bildcharakteristika zu liegen kommt. Dazu müssen zunächst entsprechende Merkmale modelliert werden. Da der Berechnungsaufwand gering gehalten werden soll, sollen die Bildverarbeitungsoperationen auf eine sehr kleine Region beschränkt bleiben.

Folgt man mit der Bildabtastung einer zum Modellumriss orthogonalen Strecke durch einen Stützpunkt  $\vec{n}(s_i)$  wie in Bild 5.3 b, muß man im Übereinstimmungsfall von Bild und Modell auf einen Grauwertsprung treffen. Dieser Sprung kann mit Hilfe eines Kantenoperators  $C$  lokalisiert werden, der Grauwertsprünge  $M$  modelliert (siehe Abbildung 5.3 c). Die Suche entspricht dann einem Extremwertproblem. Sei in  $N$  die Abtastung entlang  $\vec{n}(s_i)$  gegeben. Dann gilt:

$$M = \max_n \left[ \sum_{m=-N_C}^{N_C} C_m N_{n+m} \right] \quad (5.10)$$

Die Merkmalsdetektion entlang aller Orthogonalen durch die Modellstützpunkte  $\vec{n}(s_n)$  inkrementiert einen Zähler  $z$ , der die Detektion des Objektes anzeigt, sobald ein bestimmter Schwellwert überschritten wird:

$$z := \sum_{n=0}^{N_B-1} M(\vec{n}(s_n))$$

Dieser Zähler  $z$  wird an allen Bildpositionen und für alle Drehwinkel  $\theta := k \cdot \delta$  mit einer Schrittweite  $\delta$  bestimmt. Vorteil dieser Methode ist die Freiheit bei der Wahl geeigneter Bildmerkmale. Helle Objekte können beispielsweise durch die Transition von einem hellen zu einem dunklen Grauwert detektiert werden. Die Beschränkung auf die Merkmalsuche in einem stark eingeschränkten Suchraum beschleunigt die Detektion erheblich. Beim Suchen rotierter Modelle müssen außerdem immer nur wenige Abtastpunkte rotiert werden.

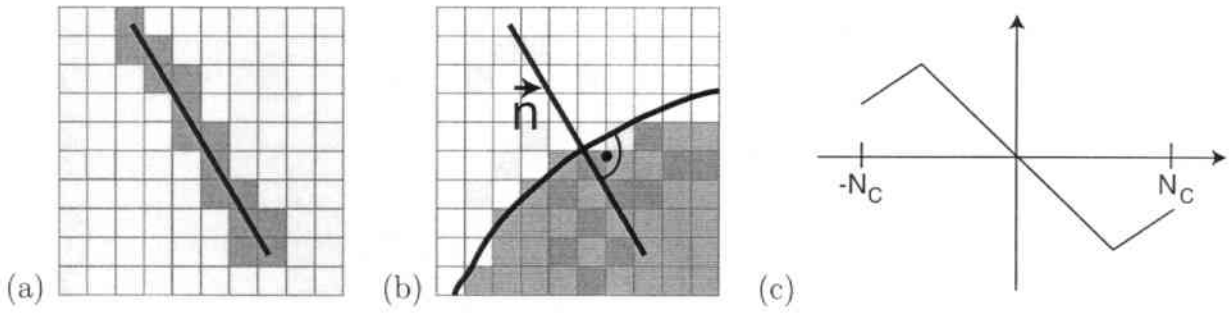


Abbildung 5.3: Bildabtastung entlang einer Linie (a), Merkmalsuche entlang der Konturorthogonalen (b) und Kantendetektor  $C$  (c). Der Kantendetektor  $C$  gibt die Modellierung eines zu findenden Grauwertverlaufs zwischen  $2N_C$  Bildpunkten an. Er dient zum Vergleich mit entlang der Konturorthogonalen  $\vec{n}$  abgetasteten Bildpunkten.

Kleine Winkelinkremente  $\delta$  geben oft gleiche Detektionsresultate  $z$  an derselben Objekt-position zurück. Zur genauen Lagebestimmung werden die resultierenden Winkel  $\theta$  daher gemittelt.

### Allgemeine Hough-Transformation

Methoden zur Muster- oder Graphanpassung benötigen erhebliche Rechenkapazität dafür, die Kongruenz zwischen Bild und Modell zu jedem Bildpunkt und für jede Rotation oder Skalierung zu bestimmen. Bei der allgemeinen Hough-Transformation<sup>5</sup> [Ballard 81] wird genau entgegengesetzt vorgegangen: im Bild vorkommende relevante Merkmale erhöhen hier Zähler für in Frage kommende Objektpositionen.

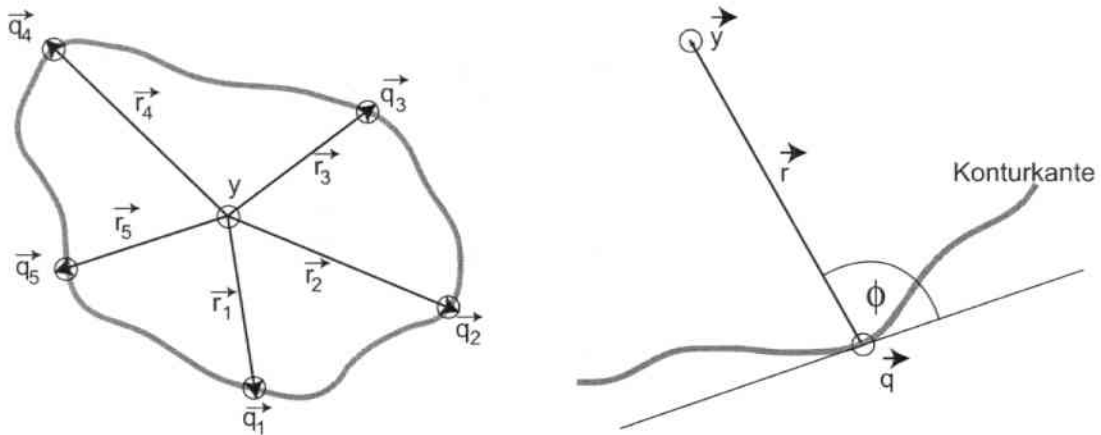


Abbildung 5.4: Konturmodell mit Parametern  $x$ ,  $y$  und  $r$

Das Konturmodell  $\vec{r}$  wird dabei durch die Menge der Stützpunkte  $\vec{q}_n$  approximiert. Jeder der Punkte in  $\vec{q}_n$  lässt sich mit Bezug auf einen Referenzpunkt  $\vec{y}$  innerhalb der Kontur

<sup>5</sup>Die allgemeine Hough-Transformation ist eine Erweiterung der Hough-Transformation, siehe [Hough 62, Duda 72]

durch einen Vektor  $\vec{r}_n = \vec{y} - \vec{q}_n$  beschreiben (siehe auch Abbildung 5.4). Für die Wahl von  $\vec{y}$  eignet sich beispielsweise der Schwerpunkt von  $\vec{r}$ . Alle Vektoren  $\vec{r}_n$  werden für ein Objekt in der sogenannten  $R$ -Tabelle gespeichert.

Für jedes Kantenbild  $I'$  wird nun eine Tabelle  $H$  mit der Dimension von  $I'$  angelegt. Jeder Kantenpunkt  $\vec{p}_E$  in  $I'$  wird auf eine Punktmenge  $P_H = \{\vec{p}_E + \vec{r}_i | \vec{r}_i \in R\text{-Tabelle}\}$  in  $H$  abgebildet. Es werden nun alle Elemente in  $H$ , dem sogenannten Hough-Puffer, um 1 erhöht, falls sie in  $P_H$  auftreten. Ein Objekt gilt als detektiert, sobald ein Element in  $H$  über einem Schwellwert liegt.

Diese Methode kann analog zur Graphendetektion zur Erfassung gedrehter und skaliertes Konturen erweitert werden. Für jeden Winkel  $\theta$  und jeden Skalierungsfaktor  $\sigma$  wird dazu ein neuer Hough-Akkumulator  $H_{\theta,\sigma}$  allokiert. Genauso wird auch die  $R$ -Tabelle um  $\theta$  und  $\sigma$  erweitert. Für die Objekterkennung von mit einer Schrittweite  $\delta$  gedrehten Modellen gilt bei entsprechender Modelltransformation  $T_\delta$  und  $R(\Theta)$  als erweiterter  $R$ -Tabelle:

$$T_\delta[R(\Theta)] = \text{Rot}\{[(\Theta - \delta) \bmod 2\pi], \delta\} \quad (5.11)$$

Das heißt, dass alle Indizes  $n$  um  $-\delta$  in der Tabelle erhöht werden. Die Skalierungstransformation erfolgt analog: mit  $\sigma$  als Skalierungsfaktor,  $R(\Sigma)$  als erweiterter  $R$ -Tabelle und  $T_\sigma$  als Transformation kann man schreiben:

$$T_\sigma[R(\Sigma)] = s \cdot R(\Sigma) \quad (5.12)$$

Jeder Vektor  $\vec{r}_i$  der  $R$ -Tabelle wird somit um den Faktor  $\sigma$  skaliert.

Den Ungenauigkeiten des Modells oder Lichtreflexionen kann folgendermaßen begegnet werden: statt der Erhöhung der Zähler für die Punkte  $\vec{p}_E + \vec{r}_i$  in  $H$  um 1 wird ein Inkrement in Form einer Gausskurve um die nächsten Nachbarn addiert. In Abbildung 5.5 sind ein Kamerabild und zwei typische Hough-Puffer  $H$  für die Drehwinkel  $\delta = 0^\circ$  und  $\delta = 120^\circ$  wiedergegeben, bei denen diese Art der Inkrementierung gewählt wurde. Hier ist deutlich zu sehen, dass der zentrale Punkt im rechten oberen Bild heller als seine Nachbarn ist und damit die Objektlage markiert. Das rechte untere Teilbild zeigt das Kantenbild mit überlagertem Konturmodell  $\vec{r}$ .

### 5.1.2 Musteranpassung

Zur Ausnutzung der Objekttextur bietet es sich an, Teilbilder oder Ansichtsbilder der fraglichen Objekte für die Objekterkennung zu nutzen. Neben der Verwendung von Eigenwertverfahren werden hierzu hauptsächlich Verfahren zur Schablonen- bzw. Musteranpassung eingesetzt.

Die Ansätze zur Muster- oder Schablonenanpassung sind wegen ihrer Einfachheit die weitverbreitet. Hier werden normalerweise die quadrierten Differenzen zwischen Kamerabild  $I$  und Bildmuster  $M$  punktweise aufsummiert, um ein Maß für die Ähnlichkeit zwischen dem Bild der Schablone und dem Kamerabild zu erhalten:

$$d(x, y) = \sum_{i \in M} \sum_{j \in M} (I(x + i, y + j) - M(i, j))^2 \quad (5.13)$$

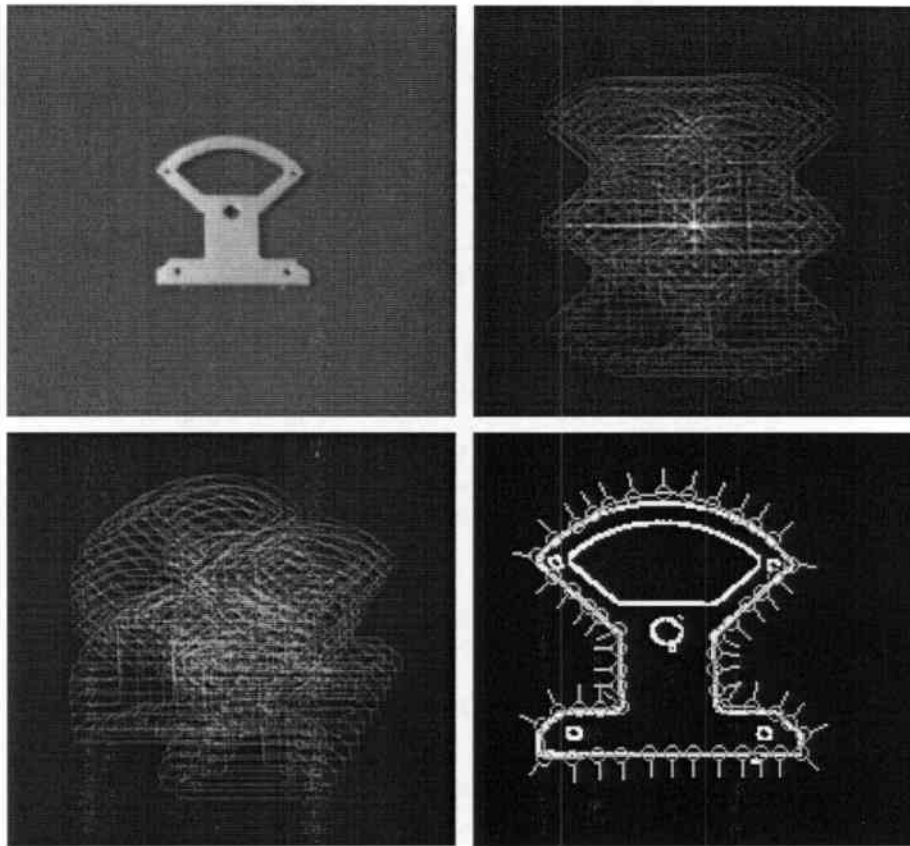


Abbildung 5.5: Kamerabild  $I$ , zwei Hough-Puffer  $H$  bei verschiedenen Drehwinkeln sowie das Resultat der Objekterkennung.

Ein Objekt gilt als gefunden, wenn das Ergebnis  $d(x, y)$  an einem Bildpunkt einen vorgegebenen Schwellwert unterschreitet. Eine Variante zu Formel 5.13 ist die folgende Formel, die zur Stabilisierung gegenüber Lichtschwankungen vorgeschlagen wurde:

$$d = \frac{\sum IM - \sum I \cdot \sum M}{\sqrt{(n \sum I^2 - (\sum I)^2)(n \sum M^2 - (\sum M)^2)}} \quad (5.14)$$

wobei  $n$  die Anzahl der Bildpunkte des Musterbildes ist. Zur Beschleunigung der Suche können hier alle Terme außer  $\sum I$ ,  $\sum I^2$  und  $\sum IM$  vorberechnet werden.

Für die Erfassung gedrehter Objekte im Kamerabild muss bei diesem Verfahren das Vergleichsmuster  $M$  um einen Winkel  $\delta$  gedreht und neu gesucht werden. Erfahrungsgemäß sind hier Schrittweiten von  $\delta \approx 5^\circ$  ausreichend.

Ohne Einschränkung der Stabilität des Verfahrens kann die Mustersuche durch einen hierarchischen Ansatz drastisch beschleunigt werden: eine vorberechnete Anzahl kleinerer und geringer aufgelöster Versionen des Kamerabildes sowie der Musterbilder dient dazu, die Suche auf einer wesentlich kleineren Skala zu beginnen und die interessanten Regionen in Bildern höherer Auflösung weiterzuverfolgen. Einen Eindruck von dieser sogenannten Pyramiden- oder Gausssuche gibt Abbildung 5.6 wieder.

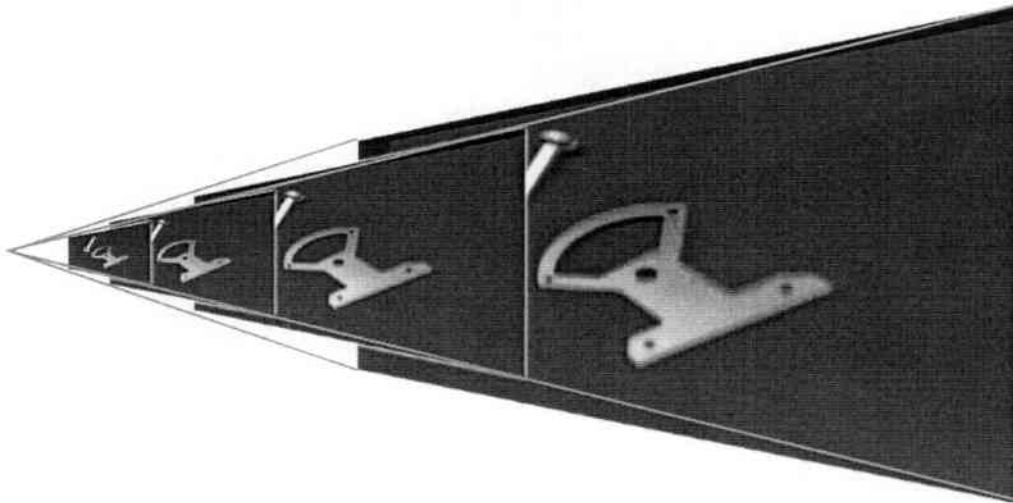


Abbildung 5.6: Hierarchische Verkleinerung von Kamera- und Musterbild

### 5.1.3 Farbbasierte Objektdetektion

Sobald sich die beobachteten Objekte nicht mehr partiell verdecken, also vereinzelt präsentiert vorliegen, wird die Unterscheidung von Bildpunkten, die zu einem Objekt oder zum Bildhintergrund gehören, wesentlich erleichtert. Diese Segmentierung kann ebenso wie die Klassifikation bei unstrukturiertem Hintergrund durch Farbbetrachtungen erfolgen. Damit können zwei der eingangs in Abschnitt 5.1 erwähnten Forderungen entfallen. Die dadurch erreichbaren Arbeitsgeschwindigkeiten erlauben Echtzeitbildverarbeitung, was für die interaktive Systemanwendung erforderlich ist. Für die Ausführumgebung ist daher diese Einschränkung ausreichend, da dort der Mensch mit dem Robotersystem direkt kooperiert.

Zur Segmentierung von Objekten lässt sich das Farbreduktionsverfahren nach Comaniciu [Comaniciu 97] einsetzen. Hiermit werden die signifikanten Farben im Kamerabild ermittelt und die häufigste Farbqualität ausgeblendet. Das Verfahren läuft in mehreren Schritten über dem *RGB*-Raum als Merkmalsraum ab:

1. Signifikanten Farben entsprechen Dichteregionen im Merkmalsraum. Die Dichte in diesem Raum wird in der Umgebung von  $m$  zufällig gewählten Startpositionen im Kamerabild untersucht. Dazu wird der Farbmittelwert der Nachbarpixel bestimmt und in den Merkmalsraum abgebildet. Derjenige Vektor mit der höchsten Dichte wird ausgewählt und weiterverarbeitet.
2. Eine Näherung an das Dichtemaximum im Merkmalsraum wird durch den Mittleren Verschiebungsvektor<sup>6</sup>  $\vec{x}$  erreicht. Die Suchkugel wird um diesen Vektor solange verschoben, bis  $||\vec{x}'|| < \epsilon$  gilt (siehe Abbildung 5.7). Der Verschiebungsvektor  $\vec{x}$  berechnet sich proportional zum Dichtegradienten im Merkmalsraum und realisiert damit ein Aufstiegsverfahren.

<sup>6</sup>engl.: Mean Shift



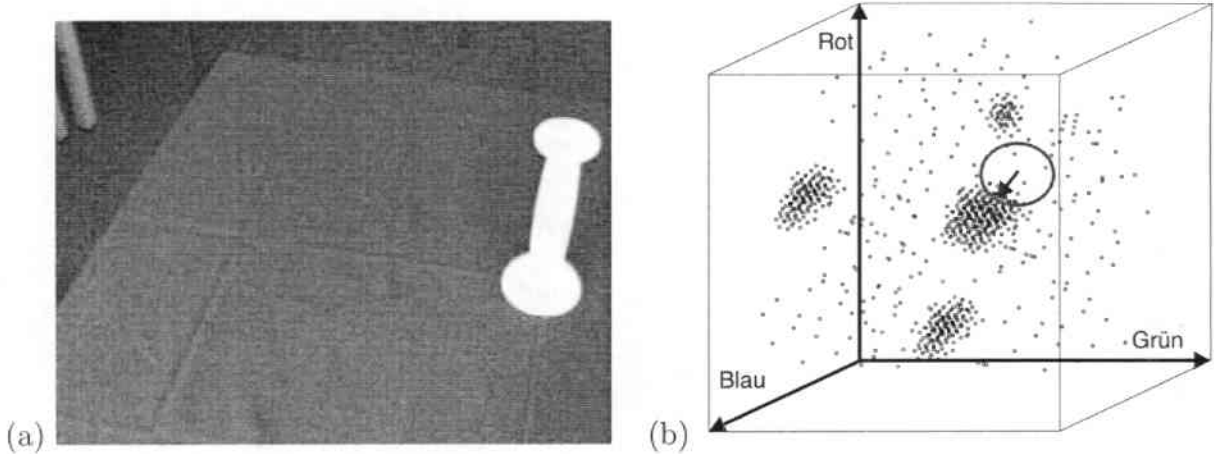


Abbildung 5.7: Kamerabild (a) und Mittlerer Verschiebungsvektor im Merkmalsraum (b).

3. Aus dem Bildraum und aus dem Merkmalsraum werden daraufhin alle Pixel, die Merkmalsvektoren im Suchfenster erzeugt haben, mit ihren 8-Nachbarn und entsprechenden Merkmalsvektoren gelöscht.
4. Diese Schritte werden solange iterativ abgearbeitet, bis kein Suchfenster mehr über mehr als  $n_{\min}$  Merkmalsvektoren verfügt. Dies ist die Mindestanzahl von Bildpunkten für signifikante Farben.

Mit Hilfe der signifikanten Farben wird eine neue Farbpalette für das Kamerabild bestimmt. Dabei werden Farbverschiebungen und Rauschen beseitigt. Betrachtet man das Histogramm eines solcherart gefilterten Bildes wie in Abbildung 5.8 wiedergegeben, lässt sich unmittelbar die Hintergrundfarbe ermitteln. Als Objekte lassen sich anschließend diejenigen Regionen bestimmen, die eine Umgebung in Hintergrundfarbe besitzen. Mit Hilfe einer Maske können dann der Hintergrund und uninteressante Objekte ausgeblendet werden.

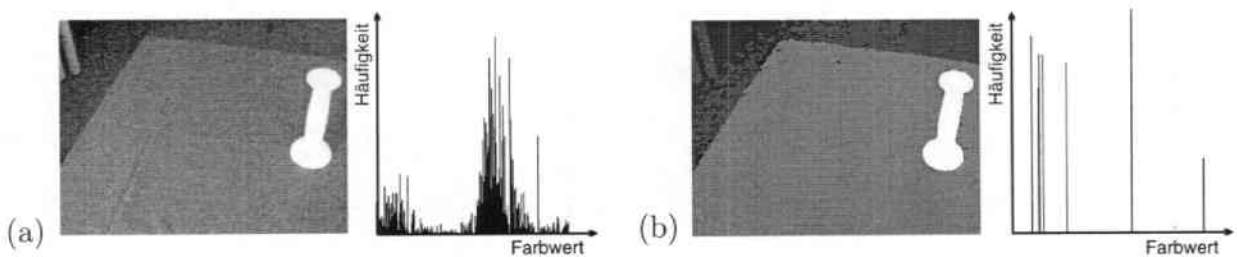


Abbildung 5.8: Kamerabild (a) und gefiltertes Bild mit Histogramm (b).

Sobald ein Objekt vom Hintergrund segmentierbar und beobachtbar ist, können dessen charakteristische Eigenschaften  $c$  gespeichert werden. Im folgenden sind dies die Farb-, Intensitäts- und Sättigungshistogramme  $\vec{h}^H$ ,  $\vec{h}^L$  und  $\vec{h}^S$  sowie einfache geometrische Eigenschaften wie das Höhe/Breite-Verhältnis. Zur Erhöhung der Robustheit gegenüber Rauschen und Farbverschiebungen (siehe Anhang D) werden die Histogramme  $\vec{h}$  durch Mittelung von je-

weils 10 Farbwerten in  $\vec{h}'$  komprimiert, d.h. die  $\vec{h}'$  berechnen sich anhand der Formel:

$$\forall i \in \{0, \dots, 24\} : h'_i := \frac{\sum_{k=\lfloor 256i/24 \rfloor}^{\lfloor 256(i+1)/24 \rfloor} h_k}{\lfloor 256(i+1)/24 \rfloor - \lfloor 256i/24 \rfloor} \quad (5.15)$$

Zwischen gespeicherten Merkmalen für ein Objekt  $o$  und einem im Kamerabild segmentierten Objekt  $o'$  können nun Ähnlichkeitsmessungen anhand dieser Merkmale  $c$  bestimmt werden, wobei jedes Merkmal ein spezifisches Gewicht  $w$  erhält:

$$\|o - o'\| = \sum w_i \|c^o - c^{o'}\| \quad (5.16)$$

Dabei wird der Gewichtungsfaktor  $w$  für den Farbanteil auf 10, für die anderen beiden Histogramme auf 1 gesetzt, um gegenüber Helligkeitsschwankungen robust zu sein. Als für Abstandsmessungen über den komprimierten Histogrammen geeignete Metrik hat sich die folgende Formel bewährt:

$$\|h^o - h^{o'}\| = \sqrt{\sum_{i=0}^{24} \|h_i^o - h_i^{o'}\|} \quad \text{mit} \quad (5.17)$$

$$\|h_i^o - h_i^{o'}\| = \min_{j=i-\Delta}^{i+\Delta} |i - j| \cdot |h_i^o - h_j^{o'}| \quad (5.18)$$

Die Differenz der beiden Histogramme wird nach Gleichung 5.18 immer in einer Umgebung  $\Delta$  „wohlwollend“ bestimmt: der am besten übereinstimmende Wert wird mit dem Ausgangsdatum verglichen. Mit der Festsetzung der Umgebungsgröße  $\Delta$  lassen sich lokale Farbverschiebungen robust behandeln. In den Versuchen wurde  $\Delta = 5$  gesetzt. Sobald  $\|o - o'\|$  unter eine Schwelle  $\theta$  fällt, gilt das segmentierte Objekt als klassifiziert. Der Vorteil dieser Methode liegt darin, dass das Lernen neuer Objekte im Abspeichern der Charakteristika  $c$  besteht. Es ist keine Parametereinstellung und keine manuelle Modellierung von Objekten notwendig.

### 5.1.4 Diskussion

Jede der vorgestellten Methoden hat spezifische Vor- und Nachteile. Im Folgenden werden die Resultate der Untersuchung im Hinblick auf die vorliegenden Einsatzbedingungen diskutiert. Zeitaufwandsbetrachtungen beziehen sich dabei immer auf die Suche nach einem Objekt im gesamten Bild.

**Graphanpassung:** In komplexeren Bildern, die mehrere Objekte enthalten oder einen stark gemusterten Hintergrund enthalten, zeigt die Graphanpassung keine zufriedenstellenden Resultate. Der Aufwand des Verfahrens liegt in der Klasse  $O(n^{kl})$  bei  $n$  Stützpunkten,  $k$  Winkelstellungen und  $l$  abzusuchenden Bildpositionen. Die Suche liegt damit trotz Verwendung der Gaußpyramiden auf derzeitigen Rechensystemen<sup>7</sup> bei etwa 2 Minuten. Dabei wurden Kamerabilder mit  $640 \times 480$  Bildpunkten mit Graphen, die durch 50 Stützpunkte definiert worden waren, untersucht. Die gefundenen Positionen und Orientierungen zeigten sich jedoch sehr genau.

<sup>7</sup>Verwendet für die Experimente wurde ein bei 300MHz getakteter PC mit zwei Pentium2-Prozessoren

Ein großer Nachteil der Methode ist die Parametervielfalt. Die Länge der Orthogonalen, ihre Anzahl und die Definition der relevanten Merkmale, nach denen gesucht werden soll, haben wesentlichen Einfluß auf das Suchergebnis und müssen bei Änderungen der Beleuchtungsbedingungen sowie einzeln für jedes Objekt neu gewählt werden.

Kleine Objekte sind aufgrund ähnlicher Konturen nicht mehr gut zu klassifizieren. Außerdem kann die Orientierung für symmetrische Objekte nicht exakt bestimmt werden. Dies trifft jedoch auf alle konturbasierten Methoden zu.

**Allgemeine Houghtransformation** Methoden auf Basis der Houghtransformation haben sehr kurze Laufzeiten. Der Aufwand liegt hier zwar in derselben Klasse  $O(n^{kl})$ ,  $l$  ist aber wegen der ausschließlichen Betrachtung von Kantenpunkten wesentlich kleiner als im Fall der Graphanpassung. Bei einem Winkelinkrement von  $\delta = 3$  ohne Gausspyramiden waren 10 Sekunden zur Modellsuche notwendig<sup>8</sup>. Der Geschwindigkeitsvorteil wird hier jedoch in Speicherverbrauch aufgelöst. Im Falle von  $\delta = 3$  werden ohne Skalierungsbetrachtung über 35MB Speicher für die Houghpuffer verwendet. Die Positions- und Orientierungsergebnisse sind jedoch sehr genau.

Verdeckungen und Rauschen im Bild können hier sehr gut behandelt werden. Wie die Graphanpassung zeigt sich die allgemeine Houghtransformation jedoch als ungeeignet für die Detektion kleiner Objekte, die kaum aufgrund ihres Umrisses unterschieden werden können.

Muster im Bildhintergrund inkrementieren Elemente in  $H$  und machen eine gute Objektdetektion schwieriger. Die Schwellwerte müssen daran jeweils angepasst werden.

**Musteranpassung** Der theoretische Aufwand der Musteranpassung liegt in derselben Klasse wie die obigen Algorithmen, wobei  $n$  hier nicht die Stützpunkte, sondern die Bildpunktanzahl bestimmt. Die Detektionszeit mit einem Winkelinkrement von  $\delta \leq 5$  liegt hier zwischen 2–4 Sekunden bei stabilen und genauen Ergebnissen. Der Geschwindigkeitsvorteil ist hierbei jedoch hauptsächlich der Verwendung der Gausspyramiden und der Verwendung der Matrox-Bibliothek geschuldet, die die Unterstützung von digitalen Signalprozessoren ausnützt.

Der Ansatz erlaubt jedoch nicht die Detektion verdeckter Objekte. Außerdem können leichte Drehungen um eine Achse, die parallel zum Bildsensor liegt, aufgrund der Texturveränderungen nicht behandelt werden. Flexible Objekte oder Objekte mit reflektierenden Oberflächen ändern die Erkennungsleistung ebenfalls inakzeptabel stark.

**Farbhistogrammvergleich:** Bedingung für den Einsatz des Farbhistogrammvergleichs ist die Fähigkeit zur Segmentierung. Diese ist nur bei Hintergründen homogener Farben zu schaffen. Beispiele hierfür sind im Anwendungsszenario einfarbige Tischdecken. Der Ablauf erfolgt dann mit einem Aufwand aus  $O(nk)$ . Die Anzahl der Objekte sei hierbei mit  $n$  bezeichnet, während  $k$  die Größe der Histogramme beschreibt. In der Praxis zeigt sich dieser Ansatz mit Laufzeiten  $\approx 1s$  als sehr schnell und robust. Der Vorteil dieses Verfahrens liegt in der Parameterarmut, die das automatische Lernen neuer Objekttypen gestattet.

Methode	GA	AHT	MA	FH
Zeitaufwand	-	+	+	++
Speicheraufwand	++	--	++	++
Stabilität bei Verdeckungen	+	++	-	--
Muster im Bildhintergrund	o	-	++	--
Orientierung	+	++	+	o
Skalierung	+	++	--	+
Parametrierung	--	-	o	++

Tabelle 5.1: Vergleich der betrachteten Methoden zur Objekterkennung (GA=Graphanpassung, AHT=Allgemeine Hough Transformation, MA=Musteranpassung und FH=Farbhistogrammvergleich)

Die Ergebnisse dieser Diskussion sind in Tabelle 5.1 zusammengefasst. Da im Rahmen von Interaktionen sehr schnell Erkennungsleistungen erfolgen müssen, wird für die Ausführungsumgebung die farbbasierte Histogrammbetrachtung als Methode zur Objektdetektion ausgewählt. Hier ist außerdem die Möglichkeit, neue Objekte durch einfache Benennung zu lernen, von Vorteil.

In der Vorführungsumgebung bieten der Mustervergleich durch Schablonenanpassung und die allgemeine Houghtransformation gute Lösungen für die Objektdetektion mit unterschiedlichen Schwächen in Abhängigkeit von der Objektgröße, -struktur und -textur. Bei der Anwendung im Rahmen des Programmierens durch Vormachen wird daher jedes Objekt mit einem Parameter zur Bestimmung einer geeigneten Detektionsmethode ausgestattet. Dies erhöht die Anzahl korrekter Identifikationen. Beispiele für die Auswahl sind nach der obigen Diskussion:

- Metallische Objekte werden generell aufgrund störender Reflexionen besser durch konturgestützte Methoden erkannt.
- Bei der Klassifikation kleiner Objekte bietet die Textur die Grundlage für die Klassifikation. Deshalb sollten hier ansichtsbasierte Methoden gewählt werden.
- Objekte, die häufig partiell verdeckt auftreten, können mit Hilfe der allgemeinen Houghtransformation besser als mit den diskutierten anderen Methoden erkannt werden.

Für jedes Objekt aus der Datenbank wird dann mit der günstigsten Methode im Bild gesucht. Die Ergebnisse werden am Ende der Detektion fusioniert: Treffer mit hoher Detektionswahrscheinlichkeit werden bei ähnlichen Positionen anderen vorgezogen und letztere nicht betrachtet<sup>9</sup>.

<sup>8</sup>Diese Zeitmessung ergab sich auf demselben oben erwähnten Rechner

<sup>9</sup>Der genaue Ablauf wird in Algorithmus 5.1 vorgestellt.

### 5.1.5 Positionsbestimmung

Sobald ein Objekt im Kamerabild detektiert und klassifiziert ist, kann dessen Raumlage bestimmt werden. Hierzu sind drei Schritte sinnvoll: zunächst muss das Korrespondenzproblem gelöst werden, d.h. korrespondierende Punkte in den entsprechenden Stereokamerabilddern müssen identifiziert werden. Sind für jede Kamera die korrespondierenden Bildpunkte aus der betrachteten Szene bekannt, kann die Rekonstruktion dieser Szenepunkte berechnet werden. Um diese Abbildung zu bestimmen, müssen die Kameras zuvor kalibriert werden. Da die Methoden hierzu in der Literatur gut belegt sind [Faugeras 93, Tsai 87, Wilson 94], werden die drei Verarbeitungsprozesse Kamerakalibrierung, Korrespondenzfindung und Rekonstruktion im Folgenden nur kurz erläutert.

#### Kamerakalibrierung

Eine Bildaufnahme einer bekannten Szenensituation mit genau vermessenen Punkten kann dazu dienen, die Abbildungsvorschrift von Szenen- zu Bildpunkten zu bestimmen. Dazu wird im Allgemeinen ein Lochkameramodell mit linearer Transformation vorausgesetzt<sup>10</sup>. Es müssen also die direkten linearen Transformationen (*DLT*-Matrizen)  $M = (m_{ij})$  ausgerechnet werden, die Szenepunkte  $(x, y, z)^T$  auf Bildpunkte  $(u, v)$  abbilden. Nach der Erweiterung um homogene Koordinaten zur Erfassung von Translationen lässt sich dieses Problem wie folgt notieren ([Weckesser 97]):

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (5.19)$$

Diese Gleichung lässt sich nach  $u$  und  $v$  auflösen:

$$u = \frac{m_{1,1}x + m_{1,2}y + m_{1,3}z + m_{1,4}}{m_{3,1}x + m_{3,2}y + m_{3,3}z + m_{3,4}} \quad (5.20)$$

$$v = \frac{m_{2,1}x + m_{2,2}y + m_{2,3}z + m_{2,4}}{m_{3,1}x + m_{3,2}y + m_{3,3}z + m_{3,4}} \quad (5.21)$$

Durch die Verwendung homogener Koordinaten stellt jede Matrix  $k \cdot M$  ( $k \neq 0$ ) die gleiche Abbildung dar. Um bei der Berechnung nicht die triviale Lösung zu erhalten, muss dieser Freiheitsgrad eingeschränkt werden. Dies wird erreicht, indem man einen der Koeffizienten  $m_{i,j}$  festlegt. Üblicherweise wird der Koeffizient  $m_{3,4}$  auf den Wert eins gesetzt.

Seien  $(u_i, v_i)$  die Bildkoordinaten und  $(x_i, y_i, z_i, 1)$  die Raumkoordinaten des bekannten Referenzpunktes  $\vec{p}_i$ , so folgt aus den Gleichungen 5.20 und 5.21:

$$u_i x_i m_{3,1} + u_i y_i m_{3,2} + u_i z_i m_{3,3} - x_i m_{1,1} - y_i m_{1,2} - z_i m_{1,3} - m_{1,4} = -u_i \quad (5.22)$$

$$v_i x_i m_{3,1} + v_i y_i m_{3,2} + v_i z_i m_{3,3} - x_i m_{2,1} - y_i m_{2,2} - z_i m_{2,3} - m_{2,4} = -v_i \quad (5.23)$$

<sup>10</sup>Der Einfluss radialer Verzerrungen war bei der verwendeten Brennweite von 6 – 8mm vernachlässigbar.



Für die Koeffizienten  $m_{i,j}$  der Matrix  $M$  ergibt sich daraus:

$$\begin{pmatrix} x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i z_i \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i x_i & -v_i y_i & -v_i z_i \end{pmatrix} \cdot \begin{pmatrix} m_{1,1} \\ m_{1,2} \\ m_{1,3} \\ m_{1,4} \\ m_{2,1} \\ m_{2,2} \\ m_{2,3} \\ m_{2,4} \\ m_{3,1} \\ m_{3,2} \\ m_{3,3} \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \end{pmatrix} \quad (5.24)$$

Eine Raum-Bildpunkt-Kombination ergibt zwei Gleichungen mit 11 Unbekannten. Um die Matrix  $M$  zu bestimmen, braucht man daher mindestens sechs Referenzpunkte. Mit diesen sechs Referenzpunkten ergibt sich folgendes Gleichungssystem, aus dem sich die Matrix  $M$  berechnen lässt:

$$\underbrace{\begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 x_1 & -u_1 y_1 & -u_1 z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1 x_1 & -v_1 y_1 & -v_1 z_1 \\ \vdots & & & & & & & & & & \vdots \\ x_6 & y_6 & z_6 & 1 & 0 & 0 & 0 & 0 & -u_6 x_6 & -u_6 y_6 & -u_6 z_6 \\ 0 & 0 & 0 & 0 & x_6 & y_6 & z_6 & 1 & -v_6 x_6 & -v_6 y_6 & -v_6 z_6 \end{pmatrix}}_{:=A} \cdot \underbrace{\begin{pmatrix} m_{1,1} \\ m_{1,2} \\ \vdots \\ m_{3,3} \end{pmatrix}}_{:=\vec{m}} = \underbrace{\begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ u_6 \\ v_6 \end{pmatrix}}_{:=\vec{p}'} \quad (5.25)$$

Benutzt man zur Berechnung mehr als 6 Referenzpunkte, erhält man ein überbestimmtes Gleichungssystem, das sich aufgrund von Messfehlern in den Szenen- und Bildkoordinaten im Allgemeinen nicht exakt lösen lässt. Man verlangt stattdessen, dass die in den Gleichungen auftretenden Abweichungen minimal werden. Deshalb wird an dieser Stelle das Verfahren der kleinsten Fehlerquadrate verwendet, um Gleichung 5.25 zu lösen:

$$A^T \cdot A \cdot \vec{m} = A^T \cdot \vec{p}' \quad (5.26)$$

Dadurch wird eine Abschätzung der Koeffizienten der Transformationsmatrix geliefert. Im vorliegenden System wurde ein Kalibrierobjekt mit 200 Messpunkten verwendet. Jeder Messpunkt liegt zur hochgenauen Bildpunktbestimmung im Zentrum eines schwarzen Kreises mit

einem Zentimeter Radius und kann durch Schwerpunktbestimmung aus dem Kamerabild gewonnen werden (Abbildung 5.9). Die Kalibrierobjekte werden zur Kalibrierung so positioniert, dass ihre Entfernung zu den Kameras der erwarteten Distanz üblicher Vorführungen entspricht. Im Falle der Ausführungsumgebung ist dies ca. 130cm vom Kamerakopf entfernt, im Falle der Vorführungsumgebung ca. 150cm (siehe Abbildung 4.2).

### Korrespondenzproblem

Vor der Rekonstruktion auf der Basis von bi- oder trinokularen Stereokameras sind die einem Objektmerkmal in der Szene entsprechenden Bildpunkte in allen Kamerabildern zu identifizieren. Da interessante Objekte mit Hilfe der oben genannten Methoden detektiert werden können, kann davon ausgegangen werden, dass solche Merkmale in einem Kamerabild bereits vorliegen. Dies wird genutzt, um einen Mustervergleich mit einem Bildausschnitt um den Schwerpunkt des fraglichen Objekts in den anderen Kamerabildern durchzuführen. Die dabei in den anderen Kamerabildern auftretenden Korrespondenzkandidaten werden z.B. mit Hilfe der epipolaren Geometrie auf ihre Plausibilität hin untersucht. Weicht ihre Position zu sehr von der Epipolarlinie ab, werden sie verworfen. Abbildung 5.10 zeigt zwei Kamerabilder, von denen das linke zur Objektdetektion und Klassifikation genutzt wurde. Die Ergebnisse sind dabei direkt in das Bild eingetragen. Im rechten Kamerabild sind aufgrund des Mustervergleichs von Bildausschnitten fünf Kandidaten für die drei Objekte gefunden worden. Der Abstand zu den drei eingezeichneten Epipolarlinien zeigt jedoch, dass die beiden linken Kandidaten wieder verworfen werden konnten.

### Rekonstruktion

Die bei der Kalibrierung für jede Kamera  $k$  berechneten *DLT*-Matrizen  $M^k$  bilden die Szenekoordinaten auf die Bildkoordinaten  $(u, v)^k$  ab. Für die Rekonstruktion ist nun ein Szenepunkt  $\vec{p} = (x, y, z)^T$  zu bestimmen, für den gilt:

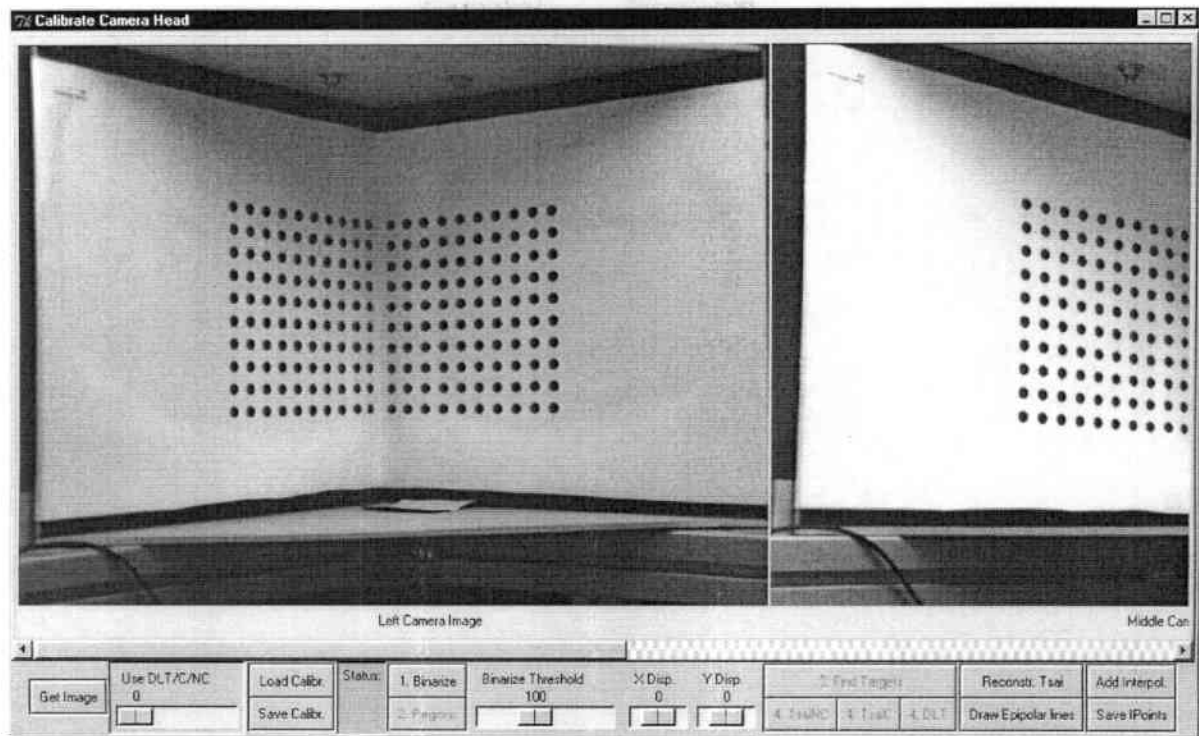
$$\forall k : (u, v)^k = M^k \cdot \vec{p} \quad (5.27)$$

Da ein einzelnes Gleichungssystem 5.27 eine Gerade als Lösung hat, wird  $\vec{p}$  durch die Kombination mindestens zweier dieser Gleichungen gewonnen:

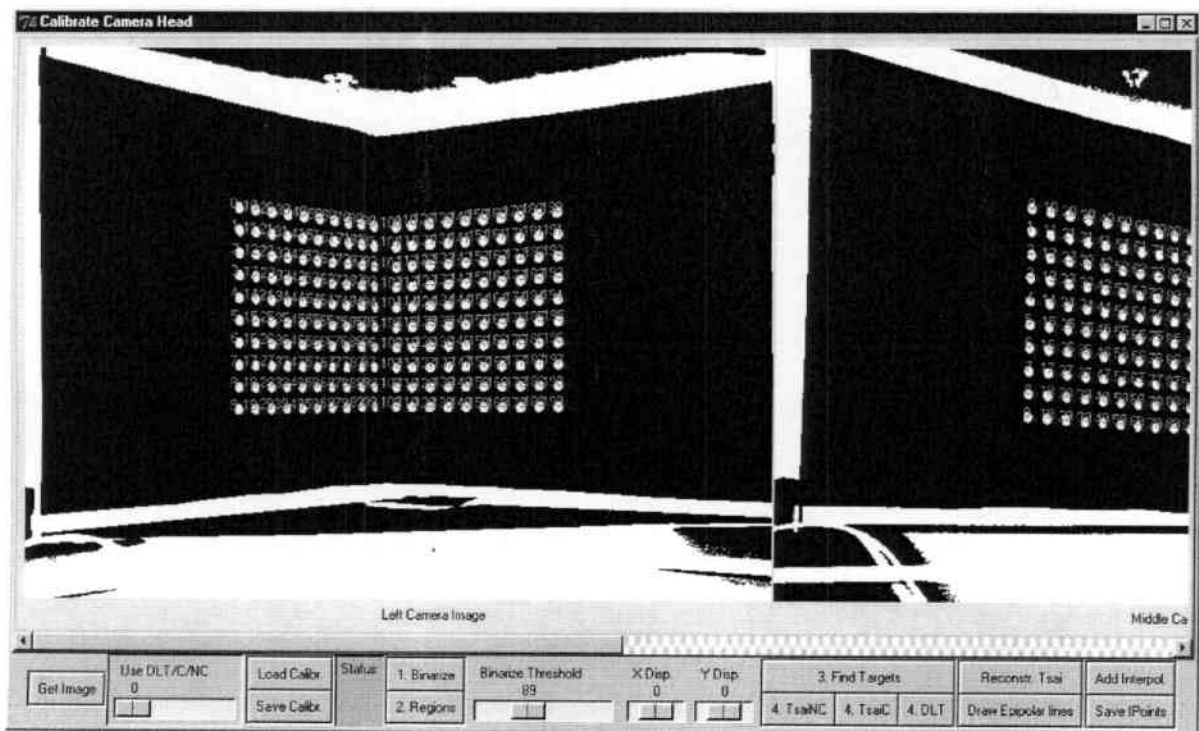
$$\begin{pmatrix} m_{31}^0 u^0 - m_{11}^0 & m_{32}^0 u^0 - m_{12}^0 & m_{33}^0 u^0 - m_{13}^0 \\ m_{31}^0 v^0 - m_{21}^0 & m_{32}^0 v^0 - m_{22}^0 & m_{33}^0 v^0 - m_{23}^0 \\ m_{31}^1 u^1 - m_{11}^1 & m_{32}^1 u^1 - m_{12}^1 & m_{33}^1 u^1 - m_{13}^1 \\ m_{31}^1 v^1 - m_{21}^1 & m_{32}^1 v^1 - m_{22}^1 & m_{33}^1 v^1 - m_{23}^1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} m_{14}^0 - m_{34}^0 u^0 \\ m_{24}^0 - m_{34}^0 v^0 \\ m_{14}^1 - m_{34}^1 u^1 \\ m_{24}^1 - m_{34}^1 v^1 \end{pmatrix} \quad (5.28)$$

Der Szenepunkt  $\vec{p}$  ist damit im Kalibrierkoordinatensystem bestimmt. Diese Koordinaten sind noch in das Weltkoordinatensystem zu transformieren. Da die genutzte Stereokamera über mehrere Drehgelenke rotierbar ist<sup>11</sup>, kommen dabei sowohl Translationen  $T$  als auch Rotationen  $R$  zum Einsatz. Beide lassen sich bei Verwendung homogener Koordinaten

<sup>11</sup>Das System lässt sich schwenken und neigen



(a)



(b)

Abbildung 5.9: Aufnahme des Kalibrierobjekts (a) und identifizierte Markerzentren (b)

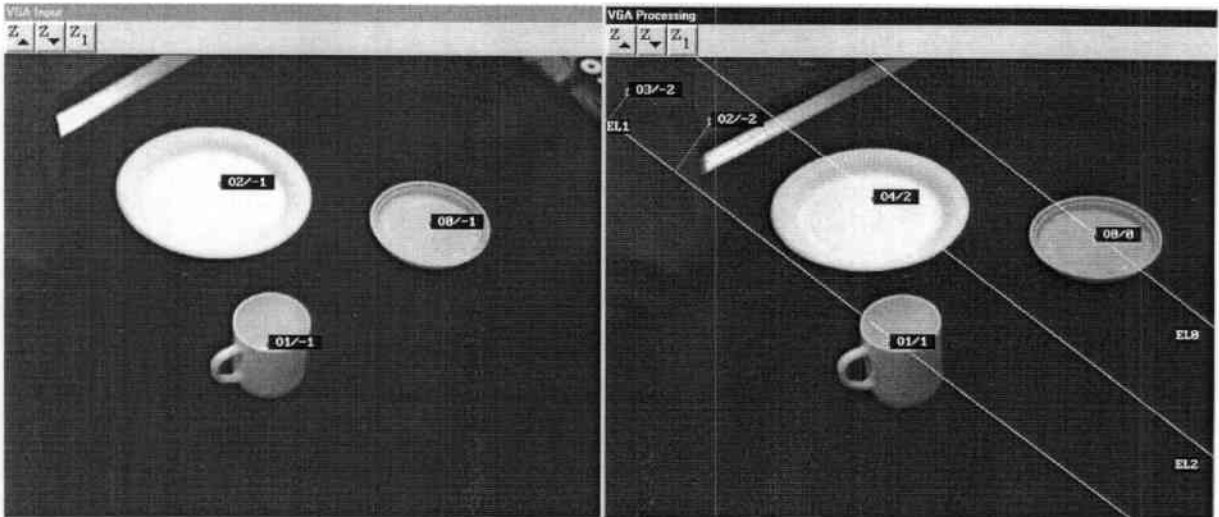


Abbildung 5.10: Ergebnis der Objekterkennung (links) und der Lösung des Korrespondenzproblems (rechts).

als lineare Operationen berechnen. Die einzelnen Koordinatensysteme sind hier sukzessiv das Kalibrierobjektkoordinatensystem, das Kamerakoordinatensystem, das erste und zweite Drehzentrum und schließlich das Weltkoordinatensystem. Die gesamte Transformation von  $P$  erweitert also Gleichung 4.1 auf die Gestalt:

$$\begin{aligned}
 P_W &= T'_{\text{Welt, TCP}} \cdot P \\
 &= T_{\text{Welt, Kamerakopf}} \cdot TR_{\text{Kamerakopf, Kameras}} \cdot \\
 &\quad T_{\text{Kameras, Kalibrierobjekt}} \cdot P
 \end{aligned} \tag{5.29}$$

wobei

$$\begin{aligned}
 TR_{\text{Kamerakopf, Kameras}} &= T_{\text{Kamerakopf, Drehgelenk2}} \cdot R_\beta \cdot \\
 &\quad T_{\text{Drehgelenk2, Drehgelenk1}} \cdot R_\alpha \cdot \\
 &\quad T_{\text{Drehgelenk1, Kameras}}
 \end{aligned} \tag{5.30}$$

mit  $\alpha$  Neige- und  $\beta$  Drehwinkel der jeweiligen Drehmodule.

### 5.1.6 Operator zur Objektdetektion

Die einzelnen Schritte der Objektdetektion mit der jeweiligen Positionsschätzung fasst Algorithmus 5.1 zusammen. Hier wird zunächst der Kamerakopf auf den interessierenden Szenenausschnitt bewegt (Zeilen 1 bis 3). Anschließend wird nach allen in der Objektdatenbank vorhandenen Modellen in einem der Kamerabilder gesucht. Dabei werden die assoziierten Detektionsmethoden verwendet (Zeilen 4 bis 7, die einzelnen Detektionsmethoden aus den Abschnitten 5.1.1, 5.1.2 und 5.1.3 seien über eine Funktion „Detektiere“ abrufbar). Die detektierten Merkmale können sich nun überschneiden. Aus der Liste der gefundenen Objekte müssen daher überzählige Typen entfernt werden. Das ist Aufgabe der Zeilen 8 bis 18: für

jedes gefundene Objekt wird überprüft, ob derselbe Typ an derselben Stelle mit einer ähnlichen Lage gefunden wurde. Danach werden für übriggebliebene Objekte korrespondierende Bildmerkmale in den anderen Kamerabildern gesucht und die Rekonstruktion angestoßen (Zeilen 19 bis 23). Die Registrierung erfolgt im Anschluß daran mit den ermittelten Positionen und Drehlagen im Weltmodell.

Die Operatoren zur Szenenanalyse sind für beide Umgebungen identisch. Da in der Ausführungsumgebung jedoch nur eine Methode zur Objektdetektion ausgewählt wurde, ist hier die Fusionierung bedeutungslos und die Zeilen 8 bis 18 entfallen.

## 5.2 Handverfolgung

Da die Anforderungen an die Bewegungsverfolgung in Abhängigkeit der geplanten Anwendung sehr unterschiedlich sind, werden verschiedene Ansätze vorgeschlagen und geprüft. In der Vorführungsumgebung steht die präzise Rekonstruktion der Handbewegung im Vordergrund, während in der Ausführungsumgebung die Hand ohne zusätzliche Marker oder Hilfsmittel verfolgt werden können soll.

Die Handverfolgung soll zur Nachführung des aktiven Stereokamerakopfs auch schnellen Benutzerbewegungen folgen können. Es ist nicht zu erwarten, dass mit Modifikationen der Algorithmen zur Objektdetektion signifikante Beschleunigungen erreicht werden können, die dafür ausreichend sind. Vielmehr sind hier direkt auf die Anwendung zugeschnittene Ansätze sinnvoll.

### 5.2.1 Handverfolgung in der Vorführungsumgebung

Zur Handverfolgung in der Vorführungsumgebung wurde im Rahmen dieser Arbeit zunächst der Einsatz eines magnetfeldbasierten Verfolgungssystems<sup>12</sup> untersucht und danach eine kontur- sowie eine markerbasierte Methode präsentiert, die auf dem Einsatz eines aktiven Stereokamerasystems beruhen. Beide visuelle Verfahren lassen sich mit dem Magnetfeldsystem kombinieren, um eine höhere Robustheit und Genauigkeit zu erzielen.

#### Magnetfeldbasierte Handverfolgung

Für die Handverfolgung können die von einem magnetfeldbasierten Trackingsystem gelieferten Messwerte benutzt werden. Die Genauigkeit dieses Systems bei der Positionsbestimmung wurde in Testreihen ermittelt [Stasch 97]. Die Ergebnisse sind in Abbildung 5.11 graphisch dargestellt.

Es zeigt sich, daß innerhalb eines Bereiches von 60 cm zum Sender die translatorischen Abweichungen zwischen gemessener und tatsächlicher Handposition bis zu ca. 5 cm betragen können. Außerhalb dieses Bereiches steigen die Meßfehler nichtlinear an. Die rotatorischen Abweichungen verhalten sich proportional zu den translatorischen. Im Gegensatz zu dieser großen globalen Ungenauigkeit des Trackingsystems ist lokal betrachtet nur ein geringes Rauschen vorhanden.

<sup>12</sup>engl.: Tracking system, siehe Abschnitt 4.3.1



```

Eingabe: Raumkoordinaten  $P$  einer Region der Szene, Objektdatenbank  $D$ 
mit Objektbeschreibungen und assoziierten Detektionsmethoden,
Schwellwerte  $\theta_d$  und  $\theta_\alpha$  zur Bestimmung der Ähnlichkeit von
Objektpositionen und Objektlagen.
Ausgabe: Gefundene Szenenobjekte mit Raumposition.

/*Kopfpositionierung*/
1:  $(\alpha, \beta) \leftarrow \text{Drehwinkel}(P)$ 
2:  $\text{BewegeKopf}((\alpha, \beta))$ 
3:  $(I_1, I_2, I_3) \leftarrow \text{DigitalisiereKamerabilder}(Kamera_1, Kamera_2, Kamera_3)$ 
/*Objektdetektion*/
4:  $\text{Szenenobjekte} \leftarrow \emptyset$ 
5: for all  $d \in D$  do
6:    $\text{Szenenobjekte} \leftarrow \text{Szenenobjekte} \cup \text{Detektiere}(I_1, d.\text{Modell}, d.\text{Methode}, d.\text{Parameter})$ 
7: end for
/*Fusion*/
8:  $\text{Auswahl} \leftarrow \emptyset$ 
9: for all  $o \in D$  do
10:  for all  $p \in D$  do
11:    if  $(o \neq p) \wedge (o.\text{Typ} = p.\text{Typ}) \wedge (|o.\text{Pos} - p.\text{Pos}| < \theta_d) \wedge (|o.\text{Winkel} - p.\text{Winkel}| < \theta_\alpha)$ 
    then
12:       $\text{Auswahl} \leftarrow \text{Auswahl} \cup \{o, p\}$ 
13:    end if
14:  end for
15:   $\text{SortiereNachErkennungswahrscheinlichkeit}(\text{Auswahl})$ 
16:   $\text{Auswahl} \leftarrow \text{Auswahl} \setminus \text{ErstesElement}(\text{Auswahl})$ 
17:   $\text{Szenenobjekte} \leftarrow \text{Szenenobjekte} \setminus \text{Auswahl}$ 
18: end for
/*Positionsschätzung*/
19: for all  $o \in \text{Szenenobjekte}$  do
20:   $\text{FindeKorrespondenz}(o, I_1, I_2)$ 
21:   $\text{FindeKorrespondenz}(o, I_1, I_3)$ 
22:   $\text{Rekonstruiere3D}(o)$ 
23: end for
/*Registrierung im Weltmodell*/
24:  $\text{Registriere}(\text{Szenenobjekte})$ 

```

**Algorithmus 5.1:** Elementarer kognitiver Operator zur Szenenanalyse

Die Ungenauigkeiten des Trackingsystems sind vor allem auf Störeinflüsse auf das elektromagnetische Feld des Senders zurückzuführen. Ursachen für diese Ablenkungen und Verstärkungen sind

- Hindernisse zwischen Sender und Empfänger,
- fremde elektromagnetische Felder, verursacht durch Monitore, Netzgeräte oder stromführende Leitungen und

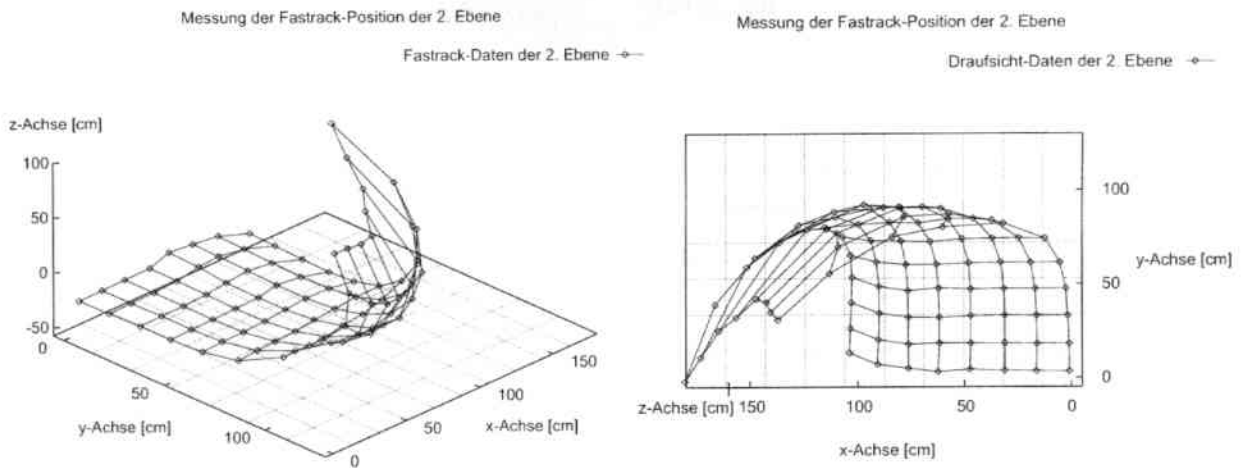


Abbildung 5.11: Perspektivische Ansicht und Draufsicht einer mit dem Trackingsystem vermessenen Ebene

- ferromagnetische Gegenstände im Raum, die Störungen in den Magnetfeldern des Sensorsystems verursachen.

Die geschilderte Empfindlichkeit des Trackingsystems gegenüber elektromagnetischen Störungen verlangt nach Entstörungsmaßnahmen sowie Signalfilterungsprozessen, um die beschränkte Genauigkeit zu erhöhen. Im Experiment zeigte sich, daß Geräte wie Bildschirme in der Nähe des Trackingsystems oder Rechner den Verlauf des Magnetfeldes signifikant stören können. Der vorgeschlagene Ansatz, das Referenzmagnetfeld zu vermessen und durch inverse Filter die Messwerte zu verbessern, stellte sich als nicht praktikabel heraus. In [Friedrich 98] findet sich der Vorschlag, bildgebende Sensoren während der Ausführung anzuwenden. Da der Magnetfeldsensor in einer eng umgrenzten Kugel um den Emitter sehr genau ist (ca. 1mm Abweichung), bietet sich der kombinierte Einsatz beider Sensoren auf der Basis einer geeigneten Sensordatenfusion an. Damit kann gegenüber Objektverdeckungen eine erhöhte Robustheit erzielt werden.

Im Folgenden werden zwei Verfahren zur bildgestützten Bewegungsverfolgung vorgestellt, die für eine Fusion mit einem Magnetfeldtracker geeignet sind. Das Verfahren zur Kombination der Messungen wird zusammen mit dem entsprechenden kognitiven Operator diskutiert.

### Bildbasierte Handverfolgung auf Basis von Konturmerkmalen

Bewegte Objekte bildbasiert zu verfolgen erfordert aufgrund der hohen Dimensionalität von Kamerabildern und Bildraten einen hohen Aufwand an Rechenleistung. Es ist deshalb naheliegend, zur Senkung des Rechenaufwands nicht auf den gesamten Basisbilddaten, sondern auf lokalen Fenstern zu operieren und dort z.B. konturbasierte Verfahren einzusetzen (siehe Abschnitt 2.3.2). Dabei bietet es sich an, die Objektlagen aus dem vorherigen Kamerabild  $I_{t-1}$  zu nutzen und lediglich in deren Umgebung nach geeigneten Merkmalen suchen. Modellwissen der Hand oder der Szenenobjekte ist hierzu hilfreich. Das Modellwissen sei entsprechend Abschnitt 5.1.1 in einem Stützpunktvektor  $\vec{Q}$  gegeben. Für die folgenden Betrachtungen seien zusätzlich in der Merkmalskurve  $\vec{Q}_f$  die im Bild gefundenen Merkmalskanten des aktuellen Objekts gespeichert. Bei einem gegebenen Konturmodell der Hand oder eines

anderen Objekts lässt sich dann die Anpassung an das Kamerabild als Minimierungsproblem beschreiben. Ziel ist es, denjenigen Vektor  $\vec{X}$  zu finden, der die entsprechende Transformation bei minimalem Abstand zu seinem Vorgänger  $\vec{X} - \vec{X}_{t-1}$  beschreibt (siehe Gleichung 5.7). Er lautet:

$$\vec{X}_{\min} = \min_{\vec{X}} \left[ \underbrace{(\vec{X} - \vec{X}_{t-1})^T S_{t-1} (\vec{X} - \vec{X}_{t-1})}_{\substack{\text{kontrollierte Abweichung} \\ \text{der neuen Kurve } \vec{X} \\ \text{von der} \\ \text{gegebenen Kurve } \vec{X}_{t-1}}} + \underbrace{\|\vec{Q} - \vec{Q}_f\|_{\vec{n}}^2}_{\substack{\text{Abweichung der} \\ \text{neuen Kurve } \vec{Q} \\ \text{von der Merk-} \\ \text{malskurve } \vec{Q}_f}} \right] \quad (5.31)$$

Die Regularisierungsmatrix  $S$  im ersten Summanden dient der Gewichtung von Transformationsparametern in  $\vec{X}$ . Hier kann beispielsweise Rotationen eine geringere Gewichtung zugeordnet werden, wenn sie als unwahrscheinlich gelten. Der Vektor  $\vec{X}$  sei wieder definiert als  $\vec{X} = (x_1, \dots, x_6)$  zur Erfassung aller affinen Transformationen (siehe Gleichung 5.7 ff). Die Abweichung der neuen Kurve  $\vec{Q}$  im zweiten Term ergibt sich nach Gleichung 5.7 zu  $W\vec{X} + \vec{Q}_0 - \vec{Q}_f$ . Sei  $\vec{X}_f$  derjenige Vektor, der die Kontur optimal an die Bildinformation anpasst. Es gilt dann

$$\vec{X}_f = W^+(\vec{Q}_f - \vec{Q}_0) \quad (5.32)$$

mit

$$W^+ := (W^T A W)^{-1} W^T A \quad \text{und} \quad (5.33)$$

$$A := \frac{1}{L} \int_0^L \begin{pmatrix} B^T(s)B(s) & \mathbf{0} \\ \mathbf{0} & B^T(s)B(s) \end{pmatrix} ds \quad (5.34)$$

Damit hat man

$$\vec{X}_{\min} = (S_{t-1} + W^T A W)^{-1} (S_{t-1} \vec{X}_{t-1} + W^T A W \vec{X}_f) \quad (5.35)$$

Die Messung der Abweichung der Merkmalskurve von den Modelldaten kann durch die Abweichung in den Orthogonalen  $\vec{n}(s_i)$  angenähert werden. Bei  $N_{B-1}$  Stützpunkten  $s_i$  wird dies beschrieben durch

$$\|\vec{r} - \vec{r}_f\|_{\vec{n}}^2 \approx \frac{0}{N_B - 1} \sum_{i=1}^N [(\vec{r}_f(s_i) - \vec{r}(s_i))^T \vec{n}_{t-1}(s_i)]^2 \quad (5.36)$$

Benutzt man entsprechend Gleichung 5.6 die Notation

$$\vec{r}(s_i) = \begin{pmatrix} B^T(s_i) & \mathbf{0} \\ \mathbf{0} & B^T(s_i) \end{pmatrix} (W\vec{X} + \vec{Q}_0) \quad (5.37)$$

erhält man mit

$$\nu_i := [\vec{r}_f(s_i) - \vec{r}_{t-1}(s_i)]^T \vec{n}_{t-1}(s_i) \quad \text{und} \quad (5.38)$$

$$\vec{h}(s_i) := W^T \begin{pmatrix} B^T(s_i) & \mathbf{0} \\ \mathbf{0} & B^T(s_i) \end{pmatrix} \vec{n}_{t-1}(s_i) \quad (5.39)$$

die Abschätzung:

$$\|\vec{r} - \vec{r}_f\|_{\vec{n}}^2 \approx \frac{1}{N_B} \sum_{i=0}^{N_B-1} (\nu_i - \vec{h}^T(s_i) [\vec{X} - \vec{X}_{t-1}])^2 \quad (5.40)$$

Anstelle der gleichförmigen Gewichte  $1/N$  lassen sich auch unterschiedliche Faktoren benutzen. Gleichung 5.40 hat dann die Form:

$$\|\vec{r} - \vec{r}_f\|_{\vec{n}}^2 \approx \frac{\sum_{i=0}^{N_B-1} w_i (\nu_i - \vec{h}^T(s_i) [\vec{X} - \vec{X}_{t-1}])^2}{\sum_{i=0}^{N_B-1} w_i} \quad (5.41)$$

Dementsprechend lässt sich Gleichung 5.31 mit Gewichten  $1/\sigma_i^2 := w_i$  auch schreiben als:

$$\vec{X}_{\min} = \min_{\vec{X}} \left[ (\vec{X} - \vec{X}_{t-1})^T S_{t-1} (\vec{X} - \vec{X}_{t-1}) + \sum_{i=0}^{N_B-1} \frac{1}{\sigma_i^2} (\nu_i - \vec{h}^T(s_i) [\vec{X} - \vec{X}_{t-1}])^2 \right] \quad (5.42)$$

Die Lösung dieses Problems lässt sich mit Hilfe des von Blake [Blake 98b] vorgeschlagenen Vorgehens iterativ berechnen. Der gesamte Vorgang zur Konturverfolgung ist in Algorithmus 5.2 wiedergegeben, die Zeilen 11 bis 19 betreffen hierbei die Bestimmung des optimalen Transformationsvektors  $\vec{X}$ . Dazu ist eine  $2 \times 2N_B$  Matrix  $U(s_i)$  notwendig, die jeweils einen Stützpunkt  $s_i$  zur Berechnung der Lageabweichung auswählt. Wenn  $N_B$  Stützpunkte in einem Konturmodell  $\vec{r}$  vorhanden sind, sei  $U(s_i)$  folgendermaßen definiert:

$$U(s_i) := \begin{pmatrix} 0 & \dots & 0 & \overbrace{1}^i & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \underbrace{1}_{i+N_B} & 0 & \dots & 0 \end{pmatrix} \quad (5.43)$$

In Zeile 24 kann aus dem bestimmten Vektor  $\vec{X}'$  die Lageinformation extrahiert werden. Die ersten beiden Komponenten enthalten direkt die Bildpunktkoordinaten, die zur Rekonstruktion genutzt werden können.

### Bildbasierte Handverfolgung auf Segmentierungsbasis

Wie sich in den experimentellen Untersuchungen der konturorientierten Verfolgung in Abschnitt 6.1.2 gezeigt hat, sind diese gegenüber schnellen Bewegungen nicht robust genug, um eine sichere Trajektorienaufzeichnung garantieren zu können. Es wird deshalb ein weiteres Verfahren entwickelt, das bei hoher Verarbeitungsgeschwindigkeit auch einen großen Bildausschnitt zur Verfolgung betrachten kann. Grundlage für dessen Verwendung ist ein Marker, der mit einem  $3 \times 5$ cm großen Muster beklebt ist (siehe Abschnitt 4.6.1). Das auf Kreisstrukturen basierende Muster hat den Vorteil, auch bei Verkippungen gut erkennbar zu bleiben (Abbildung 5.12 links). Es zeigt sich auch bei strukturierten Hintergründen als signifikante Region. In Abbildung 5.12 ist rechts ein Kamerabild zusammen mit dem Binarisierungsergebnis bei einer Schwelle von  $\theta = 160$  angegeben. Der Marker lässt sich hier zweifelsfrei identifizieren. Gegenüber der Erkennung aktiver Marker wie Leuchtdioden mit dem Einsatz von Infrarotfiltern zeigte sich in Versuchen mit diesem Ansatz kein Nachteil.

**Eingabe:** Polygonzug  $\vec{r}$  mit  $N_B - 1$  Stützpunkten  $s_i$ , Kamerabilder  $I_1^t, I_2^t$ , Polygonzug  $\vec{r}_{t-1}$  im Kamerabild  $I_1^{t-1}$ , Transformationsvektor  $\vec{X}_{t-1}$  und Regularisierungsmatrix  $S_{t-1}$ , Merkmalsdetektor  $C$ , Minimale Merkmalsanzahl  $\theta$

**Ausgabe:** Raumpunkt  $P_t$  des verfolgten Objekts.

```

/*Finde Stützpunkte im Kamerabild anhand der Modelllage in  $I_1^{t-1}$ */
1:  $z \leftarrow 0$ 
2: for all  $i \in \{0, \dots, N_B - 1\}$  do
3:    $\vec{n}_{t-1}(s_i) \leftarrow \text{BestimmeOrthogonaleIn}(\vec{r}_{t-1}(s_i))$ 
4:   if  $\text{FindeMerkmalEntlang}(\vec{n}_{t-1}(s_i), I_1^t, C)$  then
5:      $z \leftarrow z + 1$ 
6:      $\vec{r}_f(s_i) \leftarrow \text{Position}_C$ 
7:   else
8:      $\vec{r}_f(s_i) \leftarrow \text{nil}$ 
9:   end if
10: end for
/*Bestimme Beobachtungsvektor  $\vec{Z}$  mit statistischer Information  $S$ */
11:  $\vec{Z}_0 \leftarrow \vec{0}, \quad S_0 \leftarrow \mathbf{0}$ 
12: for all  $i \in \{0, \dots, N_B - 1\}$  do
13:    $\nu_i \leftarrow (\vec{r}_f(s_i) - \vec{r}_{t-1}(s_i)) \cdot \vec{n}_{t-1}(s_i)$ 
14:    $\vec{h}(s_i)^T \leftarrow \vec{n}_{t-1}(s_i)^T U(s_i) W$ 
15:    $S_i \leftarrow S_{i-1} + \frac{1}{\sigma_i^2} \vec{h}(s_i) \vec{h}(s_i)^T$ 
16:    $\vec{Z}_i \leftarrow \vec{Z}_{i-1} + \frac{1}{\sigma_i^2} \vec{h}(s_i) \nu_i$ 
17: end for
18:  $\vec{Z} \leftarrow \vec{Z}_N, \quad S \leftarrow S_N$ 
/*Ermittlung der am besten passenden Transformation  $\vec{X}'$ */
19:  $\vec{X}' \leftarrow \vec{X}_{t-1} + (S_{t-1} + S)^{-1} \vec{Z}$ 
/*Genügend Merkmale für sichere Verfolgung vorhanden?*/
20: if  $z < \theta$  then
21:    $\text{Verloren} \leftarrow \text{true}$ 
22: else
23:    $\text{Verloren} \leftarrow \text{false}$ 
24:    $(u_1, v_1) \leftarrow (\vec{X}'_1, \vec{X}'_2)$ 
25:    $(u_2, v_2) \leftarrow \text{FindeKorrespondenz}((u_1, v_1), I_1, I_2)$ 
26:    $P_t \leftarrow \text{Rekonstruiere3D}((u_1, v_1), (u_2, v_2))$ 
27: end if

```

**Algorithmus 5.2:** Konturverfolgungsalgorithmus nach Blake

Zur Geschwindigkeitssteigerung erfolgt die Verarbeitung hier mit einfachen Operationen. Die Kamerabilder werden zunächst mit einem Schwellwert  $\theta$  in Binärbilder konvertiert. In der resultierenden Bildmatrix werden zusammenhängende Bereiche gefunden und markiert. Von diesen Regionen werden die folgenden charakteristischen Eigenschaften bestimmt:

- Umfang  $U$ ,



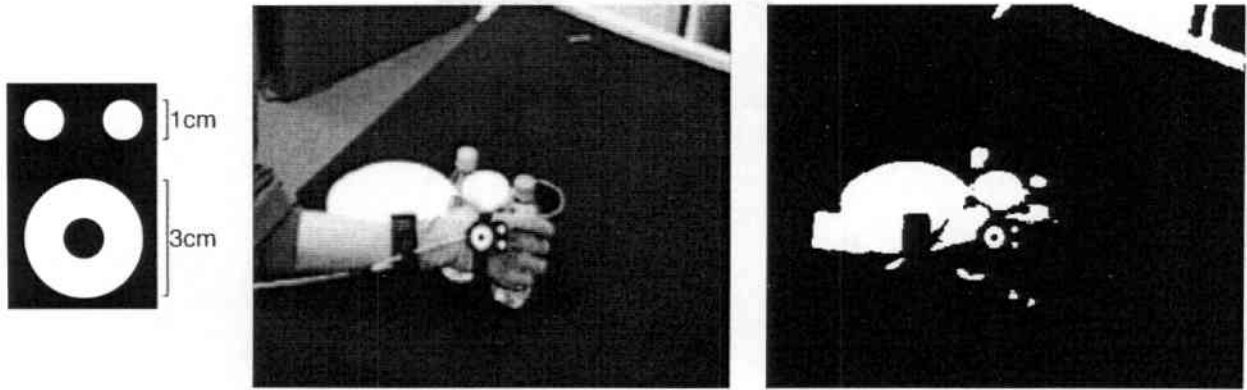


Abbildung 5.12: Markerpattern, Kamerabild und Binärbild

- Flächeninhalt  $A$ ,
- Löcheranzahl  $L$  und der
- Schwerpunkt  $(m_x, m_y)$ .

Umfang  $U$  und Flächeninhalt  $A$  dienen zur Berechnung der Kompaktheit  $K$ :

$$K := \frac{U^2}{4\pi A} \quad (5.44)$$

Sie liefert bei kreisförmigen Strukturen einen Wert nahe bei 1. Das Prädikat *Marker* wird über den Bildregionen  $R$  mit Hilfe dieser Eigenschaften als boolesches Produkt definiert durch:

$$\text{Marker}(R) := \begin{cases} \text{true} & \text{wenn } (K(R) < \theta_K) \wedge \\ & (A(R) < \theta_A) \wedge \\ & (L(R) == 1) \wedge \\ & (\exists_2 R' : (|R - R'| < \theta_d) \wedge A(R') > \theta_{A'}) \\ \text{false} & \text{sonst} \end{cases} \quad (5.45)$$

$R$  soll eine gewisse Kompaktheit und Fläche besitzen sowie ein Loch aufweisen und genau zwei Regionen innerhalb eines definierten Abstands aufweisen. Das Zentrum des Großen der drei Kreise im Markerpattern gilt als Zentrum bzw. Koordinatenbasis des Markers, dessen Raumposition rekonstruiert werden muss. In Algorithmus 5.3 sind die einzelnen Schritte aufgeführt. Ist der Marker lokalisiert, kann die Suche in den nächsten Kamerabildern auf einen Bereich in der Nähe beschränkt werden (Zeilen 8 und 9). Bei Verlust des Markers (Zeile 21) z.B. durch Verdeckung wird zunächst wieder im Gesamtbild gesucht und dazu die Variable „Verloren“ gesetzt. Diese Variable dient dazu, den Verlust durch Nutzung der Werte des magnetfeldbasierten Verfolgungssystem auszugleichen. Bei Richtungswechseln oder sehr schnellen Bewegungen kann der Marker an den Rand des lokalen Fensters gelangen. Dies wird durch Betrachtung des ganzen Kamerabildes ausgeglichen.

```

Eingabe: Kamerabilder  $I_{1,2,3}$ , Binarisierungsschwellwert  $\theta$ ,
Breite  $lb$  und Höhe  $lh$  des lokalen Fensters, Ausfalltoleranz  $c$ .
Ausgabe: Raumpunkt  $P_t$  des Markers.
/*Initialisierung: Beginne mit ganzem Fenster, noch kein Ausfall*/
1:  $GanzesFenster_{1,2,3} \leftarrow \text{true}$ ,  $z \leftarrow 0$ 
2: for all Kamerabilder  $I$  do
3:                                     /*Positionierung und Bemaßung des lokalen Fensters*/
4:   if  $GanzesFenster$  then
5:      $W_x \leftarrow W_y \leftarrow 0$ ,  $W_b \leftarrow \text{Breite}_I$ ,  $W_h \leftarrow \text{Höhe}_I$ 
6:      $W \leftarrow I$ 
7:   else
8:      $W_x \leftarrow mx - \frac{lb}{2}$ ,  $W_y \leftarrow my - \frac{lh}{2}$ ,  $W_b \leftarrow lb$ ,  $W_h \leftarrow lh$ 
9:      $W \leftarrow \text{Teilfenster}(I, W_x, W_y, lb, lh)$ 
10:  end if
                                     /*Binarisierung und Markerdetektion*/
11:   $W_b \leftarrow \text{Binarisiere}(W, \theta)$ 
12:   $\vec{R} \leftarrow \text{Zusammenhangsanalyse}(W_b)$ 
13:   $\vec{R}' \leftarrow \text{Regionenanalyse}(W_b, \vec{R}, K, L, A)$ 
                                     /*Detektions- und Fehlerbehandlung*/
14:  if  $\exists R_M \in \vec{R}' : \text{Marker}_{K,L,A,\vec{R}}(R_M)$  then
15:     $GanzesFenster \leftarrow \text{false}$ 
16:     $Verloren \leftarrow \text{false}$ 
17:     $(mx, my) \leftarrow \text{Schwerpunkt}(R_M)$ 
18:  else
19:     $GanzesFenster \leftarrow \text{true}$ 
20:     $z \leftarrow z + 1$ 
21:    if  $z > c$  then
22:       $Verloren \leftarrow \text{true}$ 
23:    end if
24:  end if
25: end for
                                     /*Rekonstruktion*/
26: if (  $Verloren_{1,2,3} = \text{false}$  ) then
27:   $(u_1, v_1) \leftarrow (mx_1 + W_{x,1}, my_1 + W_{y,1})$ 
28:   $(u_2, v_2) \leftarrow (mx_2 + W_{x,2}, my_2 + W_{y,2})$ 
29:   $(u_3, v_3) \leftarrow (mx_3 + W_{x,3}, my_3 + W_{y,3})$ 
30:   $P_t \leftarrow \text{Rekonstruiere3D}((u_1, v_1), (u_2, v_2), (u_3, v_3))$ 
31: end if

```

**Algorithmus 5.3:** Algorithmus zur segmentierungsbasierten Verfolgung von Markerbewegungen

Zur Beschleunigung wurde ein Ringpuffer für die Kamera- und Binärbilder verwendet. Dies unterstützt die parallele Ausführung von Binarisierung und Markersuche sowie der Positionsschätzung und der Kamerakopfsteuerung. Daraus ergeben sich auf Mehrprozessor-

rechnern Geschwindigkeitsvorteile.

Die Binarisierungsschwelle  $\theta$  kann aufgrund der Einfachheit des Markers in Grenzen adaptiv nachgeführt werden. Zur Bestimmung der Binarisierungsqualität werden der minimale und maximale Feret-Durchmesser des großen Markerkreises gemessen und bei Verlassen eines Toleranzbereichs  $\theta$  angehoben bzw. gesenkt.

### Fusion aus visuellen und magnetfeldbasierten Beobachtungsdaten

Die hohe Genauigkeit der Objektlokalisierung auch bei größerer Entfernung der Objekte zum Sensor durch die Bildverarbeitung ergänzt die ungenaueren Messungen des Magnetfeldsystems, das bei Verdeckungen immer referenziert werden kann. Tabelle 5.2 fasst die Vor- und Nachteile der insgesamt drei zur Verfügung stehenden Verfahren noch einmal zusammen. Eine Betrachtung des konturbasierten und des magnetfeldbasierten Verfahrens zeigt hier nachvollziehbar, dass sich die Stärken und Schwächen der beiden Verfahren komplementär ergänzen. Die Schwachpunkte der beiden Sensoren lassen sich deshalb weitgehend kompensieren, wenn geeignete Sensordatenfusionsverfahren angewandt werden.

Methode	MF	BBK	BBS
Zeitaufwand	++	+	
Speicheraufwand	++		
Stabilität bei Verdeckungen	++	--	-
Muster im Bildhintergrund	++	++	
Parametrierung	++	+	--
Genauigkeit in Sensornähe	++	+	+
Genauigkeit im Mittel	--	++	++

Tabelle 5.2: Vergleich von bild- und magnetfeldbasierter Handverfolgung (MF=Magnetfeldbasiert, BBK=Bildbasiert mit Verwendung von Konturmerkmalen und BBS=Bildbasiert mit Verwendung der Segmentierung)

Die erfassten Messwerte werden dazu je nach Sensormodell mit einem Vertrauens- bzw. Fehlerfaktor gewichtet, interpoliert und auf derselben Zeitbasis abgetastet (siehe Abbildung 5.13)<sup>13</sup>. Vor deren Fusion können noch Messungen ausgeschlossen werden: dies kann aufgrund von Heuristiken geschehen (etwa bei Auftreten zu großer Sprünge) oder aufgrund von sensorinternen Zuständen (etwa bei Melden des Verlustes des Markers aus dem Sichtbereich der Bildverarbeitung). In der Abbildung ist dies durch den ersten Verarbeitungsblock „Heuristische Vorverarbeitung“ angedeutet.

<sup>13</sup>Das gesamte Verfahren ist erstmals in [Ehrenmann 01c] veröffentlicht worden.

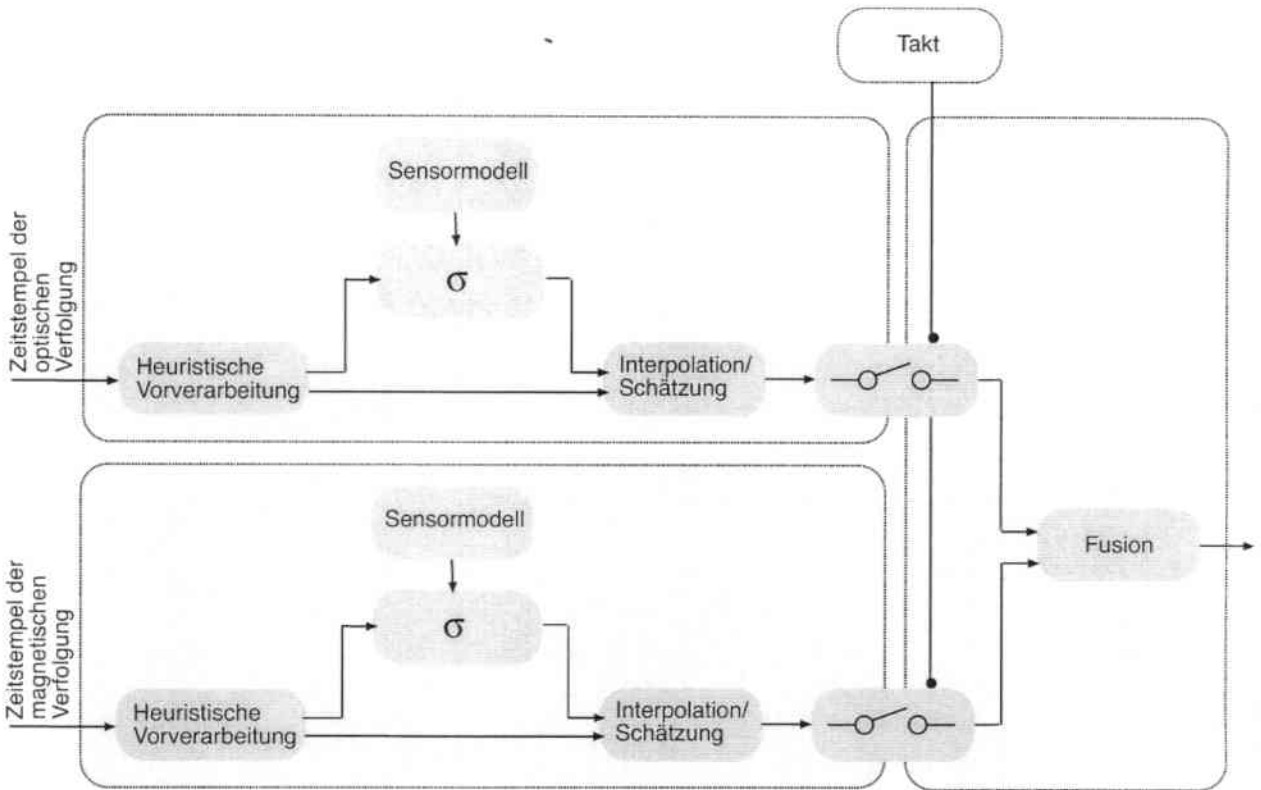


Abbildung 5.13: Schema der Sensordatenfusion aus optischen und magnetischen Messungen

Mathematisch lässt sich das Problem der Fusion durch einen gewichteten Mittelwert beschreiben. Bezeichnen  $\vec{e}_1$ ,  $\vec{e}_2$  die Unsicherheiten der beiden Sensortypen und  $\vec{x}_1$ ,  $\vec{x}_2$  deren Messungen zu einem Zeitpunkt, kann man notieren:

$$\vec{f} = \frac{\vec{x}_1 \cdot \vec{e}_2 + \vec{x}_2 \cdot \vec{e}_1}{\vec{e}_1 + \vec{e}_2} \quad (5.46)$$

Der Fehler  $\vec{e}$  wird hier der Einfachheit halber in allen Dimensionen gleich angenommen. Gesucht ist nun eine geeignete Bestimmung der Unsicherheiten für beide Sensoren.

**Magnetfeldbasierte Messungen:** Die Genauigkeit des Magnettrackingsystems hängt von der Stärke und Güte des erzeugten Magnetfeldes ab. Da dies quadratisch mit der Entfernung abnimmt, wird die Fehlerbeschreibung über die Distanz zum Emitter modelliert:

$$\vec{e} = f(\|\vec{P}_{\text{Emitter}} - \vec{P}_{\text{Empfänger}}\|) \quad (5.47)$$

Da das Magnetfeld wegen der in der Laborumgebung vorhandenen Metallgegenstände und elektrischen Leitern eine sehr starke Ablenkung erfährt, wird  $f$  durch eine Exponentialfunktion beschrieben:

$$\vec{e} = \begin{cases} k_1 \cdot \exp(k_2 \cdot \|\vec{P}_{\text{Emitter}} - \vec{P}_{\text{Receiver}}\|) & \text{wenn } \|\vec{P}_{\text{Emitter}} - \vec{P}_{\text{Receiver}}\| \leq \theta \\ \infty & \text{sonst} \end{cases} \quad (5.48)$$

mit  $k_1, k_2$  konstant. Wird eine gewisse Schwelle  $\theta$  überschritten, werden die Werte als fehlerhaft betrachtet, um die Messung nicht durch zu große Fehlmessungen zu beeinträchtigen.

**Bildbasierte Messungen:** Anders als im Fall der magnetbasierten Positionsmessung schließt im Fall der bildgestützten die Fehlerberechnung eine statische und eine dynamische Betrachtung ein:

$$\vec{e}_{\text{gesamt}} = \vec{e}_{\text{statisch}} + \vec{e}_{\text{dynamisch}} \quad (5.49)$$

$$\vec{e}_{\text{statisch}} = f_s(\|\vec{P}_{\text{Kamera}} - \vec{P}_{\text{Objekt}}\|, \text{Auflösung}) \quad (5.50)$$

$$\vec{e}_{\text{dynamisch}} = f_d(\vec{v}_{\text{Objekt}}) \quad (5.51)$$

Bei der Verfolgung von Objekten kommen durch Bewegungsunschärfe im Bild und Zeitversatz beim Auslesen der Gelenkencoder Fehler zum statischen Fall hinzu. Diese werden proportional zur Geschwindigkeit mit einem festen Faktor  $k_3$  modelliert:

$$\vec{e}_{\text{dynamisch}} = \vec{k}_3 \cdot \vec{v}_{\text{Objekt}} \quad (5.52)$$

Im statischen Fall wird der Zusammenhang für  $f$  über eine geometrische Überlegung gezeigt. Es gelte dabei näherungsweise:  $e_{\text{statisch}} = (e_x + e_y + e_z)/3$ . Das Schneiden der optischen Achsen (optische Zentren  $c_1, c_2$ ) der Kameras und dem Zielpunkt in der Szene zur Positionsrekonstruktion stellt sich wie in Abbildung 5.14 dar.

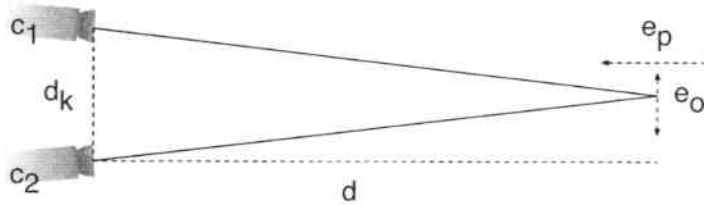


Abbildung 5.14: Rekonstruktionsfehler bei der bildbasierten Objektverfolgung

Fehler können hier orthogonal zur Bildaufnahmeffläche ( $e_o$ ) oder parallel zu ihr auftreten ( $e_p$ ). Dabei gilt im Regelfall  $e_p \gg e_o$  wegen  $d \gg d_k$ . Die maximale Abweichung eines Strahls sei  $e_{\text{max}}$ . Dieser Wert hängt ab von der Kameraauflösung, dem Kameraabstand  $d$  und der Brennweite:

$$e_{\text{max}} = f_k(d, \text{Auflösung}, \text{Brennweite}) \quad (5.53)$$

Die Berechnung von  $e_{\text{max}}$  erfolgt mit dem Strahlensatz (siehe Abbildung 5.15).

Wegen  $d \gg d_k$  kann  $\max e_o \sim e_{\text{max}}$  genähert werden.

Mit dem Szenenpunkt  $p$  kann aus Abbildung 5.16 abgelesen werden:

$$\sin \delta = \frac{e_{\text{max}}}{e_p} \quad (5.54)$$

$$e_p = \frac{e_{\text{max}}}{\sin \delta} \quad (5.55)$$



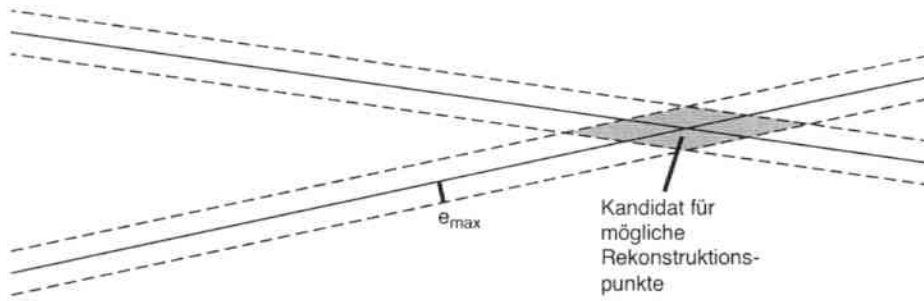


Abbildung 5.15: Lokalisierungsungenauigkeit

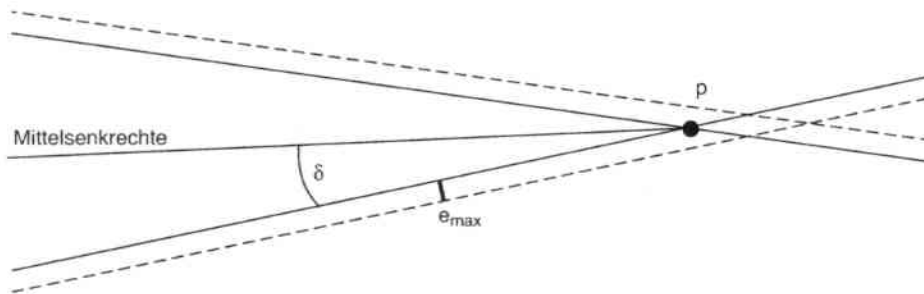
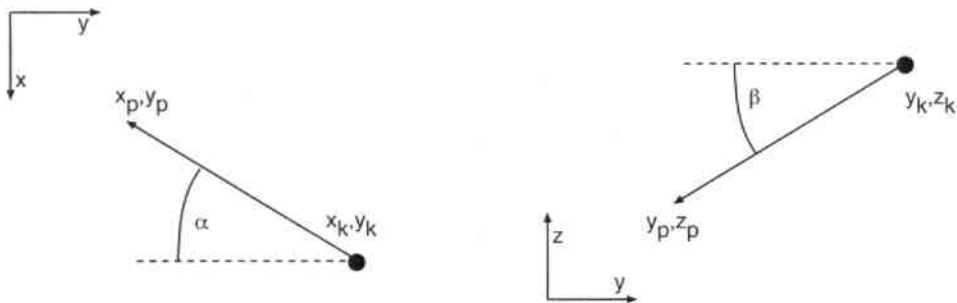


Abbildung 5.16: Winkelbestimmung

Abbildung 5.17: Strahlverfolgung in der  $x/y$ - (links) und in der  $y/z$  Ebene (rechts)

Damit können die Fehler  $e_x$ ,  $e_y$  und  $e_z$  aus  $e_p$  und  $e_o$  hergeleitet werden. Dabei sollen  $(x_k, y_k, z_k)$  die Kameraposition und  $(x_p, y_p, z_p)$  die Szenepunktkoordinaten bezüglich des Weltkoordinatensystems sein.

Aus Abbildung 5.17 folgt:

$$e_x = \frac{\cos \alpha \cdot e_o + \sin \alpha \cdot e_p}{\cos \alpha + \sin \alpha} \quad (5.56)$$

$$e_y = \frac{\sin \alpha \cdot e_o + \cos \alpha \cdot e_p}{\cos \alpha + \sin \alpha} \quad (5.57)$$

$$e_z = \frac{\cos \beta \cdot e_o + \sin \beta \cdot e_p}{\cos \beta + \sin \beta} \quad (5.58)$$

Damit lässt sich der Algorithmus 5.3 um die Fusion der kamera- und magnetfeldbasierten Positionsmessungen erweitern. Das zusätzliche Programmfragment ist in Algorithmus 5.4

aufgelistet.

```

Eingabe: Variablen  $P_t^B$  und Verloren aus Algorithmus 5.3 oder 5.2,
    synchrone Datenhandschuhmessung  $P_t^H$ .
Ausgabe: Fusionierter Raumpunkt  $P_t$  der Benutzerhand.
    /*Bei Verdeckung Rückgriff auf Magnetfeldsensor*/

1: if ( Verloren = true ) then
2:    $(\alpha, \beta) \leftarrow \text{Drehwinkel}(P_t^H)$ 
3:   BewegeKopf( $\alpha, \beta$ )
4:    $P_t \leftarrow P_t^H$ 
    /*Sonst Fusion*/

5: else
6:    $P_t \leftarrow \vec{f}(P_t^B, P_t^H)$ 
7: end if
8: Registriere( $P_t$ )
  
```

**Algorithmus 5.4:** Fusion von bild- und magnetfeldbasierter Positionsschätzung

### 5.2.2 Handverfolgung in der Ausführungsumgebung

In der Ausführungsumgebung des Roboters soll auf die Anwendung von Hilfsmitteln wie Markern oder invasiver Sensorik verzichtet werden. Daher können die bereits vorgestellten Verfahren nicht zum Einsatz kommen. Da der Bildhintergrund in Kamerabildern von Innenräumen eine starke Strukturierung aufweist, ist es nicht einfach, Merkmale für eine robuste Verfolgung auf Basis von Algorithmus 5.2 zu definieren. Es wird daher ein anderes Verfahren vorgeschlagen.

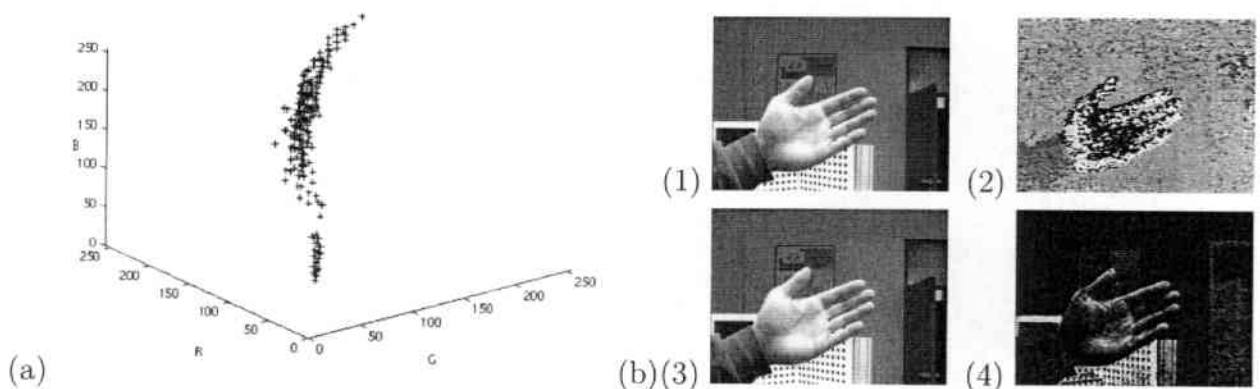


Abbildung 5.18: Verteilung von Hautfarbe bei konstanter Beleuchtung im *RGB*-Farbraum nach Yang (a) und Zerlegung eines Bildes (b 1) in Farb-, Helligkeits- und Sättigungsanteile (b, von 2 bis 4)

Die interessanten Regionen in Bildern mit Menschen sind deren Gesichter und Hände. Sie sind durch die Hautfarbe spezifisch charakterisiert. Eine Untersuchung der Hautfarbwerte

von 48 Personen in [Yang 98] zeigt, dass diese einen sehr eng umgrenzten Bereich im Farbraum einnehmen (Abbildung 5.18 a). Zusätzlich kann davon ausgegangen werden, dass Hautfarbe sehr selten bei anderen Objekten und Szenen in der Natur vorkommt. Die Hautfarbe ist ein guter Kandidat für die Segmentierung und der darauf aufbauenden Bewegungsverfolgung der Benutzerhand sowie des menschlichen Gesichts.

Eine Schwierigkeit stellt dabei jedoch die Farbkonstanz der Kameramessungen dar. Wenn farbliche Merkmale zur Objekterkennung oder Verfolgung benutzt werden, muss sichergestellt sein, dass dabei die Farbcharakteristik innerhalb gewisser Grenzen erhalten bleibt. Dies ist jedoch schon bei unterschiedlichen Lichtquellen nicht mehr der Fall. Abbildung 5.19 gibt die spektrale Zusammensetzung von Tageslicht bei bewölktem oder unbewölktem Himmel und verschiedenen künstlichen Lichtquellen an. Die Spektren zeigen sich hier als sehr unterschiedlich.

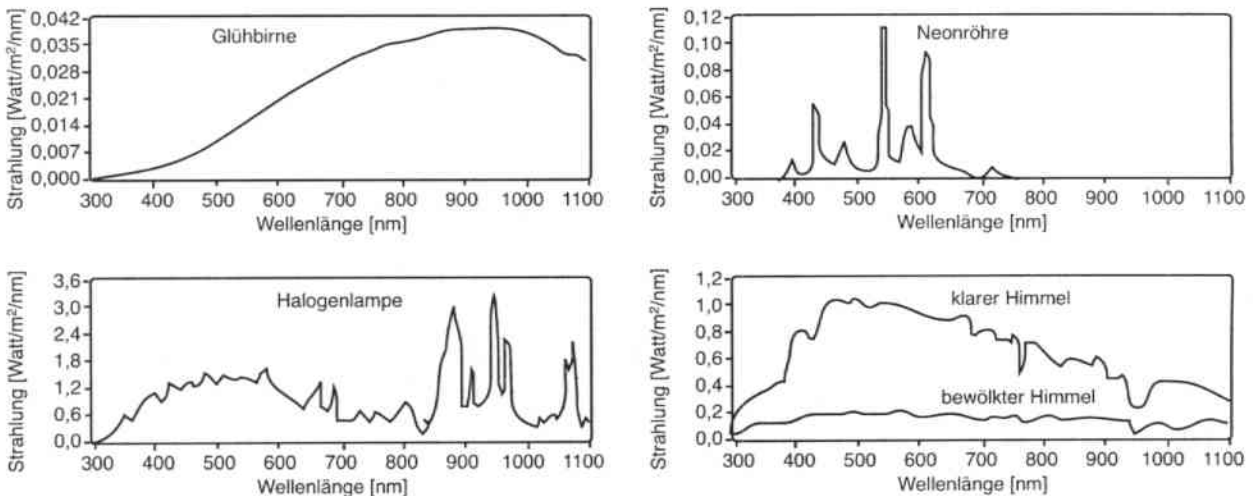


Abbildung 5.19: Lichtspektren verschiedener Beleuchtungsquellen nach [Störring 99]

Die digitalen Werte der Farbkanäle werden aufgrund von Beleuchtungsänderungen oft als im Farbwinkel verschoben wahrgenommen, was als „Farbstich“ bezeichnet wird. Für das Problem der Farbkonstanz gibt es noch keine zufriedenstellende technische Lösung [Störring 99]. Zum Ausgleich des Farbstiches besitzen Kameras oft behelfsmäßige Schaltungen mit Voreinstellungen der Verstärkungsfaktoren. Die in dieser Arbeit verwendeten Farbkameras<sup>14</sup> besitzen Modi für Innenraumbeleuchtung, Tageslicht und variable Beleuchtung, wobei letzterer die Farbanteile im Bild betrachtet, um eine adaptive Einstellung zu finden. Hierbei liegt eine *Graue-Welt-Annahme* zugrunde, nach der alle Farbanteile im Bild mit gleicher Häufigkeit vertreten sein müssen. Die Effekte dieser Einstellungen und eine Untersuchung verschiedener Beleuchtungsverhältnisse finden sich im Anhang D.

Da sich Roboter in der Regel in Räumen mit mehreren Lichtquellen bewegen, verbietet sich hier der Ansatz, mit Hilfe der Gamut-Mapping-Methode [Land 86, Jobson 95] eine farb-

<sup>14</sup>technische Daten siehe Tabelle B.1 im Anhang B

konstante Wahrnehmung zu modellieren (vgl. Abschnitt 2.3.2). Versuche mit dem Retinex-Verfahren zeigten einen Rechenaufwand, der Echtzeitbedingungen nicht genügt [Ly Duc 01]. Da sich die Einflüsse von Kameraeinstellung und verschiedenen Lichtquellen überlappen und ständig ändern, muss die Hautfarbsegmentierung also adaptiv erfolgen. Dies geschieht unter der Prämisse, dass sich die spektrale Zusammensetzung der Hautfarbe nicht sprunghaft ändert. Abbildung 5.18 zeigt rechts die Zerlegung eines Kamerabildes in seine Farb-, Helligkeits- und Sättigungsanteile (*HLS*-Anteile). Da Lichtschwankungen die Hautfarbsegmentierung nicht stören dürfen, wird diese Darstellung wegen der expliziten Codierung der Helligkeit zur Bildrepräsentation anstelle des *RGB*-Formats genutzt<sup>15</sup>. Versuche, die Farbe und Sättigung in die Segmentierung einzubeziehen, haben gezeigt, dass die Sättigung keine große Spezifität für Hautfarbe hat. Dies wird auch in der Abbildung deutlich: im Teilbild für die Farbe zeigt die Hand sehr helle und sehr dunkle Flächen. Der Farbwinkel um 0 bzw. 255 repräsentiert dementsprechend Rot. Unterschiedliche Sättigungscharakteristika treten jedoch überall im Bild auf. Das Adaptionsverfahren betrachtet daher ausschließlich den Farbwert und besteht aus drei Stufen:

**Initialisierung:** Zunächst werden Startschätzungen für den Mittelwert und die Varianz der Hautfarbpräferenz ermittelt. Dazu werden aus einem festen Bildausschnitt in der Bildmitte alle Bildpunkte als hautfarben angenommen, um den arithmetischen Durchschnitt und die Varianz des Farbwertes im *HLS*-Raum zu bestimmen. Der Benutzer muss dazu seine Hand kurz vor die Kamera halten. Die Größe des Ausschnitts wurde auf ein Fünftel der Bildbreite und -höhe festgelegt; diese bezieht keine Nichthautareale in die Mittelung ein und ist dennoch repräsentativ, da Überbelichtungen und Schattenwürfe in der Regel ausgeglichen werden.

**Segmentierung:** Die Binarisierung in Haut- und Nichthautfarbareale erfolgt mit Hilfe der ermittelten Referenzwerte  $\mu_{\text{Haut}}$ ,  $\sigma_{\text{Haut}}$  und einem manuell wählbaren Faktor  $k$  zur Anpassung des Segmentierungsintervalls:

$$p(x, y) := \begin{cases} 1 & \text{wenn } \mu - k \cdot \sigma < p(x, y) < \mu + k \cdot \sigma \\ 0 & \text{sonst} \end{cases} \quad (5.59)$$

Anschließend wird das segmentierte Binärbild durch die Anwendung von *Open*- und *Close*-Filtern geglättet (zur Erklärung der Filter siehe beispielsweise [Matrox 98b]).

Bevor die Hand als solche identifiziert ist, wird über eine Heuristik eine Unterscheidung zwischen Kopf, linker und rechter Hand gemacht (siehe Abbildung 5.20). Hierbei gelten Hautfarbregionen mit einer gewissen Größe als Kandidaten für Kopf oder Hand. Formmerkmale werden in dieser Arbeit nicht betrachtet. Sobald drei Kandidaten gefunden sind, wird diejenige Region mit dem kleinsten Ordinatenwert als Kopf markiert, die beiden anderen als Hände. Zur Initialisierung des Vorgangs müssen die folgenden Voraussetzungen erfüllt sein:

<sup>15</sup>Die Definitionen der Farbräume finden sich im Anhang C.

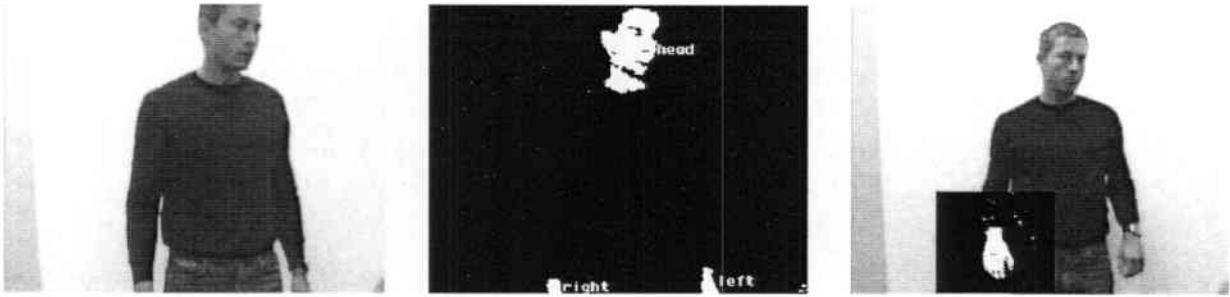


Abbildung 5.20: Identifikation von Kopf und zu verfolgender Hand

- Der Benutzer wendet dem Kamerakopf die Vorderseite zu.
- Der Benutzer steht aufrecht.
- Die Hände des Benutzers werden nicht über Kreuz gehalten.

Um die identifizierte rechte Hand wird dann ein lokales Fenster gelegt, anhand dessen die Adaption vorgenommen wird.

**Adaption:** Aus den nach der Segmentierung verbleibenden Arealen werden die größten Flecken im Binärbild selektiert und ihre korrespondierenden Bildpunkte im Originalbild zur Neuermittlung der Varianz und des Mittelwerts genutzt. Die Adaption dieser Referenzwerte erfolgt anschließend über eine stochastische Approximation einer gewichteten Summe aktueller und neu berechneter Referenzwerte:

$$\sigma_t = (1 - \alpha)\sigma_{t-1} + \alpha \cdot \sigma_{\text{segmentiert}} \quad (5.60)$$

$$\mu_t = (1 - \alpha)\mu_{t-1} + \alpha \cdot \mu_{\text{segmentiert}} \quad (5.61)$$

Die Adaptionsrate  $\alpha$  ist hier ein wählbarer Faktor, der den Einfluß von Farbveränderungen festlegt.

Zur Verkürzung der Verarbeitungszeit und zur Reduktion fehlerhafter Einflüsse ähnlichfarbener Bildregionen arbeiten die Prozesse zur Segmentierung und Adaption in einem lokalen Fenster. Zur Dimensionierung des lokalen Fensters hat sich eine Verlängerung der beiden Hauptachsen des Hautfarbbereichs um 20% bewährt, die jeweils neu berechnet wird. Der Ablauf dieser Schritte ist in Algorithmus 5.5 zusammengefasst; die Positionsberechnung für das lokale Fenster erfolgt analog zu Algorithmus 5.3.

### 5.2.3 Operator zur Bewegungsverfolgung

Sei  $P_t$  der in den Algorithmen 5.5 oder 5.4 nebenläufig gewonnene Raumpunkt der Benutzerhand,  $(u_i, v_i)$  die Bildpunktkoordinaten in den Kamerabildern. Dann kann mit Hilfe der in Algorithmus 5.6 genannten Befehlsfolge die Bewegung der Benutzerhand im Weltmodell registriert werden. Der Kamerakopf wird zur Verfolgung der Hand immer nachgeführt. Diese Aufgabe erfüllt der Operator ebenfalls, solange das Handlungsmodell den Fokus auf die Benutzerhand setzt. Die Regelung erfolgt über den Abstand  $\Delta$  der Hand zum Bildzentrum. Dazu berechne eine Funktion  $f$  die Zusammenhänge zwischen  $\Delta$  und den Drehwinkeln. In

**Eingabe:** Farbkamerabilder  $I_1, I_2$ , Kalibrierausschnitt  $W_0$ , Farbakzeptanzparameter  $k$ , Mindest- und Obergrenze  $N_{\min}, N_{\max}$  für die Bildpunktanzahl der Hand, Adaptionsschrittweite  $\alpha$ , Faktor  $k$  zur Anpassung des Segmentierintervalls.

**Ausgabe:** Raumpunkt  $P_t$  der Benutzerhand.

```

/*Initialisierung zur Festlegung der Referenzwerte*/
1:  $W \leftarrow W_0$ 
2: Bestimme  $\mu, \sigma$  über  $W$ 
/*Segmentierung im Binärfeld Hautfarbe $_W$  */
3: for all Bildpunkte  $p \in W$  do
4:   if  $\mu - k \cdot \sigma < \text{Farbwert}_p < \mu + k \cdot \sigma$  then
5:     Hautfarbe $_p \leftarrow \text{true}$ 
6:   else
7:     Hautfarbe $_p \leftarrow \text{false}$ 
8:   end if
9: end for
10: Appliziere Open- und Closefilter auf Hautfarbe $_W$ 
/*Adaption der Hautfarbeparameter  $\mu, \sigma, W$ */
11:  $\hat{\mu} \leftarrow$  Mittelwert von  $\{\text{Farbwert}_p | p \in W \text{ mit Hautfarbe}_p\}$ 
12:  $\hat{\sigma} \leftarrow$  Varianz von  $\{\text{Farbwert}_p | p \in W \text{ mit Hautfarbe}_p\}$ 
13:  $\mu \leftarrow (1 - \alpha)\mu + \alpha \cdot \hat{\mu}$ 
14:  $\sigma \leftarrow (1 - \alpha)\sigma + \alpha \cdot \hat{\sigma}$ 
15: Berechne Schwerpunkt  $s$ , Breite  $b$  und Höhe  $h$  von  $\{p \in W | \text{Hautfarbe}_p\}$ 
/*3D-Rekonstruktion*/
16:  $(u_1, v_1) \leftarrow (s_x, s_y)$ 
17:  $(u_2, v_2) \leftarrow \text{FindeKorrespondenz}(s, I_1, I_2)$ 
18:  $P_t \leftarrow \text{Rekonstruiere3D}((u_1, v_1), (u_2, v_2))$ 
19: Setze  $W$  entsprechend neu und erweitere um einen Randbereich
20: if  $N_{\min} < |\{p \in W | \text{Hautfarbe}_p\}| < N_{\max}$  then
21:   Verloren  $\leftarrow \text{false}$ 
22: else
23:   Verloren  $\leftarrow \text{true}$ 
24: end if

```

Algorithmus 5.5: Adaptive Hautfarbe-segmentierung

Abhängigkeit von  $|\Delta|$  werden von der Funktion „BewegeKopf“ auch die Drehgeschwindigkeiten  $(\omega_\alpha, \omega_\beta)$  für die Kopfmodule parametrisiert.

## 5.3 Grifferkennung

Das Problem der Grifferkennung mit den Sensorwerten des Datenhandschuhs stellt sich wie folgt dar: Die zu klassifizierenden Trainingsdaten sind hochdimensional und liegen ausschließlich numerisch vor. Die Klassifikation erfolgt datenbasiert und zur Erstellung der Klassifikatoren werden bereits vorhandene Trainingsdaten vorklassifiziert. Sie bestehen dann aus



**Eingabe:** Raumpunkt  $P_t$  der Benutzerhand, Variable *Verloren*,  
 Bildpunktkoordinaten  $(u_i, v_i)$  der Benutzerhand,  
 Breite  $b$  und Höhe  $h$  des Kamerabildes  $I$ , Zusammenhang  $f, f'$  zwischen Abstand zur fovealen Position und Kamerakopfdrehwinkeln.

**Ausgabe:** Kamerakopfbewegung und Registrierung des Raumpunktes  $P_t$  der Benutzerhand.  
 /\*Registrierte geschätzte Koordinaten\*/

```

1: Registriere( $P_t$ )
                                                    /*Fokussiere Kamerakopf*/
2: if  $\neg$  Verloren then
3:    $c_x \leftarrow b/2, \quad c_y \leftarrow h/2$ 
4:    $\Delta_x \leftarrow u_1 - c_x, \quad \Delta_y \leftarrow u_2 - c_y$ 
5:    $(\alpha, \beta) \leftarrow (\alpha, \beta) + (f(\Delta_x), f'(\Delta_y))$ 
6:   BewegeKopf( $(\alpha, \beta)$ )
7: end if
  
```

**Algorithmus 5.6:** Elementarer kognitiver Operator zur Bewegungsverfolgung

Paarungen  $(W, G)$  mit einem Fingerwinkelvektor  $W$  und dem entsprechenden Griffotyp  $G$  entsprechend der Modellierung in Abschnitt 4.6.2. Während wiederholter Nutzungen des Programmiersystems fällt jedesmal erneut eine Menge von Trainingsdaten an, die zur Verbesserung der Klassifikatoren genutzt werden können. Die nutzbare Trainingsmenge wächst daher inkrementell.

### 5.3.1 Kalibrierung des Datenhandschuhs

Problematisch sind die geometrisch und kinematisch unterschiedlichen Hände unterschiedlicher Benutzer. So treten bei Nutzung des Datenhandschuhs entsprechend des jeweiligen Benutzers Variationen in den Gelenkdaten auf, da der Handschuh in der Regel nicht genau in derselben Form anliegt und bei der Beugung einzelner Gelenke Verziehnungen des Handschuhs auftreten. So kann z.B. das Beugen des Zeigefingers bei gleichzeitiger Streckung des Mittelfingers eine Veränderung des Spreizwinkels zwischen Zeige- und Mittelfinger nach sich ziehen, obwohl keine Veränderung desselben aufgetreten ist. Die Temperaturabhängigkeit der Dehnungsmeßstreifen trägt zudem in geringem Maße dazu bei, die Meßwerte ohne Änderung der übrigen Umweltbedingungen schwanken zu lassen. Außer den genannten Störungen haben die Handgeometrien einen großen Einfluß auf die gemessenen Werte  $W$ , da sich mit unterschiedlichen Ausprägungen die Lage der Dehnungsmeßstreifen ändert.

Die Gelenkdaten  $W$  werden daher zunächst normalisiert. Dazu werden in einem Kalibrierungsschritt der minimale und der maximale Winkel jedes einzelnen messbaren Gelenks ermittelt. Die Differenz zwischen einem später gemessenen Winkel und dem Minimum wird dann im Verhältnis zum gesamten überdeckbaren Winkelbereich betrachtet:

$$w_i^n = \frac{w_i - w_{i,\min}}{w_{i,\max} - w_{i,\min}} \quad (5.62)$$

Die Messwerte  $\vec{w} = (w_1, \dots, w_{20})^T$  liegen damit alle im Bereich  $[0, 1]$  und sind von der Handgeometrie unabhängig, da jede Person ihre maximalen Beugungs- und Spreizungswin-

kel vorgibt<sup>16</sup>. Durch die Varianz in einer ausreichend großen Mustermenge kann eine weitere Stabilisierung der Erkennung erreicht werden.

### 5.3.2 Schichtklassifikator mit Verwendung objektspezifischer Griffinformation

Da die Griffklassifikation auch Klassen anhand der manipulierten Objekte unterscheidet, wurden zur Aufnahme der Trainingsdaten  $W$  entsprechende Objekte gehalten und der resultierende Satz von Fingergelenkwinkeln aufgezeichnet. Wenn alle Objekte im Weltmodell mit dem in Tabelle 4.1 modellierten Attribut ausgestattet sind, das über mögliche Griffotypen informiert, lassen sich die Klassifizierungen robuster gestalten bzw. Rückfragen im Fehlerfall stellen. Dazu wird bei einer entsprechenden Fingerkonfiguration  $\vec{w}$  überprüft, ob die der Hand nächsten Objekte eine zum erkannten Griffotyp passende Griffart zulassen.

Der angewandte Klassifikator basiert auf der Arbeit [Friedrich 98]. Dort wurden konnektionistische Verfahren eingesetzt. Als Aktivierungsfunktionen dienen lokale Aktivierungsfunktionen, da sich diese bei der Realisierung inkrementeller Lernaufgaben globalen Aktivierungsfunktionen gegenüber überlegen erweisen. Die vorliegende Lernaufgabe wurde mit Hilfe von dreischichtigen RBF-Netzen gelöst, die auf Basis aufgenommener und vorklassifizierter Beispieldatensätze  $(W, G)$  für jede Hierarchieebene der Cutkosky-Griffhierarchie einen speziellen Klassifikator zur Verfügung stellen (Abbildung 5.21). Die Entscheidung für eine Griffklasse fällt zunächst zugunsten derjenigen Ausgabeneuronen, welche die höchste Aktivierung aufweisen. Der Klassifikationsvorgang von Fingerwinkeldatensätzen erfolgt bei diesen hierarchisch angeordneten Klassifikatoren ähnlich einem Entscheidungsbaumverfahren. Begonnen wird mit dem Schicht-1 Netzwerk, welches zwischen Kraft- und Präzisionsgriffen unterscheidet. Je nach Klassifikationsergebnis wird dann das Schicht-2 oder das Schicht-3 Netzwerk bzgl. des zu klassifizierenden Datensatzes ausgewertet und so fort. Der Klassifikationsprozeß endet, sobald die erkannte Ausgabeklasse eine Griffklasse repräsentiert.

Zusätzlich zu diesem Klassifikator werden weitere Informationen zur Griffbestimmung genutzt. Die endgültige Entscheidung für die Detektion eines Griffotyps wird unter Betrachtung der dem Endeffektorpunkt nahe liegenden Objekte gefällt. Diese Objekte werden mit steigender Distanz sortiert und ihre Greifklassen mit den höchstaktivierten Ausgabeneuronen verglichen. Wird kein passendes Paar gefunden, muss eine Rückfrage an den Benutzer gestellt werden.

Die Detektion von Griffen kann robust gestaltet werden, wenn zu der Handkonfiguration weitere Information betrachtet wird. So belegen Studien, dass die Trajektorie der Hand bei Menschen zwischen Griffen in kartesischen Koordinaten nahezu eine gerade Linie mit glockenförmigem Geschwindigkeitsprofil ist [Hauck 98]. Die Winkelmessungen brauchen also

<sup>16</sup>Die Messwerte für die Beugung des Handgelenks werden nicht betrachtet. Auf die Klassifikation des Griffotyps haben sie keinen Einfluss.

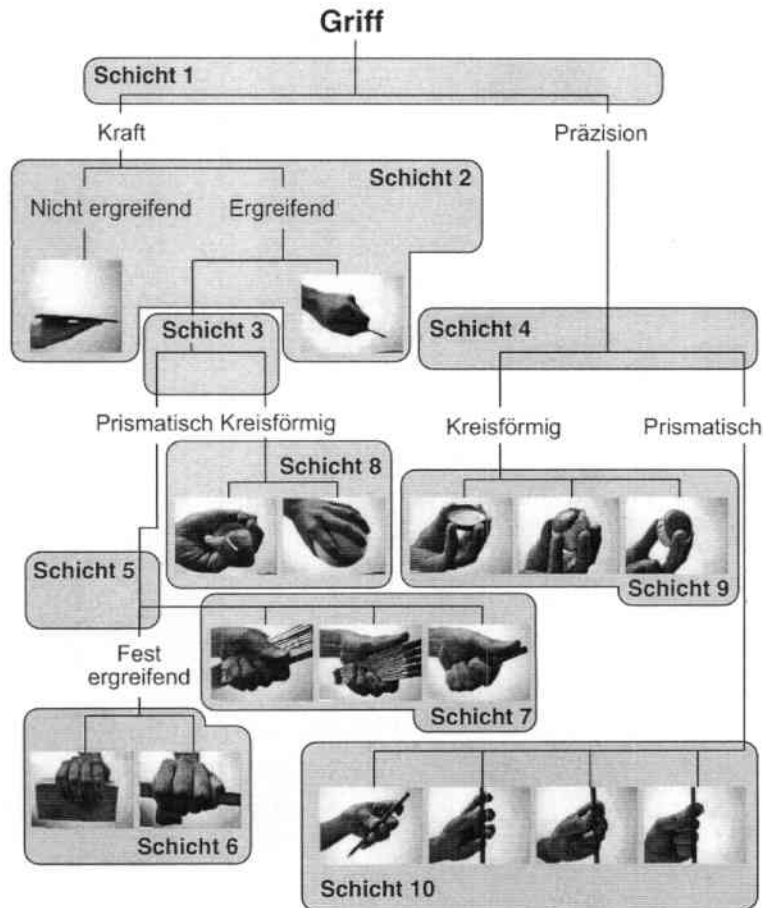


Abbildung 5.21: Schichten der Cutkosky-Griffhierarchie und korrespondierende Klassifikatoren

nur dann durch den Klassifikator propagiert zu werden, wenn die Bewegungsgeschwindigkeit sowohl der Hand mit dem Unterarm als auch der einzelnen Fingergelenke gering genug ist. Dies hat zur Folge, dass:

- große, kurzzeitige Störungen gar nicht den Klassifikator erreichen.
- Handhaltungen, die im Übergang zwischen Gesten oder Griffen kurzzeitig auftreten, nicht berücksichtigt werden müssen.

Das Geschwindigkeitsmaß hat sowohl die Bewegung der Hand als auch der Finger zu berücksichtigen. Es lässt sich definieren als euklidische Distanz zwischen zwei Abtastungen der Hand mit den Gelenkwinkeln  $w_i$ , den Orientierungswinkeln  $\alpha_i$  und den Positionsmessungen  $x_k$  entlang der Koordinatenachsen:

$$v_G = a \cdot \frac{\sum_i |w_{i,t} - w_{i,t-1}|}{\Delta t} + b \cdot \frac{\sum_j |\alpha_{j,t} - \alpha_{j,t-1}|}{\Delta t} + c \cdot \frac{\sum_k |x_{k,t} - x_{k,t-1}|}{\Delta t} \quad (5.63)$$

Dabei sind die einzelnen Summanden entsprechend ihres Einflusses mit den Faktoren  $a$ ,  $b$  und  $c$  zu gewichten. Insbesondere gehen die Orientierungswerte mit höheren Gewichtungen

ein, da die Absolutwerte der radianten Winkeldarstellung relativ klein ausfallen.

### 5.3.3 Operator zur Griffdetektion

Algorithmus 5.7 fasst die obigen Schritte zusammen. Die Hand- und Fingergeschwindigkeit  $v_G$  berechnet sich hier nach Gleichung 5.63. Voraussetzung ist die Existenz einer Funktion „Propagiere-Hierarchisch“, welche die Eingabekonfiguration  $\vec{w}$  entsprechend Abbildung 5.21 durch die einzelnen Klassifikatornetze propagiert, bis die Ausgabeschicht eines der Schichtnetze erreicht wird.

Es wird überprüft, ob sich die Benutzerhand in der Nähe eines Objektes befindet und sich dort langsam bewegt (Zeilen 1 und 2). In diesem Fall wird die Handkonfiguration an den Klassifikator angelegt und diejenigen Ausgabeneuronen mit den höchsten Aktivierungen gespeichert (Zeilen 3 bis 7). Entsprechend werden die nächstliegenden Objekte gespeichert (Zeilen 8 und 9). Anschließend werden Paare gebildet, die hinsichtlich der Plausibilität des Grifftyps überprüft werden (Zeilen 10 bis 17). Das erste gefundene Paar wird registriert. Höheren neuronalen Aktivierungen wird dabei der Vorrang gegenüber den Objektabständen gegeben.

## 5.4 Gestenerkennung

Wie im Fall der Grifferkennung sollen auch Gesten möglichst benutzerunabhängig erkannt werden. Die Detektion sollte robust gegenüber Positionswechseln, Drehungen oder Verkip-pungen der Hand sein. Während die Gestenerkennung in der Vorführungsumgebung über die Fingerstellungsmessungen des Datenhandschuhs erfolgen kann, werden dazu im Fall der Ausführungsumgebung bildbasierte Methoden eingesetzt.

### 5.4.1 Erkennung statischer Gesten in der Vorführungsumgebung

Für die Problemstellung bei der Gestenklassifikation auf Basis der Sensorwerte des Datenhandschuhs gilt das in Abschnitt 5.3 gesagte. Die zur Verfügung stehenden Trainingsdaten bestehen hier ebenfalls aus Paarungen  $(W, G)$  mit einem Fingerwinkelvektor  $W$  und einem Gestentyp  $G$ . Auch hier sind daher konnektionistische Verfahren zur Klassifikation geeignet.

Während die Trainingsdaten zur initialen Erstellung der Klassifikatornetze im Fall der Griffe aus Mustern bestehen, bei denen ein Objekt in einer gewissen Weise gegriffen wird und damit die Variationsbreite der Handkonfigurationen stark eingeschränkt ist, erhielten die Probanden zur Generierung der Muster, die zu Trainingszwecken gesammelt wurden, lediglich eine verbale Beschreibung der Gestentypen. Dies hat den Zweck, Handhaltungen als Muster zu nutzen, die natürlich ausgeführt werden.

Da die Modellgesten sich nicht sinnvoll hierarchisch anordnen lassen, wurde zur Lösung des Klassifikationsproblems ein einzelnes dreischichtiges neuronales Vorwärts-Netz<sup>17</sup> gewählt.

<sup>17</sup>engl.: Feed Forward Network

**Eingabe:** Gemessene Handkonfiguration  $\vec{w}_t$  und  $\vec{w}_{t-1}$  des Datenhandschuhs, sowie Position  $\vec{p}_t$  und  $\vec{p}_{t-1}$  und Orientierung  $\vec{\phi}_t$  und  $\vec{\phi}_{t-1}$  der Benutzerhand, Schwellwerte  $\theta_H$ ,  $\theta_O$  und  $\theta_G$  für die Erfassung der Handbewegung und ihrer Nähe zu Objekten sowie als Mindestwert für die neuronale Aktivierung, trainierte neuronale Schichtnetze  $N$ , Objektliste  $D$  des Weltmodells mit assoziierten Positionen.

**Ausgabe:** Registrierung des Grifftyps mit Objekt und Zeitpunkt.

```

1: if  $v_H(\vec{w}_t, \vec{w}_{t-1}, \vec{p}_t, \vec{p}_{t-1}, \vec{\phi}_t, \vec{\phi}_{t-1}) < \theta_H$  then
2:   if  $\exists d \in D : \|\vec{d}_p - \vec{p}_t\| < \theta_O$  then
3:     BelegeEingabeschicht( $N, \vec{w}_t$ )
4:     Propagiere-Hierarchisch( $N$ )
5:     Rückfrage  $\leftarrow$  True
6:      $N' \leftarrow \{n \in N : a_n > \theta_G\}$ 
7:     SortiereAbsteigend $_a(N')$ 
8:      $D' \leftarrow \{d \in D : \|\vec{d}_p - \vec{p}_t\| < \theta_O\}$ 
9:     SortiereAufsteigend $_d(D')$ 
10:    for all  $n \in N'$  do
11:      for all  $d \in D'$  do
12:        if GriffTyp( $n$ )  $\in$  GriffTypenFürObjekt( $d$ ) then
13:          Registriere(GriffTyp( $n$ ),  $t, \vec{p}_t, \vec{\phi}_t, d$ )
14:          Rückfrage  $\leftarrow$  False
15:        end if
16:      end for
17:    end for
18:    if Rückfrage=True then
19:      Registriere(Rückfrage,  $t, \vec{p}_t, \vec{\phi}_t, D', N'$ )
20:    end if
21:  end if
22: end if

```

**Algorithmus 5.7:** Elementarer kognitiver Operator zur Griffdetektion

Die minimale Anzahl der notwendigen Neuronen in der versteckten Schicht zur bestmöglichen Klassifikation wurde experimentell ermittelt. Dazu wurden die aufgenommenen Musterbeispiele  $W$  in eine Trainings-, Test- und Topologiemenge unterteilt (siehe Abschnitt 6.1.4). Die Gewichte des Netzes wurden mit der Trainingsmenge unter Verwendung des RProp-Verfahrens trainiert [Riedmiller 92]. Dieses Verfahren zeichnet sich durch glatte Lernkurven aus und vermeidet durch frühes Unterbrechen des Lernvorgangs<sup>18</sup> eine Überanpassung und damit den Verlust der Generalisierungsfähigkeit.

<sup>18</sup>engl.: Early Stopping

Es ist sinnvoll, sich bei der Betrachtung statischer Gesten auf solche zu konzentrieren, die ohne Bewegung der Hand ausgeführt werden. In diesem Fall treffen die Einschränkungen des Entscheidungskriteriums der Griffklassifikation auch hier zu (siehe Abschnitt 5.3.3). Die Fingerwinkel  $W$  werden daher lediglich dann durch den Klassifikator propagiert werden, wenn die Bewegungsgeschwindigkeit der Hand und der Finger unter einer Schwelle liegen.

### 5.4.2 Erkennung statischer Gesten in der Ausführungsumgebung

Die bildbasierte Erkennung statischer Gesten wird in der Ausführungsumgebung eingesetzt. Sie verwendet als Grundlage der Klassifikation von Handkonfigurationen den Umriss der Hand-silhouette. Deshalb kann sie auf Basis von Differenzbildern oder der Hautfarbsegmentierung aus Abschnitt 5.2.2 ablaufen<sup>19</sup>.

Die Analyse der Handstellung besteht aus den drei Schritten Vorverarbeitung, Merkmalsextraktion und der eigentlichen Klassifikation der ausgeführten Geste:

- Die Segmentierung der Hand und Glättung der Handregion erfolgt in der Ausführungsumgebung nach Algorithmus 5.5. Danach wird der Umriss der Hand verfolgt und in Form der abgelaufenen Bildpunkte<sup>20</sup> gespeichert. Diese Schritte finden sich anschaulich in Abbildung 5.22 wieder.

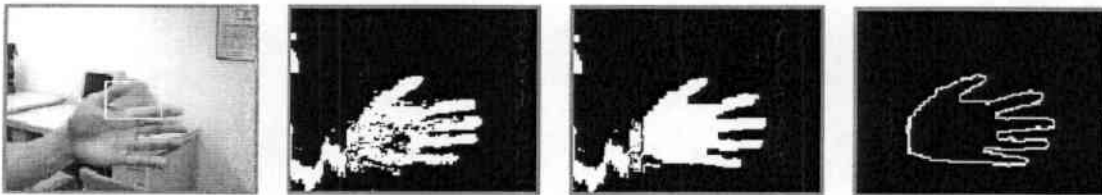


Abbildung 5.22: Vorverarbeitungsschritte zur Gestenerkennung

- Die gewonnene Konturkette wird nun an einer festen Anzahl von Abtastpunkten betrachtet, um Vergleiche durchführen zu können und den Rechenaufwand zu minimieren. Aus den Abtastpunkten werden mittels schneller Fouriertransformation<sup>21</sup> (FFT) Deskriptoren berechnet, die auf positions- und orientierungsunabhängige Werte normiert werden können.
- Der Deskriptorvektor wird anschließend mit der Referenzmenge von Deskriptorvektoren verglichen, um den Abstand zum Ähnlichsten zu bestimmen und die Geste falls möglich zu klassifizieren.

Das Vorgehen im zweiten und dritten Schritt wird im Folgenden näher diskutiert.

<sup>19</sup>Die Gestenerkennung auf Basis der Hautfarbsegmentierung wurde erstmalig in [Ehrenmann 02] publiziert. Dort wird neben der Verwendung der Gesten auch die reaktive Komponente des Roboters *Albert* besprochen.

<sup>20</sup>engl.: Chain Code

<sup>21</sup>engl.: Fast Fourier Transform



## Merkmalsextraktion

Die aus der Segmentierung erhaltene Konturkette besteht aus einer veränderlichen Punktmenge. Die Länge der Kette hängt von verschiedenen Faktoren ab: der Geste selbst (z.B. bei geschlossenen oder gespreizten Fingern), der Handgeometrie des Benutzers sowie dem Abstand zwischen Hand und Kamera. Im Abstand zwischen einem und zwei Metern schwankt die Punktezahl üblicherweise zwischen 200 und 1000 Elementen. Durch die Abtastung wird daraus eine kleinere, repräsentative Anzahl von Punkten ausgewählt. Eine feste Zweierpotenz als Anzahl von Punkten macht dabei die FFT direkt anwendbar und die Transformation in den Frequenzraum sehr schnell berechenbar.

Als Abtastpunkte werden in der Literatur solche an stark gekrümmten Stellen der Kontur vorgeschlagen, da diese hohen Informationswert haben [Kindratenko 96]. Diese zeigten sich in Versuchen bei der Rekonstruktion der Kontur aus den Deskriptoren jedoch als stärker rauschanfällig als äquidistant gelegene Punkte. Deshalb wurden letztere gewählt. Benutzt werden dabei 32 Abtastpunkte — diese Anzahl hat bei Versuchen zur Erkennungsleistung keine Nachteile gegenüber der doppelten Anzahl gebracht. Bei 8 bzw. 16 Punkten liegt die Konturabtastung jedoch oft unterhalb der Nyquiststrate (siehe Abbildung 5.23).

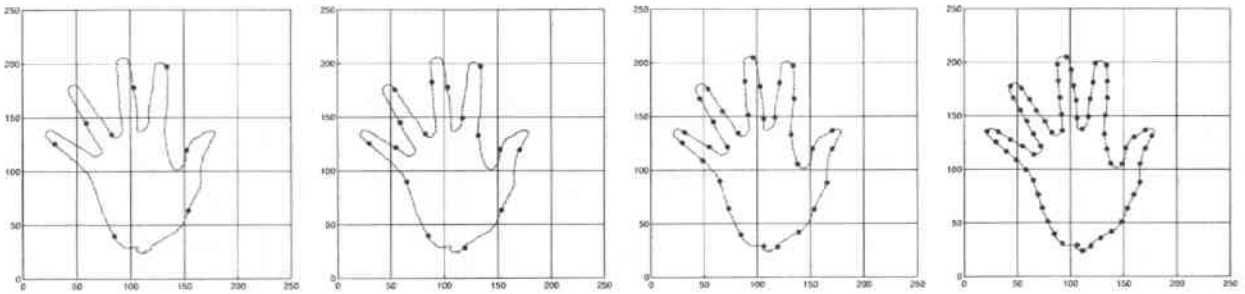


Abbildung 5.23: Abtastung der Handkontur mit 8, 16, 32 und 64 Abtastpunkten

Die abgetasteten Punkte können nun transformiert werden. Die somit erhaltenen Deskriptoren sollen jedoch invariant sein gegenüber Spezifika einzelner Personen, Skalierungen oder Rotationswinkeln. Zusätzlich wird Robustheit gegenüber dem Startpunkt der Konturkette gefordert: dieselbe Silhouette kann in Abhängigkeit vom Startpunkt des Konturablaufs verschiedene Repräsentationen haben. Die Fouriertransformation ist definiert durch:

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi\left(\frac{k \cdot n}{N}\right)} \quad (5.64)$$

wobei  $x(k) \in \mathbf{C}$  der Eingabevektor der Konturpunkte,  $X$  der Deskriptor als Ergebnis der Transformation,  $N$  die Anzahl der Abtastpunkte und  $k$  und  $n$  Zählvariablen im Bild- bzw. Frequenzraum sind. Die Transformation ist linear und weist für eine Normierung interessante Eigenschaften auf, von denen einige in Tabelle 5.3 aufgelistet sind.

Die dargelegten Beziehungen geben an, dass die Wirkung von Rotation und Skalierung auf eine Kontur direkt proportional auf den ganzen Deskriptor abgebildet werden, während die translatorische Verschiebung sowie die Änderung des Konturstartpunktes auf jedes Deskriptorelement unterschiedlichen Einfluß ausüben.

Transformation	Kontur (Ortsraum)	Deskriptor (Frequenzraum)
Identität	$s(k) = s(k)$	$S(n) = S(n)$
Rotation um $\theta$	$s_r(k) = s(k)e^{j\theta}$	$S_r(n) = S(n)e^{j\theta}$
Translation um $\Delta_{xy}$	$s_t(k) = s(k) + \Delta_{xy}$	$S_t(n) = S(n) + \Delta_{xy}\delta(n)$
Skalierung um $\alpha$	$s_s(k) = \alpha s(k)$	$S_s(n) = \alpha S(n)$
Verschiebung des Anfangspunktes	$s_p(k) = s(k - k_0)$	$S_p(n) = S(n)e^{-j2\pi k_0 n/N}$

Tabelle 5.3: Eigenschaften der Fouriertransformation

Es erweist sich jedoch als sehr einfach, die abgetastete Kontur bereits vor der Transformation hinsichtlich Translation und Skalierung zu normieren. Dazu wird zunächst der Schwerpunkt  $c_g = (\sum s_x(k)/N, \sum s_y(k)/N)$  bestimmt und die Koordinaten bezüglich dieses Punktes angegeben. Anschließend werden die Abtastpunkte mit den Dividenden  $\max_{s_x(k)} |s_x(k) - c_{gx}|$  bzw.  $\max_{s_y(k)} |s_y(k) - c_{gy}|$  in das Intervall  $[0, 1]$  normiert. Abbildung 5.24 fasst diese Schritte zusammen.

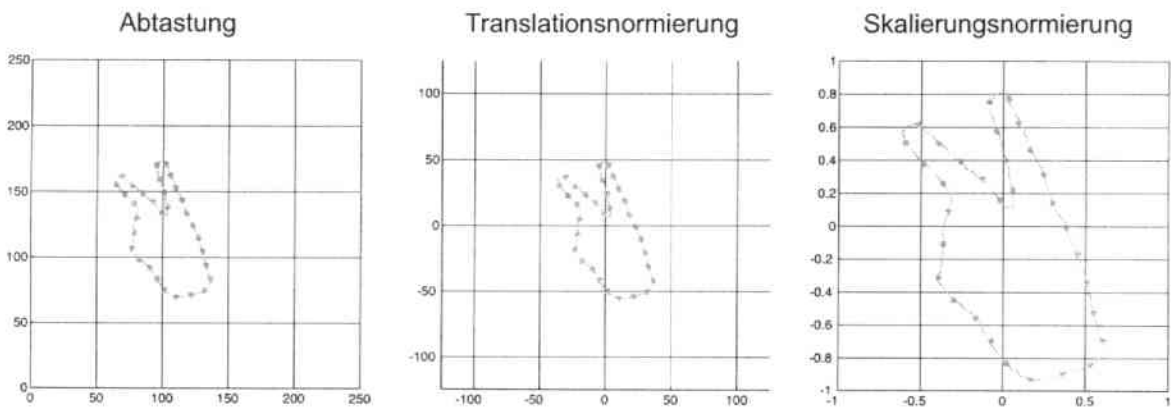


Abbildung 5.24: Vornormierung der Handkontur

Auf die so vornormierten Konturen kann dann die Fouriertransformation aus Gleichung 5.64 angewandt werden. Dabei werden die Koordinaten der Abtastpunkte als komplexe Zahlen interpretiert:

$$\{s(k)\} = \{x(k)\} + j \cdot \{y(k)\} \quad (5.65)$$

Der resultierende Fourierdeskriptor hat dieselbe Länge und die Gestalt

$$\{S(n)\} = \{S_x(n)\} + j \cdot \{S_y(n)\} \quad (5.66)$$

In Abbildung 5.25 sind links die Deskriptoren dargestellt, die bei Ausführung derselben Geste von fünf unterschiedlichen Personen erhalten wurden. Dabei waren die Hände unterschiedlich groß und unterschiedlich gedreht worden. Jeder Deskriptor ist in einer anderen Helligkeit

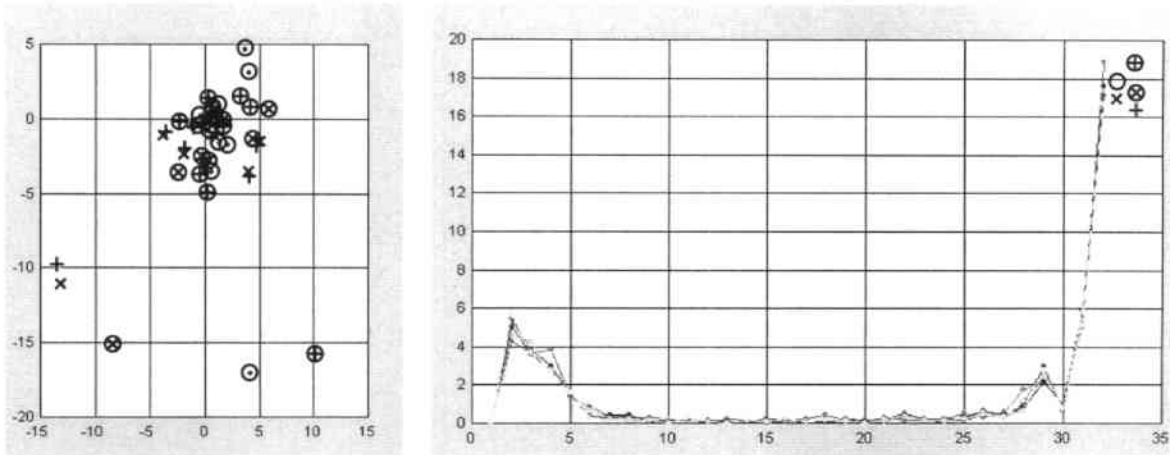


Abbildung 5.25: Ergebnisse der Fouriertransformation: links dieselbe Geste bei wechselnden Personen, rechts die entsprechenden Abstände vom Ursprung

abgebildet. Es wird erkennbar, dass die Punktverteilung bis auf den Drehwinkel dieselbe Struktur aufweist. Dies korrespondiert mit dem Formelzusammenhang in Tabelle 5.3: Rotation und der Konturstartpunkt beeinflussen nur den Winkel der Deskriptorelemente, aber nicht deren Abstand. Zur Normierung bezüglich dieser beider Eigenschaften wird daher eine neue Darstellungsform des Deskriptors benutzt, die den elementweisen Betrag beschreibt:

$$\{D(n)\} = \{|S(n)|\} = \{\sqrt{S_x^2(n) + S_y^2(n)}\} \quad (5.67)$$

Das Ergebnisdiagramm von  $\{D(n)\}$  zu Abbildung 5.25 links präsentiert sich im rechten Bild derselben. Die Abszisse gibt hier die Elementnummer an, die Ordinate den Deskriptorbetrag. Die Ähnlichkeit verschiedener Ausführungen eines Gestentyps ist hier sehr gut erhalten geblieben. Unterschiedliche Gestentypen zeigen auch verschiedenartige Betragsdeskriptoren (siehe Abbildung 5.26). Damit sind verschiedene Deskriptoren vergleichbar und können klassifiziert werden.

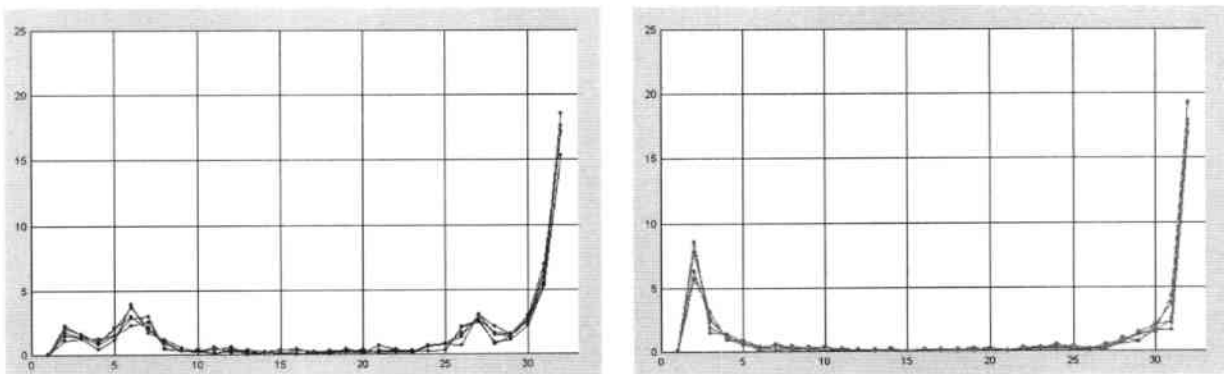


Abbildung 5.26: Betragsdeskriptoren von Gesten des Typs 4 und 6

## Klassifikation

Zur Klassifikation statischer Gesten müssen neben den Referenzdeskriptoren auch Kriterien zur Verfügung stehen, die eine Zuordnung eines Deskriptors gestatten. Eigentlich handelt es sich dabei um drei Kriterien: eines für den Vergleich zwischen Referenzen und dem fraglichen Deskriptor, eine Minimalähnlichkeit für die Akzeptanz einer Geste und ein Kriterium für die Wahl bzw. Organisation der Referenzdeskriptoren:

**Vergleichskriterium:** Als Ähnlichkeitsmaß wird hier entsprechend dem Ansatz zum Minimalabstand<sup>22</sup> [Gonzalez 93] die Euklidische Distanz  $\Delta^M$  zwischen einem Modelldeskriptor  $D^M$  und dem fraglichen Deskriptor  $D$  berechnet:

$$\Delta^M = \sum_{n=0}^{N-1} (D^M(n) - D(n))^2 \quad (5.68)$$

Abbildung 5.27 links enthält Diagramme mit den Abständen von fünf ausgeführten Gesten des Typs 1. Jeder Balken markiert den Abstand  $\Delta$  nach Gleichung 5.68 zu einer der Referenzgesten; in diesem Fall sind sieben Modelle vorhanden. Je kleiner der Balken, desto ähnlicher ist die Geste dem entsprechenden Referenzmodell. Der kleinste Ausschlag ist hier immer bei der ersten Referenzgeste.

**Entscheidungskriterium:** Ist unter Verwendung des Vergleichskriteriums aus der Bibliothek derjenige Referenzdeskriptor mit der kleinsten Distanz  $\Delta^M$  zur ausgeführten Geste gefunden, muss zur Vermeidung von Fehlern zweiter Art noch entschieden werden, ob die vorgeführte Geste auch eine geforderte Mindestqualität besitzt. Dazu wird ein Schwellwertfilter eingesetzt, der das Einhalten einer Mindestähnlichkeit fordert. In Abbildung 5.27 rechts sind die Abstände von drei unbekanntem Gesten zu den Referenzmodellen aufgetragen. Ein Vergleich mit der linken Bildhälfte zeigt, dass die Abstände hier deutlich größer sind als bei bekannten Konfigurationen.

**Modellaufbau:** Die Bibliothek, die alle klassifizierbaren Referenzdeskriptoren enthält, sollte möglichst benutzerunabhängig sein. Um zusätzlich gegenüber Rauschen in der Segmentierung sowie affinen Transformationen der Handsilhouette robust zu sein, kommt der Wahl eines guten Modells hohe Bedeutung zu. Es stellt sich genauer gesagt die Frage, wie aus einer Reihe von Gesten eines neuen Typs  $D'_{1,\dots,n}$  ein neuer Referenzdeskriptor  $D^M$  bestimmt wird. Zwei Methoden stehen dazu zur Auswahl: die Bestimmung des am besten passenden Deskriptors oder die Mittelung über mehrere.

**Auswahl des besten Deskriptors:** Im ersten Fall werden alle Musterdeskriptoren  $D^M$  mit den neuen Kandidaten  $D'_{1,\dots,n}$  verglichen. Die Euklidischen Distanzen  $\Delta$  werden für jeden Deskriptor  $D'_{1,\dots,n}$  summiert, um den Unterschied zur Bibliothek zu quantifizieren. Dann wird dasjenige Modell  $D_i$  gewählt, dessen Gesamtunterschied am größten ist.

**Mittelwertdeskriptor:** Der Durchschnittswert aus allen Deskriptoren ergibt den neuen Referenzdeskriptor, das heißt jedes Element wird als Mittelwert aus den entsprechenden Elementen der  $D'_{1,\dots,n}$  berechnet.

<sup>22</sup>engl.: Minimum Distance Classifier

Bei der ersten Methode können einzelne Werte, die im Modelldeskriptor enthalten sind, leicht Extrema annehmen. Dadurch werden in manchen Fällen die Abstände zu ausgeführten Gesten desselben Typs zu groß für eine korrekte Klassifikation. Das Vorgehen mit der elementweisen Mittelung über mehrere Beispiele behält die Information verschiedener Handgeometrien bei. Die sich ergebende Verteilung der Ähnlichkeitsmaße hat sich in Experimenten als besser erwiesen (siehe Abschnitt 6.1.4).

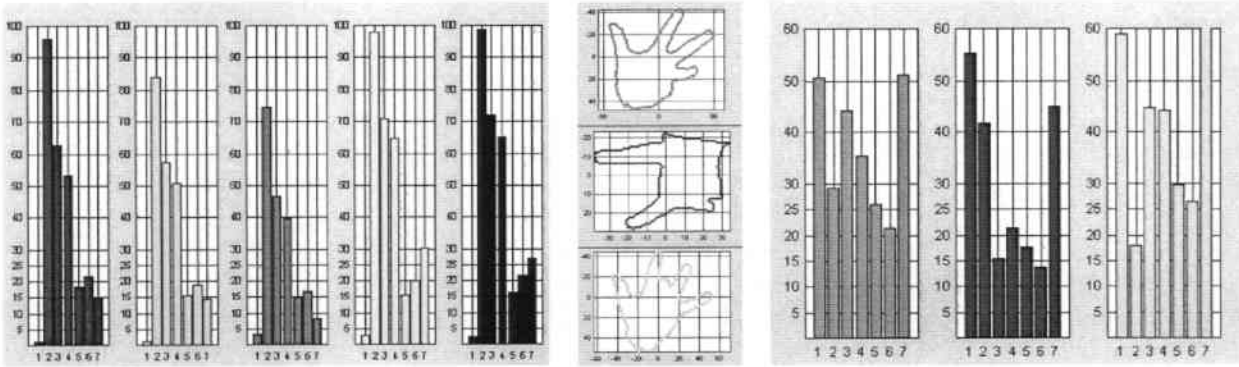


Abbildung 5.27: Distanz zwischen den Deskriptoren von fünf vorgeführten Gesten zu Referenzdeskriptoren (links), Gesten ohne Referenzmodell (Mitte) und Vergleich dieser Gesten mit den Modellen (rechts)

Zur Unterstützung der Robustheit gegen Rauschen bei der Segmentierung vor allem bei bewegten Bildern lässt sich der Umstand des kurzen Verharrens der Hand bei der Ausführung statischer Gesten ausnutzen. Die Klassifikation wird dementsprechend nur dann aktiviert, wenn die Hand für einen Moment an einer Position verharrt. Der gesamte Ablauf der bildbasierten Erkennung statischer Handgesten ist in Algorithmus 5.8 zusammengefasst.

### 5.4.3 Erkennung dynamischer Gesten in der Ausführungsumgebung

Nachfolgend werden Methoden vorgestellt, mit Hilfe derer auch die Bewegungsbahn der Hand als Geste interpretiert werden kann. Die Klassifikation selbst erfolgt auf Basis von Hidden-Markov-Modellen<sup>23</sup>. Ein wesentlicher Aspekt hierbei ist es, die Zuordnung bedeutungsloser spontaner Trajektorien zu einem Referenzmodell zu verhindern: ein fehlerhaft ausgelöster Befehl an den Roboter beeinträchtigt die Kommunikation wesentlich schwerwiegender als höhere Anforderungen an die Qualität der Vorführung. Hierzu wird der Klassifikator um einen adaptiv gewonnenen Schwellwert erweitert.

Wie auch die Erkennung statischer Gesten basiert die Erkennung dynamischer Gesten auf hautfarbsegmentierten Bildern. Das Verfahren läuft ebenfalls in mehreren Stufen ab: der Vorverarbeitung, der Merkmalsextraktion und der eigentlichen Klassifikation:

<sup>23</sup>Eine Einführung in Hidden-Markov-Modelle und den Vorwärts-, Viterbi- und Baum-Welch-Algorithmus als grundlegende Methoden zum Modellaufbau und zur Klassifikation findet sich beispielsweise in [Rabiner 89], Implementierungsbeispiele unter [Rao 99]. Das vorgestellte Verfahren selbst wurde in [Ehrenmann 01a] veröffentlicht.

**Eingabe:** Segmentiertes Kamerabild  $I$  mit ausgezeichneter Handregion  $H_I$  und Deskriptoren der Referenzgesten  $D_{1,\dots,n}^M$ , Entscheidungsschwellwert.

**Ausgabe:** Klassifizierter statischer Gestentyp.

```

1:  $K \leftarrow \text{Konturpunkte}(H_I)$ ,  $K^s \leftarrow \emptyset$ ,  $\delta \leftarrow |\{K\}|/32$ 
2: for all  $i \in \{1, \dots, 32\}$  do
3:    $K^s \leftarrow K^s + \{K_{i,\delta}\}$ 
4: end for
                                        /*Abtastung*/
5:  $cg \leftarrow \text{Schwerpunkt}(K^s)$ 
6: for all  $i \in \{1, \dots, 32\}$  do
7:    $\text{Koordinate}(K_i^s) \leftarrow \text{Koordinate}(K_i^s - cg)$ 
8: end for
                                        /*Translationsnormierung*/
9:  $max \leftarrow \max_i |\text{Koordinate}(K_i^s) - cg|$ 
10: for all  $i \in \{1, \dots, 32\}$  do
11:    $\text{Koordinate}(K_i^s) \leftarrow \text{Koordinate}(K_i^s)/max$ 
12: end for
                                        /*Skalierungsnormierung*/
13:  $D \leftarrow \text{FFT}(K^s)$ 
14: for all  $i \in \{1, \dots, 32\}$  do
15:    $\text{Koordinate}(D_i^s) \leftarrow |\text{Koordinate}(D_i^s)|$ 
16: end for
                                        /*FFT und Rotationsnormierung*/
17: for all  $i \in \{1, \dots, n\}$  do
18:    $\Delta_i^M \leftarrow \sum (D_i^M - D)^2$ 
19: end for
                                        /*Suche nach Abstandsminimum*/
20: if  $\min_i \Delta_i^M < \text{Entscheidungsschwellwert}$  then
21:   Gestentyp  $\leftarrow \text{Index}(\min_i \Delta_i^M)$ 
22:   Gebe Gestentyp zurück
23: else
24:   Gebe  $\emptyset$  zurück
25: end if
                                        /*Entscheidung*/

```

**Algorithmus 5.8:** Bildbasierte Klassifikation statischer Gesten

- Für die Klassifikation von Bewegungsmustern anhand der Verfahrbahn reicht es aus, lediglich den Schwerpunkt der Hand zu betrachten. Dieser kann einfach aus Algorithmus 5.5 gewonnen werden.
- Die bei Bewegungen entstehende Koordinatenfolge des Schwerpunkts wird geglättet und in ein eindimensionales Alphabet übersetzt. Die dabei entstehenden Wörter können mit Hilfe von Hidden-Markov-Modellen klassifiziert werden.
- Die Klassifikation wird durch Vorlage vor mehrere Referenzgesten erreicht. Jede ist



als Hidden-Markov-Modell formuliert. Mit diesem lässt sich überprüfen, mit welcher Wahrscheinlichkeit die beobachtete Trajektorie zu dem gegebenen Modell passt.

Der Gesamtprozess sieht damit aus wie in Abbildung 5.28 dargestellt. Auch hier ist das Vorgehen im dritten und vierten Schritt interessant und wird daher näher betrachtet.

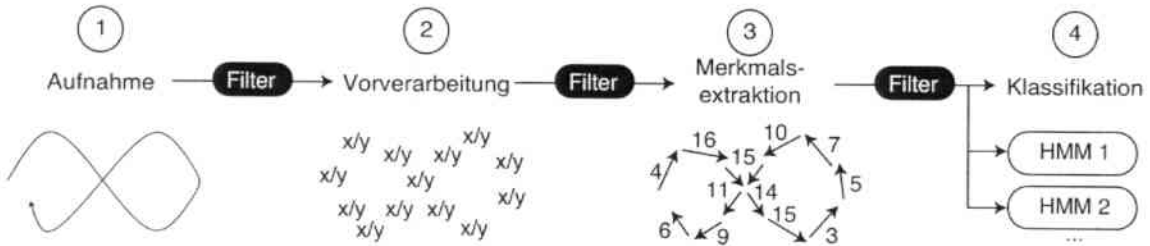


Abbildung 5.28: Der Gesten-Erkennungsprozess mit seinen Einzelschritten im Überblick

### Merkmalsextraktion

Die Glättung und Datenreduktion wird in einem mehrstufigen Filter erreicht. Die ermittelte Folge des Schwerpunkts der Benutzerhand wird zunächst an einen Nachbarschaftsfilter weitergereicht. Dieser verwirft naheliegende, aufeinanderfolgende Positionen, um die Klassifikation zu beschleunigen.

Da als Eingabe für die Hidden-Markov-Modelle keine zweidimensionalen Koordinaten dienen können, werden die Richtungsvektoren der Segmente zunächst auf ein 16elementiges Eingabealphabet  $V = \{v_1, \dots, v_{16}\}$  abgebildet (siehe Abb. 5.29).

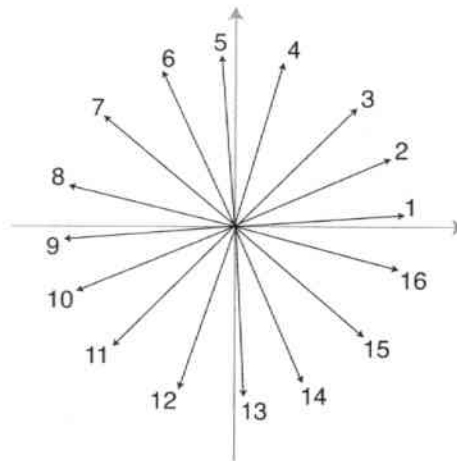


Abbildung 5.29: Das zur Vektorquantisierung verwendete Codebuch mit 16 Wörtern.

$V$  dient dazu, Beobachtungsfolgen  $O = O_1, O_2, \dots, O_T$  mit  $O_i \in V$  zu formulieren, welche klassifiziert werden können. Die Anzahl der prototypischen Beobachtungen  $|V|$  muss einerseits mächtig genug sein, um die vollzogenen Bewegungen zu repräsentieren, andererseits klein genug, um den Rechenaufwand zu beschränken. Die  $v_i$  können hier als Indizes

zu zugehörigen Richtungsvektoren interpretiert werden, welche ein spezielles Segment der vorgeführten Trajektorie repräsentieren. Gewonnen werden sie über den Steigungswinkel  $\alpha$ , der sich aus zwei sukzessiven Handschwerpunkten  $(x_{t-1}, y_{t-1})$ ,  $(x_t, y_t)$  ergibt als:

$$\alpha = \arctan \frac{y_t - y_{t-1}}{x_t - x_{t-1}}, \quad x_{t-1} < x_t \quad (5.69)$$

In Tabelle 5.4.3 wird dann der entsprechende Index  $v_i$  nachgeschlagen.

Winkelbereich	$v_i$	Winkelbereich	$v_i$
0, 0° — 22, 5°	1	180, 0° — 202, 5°	9
22, 5° — 45, 0°	2	202, 5° — 225, 0°	10
45, 0° — 77, 5°	3	225, 0° — 247, 5°	11
77, 5° — 90, 0°	4	247, 5° — 270, 0°	12
90, 0° — 112, 5°	5	270, 0° — 292, 5°	13
112, 5° — 135, 0°	5	292, 5° — 315, 0°	14
135, 0° — 157, 5°	7	315, 0° — 337, 5°	15
157, 5° — 180, 0°	8	337, 5° — 360, 0°	16

Tabelle 5.4: Zuordnung von Bewegungsvektoren zu Richtungsindizes

Die hieraus entstehende Folge  $O$  von Richtungsindizes wird durch einen Identitätsfilter weiter reduziert. Da jeder Richtungsindex aus einem Richtungsvektor entsteht und dessen Orientierung repräsentiert, reicht ein einziger Index aus, um eine Bewegungsrichtung zu verdeutlichen. Mehrere aufeinander folgende gleiche Indizes sind also redundant und können auf einen einzigen abgebildet werden.

Zur besseren Trennung von Anfang und Ende einer Geste ist vor beide Filter ein Start/Stop-Erkennen geschaltet. Er überprüft, ob die Hand kurz an einer Stelle verharrt oder nicht<sup>24</sup>. Dieses Ereignis dient als Auslöser für die Erkennung und legt Start- und Endpunkt der Trajektorie fest. Eine kontinuierliche Klassifikation wie in [Lee 99, Rigoll 97] wurde wegen höherer Verarbeitungsgeschwindigkeit und Stabilität verworfen, denn mit diesem Erkennungs-Filter können störende Punkthäufungen am Anfangs- und Endzeitpunkt der Bewegung entfernt werden. Im Umgang mit dem System stellt die Forderung nach dem kurzen Verharren der Hand keine nennenswerte Einschränkung dar.

Durch die vorgeschaltete Filterverarbeitung ergibt sich zudem der Vorteil einer weitgehenden Unabhängigkeit von der verwendeten Hardware. Zwar ist eine hohe Bilderfassungsrate we-

<sup>24</sup>Als „kurzes Verharren“ werden Bewegungen bezeichnet, die sich im Verlauf einer halben Sekunde auf einen  $\text{cm}^3$  beschränken.

sentlich bei der Handverfolgung. Andererseits enthält diese Vielzahl an Informationen einen hohen redundanten Anteil, der vor dem Erkennungsprozess herausgefiltert werden kann.

### Klassifikation

Erst nach der Übersetzung in eine eindimensionale Beobachtungsfolge können die Trajektorien den Hidden-Markov-Modellen zugeführt werden. Diese sind als endliche Automaten über den Zuständen  $S = \{s_1, \dots, s_N\}$  formuliert. Jeder Referenzgeste entspricht ein solches Modell  $\lambda$  mit der Struktur  $\lambda = (A, B, \pi)$ , wobei  $A$  eine reelle Matrix der Zustandsübergangswahrscheinlichkeiten ist.  $B$  stellt die Wahrscheinlichkeitsverteilung der Beobachtungssymbole mit  $b_j(k)$  als Wahrscheinlichkeit für das Auftreten des Symbols  $v_k$  im Zustand  $s_j$  dar und  $\pi$  ist die anfängliche Zustandsverteilung, wobei  $\pi_i$  die Wahrscheinlichkeit angibt, mit der Zustand  $s_i$  der Initialzustand ist. Die Klassifikation der Beobachtung  $O = O_1, O_2, \dots, O_T$  mit den gespeicherten Modellen  $\lambda_1, \dots, \lambda_K$  hängt von mehreren Faktoren ab. Wie im Fall der statischen Gesten sollen die für die Klassifikation relevanten Kriterien vorgestellt und diskutiert werden:

**Vergleichskriterium:** Die Beobachtung  $O$  muß zunächst mit allen Referenzmodellen  $\lambda$  verglichen werden, um die Wahrscheinlichkeit  $P(O|\lambda)$  zu erhalten, mit der das entsprechende Modell mit der vorgeführten Bewegungssequenz übereinstimmt. Der hierzu verwendete Algorithmus ist der Viterbi-Algorithmus zum Finden einer optimalen Zustandsfolge  $Q = Q_1, Q_2, \dots, Q_T$ . Dabei bedeutet optimal, dass die Zustandsfolge  $Q$  die größte Wahrscheinlichkeit für das Emittieren von  $O$  aufweist. Der Berechnungsaufwand für das Testen einer Beobachtung  $O$  steigt quadratisch mit der Zustandsanzahl in den Modellen  $\lambda$ . Um den mit den einzelnen Tests verbundenen Rechenaufwand zu begrenzen, wird eine Klassifikation der Referenzgesten eingeführt, die die Anzahl der Tests deutlich reduziert. Hierbei werden die Gesten und mit ihnen ihre Referenzmodelle gemäß ihrer Komplexität hierarchisch angeordnet. Es sind drei Komplexitätsklassen vorgesehen, die sich einfach ergänzen lassen (Abbildung 5.30).

Anhand der eingehenden Folge von Richtungsindizes wird nun entschieden, mit welchen Gesten der Test auf die Erkennung durchgeführt werden soll. Dabei wird eine Klasse gewählt und alle darin enthaltenen Referenzmodelle getestet. Die Länge der Richtungsindexfolge ist Indiz für eine mögliche Geste. Kurze Folgen brauchen nicht auf komplexe Modelle getestet werden, da eine komplexe Geste sich aus einer hohen Anzahl von Richtungsindizes zusammensetzt. Die Länge der erfassten Richtungsindexfolge kann als Kriterium für die Komplexität herangezogen werden, da der Erkennung ein Filter vorgeschaltet ist, der vor Verfälschungen schützt. Ein Innehalten bei der Handbewegung beziehungsweise eine sehr langsame Handbewegung produziert auch bei einer einfachen Geste eine hohe Anzahl von Richtungsindizes.

Bei der Verwendung einer solchen Klassifikation der Gesten gemäß ihrer Komplexität muss allerdings darauf geachtet werden, dass sich eben diese Komplexität bei verschiedenen Klassen auch deutlich unterscheidet. Sollten ähnlich komplexe Gesten in unterschiedlichen Klassen liegen, so führt die initiale Entscheidung für eine bestimmte Klasse möglicherweise zur Auswahl der falschen und verhindert die Erkennung der Geste.

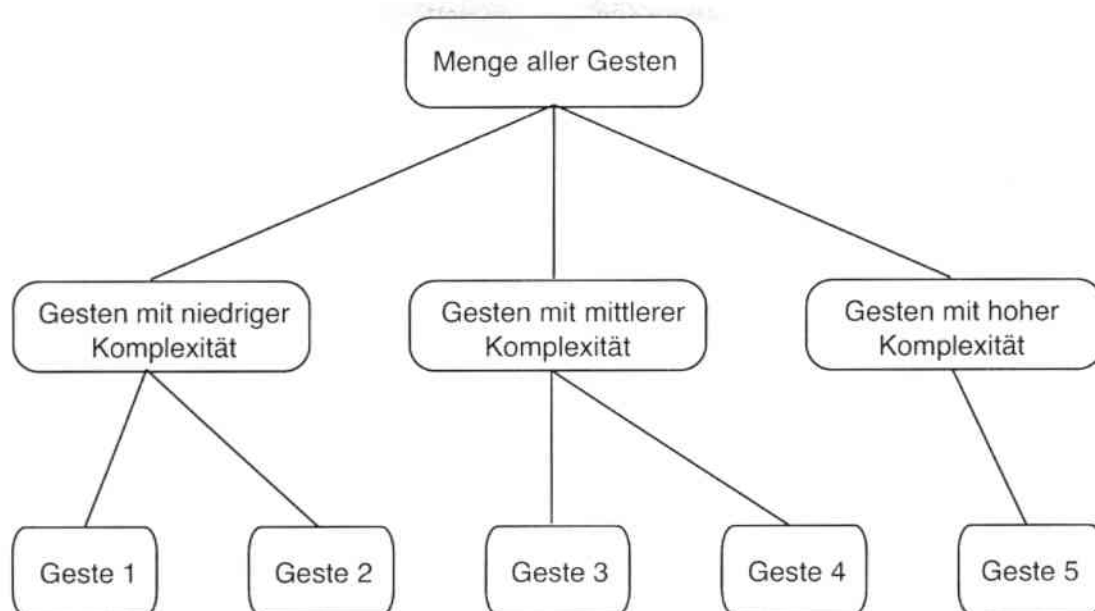


Abbildung 5.30: Klassifizierung der Referenzgesten aufgrund ihrer Komplexität

**Entscheidungskriterium:** Im Zusammenhang mit der Modellierung der Referenzmodelle ergibt sich die Frage nach der Darstellung bedeutungsloser spontaner Bewegungen, denen kein Sinn zugeordnet werden darf. Dabei können unterschiedliche Ansätze verfolgt werden. Während im Bereich der Sprach- und Handschrifterkennung ein Füllmodell definiert wird [Wilcox 92], dessen Aufgabe es ist, Laute ohne Bedeutung zu erkennen, geht [Lee 99] einen anderen Weg. Dies ist notwendig, da das Füllmodell als Ergänzung zu den ausgewählten Symbolen mit einer endlichen Anzahl von bedeutungslosen Beispiellauten trainiert werden muss. Das wäre im Bereich der Gestenerkennung zu aufwändig.

Da Hidden-Markov-Modelle selbst bei einer Geste, die einem Modell nicht entspricht, eine Wahrscheinlichkeit größer Null emittieren, muss eine Entscheidung über Erfolg oder Nichterfolg gefällt werden. Mit einem einfachen, festen Schwellwert ist es hierbei nicht getan, da die Erkennungswahrscheinlichkeiten um Größenordnungen schwanken. Deshalb wird ein sogenanntes Schwellwertmodell [Lee 99] aufgebaut. Dieses soll dem individuellen Charakter der einzelnen Modelle gerecht werden und adaptiven Charakter haben, um nicht zu große Anforderungen an die Ausführung zu stellen. Das Schwellwertmodell setzt sich aus Zustandskopien aller Referenzmodelle der trainierten Gesten des Systems zusammen und ist ein weiteres Hidden-Markov-Modell. Dieses erkennt dann alle Gesten, die sich aus Teilgesten der in den Referenzmustern modellierten zusammensetzen. Die vom Schwellwertmodell gelieferte Wahrscheinlichkeit dient als Schwellwert für die Erkennung mit einem anderen Modell und hat dabei adaptiven Charakter, denn der emittierte Wert ist umso höher und damit umso besser, je mehr bekannte Teilgesten in der untersuchten Geste enthalten sind. Hier profitiert das Schwellwertmodell von der sogenannten *internen Segmentierungseigenschaft* der

Hidden-Markov-Modelle, die besagt, dass die Zustände und die Zustandsübergänge eines trainierten Modells Teilgesten eben dieses Modells repräsentieren.

Zusammenfassend lässt sich die Nutzung des Schwellwertes wie folgt beschreiben: Das Schwellwertmodell ermittelt auf Basis der eingehenden Daten einen adaptiven Schwellwert. Anschließend werden dieselben Daten jedem Referenzmodell zur Entscheidung vorgelegt. Eine Erfolgsmeldung für eine erkannte Geste wird nur gegeben, wenn die Wahrscheinlichkeit  $P_i = P(\text{Sequenz} | i\text{-tes Referenzmodell})$  für eines der Referenzmodelle größer als der Schwellwert wird.

Eine mögliche Architektur des Schwellwertmodells lässt sich in Gestalt eines *ergodischen* Modells finden, d.h.  $a_{ij} \geq 0 \quad \forall i, j = 1, \dots, N$ . Es entsteht dadurch, dass alle Zustände der Referenzmodelle übernommen und miteinander verbunden werden. So kann jeder Zustand von jedem anderen in einem einzigen Übergang erreicht und die Erkennung der Teilgesten realisiert werden. Abbildung 5.31 skizziert eine vereinfachte Struktur eines solchen Modells. In dem so entstandenen Modell werden die Beobach-

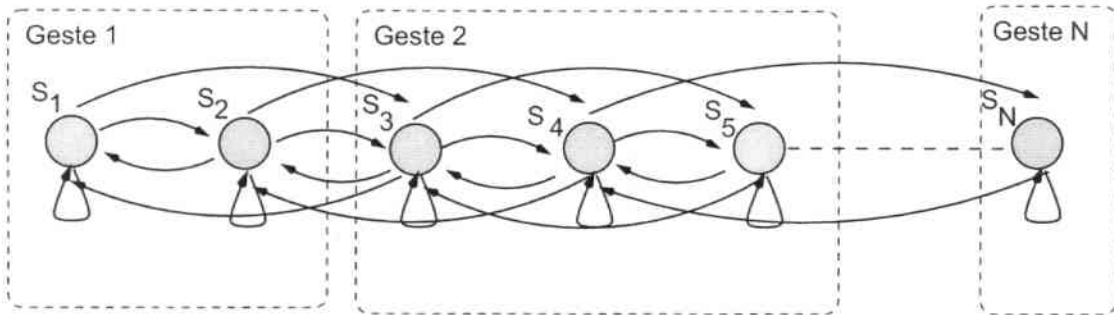


Abbildung 5.31: Struktur des Schwellwertmodells als ergodisches Hidden-Markov-Modell

tungswahrscheinlichkeiten für die Ausgabesymbole übernommen. Das Gleiche gilt für die Wahrscheinlichkeitswerte der Zustandsübergänge. Für alle anderen Zustandsübergangswahrscheinlichkeiten gilt:

$$a_{ij} = \frac{1 - a_{ii}}{N - 1}, \quad \forall i, j, i \neq j. \quad (5.70)$$

$N$  bezeichne hierbei die Anzahl der Zustände,  $a_{ij}$  die Wahrscheinlichkeit eines Übergangs von Zustand  $s_i$  nach  $s_j$ . Das Beibehalten der Wahrscheinlichkeiten für Selbstübergänge und Beobachtungen korrespondiert mit der internen Segmentierungseigenschaft der Hidden-Markov-Modelle. Es stellt sicher, dass das Schwellwertmodell jede Geste, die aus beliebigen Teilen der Referenzgesten in beliebiger Reihenfolge zusammengesetzt ist, erkennt. Durch die verringerte Zustandsübergangswahrscheinlichkeit 5.70 ist jedoch garantiert, dass die Wahrscheinlichkeit des Referenzmodells dann größer ist als die des Schwellwertmodells. Die Anfangswahrscheinlichkeit  $\pi$  wird auf alle Zustände gleichverteilt:

$$\pi_i = \frac{1}{N}, \quad i = 1, \dots, N. \quad (5.71)$$

Eine Vergabe von Werten für  $\pi$  ist notwendig, da der Vorwärts-Algorithmus zur Ermittlung des Schwellwerts diese Werte in die Berechnung mit einbezieht. Um Gewichtungen durch unterschiedlich große  $\pi_i$  und daraus resultierende Verfälschungen bei der Berechnung des Schwellwertes zu vermeiden, existiert kein hervorgehobener Startzustand.

Bei der Anwendung der hierarchischen Gestenklassifikation (Seite 118) muss für jede Hierarchiekategorie ein entsprechendes Schwellwertmodell aufgebaut werden. Dies reduziert den Rechenaufwand nochmals, da nur gegen das zur Klasse gehörende Schwellwertmodell geprüft werden muss und dieses wegen des quadratischen Wachstums erheblich kleiner ausfällt.

**Modellaufbau:** Für die Gestenerkennung werden *Links-Rechts*-Modelle eingesetzt mit einer Sprungbegrenzung von  $\Delta = 2$ , d. h. es werden folgende Bedingungen an  $\lambda$  gestellt:

$$a_{ij} = 0, \quad j > i + \Delta \quad (5.72)$$

Das heißt zunächst, dass das Modell von links nach rechts durchschritten wird (der Zustandsindex erhöht sich mit der Zeit oder bleibt gleich; Rückschritte sind verboten). Mit dieser Eigenschaft eignen sich *Links-Rechts*-Modelle zur Modellierung von Signalen, deren Eigenschaften sich über die Zeit ändern. Die Anzahl der überspringbaren Zustände wird auf 2 begrenzt, es können also keine Übergänge über mehr als zwei  $s$  stattfinden.

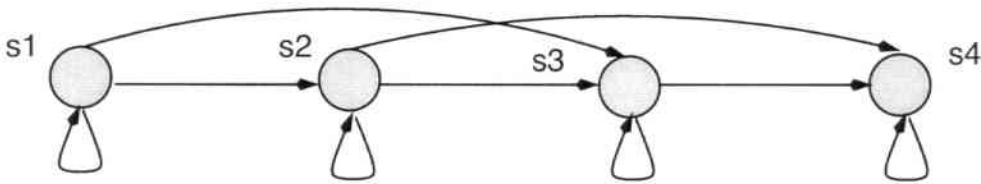


Abbildung 5.32: *Links-Rechts*-Hidden-Markov-Modell mit vier Zuständen und einer Sprungbegrenzung von  $\Delta = 2$

Eine entsprechende Übergangsmatrix hätte bei vier Zuständen beispielsweise die Gestalt aus Abbildung 5.32 mit der folgenden Zustandsübergangsmatrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \quad (5.73)$$

Weiterhin gilt, dass Zustand  $s_1$  der Initialzustand, Zustand  $s_N$  der Endzustand ist:

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (5.74)$$



Außerdem gilt bei *Links-Rechts*-Modellen immer, dass der Übergang vom Endzustand lediglich in sich selbst stattfinden kann:

$$a_{NN} = 1, \quad (5.75)$$

$$a_{Ni} = 0, \quad i < N \quad (5.76)$$

Das Training der Hidden-Markov-Modelle erfolgt mit dem Baum-Welch-Algorithmus als etablierter Methode. Er passt die Modellparameter  $(A, B, \pi)$  so an, dass die Wahrscheinlichkeit für eine Beobachtungsfolge bei gegebenem Modell maximal wird. Dies ist gleichbedeutend mit einem Training des Modells, so dass nach erfolgtem Training mit Hilfe des Viterbi-Algorithmus die wahrscheinlichste Zustandsfolge ermittelt werden kann.

Der gesamte Ablauf der Erkennung dynamischer Handgesten ist in Algorithmus 5.9 zusammengefasst. Hier findet sich zuerst die Applikation des Nachbarschaftsfilters (Zeilen 1 bis 5), dann die Übersetzung in das Eingabealphabet  $V$  (Zeile 6) und die Eliminierung gleicher sukzessiver Symbole (Zeilen 7 bis 11). Nach der Vorlage vor die gespeicherten Gestenmodelle (Zeilen 12 bis 15) erfolgt der Test gegen das Schwellwertmodell  $\lambda^\theta$  (Zeilen 16 und 17).

#### 5.4.4 Operator zur Gestendetektion

Der Operator für die Klassifikation statischer Gesten auf Basis von Messungen des Datenhandschuhs kann dem Grifferkenner in der Vorführungsumgebung sehr ähnlich konstruiert werden. Er ist in Algorithmus 5.10 aufgeführt. Zunächst wird hier anhand der Finger- und Handbewegungsgeschwindigkeit  $v_G$  geprüft, ob der Kontext für die neuronale Klassifikation gegeben ist (Zeile 1). Danach entscheidet das trainierte Netz über die Klassenzugehörigkeit (Zeile 2 bis 7).

Anders verhält es sich mit der Ausführungsumgebung: ein kurzes Verharren der Hand kann für die Vorführung einer statischen Geste ebenso wie für den Anfang einer dynamischen stehen. Die Lösung dieses Problems ist in vereinfachter Form in Algorithmus 5.11 aufgeführt. In der Implementierung ist die Zeitmessung für das Verharren der Hand bei der Gestenausführung frei wählbar. Bei einem kurzen Verharren der Hand muss sowohl die Klassifikation der Handkonfiguration erfolgen (Zeilen 4 und 5) wie auch der Start der Trajektorienaufzeichnung für die Qualifizierung einer dynamischen Geste (Zeilen 2 und 3). Die Unterscheidung des jeweiligen Zustands wird ab dann mit Hilfe der Variable *state* getroffen (Zeilen 1, 2, 6, 8, 12 und 13). Erst bei erneutem Verharren der Hand wird die aufgezeichnete Folge von Hand Schwerpunkten  $O$  klassifiziert (Zeilen 9 bis 11). Die Funktion „Registriere“ trägt dabei im Falle, dass die Klassifikatoren in  $g$  die leere Menge  $\emptyset$  zurückgeben haben, kein Ereignis in das Weltmodell ein.

## 5.5 Registrierung beobachteter Ereignisse

Handlungsrelevante Beobachtungen wie manipulierbare Objekte, Griffe, Gesten oder Handtrajektorien müssen zur späteren Interpretation der Vorführung im Weltmodell

**Eingabe:** Folge  $O$  von Schwerpunkten der Benutzerhand im Kamerabild,  
Nachbarschaftsschwellwert $_d$ , Deskriptoren der Referenzgesten  $\lambda_{1,\dots,n}^M$ ,  
Schwellwertmodell  $\lambda^\theta$ .

**Ausgabe:** Klassifizierter dynamischer Gestentyp.

```

1: for all  $[o_i, o_{i+1}] \in O$  do
2:   if  $|o_i - o_{i+1}| < \text{Schwellwert}_d$  then
3:      $O \leftarrow O \setminus [o_i]$ 
4:   end if
5: end for
/*Nachbarschaftsfilter*/

6:  $O^V \leftarrow \text{Vektorsymbol}(O)$ 
/*Merkmalsextraktion*/

7: for all  $[o_i, o_{i+1}] \in O^V$  do
8:   if  $o_i == o_{i+1}$  then
9:      $O^V \leftarrow O^V \setminus [o_i]$ 
10:  end if
11: end for
/*Gleichheitsfilter*/

12: for all  $\lambda^M$  do
13:    $p_i \leftarrow \text{Viterbi}(\lambda_i^M, O^V)$ 
14: end for
15:  $p \leftarrow \max p_i, \quad i \leftarrow \text{maxindex } p_i$ 
/*Klassifikation*/

16:  $p_\theta \leftarrow \text{Viterbi}(\lambda^\theta, O^V)$ 
17: if  $p \geq p_\theta$  then
18:   Gebe  $i$  zurück
19: else
20:   Gebe  $\emptyset$  zurück
21: end if
/*Entscheidungskriterium*/

```

**Algorithmus 5.9:** Bildbasierte Klassifikation dynamischer Gesten

registriert werden. Von der Funktion zum Registrieren ist in den Algorithmen 5.1, 5.6, 5.7 und 5.11, die die kognitiven Operatoren realisieren, bereits Gebrauch gemacht worden.

Abbildung 5.33 stellt das Zusammenwirken der Operatoren und dem Weltmodell anschaulich vor. Da die Methoden zur Detektion auf einem verteilten System ablaufen, ist bei der Registrierung der Ereignisse auf eine einheitliche Zeitmessung zu achten. Dies ist besonders bei der Fusionierung von Daten zu berücksichtigen. Bei der Bewegungsverfolgung in der Vorführungsumgebung werden dazu die erhaltenen Werte auf dem empfangenden Rechner interpoliert und auf derselben Zeitbasis ein weiteres Mal abgetastet.

Vor der Verarbeitung werden die Messdaten auf ihre Plausibilität hin überprüft. Große Sprünge werden beispielsweise in der Bewegungsverfolgung als Fehlmessungen interpretiert

**Eingabe:** Gemessene Handkonfiguration  $\vec{w}_t$  und  $\vec{w}_{t-1}$  des Datenhandschuhs sowie Position  $\vec{p}_t$  und  $\vec{p}_{t-1}$  und Orientierung  $\vec{\phi}_t$  und  $\vec{\phi}_{t-1}$  der Benutzerhand, Schwellwerte  $\theta_H$  und  $\theta_G$  für die Erfassung der Handbewegung und als Mindestwert für die neuronale Aktivierung, trainiertes neuronales Netz  $N$ .

**Ausgabe:** Registrierung des Gestentyps mit Ort und Zeitpunkt.

```

                                                                    /*Kontextfeststellung*/
1: if  $v_G(\vec{w}_t, \vec{w}_{t-1}, \vec{p}_t, \vec{p}_{t-1}, \vec{\phi}_t, \vec{\phi}_{t-1}) < \theta_H$  then
2:   BelegeEingabeschicht( $N, \vec{w}_t$ )
                                                                    /*Klassifikation*/
3:   Propagiere( $N$ )
                                                                    /*Entscheidungskriterium*/
4:   if  $\exists$  Ausgabeneuron  $n \in N$  mit Aktivierung  $a_n > \theta_G$  then
5:     Registriere( $\text{Gestentyp}(n), t, \vec{p}_t, \vec{\phi}_t$ )
6:   end if
7: end if

```

**Algorithmus 5.10:** Elementarer kognitiver Operator zur Gestendetektion in der Vorführungsumgebung

**Eingabe:** Segmentiertes Kamerabild  $I$  mit ausgezeichneter Handregion  $H_I$  und deren Position  $\vec{p}_t$ , Orientierung  $\vec{\phi}_t$ , Geschwindigkeit  $v_H$  sowie Schwerpunkt  $c_H$ , der Start/Stop-Schwellwert $_v$ , der Nachbarschaftsschwellwert $_d$ .

**Ausgabe:** Registrierung des Gestentyps mit Ort und Zeitpunkt.

```

                                                                    /*Start/Stopfilter*/
1: if  $(v_H < \text{Schwellwert}_v) \text{AND} (state = \text{Start})$  then
2:    $state \leftarrow \text{Aufnahme}$ 
3:    $O \leftarrow \emptyset$ 
4:    $g \leftarrow \text{KlassifiziereStatischeGeste}(H_I)$ 
5:   if  $g \neq \emptyset$  then
6:     Registriere( $g, t, \vec{p}_t, \vec{\phi}_t$ )
7:   end if
8: else if  $(v_H \geq \text{Schwellwert}_v) \text{AND} (state = \text{Aufnahme})$  then
9:    $O \leftarrow O + [c_H]$ 
10: else if  $(v_H < \text{Schwellwert}_v) \text{AND} (state = \text{Aufnahme})$  then
11:    $state \leftarrow \text{Stop}$ 
12:    $g \leftarrow \text{KlassifiziereDynamischeGeste}(O)$ 
13:   if  $g \neq \emptyset$  then
14:     Registriere( $g, t$ )
15:   end if
16: else if  $(v_H < \text{Schwellwert}_v) \text{AND} (state = \text{Stop})$  then
17:    $state \leftarrow \text{Start}$ 
18: end if

```

**Algorithmus 5.11:** Elementarer kognitiver Operator zur Gestendetektion in der Ausführungsumgebung

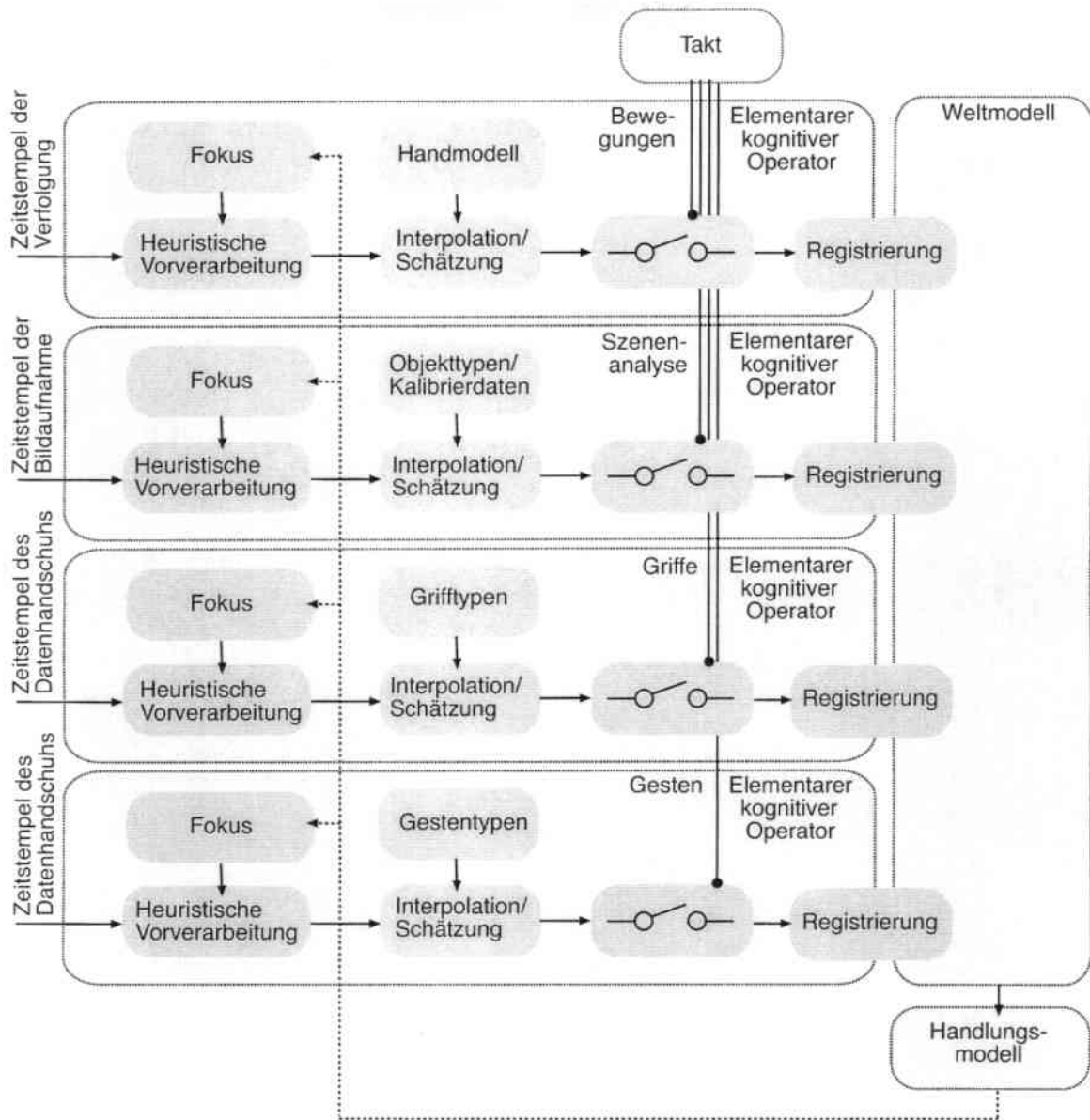


Abbildung 5.33: Registrierung beobachteter Ereignisse im Weltmodell

und gefiltert (im Bild als „Heuristische Vorverarbeitung“ markiert, vergleiche auch Abbildung 5.13). Jeder Operator nutzt außerdem zusätzlich entsprechendes Modellwissen (beispielsweise Kalibrierdaten, Objektmodelle, Gesten- und Griffotypen). Aufgrund des im Weltmodell gespeicherten Kontextes und der im Handlungsmodell gespeicherten Information kann dann der Aufmerksamkeitsfokus des Systems auf Orte zentriert werden, in denen künftige Ereignisse beobachtbar sein können. Nach der Registrierung eines Loslassvorgangs beispielsweise wird die Szenenanalyse mit dem Fokus auf die Handposition angestoßen. Grundlage für diese Entscheidung ist das in Abschnitt 4.5 vorgestellte Handlungsmodell (siehe auch Abbildung 4.9).

## 5.6 Zusammenfassung

Das vorliegende Kapitel erläuterte den Aufbau der kognitiven Operatoren, die zur Registrierung relevanter Handlungselemente und deren Parameter im Weltmodell dienen. Sie werden zur Szenenanalyse, Verfolgung der Benutzerhand und zur Gesten- und Grifferkennung eingesetzt.

Einige der Operatoren wurden der Aufteilung in Vorführ- und Ausführumgebung entsprechend speziell konstruiert. Dies ist besonders deutlich im Fall der Bewegungsverfolgung: zur Nutzung vorgeführter Trajektorien auf einem Robotersystem ist die Genauigkeit der Handverfolgung eine wesentliche Bedingung. Deshalb werden hier die Messwerte des magnetfeldbasierten und des bildbasierten Verfolgungssystems fusioniert. Zur Interaktion wird auf den Magnetfeldsensor dagegen nicht zurückgegriffen.

# Kapitel 6

## Experimentelle Validierung

Nach der Vorstellung der Funktionen der einzelnen kognitiven Operatoren im Kapitel 5 wird im Folgenden sowohl deren jeweilige Leistung untersucht werden als auch deren Zusammenspiel im Rahmen der Beobachtung einer Handlungsfolge. In diesem Kapitel werden daher zunächst separat Anwendungen der einzelnen kognitiven Operatoren vorgestellt. Anschließend wird die Verfolgung von Handlungen mit diesen Operatoren anhand zweier Experimente mit komplexeren Vorführungen diskutiert.

### 6.1 Validierung der kognitiven Operatoren

Die zu validierenden Operatoren sind die im Kapitel 5 vorgestellten Beobachtungselemente für die Szenenanalyse, Bewegungsverfolgung und Griff- und Gestenerkennung. Sie werden in ihrer Arbeitsweise der Reihe nach untersucht.

#### 6.1.1 Szenenanalyse

Entsprechend der Vorstellung des kognitiven Operators zur Szenenanalyse in Algorithmus 5.1 erfolgt unten die Diskussion der gewählten Verfahren zur Objektdetektion und der Analyse der Genauigkeit der 3D-Rekonstruktion. Die farbhistogrammbasierte Methode in der Ausführungsumgebung und der kombinierte Ansatz in der Vorführungsumgebung werden jeweils separat behandelt. In der Vorführungsumgebung wird das objektspezifische Hintergrundwissen referenziert, um eine entsprechende Detektionsmethode aufzurufen und das Ergebnis im Anschluss zu fusionieren.

#### Objektdetektion in der Vorführungsumgebung

Die Modelle zur Objektdetektion variieren in ihrer Auflösungsgröße zwischen 60 und 200 Bildpunkten Durchmesser im Fall einer ansichtsbasierten Detektion bzw. 20 bis 50 Stützpunkten im Fall der Verwendung konturbasierter Detektionsmethoden. Sie entsprechen in Perspektive und Ausschnittwahl den Bildern, die auch im CogVis-Projekt verwendet werden [Union 01]. Ansichtsmodelle können durch Bildausschnitte gewonnen werden. Die entsprechenden Konturmodelle  $\vec{r}$  werden über eine interaktive Dialogmaske eingegeben, in die im Hintergrund eine Objektansicht eingeblendet wird (Abbildung 6.1 a). Zur



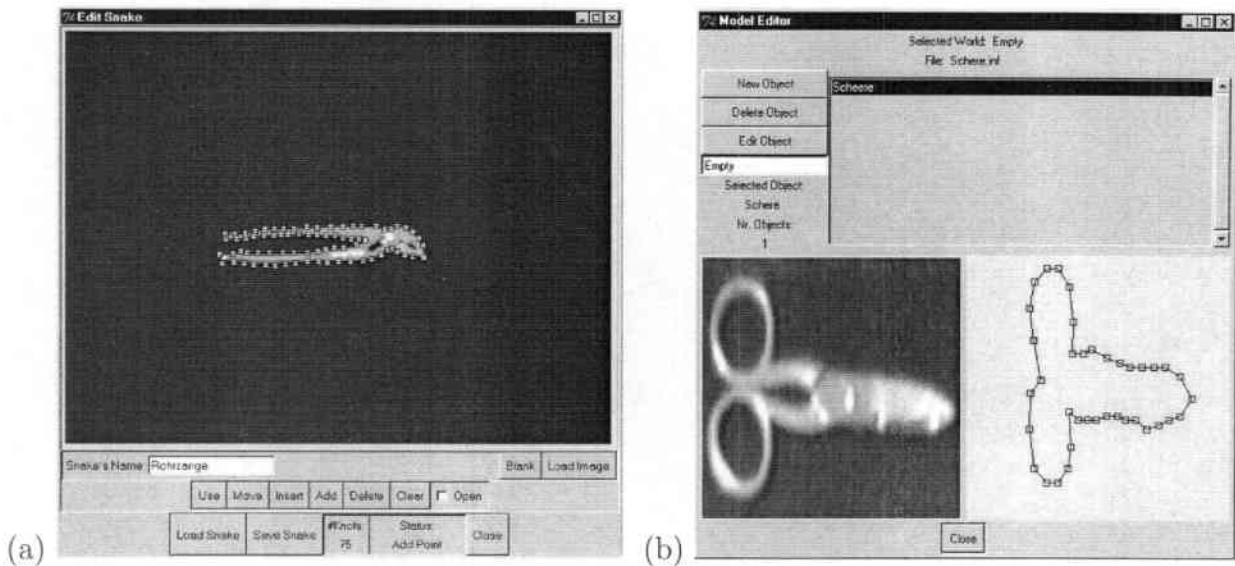


Abbildung 6.1: Testobjekte für die Szenenanalyse (a) und Modelldatenbank (b)

Objektdetektion können diese ebenso wie Ansichten des Objekts referenziert werden. In Abbildung 6.1 b sind einige charakteristische Merkmale für das Objekt „Schere“ abgebildet.

In den Experimenten wurden die Rotationsachsen des Kamerakopfes konstant gehalten und Modelle verwendet, die aus der fixierten Blickrichtung hinreichend beobachtbar waren. Es wurden 30 Bilder aufgenommen, die mehrere Objekte enthielten. Zusätzlich wurden 10 Beispiele untersucht, die partiell verdeckte Objekte enthielten. Aus Rechenzeitgründen wurden Rotationen, aber keine Skalierungen in den Algorithmen berücksichtigt. Tabelle 6.1 stellt die erzielten Klassifikationsleistungen zusammen<sup>1</sup>. Die Rubrik des Fehlers erster Art evaluiert die Rate nicht erkannter Objekte, der Fehler zweiter Art gibt die Rate vermeintlicher Detektionen von nicht vorhandenen Objekten an. Dies kann aufgrund objektähnlicher Strukturen im Hintergrund auftreten. Getestet wurde der Einsatz der einzelnen Detektionsmethoden wie auch des kombinierten Ansatzes entsprechend Algorithmus 5.1.

Als Schwellwerte für die Erkennung wurden höhere Werte bevorzugt, um den Fehler zweiter Art niedrig zu halten. Dies resultierte jedoch in höheren Ergebnissen für den Fehler erster Art. Es zeigt sich, dass der kombinierte Ansatz die Klassifikationsleistungen der Einzelerkenner wesentlich verbessert, wenn die Detektionsmethode für die Objekte entsprechend ihrer Klasseneinteilung gemäß Abschnitt 4.5.1 gewählt wird. Da die Laufzeit des Algorithmus zur Graphanpassung aufgrund des kubischen Zeitverhaltens inakzeptabel hoch ist, wird für die Detektion von Konturmodellen die Methode der Allgemeinen Hough-Transformation bevorzugt gewählt.

Die Stärke dieses Ansatzes wird vor allem bei verdeckten Objekten deutlich. Hier konnten 8 der 10 teilverdeckten Gegenstände korrekt identifiziert werden. Abbildung 6.2 oben macht die Leistung anhand zweier Beispiele deutlich. Fehlklassifikationen oder Verwechslun-

<sup>1</sup>Verwendet für die Experimente wurde ein bei 300MHz getakteter PC mit zwei Pentium2-Prozessoren

Methode	GA	AHT	SA	KA
Test	0.77	0.79	0.70	0.94
Fehler erster Art	0.23	0.21	0.30	0.06
Fehler zweiter Art	0.07	0.05	0.05	0.01
Maximaler Überdeckungsgrad	0.50	0.80	0.10	0.50
Laufzeit/Objekt in [s]	145.0	19.3	2.1	12.6

Tabelle 6.1: Erkennungsleistung der Objektdetektion in der Vorführungsumgebung (GA=Graphanpassung, AHT=Allgemeine Hough-Transformation, SA=Schablonenanpassung und KA=Kombinierter Ansatz)

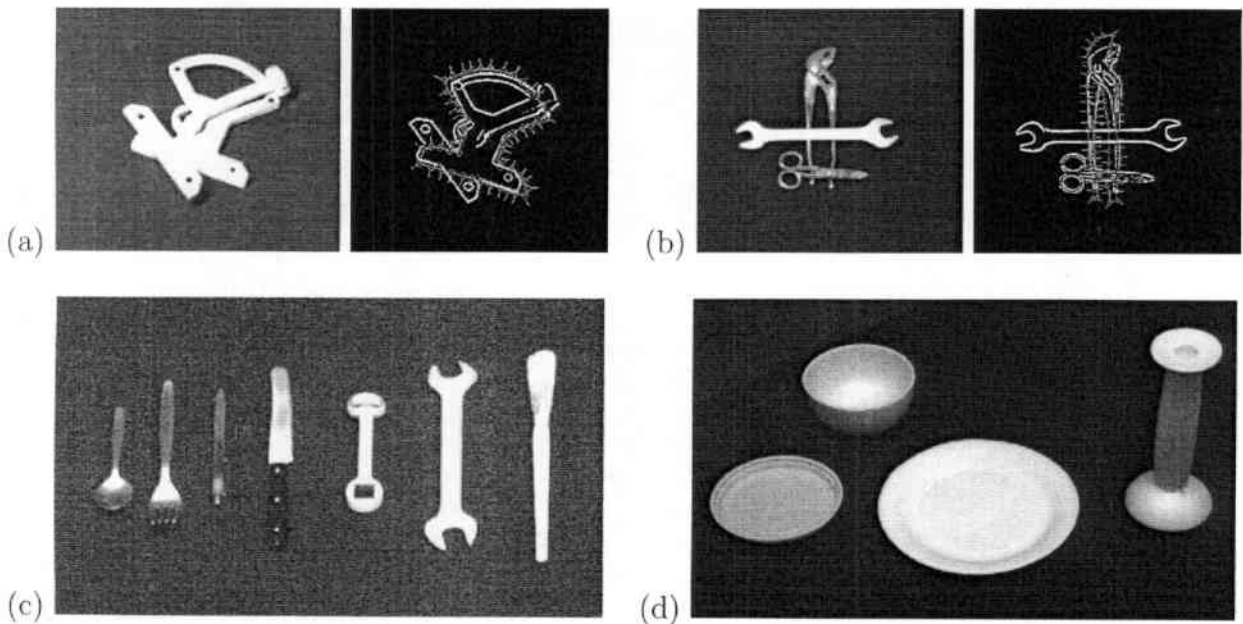


Abbildung 6.2: Ergebnisse der allgemeinen Houghtransformation bei Teilverdeckungen (oben) und Objekte mit hoher Verwechslungsrate aufgrund ähnlichen Umrisses (unten links) bzw. ähnlicher Textur (unten rechts)

gen treten beispielsweise bei kleinen Objekten auf, deren Kontur sich nur in wenigen Punkten unterscheidet. Auch Objekte ähnlicher Textur erhöhen bei ansichtsbasierten Methoden die Fehlerrate (Beispiele in Abbildung 6.2 unten).

### Objektdetektion in der Ausführungsumgebung

Objektspezifische Beschreibungen, die für die Klassifikation genutzt werden, können mit dem farbhistogrammbasierten Verfahren automatisch generiert werden. Dazu wird während der Interaktion mit dem Robotersystem ein einzelner Gegenstand auf der Manipulationsfläche präsentiert und benannt. Dieser wird segmentiert und seine charakteristischen Eigenschaf-

ten gespeichert. Abbildung 6.3 zeigt das Ausmaskieren eines Kerzenständers und dessen Histogramme für den Farbwert, die Sättigung und Helligkeit. Diese werden in der Lernphase in komprimierter Form nach Formel 5.15 zusammen mit seinem Höhe/Breite-Verhältnis gespeichert.

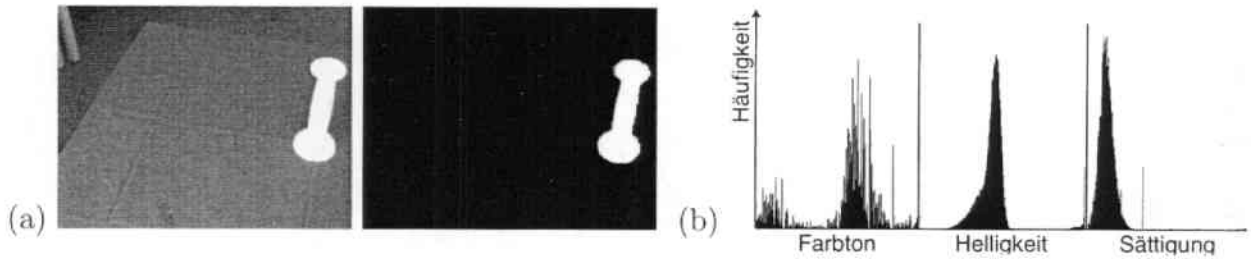


Abbildung 6.3: Lernen neuer Objektcharakteristika: Kamerabild und Maske (a), objektspezifische Histogramme (b)

Ein Schwachpunkt des Verfahrens ist die Forderung nach der Separierbarkeit der Objekte. Da die Segmentierung bei der Farbreduktion auch Nachbarpixel betrachtet, müssen in den Kamerabildern mindestens zwei Bildpunkte zwischen zwei Objekten vorhanden sein. Dies entspricht bei halber PAL-Auflösung der Kameras einem Zentimeter auf der Manipulationsfläche. Um diesen Umstand zu umgehen, können beispielsweise gestapelte Objekte als separates Objekt gelernt werden. Dies funktionierte bei einer auf eine Untertasse gestellte Tasse in den Testläufen<sup>2</sup>.

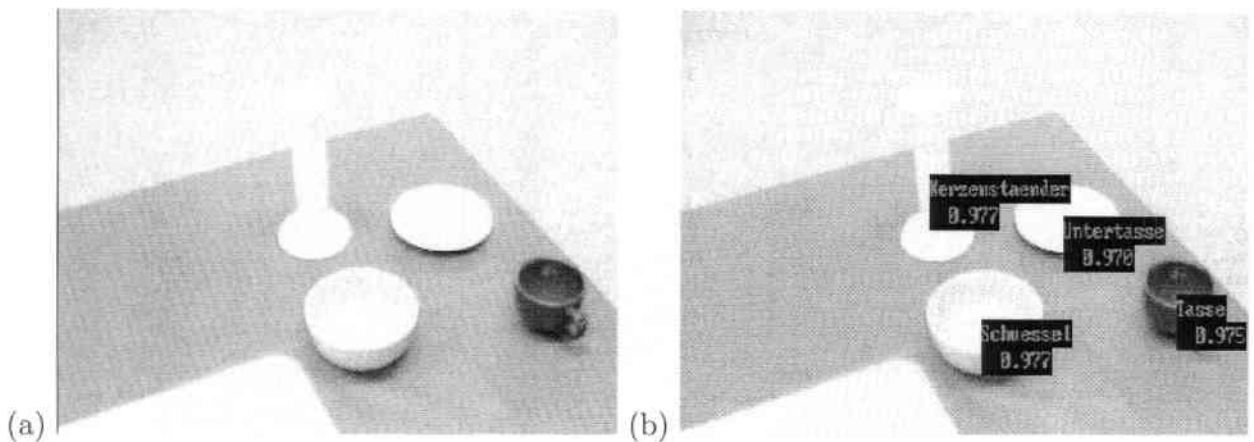


Abbildung 6.4: Objektklassifikation auf Basis von Farbhistogrammen

Die Klassifikationsleistung des Verfahrens hängt von der Anzahl der bereits gelernten Objekte abhängig. In Tabelle 6.2 sind die Klassifikationsergebnisse bei unterschiedlichen Anzahlen gelernter Objekte eingetragen. Dabei wurden 30 unterschiedlich aufgenommene Bilder verschiedener Objekte untersucht, die jeweils fünf Objekte enthielten. Abbildung 6.4 gibt einen Eindruck von der Aufnahmesituation. Eines dieser Objekte war dabei noch nicht gelernt worden, um den Fehler zweiter Art bestimmen zu können. Es zeigt sich, dass vor allem die

<sup>2</sup>Eine umfassende Darstellung der Experimente findet sich in [Lehne 01].

Anzahl der gelernten Objekte	5	8	9	12
Test	0.97	0.85	0.67	0.58
Fehler erster Art	0.03	0.15	0.33	0.42
Fehler zweiter Art	0.0	0.04	0.08	0.18

Tabelle 6.2: Erkennungsleistung der Objektdetektion in der Ausführunggebung

Anzahl der Verwechslungen und der Fehler zweiter Art bei zunehmender Zahl bekannter Objekte wächst. Bei einer Objektanzahl  $\geq 12$  wird nahezu jedes Objekt als bekannt angenommen. Dies hat seine Ursache nicht zuletzt darin, dass die gelernten Objekte möglichst unterschiedlich ausgewählt wurden und daher den Merkmalsraum gut überdecken. Eine Verkleinerung des Parameters  $\Delta$  zur Erhöhung der Trennschärfe vermindert jedoch die Robustheit gegenüber unterschiedlichen Beleuchtungssituationen. Unterschiedliche Skalierungen der Objekte in den Kamerabildern haben dagegen keinen Einfluß auf die Klassifikationsleistung gezeigt. Solange die Nähe zur Kamera ausreichend ist, um spezifische Farbmerkmale zu zeigen, bleiben die Histogramme vergleichbar.

Positiv anzumerken sind zudem sowohl die Parameterfreiheit als auch die Geschwindigkeit dieses Verfahrens, die sich stets im Rahmen einer Sekunde bewegte. Beide Merkmale zusammen machen es trotz der beschränkten Objektanzahl zu einer geeigneten Methode im Rahmen der Interaktion mit einem Robotersystem.

### Rekonstruktion

Zur Überprüfung der Kalibrierung wurden 300 Messungen im Bereich mit einem Abstand von  $\pm 300\text{mm}$  vom Kalibrierobjekt durchgeführt<sup>3</sup>. Die Ergebnisse sind nach einzelnen Koordinaten mit dem Mittelwert  $\mu$  und der Standardabweichung  $\sigma$  für die Vorführ- bzw. Ausführunggebung in Tabelle 6.3 aufgetragen.

Koordinate	$\mu$	$\sigma$
$x$	3.8	2.9
$y$	4.0	2.6
$z$	13.8	12.4

(a)

Koordinate	$\mu$	$\sigma$
$x$	1.5	1.8
$y$	2.1	1.9
$z$	3.4	2.3

(b)

Tabelle 6.3: Mittelwerte und Standardabweichungen bei der Szenenrekonstruktion in der Ausführunggebung (a) und in der Vorführunggebung (b) in [mm]

Es ist zu beobachten, dass vor allem die gemessenen  $z$ -Werte mit zunehmendem Abstand von den Kameras schlechter werden. Dieser Effekt zeigt sich deutlich in der Ausführunggebung, in der sich die Kameras sehr nahe nebeneinander befinden (20cm Linsenabstand gegenüber

<sup>3</sup>Eine umfassende Darstellung der Experimente findet sich in [Ambela 99, Spinner 01].

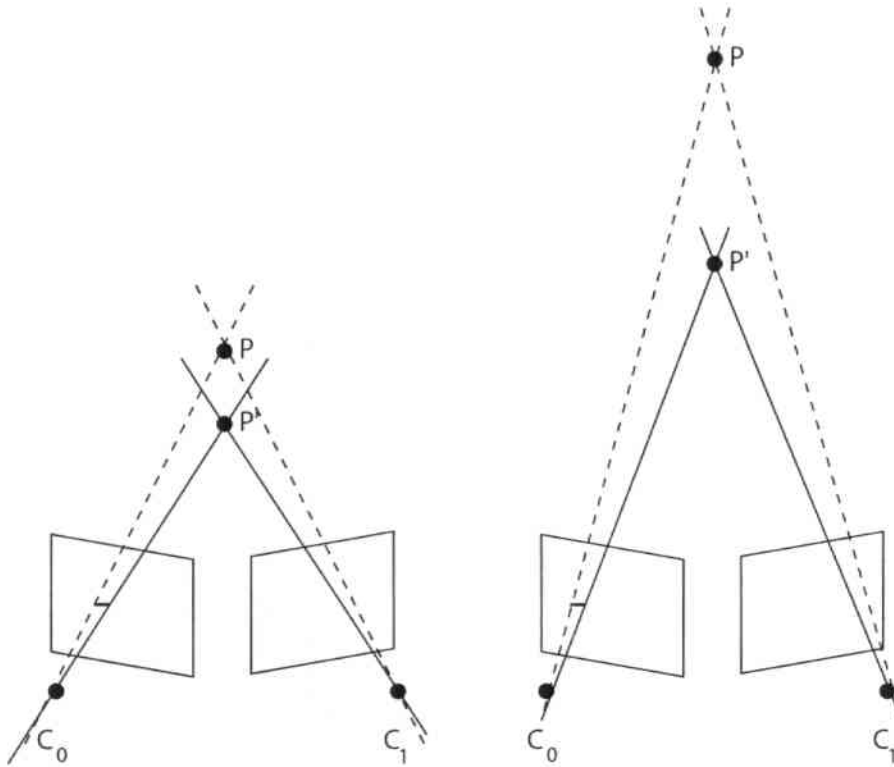


Abbildung 6.5: Auswirkung der Verlängerung der Schnittgeraden bei der Positionsbestimmung

40cm im Fall der Vorführungsumgebung). Dies lässt sich durch eine Fehleranalyse der Rekonstruktion begründen, bei der im ersten Schritt der Schnittpunkt  $P$  der Geraden  $g_0$  und  $g_1$  berechnet wird, die  $P$  mit den optischen Zentren  $C_0$  und  $C_1$  verbinden (siehe Abbildung 6.5). Mit zunehmender Entfernung von  $P$  zu den optischen Zentren der Kameras wird auch der Schnittwinkel zwischen  $g_0$  und  $g_1$  immer kleiner. Eine Abweichung bei der Bestimmung der Bildpunktkoordinaten, der sensorbedingte Auflösungsgrenzen gesetzt sind<sup>4</sup>, wirkt sich dann jedoch stärker auf die Entfernungsschätzung für  $P$  aus. Als Fehlerbereich ergibt sich damit eine 3D-Ellipse, deren längste Halbachse in  $z$ -Richtung liegt (siehe Abbildung 6.6).

### 6.1.2 Bewegungsverfolgung

Die kognitiven Operatoren zur Bewegungsverfolgung finden sowohl in der Vorführ- als auch in der Ausführungsumgebung Anwendung. Im folgenden wird daher zunächst die Handverfolgung während der Benutzerdemonstration analysiert und im Anschluss daran die auf dem Robotersystem implementierten Operatoren.

<sup>4</sup>Ein einzelner Kamerabildpunkt nimmt bei 8mm Brennweite nach Tabelle B.1 und dem Strahlensatz das reflektierte Licht einer Fläche von horizontal fast 2,1mm Durchmesser bei 150cm Entfernung auf. Die Abweichung der  $x$ -Koordinate erweist sich in der Vorführungsumgebung damit als subpixelgroß.

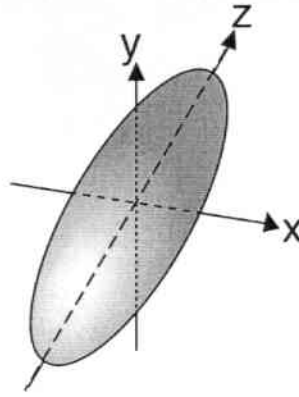


Abbildung 6.6: Fehlerellipse bei der Positionsbestimmung

### Handverfolgung in der Vorführungsumgebung

Die Kombination von einer kontur- bzw. markerbasierten Bildverarbeitungsmethode mit einem magnetfeldbasierten Positionsschätzungsverfahren hat sich, wie bereits in Kapitel 5.2 beschrieben, als günstig für die Verfolgung und Aufzeichnung von Handbewegungen erwiesen. Folgende Beobachtungen und Schlüsse konnten bei der experimentellen Erprobung der Verfahren gemacht bzw. gezogen werden:

**Bildbasierte Konturverfolgung** Als problematisch bei der Konturverfolgung erweist sich die Erzeugung eines geeigneten Referenzmodells. Da die Umrisse der Hand sehr stark variieren, müsste dieses aus dem modellierten Oberflächenmodell gewonnen werden. Dies erfordert jedoch einen hohen Rechenaufwand. Anstelle der Hand können aber auch gegriffene Objekte verfolgt werden, deren Umrisse bereits für die Objekterkennung modelliert sind. Abbildung 6.7 zeigt das Ergebnis eines solchen Ansatzes. Hierbei wurde die Länge der Orthogonalen zur Merkmalsuche auf 25 Bildpunkte festgelegt und als Kantenmerkmal ein Grauwertsprung von mindestens 30 Helligkeitsstufen akzeptiert. Die Verfolgung ist mit diesem Verfahren trotz partieller Verdeckungen und Objektverkippen möglich — trotzdem erwies es sich nicht als effizient für die Bewegungsverfolgung. Grund dafür ist die relativ aufwändige Parametrierung des Verfahrens. Zu Beginn der Verfolgung muss die initiale Position des Objekts im Bild bekannt sein — diese kann zwar mit Hilfe der Verfahren zur Objektdetektion einfach gefunden werden, benötigt jedoch im Rahmen von Handlungssequenzen zu viel Zeit (siehe Tabelle 6.1). Auch die Verfolgung selbst limitiert die Handbewegungen aufgrund der hohen Rechenzeiten auf niedrige Verfahrensgeschwindigkeiten. Die Länge der Orthogonalen kann zwar zur Verfolgung schneller Bewegungen erhöht werden, die Suche ist dann jedoch auch anfälliger gegenüber Grauwertsprüngen im Bildhintergrund. Aufgrund der aufwändigen Matrizenberechnungen erfolgte die Konturanpassung mit einer Frequenz von 15Hz.

**Bildbasierte Markerverfolgung** Im Gegensatz zur konturbasierten Verfolgung erwies sich die Verwendung der Regionenanalyse zum Auffinden des Markers als sehr schnelles Verfahren. Abbildung 6.8 gibt einen Ausschnitt aus einer Bildsequenz zur Handverfol-



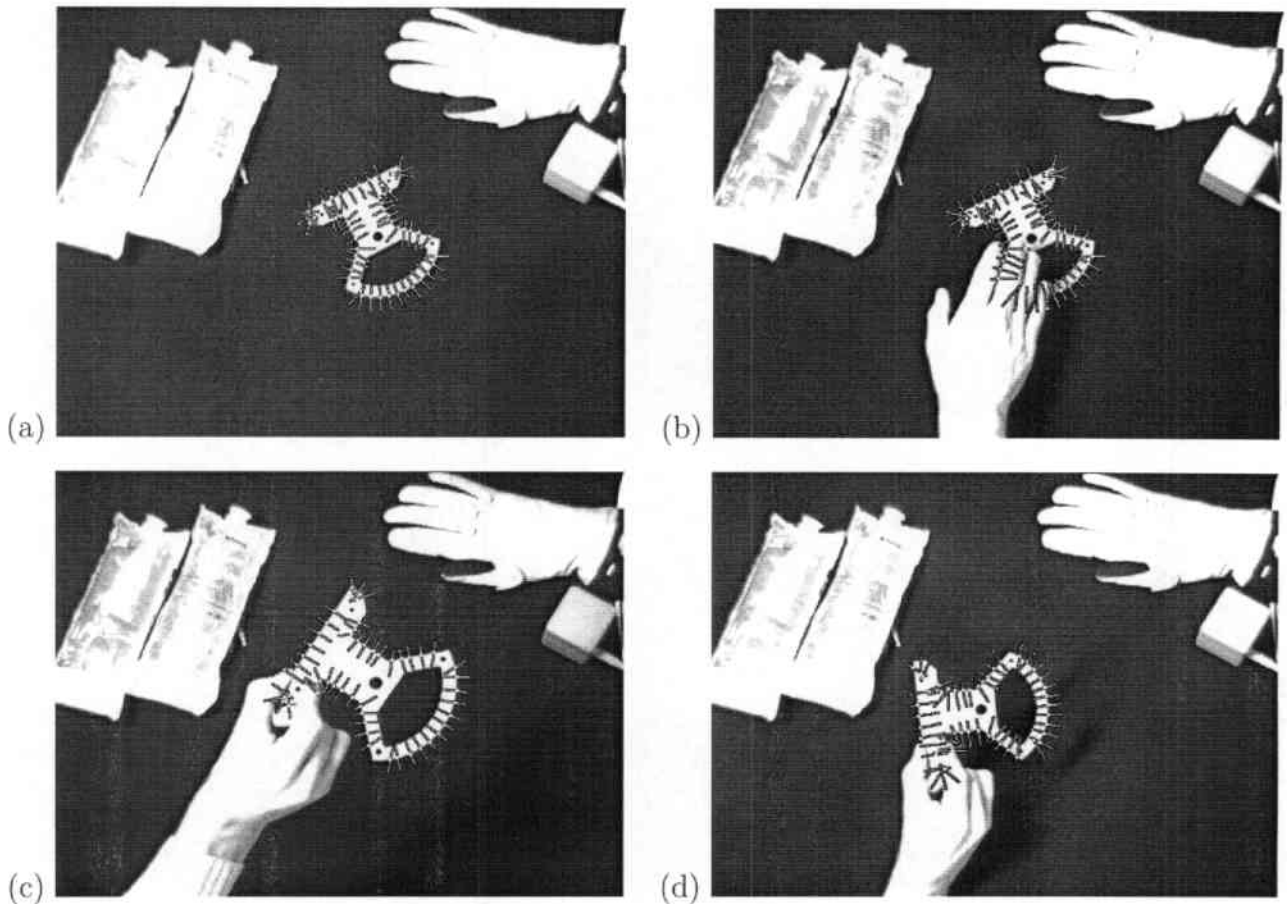


Abbildung 6.7: Bildbasierte Objektverfolgung mit aktiven Konturen

gung wieder. Da die Suche auf Halbbildern und nur im lokalen Fenster erfolgt, ist in den linken Teilbildern die Szenenübersicht aus der Perspektive des Kamerakopfs gezeigt und rechts das lokale Fenster im Kamerabild dargestellt. Im Anwendungsfall erreicht das System eine sehr stabile Leistung von 35 Hz bei abwechselnder Verwendung der beiden Halbbilder. Dabei konnte auch eine hohe Genauigkeit der Rekonstruktion erzielt werden (siehe unten). Leider ist diese zur Orientierungsbestimmung des Markers nicht ausreichend. Aufgrund des geringen Abstands zwischen den beiden kleinen und der großen kreisrunden Markierung kann die Rotation nur auf jeweils ca.  $5^\circ$  genau bestimmt werden. Hervorzuheben ist jedoch die Parameterarmut dieses Verfahrens. Bei Lichtveränderungen ist lediglich der Schwellwert für die Binarisierung anzupassen.

Zur Handverfolgung in der Vorführungsumgebung werden die einzelnen Sensormessungen und deren Fusion betrachtet. Neben der quadratischen Abnahme der Magnetfeldstärke wirken sich auf die Messungen des Magnettrackers vor allem metallische oder stromdurchflossene Gegenstände störend aus. Selbst der zu tragende Datenhandschuh zeigt Auswirkungen auf das Ergebnis. In Abbildung 6.9 sind Messwerte aufgetragen, die zum Einen mit Hilfe des Magnetsensors allein und zum Anderen mit Hilfe des am Datenhandschuh fixierten Magnetsensors gewonnen wurden. Die Verschlechterung liegt hier bereits im Zentimeterbereich. In einem kleinen Bereich um den Magnetfeldemitter kann diese Störung jedoch durch die Addition eines festen, vor jeder Vorführung jeweils neu zu kalibrierenden Ortsvektors

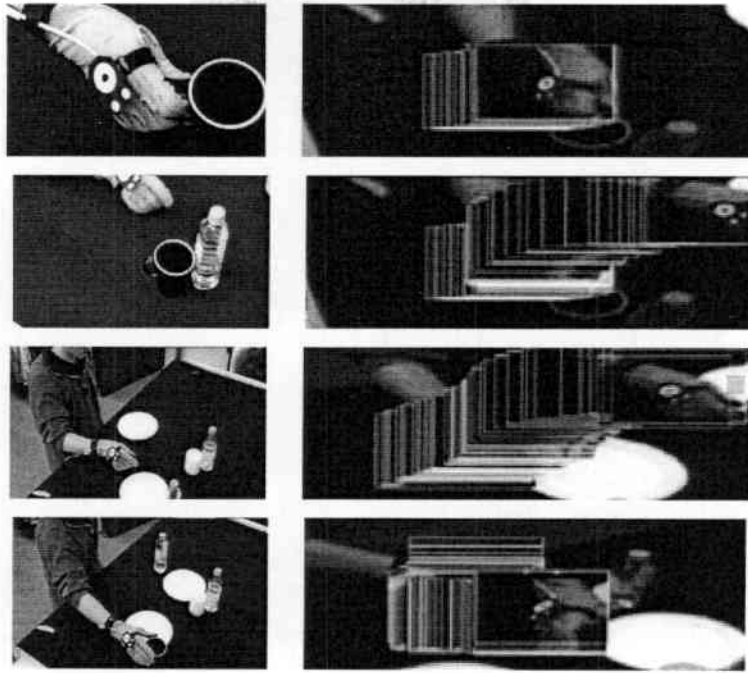


Abbildung 6.8: Bildbasierte Objektverfolgung mit Regionenanalyse (links Bilder aus Kameraperspektive, rechts Kamerabilder mit lokalem Verfolgungsfenster)

beobachtet werden.

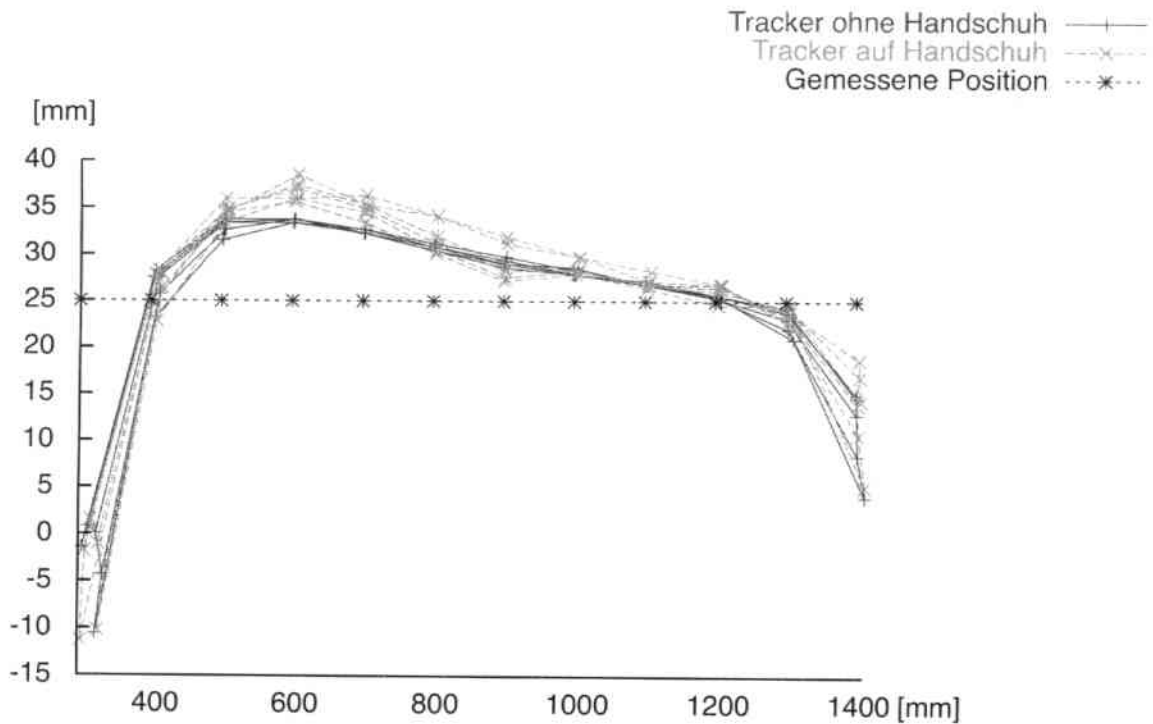


Abbildung 6.9: Vergleich von magnetfeldbasierten Positionsmessungen mit und ohne Datenhandschuh

Die folgenden Untersuchungen zur Sensordatenfusion wurden ausschließlich mit dem Magnetfeldsensor und dem verwendeten markerbasierten Verfahren durchgeführt. Über die gesamte Demonstrationsfläche wurden dazu Messungen mit beiden Sensoren gewonnen, die mit den erwähnten Methoden fusioniert wurden. Eine Statistik mit Abweichungsmittelwerten ist hier wenig aussagekräftig, da die Messqualität bei jedem Sensor von der Position abhängt. Die fehlerträchtigen Extrembereiche beeinflussen dabei die Mittelwerte überproportional. Die mittleren Messfehler in  $x$ ,  $y$  und  $z$ -Richtung von Magnetfeldsystem, Bildverarbeitung und der Fusion beider betragen in Millimetern (430.9; 158.6; 23.8), (15.6; 12.6; 20.1)<sup>5</sup> und (26.7; 20.9; 32.2). Das Untersuchungsergebnis findet sich mit allen Einzelmessungen in Abbildung 6.10. Dort lässt sich ablesen, dass die Fusion das Messergebnis wesentlich verbessert hat.

Die Möglichkeit, die ungenauen Positionsangaben des magnetfeldbasierten Trackingsystems als Schätzung an die Bildverarbeitung schicken zu können, stabilisiert die Handverfolgung. Kurzzeitige Verdeckungen des Markers werden durch den zweiten Sensor kompensiert, während die Ungenauigkeiten des Magnetsystems durch die visuell gemessenen Lagewerte ausgeglichen werden. Abbildung 6.11 zeigt die gemessenen Werte für die  $z$ -Achse während der Vorführung einer Trajektorie. Im oberen Bild ist zu sehen, wie das Magnetsystem Werte generiert, solange die Kameras den Marker noch nicht fokussieren können. Erst ab der dritten Sekunde in der Aufnahme tragen beide Sensoren zur Messung bei. Gelegentliche Verfolgungsverluste und Fehlmessungen der Bildverarbeitung fließen ohne erkennenswerte Störung in das Messergebnis ein. Das untere Bild zeigt hingegen an der Zeitposition 6 und zwischen der 20. und 25. Sekunde Ausschläge, die das Magnetfeldsystem aufgrund hoher Entfernungen vom Emitter erzeugt. Diese werden hier durch die Bildverarbeitung aufgefangen.

### Handverfolgung in der Ausführungsumgebung

Bei einer Segmentiertrate von  $\geq 12\text{Hz}$  lässt sich eine Benutzerhand nahezu kontinuierlich verfolgen<sup>6</sup>. Die Bildsequenz in Abbildung 6.12 zeigt die Farbänderung im Kamerabild mit dem entsprechenden Segmentierungsergebnis über einen Verlauf von 10 Sekunden. Dabei wurde die Szene durch Tageslicht und einen mit einem Farbfilter präparierten Strahler beleuchtet, der eine Rot-, Grün- und Blauverschiebung hervorrief. Das Segmentierungsergebnis bleibt hier stabil und lässt die Handposition schätzen. In die Bilder sind der jeweilige Handschwerpunkt, das genutzte Kalibrierfenster und die geschätzte Position direkt eingetragen.

Abbildung 6.13 zeigt den Verlauf des Mittelwerts  $\mu$  sowie der Varianz  $\sigma$  aus Algorithmus 5.5 in zwei weiteren Experimenten. Im linken Teilbild wurde dazu die Hand ruhig gehalten und ein roter Gegenstand in den Bildhintergrund geführt. Die Reaktion auf die darauf resultierende Kameraadaptation ist ab dem 60. verarbeiteten Bild als Absenkung des Mittelwertes sichtbar. Das rechte Bild zeigt den entsprechenden Verlauf bei einem Kameraschwenk durch das Labor. Die Schwankungen ergeben sich hier ebenfalls aufgrund der Veränderungen im Bildhintergrund. In beiden Fällen ist die Szene durch Tageslicht beleuchtet worden, das über

<sup>5</sup>bei den durch die Kameras gemessenen Werten ist zu beachten, dass lediglich der statische Fehler erfasst wurde.

<sup>6</sup>Eine umfassende Darstellung der Experimente findet sich in [Ly Duc 01].

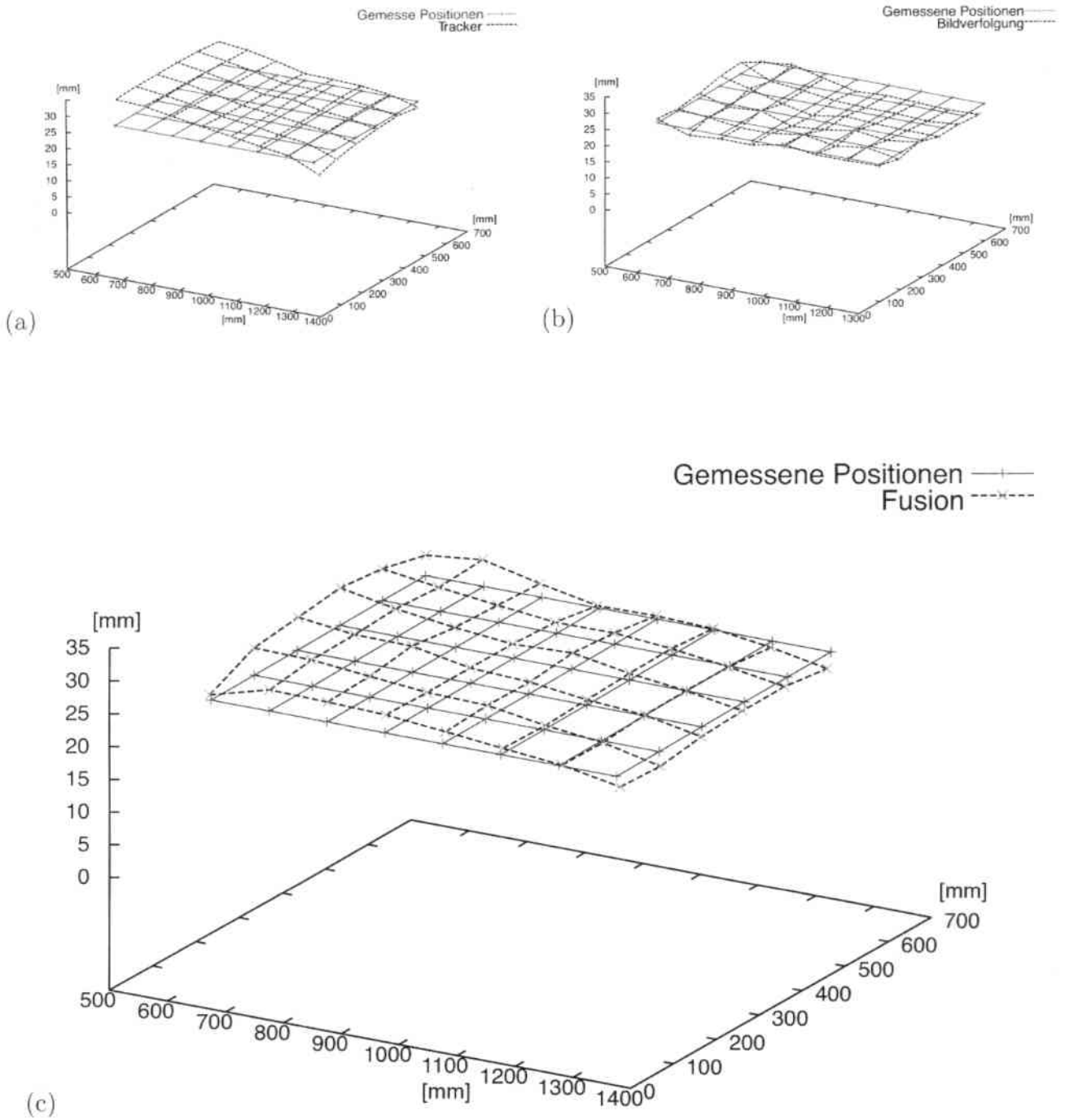


Abbildung 6.10: Vergleich von Positionsmessungen der einzelnen Sensoren: magnetfeldbasiert (a), bildbasiert (b) und fusioniert (c)

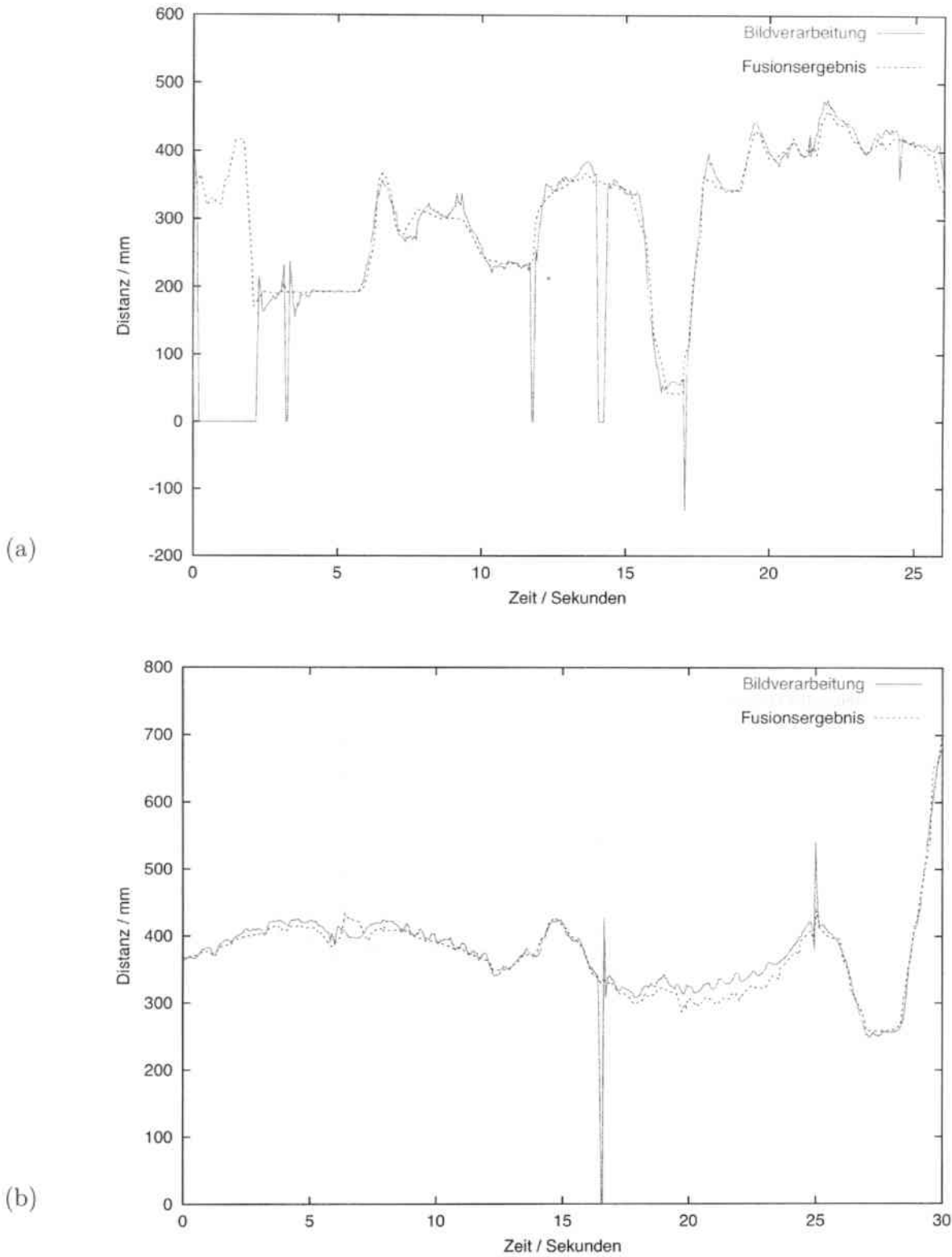


Abbildung 6.11: Detailergebnisse der Datenfusion:  $z$ -Achse einer gemessenen Handtrajektorie mit Kompensation von Markerverlusten durch Verdeckungen im Kamerabild (Nullwerte in Teilbild a) und von Magnetfeldstörungen (Ausschlag ab der 6. und 19. Sekunde in Teilbild b)

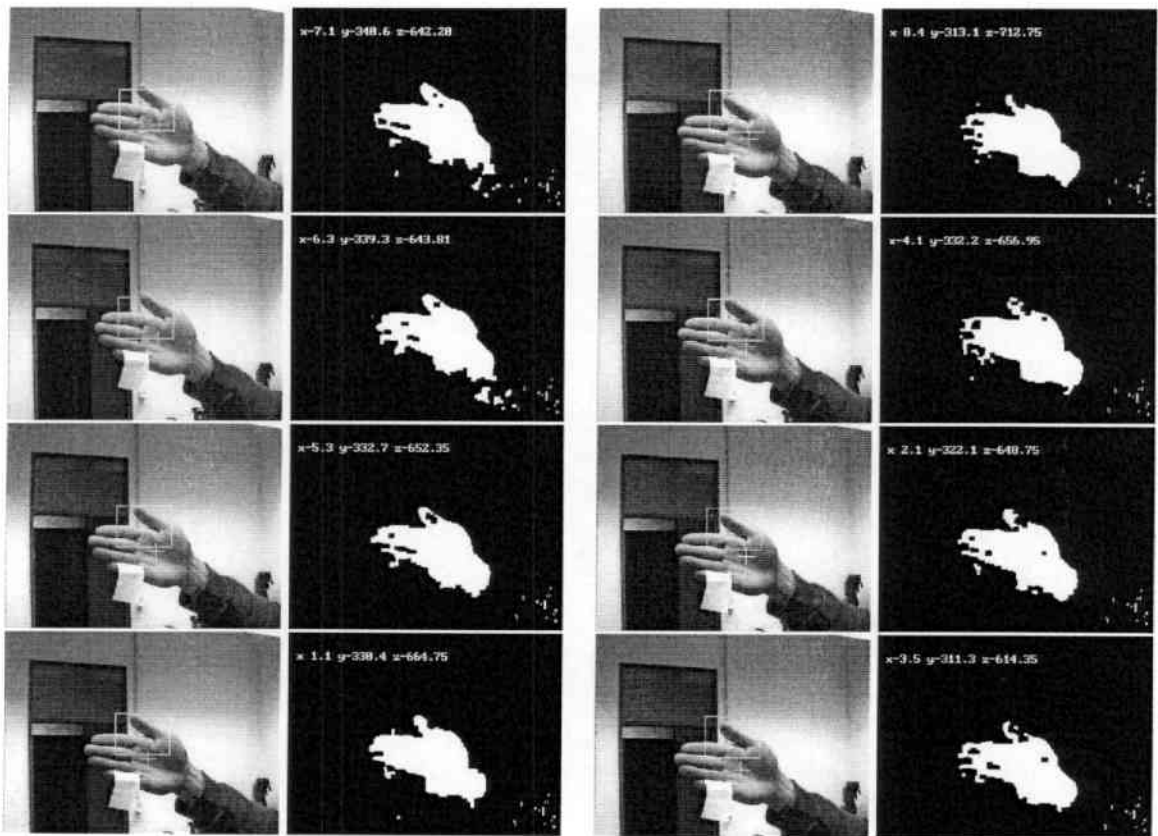


Abbildung 6.12: Experiment zur farbbasierten Segmentierung mit wechselnder Beleuchtung



den Kamerakopf hinweg auf die Hand fiel. Sobald  $\mu$  unter zu großen Störeinflüssen divergiert, ist die Segmentierung nicht mehr stabil. Die Varianz wird deshalb auf ein Maximum begrenzt.

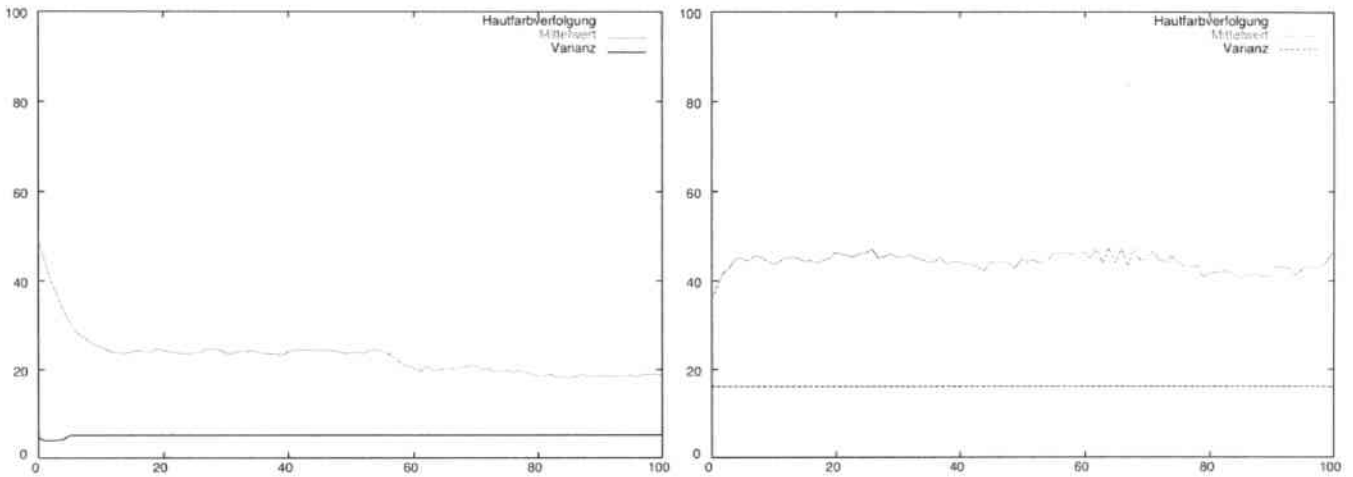


Abbildung 6.13: Verlagerung des Mittelwerts bei der Hautfarbverfolgung

Versuche, zur Lösung des Korrespondenzproblems ähnliche Bildausschnitte durch Schablonenanpassung im rechten Bild zu suchen, sind nicht befriedigend verlaufen. Zur Geschwindigkeitssteigerung wurden diese mit einem Farbkanal, der Helligkeit oder dem Farbwinkel durchgeführt, brachten aber aufgrund der sich durch die Parallaxenverschiebung ergebenden unterschiedlichen Perspektiven schlechte Ergebnisse. Stabiler hat sich hier die Segmentierung mit den im Originalbild genutzten Schwellwerten erwiesen. Auch bei sehr kameranahen Positionen wurde der Schwerpunkt der segmentierten Fläche trotz geringfügiger Unterschiede dem Ausgangsbild entsprechend gefunden.

### 6.1.3 Griffenerkennung mit Hilfe des Datenhandschuhs

Zur Klassifikation von Griffen mit Hilfe des Datenhandschuhs werden die in [Friedrich 98] untersuchten neuronalen Netze in Kombination mit einer Betrachtung von objektspezifischen Griffattributen eingesetzt. Bei den Netzen handelt es sich um dreischichtige „feed-forward“ Netze, in deren Neuronen radiale Gaußfunktionen als Aktivierungsfunktionen eingesetzt wurden.

Zur Einstellung der Gewichte wurde dazu eine Trainingsdatenmenge von 1280 Griffbeispielen (80 pro Grifftyp) für jedes zu trainierende bzw. zu testende Schichtnetz (Abbildung 5.21) kopiert und bzgl. der entsprechenden Klassifikationsaufgabe konfiguriert. Die Beispiele wurden von 10 Probanden erzeugt, um die Netze nicht benutzerabhängig zu trainieren. Die Trainingsmenge für ein Schichtnetz bestand aus allen Griffbeispielen der Grifftypen der Cutkosky-Hierarchie, die sich in den Teilbäumen der Grifffhierarchie befanden, welche das Netz unterscheiden sollte. Alle in einer Schicht nicht benötigten Trainingsbeispiele wurden entfernt. Die Trainingsbeispiele wurden bzgl. der zu unterscheidenden Ausgabeklassen markiert.

Werden zu klassifizierende Handschuhmessdaten gemäß der vorgeschlagenen hierarchischen

Daten	Griffart							
	1	2	3	4	5	6	7	8
Test	0.95	0.90	0.90	0.86	0.98	0.75	0.61	0.90

Daten	Griffart							
	9	10	11	12	13	14	15	16
Test	0.88	0.95	0.85	0.53	0.35	0.88	1.00	0.45

Tabelle 6.4: Klassifikationsleistung des neuronalen Klassifikators nach Friedrich bzgl. der einzelnen Griffarten auf den Testdaten

Klassifikationsstrategie durch das Gesamtnetz propagiert, so ergibt sich für die verschiedenen Griffarten die in Tabelle 6.4 dargestellte Klassifikationsleistung. Die Testdatenmenge enthielt pro Griff 40 Beispiele, insgesamt 640. Die durchschnittliche Klassifikationsleistung betrug 83,28%. Gegenüber einem Ansatz mit einem Klassifikator, der aus einem einzigen Netz besteht [Rogalla 98], konnte dabei die Erkennungsleistung wesentlich gesteigert werden.

In [Friedrich 98] wird bereits nachgewiesen, daß diejenigen Fälle, in denen die Klassifikationsleistung abnimmt, solche sind, bei denen die kinematische Information der Finger der Benutzerhand allein nicht für eine Erkennung des Griiffs ausreicht. Beispiele dafür sind die Präzisionsgriffe von scheiben- und kugelförmigen Objekten. Sobald die objektspezifische Griffinformation entsprechend Algorithmus 5.7 in die Klassifikation einbezogen wird, verbessert sich die Erkennungsleistung. In verschiedenen Situationen wurden zu Testzwecken Objekte in der Vorführungsumgebung gegriffen. Jeder Griff wurde dabei 10 mal ausgeführt. Der Plattformgriffartyp 15 kam nicht zum Einsatz, weil mit ihm einhändig keine Gegenstände aufgenommen werden können (die Ergebnisse sind in Tabelle 6.5 aufgeführt). Die Registrierung einer Rückfrageanforderung an den Benutzer zur Überprüfung des Klassifikationsergebnisses erfolgte dabei insgesamt lediglich 8 mal. Diese Ereignisse wurden nicht als positives Ergebnis gewertet.

Daten	Griffart															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Test	1.0	1.0	0.9	0.9	1.0	0.9	0.9	0.9	0.9	1.0	0.9	0.8	0.8	0.9	-	0.8

Tabelle 6.5: Klassifikationsleistung des Griffklassifikators nach Algorithmus 5.7 in Beispielsituationen

Gegenüber der ausschließlich auf Konfigurationsmustern beruhenden Klassifikation zeigt sich hier genau an den Stellen eine Verbesserung, die ohne geometrische Information nicht ge-

trennt werden können. Die Grifftypen 11 und 13 wie auch die Typen 10 und 12 können jedoch immer noch nicht sicher unterschieden werden. Diese betreffen die gleichen Objekttypen bei sehr ähnlicher Fingerkonfiguration, berühren sie jedoch entweder mit den Fingerspitzen für die präzise Manipulation oder mit der gesamten Handinnenfläche. Hier kann die Klassifikation nur mit kraft- oder berührungsempfindlicher Sensorik verbessert werden. Die Ergebnisse der Experimente mit einer durchschnittlichen Erkennungsleistung von 0.91 zeigen, daß die korrekte Klassifikation von Griffen mit dem beschriebenen Verfahren möglich ist.

### 6.1.4 Gestenerkennung

Wie im Fall der Bewegungsverfolgung wurden mehrere Ansätze zur Gestenerkennung untersucht. Diese betreffen die Gestenerkennung auf Basis der Fingergelenkmessungen des Datenhandschuhs sowie die Verfahren zur Klassifikation statischer wie dynamischer Gesten auf Basis der Bildverarbeitung.

#### Erkennung statischer Gesten mit Hilfe des Datenhandschuhs

Aufgrund der zufriedenstellenden Ergebnisse der Griffenerkennung wurden für die Gestenerkennung in der Vorführungsumgebung ebenfalls dreischichtige „feed-forward“ Netze mit radialen Gaußfunktionen als Aktivierungsfunktion eingesetzt. Die Test- und Trainingsdatenmenge hatte einen ähnlichen Umfang (pro Geste 40 Test- und 40 Trainingsmuster) von acht verschiedenen Personen<sup>7</sup>. Die Klassifikation der Handstellungsmuster funktioniert damit analog zur Griffenerkennung. Beim Gestentyp 1 und 2, den „Ja/Nein-Gesten“ und Zeigegesten muss zur Interpretation zusätzlich die Position und Orientierung der Hand betrachtet werden. Im ersten Fall lässt sich über die Orientierung des Handgelenks zweifelsfrei feststellen, ob der Daumen nach oben oder unten gerichtet ist; bei der Zeigegeste wird zur Feststellung der Zielrichtung die Raumlage der Handfläche genutzt.

Da sich die zu erkennenden symbolischen Gesten nicht hierarchisch anordnen lassen, sollte ein einziges Netz die Klassifikationsaufgabe übernehmen. Abbildung 6.14 zeigt die Klassifikationsleistung von Netzen mit unterschiedlicher Neuronenanzahl in der mittleren Schicht bei Training mit einer unabhängig von der Test- und Trainingsmenge aufgenommenen Menge von Beispielen. Bei jeder Topologie wurde die Lernphase fünfmal mit unterschiedlichen Startbelegungen der Gewichte wiederholt. Man erkennt, daß vier Neuronen für eine hohe Klassifikation ausreichend sind. Bei Training mit der Testmenge wurde eine Trefferrate von 90.8 % erzielt. Die Einzelergebnisse sind nach den Gestentypen in Tabelle 6.6 aufgeführt.

Daten	Gestentyp						
	1/2	3	4	5	6	7	8
Test	0.88	0.95	0.96	0.94	0.96	0.66	0.99

Tabelle 6.6: Klassifikationsleistung bzgl. der einzelnen Gestentypen auf den Testdaten

<sup>7</sup>Eine umfassende Darstellung der Experimente findet sich in [Ly Duc 00].

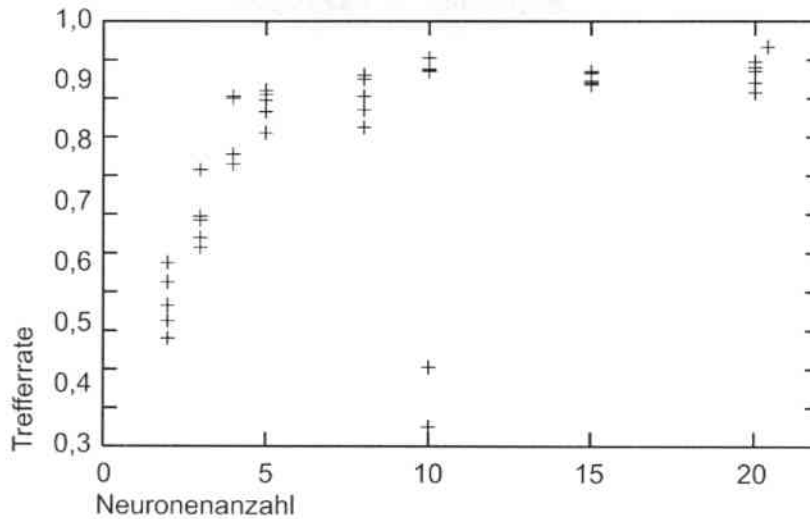


Abbildung 6.14: Klassifikationsleistung von Neuronalen Netzen zur Gestenerkennung in Abhängigkeit von der Hidden-Neuronen-Anzahl

Die Gestentypen 1 und 2 sind hier zusammengefasst, da sich in der Fingerstellung kein Unterschied zeigt. Unterscheidungsmerkmal ist hier lediglich die Zeigerichtung des Daumens. Das im Vergleich zu den anderen Gesten niedrige Ergebnis der Klassifikation der Handstellung beim Gestentyp 7 („Victory“-Zeichen) hat seine Ursache in der Ähnlichkeit zum Typ 8 (Kommandiergeste mit zwei Fingern). Dies geht aus einer Betrachtung der jeweiligen Falschklassifikationen hervor, die in Tabelle 6.7 für die Testmenge aufgelistet sind. Hier lässt sich das Klassifikationsergebnis steigern, indem Kontextinformation zur Interpretation hinzugenommen wird.

		Sollausgabe						
		1/2	3	4	5	6	7	8
Ist-Ausgabe	1/2	0	1	0	2	0	0	0
	3	0	0	0	0	1	0	0
	4	0	0	0	0	1	0	0
	5	5	0	1	0	0	5	0
	6	0	2	0	0	0	0	0
	7	0	0	0	0	0	0	0
	8	0	0	0	0	0	20	0

Tabelle 6.7: Verwechslungen auf der Testmenge des Gestenklassifikators. Die Zahlen bedeuten den zu erkennenden bzw. erkannten Gestentyp aus Tabelle 4.2

**Integration von Griff- und Gestenerkennung** Da sowohl der Gesten- wie auch der Griffkennner auf den Messungen des Datenhandschuhs operieren, ist es möglich, dass beide Klassifikatoren bei einer Messung feuern. Die Faust ist beispielsweise dem Griffotyp zum Halten kleiner zirkulärer Objekte sehr ähnlich (Griffotyp 10 und Gestentyp 5). Deshalb ist eine Untersuchung von Fehlklassifikationen zwischen beiden Erkennern notwendig.

Daten	Schwellwert $\theta$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Test	0.73	0.72	0.72	0.72	0.72	0.70	0.68	0.65	0.61	0.52
Fehler erster Art	0.27	0.28	0.28	0.28	0.28	0.30	0.32	0.35	0.39	0.48
Fehler ohne $\theta$	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.26	0.26	0.23
Rückweisungsrate	0.00	0.05	0.05	0.07	0.01	0.03	0.06	0.09	0.13	0.25
Fehler zweiter Art	0.00	0.00	0.00	0.00	0.10	0.32	0.35	0.42	0.46	0.56

Tabelle 6.8: Trefferrate des Gesamterkenners zur Griff- und Gestenklassifikation in Abhängigkeit vom Schwellwert in [%]

Akzeptiert man bei beiden Klassifikatoren das Feuern von Neuronen in der Ausgabeschicht erst ab dem Überschreiten eines Schwellwerts  $\theta$  und entscheidet sich dann für dasjenige Ausgabeneuron mit dem höchsten Aktivierungswert aus beiden Netzen, lassen sich die Erkennungsleistungen auf der vereinigten Testmenge entsprechend Tabelle 6.8 zusammenfassen. Hier lässt sich beobachten, dass mit dem Sinken des Fehlers erster Art bei steigendem  $\theta$  auch die Rate fälschlich zurückgewiesener Muster steigt. Als optimale Schwelle stellt sich ein Wert um 0.3 zur Vermeidung des Fehlers zweiter Art dar. Wie schon oben sind auch hier die Verwechslungsfehler bedingt durch Ähnlichkeiten der Handkonfigurationen. Dies beweist eine Übersicht über die auftretenden Fehlklassifikationen des Gesamtklassifikators. In Tabelle 6.9 sind diese bei einer verwendeten Schwelle von  $\theta = 0.39$  aufgetragen. Es zeigt sich beispielsweise, dass 53 Verwechslungen die Haltegeste (Gestentyp 3) und den Plattformkraftgriff (Griffotyp 15) betreffen. Weitere 28 Fehlklassifikationen betreffen die Ja/Nein-Geste (Gestentypen 1 und 2) und den seitlichen Kniff (Kraftgriff Typ 16). Diese Verwechslungen machen zusammen bereits 9% der gesamten Testmenge aus.

Aufgrund dieser Gegebenheiten macht es auch keinen Sinn, die beiden Klassifikatoren mit Handmustern der jeweils anderen Testmenge als Gegenbeispiele zu trainieren. Die fehlende Disjunktheit und damit auch schlechte Trennbarkeit der Mengen zeigte sich bei einem Experiment, bei dem ein mit Griffgegenbeispielen trainiertes Netz zur Gestenerkennung mit  $\theta = 0.3$  eine Trefferquote von 72,9% erreichte und damit unwesentlich besser abschnitt als der Gesamtklassifikator. Zwar wurden hier keine Griffe als Gesten interpretiert, die Trennung der Gestentypen wurde jedoch unstabiler.

**Fazit des Verfahrens zur Gestenklassifikation.** Mit den Tabellen 6.6 und 6.8 wird deutlich, dass nicht nur die Gesten mit dem beschriebenen Verfahren präzise klassifiziert

		Sollausgabe																							
		Gesten								Griffe															
		1/2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1/2		0	0	0	0	0	0	0	0	3	2	5	0	1	0	7	3	1	2	2	0	0	0	0	28
3	Gesten	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0
5		5	0	1	0	0	5	0	1	2	0	0	0	0	0	0	2	0	1	0	0	0	1	0	0
6		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
7		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8		0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2		0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3		0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4		0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5		0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8		0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10		0	0	9	0	0	17	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11		0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12		0	0	0	0	0	12	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14		0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15		0	53	1	0	9	9	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Gesten								Griffe															
		Ist-Ausgabe																							

Tabelle 6.9: Verwechslungen des Gesamterkenners mit Schwellwert. Die Zahlen bedeuten den zu erkennenden bzw. erkannten Griff- bzw. Gestentyp aus Tabelle 4.2 und aus der Cutkosky-Hierarchie in Abbildung 4.13

werden können, sondern dass Griffe und Gesten auch in einer Vorführungsumgebung mit den ausgewählten Sensoren gleichzeitig genutzt werden können. Durch das Handlungsmodell lässt sich wie durch eine Grammatik die Reihenfolge von Gesten und Griffe für einen Interaktions-



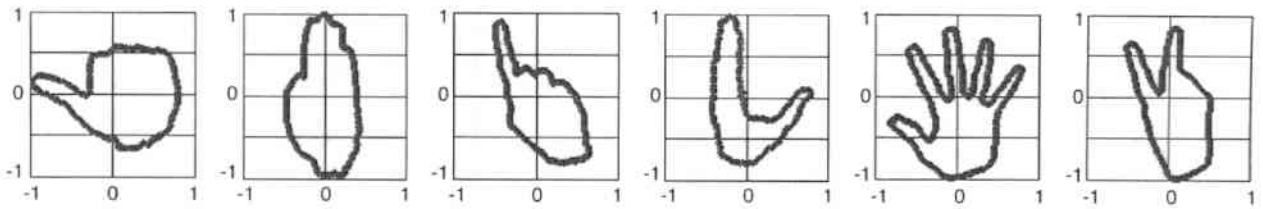


Abbildung 6.15: Statische Referenzgesten für die experimentelle Validierung

kontext charakterisieren, um die Trennung von Griffen und Gesten weiter zu erhöhen. Die damit definierten Handlungsfolgen stellen eine bestimmte Beschränkung der Abfolge möglicher Gesten und Griffen dar und geben eine gewisse Vorhersageinformation für die Wahl des Erkenners. Durch die Hinzunahme taktiler Informationen könnten die meisten Kollisionen vollständig aufgelöst werden, da diejenigen Handkonfigurationen, die zu hohen Verwechslungsraten führen, vorrangig bei den Griffen Berührungen an den Fingerspitzen erkennen lassen sollten.

### Erkennung statischer Gesten in der Ausführungsumgebung

Zum Test der kameragestützten Gestenerkennung wurden in eine Experimentierbibliothek sechs unterschiedliche Gestentypen von zwölf verschiedenen Personen zur Validierung aufgenommen<sup>8</sup>. Da die Erkennung rotationsunabhängig erfolgt, können hier wie im Fall der Erkennung über den Datenhandschuh die Gestentypen 1 und 2 zusammengefasst werden. Aufgrund der Ähnlichkeit der Silhouette der Gestentypen 3 und 5 wie auch der Typen 4 und 8 werden im Hinblick auf die Robustheit des Verfahrens keine Referenzgesten für die Typen 5 und 8 in die Untersuchung mit einbezogen. Stattdessen wird für die Ausführungsumgebung eine Handkonfiguration aufgenommen, die dem Greifen von Objekten dient. Jeder Typ wird von jeder Person in sechs Varianten für den Modellaufbau und dreißig Varianten für den Test präsentiert. Zusätzlich werden sechs verschiedene Handkonfigurationen aufgenommen, die keinem der Referenztypen angehören. Abbildung 6.15 zeigt die ausgerichteten und normalisierten Umriss.

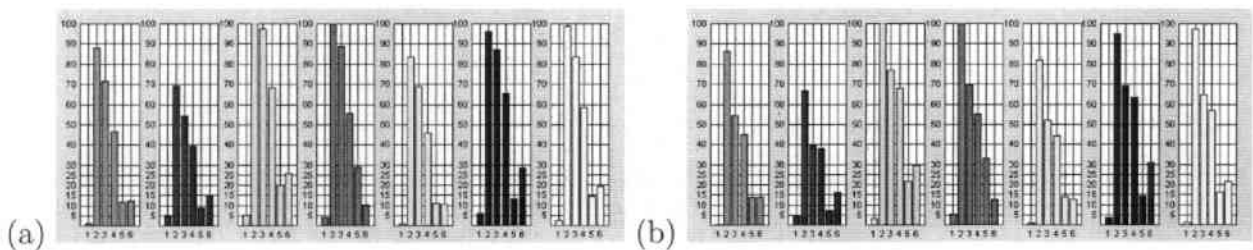


Abbildung 6.16: Ergebnisvergleich einer Geste vom Typ 3 bei gemitteltem Modell (a) und ausgewähltem besten Modell (b)

Damit kann sowohl der Modellaufbau als auch die Klassifikationsleistung beurteilt werden:

<sup>8</sup>Eine umfassende Darstellung der Experimente findet sich in [Gimeno 01].

**Modellaufbau:** Zum Vergleich des Modellaufbaus mit Hilfe der in Abschnitt 5.4.2 erläuterten Methoden nach dem Prinzip der Bestenauswahl bzw. der Mittelung über die Beispiele dient die Betrachtung mehrerer Referenzvergleiche. Exemplarisch ist in Abbildung 6.16 das Vergleichsergebnis der Euklidischen Distanz nach Präsentation einer Stoppgeste abgebildet, links bei gemittelten Modellen und rechts bei Modellauswahl nach der Bestenmethode. Hier können zwei Eigenschaften beobachtet werden:

- Die Euklidischen Distanzen fallen bei Korrespondenz zwischen Präsentation und Referenz bei den Modellen auf Basis des Durchschnitts mehrerer Präsentationen in der Regel geringer aus. Andersherum sind die Differenzen hier bei Nichtübereinstimmung größer. Der Distanzvergleich beim Entscheidungskriterium kann also bei diesem Typ robuster arbeiten.
- Die Methode der Bestenauswahl liefert in einigen Fällen gleich große Distanzen wie die Durchschnittsmethode, jedoch betragsmäßig auf höherem Niveau. Der Unterschied ist damit proportional kleiner.

**Klassifikationsleistung:** Die mit der Durchschnittsmethode gewonnenen Modelldeskriptoren der Gestenbibliothek zeigen bei einem Klassifizierungstest mit den wie oben erwähnt gewonnenen Testbeispielen die in Tabelle 6.10 aufgelisteten Ergebnisse. Die Typen entsprechen dabei den in Abbildung 6.15 gezeigten, insbesondere ersetzt die Griffgeste 5' die Faustgeste (vierte Kontur von links). Mit aufgeführt sind als letzter Typ „ $\emptyset$ “ Handstellungen, die keiner der aufgeführten Gesten entsprechen.

Daten	Gestentyp						
	1/2	3	4	5'	6	7	$\emptyset$
Test	1.00	0.96	0.96	0.83	1.00	1.00	0.96
Fehler erster Art	0.00	0.04	0.04	0.08	0.00	0.00	0.00
Fehler zweiter Art	0.00	0.00	0.00	0.08	0.00	0.00	0.04

Tabelle 6.10: Ergebnis der Klassifikation statischer Handgesten

Die Klassifikation wird mit einem Durchschnittswert von 95,9% korrekt erkannter Handkonfigurationen ihrer Aufgabe sehr gut gerecht. Das niedrigste Resultat wird bei der Griffgeste 5' erhalten. Der Grund dafür liegt in der Modellgewinnung—dieser Typ ist sehr unscharf definiert (siehe Abbildung 6.17). Die Probanden haben hier Fingerstellungen mit sehr unterschiedlichen Abständen zwischen Zeigefinger und Daumen präsentiert. Das Klassifikationsergebnis könnte hier durch Muster mit einer höheren Spezifität verbessert werden, würde dann jedoch auch bei der Erkennung eine gute Reproduktion der Referenzkonfiguration erfordern.

Es muss jedoch darauf hingewiesen werden, dass die Qualität der Konturklassifikation in hohem Maße von der Segmentierung abhängt. Bei schlechten Beleuchtungsverhältnissen und

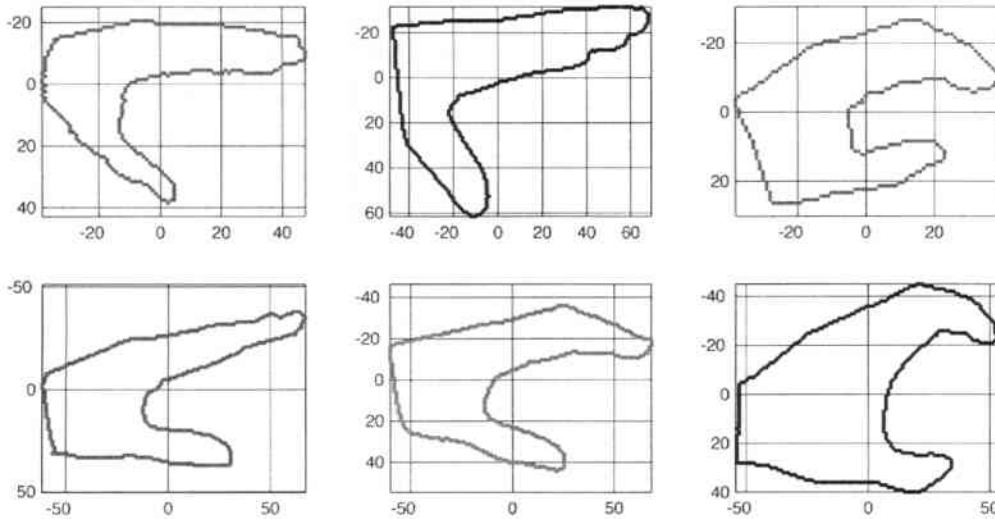


Abbildung 6.17: Musterkonturen für den Griffgestentyp 5'

entsprechendem Rauschen im Handumriss kann die Konturabtastung nicht mehr zu Abstandsdeskriptoren führen, die zu einer korrekten Klassifikation notwendig sind.

### Erkennung dynamischer Gesten

Zum Test der Klassifikation wurden die fünf dynamischen Referenzgesten mit je 10 Beispielen trainiert (siehe Tabelle 4.3)<sup>9</sup>. Es wurden keine modellierten, idealtypischen Gesten zum Training verwendet, da bei einer künstlichen Modellierung die Gefahr besteht, den Anforderungen im praktischen Einsatz nicht gerecht zu werden. Bei der Auswahl der Gesten wurde im Hinblick auf die Robustheit des Verfahrens darauf geachtet, dass die Gesten nicht Teiltrajektorien voneinander enthalten und eine große Verschiedenheit in ihren geometrischen Merkmalen aufweisen. Für die Ausführung seitens des Benutzers gilt, dass sie auch nicht zu komplex werden sollten. Bei Geste 5 beispielsweise haben mehrere Probanden die Ansicht geäußert, dass die Ausführung zu lange dauert und zu viel Konzentration erfordert. Beim Einsatz zur Kommandierung sollte daher auf solche Verfahrenen verzichtet werden.

Im folgenden sollen analog zu den statischen Gesten Ergebnisse der Methoden zum Modellaufbau und zur Klassifikation besprochen werden:

**Modellaufbau:** Nach [Lee 99] und [Rabiner 89] hat die Anzahl der Zustände der Modelle  $\lambda_i$  nur bedingten Einfluss auf die Erfolgsrate des Erkennungssystems. Obwohl bei einer bestimmten Anzahl von Zuständen ein lokales Maximum erreicht wird, führt die weitere Hinzunahme von Zuständen zu keiner signifikanten Verbesserung. Nachteilig wirkt sie sich aber auf den Berechnungsaufwand bei Training und Erkennung aus, da eine höhere Anzahl von Parametern mit einbezogen werden muss. Eine hohe Anzahl von Zuständen hat insbesondere bei der Konstruktion des Schwellwertmodells negative Folgen (Abschnitt 5.4.3).

<sup>9</sup>Eine umfassende Darstellung der Experimente findet sich in [Lütticke 00].

Tabelle 6.11 zeigt die Entwicklung der Erkennungswahrscheinlichkeiten für die Gesten eins und fünf bei wachsender Anzahl von Zuständen und bei der Anwendung auf eine Beobachtungssequenz der Länge 16. Mit angegeben ist die Anzahl der zur Erkennung notwendigen Berechnungen, die sich durch  $N^2T$  ergibt (Der Viterbi-Algorithmus hat die Komplexitätsklasse  $O(N^2T)$  über der Länge  $T$  der Beobachtungssequenz bei  $N$  Zuständen des Modells).

Zustände	$P(O  \text{Geste 1})$	$P(O  \text{Geste 5})$	Berechnungen
3	1.374206E-04	6.364537E-13	144
4	1.405220E-03	8.566302E-12	256
5	1.537310E-03	2.597614E-11	400
6	1.689013E-03	2.872422E-10	576
8	4.477910E-03	3.251160E-09	1024
10	2.431920E-02	2.010504E-08	1600
12	./.	8.175456E-07	2304
15	./.	8.065692E-05	3600

Tabelle 6.11: Erkennungswahrscheinlichkeiten bei wachsender Anzahl von Zuständen für Geste Nummer eins und fünf

Es ist erkennbar, dass mit steigender Anzahl von Zuständen auch die Erkennungswahrscheinlichkeit ansteigt. Aus der Wertentwicklung für Geste eins lässt sich erkennen, dass ab einer gewissen Anzahl von Zuständen ein signifikanter Zuwachs bei der Wahrscheinlichkeit nur durch eine deutlich höhere Zustandsanzahl erkaufte werden kann. Die verbesserten Wahrscheinlichkeiten bei Geste fünf für die aufgeführten Zustandsanzahlen begründen sich in der größeren Komplexität dieser Geste. Für ihre Modellierung sind mehr Zustände nötig als für die vom Bewegungsablauf her einfachere erste Geste.

Da mit steigender Anzahl von Zuständen eines Modells aber auch der Berechnungsaufwand quadratisch wächst, muss bei der Wahl der Zustandsanzahl abgewogen werden. Modelle mit mehr Zuständen liefern tendenziell auch bessere Ergebnisse, allerdings beschränken diese aufgrund der zusätzlichen Rechenzeit die Echtzeitfähigkeit. Im Idealfall entspräche ein Ausgabesymbol einem Zustand im Hidden-Markov-Modell. Eine solche Realisierung führt aber zu Modellen mit sehr hoher Anzahl von Zuständen, was wiederum einen hohen Berechnungsaufwand nach sich zieht. Gesten, die aus Geraden zusammengesetzt sind, können durch einen Richtungsindex pro Gerade beschrieben werden und führen zu Modellen, die einen Zustand pro Gerade aufweisen. Die Praxis aber zeigt, dass Handbewegungen entlang einer Geraden nahezu nie dem Ideal entsprechen und sich oft eine Folge zweier benachbarter und alternierender Richtungsindizes ergibt. Eine schematische Eins-zu-eins-Zuordnung würde daher zu einem unerwünsch-

ten Ergebnis führen.

Da die Zustände eines *Links-Rechts*-Modells aber in der Lage sind, mehrere Symbole auszugeben, bevor ein Übergang zum nächsten Zustand stattfindet, können die alternierenden Symbole dennoch einem Zustand im Referenzmodell zugeschlagen werden. Das Wissen über die Richtungsindizes, die in diesen Fällen behandelt werden, kann dann bei der Schätzung der initialen Parameter für die Matrix  $B$  der Symbolausgabewahrscheinlichkeiten nutzbringend eingesetzt werden.

Die aus den Versuchen resultierende Zustandsverteilung für die Referenzmodelle zeigt Tabelle 6.12.






Daten	Gestentyp				
	1	2	3	4	5
Trajektorie					
Zustände	5	5	6	6	8

Tabelle 6.12: Experimentell bestimmte optimale Zustandsanzahl der Referenzmodelle

Mit diesen Vorgaben werden die entsprechenden Modelle  $\lambda$  für die Referenzgesten trainiert. Da das Training mit mehrfachen Beobachtungssequenzen erfolgen muss, ist ein *Online*-Training hier nicht möglich. Verwendet wurden zwischen 7 und 14 Beispiele von zwei Personen, um Personenspezifika zu vermeiden. Lediglich bei der ersten Geste wurden ausschließlich Beispiele einer Person genutzt, um das Erkennungsverhalten für Probanden zu testen, von denen keine Beispiele vorliegen.

**Klassifikationsleistung:** Die Merkmalsreduktion durch Nachbarschaftsfilter und Beschneiden gleicher Nachfolger in den Vektorindexfolgen verändert die Repräsentation der beobachteten Verfahrbahn des Handschwerpunkts sehr stark. Abbildung 6.18 verdeutlicht die Wirkung des Nachbarschaftsfilters: die Anzahl der gespeicherten Punkte bei der Durchführung der Hakengeste nimmt stark ab.

Insgesamt lässt sich durch die Anwendung aller drei Filter eine Reduktion der Eingabefolgen je nach Geste zwischen 44% und 96% erzielen. Aus Abbildung 6.19 ist ebenfalls ersichtlich, welche Wirkung jeder einzelne Filter auf die Segmentanzahl hat. Die letzte Spalte zeigt jeweils die Reduzierung (R) der Informationen vor der Filterung auf die resultierende Menge in Prozent. Bei allen Werten handelt es sich um Mittelwerte von je 10 Beispielen. Der Grenzwert für den Nachbarschaftsfilter beträgt fünf Pixel. Abbildung 6.19 (a) zeigt die Wirkung des Nachbarschaftsfilters, wobei die Ausgangsdaten bezüglich der Start- und Endpunkthäufung bereinigt sind, also nur die reine Geste repräsentiert wird. In Abbildung 6.19 (b) kommt der Start/Stop-Filter hinzu. Die prozentuale Verbesserung ist hier erkennbar deutlicher als im vorhergehenden Fall. Abweichungen sind durch eine unruhige Hand möglich, da hierdurch die Punkte, die den Start- oder Endzustand signalisieren sollen, nicht nahe genug beieinander liegen und als Teil der Geste missverstanden werden können. Der Filterungserfolg ist dann



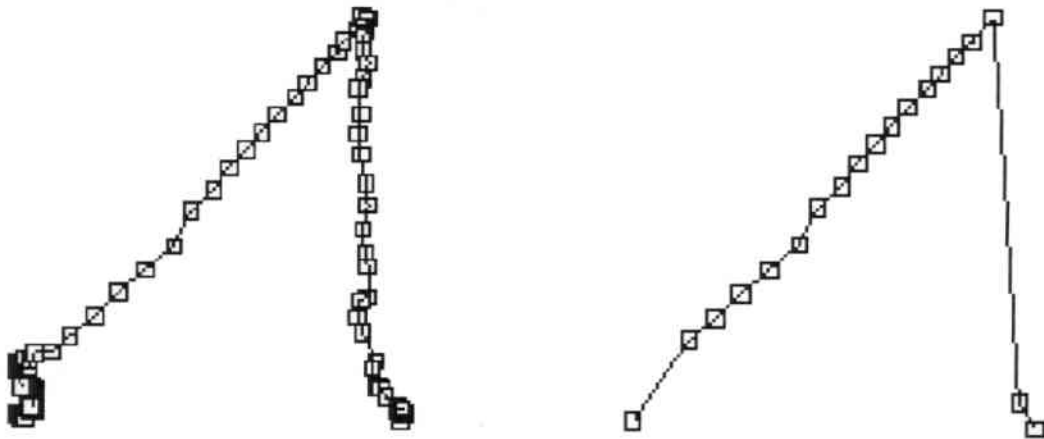


Abbildung 6.18: Aufgezeichnete und gefilterte Trajektorie (122 Segmente links, 19 Segmente rechts)

entsprechend geringer. Abschließend beschreibt Abbildung 6.19 (c) die Wirkung beim sukzessiven Einsatz aller drei Filter. Das Datenaufkommen wird dabei um 44 bis 96 Prozent reduziert. Deutlich zu erkennen sind Unterschiede in der Wirkung bei den Gesten eins bis zwei einerseits und drei bis fünf andererseits. Dies ist darauf zurückzuführen, dass es sich bei den ersten beiden um Gesten handelt, die aus Geradensegmenten zusammengesetzt sind. Deren Modelle lassen sich durch deutlich weniger Richtungsindizes beschreiben; im Idealfall sollte sogar ein einziges Symbol pro Segment ausreichend sein. Von diesem Unterschied profitiert die hierarchische Gestenklassifikation, weil hierdurch eine Einordnung in Klassen erleichtert wird. Der Unterschied zwischen den Gesten drei und vier im Vergleich zu Geste 5, die jeweils in eine andere Klasse fallen (Abbildung 5.30) ist zwar bei Betrachtung der absoluten Segmentzahl weniger deutlich, bei Auswertung der Prozentwerte aber hinreichend gut. Die Wirkung der Filter auf Beispiele der Referenzgesten findet sich graphisch dargestellt ebenfalls in Abbildung 6.20.

Beispielhaft wird in Abbildung 6.21 demonstriert, wie eine Vorführung gefiltert und das erkannte Wort auf das Modell abgebildet wird.

Ausgehend von einem Fundus an Beispielgesten muss für die hierarchische Klassifikation eine Klasseneinteilung auf Grundlage der Länge der Richtungsindexfolgen festgelegt werden. Tabelle 6.13 zeigt eine Übersicht einer solchen Menge von Beispielgesten. Für jede Geste ist die Länge der kürzesten und der längsten Richtungsindexsequenz aufgeführt, die zum Training der Referenzmodelle verwendet wurden. Abschließend ist wiederum für jede Geste das Intervall beschrieben, das auf Grundlage der Sequenzlängen gewählt wurde.

Als Intervallgrenzen kommen nicht immer die Länge der kürzesten Sequenz der Klasse und die Länge der längsten Sequenz in Frage, da sich die Intervalle dann überlappen können. Wählt man diesen Ansatz, so erhält man als Intervall für Klasse eins [3, 13] und für Klasse zwei [7, 19]. Da aber Voraussetzung für die Klassifizierung einer Geste



Geste	Segmente vorher	Segmente hinterher	R in [%]
1	64	40	38
2	55	41	25
3	75	39	48
4	65	33	49
5	77	57	26

(a) Wirkung des Nachbarschaftsfilters ohne Start- und Endzustände

Geste	Segmente vorher	Segmente hinterher	R in [%]
1	136	68	50
2	110	95	14
3	126	78	84
4	107	71	34
5	105	83	21

(b) Wirkung des Start/Stop-Filters und des Nachbarschaftsfilters mit Codierung der Start- und Endzustände

Geste	Segmente initial	Segmente nach S/T- und N-Filter	Symbole nach G-Filter	R
1	138	19	6	0.96
2	134	17	7	0.95
3	85	28	14	0.50
4	118	34	15	0.44
5	113	34	19	0.56

(c) Einsatz aller Filter, wobei S/T den Start/Stop-Filter, N den Nachbarschaftsfilter bezeichnet, G den für gleiche aufeinander folgende Symbole

Abbildung 6.19: Ergebnisse beim Filtern einer Beobachtungssequenz

die eindeutige Zuordnung zu einer Klasse ist, muss ein solches Überlappen verhindert werden. Hierbei hilft eine genauere Analyse der Sequenzen, deren Längen in Tabelle 6.13 dargestellt sind. Sind die Sequenzen in Klasse eins mehrheitlich kürzer als 12 und die in Klasse zwei meistens länger als 11, so können die Intervallgrenzen wie dargestellt festgelegt werden. Sequenzen, die in diese Klassen gehören, aber länger beziehungsweise kürzer sind und deswegen falsch zugeordnet würden, werden zur Erreichung einer eindeutigen Entscheidung in Kauf genommen.

Der durch das erkannte Endesignal ausgelöste Klassifikationsvorgang selbst kann auf mehrere Weisen umgesetzt werden. Abbildung 6.22 verdeutlicht die verschiedenen Möglichkeiten, Schwellwertmodell und Filterung mit einzubeziehen oder sogar zu kombinieren. Diese werden im Folgenden diskutiert:

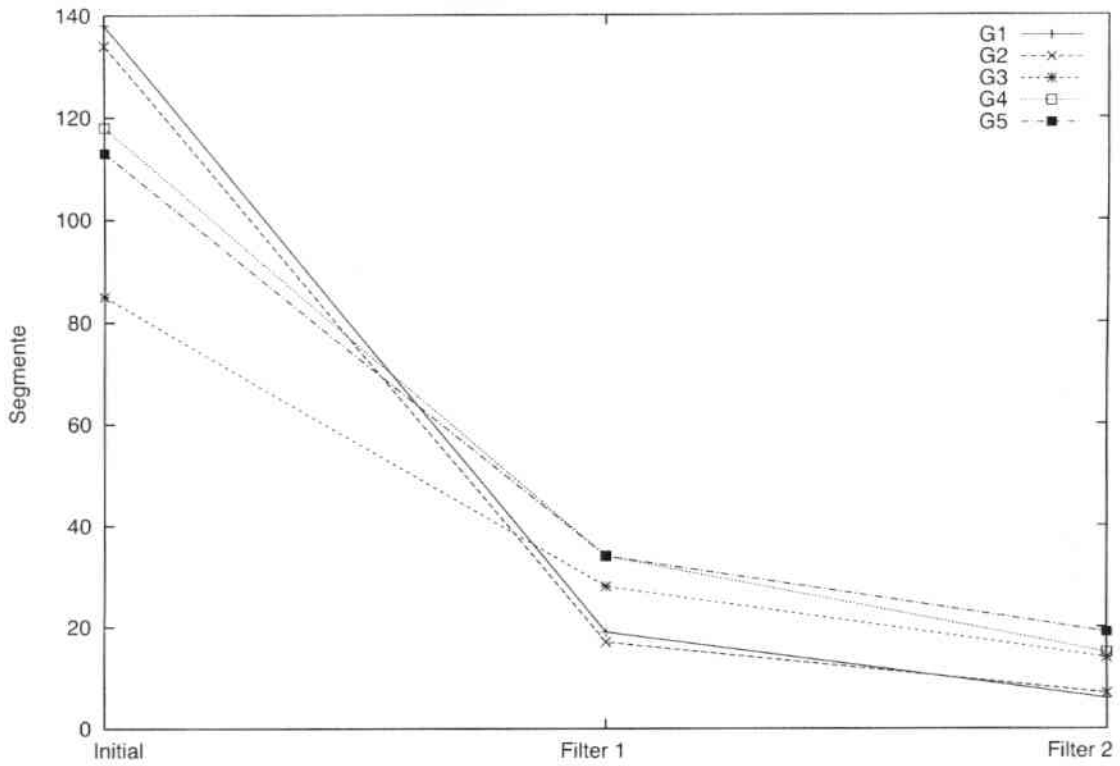
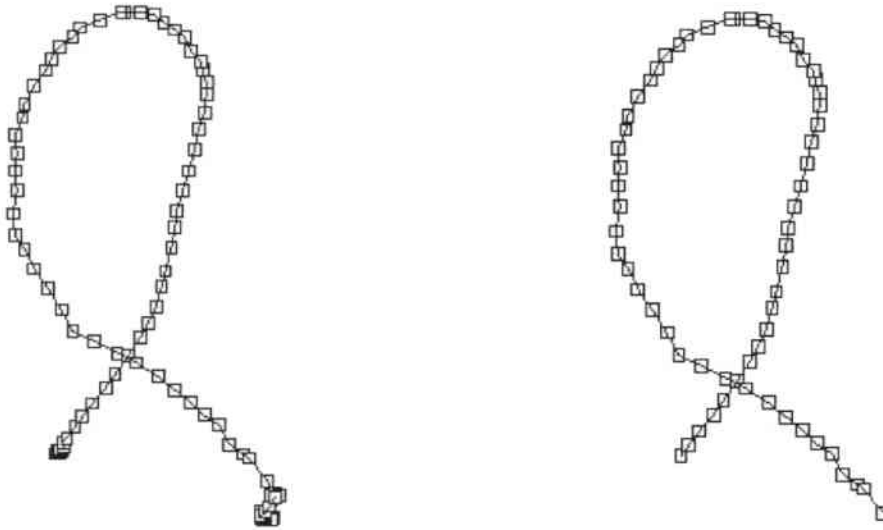


Abbildung 6.20: Wirkung der Filter auf die Anzahl der Segmente einer Bewegungsvorführung

Geste	Klasse	Min. Länge	Max. Länge	Intervall
1	1	3	13	[1,11]
2		4	12	
3	2	9	19	[12,17]
4		7	18	
5	3	16	23	[18,∞)

Tabelle 6.13: Verschiedene Sequenzlängen für die Gesten, mit deren Hilfe die Grundlage für die hierarchische Gestenklassifikation bestimmt wird

**Maximale Wahrscheinlichkeit:** Die einfachste Erkennungsvariante besteht darin, die bisherige Folge von Richtungsindizes auf alle Referenzmodelle zu testen und diejenige Geste als Treffer zu bewerten, deren Referenzmodell die höchste Wahrscheinlichkeit geliefert hat. Diese Methode hat den Vorteil, daß sie sehr tolerant gegenüber den ausgeführten Gesten ist. So können auch Gesten erkannt werden, die verglichen mit der Idealform relativ stark verzerrt oder andersartig verfälscht sind. Die durch das Referenzmodell gelieferte Erkennungswahrscheinlichkeit ist



Durch Kamera erfasste Geste  
(105 Segmente)

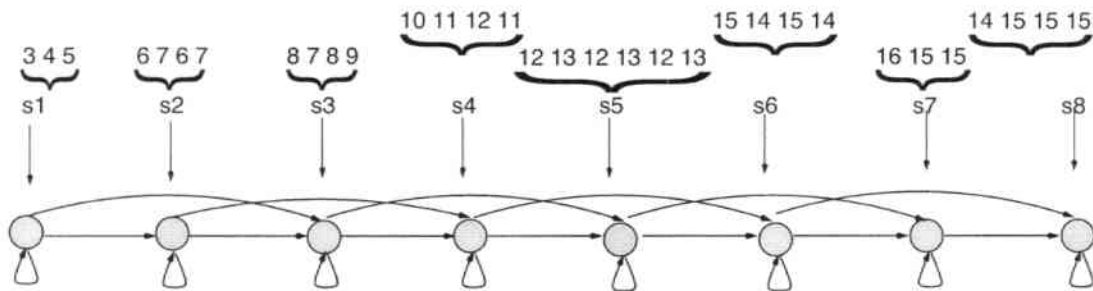
Geste nach Start/Stop-Filterung  
(63 Segmente)

3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 5  
 6 6 7 6 7 8 7 8 9 9 10 10 11 11 11 12  
 11 11 12 12 13 12 13 12 13 15 14 15  
 14 14 16 16 16 16 15 15 15 16 14 15  
 16 15

3 4 5 6 7 6 7 8 7 8 9 10 11 12 11 12  
 13 12 13 12 13 15 14 15 14 16 15 16  
 14 15 16 15

Folge von Richtungsindizes nach  
Vektorquantisierung

Folge von Richtungsindizes nach  
Filterung



Gruppierung von Richtungsindizes und Zuordnung zu Zuständen im HMM

Abbildung 6.21: Beispielablauf der Filterung und Zustandszuordnung der Beobachtung  $O$

zwar unter Umständen außerordentlich gering, aber dennoch groß genug, um aus den modellierten Gesten die Korrekte herauszufinden. Dieses Verfahren geht allerdings von der Prämisse aus, dass nur sinnvolle Gesten ausgeführt werden. Ist das nicht der Fall, so wird zwangsläufig auch eine bedeutungslose Geste als korrekt erkannt. Dieser Ansatz ist nicht in der Lage, sinnvolle von sinnlosen Gesten zu trennen.

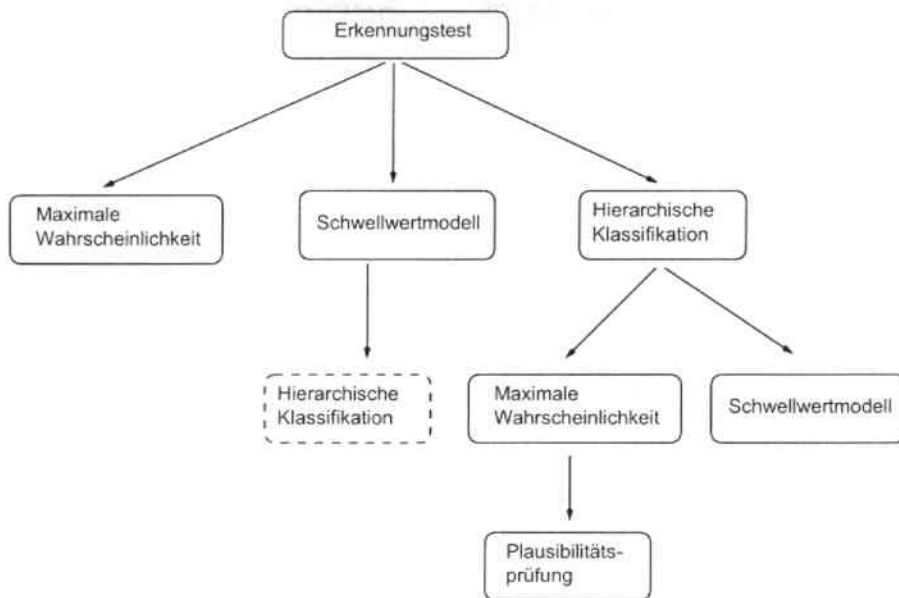


Abbildung 6.22: Testhierarchie mit möglichen Einstellungen zum Erkennungstest

**Schwellwertmodell:** Die zweite Möglichkeit, dargestellt in der Mitte der Testhierarchie in Abbildung 6.22, erlaubt das Einbeziehen des Schwellwertmodells. Auch hier wird die Richtungsindexfolge mit jedem der Referenzmodelle getestet. Darüberhinaus wird diese Folge aber auch mit dem Schwellwertmodell getestet, das durch Verschmelzung aller Referenzmodelle entsteht (Abschnitt 5.4.3). Ein Treffer wird signalisiert, falls die größte erhaltene Wahrscheinlichkeit der Referenzmodelle auch über der des Schwellwertmodells, dem Schwellwert, liegt. Durch Einsatz dieses Schwellwertes wird die irrtümliche Erkennung bedeutungsloser Handbewegungen verhindert.

**Hierarchische Klassifikation:** Die Nutzung des Schwellwertmodells läßt sich ferner mit der hierarchischen Gestenklassifikation verbinden. Allerdings ist eine solche Verbindung nur sinnvoll, wenn die hierarchische Klassifikation vor der Prüfung des Schwellwertes stattfindet. Insofern ist dieser Fall identisch mit dem in Abbildung 6.22 rechts außen veranschaulichten Fall.

Die hierarchische Gestenklassifikation ist mit beiden bereits vorgestellten Techniken kombinierbar. Die Anzahl der Richtungsindizes liefert den Anhalt für die Wahl der Klasse, mit deren Referenzmodellen die Indexfolge im Weiteren getestet wird. In der einfachen Variante wird hier die Geste des Referenzmodells ausgewählt, das die höchste Erkennungswahrscheinlichkeit innerhalb dieser Klasse aufweist. Dieser Fall entspricht dem ersten, bei dem ebenfalls die maximale Wahrscheinlichkeit einzige Grundlage für die Entscheidung ist. Diese Methode weist dementsprechend auch den Nachteil auf, dass sie bedeutungslose nicht von bedeutungstragenden Gesten trennen kann. Der Unterschied zur eingangs geschilderten Technik besteht in dem weniger aufwändigen Testen, da nicht alle Referenzmodelle mit einbezogen werden müssen. Es handelt sich bei der hierarchischen Gestenklassifikation

primär um eine Maßnahme zur Datenreduktion, die nur im Zusammenwirken mit weiteren Techniken bei der Erkennung sinnvoll einsetzbar ist.

Eine solche ergänzende Maßnahme liegt in der bereits angedeuteten Kombination mit dem Schwellwertmodell. Nach abgeschlossener Klassifikation erfolgt innerhalb der gewählten Klasse ein Testen mit allen Referenzmodellen und einem Schwellwertmodell, das durch Verschmelzung der Referenzmodelle nur dieser Klasse entstanden ist. Im Gegensatz zum globalen Schwellwertmodell, dessen Grundlage alle Referenzmodelle sind, ist ein solches lokales Schwellwertmodell auf die jeweilige Klasse beschränkt. Um eine Geste erfolgreich zu erkennen, muss auch hier die Erkennungswahrscheinlichkeit des Referenzmodells über dem Schwellwert liegen. Alternativ kann aber auch die Entscheidung nach maximaler Wahrscheinlichkeit um eine Plausibilitätsprüfung erweitert werden. Die Prüfung der Plausibilität bezieht sich allerdings auf die Klassifikation und unterstützt damit nur indirekt die Erkennung. Diese Plausibilitätsprüfung erfordert, dass das Referenzmodell mit der maximalen Erkennungswahrscheinlichkeit in dieser Klasse auch das mit der maximalen Wahrscheinlichkeit aller Modelle sein muss, um einen Treffer darzustellen. Wesentlicher Nachteil hierbei ist die Tatsache, dass zur Plausibilitätsprüfung der Test mit allen Referenzmodellen durchgeführt werden muß, also der Vorteil des ersparten Rechenaufwandes bei der hierarchischen Klassifikation zunichte gemacht wird.

Abschließend läßt sich sagen, dass, um eine zuverlässige Erkennung zu gewährleisten, das Schwellwertmodell eingesetzt werden sollte. Es kann um eine vorgeschaltete hierarchische Klassifikation erweitert werden, um den Rechenaufwand zu begrenzen. Dies wird allerdings erst bei einer steigenden Anzahl von zu erkennenden Gesten oder bei Testvorgängen zu allen diskreten Zeitpunkten interessant.

Nachfolgend werden die experimentellen Ergebnisse der in den vorhergehenden Abschnitten beschriebenden Verfahren zur Gestenerkennung wiedergegeben und erläutert. Stärken und Schwächen werden betrachtet und die Ergebnisse interpretiert. Tabelle 6.14 faßt noch einmal die Parameter für die Test- und die vorausgehenden Trainingsvorgänge zusammen.

Parameter	Eigenschaft
Anzahl der Gesten	5
Trainingsdaten pro Geste	10
Anzahl der Erkennungsverfahren	4
Anzahl Tests pro Geste und Verfahren	23

Tabelle 6.14: Parameter für das Erkennungssystem

Gemäß der in Abbildung 6.22 dargestellten Verfahren wurden Versuche mit den folgenden Erkennungsmethoden durchgeführt:

- Maximale Wahrscheinlichkeit
- Schwellwertmodell
- Hierarchische Klassifikation
- Hierarchische Klassifikation mit Schwellwertmodell

Das erste der getesteten Verfahren implementiert die Erkennung gemäß der maximalen Erkennungswahrscheinlichkeit. Als Treffer wird also diejenige Geste ausgegeben, für deren Referenzmodell  $i$  gilt:

$$P(O|\lambda_i) = \max \{P(O|\lambda_j) \mid j = 1, 2, \dots, 5\}. \quad (6.1)$$

Für dieses Verfahren ergibt sich für alle Gesten fast durchgängig eine Erkennungswahrscheinlichkeit von 100 Prozent (siehe Tabelle 6.15).

Daten	Gestentyp					
	1	2	3	4	5	Gesamt
Tests	15	15	15	15	15	75
Fehler	0	1	0	0	0	1
Erkennung	1.00	0.93	1.00	1.00	1.00	0.99

Tabelle 6.15: Erkennungserfolg der einzelnen Gesten beim Verfahren der maximalen Wahrscheinlichkeit

Diese außerordentlich guten Ergebnisse sind darauf zurückzuführen, dass die modellierten Referenzgesten einander kaum ähnlich sind. Dies führt wiederum dazu, dass sich selbst bei sehr geringen Erkennungswahrscheinlichkeiten die Testergebnisse für die einzelnen Modelle noch so deutlich unterscheiden, dass die richtige Geste zweifelsfrei identifiziert werden kann. Abbildung 6.23 zeigt Beispieltrajektorien für die dritte Geste, die alle korrekt erkannt wurden, obwohl die Erkennungswahrscheinlichkeit sogar bis zur Größenordnung von  $10^{-35}$  abnimmt. Im praktischen Einsatz ist eine Gestenerkennung mit der Methode der maximalen Wahrscheinlichkeit als einzigem Kriterium trotz der guten Ergebnisse untauglich, da bedeutungslose Gesten als solche nicht erkannt werden können.

Beim zweiten untersuchten Verfahren handelt es sich um Erkennungstests mit dem in Abschnitt 5.4.3 eingeführten Schwellwertmodell. Das aus der Verschmelzung aller Referenzmodelle entstehende Schwellwertmodell liefert einen Wert, der von der Erkennungswahrscheinlichkeit eines der Referenzmodelle übertroffen werden muss, um eine erkannte Geste zu signalisieren. Dies verhindert, daß bedeutungslose Gesten irrtümlich als erkannt gemeldet werden. Tabelle 6.16 zeigt die Ergebnisse der Tests mit dem Schwellwertmodell.



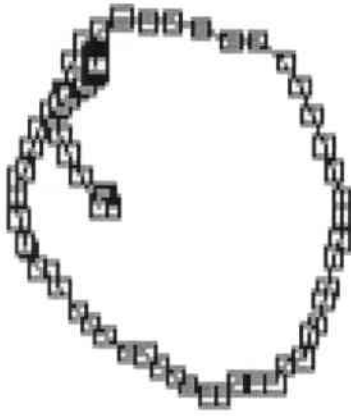
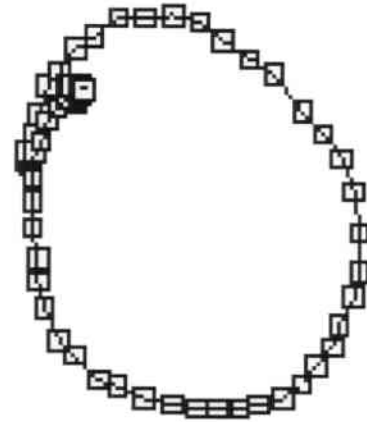
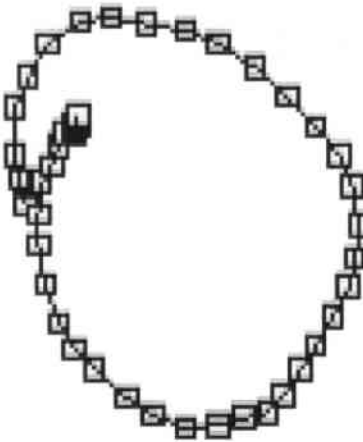
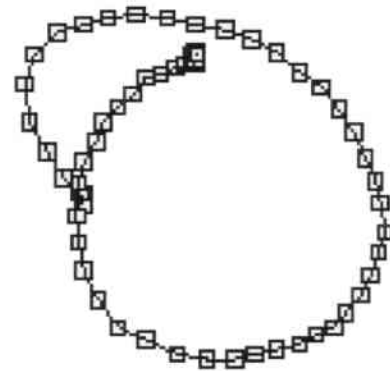
(a)  $P(\lambda|O) = 2.6845E - 21$ (b)  $P(\lambda|O) = 6.4585E - 08$ (c)  $P(\lambda|O) = 1.7190E - 35$ (d)  $P(\lambda|O) = 1.0110E - 24$ 

Abbildung 6.23: Trajektorien von erkannten Gesten des Typs 3 mit ihren jeweiligen Erkennungswahrscheinlichkeiten

Bis auf einen einzigen Fehler hätte die Methode der maximalen Wahrscheinlichkeit in allen Testfällen die korrekte Geste ermittelt. Dass das Verfahren mit Schwellwertmodell dazu in diesem Maß nicht in der Lage ist, hat mehrere Gründe. Die im Vergleich schlechteren Ergebnisse für die Gesten zwei und drei sind zum Teil auf Probleme bei der Bildverarbeitung zurückzuführen. In diesen beiden Fällen hat der Start/Stop-Filter das Ende der Gesten nicht rechtzeitig erkannt, so dass sie durch eine unnatürliche Häufung von Punkten am Ende der Trajektorie verfälscht wurden. Die Erkennungs-

Daten	Gestentyp						
	1	2	3	4	5	∅	Gesamt
Tests	30	24	25	26	24	21	150
Fehler	7	5	6	4	0	0	22
Erkennung	0.77	0.79.16	0.76	0.85	1.00	1.00	0.85

Tabelle 6.16: Erkennungserfolg der einzelnen Gesten bei Nutzung des Schwellwertmodells

wahrscheinlichkeit der betreffenden Modelle lag zwar immer höher als die der anderen, reichte aber nicht aus, um den Schwellwert zu überschreiten. Die Tatsache, dass das Referenzmodell der Geste nur mit Daten einer einzigen Person trainiert wurde, hat weiterhin Einfluss auf das weniger gute Abschneiden. Zu bedenken ist außerdem die verhältnismäßig geringe Anzahl von Gesten, mit denen die Referenzmodelle trainiert wurden (Tabelle 6.14).

Das Verfahren der hierarchischen Gestenklassifikation eignet sich zwar auch zur Gestererkennung, ist aber prädestiniert für die Datenreduktion. Bei der Auswertung der hierzu ermittelten Testergebnisse müssen die Werte für Geste eins und zwei, Geste drei und vier und Geste fünf zusammen betrachtet werden, da sie jeweils in einer gemeinsamen Klasse liegen. Die hierarchische Gestenklassifikation hat genauso wie die Methode der maximalen Wahrscheinlichkeit die Eigenschaft, immer eine Geste als erkannt zu melden. Dabei handelt es sich um die Geste mit der maximalen Wahrscheinlichkeit in der ausgewählten Klasse. Ist die Erkennung fehlerhaft, so bedeutet dies, dass der Algorithmus die falsche Klasse und folglich auch eine falsche Geste als erkannt ausgewählt hat (Tabelle 6.17).

Daten	Gestentyp						
	1	2	3	4	5	∅	Gesamt
Tests	21	25	25	25	45	19	160
Fehler	0	0	5	3	8	0	16
Erkennung	1.00	1.00	0.80	0.88	0.82	1.00	0.90

Tabelle 6.17: Erkennungserfolg der einzelnen Gesten bei Anwendung der hierarchischen Gestenklassifikation

Als problematisch bei der hierarchischen Klassifikation haben sich schnelle Handbewegungen erwiesen. Da die Auswahl der Klasse auf Grundlage der Anzahl der Richtungsindizes, die aus der Gestentrajektorie gewonnen werden, entsteht, kann eine aus wenig Punkten bestehende Trajektorie irreführend wirken. Eine schnelle Handbewegung erzeugt nun eben eine solche Trajektorie aus wenig Punkten. Die daraus resultierende ebenfalls sehr kurze Folge von Richtungsindizes deutet dann auf eine Geste in der

ersten Klasse hin. Die Experimente haben gezeigt, dass alle Fehlentscheidungen für die Gesten drei und vier in einem Sprung in Klasse eins, nicht aber in Klasse drei bestanden. Gleiches gilt auch für Geste fünf.

Unproblematisch hingegen sind langsam ausgeführte Gesten, die immer in einer langen Richtungsindexfolge resultieren. Dies und eine damit einhergehende Entscheidung für eine falsche Klasse mit Gesten aus langen Indexfolgen wird durch den Einsatz der Filter verhindert. Sowohl der Pixelfilter als auch der Gleichheitsfilter verhindern eine solche Fehlentscheidung.

Die Intervallgrößen, die festlegen, welche Längen der Richtungsindexfolgen zu welcher Klasse führen, sind experimentell bestimmt und, da sie Mittelwerte darstellen, zwangsläufig ungenau. Weil sich die Intervalle nicht überlappen dürfen, um eine eindeutige Klassifikation zu ermöglichen, muß man einige wenige Fehleinschätzungen in Kauf nehmen. Dem läßt sich allerdings entgegenwirken, indem nur Gesten gewählt werden, die so unterschiedlich sind, dass sie auch eine signifikant andere Anzahl von Richtungsindizes bei der Quantisierung hervorbringen.

Bei dem letzten eingesetzten Verfahren handelt es sich um eine Kombination aus hierarchischer Klassifikation und Schwellwertmodell. Während die Anzahl der Richtungsindizes, die die Geste ausmachen, die Wahl der Klasse vorgibt, fällt die Entscheidung über ein Erkennen innerhalb der Klasse mit Hilfe des Schwellwertmodells. Die Testergebnisse sind in Tabelle 6.18 zusammengefaßt.

Daten	Gestentyp						
	1	2	3	4	5	∅	Gesamt
Tests	40	29	25	15	15	26	150
Fehler	13	5	4	4	4	0	30
Erkennung	0.68	0.83	0.84	0.73	0.73	1.00	0.80

Tabelle 6.18: Erkennungserfolg der einzelnen Gesten bei Anwendung der hierarchischen Gestenklassifikation kombiniert mit dem Schwellwertmodell

Die im Vergleich zu den anderen Verfahren spürbar schlechteren Ergebnisse sind durch die doppelten Anforderungen und dadurch auch zweifache Fehleranfälligkeit bedingt. Um eine Geste korrekt erkennen zu können, muss nicht nur die Entscheidung für die richtige Klasse fallen, auch muss die Erkennungswahrscheinlichkeit über dem Schwellwert liegen. Ist eine dieser beiden Voraussetzungen nicht erfüllt, so scheidet die Erkennung.

Die Fehler lassen sich also zum Einen auf die Wahl der falschen Klasse zurückführen und zum anderen auf eine Ablehnung durch das Schwellwertmodell. Insbesondere das schlechte Resultat für Geste eins hat mit dem Schwellwertmodell zu tun, ist aber wesentlich durch die Tatsache bedingt, dass das zugehörige Referenzmodell nur mit Gesten eines Nutzers trainiert wurde. Eine Verbesserung der Erkennungsrate kann erzielt

werden, indem die Intervalle, die der Wahl der Klasse zugrunde liegen, angepasst werden. Das Training des Referenzmodells von Geste eins hebt sich von dem der anderen dadurch ab, dass die Trainingsgesten ausschließlich von einer einzigen Person stammen. Getestet wurde jedoch mit zwei Personen. Dabei hat sich gezeigt, dass die Gesten der Person, die auch das Referenzmodell trainiert hat, besser erkannt werden, als die der unbekannteren. Tabelle 6.19 schlüsselt die Testergebnisse nach Testperson und Verfahren auf. Person eins hat die Trainingsdaten für das Referenzmodell geliefert, wohingegen Person zwei nur zum Testen zur Verfügung stand.

	Schwellwertmodell		Hierarchische Klassifikation & Schwellwertmodell	
	Person 1	Person 2	Person 1	Person 2
Geste 1	0.87	0.67	0.75	0.60

Tabelle 6.19: Unterschiedlicher Erkennungserfolg in Abhängigkeit davon, ob auch mit Gesten der Testperson trainiert wurde

Der Erkennungserfolg beim Verfahren der maximalen Wahrscheinlichkeit wurde durch Testen mit einer am Training unbeteiligten Person nicht beeinträchtigt. Obwohl die Gesten fast vollständig korrekt erkannt wurden, haben diese Tests aber auch gezeigt, dass die Erkennungswahrscheinlichkeiten für die Person, die auch Trainingsdaten zur Verfügung gestellt hat, im Mittel höher waren. Für die hierarchische Gestenklassifikation gilt das Gleiche.

Zusammenfassend läßt sich sagen, dass die Referenzmodelle auf einzelne Personen zugeschnitten werden können, indem sie nur mit Gesten dieser einen Person trainiert werden. Sie können dann auch noch Gesten anderer Personen erkennen, allerdings zu einem weniger großen Anteil, mindestens aber mit einer weniger großen Wahrscheinlichkeit. Auf der anderen Seite lassen sich die Modelle mittels Training von Gesten, die von möglichst vielen verschiedenen Probanden stammen, gut und allgemein auf die spätere Erkennungsaufgabe vorbereiten. Die Erkennungswahrscheinlichkeiten werden dann im Mittel geringer ausfallen, als wenn nur mit einer Person trainiert wird. Dies hat allerdings keinerlei negative Konsequenzen, da in diesem Fall auch der Schwellwert, den es zu überschreiten gilt, kleiner wird. Sollen also Gesten von mehreren Personen erkannt werden, so ist ein Training mit Daten aller dieser Personen empfehlenswert.

## 6.2 Validierung der Handlungsbeobachtung

Zur Validierung der Handlungsbeobachtung dienen zwei Experimente, die das Zusammenspiel der elementaren kognitiven Operatoren in der Vorführ- und in der Ausführungsumgebung testen. Sie werden im Folgenden diskutiert.

### 6.2.1 Handlungsbeobachtung in der Vorführungsumgebung

Die Leistungsfähigkeit der vorgestellten elementaren kognitiven Operatoren zur Erkennung performativer Handlungselemente wird anhand einer konkreten Handhabungsaufgabe untersucht. Dazu wurde als Problemstellung das Tischdecken ausgewählt. Das zu platzierende Geschirr befindet sich zu Beginn in Griffweite des Demonstrierenden. Im Laufe der Vorführung werden eine Schüssel und eine Tasse mit Untertasse an passende Positionen auf einer Unterlegmatte verteilt.

Nach der Vorführung fand entsprechend Abschnitt 4.1 eine halbautomatische Analyse der aufgezeichneten Aktionen statt. Die verwendeten Griffe und Verfahrbahnen wurden in der Simulation dem Benutzer zur Überprüfung vorgespielt. Anschließend wurde die Ausführung auf den modellierten Roboter abgebildet und mit der entsprechenden kinematischen Änderung simulativ wiederholt. Danach konnte das erzeugte Programm auf dem realen Roboter zur Ausführung gebracht werden.

Die Abbildungen 6.24ff zeigen ausschnittsweise die Vorführung des Tischdeckens, die Wiederholung in der modellierten Szene, die Abbildung auf den Roboter und dessen Ausführung derselben Aufgabe. Zum Vergleich der vier Phasen finden sich die zu einem Zeitpunkt gehörenden Bilder untereinander geordnet. Die Ansichten der aus der Vorführung gewonnenen Repräsentation des Tischdeckens sind mit Hilfe des KAVIS-Visualisierers erzeugt worden. Sie zeigen das Umweltmodell mit der Benutzerhand und deren Trajektorie. Es wird deutlich, dass die Ausführung unter Verwendung der präsentierten Griffe und Trajektorien die Vorführung sehr gut reproduziert. Da die Anfahrttrajektorien vor Griffen mit Bezug auf die jeweiligen Objekte gespeichert werden, ist dies sogar möglich, wenn die Anordnung der zu manipulierenden Gegenstände in der Szene variiert wird. Da der ausführende Roboter *Albert* mit seiner Dreifingerhand über eine eingeschränkte Beweglichkeit verfügt, werden die Objekte mit Kraftgriffen aufgenommen. Diese erlauben dem System das stabile Aufnehmen der verwendeten Objekte (Zeitpunkte d, k und q in der Abbildung).

### 6.2.2 Handlungsbeobachtung in der Ausführungsumgebung

Ein weiteres Experiment dient zur Überprüfung der Interaktion mit einem Robotersystem. Das Robotersystem *Albert* soll dazu angewiesen werden, Objekte zu klassifizieren. Dazu wird die Blickrichtung des Systems durch eine Zeigegeste auf den vor ihm stehenden Tisch gelenkt. In Abbildung 6.29 finden sich die einzelnen Schritte des Vorgehens. Die Roboterperspektive ist hier unter der Bildsequenz aufgeführt, welche die Handlung darstellt. Die Vorführung beginnt mit einer Geste vom Typ 6. Daraufhin folgt der Kamerakopf den Handbewegungen. Zum Zeitpunkt d wird die Zeigegeste erkannt, die zur Auslösung der Klassifikation dient.

## 6.3 Zusammenfassung

Die Untersuchung der einzelnen kognitiven Operatoren hat für beide Umgebungen, in denen sie eingesetzt werden, zufriedenstellende Ergebnisse gezeigt. Die Erkennung von Griffen und Gesten ist sicher und robust möglich. Die Fusion der magnetfeldbasiert gewonnenen Positionsdaten mit denen der Bildverarbeitung weisen ebenfalls die geforderte Genauigkeit auf. Zusammen mit den Ergebnissen, die bei der Objekterkennung erreicht werden, zeigen sich die Operatoren damit als sehr gut geeignet, zur Handlungsbeobachtung verwendet zu werden.

Die implementierte Vorführungsumgebung kann damit wie auch das Beobachtungssystem auf dem Roboter *Albert* zur Beobachtung von Benutzerdemonstrationen genutzt werden. Die vorgeschlagene Systemarchitektur, die zum Ablauf und zur Koordination der Operatoren dient, konnte ihre Leistungsfähigkeit in zwei entsprechenden Experimenten unter Beweis stellen.



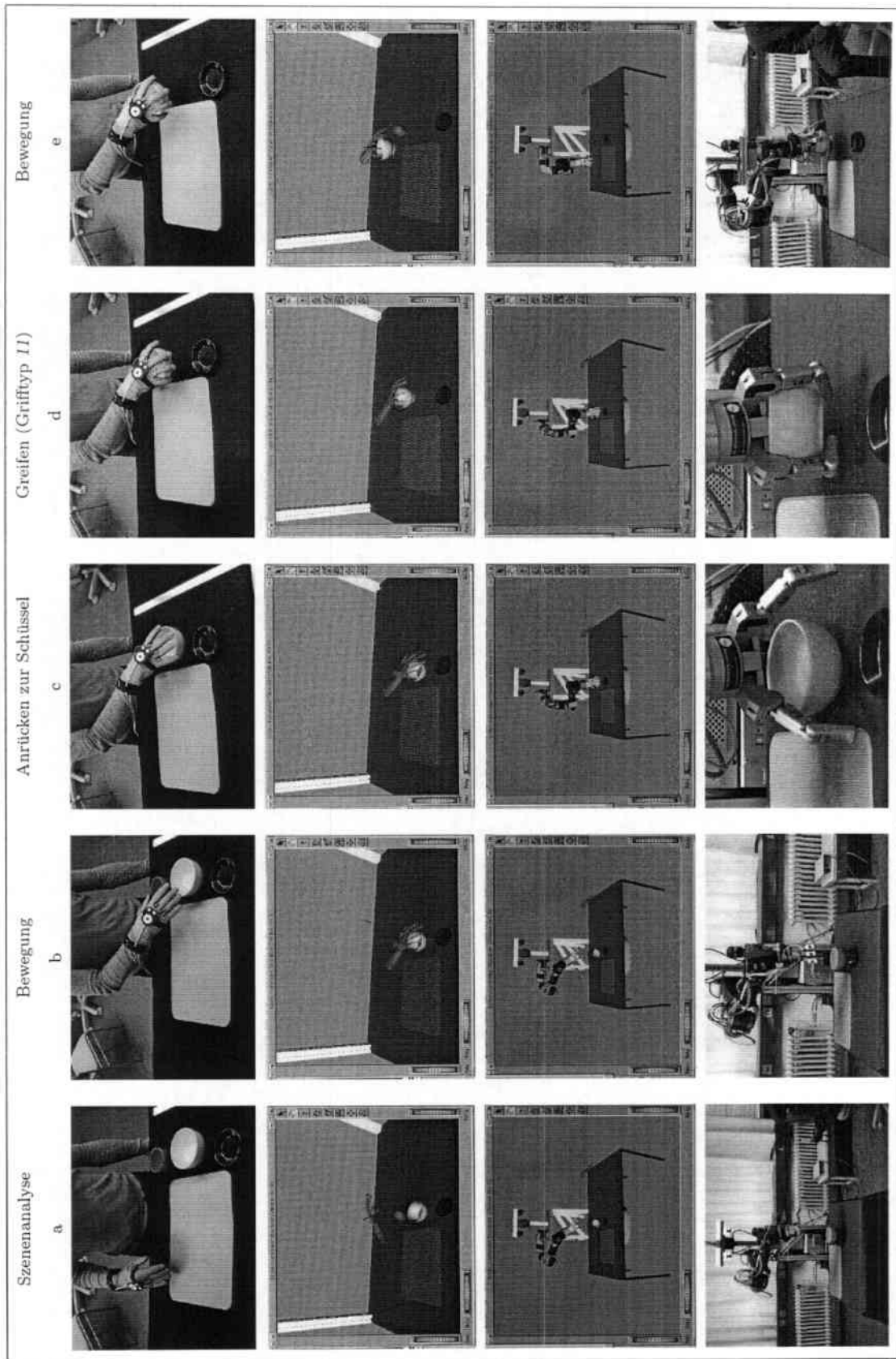


Abbildung 6.24: Experiment zur Validierung der Handlungsbeobachtung (Teil 1, von oben nach unten: Vorführung, Simulation der aufgezeichneten Vorführung, Ausführung der abgebildeten Vorführung und reale Ausführung der Handhabungsaufgabe)

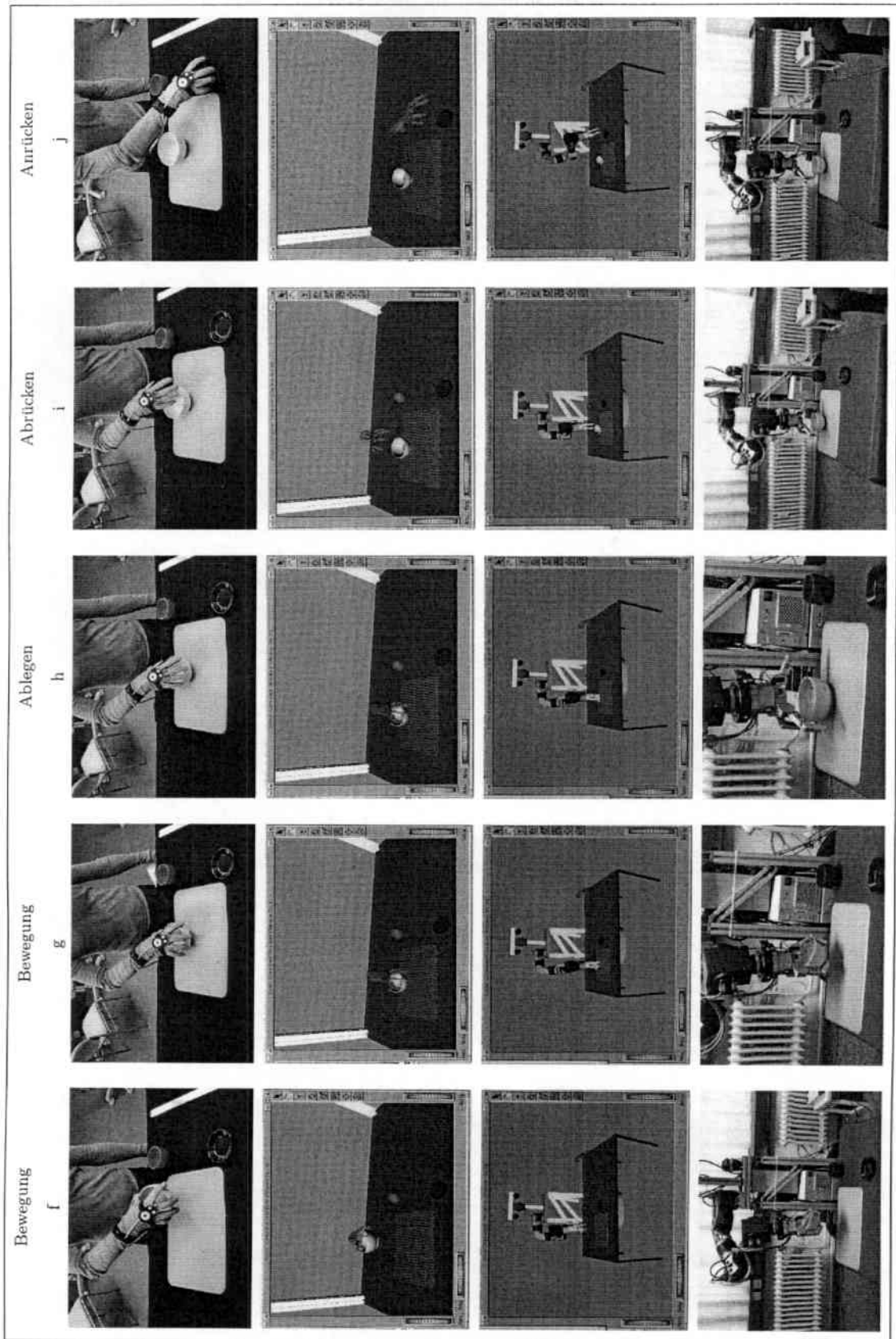


Abbildung 6.25: Experiment zur Validierung der Handlungsbeobachtung (Teil 2)

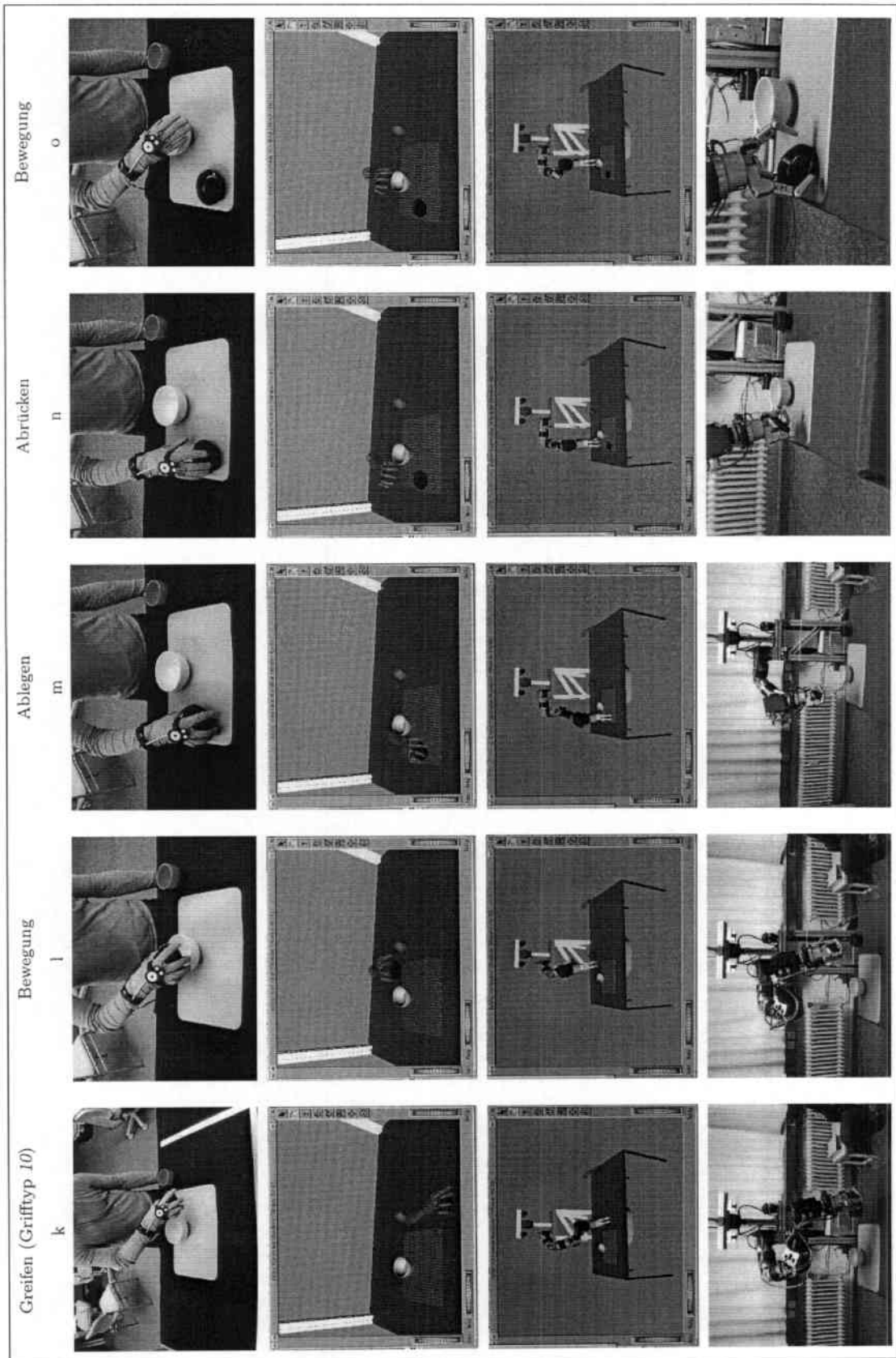


Abbildung 6.26: Experiment zur Validierung der Handlungsbeobachtung (Teil 3)

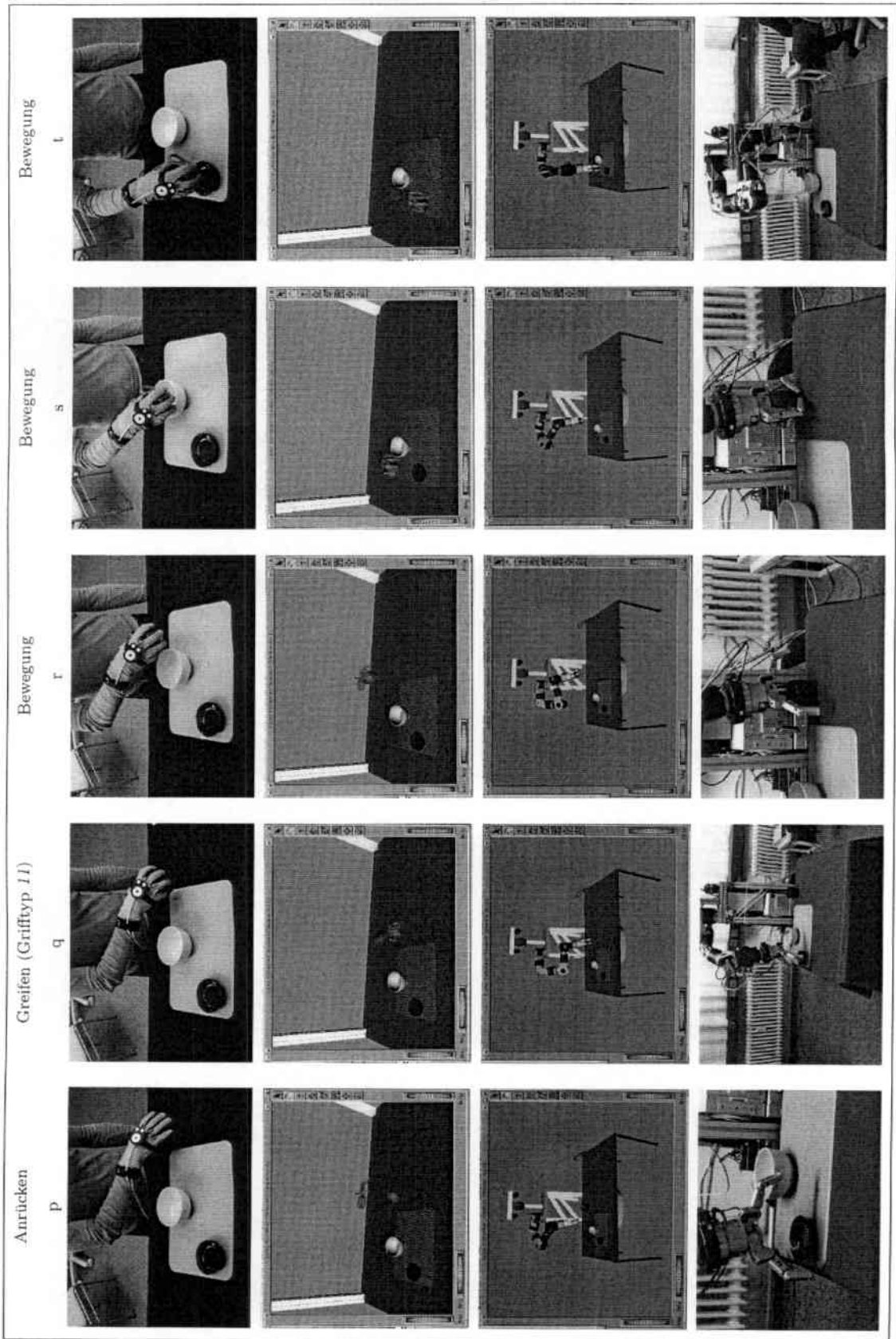


Abbildung 6.27: Experiment zur Validierung der Handlungsbeobachtung (Teil 4)





Abbildung 6.28: Experiment zur Validierung der Handlungsbeobachtung (Teil 5)

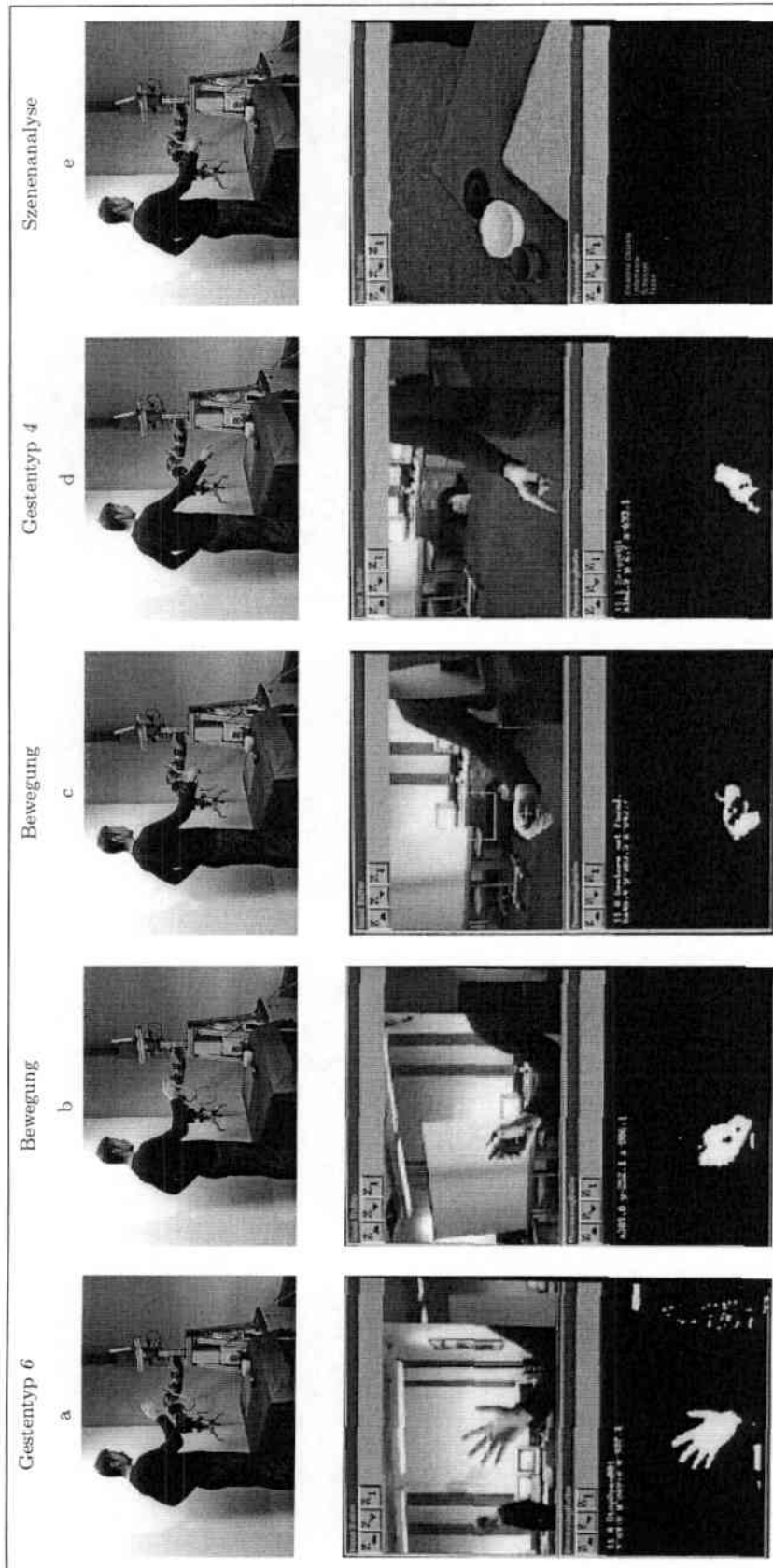


Abbildung 6.29: Experiment in der Ausführungsumgebung



# Kapitel 7

## Schlußbetrachtung

In der vorliegenden Arbeit werden neue Methoden zur Programmierung von Robotern auf Basis beobachteter Vorführungen untersucht. Die dabei gewonnenen Erkenntnisse werden in diesem Kapitel zusammengefasst und mit den erzielten Ergebnissen diskutiert und bewertet. Den Schluss des Kapitels bildet ein Ausblick auf zukünftige Ergänzungen und Erweiterungsmöglichkeiten.

### 7.1 Zusammenfassung der erzielten Ergebnisse und Erkenntnisse

Ausgangspunkt dieser Arbeit war die Feststellung, dass das Programmieren von Robotern und die Schnittstellen zur Interaktion mit Robotern auf der Basis von teach-in Verfahren oder textuellen Programmiersprachen nicht benutzerfreundlich gestaltet sind. Dies schlägt sich nicht zuletzt in hohen Programmierkosten nieder.

Es wurde deshalb ein System für die Abbildung von Benutzervorführungen in eine Repräsentation entwickelt, die einem Robotersystem die Beobachtung von Handlungen zum Zweck der Interaktion oder zum Programmieren durch Vormachen gestattet. Solche Verfahren sind deshalb von besonderer Bedeutung, da der zukünftige Einsatz von Robotern als Assistenz- oder Servicesystemen auch ohne spezielle Schulung Menschen deren Anwendung ermöglichen soll. Die Benutzer sollen durch einfache Interaktionsmechanismen befähigt werden, Programme für Manipulationsaufgaben zu erzeugen oder deren Ausführung anzustoßen, um Roboter auch in nichtindustriellen Umgebungen und in flexiblen Fertigungsumgebungen einsetzen zu können. Das Programmieren durch Vormachen ist eine für diesen Zweck geeignete Methode.

Die Analyse von Vorführungen, wie sie Menschen gegenüber Menschen ausführen, zeigte, dass existierende Ansätze nur einen kleinen Ausschnitt der Aspekte komplexer Handlungsfolgen herausgreifen und beobachten. Selten enthalten Vorführungen jedoch ausschließlich performative Elemente, meist werden sie durch Kommandos während der Ausführung unterstützt. Die Interpretation performativer Handlungen im Hinblick auf den flexiblen Transfer der Vorführung auf einen Manipulator ist ohne Kommentierung schlechthin gar

nicht möglich: das System könnte im Beispiel des Tischdeckens nicht entscheiden, ob das Messer in Relation zum Teller oder zur Untertasse zu liegen kommen muss.

Dieser Beobachtung folgend wurden Operatoren definiert, deren Aufgabe die Beobachtung eines elementaren Handlungssegments ist, sei es ein performatives, kommandierendes oder kommentierendes. Jeder Operator erfasst handlungsspezifische relevante Informationen und registriert diese in einem Umweltmodell. Das entwickelte Handlungsmodell spricht die einzelnen Operatoren integriert an und löst sie aus. Zudem legt es die Rahmenbedingungen für deren zeitliche Abfolge bei der Beobachtung fest.

Die methodische Realisierung der Operatoren ist abhängig von den verwendeten Sensortypen. Zur positionsgenauen Verfolgung von Handlungen wurde deshalb eine separate Vorführungsumgebung geschaffen, in der mit Hilfe eines Datenhandschuhs, eines magnetfeldbasierten Verfolgungssystems und eines aktiven Kamerakopfes Operationen im Rahmen des Programmierens durch Vormachen untersucht werden konnten. Zur Interaktion und Auslösung der dort gewonnenen Programme kann auf einem Roboter der dort vorhandene Kamerakopf eingesetzt werden. Die Grundlagen der kognitiven Operatoren sind im Einzelnen:

- Verschiedene Ansätze zur Objekterkennung basierend auf Farbhistogrammen oder Ansichten sowie die Lokalisierung der erkannten Objekte,
- Verfahren zur Verfolgung von Bewegungen mittels der Fusionierung von Messungen eines magnetfeldbasierten Trackingsystems und markerbasierten bildverarbeitenden Methoden oder mittels adaptiver Hautfarbsegmentierung,
- Neuronale Netze zur Erkennung von Griffen, die als Eingabe Messwerte des Datenhandschuhs erhalten,
- Neuronale Netze zur Erkennung statischer Gesten,
- Die bildbasierte Klassifikation statischer Gesten anhand der Handkontur,
- Die bildbasierte Klassifikation dynamischer Gesten anhand von Bewegungsmustern.

Zur Evaluierung der Stabilität der Handlungsverfolgung wurden zahlreiche Experimente durchgeführt. Dabei wurden sowohl die einzelnen Operatoren als auch das gesamte Beobachtungssystem in den beiden Ausprägungen in der Ausführ- und in der Vorführungsumgebung getestet. Hierbei konnte gezeigt werden, dass aus den bei der Beobachtung generierten Datenstrukturen Roboterprogramme abgeleitet werden konnten und das System interaktiv auf Kommandos reagieren kann. Die einfache Instruktion komplexer Manipulationsaufgaben und ihre Kommandierung wird damit ermöglicht.

## 7.2 Diskussion

Damit die in dieser Arbeit vorgestellte Beobachtungsmethodik erfolgreich zur Roboterinstruktion angewendet werden kann, müssen einige Voraussetzungen gegeben sein:

1. Das Programmiersystem muss für die erfolgreiche Programmierung in der Vorführungsumgebung überhaupt in der Lage sein, die relevanten Informationen zu erfassen. Vorführungen von Schiebe-, Füge- oder Balancierhandlungen sowie zweihändige Vorführungen werden beispielsweise nicht adäquat beobachtet. Da in das System auch keine kraftaufnehmenden Sensoren integriert sind, ist bislang noch nicht die Ableitung kraft geregelter Operationen möglich. Dieses Problem wird derzeit durch objektspezifisches Hintergrundwissen gelöst.
2. Der Benutzer muss in der Lage sein, die zu programmierende Manipulationsaufgabe kompetent in seiner Vorführung zu spezifizieren. Da die Bewegungsbahnen direkt auf dem ausführenden System nutzbar sein sollen und daher vollständig im Umweltmodell registriert werden, ist darauf zu achten, Bewegungen mit Rücksicht auf eventuelle Manipulatorrestriktionen zu zeigen. Dies trifft ebenso auf die verwendeten Greifoperationen zu.
3. Die Vereinbarung zur gezielten Auswahl von Roboterfunktionen über Gesten muss dem Benutzer bekannt sein. Im Rahmen der Interaktion muss er ständig über die Bedeutung seiner Vorführung orientiert sein. Zwar sind viele symbolische Gesten wie das Stoppzeichen oder deiktische Referenzen allgemeinverständlich, im Rahmen spezieller Operationen kann die Interpretation jedoch nicht eindeutig erfolgen. So ist es beispielsweise möglich, eine Zeigegeste als Zielrichtungsangabe oder als Objektauswahl zu verstehen.
4. Im Falle der Ausführungsumgebung muss der Benutzer auf Schwankungen der Lichtverhältnisse und Verdeckungen Rücksicht nehmen. Da die Handverfolgung ausschließlich auf der adaptiven Hautfarbsegmentierung basiert, treten bei plötzlichen Änderungen der Hintergrundbeleuchtung kamerabedingt spektrale Verschiebungen auf, welche eine robuste Verfolgung unmöglich machen. Eine stabile Verfolgung allein auf Basis der Segmentierung nach der Hautfarbe ist sehr schwer zu realisieren.

Sind diese Bedingungen erfüllt, können mit Hilfe der vorgestellten Methode und unter Anwendung der entwickelten Algorithmen Roboter instruiert, d.h. Handlungsfolgen durch Vormachen spezifiziert und Kommandos durch Gesten gegeben werden.

Die vorgestellte Methodik ermöglicht die Programmierung von Manipulationsaufgaben und Interaktion auch solchen Benutzern, die nur über eingeschränkte Programmierkenntnisse verfügen. Die klassische Programmierung, in der allein der Benutzer aktiv ist, wird durch einen Dialog zwischen Benutzer und Programmiersystem ersetzt. Über den Stand der Technik geht diese Arbeit durch die gezielte Fusionierung spezieller Sensorik hinaus: die Kombination von Datenhandschuh, magnetfeldbasiertem Verfolgungssystem und aktivem Kamerakopf in der Vorführungsumgebung hat sich für die Aufnahme von Demonstrationen als sehr gut geeignet erwiesen. Neben Fortschritten bei der Verbesserung der in den kognitiven Operatoren verwendeten Verfahren (Fusion von Methoden zur Objektdetektion, Fusion von magnetfeld- und bildbasierten Messdaten, Nutzung von Objektwissen zur Grifferkennung, Einsatz von Fourierdeskriptoren und neuronalen Netzen zur Gestenerkennung und Entwurf eines hierarchi-

schen Verfahrens zur Klassifikation dynamischer Gesten) ist das systemische Zusammenspiel dieser Detektoren in einem verteilten System neuartig.

### 7.3 Ausblick

Der in dieser Arbeit entwickelte Ansatz zur Beobachtung von Benutzerhandlungen stellt eine allgemeine Lösung für das Problem der Interaktion mit Assistenzsystemen und deren Programmierung dar. Diese Lösung kann auf vielfältige Weise erweitert und an spezielle Verhältnisse angepaßt werden. Komplett oder in Teilen kann sie auch in anderen Domänen als der Roboterinteraktion im Speziellen oder der Robotik im Allgemeinen eingesetzt werden. So ist zum Beispiel die Überwachung handhabender Tätigkeiten im Bereich der Mensch-Roboter-Kooperation zur Situationserkennung denkbar oder die Interaktion mit anderen Systemen als Assistenten. Das entwickelte Verfahren ist daher als Prototyp eines allgemeinen Entwicklungsmodells für die benutzer- und aufgabenorientierte interaktive Programmierung anzusehen.

Neben der Anpassung an andere potentielle Einsatzgebiete bestehen sowohl in methodischer als auch in praktischer Hinsicht Möglichkeiten, den vorgestellten Ansatz und seine Komponenten weiterzuentwickeln. Einige der Erweiterungsmöglichkeiten sind oben bereits angeklungen. Als besonders interessant stellen sich die folgenden dar:

- Eines der grundlegenden Probleme im Rahmen des Programmierens durch Vormachen stellt die Interpretation der vorgeführten Handhabungen dar. Wie in Kapitel 3 beobachtet, erklären Menschen ihre Handlungen durch kontinuierliche parallele verbale Beschreibungen. Es würde die Auswahl der wesentlichen Änderungen in den relationalen Deskriptoren über dem Weltmodell wesentlich verbessern, wenn eine solche sprachliche Kommentierung zusätzlich zu den gestischen Handlungen nutzbar würde. Außerdem würde die Generierung von Systemhypothesen über erfolgte Handlungen sicherer werden.
- Die Interpretation der Vorführung findet momentan im Wesentlichen nach der Vorführung in einem separaten Schritt statt. Wenn Teile dieser Interpretation vorverlegt würden und bereits während der Vorführung die Korrektheit von Systemhypothesen durch Rückfragen bestätigt würden, könnte der Prozess der Datenaufnahme eine tiefgreifende Beschleunigung erfahren. Der Benutzer wäre dann nicht mehr gefordert, sich in einem zweiten Schritt die Aufnahme seiner Aktionen bestätigen zu lassen und Systemhypothesen zu überprüfen, sondern könnte sie gleich an Ort und Stelle bestätigen bzw. falsifizieren und seine Vorführung gegebenenfalls erneut durchführen.
- Ein breiteres Spektrum von Handhabungsaufgaben lässt sich durch die Erweiterung der Anzahl elementarer kognitiver Operatoren abdecken. So können Operatoren für das Schieben mit Kontakt, für Fügeoperationen oder spezielle Detektoren für das Drücken von Knöpfen oder Öffnen von Türen und Schubladen bereitgestellt werden. Dazu ist lediglich der Operator zu spezifizieren und zu implementieren und ein zugehöriges Ereignis vorzusehen, das im Weltmodell registriert werden kann. Es wäre ein vielversprechendes Forschungsziel, eine Kopplung des in dieser Arbeit entwickelten Prozesses und

der entworfenen Algorithmen mit Methoden aus dem Bereich der Akquisition von Elementarfähigkeiten zu erreichen [Kaiser 96]. Ein integrierter Prozess, bei dem bei Bedarf noch nicht vorhandene, aber benötigte Elementarfähigkeiten und ihre entsprechenden kognitiven Operatoren in derselben Vorführungsumgebung interaktiv programmiert werden, wäre dabei besonders wünschenswert. Diese neuen Fähigkeiten könnten dann direkt im Rahmen der Programmierung umfangreicherer Aufgaben Verwendung finden.

- Da Menschen Aufgaben meist unter Nutzung beider Hände ausführen, ist das Einsatzgebiet der vorgestellten Verfahren durch Beobachtung beider Hände wesentlich erweiterbar. Die Operatoren für die Griffe und Gesten selbst sind dazu relativ einfach durch eine gespiegelte Kopie für die zusätzliche Hand zu erweitern, es wären jedoch für komplexere Handlungen spezielle kognitive Fähigkeiten für die Funktionsbestimmung der dominanten und zweiten Hand notwendig.
- Die Qualität und der Komfort der Handlungserkennung wird ebenso wesentlich durch den Einsatz von kraftaufnehmenden Sensoren für die Griffbehandlung erweitert. Dadurch würde nicht nur weitere Information aus der Vorführung nutzbar, sondern auch die Erkennung dynamischer Griffe einfacher. Dazu existieren bereits vielversprechende Ansätze [Zöllner 01].
- Die Erweiterung der Beobachtung in der Ausführungsumgebung durch zusätzliche Merkmale wie den optischen Fluss oder Tiefeninformation könnte die Handverfolgung robuster machen und außerdem die Erkennung dreidimensionaler Gesten ermöglichen. Dies würde auch bestimmte Interaktionsmuster wie die Richtungsangabe wesentlich vereinfachen. Auch auf diesem Gebiet gibt es bereits erfolgreiche Untersuchungen [Nickel 03].
- Menschen artikulieren sich nicht nur verbal oder gestikulierend. Deshalb ist auch die Erkennung mimischer Aspekte oder der Blickrichtung zur Aufmerksamkeitsverfolgung als Merkmal für die Interaktion zwischen Mensch und Maschine nutzbar. Es wäre für Benutzer ebenso intuitiv verständlich, wenn die Maschine ihrerseits Zuständen durch mimische Verzerrungen, Kopfbewegungen oder Blickrichtungsänderungen Ausdruck verleiht.

Alle diese Erweiterungen dienen dazu, den Wissenstransfer für Mensch und Maschine noch einfacher und effizienter zu gestalten und den Methoden durch Weiterentwicklung neue Anwendungsgebiete zu eröffnen. Bereits jetzt ist mit dem entwickelten Beobachtungssystem ein Fortschritt in Richtung der Programmierung technischer Systeme und der Interaktion mit ihnen durch Endbenutzer erreicht worden.



# Anhang A

## Definitionen

Der folgende Abschnitt dient der genauen Definition von Begriffen, die im Text verwendet werden.

**Definition A.1 Weltmodell:** Das Weltmodell  $w$  ist die Vereinigung der Beschreibungen aller in einer Szene enthaltenen Objekte und ihrer Eigenschaften in der Zeit.  $w$  ist damit ein temporallogischer Ausdruck, der sich aus der konjunktiven Verknüpfung einzelner Objekte, Eigenschaften und Ereignissen zusammensetzt:

$$w = E_1 \wedge E_2 \wedge \dots \wedge E_n \quad (\text{A.1})$$

**Definition A.2 Ereignis:** Ein Ereignis  $e$  ist ein Paar  $(g, t)$ , das ein Geschehnis  $g$  beschreibt. Bestandteil von  $e$  ist der Zeitpunkt  $t$ , zu dem die Beobachtung von  $g$  gemacht wurde. Ereignisse können im Rahmen der vorliegenden Arbeit insbesondere von folgender Qualität sein:

- Die Beschreibung des Vorkommnis' eines Objekts oder der Lageveränderung eines Objekts.
- Die Beschreibung einer Positionsänderung der Benutzerhand.
- Die Beschreibung einer ausgeführten Geste.
- Die Beschreibung eines ausgeführten Grifftyps.

**Definition A.3 Beobachtung:** Eine Beobachtung eines Sensorsystems ist die Generierung eines Ereignisses  $e = (g, t)$  zum Zeitpunkt  $t$ . Die Beobachtung ist nur dann korrekt, wenn  $g$  einem Geschehnis oder einem Objekt in der Szene entspricht.

**Definition A.4 Registrierung:** Die Registrierung eines Ereignisses  $e$  ist die Aufnahme von  $e$  in das Weltmodell  $w$ .

**Definition A.5 Objektlage:** Die Lage eines Objektes  $o$  ist beschrieben durch das 6-Tupel  $(x, y, z, \alpha, \beta, \gamma)$ . Neben translatorischen Positionsangaben sind rotatorische zur vollständigen Lagebeschreibung notwendig.



**Definition A.6 Elementarer kognitiver Operator:** Ein elementarer kognitiver Operator  $O$  ist ein 4-Tupel  $(N, S, R, K)$ . Es gilt:

$N$  = der Name des elementaren kognitiven Operators  $O$ .

$S$  = die Sensortypen, deren Messungen der Operator  $O$  verarbeitet.

$R$  = die Ereignistypen, die von  $O$  im Weltmodell  $w$  registriert werden.

$K$  = das ausführbare Programm des Operators.

# Anhang B

## Technische Daten der verwendeten Sensorik

Technische Merkmale	Pulnix TM-765i (Vorführungsumgebung)	Sony 777AP (Ausführungsumgebung)
Bildaufnahmeverfahren	Zeilensprungverfahren	Zeilensprungverfahren
Farbkodierung	8 Bit Grauwerte	24 Bit RGB
Chip-Größe	2/3" CCD-Sensor	1/3" CCD-Sensor
Auflösung	765 × 581 Pixel	768 × 494 Pixel
Pixelgröße	11,0 μm × 11,0 μm	6,25 μm × 7,29 μm
Verschlussgeschwindigkeit	1/60 – 1/31.000 s	1/60 – 1/4.000 s
Brennweite	6 – 8 mm	6 – 8 mm

Tabelle B.1: Technische Merkmale der verwendeten Kameras

Technische Merkmale	Drehmodul MoRSE
Motor	DC, kollektorlos, elektronisch kommutiert
Leistung	100 W
Versorgung	24 V DC
Winkelgeschwindigkeit	0...180°/s
Enkoder	2000 Impulse/Umdrehung
Getriebe	Wellengetriebe 100:1
Gewicht	1,72 Kg
Abmessungen	70 mm × 70 mm
Bewegung	320°

Tabelle B.2: Technische Merkmale der Drehmodule nach Amtec

# Anhang C

## Farbräume

Farbräume dienen zur Repräsentation der farbgebenden Komponenten in unterschiedlichen Medien. Der folgende Abschnitt kann nur einen kleinen Ausschnitt aus vielen weiteren Gebräuchlichen darstellen. In der Computergraphik häufig verwendete Farbräume sind der *RGB*-<sup>1</sup>, der *HLS*-<sup>2</sup> und der *CMY*-<sup>3</sup>Farbraum. Diese Verfahren verwenden drei numerische Parameter, um eine Farbe zu charakterisieren. Hier wird bereits durch die Namensgebung ein Bild des Farbmischvorgangs vermittelt: im Beispiel des *HLS*-Verfahrens beschreibt der *Hue*-Parameter als Farbwinkel den „Farbstich“, der *Luminance*-Parameter die Helligkeit der Farbe und der *Saturation*-Parameter deren Sättigung. Weitere Details zu Farbräumen finden sich in [Foley 94].

**RGB:** Kameras und Monitore verwenden üblicherweise zur Farbbeschreibung das *RGB*-Format zur Angabe des Rot-, Grün- und Blauanteils einer Farbe (siehe Abbildung C.1). Im folgenden werden daher die anderen Farbräume durch die Umrechnungsvorschrift aus diesem System vorgestellt.

**HLS:** Die Parameter des *HLS*-Raumes beschreiben Farbwert, Helligkeit und Sättigung. Die Umrechnungsvorschrift  $RGB \rightarrow HLS$  lautet:

$$\cos H = \frac{2R - G - B}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \quad (\text{C.1})$$

$$L = \frac{1}{3}(R + G + B) \quad (\text{C.2})$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \quad (\text{C.3})$$

$$(\text{C.4})$$

Der Farbraum lässt sich als Konus vorstellen, entlang dessen Hauptachse die Helligkeitskoordinate verläuft. Der Abstand der Hauptachse gibt die Sättigung eines Farbpunkts an, wobei der Rand die höchste Farbigeit aufweist. Der Drehwinkel um die Hauptachse definiert den Farbwert (siehe Abbildung C.2).

---

<sup>1</sup>engl.: Red, Green, Blue

<sup>2</sup>engl.: Hue, Luminance, Saturation

<sup>3</sup>engl.: Cyan, Magenta, Yellow

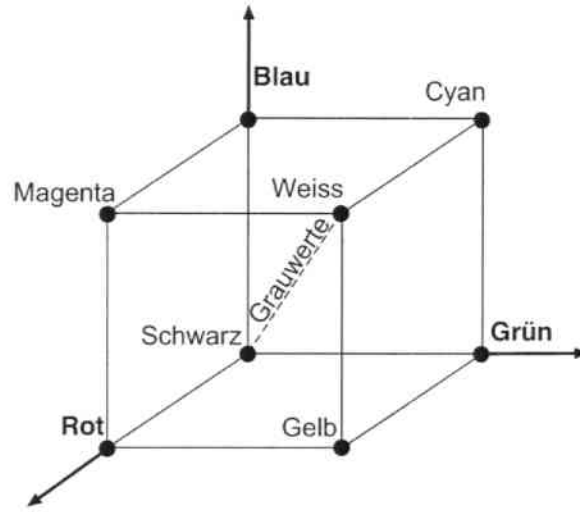


Abbildung C.1: Farbdarstellung im *RGB*-Würfel

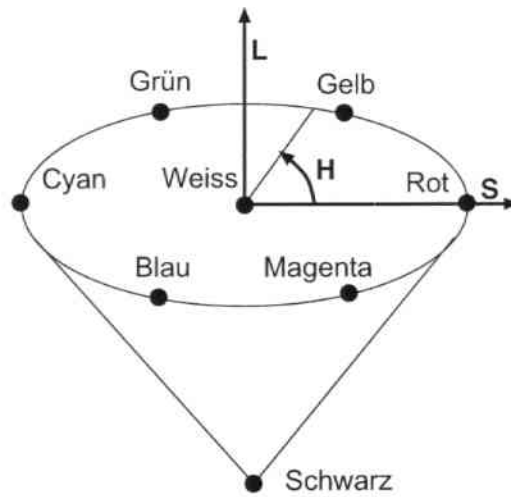


Abbildung C.2: Farbdarstellung im *HLS*-Raum

**RG:** Beim  $RG$ -Raum wird davon ausgegangen, dass der Blauanteil einer Farbe für viele Anwendungen nicht ins Gewicht fällt. Betrachtet werden stattdessen lediglich die Rot- und Grünanteile an der Helligkeit eines Punktes. Die Formeln  $RGB \rightarrow RG$  lauten:

$$R = \frac{R}{R+G+B} \quad (\text{C.5})$$

$$G = \frac{G}{R+G+B} \quad (\text{C.6})$$

$$(\text{C.7})$$

**CMY:** Im Printbereich werden aufgrund des subtraktiven Mischverhaltens von Druckfarben auf einem Träger Farbräume wie der  $CMY^4$  verwendet. Die Umrechnung ist daher:

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{C.8})$$

---

<sup>4</sup>engl.: Cyan, Magenta, Yellow

# Anhang D

## Farbkonstanz

Während Menschen die Fähigkeit haben, Farben auch unter verschiedensten Beleuchtungsverhältnissen als konstant wahrzunehmen, ist dieses Problem technisch noch nicht zufriedenstellend gelöst.

Es kommt erschwerend hinzu, dass die in den Kameras verwendeten Bildsensoren aufgrund der eingesetzten Farbfilter nicht bei allen Wellenlängen dieselbe spektrale Empfindlichkeit aufweisen (siehe Abbildung D.1). Moderne Kameras besitzen zum Ausgleich dieses Umstands und unterschiedlicher Beleuchtungsverhältnisse oft mehrere Betriebsmodi. Meist existieren hierfür feste Verstärkerschaltungen für typische Spektren von Innenraum- und Außensituationen. Eine adaptive Farbanpassung läuft meist nach der Graue-Welt-Annahme, nach der Rot-, Blau- und Grünanteile im Gesamtbild gleichen Farbanteil haben müssen.

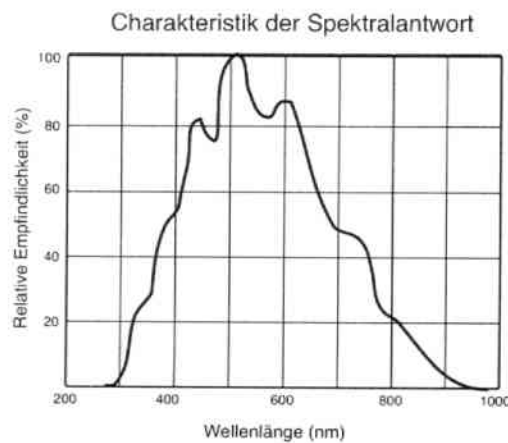


Abbildung D.1: Empfindlichkeitskurve des verwendeten CCD-Sensors in den Farbkameras

Abbildung D.2 zeigt die Auswirkungen dieser Einstellungen auf eine Situation unter verschiedenen Beleuchtungen. Die Kameraeinstellungen sind waagrecht und die Beleuchtungsquellen senkrecht eingetragen. Wie die Spalte der Innenbeleuchtung zeigt, besitzt das Tageslicht einen höheren Blauanteil als die Innenbeleuchtung. Bei Beleuchtung durch eine Leuchtstoffröhre mit der Kameraeinstellung „Innen“ zeigt das Bild einen guten Farbausgleich. Dieselbe Einstellung unter Tageslicht erzeugt jedoch einen deutlichen Blaustich. Die



variable Einstellung erzeugt ausgeglichene Farbintensitäten.

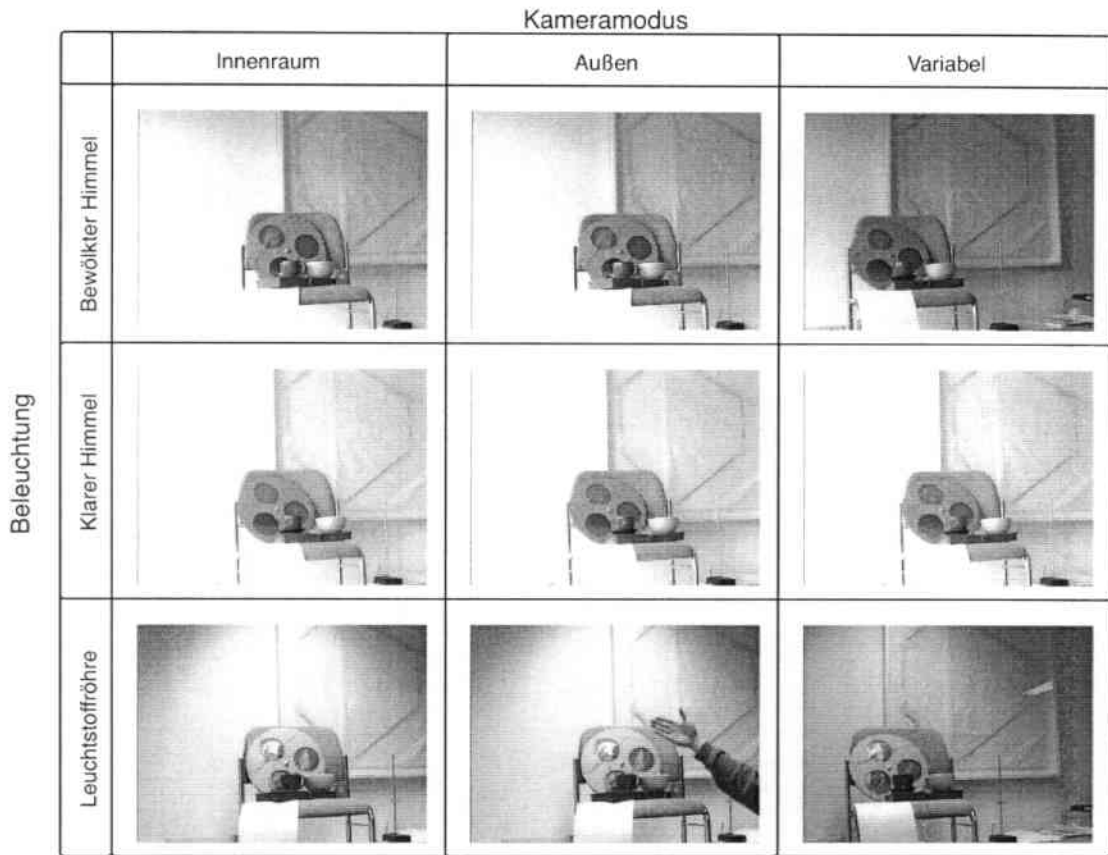


Abbildung D.2: Szene in verschiedenen Beleuchtungssituationen bei unterschiedlichen Kameraeinstellungen

Ein Problem beim Effekt des adaptiven Farbausgleichs ist in Abbildung D.3 wiedergegeben. Hier werden drei Bücher in den Grundfarben in das Kamerabild gehalten, die zu einer drastischen Anhebung der Farbverstärkung der jeweils anderen beiden Grundfarben führen. Im linken Bild führt beispielsweise das grüne Buch zu einer Anhebung des Grünanteils im Bildfarbspektrum. Daraufhin wird die Grünintensität abgesenkt und die Blau- und Rotanteile angehoben. Es zeigt sich, dass die Farbwiedergabe eines Objektes sehr stark vom farblichen Inhalt der Gesamtszene abhängt. Daraus resultieren Probleme für die Objekterkennung bzw. farbbasierte Verfolgung.

Zusammenfassend lässt sich sagen, dass keine Einstellung zufriedenstellende Ergebnisse hinsichtlich der Farbkonstanz zeigt. Die einzige, die bei Verwendung einer adaptiven Hautfarbsegmentierung stabile Ergebnisse zeigte, war diejenige mit der Implementierung der Graue-Welt-Annahme. Beide anderen Einstellungen zeigten eine so starke Farbverschiebung bei unterschiedlichen Beleuchtungen, dass eine Adaption nicht mehr möglich war.

Ein weiteres Problem der Farbdarstellung ergibt sich in Szenen mit stark variierender Helligkeit. Bei starren Blenden kommt es dabei schnell zu Rauschen des Sensors, weil die Spannungsbereiche der CCD-Sensoren durch die Verstärker nicht adäquat in die Eingangsspan-

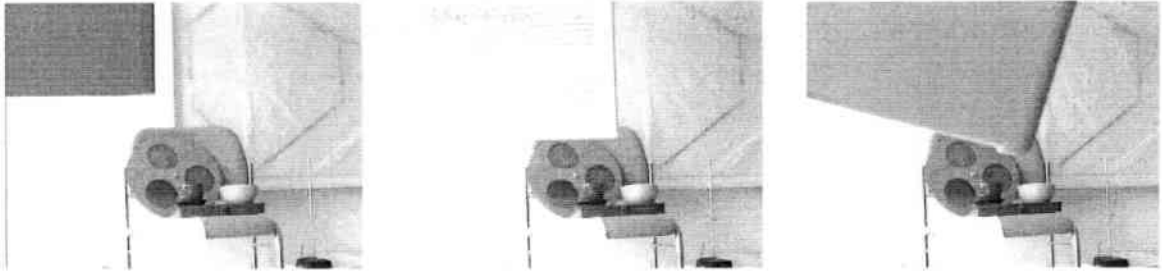


Abbildung D.3: Resultate der adaptiven Farbverstärkung auf Basis der Grauen-Welt-Annahme

nungsbereiche der AD-Wandler abgebildet werden. In der Nähe heller oder dunkler *RGB*-Werte kann jedoch der Farbwert nicht mehr exakt reproduziert werden (siehe die Formeln im Anhang C). Damit schwanken die Farbwerte zu stark für eine vernünftig gewählte Adaptionsschrittweite bei der Segmentierung.

# Abbildungsverzeichnis

1.1	Industrieroboter im Automobilbau und Roboterassistent <i>HERMES</i> . . . . .	2
1.2	Abgrenzung von Roboterassistenten zu anderen Robotersystemen . . . . .	3
1.3	Bedienschnittstelle für Mehrachsroboter der Firma Kuka (a) und haptische Schnittstelle am Endeffektor des Roboterherstellers Reis (b) . . . . .	5
2.1	Kommerzielle Datenhandschuhe . . . . .	16
2.2	Magnetische Trackingsysteme für den ganzen Körper . . . . .	17
2.3	Bildbasierte Trackingsysteme für den ganzen Körper . . . . .	17
2.4	Datenanzug mit Beugungswinkelmessung und Faro-Arm . . . . .	18
2.5	Handverfolgung nach Wu und Stereoverfolgung nach Rehg . . . . .	23
2.6	Gestenklassifikation nach Triesch . . . . .	24
2.7	Der <i>aktive Arbeitsplatz</i> und die Verfolgung zweier Hände nach Sagerer . . . . .	28
2.8	Handverfolgung in <i>APO</i> . . . . .	30
2.9	Handverfolgung in <i>LFO</i> . . . . .	31
2.10	Handlungsbeobachtung bei interaktiver Belehrung des Roboters <i>CORA</i> . . . . .	35
3.1	Beispiele von Handhabungsaufgaben im Haushalt . . . . .	42
3.2	Beispiel für Kooperation bei Handhabungsaufgaben . . . . .	43
3.3	Beispiel für eine Lösungsvorführung . . . . .	44
4.1	Interpretationsprozess für komplexe performative Handlungen . . . . .	48
4.2	Die Vorführungsumgebung mit einem trinokularen aktiven Sichtsystem . . . . .	51
4.3	Ein Datenhandschuh mit magnetfeldbasiertem Positionssystem . . . . .	52
4.4	Softwarearchitektur des Gesamtsystems zum Programmieren durch Vormachen . . . . .	53
4.5	Binokulares aktives Sichtsystem des Robotersystems <i>Albert</i> . . . . .	55
4.6	Softwarearchitektur des Gesamtsystems in der Ausführungsumgebung . . . . .	56
4.7	Aufbau des Beobachtungssystems in der Vorführ- und Ausführungsumgebung . . . . .	57
4.8	Objekte und Attribute . . . . .	59
4.9	Handlungsmodell in der Formulierung eines endlichen Automaten . . . . .	60
4.10	Transformationen im Handmodell und Ort des Endeffektors . . . . .	61
4.11	Bezugskoordinatensysteme in der Vorführungsumgebung . . . . .	62
4.12	Bezugskoordinatensysteme in der Ausführungsumgebung . . . . .	63
4.13	Grifftaxonomie nach Cutkosky . . . . .	64
5.1	Kognitiver Operator und Zusammenspiel der kognitiven Operatoren . . . . .	69
5.2	Konturmodell $\vec{r}$ einer 2D-Objektansicht mit Stützpunkten und Orthogonalen . . . . .	71

5.3	Bildabtastung, Merkmalsuche und Kantendetektor . . . . .	73
5.4	Konturmodell mit Parametern $x$ , $y$ und $r$ . . . . .	74
5.5	Kamerabild und Hough-Puffer verschiedener Drehwinkel . . . . .	76
5.6	Hierarchische Verkleinerung von Kamera- und Musterbild . . . . .	77
5.7	Kamerabild und mittlerer Verschiebungsvektor im Merkmalsraum . . . . .	78
5.8	Kamerabilder und Histogramme bei Anwendung der Farbfilterung . . . . .	78
5.9	Aufnahme des Kalibrierobjekts und identifizierte Markerzentren . . . . .	85
5.10	Ergebnis der Objekterkennung und der Lösung des Korrespondenzproblems .	86
5.11	Eine mit dem Trackingsystem vermessene Ebene . . . . .	89
5.12	Markermuster, Kamerabild und Binärbild . . . . .	93
5.13	Schema der Sensordatenfusion aus optischen und magnetischen Messungen .	96
5.14	Rekonstruktionsfehler bei der bildbasierten Objektverfolgung . . . . .	97
5.15	Lokalisierungsungenauigkeit . . . . .	98
5.16	Winkelbestimmung . . . . .	98
5.17	Strahlverfolgung in der $x/y$ - und in der $y/z$ Ebene . . . . .	98
5.18	Hautfarbenverteilung im $RGB$ -Farbraum und Zerlegung eines Bildes in $HLS$	99
5.19	Lichtspektren verschiedener Beleuchtungsquellen nach Stör링 . . . . .	100
5.20	Identifikation von Kopf und zu verfolgender Hand . . . . .	102
5.21	Schichten der Cutkosky-Griffhierarchie und korrespondierende Klassifikatoren	106
5.22	Vorverarbeitungsschritte zur Gestenerkennung . . . . .	109
5.23	Abtastung der Handkontur mit 8, 16, 32 und 64 Abtastpunkten . . . . .	110
5.24	Vornormierung der Handkontur . . . . .	111
5.25	Beispielsergebnisse der Fouriertransformation . . . . .	112
5.26	Betragsdeskriptoren von Gesten des Typs 4 und 6 . . . . .	112
5.27	Distanz zwischen Deskriptoren bekannter und unbekannter Gesten . . . . .	114
5.28	Der Gesten-Erkennungsprozess mit seinen Einzelschritten im Überblick . . .	116
5.29	Das zur Vektorquantisierung verwendete Codebuch mit 16 Wörtern. . . . .	116
5.30	Klassifizierung der Referenzgesten aufgrund ihrer Komplexität . . . . .	118
5.31	Struktur des Schwellwertmodells als ergodisches Hidden-Markov-Modell . . .	120
5.32	<i>Links-Rechts</i> -Hidden-Markov-Modell mit vier Zuständen . . . . .	121
5.33	Registrierung beobachteter Ereignisse im Weltmodell . . . . .	125
6.1	Testobjekte für die Szenenanalyse und Modelldatenbank . . . . .	128
6.2	Ergebnisse der Houghtransformation . . . . .	129
6.3	Lernen neuer Objektcharakteristika . . . . .	130
6.4	Objektklassifikation auf Basis von Farbhistogrammen . . . . .	130
6.5	Auswirkung der Verlängerung der Schnittgeraden bei der Positionsbestimmung	132
6.6	Fehlerellipse bei der Positionsbestimmung . . . . .	133
6.7	Bildbasierte Objektverfolgung mit aktiven Konturen . . . . .	134
6.8	Bildbasierte Objektverfolgung mit Regionenanalyse . . . . .	135
6.9	Positionsmessungen mit und ohne Datenhandschuh . . . . .	135
6.10	Vergleich von Positionsmessungen der einzelnen Sensoren und der Fusion . .	137
6.11	Detaillierergebnisse der Datenfusion . . . . .	138
6.12	Experiment zur farbbasierten Segmentierung mit wechselnder Beleuchtung .	139

6.13	Verlagerung des Mittelwerts bei der Hautfarbverfolgung . . . . .	140
6.14	Klassifikationsleistung von Neuronalen Netzen zur Gestenerkennung . . . . .	142
6.15	Statische Referenzgesten für die experimentelle Validierung . . . . .	146
6.16	Ergebnisvergleich einer Geste bei gemitteltem Modell und bestem Modell . . .	146
6.17	Musterkonturen für den Griffgestentyp 5' . . . . .	148
6.18	Aufgezeichnete und gefilterte Trajektorie . . . . .	150
6.19	Ergebnisse beim Filtern einer Beobachtungssequenz . . . . .	151
6.20	Wirkung der Filter auf die Anzahl der Segmente einer Bewegungsvorführung	152
6.21	Beispielablauf der Filterung und Zustandszuordnung der Beobachtung $O$ . .	153
6.22	Testhierarchie mit möglichen Einstellungen zum Erkennungstest . . . . .	155
6.23	Trajektorien der Geste 3 und Erkennungswahrscheinlichkeiten . . . . .	158
6.24	Experiment zur Validierung der Handlungsbeobachtung (Teil 1) . . . . .	164
6.25	Experiment zur Validierung der Handlungsbeobachtung (Teil 2) . . . . .	165
6.26	Experiment zur Validierung der Handlungsbeobachtung (Teil 3) . . . . .	166
6.27	Experiment zur Validierung der Handlungsbeobachtung (Teil 4) . . . . .	167
6.28	Experiment zur Validierung der Handlungsbeobachtung (Teil 5) . . . . .	168
6.29	Experiment in der Ausführumgebung . . . . .	169
C.1	Farbdarstellung im <i>RGB</i> -Würfel . . . . .	182
C.2	Farbdarstellung im <i>HLS</i> -Raum . . . . .	182
D.1	Empfindlichkeitskurve des verwendeten CCD-Sensors in den Farbkameras . .	185
D.2	Szene in verschiedenen Beleuchtungssituationen . . . . .	186
D.3	Resultate der adaptiven Farbverstärkung auf Basis der Grauen-Welt-Annahme	187

# Tabellenverzeichnis

1.1	Zusammenhang von Daten, Information und Wissen . . . . .	4
1.2	Ebenen der Roboterprogrammierung . . . . .	6
2.1	Überblick über Hand- und Armverfolgungsansätze . . . . .	24
2.2	Sensoreinsatz bei den Ansätzen zur roboterspezifischen Handlungsbeobachtung . . . . .	37
2.3	Beobachtete Merkmale der vorgestellten Systeme . . . . .	38
3.1	Beispiel von Syntax, Semantik und Pragmatik beim Tischdecken . . . . .	45
4.1	Korrespondenzen zwischen Objektgeometrie und Griffotypen . . . . .	65
4.2	Verwendete statische Gesten . . . . .	67
4.3	Verwendete dynamische Gesten . . . . .	67
5.1	Vergleich der betrachteten Methoden zur Objekterkennung . . . . .	81
5.2	Vergleich von bild- und magnetfeldbasierter Handverfolgung . . . . .	95
5.3	Eigenschaften der Fouriertransformation . . . . .	111
5.4	Zuordnung von Bewegungsvektoren zu Richtungsindizes . . . . .	117
6.1	Erkennungsleistung der Objektdetektion in der Vorführungsumgebung . . . . .	129
6.2	Erkennungsleistung der Objektdetektion in der Ausführungsumgebung . . . . .	131
6.3	Mittelwerte und Standardabweichungen bei der Szenenrekonstruktion . . . . .	131
6.4	Klassifikationsleistung des neuronalen Griffklassifikators . . . . .	141
6.5	Klassifikationsleistung des Griffklassifikators . . . . .	141
6.6	Klassifikationsleistung bzgl. der einzelnen Gestentypen . . . . .	143
6.7	Verwechslungen auf der Testmenge des Gestenklassifikators . . . . .	143
6.8	Trefferrate des Gesamterkenners zur Griff- und Gestenklassifikation . . . . .	144
6.9	Verwechslungen des Gesamterkenners mit Schwellwert. Die Zahlen bedeuten den zu erkennenden bzw. erkannten Griff- bzw. Gestentyp aus Tabelle 4.2 und aus der Cutkosky-Hierarchie in Abbildung 4.13 . . . . .	145
6.10	Ergebnis der Klassifikation statischer Handgesten . . . . .	147
6.11	Erkennungswahrscheinlichkeiten bei wachsender Anzahl von Zuständen . . . . .	149
6.12	Experimentell bestimmte optimale Zustandsanzahl der Referenzmodelle . . . . .	150
6.13	Sequenzlängen für die hierarchische Gestenklassifikation . . . . .	154
6.14	Parameter für das Erkennungssystem . . . . .	156
6.15	Erkennungserfolg beim Verfahren der maximalen Wahrscheinlichkeit . . . . .	157
6.16	Erkennungserfolg bei Nutzung des Schwellwertmodells . . . . .	159
6.17	Erkennungserfolg bei Anwendung der hierarchischen Gestenklassifikation . . . . .	159



6.18	Erkennungserfolg bei Anwendung der hierarchischen Gestenklassifikation mit dem Schwellwertmodell . . . . .	160
6.19	Unterschiedlicher Erkennungserfolg in Abhängigkeit davon, ob auch mit Gesten der Testperson trainiert wurde . . . . .	161
B.1	Technische Merkmale der verwendeten Kameras . . . . .	179
B.2	Technische Merkmale der Drehmodule nach Amtec . . . . .	179

# Algorithmenverzeichnis

5.1	Elementarer kognitiver Operator zur Szenenanalyse . . . . .	88
5.2	Konturverfolgungsalgorithmus nach Blake . . . . .	92
5.3	Algorithmus zur segmentierungsbasierten Verfolgung von Markerbewegungen	94
5.4	Fusion von bild- und magnetfeldbasierter Positionsschätzung . . . . .	99
5.5	Adaptive Hautfarbsegmentierung . . . . .	103
5.6	Elementarer kognitiver Operator zur Bewegungsverfolgung . . . . .	104
5.7	Elementarer kognitiver Operator zur Griffdetektion . . . . .	108
5.8	Bildbasierte Klassifikation statischer Gesten . . . . .	115
5.9	Bildbasierte Klassifikation dynamischer Gesten . . . . .	123
5.10	Elementarer kognitiver Operator zur Gestendetektion in der Vorführungsumgebung	124
5.11	Elementarer kognitiver Operator zur Gestendetektion in der Ausführungsumgebung	124

## Literaturverzeichnis

- [Aloimonos 93] Y. Aloimonos. *Active Vision Revisited*, Kapitel „Introduction“. Y. Aloimonos (Hrsg.), Hillsdale, 1993.
- [Ambela 99] Despina Ambela. Positionsbestimmung und Objekterkennung für Verfahren mit aktiven Konturen. Diplomarbeit, Universität Karlsruhe, Institut für Prozessrechen-technik, Automation und Robotik, 1999.
- [Amtec 00] Amtec. *Products: Rotary*, 2000. <http://www.amtec-robotics.com>.
- [Anderson 89] J. Anderson. *Kognitive Psychologie, 2. Auflage*. Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, 1989.
- [Arbib 85] M. Arbib, T. Iberall, D. Lyons, R. Linscheid. *Hand Function and Neocortex*, Kapitel „Coordinated control programs for movement of the hand“, Seiten 111–129. A. Goodwin and T. Darian-Smith (Hrsg.), Springer-Verlag, 1985.
- [Archibald 93] C. Archibald, E. Petriu. Computational paradigm for creating and executing sensorbased Robot Skills. *24th International Symposium on Industrial Robots*, Seiten 401–406, 1993.
- [Arsenio 97] A. Arsenio, J. Santos-Victor. Robust visual tracking by an Active Observer. Tagungsband: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Band 3, Seiten 1342–1347, 1997.
- [Asada 91] H. Asada, S. Liu. Transfer of human skills to neural net robot controllers. Tagungsband: *IEEE International Conference on Robotics and Automation (ICRA)*, Seiten 2442–2448, 1991.
- [Ascension 01] Ascension. *Flock of Birds Specification*. P.O. Box 527, Burlington, Vermont 05402, USA, 2001. <http://www.ascension-tech.com/products/flockofbirds/>.
- [Avrithis 00] Y. Avrithis, N. Tsapatsoulis, S. Kollias. Broadcast News Parsing using Visual Cues: a robust Face Detection Approach. Tagungsband: *2000 IEEE International Conference on Multimedia and Expo (ICME)*, Band 3, Seiten 1469–1472, 2000.
- [Badler 00] N. Badler, M. Costa, L. Zhao, D. Chi. To gesture or not to gesture: what is the Question? Tagungsband: *Computer Graphics International*, Seiten 1–6. Computer Graphics Society and British Computer Society, 19.-23. Juni 2000.
- [Bajcsy 92] R. Bajcsy, M. Campos. Active and exploratory Perception. *CVGIP: Image Understanding*, 56(1):31–40, 1992.
- [Ballard 81] D. Ballard. Generalizing the Hough transform to detect arbitrary Shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [Ballard 92] D. Ballard, C. Brown. Principles of Animate Vision. *CVGIP: Image Understanding*, 56(1):3–20, 1992.

- [Banarse 96] D. Banarse, A. Duller. Deformation Invariant Pattern Classification for Recognising Hand Gestures. Tagungsband: *Proceedings of the IEEE International Conference on Neural Networks*, Band 3, Seiten 1812–1817, 1996.
- [Bandlow 98] T. Bandlow, A. Hauck, T. Einsele, G. Färber. Recognising Objects by their Silhouette. Tagungsband: *IMACS Conference on Computer Engineering in Systems Applications (CESA)*, Seiten 744–749, April 1998.
- [Baratoff 99] G. Baratoff, I. Ahrns, C. Toepfer, H. Neumann. Ortsvariantes aktives Sehen: von biologischer Motivation zu technischer Realisierung. *Künstliche Intelligenz 1/99*, Seiten 33–35, 1999.
- [Barnard 95] K. Barnard. Computational Colour Constancy: taking Theory into Practice. Diplomarbeit, Simon Fraser University, School of Computing, 1995.
- [Bernardino 98] A. Bernardino, J. Santos-Victor. Visual behaviours for binocular tracking. *Robotics and Autonomous Systems*, 25:137–146, 1998.
- [Beyer 96] U. Beyer, F. Śmieja. A model-based Approach to Recognition and Measurement of partially hidden Objects in complex Scenes, Report Nr. 96/3. Technischer Bericht, Research Group for Adaptive Systems GMD Laboratory, 1996.
- [Bischoff 98] R. Bischoff. Design Concept and Realization of the Humanoid Service Robot HERMES. In A. Zelinsky, Hrsg., *Field and Service Robotics*, Seiten 485–492. Springer, 1998.
- [Blake 98a] A. Blake, M. Isard. *Active Contours*. Springer, 1998.
- [Blake 98b] A. Blake, M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag Berlin Heidelberg New York, 1998.
- [Braten 99] S. Braten, E. Dickmanns, M. Pellkofer, A. Rieder. Aktive Blickrichtungssteuerung in autonomen Fahrzeugen. *Künstliche Intelligenz 1/99*, Seiten 13–17, 1999.
- [Breazeal 00] C. Breazeal. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. Dissertation, Department of Electrical Engineering and Computer Science, 2000.
- [Brockmann 99] W. Brockmann, W. Heide, R. Kluthe, D. Kömpf, E. Maehle, A. Sprenger. Modellierung der kortikalen Steuerung von Blicksakkaden bei der visuellen Exploration. *Künstliche Intelligenz 1/99*, Seiten 25–30, 1999.
- [Brooks 86] R. Brooks. Achieving Artificial Intelligence through building Robots (A. I. Memo 899). Technischer Bericht, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Mai 1986.
- [Brunelli 95] R. Brunelli, T. Poggio. Template Matching: Matched Spatial Filters and beyond (Memo Nr. 1549). Technischer Bericht, MIT, AI Laboratory and Center for biological and computational learning, Oktober 1995.

- [Büker 99] U. Büker, S. Drüe, N. Götze, G. Hartmann, R. Stemmer, R. Trapp. Aktive Objekterkennung und -vermessung zur Steuerung eines Demontageroboters. *Künstliche Intelligenz 1/99*, Seiten 25–30, 1999.
- [Bundesministerium für Bildung und Forschung 01] Bundesministerium für Bildung und Forschung. *Leitprojekt Intelligente anthropomorphe Assistenzsysteme*, 2001. <http://www.morpha.de>.
- [Comaniciu 97] D. Comaniciu, P. Meer. Robust Analysis of Feature Spaces: Color Image Segmentation. Tagungsband: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 750–755, Juni 1997.
- [Corke 00] P. Corke, S. Hutchinson. Real-Time Vision, Tracking and Control. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco*, Seiten 622–628, April 2000.
- [Cutkosky 89] M. R. Cutkosky. On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989.
- [D. Huttenlocher 99] R. Lilien D. Huttenlocher, C. Olson. View-based recognition using an Eigenspace approximation to the Hausdorff measure. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 21(9), September 1999.
- [Devlin 99] K. Devlin. *Turning Information into Knowledge*. Freeman, 1999.
- [Dillmann 99] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zöllner, M. Bordegoni. Learning Robot Behaviour and Skills based on Human Demonstration and Advice: the Machine Learning Paradigm. Tagungsband: *9th International Symposium of Robotics Research (ISRR 1999)*, Seiten 229–238, Snowbird, Utah, USA, 9.-12. Oktober 1999.
- [Dornaika 97] F. Dornaika, C. Garcia. Object Pose by Affine Iterations, Report Nr. 97/4. Technischer Bericht, RWCP Theoretical Foundation GMD Laboratory, 1997.
- [Duda 72] R. O. Duda, P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [Ehrenmann 00] M. Ehrenmann, D. Ambela, P. Steinhaus, R. Dillmann. A Comparison of Four Fast Vision-Based Object Recognition Methods for Programming by Demonstration Applications. Tagungsband: *Proceedings of the 2000 International Conference on Robotics and Automation (ICRA)*, Band 1, Seiten 1862–1867, San Francisco, Kalifornien, USA, 24.–28. April 2000.
- [Ehrenmann 01a] M. Ehrenmann, T. Lütticke, R. Dillmann. Dynamic Gestures as an Input Device for Directing a Mobile Platform. Tagungsband: *Proceedings of the 2001 International Conference on Robotics and Automation (ICRA)*, CD-ROM, Seoul, Korea, 23.–25. Mai 2001.

- [Ehrenmann 01b] M. Ehrenmann, O. Rogalla, R. Zöllner, R. Dillmann. Teaching Service Robots complex Tasks: Programming by Demonstration for Workshop and Household Environments. Tagungsband: *Proceedings of the 2001 International Conference on Field and Service Robots (FSR)*, Band 1, Seiten 397–402, Helsinki, Finnland, 11.–13. Juni 2001.
- [Ehrenmann 01c] M. Ehrenmann, R. Zöllner, S. Knoop, R. Dillmann. Sensor Fusion Approaches for Observation of User Actions in Programming by Demonstration. Tagungsband: *Proceedings of the 2001 International Conference on Multi Sensor Fusion and Integration for Intelligent Systems (MFI)*, Band 1, Seiten 227–232, Baden-Baden, 19.–22. August 2001.
- [Ehrenmann 02] M. Ehrenmann, R. Zöllner, O. Rogalla, R. Dillmann. Programming Service Tasks in Household Environments by Human Demonstration. Tagungsband: *Proceedings of the 2002 11th IEEE International Workshop on robot and Human Interactive Communication (ROMAN)*, Seiten 460–467, Berlin, 25.–27. September 2002.
- [Ehrenmann 98] M. Ehrenmann. Objekterkennung in Kamerabildern in einem fusionierten Ansatz von Maschinensehen und Griffwinkelbetrachtung. Diplomarbeit, Universität Karlsruhe, 1998.
- [Elsen 98] I. Elsen. A Pixel-Based Approach to View-Based Object Recognition with Self-Organizing Neural Networks. Tagungsband: *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronic Society (IECON)*, Band 4, Seiten 2040–2044, 1998.
- [Faro 01] Faro. *Produktspezifikation zum Faro-Arm*. <http://www.faro.com>, 2001.
- [Faugeras 93] O. Faugeras. *Three Dimensional Computer Vision - A Geometric Viewpoint*. MIT Press Cambridge, Massachusetts, 1993.
- [Feyrer 99] S. Feyrer, A. Zell. Personentracking mit einer mobilen Roboterplattform unter Verwendung eines multimodalen Detektionsansatzes. *Künstliche Intelligenz 1/99*, Seiten 7–12, 1999.
- [Finlayson 96] G. Finlayson. Color in Perspective. Tagungsband: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Band 18, Seiten 1034–1038, Oktober 1996.
- [Foley 94] James D. Foley. *Grundlagen der Computergraphik: Einführung, Konzepte, Methoden*. Addison-Wesley, 1994.
- [Forney 73] G. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, März 1973.
- [Frankfurter Allgemeine Zeitung 00] Die Roboter dringen in ganz neue Bereiche vor. *Frankfurter Allgemeine Zeitung*, 22. März 2000.
- [Frankfurter Allgemeine Zeitung 01a] Dienstleistung avanciert zum neuen Geschäftsfeld für Maschinenbauer. *Frankfurter Allgemeine Zeitung*, 25. Januar 2001.



- [Frankfurter Allgemeine Zeitung 01b] Japan will die ersten Humanoiden marktreif machen. *Frankfurter Allgemeine Zeitung*, 26. Februar 2001.
- [Frankfurter Allgemeine Zeitung 99a] Die amerikanischen Automobilhersteller wollen die Produktion mit Hilfe von Robotern vorantreiben. *Frankfurter Allgemeine Zeitung*, 13. Oktober 1999.
- [Frankfurter Allgemeine Zeitung 99b] Sensoren bestimmen das Leistungsvermögen von Service- und Montagerobotern. *Frankfurter Allgemeine Zeitung*, 4. Oktober 1999.
- [Friedrich 98] H. Friedrich. *Interaktive Programmierung von Manipulationssequenzen*. Dissertation, Universität Karlsruhe, 1998.
- [Friedrich 99] H. Friedrich, V. Grossmann, M. Ehrenmann, O. Rogalla, R. Zöllner, R. Dillmann. Towards Cognitive Elementary Operators: Grasp Classification using Neural Network Classifiers. Tagungsband: *Proceedings of the IASTED International Conference on Intelligent Systems and Control (ISC)*, Band 1, Seiten 88–93, Santa Barbara, Kalifornien, USA, 28.-30. Oktober 1999.
- [Fritsch 00] J. Fritsch, F. Lömker, M. Wienecke, G. Sagerer. Detecting Assembly Actions by Scene Observation. Tagungsband: *Proceedings International Conference on Image Processing*, Band I, Seiten 212–215, Vancouver, September 2000. IEEE.
- [Garcia 99] C. Garcia, G. Tziritas. Face Detection using Quantized Skin Color Regions Merging and Wavelet Packet Analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, September 1999.
- [Gavrila 95] D. Gavrila, L. Davis. Towards 3d model-based Tracking and Recognition of Human Movement: a Multi-View Approach. In Martin Bichsel, Hrsg., Tagungsband: *International Workshop on Face and Gesture Recognition*, Zürich, Seiten 272–277, Juni 1995.
- [Gimeno 01] Elena Gimeno. Bildbasierte Erkennung statischer Gesten. Diplomarbeit, Universität Karlsruhe, Institut für Prozessrechentechnik, Automation und Robotik, 2001.
- [Goncalves 95] L. Goncalves, E. Di Bernardino, E. Ursella, P. Perona. Monocular Tracking of the Human Arm in 3d. Tagungsband: *Fifth International Conference on Computer Vision (ICCV)*, Seiten 764–770, 20.-23. Juni 1995.
- [Gonzales 93] R. C. Gonzales, P. Wintz. *Digital Image Processing*. Addison-Wesley, 1993.
- [Gonzalez-Linares 99] J. Gonzalez-Linares, N. Guil, P. Pérez, M. Ehrenmann, R. Dillmann. An Efficient Image Processing Algorithm for High-Level Skill Acquisition. Tagungsband: *Proc. of the International Symposium on Assembly and Task Planning (ISATP)*, Porto, Portugal, Seiten 262–267, Juli 1999.
- [Gonzalez 93] R. C. Gonzalez, R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.

- [Haberäcker 95] P. Haberäcker. *Praxis der digitalen Bildverarbeitung und Mustererkennung*. Carl Hanser Verlag München Wien, 1995.
- [Hägele 01] M. Hägele, J. Neugebauer, R. D. Schraft. From Robots to Robot Assistants. Tagungsband: *Proceedings of the 32nd International Symposium on Robotics (ISR)*, Band 1, Seiten 404–409, Seoul, Korea, 19.–21. April 2001.
- [Hauck 98] A. Hauck, M. Sorg, G. Färber, T. Schenk. A biologically motivated model for the control of visually guided reach-to-grasp movements. Tagungsband: *Proceedings of the International Conference on Intelligent Systems*, Seiten 295–300, 1998.
- [Heap 95] T. Heap, F. Samaria. Real-time hand tracking and gesture recognition using smart snakes. Technischer Bericht, Olivetti Research Limited, 20. Juni 1995.
- [Heap 97] T. Heap, D. Hogg. Improving Specificity in PDMs using a Hierarchical Approach. Tagungsband: *Proceedings of the British Machine vision Conference, Essex*, Seiten 80–89, September 1997.
- [Heikkilä 00] Janne Heikkilä. Geometric Camera Calibration using Circular Control Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, Oktober 2000.
- [Heise 92] R. Heise. Programming Robots by Example. Technischer Bericht, Department of Computer Science, The university of Calgary, 1992.
- [Hesse 88] H. Hesse, K. Goser. Die hough-transformation - ein verfahren zur klassifizierung beliebiger konturen. *Robotersysteme*, 4:27–32, 1988.
- [Hough 62] P. V. C. Hough. Method and Means for Recognizing Complex Patterns, 1962. U.S. Patent 3,069,654. Patentiert am 18. Dezember 1962.
- [Iberall 94] T. Iberall, G. Sukhatme, D. Beattie, G. Bekey. On the Development of EMG Control for a Prosthesis using a Robotic Hand. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Diego, Kalifornien, Band 2, Seiten 1753–1758, 1994.
- [Ikeuchi 94] K. Ikeuchi, T. Suehiro. Towards an Assembly Plan from Observation, Part I: Task Recognition with Polyhedral Objects. *IEEE Transactions on Robotics and Automation*, 10(3):368–385, 1994.
- [Imagawa 98] K. Imagawa, Shan Lu, S. Igi. Color-Based Hands Tracking System for Sign Language Recognition. Tagungsband: *Third IEEE International Conference on Automatic Face and Gesture Recognition*, Seiten 462–467, 1998.
- [Intel 03] Intel. *Open Source Computer Vision Library*, 2003. <http://www.intel.com/research/mrl/research/opencv/>.
- [ISO 00] ISO. *Manipulating Industrial Robots—Object Handling with Grasp-Type Grippers. Vocabulary and Presentation of Characteristics*, November 2000. 14539:2000.

- [Jähne 97] B. Jähne. *Digitale Bildverarbeitung*. Springer-Verlag Berlin Heidelberg New York, 4., vollst. neubearb. Aufl. Auflage, 1997.
- [Jain 95] R. Jain, R. Kasturi, B. G. Schunck. *Machine Vision*. McGraw Hill, Inc., 1995.
- [Jeanneroud 84] M. Jeanneroud. The Timing of Natural Prehensions Movements. *Journal of motor behaviour*, 16(3):235–254, 1984.
- [Jiar 96a] Y. Jiar, M. Wheeler, K. Ikeuchi. Hand Action Perception and Robot Instruction. Tagungsband: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Band 3, Seiten 1586–93, 1996.
- [Jiar 96b] Y. Jiar, M. Wheeler, K. Ikeuchi. Hand Action Perception and Robot Instruction. Technischer Bericht, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 1996.
- [Jobson 95] D. Jobson. Properties of a Center/Surround Retinex: Part 2. Surround Design. Technischer Bericht, NASA, Langley Research Center, 1995.
- [Jojic 00] N. Jojic, B. Brumitt, B. Meyers, S. Harris, T. Huang. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. Tagungsband: *IEEE International Conference on Face and Gesture Recognition*, Seiten 468–475, 2000.
- [Kaiser 96] M. Kaiser. *Interaktive Akquisition elementarer Roboterfähigkeiten*. Dissertation, Universität Karlsruhe (TH), 1996.
- [Kakadiaris 96] I. Kakadiaris, D. Metaxas. Model-Based Estimation of 3d Human Motion with Occlusion Based on Active Multi-Viewpoint Selection. Tagungsband: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 81–87, 18.–20. Juni 1996.
- [Kang 93] S. Kang, K. Ikeuchi. Temporal Segmentation of Tasks from Human Hand Motion. Technischer Bericht CMU-CS-93-150, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, April 1993.
- [Kang 94] S. Kang. *Robot Instruction by Human Demonstration*. Dissertation, Carnegie Mellon University, Pittsburg, Pennsylvania, 1994.
- [Kang 97] S. Kang, K. Ikeuchi. Toward Automatic Robot Instruction from Perception: Mapping Human Grasps to Manipulator Grasps. *Robotics and Automation*, 13(1):81–95, Februar 1997.
- [Kass 88] M. Kass, A. Wittkin, D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 2(3):321–331, 1988.
- [Kefalea 97] E. Kefalea, O. Rehse, Chr. von der Malsburg. Object Classification Based on Contours with Elastic Graph Matching. Tagungsband: *Proceedings of the IWVF3 97, World Scientific, Singapore*, <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/1997/>, 1997.

- [Kennedy 93] R. Kennedy, N. Lane, K. Berbaum, M. Lillenthal. A Simulator Sickness Questionnaire: A new Method for Quantifying Simulator Sickness. *International Journal of Aviation Psychology*, 3(3):203–220, 1993.
- [Kestler 96] H. Kestler, M. Borst, H. Neumann. Einfache Handgestikerkennung mit einem zweistufigen Nearest-Neighbour Klassifikator. Technischer Bericht, Universität Ulm, SFB 527, 96/6, 1996.
- [Kestler 99] H. Kestler, S. Simon, A. Braune, F. Schwenker, G. Palm. Object classification using Simple, Colour-Based Visual Attention and a Hierarchical Neural Network for Neuro-Symbolic Integration. Technischer Bericht, Universität Ulm, SFB 527, 99/4, 1999.
- [Kim 96] J. Kim, W. Jang, Z. Bien. A Dynamic Gesture Recognition System for the Korean Sign Language (KSL). *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 26(2):354–359, April 1996.
- [Kim 99] H. Kim, B. You, G. Hager, S. Oh, C. Lee. Three-Dimensional Pose Determination for a Humanoid Robot using Binocular Head System. Tagungsband: *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Band 2, Seiten 1204–1209, 1999.
- [Kindratenko 96] V. Kindratenko, P. van Espen. Classification of Irregularly Shaped Micro-Objects using Complex Fourier Descriptors. Tagungsband: *ICPR*, Seite B75.14, 1996.
- [Klaus 87] B. Klaus, P. Horn. *Robot Vision*. MIT Press, 1987.
- [Kohler 00] M. Kohler. *Übersicht zur Handgestenerkennung*. <http://ls7-www.cs.uni-dortmund.de/research/gesture/>, 2000.
- [Kollnig 95] H. Kollnig, H.-H. Nagel. 3D Pose Estimation by Fitting Image Gradients directly to Polyhedral Models. *Fifth International Conference on Computer Vision*, Seiten 569–574, 1995.
- [Kuniyoshi 93] Y. Kuniyoshi, H. Inoue. Qualitative recognition of ongoing human action sequences. Tagungsband: *13th International Joint Conference on Artificial Intelligence*, Seiten 1600–1609, 1993.
- [Kuniyoshi 94] Y. Kuniyoshi, M. Inaba, H. Inoue. Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.
- [Lai 94] K. Lai. *Deformable Contours: Modeling, Extraction, Detection and Classification*. Dissertation, University of Wisconsin-Madison, 1994.
- [Lai 95] K. F. Lai, R. T. Chin. Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1084–1090, 1995.

- [Lamar 99] M. Lamar, M. Bhuiyan, A. Iwata. Hand Gesture Recognition using morphological Principal Component Analysis and an improved CombNET-II. Tagungsband: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Band 4, Seiten 57–62, 1999.
- [Land 86] E. Land. An alternative Technique for the Computation of the Designator in the Retinex Theory. Tagungsband: *Proceedings of the National Academy of Sciences*, Band 83, Seiten 3078–3080, 1986.
- [Lee 96] C. Lee, Y. Xu. Online, Interactive Learning of Gestures for Human/Robot Interfaces, Minneapolis, Minnesota. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation*, Band 4, Seiten 2982–2987, April 1996.
- [Lee 99] H. Lee, J. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, Oktober 1999.
- [Lehne 01] Christian Lehne. Farbbasierte objekterkennung für einen mobilen roboter. Diplomarbeit, Universität Karlsruhe, Institut für Prozessrechentechnik, Automation und Robotik, 2001.
- [Leibe 01] B. Leibe, D. Minnen, J. Weeks, T. Starner. Integration of Wireless Gesture Tracking, Object Tracking, and 3D Reconstruction in the Perceptive Workbench. Tagungsband: *Proceedings of 2nd International Workshop on Computer Vision Systems*, Band 2095, Seiten 73–92. Springer, Berlin, 2001.
- [Lin 00] J. Lin, Y. Wu, T. Huang. Modeling the Constraints of Human Hand Motion. Tagungsband: *Proceedings of the IEEE Workshop on Human Motion*, Seiten 121–126, 2000.
- [Luong 97] Q.-T. Luong, O. Faugeras. Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices. *International Journal on Computer Vision*, 22(3):261–289, 1997.
- [Lütticke 00] Tobias Lütticke. Erkennung dynamischer Gesten zur Kommandierung mobiler Roboter. Diplomarbeit, Universität Karlsruhe, Institut für Prozessrechentechnik, Automation und Robotik, 2000.
- [Ly Duc 00] Nguyen Ly Duc. Erkennung statischer Gesten mittels Neuronaler Netze. Studienarbeit, Universität Karlsruhe, Institut für Prozessrechentechnik, Automation und Robotik, 2000.
- [Ly Duc 01] Nguyen Ly Duc. Hautfarbenadaption und Handverfolgung. Diplomarbeit, Universität Karlsruhe, Institut für Prozessrechentechnik, Automation und Robotik, 2001.
- [MacKenzie 94] C. MacKenzie, T. Iberall. The Grasping Hand. *Advances in psychology*, 104:15–46, 1994.

- [Mandel 77] M. Mandel. Iconic devices in American sign language. In L. A. Friedman, Hrsg., *On the other hand. New Perspectives on American Sign language*, Seiten 57–107. Academic Press, 1977.
- [Marchand 98] E. Marchand, G. Hager. Dynamic Sensor Planning in Visual Servoing. Tagungsband: *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA)*, Band 3, Seiten 1988–1993, 1998.
- [Marr 82] D. Marr. *Vision*. Freeman, New York, 1982.
- [Marsh 98] T. Marsh. An Iconic Gesture is worth more than a thousand Words. Tagungsband: *Proceedings of the IEEE Conference on Information Visualization*, Seiten 222–223, 1998.
- [Matrox 98a] *Genesis Installation and Hardware Reference. Manual No. 10503-MN-0121*, 1998. <http://www.matrox.com>.
- [Matrox 98b] *Matrox Imaging Library Command Reference. Version 5.1. Manual No. 10512-MN-0501*, 1998. <http://www.matrox.com>.
- [Matrox 98c] *Matrox Meteor Installation and Hardware Reference. Manual No. 10529-MN-0110*, 1998. <http://www.matrox.com>.
- [McInerney 96] T. McInerney, D. Terzopoulos. Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis*, 1(2):91–108, 1996.
- [McNeil 92] D. McNeil. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago, 1992.
- [MetaMotion 01] MetaMotion. *Gypsy Specification*. Meta Motion, 268 Bush St. 1, San Francisco, California 94104, USA, 2001. <http://www.MetaMotion.com/motion-capture/magnetic-motion-capture-2.htm>.
- [Mindflux 00] Mindflux. *Data Glove 5 Specifications*, 2000. <http://www.mindflux.com.au/products/5dt/glove.html>.
- [Miura 92] J. Miura, H. Kawarabayashi, M. Watanabe, T. Tanaka, M. Asada, Y. Shirai. Tracking a Moving Object by an Active Vision System: PANTHER-VZ. Tagungsband: *Proceedings of the International Symposium on Robotics, Mechatronics and Manufacturing Systems 92, Kobe, Japan*, Seiten 957–962, September 1992.
- [Monkman 91] G. Monkman. Sensor fusion in robot programming. *IEEE Colloquium on Principles and Applications of Data Fusion*, Seiten 8/1–8/5, 1991.
- [Moravec 00] Hans Moravec. Die Robotik – eine Vorankündigung. *Frankfurter Allgemeine Zeitung*, 26. Juli 2000.



- [Mori 97] T. Mori, Y. Kamisuwa, H. Mizoguchi, T. Sato. Action Recognition System based on Human Finder and Human Tracker. Tagungsband: *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Band 3, Seiten 1334–1341, 1997.
- [MOTEK 01] MOTEK. *PRIMAS Specification*. MOTEK European Headquarters, Nieuwe Hemweg 6a, 1013 BG Amsterdam, Niederlande, 2001. <http://osh.motek.org>.
- [Nagel 95] H.-H. Nagel. *Kognitive Systeme, Skriptum zur Vorlesung. Überarbeitete Auflage von R. Dillmann*. Institut für Prozessrechentchnik Automation und Robotik, Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik, Universität Karlsruhe, 1995.
- [Napier 56] J. Napier. The Prehensile Movements of the Human Hand. *The Journal of Bone and Joint Surgery*, 38B(4):902–913, 1956.
- [Nemire 94] K. Nemire. Building Usable Virtual Environment Products. *CyberEdge Journal*, Seiten 8–12, September/Okttober 1994.
- [Newtonson 77] D. Newtonson, et al. The Objective Basis of Behaviour Units. *Journal of Personality and Social Psychology*, 35(12):847–862, 1977.
- [Nickel 03] Kai Nickel. Erkennung von Zeigegesten basierend auf 3D-Tracking von Kopf und Händen. Diplomarbeit, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme, 2003.
- [Nordlund 96] P. Nordlund, T. Uhlin. Closing the Loop: Detection and Pursuit of a Moving Object with Sensors Onboard a Moving Robot. *Image Vision and Computing*, 14(4):265–275, 1996.
- [Ogata 94] H. Ogata, T. Takahashi. Robotic Assembly Operation Teaching in a Virtual Environment. *IEEE Transactions on Robotics and Automation*, 10(3):391–399, 1994.
- [Ogata 97] H. Ogata, T. Takahashi. A Geometric Approach to Task Understanding and Playback: Compact and Robust Task Description for Complex Environments. Tagungsband: *Proceedings of the IEEE International Conference on Advanced Robotics (ICAR)*, Monterey, USA, Seiten 693–698, 7.-9. Juli 1997.
- [Onda 97] H. Onda, H. Hirukawa, F. Tomita, T. Suehiro, K. Takase. Assembly Motion Teaching System using Position/Force Simulator—Generating Control Program. Tagungsband: *10th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 389–396, Grenoble, Frankreich, 7.-11. September 1997.
- [Open Inventor Architecture Group 94] Open Inventor Architecture Group. *Open Inventor™ C++ Reference Manual, The Official Reference Document for Open Inventor*. McGraw-Hill, 1994.

- [Park 00] J. Park, J. Seo, D. An, S. Chung. Detection of Human Faces using Skin Color and Eyes. Tagungsband: *2000 IEEE International Conference on Multimedia and Expo (ICME)*, Band 1, Seiten 133–136, 2000.
- [Paul 95] G. Paul, K. Ikeuchi. Modelling planar Assembly Tasks: Representation and Recognition. Tagungsband: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Pittsburgh, Pennsylvania*, Band 1, Seiten 17–22, 5.–9. August 1995.
- [Pauli 98] J. Pauli. *Machine Learning*, Band 31, Kapitel „Learning to Recognize and Grasp Objects“, Seiten 239–258. Kluwer Academic Publishers, Boston, 1998.
- [Peixoto 00] P. Peixoto, J. Batista, H. Araújo. Integration of Information from several Vision Systems for a common Task of Surveillance. *Robotics and Autonomous Systems*, 31:99–108, 2000.
- [Pentland 00] A. Pentland. Looking at People: Sensing for Ubiquitous and Wearable Computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, Januar 2000.
- [Perzanowski 01] D. Perzanowski, A. Schultz, W. Adams, E. Marsh, M. Bugajska. Building a Multimodal Human-Robot Interface. *IEEE Journal on Intelligent Systems*, Seiten 16–21, Januar–Februar 2001.
- [Phoenix Technologies 01] Phoenix Technologies. *Visualeyez Specification*. Phoenix Technologies Incorporated, 4302 Norfolk St., Burnaby., British Columbia, Kanada V5G 4J9, 2001. [http://www.ptiphoenix.com/home\\_page.htm](http://www.ptiphoenix.com/home_page.htm).
- [Plamondon 00] R. Plamondon, S. Srihari. On-Line and Off-Line Handwriting Recognition: A comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, Januar 2000.
- [Poggio 93] T. Poggio, A. Hurlbert. Observations on Cortical Mechanisms for Object Recognition and Learning, Memo Nr. 1404. Technischer Bericht, MIT, AI Laboratory and Center for Biological and Computational Learning, Dezember 1993.
- [Polhemus 02] Polhemus. *StarTrak Specifications*, 2002. <http://www.polhemus.com/stards.htm>.
- [Polhemus 93] *Polhemus 3SPACE<sup>TM</sup> FASTRAK<sup>TM</sup>*. P.O. Box 560, Colchester, Vermont 05446, USA, 1993.
- [Probst 99] G. Probst, S. Raub, K. Romhardt. *Wissen managen: wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Gabler, 3. Auflage, 1999.
- [Rabiner 89] L. Rabiner. A tutorial on Hidden Markov Models and selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–285, Februar 1989.

- [Rao 96] R. Rao. Robust Kalman Filters for Prediction, Recognition, and Learning. Report Nr. 645. Technischer Bericht, University of Rochester, Computer Science Department, Dezember 1996.
- [Rao 99] A. Rao. Report of Project II: Hidden Markov Model (HMM). Technischer Bericht CSE655 Pattern Recognition, State University of New York at Buffalo, Mai 1999.
- [Rehg 93] J. Rehg, T. Kanade. DigitEyes: Vision-Based Human Hand Tracking, CMU-CS-93-220. Technischer Bericht, Carnegie Mellon University, USA, Dezember 1993.
- [Rehg 94] J. Rehg, T. Kanade. Visual Tracking of high DOF Articulated Structures: an Application to Human Hand Tracking. Tagungsband: *ECCV*, Seiten 35–46, 1994.
- [Rehg 95] J. Rehg, T. Kanade. Model-Based Tracking of Self-Occluding Articulated Objects. Tagungsband: *Fifth International Conference on Computer Vision (ICCV)*, Seiten 612–617, 20.–23. Juni 1995.
- [Riedmiller 92] M. Riedmiller, H. Braun. RPROP- A Fast Adaptive Learning Algorithm, 1992.
- [Rigoll 97] G. Rigoll, A. Kosmala, S. Eickeler. High Performance Real-Time Gesture Recognition using Hidden Markov Models. Tagungsband: *Proceedings of the Gesture Workshop (GW)*, Seiten 69–80, September 1997.
- [Rime 91] B. Rime, L. Schiaratura. Gesture and Speech. In R. Feldman, B. Rime, Hrsg., *Fundamentals of nonverbal behavior*, Seiten 239–281. Cambridge University Press, 1991.
- [Rogalla 00] O. Rogalla, K. Pohl, R. Dillmann. A general Approach for Modeling Robots. Tagungsband: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Band 3, Seiten 1963–1968, 2000.
- [Rogalla 98] O. Rogalla, M. Ehrenmann, R. Dillmann. A sensor fusion approach for PbD. Tagungsband: *Proc. of the IEEE/RSJ Conference Intelligent Robots and Systems, IROS'98*, Band 2, Seiten 1040–1045, 1998.
- [Ruf 97] A. Ruf, M. Tonko, R. Horaud, H.-H. Nagel. Visual Tracking of an End-Effector by Adaptive Kinematic Prediction. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2:893–899, 1997.
- [Ryan 93a] M. S. Ryan, G. R. Nudd. Dynamic Character Recognition Using Hidden Markov Models. Warwick Research Report RR 244, Department of Computer Science, University of Warwick, Coventry, Mai 1993. [www.dcs.warwick.ac.uk/pub/reports/rr/244.html](http://www.dcs.warwick.ac.uk/pub/reports/rr/244.html).
- [Ryan 93b] M. S. Ryan, G. R. Nudd. The Viterbi Algorithm. Warwick Research Report RR 238, Department of Computer Science, University of Warwick, Coventry, Februar 1993. [www.dcs.warwick.ac.uk/pub/reports/rr/238.html](http://www.dcs.warwick.ac.uk/pub/reports/rr/238.html).
- [Rybski 99] P. E. Rybski, R. Voyles. Interactive Task Training of a Mobile Robot through Human Gesture Recognition. *IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, Michigan, Seiten 664–669, Mai 1999.

- [Schaude 96] H. Schaude. Dokumentation zu KaVis. Technischer Bericht, Institut für Prozessrechentchnik, Automation und Robotik, Universität Karlsruhe (TH), 1996.
- [Schlesinger 19] G. Schlesinger. *Ersatzglieder und Arbeitshilfen*. Springer-Verlag, Berlin, 1919.
- [Schmidt 98] J. Schmidt, H. Süße. 3D-Object Recognition using Model-Based Invariants. Tagungsband: *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronic Society (IECON)*, Seiten 2034–2039, 1998.
- [Sensable 01] Sensable. *Produktspezifikationen zum Phantom*. <http://www.sensable.com>, 2001.
- [Shepherd 93] B. Shepherd. Applying Visual Programming to Robotics. *Proceedings of the IEEE International Conference on Robotics and Automation (ICORA)*, 2:707–712, 1993.
- [Shiga 00] Y. Shiga, H. Ebine, M. Ikeda, O. Nakamura. Human Face Extraction based on Color and Moving Information and the Recognition of Expressions. Tagungsband: *Canadian Conference on Electrical and Computer Engineering*, Band 2, Seiten 1100–1108, 2000.
- [Shirai 73] Y. Shirai, H. Inoue. Guiding a Robot by Visual Feedback in Assembling Tasks. Tagungsband: *IEEE Transactions on Pattern Recognition*, Band 5, Seiten 99–108, 1973.
- [Sidenbladh 99] H. Sidenbladh, D. Kragic, H. Christensen. A Person Following Behaviour for a Mobile Robot. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, USA*, Seiten 670–675, April 1999.
- [Sigal 00] L. Sigal, S. Sclaroff, V. Athitsos. Estimation and Prediction of Evolving Color Distributions for Skin Segmentation under Varying Illumination. Tagungsband: *IEEE Conference on Computer Vision and Pattern Recognition*, Band 2, Seiten 152–159, 2000.
- [SimGraphics 01] SimGraphics. *VActor Specification*. 1137 Huntington Drive, South Pasadena, California 91030-4563, USA, 2001. <http://www.simg.com>.
- [Singer 00] Wolf Singer. Wir benötigen den neuronalen Code. *Frankfurter Allgemeine Zeitung*, 24. August 2000.
- [Speidel 02] S. Speidel. Historische Entwicklung humanoider Roboter—Aspekte. In *Seminar „Humanoide Roboter“*. Fakultät für Informatik, Universität Karlsruhe (TH), Januar 2002.
- [SPIEGEL ONLINE 01] SPIEGEL ONLINE. *Roboter vs. Mensch*. <http://www.spiegel.de/auto/news/00,1518,123786,00.html>, 23. März 2001.
- [Spinner 01] Natalie Spinner. Kalibrierung und Tiefenschätzung bei parallel ausgerichteten Kameras. Studienarbeit, Universität Karlsruhe, Institut für Prozessrechentchnik, Automation und Robotik, 2001.

- [Starner 00] T. Starner, B. Leibe, B. Singletary, J. Pair. MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Gaming Interface through Wearable Computers and Multi-Modal Input and Output. Tagungsband: *International Conference on Intelligent User Interfaces*, Seiten 256–259, New York, Januar 2000. ACM, ACM Press.
- [Starner 95] T. Starner. Real-Time American Sign Language Recognition from Video using Hidden Markov Models. Tagungsband: *Proceedings of the IEEE International Symposium on Computer Vision*, Seiten 265–270, 1995.
- [Stasch 97] M. Stasch. Entwicklung und Aufbau eines magnetfeldbasierten Positionssensors. Diplomarbeit, Universität Karlsruhe, 1997.
- [Steinhage 00a] A. Steinhage. The Dynamic Approach to Anthropomorphic Robotics. Tagungsband: *4th Portuguese Conference on Automatic Control*, Proceedings of Control, 2000. <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/accel/top.html>.
- [Steinhage 00b] A. Steinhage, v. Seelen. Dynamische Systeme zur Verhaltensgenerierung eines anthropomorphen Roboters. Tagungsband: *Autonome Mobile Systeme (AMS)*, Informatik aktuell, Seiten 260–269, Karlsruhe, 2000. Springer-Verlag.
- [Steinhaus 97] P. Steinhaus. Prinzipielle Komponenten-Analyse versus multimediale Filter-histogramme. Diplomarbeit, Universität Karlsruhe (TH), 1997.
- [Steinhaus 99] P. Steinhaus, M. Ehrenmann, R. Dillmann. MEPHISTO: A Modular and Extensible Path Planning System using Observation. Tagungsband: *First international Conference on Computer Vision Systems (ICVS)*, LNCS, Band 1, Seiten 361–375, Las Palmas, Gran Canaria, Spanien, Januar 1999.
- [Störring 99] M. Störring, H. Andersen, E. Granum. Skin Colour Detection under Changing Lighting Conditions. Tagungsband: *7th symposium on Intelligent Robot Systems*, Seiten 187–195, 1999.
- [Sturman 94] D. Sturman, D. Zeltzer. A Survey on Glove-Based Input. *IEEE Computer Graphics and Applications*, 14(1):30–39, 1994.
- [Takahashi 92] T. Takahashi, H. Ogata. Robotic Assembly Operation based on Task-Level Teaching in Virtual Reality. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation*, Nizza, Frankreich, Seiten 1083–1088, Mai 1992.
- [Takahashi 96] T. Takahashi. Time Normalization and Analysis Method in Robot Programming from Human Demonstration Data. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, USA, Band 1, Seiten 37–42, April 1996.
- [Tanaka 00] K. Tanaka, N. Abe, M. Ooho, H. Taki. Registration of Virtual Environment Recovered from Real One and Task Teaching. Tagungsband: *Proceedings of the IROS 2000*, Seoul, Korea, 2000.



- [Tatsuno 96] J. Tatsuno, S. Matsuyama, Y. Kokubo, K. Kawabata, H. Kobayashi. Human Friendly Teaching for Industrial Robots. Tagungsband: *IEEE International Workshop on Robot and Human Communication*, Seiten 456–460, 1996.
- [Terrillon 98] J.-C. Terrillon, M. David, S. Akamatsu. Detection of Human Faces in Complex Scene Images by use of a Skin Color Model and of Invariant Fourier-Mellin Moments. Tagungsband: *Fourteenth International Conference on Pattern Recognition*, Band 2, Seiten 1350–1355, 1998.
- [Terzopoulos 88] D. Terzopoulos, A. Witkin, M. Kass. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial Intelligence*, 36:91–123, 1988.
- [Theis 01] C. Theis. Stereoskopische Lokalisierung und Erkennung von Objekten im Greifraum eines autonomen Montageroboters. Diplomarbeit, Ruhr Universität Bochum, Institut für Neuroinformatik, Lehrstuhl für theoretische Biologie, 2001.
- [Tonko 97] M. Tonko, J. Schurmann, K. Schafer, H.-H. Nagel. Visually Servoed Gripping of a used Car Battery. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1:49–54, 1997.
- [Triesch 01] J. Triesch, Chr. von der Malsburg. A system for person-independent hand posture recognition against complex backgrounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1449–1453, Dezember 2001.
- [Tsai 87] R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.
- [Tso 95] S. Tso, K. Liu. Automatic Generation of Robot Program Codes from Perception of Human Demonstration. Tagungsband: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Seiten 23–28, 1995.
- [Tsuda 99] M. Tsuda, T. Takahashi, H. Ogata. Creating an Assembly-Task Model by Human Demonstration. Tagungsband: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Band 4, Seiten 1007–1013, 1999.
- [Tung 95] C. Tung, A. Kak. Automatic Learning of Assembly Tasks using a Dataglove System. Tagungsband: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, Seiten 1–8, 1995.
- [Ude 94] A. Ude, H. Bröde, R. Dillmann. Object Localization using Perceptual Organization and Structural Steropsis. Tagungsband: *Proceedings of the Third International Conference on Automation, Robotics and Computer Vision, Singapur*, Seiten 197–201, November 1994.
- [Ude 96] A. Ude. *Rekonstruktion von Trajektorien aus Stereobildfolgen für die Programmierung von Roboterbahnen*. Dissertation, Universität Karlsruhe, 1996. Erschienen in: VDI Verlag, Fortschr. Ber. VDI Reihe 10 Nr. 448. Düsseldorf.



- [Uhlen 95] T. Uhlin, P. Nordlung, A. Maki, J. Eklundh. Towards an Active Visual Observer. Tagungsband: *Fifth International Conference on Computer Vision*, Seiten 679–686, 1995.
- [Union 01] Europäische Union. *Information Society Technologies Programme: Cognitive Vision Systems*, Februar 2001. <http://cogvis.nada.kth.se>.
- [Vincze 00] M. Vincze. Dynamics and System Performance of Visual Servoing. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco*, Seiten 644–649, April 2000.
- [Virtex 00] Virtex. *Cyberglove Specifications*, 2000. <http://www.virtex.com>.
- [Virtual 95] *Virtual Technologies Inc. CyberGlove<sup>TM</sup>*. Palo Alto, California, USA, 1995.
- [Voyles 95] R. Voyles, P. Khosla. Tactile Gestures for Human/Robot Interaction. Tagungsband: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Band 3, Seiten 7–13, 1995.
- [Voyles 99a] R. Voyles, P. Khosla. Gesture-Based Programming: A Preliminary Demonstration. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, Michigan*, Seiten 708–713, Mai 1999.
- [Voyles 99b] R. Voyles, J. Morroy, P. Khosla. Gesture-Based Programming for Robotics: Human Augmented Software Adaption. *IEEE Intelligent Systems*, Seiten 22–29, November/Dezember 1999.
- [Wachter 97] S. Wachter, H.-H. Nagel. Tracking of Persons in Monocular Image Sequences. *Proceedings of the Nonrigid and Articulated Motion Workshop*, Seiten 2–9, 1997.
- [Weckesser 97] P. Weckesser. *Aktiver Einsatz eines Multisensorsystems zur Exploration der Umwelt mit einem mobilen Roboter*. Dissertation, Universität Karlsruhe, 1997. Erschienen in: VDI Verlag, Fortschr. Ber. VDI Reihe 8 Nr. 654. Düsseldorf.
- [Wheeler 95] M. Wheeler, K. Ikeuchi. Sensor Modelling Probabilistic Hypothesis Generation and Robust Localization for Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):252–265, 1995.
- [Wilcox 92] L. Wilcox, M. Bush. Training and Search Algorithms for an Interactive Word Spotting System. Tagungsband: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Band II, Seiten 97–100, 1992.
- [Wilson 94] R. Wilson. *Modeling and Calibration of Automated Zoom Lenses*. Dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.
- [Wu 01] Y. Wu, T. Huang. Hand Modeling, Analysis and Recognition. *IEEE Signal Processing Magazine*, 18(3):51–60, Mai 2001.
- [Xu 98] G. Xu, T. Sugimoto. Rits Eye: A Software-Based System for Real-Time Face Detection and Tracking using Pan-Tilt-Zoom Controllable Camera. Tagungsband: *Fourteenth International Conference on Pattern Recognition*, Band 2, Seiten 1194–1197, 1998.

- [Yamamoto 91] M. Yamamoto, K. Koshikawa. Human Motion Analysis based on a Robot Arm Model. Tagungsband: *IEEE Transactions on Computer Vision and Pattern Recognition*, Seiten 664–665, 1991.
- [Yang 98] J. Yang, W. Lu, A. Waibel. Skin-Color Modeling and Adaptation. Tagungsband: *Proceedings of ACCV, Hong Kong*, Band 2, Seiten 687–694, 1998.
- [Yeasin 00] M. Yeasin, S. Chaudhuri. Toward automatic robot programming: learning human skill from visual data. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 30(1):180–185, Februar 2000.
- [Yeasin 97] M. Yeasin, S. Chaudhuri. Automatic Robot Programming by Visual Demonstration of Task Execution. Tagungsband: *ICAR 97*, 7.-9. Juli 1997.
- [Yuan 97] X. Yuan, H. Sun. Mechanical Assembly with Data Glove Devices. Tagungsband: *IEEE Canadian Conference on Engineering Innovation: Voyage of Discovery.*, Band 1, Seiten 177–180, 1997.
- [Zamperoni 89] P. Zamperoni. *Methoden der digitalen Bildsignalverarbeitung*. Vieweg und Sohn Verlagsgesellschaft, Braunschweig, Wiesbaden, 1989.
- [Zhang 00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.
- [Zhang 99] J. Zhang, Y. von Collani, A. Knoll. Interactive Assembly by a Two-Arm-Robot Agent. *Robotics and Autonomous Systems*, 1(29):91–100, 1999.
- [Zöllner 01] R. Zöllner, O. Rogalla, J. Zöllner, R. Dillmann. Dynamic Grasp Recognition within the Framework of Programming by Demonstration. Tagungsband: *The 10th IEEE International Workshop on Robot and Human Interactive Communication (Roman)*, Seiten 418–423, 18.-21. September 2001.

# Index

- Abbildung, 10, 48, 52, 54, 169
- Ablaufbeschreibung, 59
- Abstraktion, 48, 54
- Abtastung
  - Bildabtastung, 73
  - Konturabtastung, 109
- Aktiver Beobachter, 20
- Aktives Sehen, 20
- Algorithmus
  - Condensation*-Algorithmus, 27
  - Allgemeine Hough-Transformation, 70, 74, 80
  - Baum-Welch-Algorithmus, 122
  - Farbhistogramm, 70, 77, 80, 129
  - Gamut-Mapping, 21, 100
  - Graphanpassung, 79, 128
  - Graphendetektion, 70, 72
  - Hough-Transformation, 128
  - Konturanpassung, 91
  - Musteranpassung, 70, 75, 80
  - Schnelle Fourier-Transformation, 109
  - Viterbi-Algorithmus, 118
- Alphabet, 114
- Anweisung, 10
- Aufmerksamkeitsfokus, 50, 123
- Ausführungsumgebung, 10, 48, 50, 162, 171
- Ausführung, 49, 53, 54
- Ausführungsumgebung, 55
- Avatar, 16
  
- B-Splines*, 71
- Basisfunktion, 71
- Bedeutung, 4
- Beleuchtung, 183
- Beobachtung, 10, 13, 43, 47, 52, 54, 56, 60, 169, 175
  - aktive, 20
- Berechnungsaufwand, 15
- Bewegungsverfolgung, 14, 59–61, 70, 88, 102, 132–137
  - Visual Servoing*, 14, 20
  - in Echtzeit, 14, 20
  - Zykluszeit, 14
- Bild
  - Bildfolgenanalyse, 15
  - Halbbild, 133
  - Tiefenbild, 29
- Binarisierung, 91
- Brennweite, 19, 97, 177
  
- Daten, 4
- Datenanzug, 16
- Datenbank, 54
- Datenhandschuh, 16, 25, 26, 29, 34, 37, 50, 51, 102, 107, 140, 141, 170
  - Kalibrierung, 104
- Dehnmessstreifen, 51
- Dehnungsmessstreifen, 16, 17, 104
- Dialog, 24, 66
- Differenzbild, 14, 21
- Digitalisierung, 55
- Disamiguisierung, 27
- Disparität, 14
- Drehmodul, 51, 55, 97, 128, 177
  
- Eigenverfahren, 19
- Endlicher Automat, 59
- Epipolarlinien, 84
- Ereignis, 56, 123, 175
- Ergonomie, 13
- Exoskelett, 17
- Experiment, 170
  
- Farbe
  - Farbreduktion, 77

- Histogramm, 78
- Segmentierung, *siehe* Segmentierung
- Farbfilter, 183
- Farbkonstanz, *siehe* Kamera
- Farbraum
  - CMY-Farbraum, 179
  - HLS-Farbraum, 100, 179
  - HSY-Farbraum, 21
  - RGB-Farbraum, 21, 100, 179
  - RG-Farbraum, 181
- Fehler
  - Fehler erster Art, 128
  - Fehler zweiter Art, 128
- Filter
  - Farbfilter, 137
  - Identitätsfilter, 116, 149
  - Kalman-Filter, 20, 27
  - Nachbarschaftsfilter, 116, 149
  - Start-Stop-Filter, 117, 149
- Fingerstellung, 16, 23
- Fokussierbewegung, 14
- Gaussuche, 76
- Generalisierung, 48, 54
- Geschlossene Welt, 49
- Geste, 10, 59, 65, 171
  - deiktisch, 66
  - Deskriptor, 109
  - dynamisch, 24, 66, 114, 147–161, 170
  - emblematisch, 66
  - Gestenerkennung, 18, 66, 70, 107–123, 141–161
  - Gestenhierarchie, 65
  - ikonisch, 65
  - metaphorisch, 66
  - statisch, 24, 27, 66, 107, 109, 141–147, 170
  - symbolisch, 66, 141
  - symbolische, 34
  - Taktgeste, 66
  - taktile, 34
  - Zeigegeste, 27, 42, 66
- Graue-Welt-Annahme, *siehe* Kamera
- Griff
  - Grifferkennung, 16, 18, 26, 58, 63, 70, 102–107, 140–141, 170
  - Griffhierarchie, 26, 63, 105, 140
  - Grifftyp, 26, 50, 63, 102
  - Kraftgriff, 63
  - Präzisionsgriff, 63
- Hand
  - dominante Hand, 42
  - Handkonfiguration, 65, 66, 141
  - Handmodell, 22, 59
  - Handverfolgung, 14, 22, 50, 61, 88–102, 133–137, 173
  - Handzeichen, 42
- Handhabung, 13, 41
- Handlung, 10, 59
  - Handlungsbeobachtung, 18, 30, 161
  - Handlungserkennung, 27, 42
  - Handlungsmodell, 59
  - Handlungsrepräsentation, 36
  - Interpretation, 47
  - kommandierende, 43
  - kommentierend, 44
  - performative Handlung, 43, 47, 55, 161, 169
  - Spontanhandlung, 49
  - Teilhandlung, 47
- Handlungserkennung, 27
- Handschrifterkennung, 26
- Haushalt, 42
- Hautfarbe, 21
- Heuristik, 54
- Hintergrundwissen, 170
- Industrie, 1
- Information, 4
- Informationsverarbeitung, 56
- Instruierung, 9, 41, 54, 170
- Intelligenz, 3
  - künstliche, 3
- Interaktion, 3, 9, 28, 33, 42, 66, 170, 171
  - mimisch, 173
  - multimodal, 38
- Interpretation, 43, 47, *siehe* Handlungsinterpretation, 48, 53, 54, 56, 172

- Interpretation einer Vorführung, 10
- Körperlichkeit, 3
- Kamera
- Datenblatt, 177
  - Deckenkamera, 14
  - Farbkamera, 14, 55
    - Graue-Welt-Annahme, 183
  - Farbkonstanz, 21, 99, 183–185
  - Grauwertkamera, 14
  - Kamerakalibrierung, 19, 82, 131
    - mit Referenzobjekt, 19
    - Selbstkalibrierung, 19
  - Kamerakopf, 50, 51, 60, 128
    - binokular, 14
    - Linsenverzerrung, 19
    - Lochkameramodell, 82
    - Multikamerasystem, 16
- Kantendetektor, 71, 89
- KAVIS, 58, 161
- Klassifikation, *siehe* Objekt-, Griff- oder Gestenerkennung
- Kognition, 41
- Kognitive Psychologie, 41
- Kommandierung, 42, 43, 55, 66
- Kommentierung, 10, 44, 55
  - durch Gesten, 49
  - verbal, 49, 172
- Konfidenzfaktor, 95
- Kontaktzustand, 29
- Kontext, 60
  - Kontextstabilität, 49
  - Kontextwechsel, 41, 49, 54
- Kontrollpunkt, 71
- Kontur, 71, 109
  - aktive Kontur, 19
  - Konturanpassung, 19, 89
  - Konturmodell, 21, 22
  - Konturverfolgung, 24, 133
- Koordinatensystem
  - Endeffektorkoordinatensystem, 61
  - Markerkoordinatensystem, 62
  - Weltkoordinatensystem, 61
- Korrektur, 53
- Korrespondenzproblem, 71, 84, 137
- Kraftrückkopplung, 18
- Lichtverhältnisse, 171
- Lokalisierung, *siehe* Objektlokalisierung
- Marker, 20, 91, 133
  - Leuchtdiode, 16, 91
- Matrix
  - DLT*-Matrix, 82
  - Regularisierungsmatrix, 90
- Merkmal
  - Kontur, 58
  - Merkmalssuche, 133
- Messfehler, 88
- Modell
  - Ansichtsmuster, 71, 127
  - Attribute, 58
  - Benutzermodell, 57
  - Ein-Hand-Modell, 49
  - Farbcharakteristika, 71
  - geometrisch, 58
  - Griffmodell, 63
  - Handlungsmodell, 57, 59, 60
  - Handmodell, 57, 59
  - Hidden Markov Modell*, 25, 34, 114–122, 148
  - Kante im Bild, 71
  - Konturmodell, 71, 74, 89, 127
  - Modellierung, 41, 57
  - Referenzmodell, 114, 150, 160
  - Robotermodell, 54
  - Schwellwertmodell, 119, 154
  - Sensormodell, 95
  - Umweltmodell, 52, 61
  - Weltmodell, 54, 57, 105, 123, 175
  - Zeit, 123
  - Zwei-Hand-Modell, 173
- MORPHA, 9
- Neuronales Netz, 20, 24, 27, 105, 107, 140, 141, 170
- Nyquistrate, 110
- Objekt, 54, 59, 70
  - Lokalisierung, 22, 70, 82, 84, 97, 131
  - Objektansicht, 58

- Objektdetektion, 70, 127
  - farbbasiert, 77
  - konturbasiert, 71, 72, 74
  - Musterbasiert, 75
- Objekteigenschaften, 58, 140
- Objekterkennung, 15, 19, 55, 58, 60
  - Methode, *siehe* Algorithmus
- Objektgeometrie, 65
- Objektklassifikation, 70
- Objektlage, 47, 175
- Objektmodell, *siehe* Modell, 105
- Objektreferenzierung, 42
- Objektverfolgung, 18, 20
- Objekterkennung, 127–131, 170
- Operator, 47, 170
  - elementarer kognitiver, 60, 69–123, 127–161, 172, 176
  - Kantenoperator, 73
  - Makrooperator, 53, 55
  - Operatorauswahl, 41
- Orthogonale, 72, 73, 133
- Positionsbestimmung, *siehe* Objektlokalisierung
- Positionssensor
  - magnetfeldbasierter, 15, 16, 25, 50, 52, 61, 88, 96, 133
  - manipulatorbasierter, 18
- Programmierung, *siehe* Roboterprogrammierung, 13
- Pyramidenverfahren, *siehe* Gausssuche
- Rückfrage, 60
- Rückfragen, 172
- Regelung, 56
- Region
  - Region-Growing*, 21
  - Analyse, 91, 133
  - Flächeninhalt, 91
  - Handregion, 109
  - Hautfarbregion, 99
  - Kompaktheit, 91
  - Schwerpunkt, 91
  - Umfang, 91
- Registrierung, 10, 86, 123, 171, 175
- Rekonstruktion, *siehe* Objektlokalisierung
- Repräsentation, 47, 58, 169
- Roboter, 50
  - Assistent, 2, 3, 10, 13, 56
  - Industrieroboter, 2
  - mobiler Roboter, 14
  - Serviceroboter, 2, 10
- Roboter manipulator, 13, 14
- Roboterprogramm, 54
- Roboterprogrammierung, 4
  - durch Vormachen, 8, 9, 28, 169
    - Prozessphasen, 49
  - durch *Teach-In*, 13
- Ebenen, 5
  - Aufgabenebene, 5
  - Gelenkebene, 5
  - Manipulatorebene, 5
  - Objektebene, 5
- manuell, 4, 7
- symbolisch, 7
- System
  - ALAT*, 32
  - APO*, 29
  - ARP*, 33
  - CORA*, 35
  - ETAR*, 7
  - GBP*, 34
  - IPOR*, 33
  - LFO*, 31
  - MA*, 33
  - SKORP*, 7
  - TLT*, 31
  - von *Onda*, 8
  - von *Shepherd*, 7
- textuell, 4
  - aufgabenorientiert, 6
- Sakkaden, 14, 20
- Schablonenanpassung, 14, 19, 75
- Schnittstelle, 66
  - blickwinkelbasiert, 2, 173
  - gestenbasiert, 2
  - graphische, 54
  - mimisch, 173
  - multimodal, 2, 9, 13



- sprachbasiert, 2
- Segmentierung
  - Handlungssegmentierung, 43, 47, 52, 54
  - nach Farbe, 77, 129, 137
  - nach Hautfarbe, 21, 35, 99–102, 109, 114
  - nach Helligkeit, 91, 133
- Sehen
  - aktives, 20
  - anregendes, 20
  - Sehen und Handeln, 20
- Sensorik, 14–18, 47, 50–52, 55–56, 170
  - aktive, 36, 60
  - bildgebende, 14
  - Fusion, 89, 95, 133, 170
  - haptische, 141
  - Kraftsensorik, 173
  - magnetfeldbasiert, 16
  - Rauschen, 49, 88, 113
- Simulation, 48, 53, 54, 161
- Softwarearchitektur, 52, 56
- Symbol, 48, 53, 59
- Syntax, 4
- Systemhypothese, 49
- Szenenanalyse, 18, 19, 69–87, 127
- Szeneninterpretation, 15
  
- Teilzielzerlegung, 41
- Theorie
  - Theorie des rechnenden Sehens, 20
  - Theorie dynamischer Systeme, 35
- Tiefenrekonstruktion, *siehe* Objektlokalisierung
- Trajektorie, 28, 43, 49, 53, 61, 66, 105
- Transformation, 61
  - affine, 72
  - Fourier-Transformation, 109
  - Hough-Transformation, 19, 70, 74
  
- Umweltmodell, 171
- Umweltzustand, 47
  
- Verdeckung, 128
- Verdeckungen, 15
  
- Vergenzbewegung, 14
- Verhalten
  - Verhaltensorganisation, 35
- Virtueller Raum, 36
- Vorführgerät, 13
- Vorführungsumgebung, 10, 49, 50, 54, 170
- Vorführung, 10, 41, 47, 54, 161, 170
  - Analyse, 169
  - graphische, 13
  - physische, 13, 36
  - Randbedingung, 49
  - symbolische, 13
- Vorführungsumgebung, 50
  
- Wissen, 4, 173
  - Ausführungswissen, 54
  
- Zeichen, 4
- Zeichensprache, 25
- Zeitmessung, *siehe* Modell der Zeit
- Zielgerichtetheit, 41