

Intuitive Schnittstelle

eine neue Computerumgebung für die Planung
bildkünstlerischer Prozesse

Michael Hoch

Michael Hoch
Email: micha@khm.de

Vollständiger Abdruck
der vom Fachbereich Informatik der Universität Dortmund
zur Erlangung des Grades eines Doktors der Naturwissenschaften
genemigten Dissertation.

Betreuer:	Prof. Dr. Heinrich Müller Lehrstuhl Graphische Systeme Universität Dortmund
	Prof. Dr. Siegfried Zielinski Fächergruppe Kunst- und Medienwissenschaften Kunsthochschule für Medien Köln

Tag der mündlichen Prüfung: 9. März 1999

Dekan/Dekanin: Prof. Dr. Heinrich Müller

Gutachter: Prof. Dr. Heinrich Müller
Prof. Dr. Wolfgang Banzhaf
Prof. Dr. Siegfried Zielinski

Vorwort

Der Einsatz des Computersehens zur Erschließung neuer Interaktionsmöglichkeiten mit rechnerbasierten Systemen hat in den letzten Jahren stark anwachsendes Forschungsinteresse gefunden. In dem vorliegenden Werk wird ein Interaktionsszenario auf dieser Grundlage vorgeschlagen, existierenden Ansätzen gegenübergestellt, die technische Lösung entwickelt und als Prototypsystem implementiert. Das Interaktionssystem wird exemplarisch im Rahmen eines prototypischen Filmplanungssystems eingesetzt und auf seine Akzeptanz hin evaluiert.

Hinter dieser nüchternen Zusammenfassung verbirgt sich eine außerordentlich vielsichtige Darstellung der Gegebenheiten, Möglichkeiten, aber auch Grenzen der computersehbasierten Interaktion. Herr Hoch, um dessen Dissertation es sich bei dem vorliegenden Buch handelt, hat diese in seiner Zeit als Mitarbeiter an der Kunsthochschule für Medien zu Köln angefertigt. Diese für einen Diplominformatiker zum Teil ganz andere Welt des Denkens und Vorgehens geht in diese Arbeit wahrnehmbar ein. Sie bietet eine sehr gut lesbare, umfassende, fundierte Einordnung des Themas aus historischer, philosophischer, psychologischer und technischer Sicht. Diese interdisziplinäre Sicht begründet nachvollziehbar den Weg, der in der Arbeit beschritten wird. Ausgehend davon wird das Interaktionsszenario der Dissertation entwickelt, das die aufgestellte Forderung einer "Intuitiven Schnittstelle" erfüllen soll.

Dieses Szenario wird in einer Systemarchitektur umgesetzt, die den Rahmen für verschiedene interessante technische Lösungen bietet und die in ein außerordentlich robustes Prototypsystem umgesetzt wurde. Aus technischer Sicht ist Herrn Hoch eine Lösung des Personenverfolgungsproblems mit Neuheitswert gelungen, die mit äußerst geringem Markierungsaufwand (Marker an der Brust) auskommt und die eine hohe Zuverlässigkeit auch unter nicht ganz idealen Randbedingungen, etwa die Anwesenheit weiterer Personen, aufweist.

Aus ergonomischer Sicht ist das Interaktionsszenario aus einer Kombination von Zeigen, Raumnutzung und Interaktion mit physischen Platzhalterobjekten als neuartig einzuordnen. Einen besonderen Schwerpunkt stellen die mit Methoden der empirischen Sozialforschung durchgeführten Untersuchungen des Interaktionskonzepts an einem interaktiven Filmplanungssystem dar. Aus diesen ist zwar zu schließen, daß das Ziel, eine bessere Interaktionsmöglichkeit für bildkünstlerische Prozesse zu schaffen, nur zum Teil erreicht wurde. Die ungeschönte Darlegung dieses Ergebnisses mit detaillierten Analysen eröffnet jedoch Perspektiven für die weitere Forschung in diese Richtung. Sie kann in diesem Sinn als wichtiger Schritt verstanden werden, für die Technik der computersehbasierten Interaktion angemessene Einsatzmöglichkeiten zu erschließen und so die häufig unkritische Begeisterung über die Möglichkeiten dieses Ansatzes der Mensch-Rechner-Interaktion in solide, tragfähige Bahnen lenken.

Dortmund, im Mai 1999

Heinrich Müller

Vorwort: Medium – eine Welt dazwischen

Die Untersuchung Michael Hochs wurde stark durch eine Beobachtung angeregt, die selbst im Alltag einer Kunsthochschule für Medien immer noch zu machen ist und bei Informatikern für Frustrationen sorgt: Es gibt eine quasi natürliche Schwellenangst von Künstlern gegenüber symbolischen Maschinen. Besonders bei visuell arbeitenden Künstlern sind die Vorbehalte gegenüber Artefakten und Prozessen, die einer anderen Struktur als der ihrer konkreten Einbildungskraft folgen, häufig noch enorm hoch. Michael Hoch analysierte als Ursache hierfür vor allem die traditionelle Konfiguration der Schnittstelle heraus, also jenes Ortes, an dem der Nutzer auf die Maschinen trifft und umgekehrt. Hemmend für die Interaktion von Bildkünstlern mit der Hardware und den Programmen seien hier vor allem die traditionellen Eingabegeräte, die eher der symbolischen Welt der Mathematik zugehörig seien und der Welt der sinnlichen Wahrnehmung abgewandt gegenüberstehen. Aus dieser Beobachtung heraus entwickelt der Autor die Forderung, daß die Schnittstelle in Zukunft großzügiger gestaltet, das heißt, stärker an die verschiedenen Bedürfnisse sehr unterschiedlicher Nutzer und Typen von Nutzern angepaßt werden müsse.

Geisteswissenschaftler würden es gemeinhin bei der ausformulierten Kritik und der Forderung nach Verbesserungen bewenden lassen. Als Informatiker und Techniker ging Michael Hoch aber beträchtlich weiter. Er entwickelte für einen spezifischen Benutzerkreis nicht nur das Modell einer alternativen Schnittstellen-Konfiguration, sondern unterzog dieses Modell darüberhinaus auch empirischen Tests, um die Richtigkeit und Tragweite seines Konzeptes überprüfen zu können. Im Ergebnis der Arbeit wird klar, daß es hier nicht um den Entwurf einer künftigen *universalen Schnittstelle* gehen kann, sondern um gezielte und notwendige Experimente für deren ausdifferenziertere Gestaltung im Interesse der Nutzer.

Bei der Formulierung der geisteswissenschaftlichen Annahmen, die in das Konzept der *Intuitiven Schnittstelle* eingehen, hat es sich Michael Hoch nicht leicht gemacht. Ausgehend von der These, daß Körper, Raum, Wahrnehmung und Subjektkonstituierung aufs engste miteinander verknüpft sind, hat er sich durch grundlegende Werkaspekte Merleau-Pontys gearbeitet, hat sich die in der Kinodebatte Anfang der siebziger Jahre entstandene Apparatus-Theorie angeeignet und sich mit dem komplexen Standardwerk von Yates zum Zusammenhang von Raumwahrnehmung und Erinnerung auseinandergesetzt. Die daraus gewonnenen Erkenntnisse führten zu einem in der Studie immer wieder betonten Kerngedanken: Für Computernutzer, die mit ihrer professionellen Tätigkeit insbesondere in der Produktion von sinnlichem Material zu Hause sind, wie z. Bsp. für Filmemacher, müsse die Konfiguration einer Schnittstelle angemessen sein, innerhalb derer sich die Akteure mit den Mitteln körperlicher Interaktion einbringen können (Gestik, Stimme, Bewegung im Raum).

Aus medienwissenschaftlicher Sicht spreche ich der Studie meine Hochachtung aus. Sie folgt einer originellen und einleuchtenden Arbeitshypothese zu einem Brennpunkt gegenwärtiger Medientheorie und -praxis, operationalisiert diese im Hinblick auf ein handhabbares Modell und gelangt schließlich bis zur Verifizierung/Falsifizierung einiger der Grundannahmen und zu ihrer Modifikation. Die transdisziplinäre Anlage der Untersuchung, die für Michael Hoch mit einem erheblichen Forscherrisiko verbunden war, ist beispielhaft und hoffentlich richtungsweisend für künftige Studien in dem spannenden Feld zwischen Informatik, Gestaltung und Kunst.

Siegfried Zielinski, Köln im Juni 1999

Zusammenfassung

Computergestützte Systeme für bildkünstlerische Prozesse sind oft durch eine spezielle Bedienstation mit ein oder mehreren Bildschirmen realisiert. Diese Systeme weisen einen hohen Grad an symbolischen Hindernissen auf, die sich negativ auf eine kreative Gestaltung auswirken können. Bei der traditionellen künstlerisch-/gestalterischen Arbeit ist man aber stets gegenüber einem „Werkstück“ situiert und gebraucht seine Muskelkraft. In der vorliegenden Arbeit werden daher Methoden skizziert und untersucht, die es gestatten, den Computer aus der herkömmlichen, wenig flexiblen Anordnung rund um den Monitor zu „befreien“ und in die alltägliche Umgebung des Nutzers zu integrieren. Es wird das Konzept einer Schnittstelle geschaffen, die sich von Maus und Tastatur und der Bildschirmarbeit als primär einer Tätigkeit loslöst, die spezielle Fähigkeiten im Umgang mit Maschinen und Programmen voraussetzt; eine Schnittstelle, die in die Interaktion zwischen Mensch und Maschine den realen Raum und mögliche Bezüge zur Dingwelt einbezieht. Im Gegensatz zu einer Vielzahl von traditionellen Computeranwendungen, wird bei der hier beschriebenen *Intuitiven Schnittstelle* die besondere Rolle des Körpers des Menschen in der Wahrnehmung betrachtet und versucht, die Situation bzw. das Situiertsein des Benutzers in seiner Umgebung explizit zu berücksichtigen.

Damit der Rechner über die eigene Armlänge hinaus bedient werden kann, gleichzeitig jedoch eine weitgehende Bewegungsfreiheit des Benutzers besteht, muß dessen Körperhaltung mit Verfahren der Mustererkennung identifiziert werden. Im Rahmen dieser Arbeit wurde ein videobasiertes Verfolgungssystem entwickelt, das es gestattet, die 3D-Position und Zeigerichtung einer Person im Raum in Echtzeit auf einem Standard-PC durchzuführen. Die Architektur dieses Softwaresystems stellt ein offenes, erweiterbares System dar, das insbesondere Aspekte der Echtzeitverarbeitung berücksichtigt. Das System basiert auf der Verfolgung farbiger, zusammenhängender Regionen unterstützt durch verschiedene modellbasierte Verfahren: Das in einem ersten Schritt zur Unterstützung der Bildverarbeitung verwendete einfache Armmodell lieferte nicht für jede Benutzer-Aktions-Kombination befriedigende Ergebnisse. Die Erweiterung des Systems um ein komplexeres Modell des Körpers und der Integration einer Wissensbasis mit zusätzlichen Regeln und Aktionen erlaubt schließlich eine robuste, kontinuierliche Verfolgung. Eine weitere Verbesserung auch der Segmentierung lieferte eine qualitative und quantitative Untersuchung von Beleuchtungseinflüssen, die zu einem automatischen Kalibrierungsverfahren führte.

Zur Überprüfung des Konzepts der *Intuitiven Schnittstelle* wurde ein Prototyp entwickelt und einer empirischen Untersuchung unterzogen. Die prototypische Anwendung realisiert einige wesentliche Komponenten eines Systems zur Filmplanung, bei dem Objekte der Szene durch Zeigegestik und Sprachkommandos ausgewählt werden können und neue Objekte in Abhängigkeit von der Position des Benutzers im Raum der Szene hinzugefügt werden können. Weiterhin können Realgegenstände in die Interaktionsabläufe einbezogen werden. Für die empirische Untersuchung wurden verschiedene Regisseure zu einzelnen Aspekten des Systems und der Schnittstelle befragt. Die Auswertung zeigt, daß, zumindest für die befragten Personen, eine Verwendung von Raum zur Schaffung einer sinnlichen, körperlichen Kommunikation nicht zwingend notwendig ist. Alle Befragten konnten sich aber ein derartiges System besonders bei der Planung der Ausstattung und bei der Planung von Szenenauflösungen mit Ausstatter, Kameramann und Beleuchter als sehr nützlich vorstellen. Somit kann der praktische Nutzen der Intuitiven Schnittstelle auf ein engeres Berufsfeld innerhalb der Filmplanung konkretisiert werden. Weiterhin werden im Ausblick dieser Arbeit konkrete Vorschläge für eine Weiterentwicklung angegeben, z.B. eine Variante in Form eines modellhaften Systems mit kleinen Figuren, welches besonders einfach zu bedienen wäre.

Danksagung

Die hier vorliegende Arbeit entstand während meiner Tätigkeit als künstlerisch-wissenschaftlicher Mitarbeiter an der Kunsthochschule für Medien in Köln in Zusammenarbeit mit dem Fachbereich Informatik der Universität Dortmund. Mein besonderer Dank gilt Prof. Heinrich Müller und Prof. Siegfried Zielinski für die wissenschaftliche Betreuung, sowie Prof. Wolfgang Banzhaf für sein Interesse an der Arbeit und die Begutachtung. Ebenso danke ich Prof. Georg Trogemann für die konzeptionellen Diskussionen.

Mein Dank gilt weiterhin den Regisseuren Jeanine Meerapfel, Reimund Boy, Peter Braatz, Peter Bringmann, Hellmuth Costard und Jörn Hintzer für ihre Bereitschaft, bei der empirischen Untersuchung mitzuwirken, sowie Egon Bunne und Werner Barg für ihre Unterstützung. Für Ihre Mitarbeit danke ich Gregor Hammes, Benno Graffmann, Thomas Kulessa und Harald Tillmann. Für anregende fachkundige Diskussionen danke ich Detlev Schwabe sowie Prof. Bernd Girod.

Für Elviz und Jolanthe

1	EINLEITUNG	1
1.1	Der Begriff Schnittstelle	1
1.2	Nutzung von Raum	2
1.3	Problemstellung	3
1.4	Arbeitsmethode	4
1.5	Resultate	5
2	SZENARIO	8
2.1	Körper und Raum	8
2.1.1	Wahrnehmungserfahrung	8
2.1.2	Körperraum und Außenraum	9
2.1.3	Der Leib und sein Spiegelbild	10
2.1.4	Der Mensch als wahrnehmendes Subjekt	11
2.2	Forschungsstand	12
2.2.1	Intuitive Interaktion	13
2.2.2	Reaktive Umgebungen	16
2.2.3	Reale Interaktion	18
2.3	Konzept der Intuitiven Schnittstelle	22
2.3.1	Überblick	23
2.3.2	Technischer Aufbau	24
2.3.3	Anwendungen	24
3	INTERAKTIONSTECHNIKEN	31
3.1	Gestaltung interaktiver Systeme	31
3.1.1	Elementare Interaktionsformen	32
3.1.2	Dialoge	33
3.1.3	Systemverhalten	34
3.2	Einordnung der Intuitiven Schnittstelle	35
3.2.1	Geforderte Funktionalitäten	35
3.2.2	Besonderheiten des Szenarios	35
3.3	Basisinteraktionen	36
3.3.1	Zeigegestik	36
3.3.2	Sprachkommandos	38
3.3.3	Menüauswahl	38
3.3.4	Rauminteraktion	39
3.3.5	Verwendung von Realobjekten	41
3.4	Dialoggestaltung	41
3.4.1	Umsetzung der Funktionalitäten	41
3.4.2	Ergonomie	43
3.4.3	Dialogablauf	43
4	PERSONENVERFOLGUNG	45
4.1	Forschungsstand	45
4.2	Segmentierung	47

4.2.1	Farbräume	48
4.2.2	Regionensegmentierung	49
4.2.3	Einfluß von Beleuchtung und Raumposition	51
4.2.4	Kalibrierung	53
4.3	3D Rekonstruktion	54
4.3.1	Geometrische Transformationen	54
4.3.2	Lochkameramodell	55
4.3.3	Kamerakalibrierung	56
4.3.4	Rücktransformation	59
4.4	Verfolgung	65
4.4.1	Kalman-Filterung	65
4.4.2	Verfolgen über Silhouette	67
4.5	Modellbasierte Verfahren	68
4.5.1	Einfaches Armmodell	69
4.5.2	Körpermodell	74
4.6	Einbeziehung realer Gegenstände	84
4.7	Leistungsdaten	85
4.7.1	3D-Rekonstruktion	85
4.7.2	Zeigeoperation	86
4.7.3	Geschwindigkeit der Verfolgung	86
4.7.4	Zuverlässigkeit	87

5 SYSTEMARCHITEKTUR **90**

5.1	Allgemeine Struktur eines Musteranalysesystems	90
5.2	Systemüberblick	92
5.2.1	Datenfluß	93
5.2.2	Wissensbasis	94
5.2.3	Kontrollstruktur	94
5.3	Objektorientierter Entwurf	96
5.3.1	Struktur des Entwurfs	96
5.3.2	Modellierung des Verfolgungssystems	96
5.3.3	Modellierung der physikalischen Umgebung	99
5.3.4	Frontend	100
5.3.5	Szenenbeschreibung	102
5.3.6	Zusammenspiel der Komponenten	104
5.4	Entwicklerschnittstelle	105

6 EMPIRISCHE UNTERSUCHUNG: TESTS EINES PROTOTYPS **109**

6.1	Empirische Sozialforschung	110
6.1.1	Methoden	110
6.1.2	Intensivinterview	111
6.1.3	Teilnehmende Beobachtung	112
6.1.4	Experiment	112
6.2	Prototyp zur intuitiven Filmplanung	113
6.2.1	Systemaufbau	114
6.2.2	Beispielszene	117
6.2.3	Manipulationsmöglichkeiten	117
6.2.4	Sprachkommandos	120
6.3	Forschungsplan	121

<i>Inhalt</i>	<i>xi</i>
6.3.1 Konzeptualisierung	121
6.3.2 Untersuchungsplan	123
6.3.3 Leitfaden	125
6.4 Durchführung	127
6.4.1 Pretest	127
6.4.2 Befragte Personen	127
6.4.3 Interviewsituation	129
6.4.4 Auswertung	130
 7 AUSBLICK	 135
 8 LITERATUR	 138

1 Einleitung

Ein Computersystem, das mit den herkömmlichen technischen Accessoires für den Nutzer ausgestattet ist, ist weniger imstande, die Person wahrzunehmen, die mit ihm arbeiten möchte, als moderne Toiletten oder Außenlichter mit Bewegungsmeldern. Und obwohl die Erweiterung des Computers um die Fähigkeit, seine Umwelt zu registrieren und auf sie aktiv zu reagieren, bei einem solchen komplexen technischen System naheliegen würde, hat sich die Apparatur der Schnittstelle in den letzten 15 Jahren kaum verändert; die Kommunikation zwischen Nutzer und Computer geschieht in manueller Hinsicht fast ausschließlich über Maus und alphanumerische Tastatur. Dabei wäre es in vielen Anwendungssituationen sehr hilfreich, den Computer über die eigene Armlänge hinaus benutzen zu können. In der vorliegenden Arbeit werden Methoden skizziert und untersucht, die es gestatten, den Computer aus der herkömmlichen, wenig flexiblen Anordnung rund um den Monitor zu „befreien“ und in die alltägliche Umgebung des Nutzers zu integrieren. Dazu soll der Rechner seinen Benutzer wahrnehmen und auf dessen äußere Bewegungen und Artikulationen reagieren, ihn faktisch also sehen und hören können. Es wird das Konzept einer Schnittstelle geschaffen, die sich wegbewegt von Maus und Tastatur und der Bildschirmarbeit als primär einer Tätigkeit, die spezielle Fähigkeiten im Umgang mit Maschinen und Programmen voraussetzt; eine Schnittstelle, die in die Interaktion zwischen Mensch und Maschine den realen Raum und mögliche Bezüge zur Dingwelt einbezieht.

1.1 Der Begriff Schnittstelle

Der Engpaß beim Umgang mit dem Rechner ist häufig die Schnittstelle, d.h. im technologischen Sinne die Sprachen, Programme und Geräte, die dem Benutzer für den Umgang mit einem Datenverarbeitungssystem zur Verfügung gestellt werden. Für diese Arbeit wird die Schnittstelle definiert als der Ort, an dem die Koppelung zwischen den unterschiedlichen Systemen Mensch und Computer organisiert wird und stattfindet. Sie soll die Übersetzungs- und Vermittlungsfunktion zwischen der vom Benutzer eingegebenen Befehlsfolge und den internen Repräsentationen eines Computerprogramms bilden (Halbach 94). Zur Schnittstelle gehören auf Seiten der Hardware die Eingabegeräte, wie etwa Maus und Tastatur, sowie auf Seiten der Software die Benutzeroberfläche: das, was der Benutzer vom laufenden Programm überhaupt sieht oder hört, also sinnlich wahrnimmt. Eingabegeräte und die Gestaltung der Oberfläche definieren die Nutzungs- und Interaktionsmöglichkeiten zwischen Mensch und Maschine, die je nach Problemstellung und Anwendungsbereich mehr oder weniger geeignet sein können. So ist beispielsweise die Maus zur Manipulation von Bildern und Grafiken besser geeignet als die Tastatur. Bei komplexeren Problemen, wie z.B. die Animation von Objekten in einer 3D-Szene, muß das Problem in zahlreiche Teilaspekte untergliedert werden, um noch mit Maus und Tastatur gelöst werden zu können. Hierbei erfordert die Bedienung des Computers besondere Aufmerksamkeit, und es ist ein ständiges Fokussieren und Defokussieren zwischen der Lösung der Aufgabe und der Benutzung der Werkzeuge der Benutzeroberfläche notwendig. Dieses Dilemma wird auch mit neuartigen Hardware-Schnittstellen wie dem Grafiktablett, dem Datenhandschuh oder gar Datenhelm nicht gelöst, sondern lediglich verschoben. Die Entweder/Oder-Mentalität (ist der Lichtgriffel besser als das Grafiktablett?) geht davon aus, daß es eine optimale Lösung gäbe. Dabei wird nicht berücksichtigt, daß die Nutzer in ihren Zugängen zur und Anforderungen an die Technik verschieden sind, daß sich die Arbeitssituationen je nach Bedarf ändern können und die Kanäle für den Zugang

zum Medium (wie Maus, Bildschirm, Sprache, Gestik etc.) nicht eindimensional limitiert, sondern vielfältig ausgefächert sein sollten, je nachdem, wie es die unterschiedlichen Nutzertypen einfordern. Eine universelle Schnittstellenkonfiguration kann es daher nicht geben. Eine intuitiv zu benutzende Schnittstelle wird prinzipiell eine multimodale sein, die es einerseits erlaubt, über mehrere verschiedene Kanäle mit dem Rechner zu kommunizieren und es ebenfalls zuläßt, die gleiche Information auf semiotisch verschiedene Art und Weise, in unterschiedlichen Ausdrucksformen (einen Bericht zum Beispiel als Audio- oder Videoreport oder als Diagramm) zur Kenntnis zu nehmen.

Bei der Gestaltung von Mensch-Maschine-Schnittstellen beschäftigen sich Ingenieure, Gestalter und Informatiker hauptsächlich mit der Frage, mit welchen Mitteln es dem Benutzer erleichtert werden kann, den in seiner Konfiguration festgelegten Rechner zu benutzen und wie die Reaktionszeit der menschlichen Interaktion optimiert werden kann. Statt dessen könnte man sich aber auch umgekehrt fragen, wie der Computer so verändert werden kann, daß er adäquater mit den Benutzern umgehen könnte, damit der Mensch intuitiv auch ohne dazwischengeschalteten Lernvorgang und symbolische Interaktionen einen Computer bedienen könnte. Der Austausch fände so auf einer nichtlogischen Ebene statt. Eine intuitive Schnittstelle, die sich in den alltäglichen Lebensraum des Menschen integriert –idealerweise eine Schnittstelle, die sich als solche gar nicht zu erkennen gibt–, kann dazu beitragen, den Umgang mit dem Rechner zu erleichtern und die Schwellenangst gegenüber der symbolischen Maschine abzubauen. Die Interaktion mit dem Interface geriete dadurch in den Hintergrund und würde nicht mehr im Fokus der Interaktion mit dem Medium stehen (Halbach 94). Neben Fragen der Ergonomie ermöglicht gerade die natürliche – im Sinne von gewohnter – Umgebung, in die der Computer integriert ist, einen ungezwungenen Umgang. Um zur Verdeutlichung eine Analogie aus dem Alltagsleben zu benutzen: Die Bedienung einer Waschmaschine oder Mikrowelle, die oft mehr als einen Computer enthalten, bedeutet heute für die meisten Menschen kein wirkliches Hindernis mehr. Im Gegensatz zum Ingangsetzen einer Waschmaschine begibt sich der Computernutzer, wenn er sich vor den Rechner setzt, aus seiner natürlichen Umgebung heraus in eine künstliche Umgebung. Hingegen könnte der Umgang weicher gestaltet werden, wenn der Nutzer in seiner vertrauten Umgebung verbliebe und der Computer auf ihn und seine Anweisungen reagierte.

Ziel ist also die Konstruktion einer Schnittstelle, die einen möglichst großen physischen Bewegungsspielraum des Benutzers erlaubt und ihn nicht durch das Anlegen von technischen Prothesen (Datenhandschuhe, Brillen etc.) behindert. Zu diesem Zweck können zum Beispiel Kameras verwendet werden, welche die Szene beobachten. Die Aufgabe des Rechners bzw. des Programms besteht nun darin, mittels Bildverarbeitung aus den Kamerabildern zu erkennen, wie eine Person innerhalb der definierten Szene gegenüber der Maschinerie agiert und welche Art von Befehlen die Person ihr durch demonstrative Gestik erteilt. Zusätzlich zur Bildverarbeitung können einfache Befehle auch mittels Sprachverarbeitung erkannt werden. Die Bildausgabe wiederum kann über eine Projektionswand in die Umgebung des Raumes integriert werden.

1.2 Nutzung von Raum

Im alltäglichen Leben erfahren wir die Welt nicht zuletzt durch unsere eigene Körpererfahrung bzw. gemäß Edmund Husserl durch eine aktive, bewußte „Leibbewegung“; erst durch unsere Bewegung relativ zu anderen Personen oder Objekten konstituieren wir also für uns Räumlichkeit und Wirklichkeit. Ein Medium, das auf die Vermittlung von Welt aus ist, muß genau diese Leibbewegung ermöglichen (Halbach 94). Damit der Benutzer den Eindruck hat, daß er das System durch die in die Außenwelt eingreifenden Leibbewegungen manipuliert, sind nach Halbach erstens eine adäquate Positionierung des Be-

nutzers relativ zu den in der virtuellen Welt enthaltenen Objekten und zweitens die Repräsentation des Benutzers in dieser Welt wichtig. In der hier beschriebenen Intuitiven Schnittstelle arbeitet der Benutzer vor einer umgebungsartigen Rückprojektionswand und ist damit nicht nur adäquat gegenüber der virtuellen Welt positioniert, sondern er befindet sich gleichermaßen in der Realwelt vor der Projektion, in der die intentionalen Bewegungen des Körpers raumgreifend stattfinden können. Der Benutzer ist damit auch in der virtuellen Welt durch seine eigene Aktion repräsentiert. Die starke Einbeziehung von Ort und Raum ist für das Sicheinlassen des Benutzers und damit für das Funktionieren der Schnittstelle ausschlaggebend: ein Prinzip, das beispielsweise beim Apparat Kino, wo der Besucher sich freiwillig für eine bestimmte Zeit an einen klar definierten Ort begibt, um Film zu erleben, sehr gut funktioniert (Cha 80), bei den meisten Anwendungen der *Virtual Reality* aber nicht funktioniert, da der Benutzer seines gewohnten Körpererlebnisses enthoben und durch technische Prothesen (Datenhandschuh und -brille) in seiner Bewegungsfreiheit wie eine Marionette manipuliert wird. Die Umgebung der Intuitiven Schnittstelle, in der sich der Benutzer bewegt, ist nicht der Raum um den Computer, sondern der Realraum selbst, in den der Computer integriert ist. Die Schnittstelle ist hierbei nicht nur die Bildschirmoberfläche, das künstliche Gesicht des Computers, sondern die gesamte Interaktionsumgebung wird zum Interface.

Eine wichtige intellektuelle Kapazität des Menschen ist es, räumliche Bezüge herzustellen: Simonides von Keos (556-468 v. Chr.) hatte die Fähigkeit, Teile eines gesprochenen Vortrags mit bestimmten Orten im Tempel zu verbinden; er ging in Gedanken im Tempel umher und erinnerte sich so an die entsprechenden Teile seiner Rede (Yates 66). Diese Technik wurde als *Ars Memoriae* bezeichnet; die ersten Jesuiten in China nannten diesen Prozeß "Bau von Palästen des Geistes". Eher gegensätzlich verhält es sich beispielsweise bei den Schnittstellen zu modernen Datennetzen wie dem Internet: Sogenannte Browser wie Netscape lassen keine räumlichen Bezüge zu den abstrakten virtuellen Objekten bzw. Dokumenten zu, so daß eine Navigation nicht einfach ist. Hilfreich wäre das Herstellen solcher Bezüge zu virtuellen Welten im Datennetz, die sich im realen Raum, d.h. zwischen Computer und Benutzer, materialisieren. Eine intuitive Schnittstelle ist daher auch eine solche, die es dem Benutzer erlaubt, sich frei im Raum zu bewegen und Bezüge zwischen Objekten im Datenraum und Objekten im Realraum herzustellen.

1.3 Problemstellung

Das Problem, das in dieser Arbeit angegangen wird, ist folgendes: Damit der Rechner über die eigene Armlänge hinaus bedient werden kann, gleichzeitig jedoch eine weitgehende Bewegungsfreiheit des Benutzers besteht, muß dessen Körperhaltung mit Verfahren der Mustererkennung identifiziert werden. Der Computer soll in der Lage sein zu entscheiden, ob sich eine Person im Raum aufhält, wo sie sich aufhält und ob sie Kommandos an den Rechner gibt oder sich nur frei bewegt. Nicht behandelt und damit auch nicht erkannt werden sollen eher subtile Gestiken des Menschen oder andere für Steuerkommandos schwer identifizierbare Interaktionsformen wie etwa Gefühle oder Mimik. Die dabei zu lösenden wissenschaftlichen Probleme unterteilen sich in die Bildverarbeitung, genauer: die Segmentierung von relevanter Information aus den Videobildern der Kameras, die Verfolgung der Körperbewegungen der Nutzer (das sind die Algorithmen zur Erkennung von Raumposition, Körperhaltung und Zeigegestiken einer Person) und die Interaktion mit dem System, d.h. die Software-Schnittstelle, die einen intuitiven Umgang ermöglichen soll. Beim Entwurf eines Bildverarbeitungssystems ist neben der besonderen Situation der Verfolgungsaufgabe auch eine minimale Bildrate von 12 Bildern pro Sekunde zu berücksichtigen, damit der Benutzer den Eindruck der Interaktivität behält (in Stary 96 werden 10 Bilder pro Sekunde als unterste Grenze angegeben). Für einen flexiblen Einsatz sollte diese Bildrate auf Standard PC erreichbar sein, weshalb sowohl

verschiedene leistungsfähige aber komplexe Mustererkennungsalgorithmen keine Anwendung finden, als auch in der Literatur bekannte allgemeingültige Systeme zur Bildverarbeitung ausscheiden, da der interne Verwaltungsaufwand für die Aktualisierung der Datenstrukturen zu groß ist, als daß eine Echtzeitverarbeitung möglich wäre.

Computergestützte Systeme für bildkünstlerische Prozesse sind oft durch eine spezielle Bedienstation mit ein oder mehreren Bildschirmen realisiert. Beispiele hierfür sind Effektgeräte zur Nachbearbeitung von Videomaterial (z. B. *Flint*), Softwaresysteme zur Generierung von Computeranimationen (z. B. *SOFTIMAGE*) oder Systeme zum Schneiden von Filmmaterial (z. B. *Avid*). Diese Systeme weisen einen hohen Grad an symbolischen Hindernissen auf, die sich negativ auf eine kreative Gestaltung auswirken können. Wenn ich aber gestalten will, kreativ formen will, so bin ich traditionell stets gegenüber einem „Werkstück“ situiert und gebrauche meine Muskelkraft. Ein Hauptaugenmerk dieser Arbeit soll daher in der Untersuchung von Methoden liegen, Raum und Realobjekte in das Interface einzubeziehen, um die Lebhaftigkeit des Benutzers zu gewährleisten. Ein intuitiver, im Sinne von gewohntem Umgang, soll eine Entspezialisierung der Schnittstelle bewirken, es also ermöglichen, die Kommunikation mit dem Computer in die natürliche Umgebung einzugliedern. Zur Überprüfung dieser Methoden ist ein Prototyp zu entwickeln und einer empirischen Untersuchung zu unterziehen. Der Prototyp sollte eine Anwendung aus dem Bereich künstlerischer Bildproduktion, wie etwa Film, realisieren, deren Überprüfung nach Möglichkeit Hinweise für eine Weiterentwicklung oder sogar eine konkrete Systembeschreibung liefert.

1.4 Arbeitsmethode

Das Manko vieler Forschungsprojekte ist, daß durch die starke Orientierung an Einzelprojekten und Einzelfachforschung die übergreifenden Probleme des informations- und systemtechnologischen Zeitalters, insbesondere auch ihre sozialwissenschaftlich und sozial- wie kulturphilosophisch zu erfassenden Aspekte, kaum betrachtet werden. Ropohl schreibt hierzu: *„Die verschiedenen Technikwissenschaften, (wie Bauingenieurwesen, Maschinenbau, Elektrotechnik usw.) konzentrieren sich in der Regel zu sehr auf die mannigfachen Gegenstände ihres Spezialgebietes, als daß sie in der Lage wären die Grenzen ihres Bereiches zu überschreiten und allgemeine Probleme der Technik ins Auge zu fassen;...“* (Ropohl 79, S. 27). Im Unterschied zur Technologie, bezieht sich Technik immer auf einen Objektbereich, d.h. auf künstliche Objekte, also Artefakte, von Menschen erzeugt und für bestimmte Zwecke verwendet, z.B. Steinbrocken mit einer scharfkantigen Schneide versehen und zu einem Faustkeilwerkzeug gemacht, Fasern gewebt und zu Kleidungsstücken verarbeitet, die dem Witterungsschutz oder auch dem Schmuckbedürfnis dienen, oder wenn aus einer Fülle verschiedener Baumaterialien und Aggregaten ein Kraftwerk erstellt wird, das elektrische Energie liefert (Ropohl 79, S. 31). *„Die Technik umfaßt die gegenständlichen Artefakte, deren Entstehung und deren Verwendung, wobei die Verwendung technischer Gebilde wiederum der Hervorbringung neuer Artefakte dienen kann. Das Beziehungsgeflecht zwischen Entstehungs-, Sach- und Verwendungszusammenhängen hat dabei eine naturale, eine humane und eine soziale Dimension: Technik ereignet sich zwischen der Natur, dem Individuum und der Gesellschaft.“* (Ropohl 79, S. 43).

Ein weitertragender Beitrag zu einem neuen Umgang mit dem Computer, der über die bloße Erstellung technischer Artefakte hinausgeht, kann daher nur durch eine fachübergreifende Behandlung des Themas erfolgen und bedeutet, daß auch sozialwissenschaftliche Aspekte, insbesondere humane (psychologische) und soziologische, betrachtet werden müssen. Im Kern liegen hier also zwei Forschungsprobleme vor: Das der Informatik und das der empirischen Sozialforschung. Zur Schaffung einer intuitiven Schnitt-

stelle wird daher in dieser Arbeit das entwickelte System einer sozialwissenschaftlichen Kontrolle in Form einer empirischen Untersuchung unterzogen.

1.5 Resultate

In dieser Arbeit wurde die notwendige Technologie zur Personenverfolgung bezüglich ihrer Zielsetzung der Schaffung einer intuitiven Schnittstelle unter Einbeziehung von Körper und Raum entwickelt. Die Zielgruppe wurde auf einen Benutzerkreis präzisiert, der aus dem Arbeitsbereich künstlerischer/gestalterischer Bildproduktion, z.B. einen Filmregisseur, stammt. Zusätzlich wurde diese Technik anhand eines Prototypen einer sozialwissenschaftlichen Kontrolle in Form einer empirischen Untersuchung unterzogen. Dazu wurden anhand einer Beispielanwendung zur Filmplanung verschiedene Regisseure zu einzelnen Aspekten des Systems und der Schnittstelle befragt. Die Ergebnisse, sowohl der technischen Entwicklung, als auch der Untersuchung, liefern Hinweise für die Verwendung dieser Technik im Hinblick auf ihren Bestimmungszusammenhang der Planung bildkünstlerischer Prozesse.

Intuitive Schnittstelle. Als Ergebnis dieser Arbeit lassen sich die Hauptunterschiede und die Anforderungen der intuitiven Schnittstelle im Vergleich zu traditionellen Systemen benennen. Sie führen zu anderen Lösungen bei Objekt- und Menüanwahlen und zu neuen Konzepten zur Einbeziehung von realem Raum, die bei traditionellen Systemen nicht zu finden sind. Die verschiedenen Interaktionstechniken werden in Kapitel 3 beschrieben.

- Das Zeigen auf eine große Projektionswand hat einen eher richtungsweisenden Charakter und ist nicht so exakt wie durch die Verwendung einer Maus.
- Die Positionierung des Benutzers in seiner Umgebung stellt eine neue Ebene bei der Interaktion dar, die ausgenutzt werden kann aber auch speziell berücksichtigt werden muß.
- Die Projektionswand hat auf Grund ihrer Größe eine andere Qualität im Erscheinungsbild als ein Computermonitor, durch die der Benutzer die Darstellung eines Menüs anders erfährt.
- Der Realraum und die Körperbewegungen des Benutzers können als Teil der Schnittstelle verwendet werden. Sie haben jedoch keine Entsprechung in traditionellen Computersystemen mit Maus und Tastatur.

Verfolgungssystem. Im Rahmen dieser Arbeit wurde ein videobasiertes Verfolgungssystem entwickelt, das es gestattet, die 3D-Position und Zeigerichtung einer Person im Raum in Echtzeit auf einem Standard-PC durchzuführen. Das System basiert auf der Verfolgung farbiger, zusammenhängender Regionen, sogenannter *Blobs*. Aufgrund stetiger Probleme, die bei der Verfolgung der Regionen unter verschiedenen Beleuchtungsverhältnissen und an verschiedenen Orten im Raum auftraten, wurde eine qualitative und quantitative Untersuchung von Beleuchtungseinflüssen durchgeführt, die zu einem automatischen Kalibrierungsverfahren führte. Das in einem ersten Schritt zur Unterstützung der Bildverarbeitung verwendete einfache Armmodell lieferte nicht für jede Benutzer-Aktions-Kombination befriedigende Ergebnisse. Die Erweiterung des Systems um ein komplexeres Modell des Körpers und der Integration einer Wissensbasis mit zusätzlichen Regeln und Aktionen erlaubt eine kontinuierliche Verfolgung. Die Ergebnisse sind auch hier nicht immer korrekt im Sinne einer meßbaren Genauigkeit, jedoch ist eine absolute Genauigkeit einer Erkennung in dem vorgestellten Szenario nicht entscheidend für das Funktionieren der Schnittstelle, vielmehr ist es wesentlich wichtiger, ständig „brauchbare“ Ergebnisse zu erhalten. Details zur Personenverfolgung finden sich in Kapitel 4.

Architektur. Die Softwarearchitektur der Intuitiven Schnittstelle stellt ein flexibles, erweiterbares System dar. Verfolgungssystem und Benutzerschnittstelle können mittels einer Serverstruktur und Netzwerkkommunikation auf verschiedenen Rechnern ablaufen. So kann für die Benutzerschnittstelle sowohl ein Standard-PC als auch eine Hochleistungs-Grafikworkstation mit spezieller 3D-Grafikunterstützung zur Generierung besonders realistischer Szenen verwendet werden. Die sogenannte Szenenbeschreibung erlaubt auf einfache Weise die Verknüpfung von virtuellen Objekten mit Objekten der realen Welt, d.h. den Dingen, die im Kamerabild verfolgt werden. Das System stellt für die Verfolgung eine Reihe von Basisobjekten zur Verfügung, die durch die Erweiterbarkeit des Systems um weitere Objekte ergänzt werden können, ohne das Gesamtsystem neu zu übersetzen. Mit Hilfe dieser Entwicklerschnittstelle wurden beispielsweise die Segmentierung auf Basis von Lookup-Tabellen und das komplexere Körpermodell integriert. Die Architektur mit den Basisobjekten wird in Kapitel 5 beschrieben.

Anwendungen. In einer Netscape-Anwendung verwaltet der Benutzer Informationseinheiten (Internet-Verweise) durch seine Körperbewegungen und -positionen im Raum nach dem Prinzip der *Ars Memoriae* (Yates 66). Die Anwahl von derartigen Verweisen geschieht über Sprachkommandos oder durch eine Zeigegestik. Die zweite Anwendung realisiert einige wesentliche Komponenten eines Systems zur Filmplanung, bei dem Objekte der Szene durch Zeigegestik und Sprachkommandos ausgewählt werden können und neue Objekte in Abhängigkeit von der Position des Benutzers im Raum der Szene hinzugefügt werden können. Weiterhin können Realgegenstände in die Interaktionsabläufe einbezogen werden. Die Umgebung der Intuitiven Schnittstelle, in der sich der Benutzer bewegt, ist hierbei nicht der Raum um den Computer, sondern der Realraum selbst. In drei weiteren Anwendungen, in Form von interaktiven Installationen, hat sich die Softwarearchitektur der Intuitiven Schnittstelle ebenfalls als nützlich erwiesen. Anwendungen und Installationen werden in Kapitel 2 beschrieben, die Anwendung zur Filmplanung detaillierter im Kapitel 6.

Empirische Untersuchung. Für die Entwicklung der Intuitiven Schnittstelle wurde die prototypische Anwendung zur Filmplanung einer empirischen Kontrolle unterzogen. Bei dem Prototyp handelt es sich um ein Vorbereitungswerkzeug, es geht also nicht um die Ersetzung der Filmproduktion mit ihrer komplexen Prozeßhaftigkeit, sondern um die Planung von Vorgängen vor der Produktion. Film ist ein hochkomplexer Prozeß, der teilweise 350 bis 500 Personen an einem Drehtag benötigt und bei dem fünf Arbeitstage leicht Kosten von einigen Millionen entsprechen, so daß eine genaue Planung schon im Sinne der Ökonomie notwendig erscheint. Von den zur Bewertung üblicherweise herangezogenen Techniken wurde eine Kombination aus einer Interviewform als Art der Befragung und einem Experiment ausgewählt und ein detaillierter Forschungsplan erstellt. Die gesamte Untersuchung wurde durch Videoaufzeichnung protokolliert, durch die eine Beobachtung des Benutzers bei der Bedienung des Systems möglich wurde. Die Auswertung dieser Protokolle zeigt, daß, zumindest für die befragten Personen, eine Verwendung von Raum zur Schaffung einer sinnlichen, körperlichen Kommunikation nicht zwingend notwendig ist, obgleich eine Projektion für ein Raumgefühl hilfreich ist. Alle Befragten konnten sich aber ein derartiges System besonders bei der Planung der Ausstattung und bei der Planung von Szenenauflösungen mit Ausstatter, Kameramann und Beleuchter als sehr nützlich vorstellen. Besonders bei komplizierten Szenen erscheint dies als sehr hilfreich. Szenen also, die eher bei großen Spielfilmen oder Actionfilmen die Regel sind, als beispielsweise bei einem Dokumentarfilm oder einem klassischen Autorenfilm. Somit kann der praktische Nutzen der Intuitiven Schnittstelle auf ein engeres Berufsfeld innerhalb der Filmplanung konkretisiert werden. Eine weiterführende Untersuchung mit entsprechend gezielt ausgewählten Personen könnte nähere Aufschlüsse

geben. Kapitel 6 beschreibt sowohl die Funktionalität des Prototyps als auch den Forschungsplan und die Auswertung der Untersuchung.

Publikationen und Ausstellungen. Die Ergebnisse dieser Arbeit wurden in folgenden Beiträgen publiziert:

- Hoch M (1998), *Interaktion, Körper und Realraum*, Informatik '98, Informatik zwischen Bild und Sprache, 28. Jahrestagung der Gesellschaft für Informatik, Springer 1998.
- Kulesa T, Hoch M (1998-2), *Efficient Color Segmentation under Varying Illumination Conditions*, Tenth IEEE Image and Multidimensional Digital Signal Processing (IMDSP) Workshop, July 12-16, 1998, Alpbach, Austria.
- Hoch M (1998), *A Prototype System for Intuitive Film Planning*, Third IEEE International Conference on Automatic Face and Gesture Recognition (FG'98), April 14-16, Nara, Japan 1998, pp 504-509.
- Hoch M (1997), *Object Oriented Design of the Intuitive Interface*, Proc. of 3D Image Analysis and Synthesis, Erlangen November 17-19, Infix 1997, pp 161-167.
- Hoch M (1997), *Human Body Tracking for the Intuitive Interface*, Proc. of 3. Workshop Farb-bildverarbeitung, IRB-Verlag Fraunhofer Gesellschaft, Stuttgart 1997, pp. 73-79.
- Hoch M (1997), *Intuitive Schnittstelle*, Lab, Jahrbuch 1996/97 für Künste und Apparate, Verlag Walther König, Köln 1997.
- Hoch M, Fleischmann G (1996), *Social Environment: Towards an Intuitive User Interface*, Proc. of 3D Image Analysis and Synthesis, Infix 18.-19. November 1996, pp 155-161

Das Softwaresystem der Intuitiven Schnittstelle wurde bei folgenden interaktiven Installationen verwendet und ausgestellt:

- *Traces/Pattern*, interaktive Installation von Ursula Damm, gezeigt auf der ArtCologne 1997 und Medienforum NRW 1998 in Köln.
- *Machtspiel*, interaktive Installation von Aurelia Mihai, ausgestellt auf ConverArt, Die Kunst der Abrüstung, Münster 1998.
- *VbodyRbody*, interaktive Installation von Akke Wagenaar, Masahiro Miwa, Michael Hoch und Matthias Melcher, ausgestellt in Köln 1996.

Zusammenfassend wurden in dieser Arbeit folgende neue Beiträge geleistet: Das Konzept der Intuitiven Schnittstelle basiert nicht auf einer fachlichen Abgrenzung, es setzt den Körper des Menschen in den Vordergrund und schafft dadurch Verbindungen und Querbezüge zu anderen Forschungsbereichen. Zur Realisierung wurden Möglichkeiten zur Interaktion im Raum untersucht und Alternativen zu einer traditionellen Menüauswahl in dieser Situation gefunden, bei der der Einsatz von sensitiven Flächen am Projektionsrand nicht den gewünschten Erfolg lieferte. Dadurch konnten die oben erwähnten Unterschiede zu traditionellen Systemen definiert und Methoden zur Rauminteraktion als Lösung gefunden werden. Für die Personenverfolgung konnte durch eine Untersuchung der Auswirkung von Beleuchtungsschwankungen ein Kalibrierungsverfahren angegeben werden. Weiterhin liefert die Kombination eines modellbasierten Verfahrens mit Regeln und Aktionen ein stabiles Verfolgungsverfahren und ein natürlicher aussehendes Modell als es aus den Daten der Bildverarbeitung allein möglich wäre. Die Architektur des Softwaresystems stellt ein offenes System dar, das sich auch bei interaktiven Installationen als nützlich erwiesen hat.

2 Szenario

In diesem Kapitel wird das Szenario der Intuitiven Schnittstelle beschrieben. Dazu werden zuerst einige Aspekte der Phänomenologie der Wahrnehmung in Bezug auf Körper und Raum erläutert. Anschließend werden in einer Bestandsaufnahme bestehende Ansätze beschrieben und versucht, sie bezüglich der für die Intuitive Schnittstelle relevanten Punkte zu vergleichen und einzuordnen. Nach einer Beschreibung des Konzepts werden schließlich einige Anwendungen der Intuitiven Schnittstelle bzw. des im Rahmen dieser Arbeit entwickelten Softwaresystems erläutert.

2.1 Körper und Raum

Über weite Strecken des 17. und des 18. Jahrhunderts war der Gedanke vom menschlichen Körper als Apparat, als ein mechanisches System von Mechanismen, weit verbreitet. Dies wird beispielsweise an dem Aufsatz von La Mettrie (1709-1751): *Der Mensch als Maschine* und den technischen Analogien der menschlichen Organe (das Herz als Blasebalg) deutlich. Ein Gedanke, der interessanterweise in der *Künstlichen Intelligenz* in abgewandelter Weise (der Mensch als Programm) wiederkehrt. Die Romantik des späten 18. Jahrhunderts hat sich zwar energisch gegen den Gedanken des Menschen als Mechanismus gewehrt, trotzdem war Ende des 19. Jahrhunderts der Körper für viele Denker ein Stück Materie, ein Bündel von Mechanismen. Ein Gedanke, der sich auf der wissenschaftlichen Ebene jetzt auflöste, gleichzeitig jedoch auf der populären Ebene einen enormen Schub bekam. Ende des 19. Jahrhunderts setzte mit Sigmund Freud die Psychoanalyse ein, in deren Folge das Psychische wieder eine wesentlich größere Bedeutung bekam. Das 20. Jahrhundert hat den Begriff des Leibes, das heißt des belebten und beseelten Körpers, wiederentdeckt und vertieft. Der Leib und besonders seine Fähigkeit zur Bewegung werden hierbei zur Bedingung der Wahrnehmung von Gegenständen und der Erfahrung von Welt. Im folgenden werden einige Punkte der Phänomenologie aufgegriffen und in Bezug zu einer Arbeitssituation am Computer gesetzt. Ich beziehe mich dabei im Wesentlichen auf eine der herausragenden Arbeiten dieses Jahrhunderts zur Wahrnehmung und Wechselbeziehung zwischen Körper und Raum (Merleau-Ponty 66).

2.1.1 Wahrnehmungserfahrung

Zur Wahrnehmung von Gegenständen in der natürlichen Umgebung besagt die Gegenstand-Horizont-Struktur, daß beim Betrachten eines Gegenstandes nicht nur dieser selbst, sondern gleichsam seine Beziehung zu allen anderen Gegenständen - dem sogenannten Horizont - erfaßt wird. Wenn wir anschließend einen Gegenstand erblicken, den wir eben zuvor mit dem Blick flüchtig streiften, so werden diese anderen Gegenstände zum Horizont. Wir vergleichen hierbei nicht die jetzt gesehenen Einzelheiten mit dem vorherigen Überblick, sondern vollziehen eine wirkliche Identifikation des Gegenstandes in seiner räumlichen Situation. Eine Identifikation, die bei der Betrachtung von Bildern nicht vollzogen wird: Erfaßt beispielsweise im Film eine Kamera einen Gegenstand und nähert sich ihm, um ihn in Großaufnahme zu zeigen, so wissen wir, daß es sich um den zuvor gesehenen handelt, wirkliche Identifikation vollziehen wir nicht (Merleau-Ponty 66, S. 92). Der Horizont bildet eine wichtige Basis für die Wahrnehmung überhaupt: Ohne einen Horizont bzw. eine Peripherie zu der sich Gegenstände in Beziehung stellen lassen, ist die gegenständliche Wahrnehmung nicht möglich. Das bedeutet beispielsweise: „*Nehme ich diesen Tisch hier war, so kann meine Wahrnehmung der Platte die der Beine nicht gänzlich ignorieren, da sonst der Gegenstand auseinanderfiele*“ (Merleau-Ponty 66, S. 464).

Die Peripherie ist es auch, der bei fast allen Computeranwendungen zu wenig Beachtung geschenkt wird (Ishii 97). Aufgrund der traditionellen Schnittstelle mit Maus, Tastatur und Monitor wird stets die „alte“ Information durch neue ersetzt, sie bildet nicht den Horizont als Basis einer Wahrnehmung.

Das zweite Element ist der wahrnehmende Körper selbst, der mit der sichtbaren Welt verschränkt ist, indem er einen Standort hat, von dem aus er sieht. Ein Standort von dem aus der Körper, um einen Gegenstand herumgehend, nicht nur eine Reihe perspektivischer Ansichten über ein Ding gewinnt, sondern es dabei in einen Raum setzt. Fortan kann man einerseits sagen, daß der Raum selbst sich buchstäblich durch den Körper hindurch kennt, andererseits bilden die Dinge in seinem Umkreis, da der Mensch sieht und sich bewegt, einen Teil seiner eigenen Definition (Merleau-Ponty 84, S. 52, S. 16). Der Körper ist das sich bewegende Lebewesen, durch das die wahrgenommenen Dinge, im wahrsten Sinne des Wortes, leibhaftig erfaßt werden. Die tätige Bewegung spielt dabei eine wesentliche Rolle bei der Etablierung eines für die Wahrnehmung wichtigen Raumniveaus: Bei einem Experiment setzte man einem Individuum eine Brille auf, die die Welt um 180 Grad verdreht erscheinen ließ. Der Proband hatte jetzt von seinem Körper zwei unvereinbare Vorstellungen: eine, die ihm durch Tastempfindungen von solchen Bildern vor dem Experiment und eine, die den Leib mit den Füßen nach oben zeigt. Nach einiger Zeit jedoch werden die alten Assoziationen zwischen „oben“ und „Bein fühlen“ zugunsten der neuen beseitigt, das Subjekt sieht nicht nur die Beine, sondern fühlt sie auch dort. Hierbei wurde festgestellt, daß die Verdrängung sich um so leichter vollzieht, je aktiver der Proband ist (Merleau-Ponty 66, S. 285-287).

2.1.2 Körperraum und Außenraum

Dem Körper kommt in der Gegenstand-Horizont-Struktur als Mittelpunkt der Wahrnehmung offensichtlich eine besondere Rolle zu. Die Räumlichkeit des Körpers ist im Gegensatz zu der äußerer Gegenstände keine Positionsräumlichkeit, sondern eher eine Situationsräumlichkeit. Wenn ich mich auf einem Schreibtisch abstütze, so weiß ich nicht nur, wo meine Hand sich befindet, ich weiß auch genau, wo sich der Körper und jeder seiner Bestandteile befindet, und zwar nicht aufgrund eines Verhältnisses zu anderen Positionen oder äußeren Koordinaten, sondern aufgrund der Situation des Körpers selbst. Diese Situationsräumlichkeit ist mir gegeben, ohne mir als objektive Erkenntnis bewußt zu sein, das heißt ich kann mich in einer Umgebung einrichten, ohne Leib und Umgebung erst als Objekte im Kantischen Sinne aufnehmen zu müssen. So fällt es wesentlich leichter, beispielsweise die Nase zu greifen als auf sie zu zeigen. Ein Experiment mit einem Kranken zeigt, daß die Fähigkeit des Zeigens gänzlich verloren gehen, obwohl die des Greifens mit Erfolg ausgeführt werden kann: *„Was den Körperraum betrifft, so gibt es offenbar ein Wissen des Ortes, ... Von einer Mücke gestochen, muß der Kranke die gestochene Stelle nicht erst suchen, er findet sie unverzüglich, da es ihm nicht darum geht, sie in Bezug zum Koordinatensystem des objektiven Raums zu setzen, sondern nur darum, seine phänomenale Hand an eine schmerzende Stelle seines phänomenalen Leibes zu führen, ...“* (Merleau-Ponty 66, S. 128-131).

Etwas anders verhält es sich mit sogenannten abstrakten und virtuellen Bewegungen, d.h. z.B. Bewegungen, die ich ausführe, um die Form eines Gegenstandes in der Luft zu symbolisieren (abstrakt) oder Bewegungen, die einen Gegenstand erfahrbar machen allein aufgrund taktiler Reize (virtuelle). Abstrakte Bewegungen werden erst durch eine sogenannte Projektionsfunktion ermöglicht, die Zeichen erscheinen läßt und das Handeln orientiert (wie Schilder in einem Museum den Besucher leiten). Die menschliche Leistung besteht hier gerade darin, das natürliche Verhältnis von Körper und Umgebung umzukehren, sich von der Welt abwenden und seinen Körper besitzen zu können. Der Körper ist

also nicht nur durch wirkliche Situationen in Bewegung zu setzen, er kann sich auch auf Experimente einlassen und sich im Virtuellen situieren. Genau diese Fähigkeit ist bei dem beschriebenen Kranken verloren gegangen, nicht jedoch das natürliche Verhältnis zu seinem Körperraum (Merleau-Ponty 66, S. 137-138). In Bezug auf die Betrachtung von Bildern und Szenen auf einem Computerbildschirm bedeutet dies, daß der Mensch sich sehr wohl im Virtuellen situieren kann, dabei jedoch die Erfahrung der „Welt“ nicht aus einem natürlichen Körperschema heraus bezieht, sondern diese erst mittels seines Intellekts und der beschriebenen Projektionsfunktion möglich wird. Hierbei haben wir es dann mit einem zivilisatorisch/kulturell transformierten Leib zu tun, also dem, was man gemeinhin Körper nennt.

2.1.3 Der Leib und sein Spiegelbild

In der normalen Wahrnehmung haften bestimmte Eigenschaften, wie die Tiefe, Oben und Unten, Rechts und Links, nicht in erster Linie den Dingen an, sondern entstehen stets bezüglich eines Raumniveaus. *„Sagen wir von einem Gegenstand, er sei riesig oder winzig, fern oder nahe, so doch zumeist ohne irgendeinen auch nur impliziten Vergleich mit einem anderen Gegenstand oder auch nur mit der Größe oder objektiven Stellung unseres eigenen Leibes, sondern mit Bezug nur auf eine bestimmte „Reichweite“ unserer Gesten, auf einen gewissen „Anhalt“ des phänomenalen Leibes an einer Umgebung.“* (Merleau-Ponty 66, S. 310). Ein für die Wahrnehmung wichtiges Raumniveau entsteht also erst aus der Situation des Körpers zu seiner Umgebung. Die zentrale Bedeutung, die der Körper als Nullpunkt für jegliche Größenbeziehung besitzt, wird bei der Betrachtung seines Spiegelbildes deutlich. Ein Kartesianer sieht das Spiegelbild als eine Gliederpuppe, als ein Außen an, das für ihn selbst ebensowenig ein Leib ist wie für die anderen. Das Spiegelbild ist ihm ähnlich hat aber zugleich nichts von ihm, kann insofern auch kein Raumniveau konstituieren.

Wie verhält sich das Raumniveau bzgl. Abbildungen der Realität, wie sie in der Malerei, bei einem Kupferstich oder auch bei computergenerierten Szenen unserer subjektiven Wahrnehmung erscheinen? Die Malerei zeigt uns, was das alltägliche Sehen für unsichtbar hält. Durch sie brauchen wir keinen „Muskelsinn“, um den umfänglichen Reichtum der Welt zu erfassen. Doch so lebhaft ein Bild uns Wälder, Städte oder Menschen vergegenwärtigt, so ähnelt es ihnen - ähnlich dem Spiegelbild des Leibes - doch nicht, ist es *„nur etwas Farbstoff, der hier und da aufs Papier gebracht wurde“* (Merleau-Ponty 84, S. 24). Aus dieser Überlegung heraus können sich die Dinge eines computergenerierten Bildes, sichtbar durch den „Farbstoff“ auf dem Monitor, nur bezüglich eines Raumniveaus des Arbeitsplatzes konstituieren. Dies liegt an der Verankerung des Subjekts in der natürlichen Umgebung, der Umgebung, in der der Bildschirm situiert ist. Ein bei Merleau-Ponty (66, S. 290-293) beschriebener Versuch mit einem in einem Spiegel schräg dargestellten Zimmer zeigt jedoch, daß dies sehr wohl möglich ist, falls dem Subjekt die Verankerung nicht mehr gegenwärtig ist:

„Wenn das Subjekt aber seine vorherige Verankerung nicht dadurch befestigt, daß es den Blick vom Spiegel entfernt, geschieht nach einigen Minuten das Wunder, daß das reflektierte Zimmer ein Subjekt hervorruft, das in ihm zu leben vermag. Dieser virtuelle Leib verdrängt den wirklichen Leib so weitgehend, daß das Subjekt sich nicht mehr in der Welt fühlt, in der es sich tatsächlich befindet und statt seiner wirklichen Arme und Beine solche Arme und Beine empfindet, die es haben mußte, um in dem reflektierten Zimmer gehen und tätig sein zu können, es bewohnt das Schauspiel. In diesem Moment gerät das Raumniveau ins Schwanken und etabliert sich in der neuen Lage.“

Dieser Effekt wird bei der computergestützten Generierung virtueller Welten als Immersion bezeichnet. Ein Effekt der im Apparat Kino sehr wohl funktioniert, bei den meisten computergestützten Anwendungen jedoch versagt, da nicht, wie beim Kino, eine narrative mit Identifikationsangeboten besetzte Umgebung vorliegt, in die der Mensch sich freiwillig begibt. Statt dessen wird eine zweite Welt geschaffen, die ein anderes als das konstituierte Raumniveau des Betrachters erfordern würde.

2.1.4 Der Mensch als wahrnehmendes Subjekt

Der Mensch ist also ein Subjekt, das durch seinen Leib, bzw. seinen Körper, ein Raumniveau herstellt, bezüglich dessen Welt erfahren wird. Alle Aktionen, alle tätigen Leibbewegungen, ja jegliche Wahrnehmung geschieht stets aus dieser Situation heraus:

„Doch welchen Sinn würde das Wort „auf“ für ein Subjekt haben, das nicht durch seinen Leib einer Welt gegenüber situiert wäre? ... Wenn ich von einem Gegenstand sage, er liege auf dem Tisch, so versetze ich mich immer in Gedanken in diesen Gegenstand und in den Tisch und wende auf beide einen Begriff an, der ursprünglich beheimatet ist in den Verhältnissen meines Leibes zu äußeren Gegenständen. ... Vertikale und Horizontale, Nähe und Ferne sind nur abstrakte Bezeichnungen für ein einziges Sein-in-Situation und gründen sich auf ein und dasselbe „Vis-à-vis“ von Subjekt und Welt.“ (Merleau-Ponty 66, S. 126, 311)

Die Verbindung des Subjekts mit der Idee des Leibes und der Idee der Welt ist hierbei kein gedachtes Verhältnis, denn dann könnte das Subjekt absolut unabhängig als Denker bestehen und wäre nicht situiert. Diese Überlegungen lassen den Schluß zu, daß die traditionelle Arbeit mit dem Computer, mit Tastatur, Maus und Bildschirm, das Subjekt auf eine Weise seiner natürlichen Situation enthebt, auf eine andere Weise dieser Versuch aber scheitern wird, aufgrund seiner Verankerung in der Umgebung. Lediglich auf einer abstrakten Ebene, wie es beim Lesen eines Buches geschieht oder nur, wenn die Verankerungspunkte „unsichtbar“ gemacht werden, wie es im Kino in der Regel der Fall ist, wird hier Welt erfahren. Bestimmte Computeranwendungen versuchen durch Realismus und Detailreichtum Welt so zu vermitteln, wie wir sie naturgegeben gewohnt sind, sie können jedoch erst erfolgreich sein, wenn sie die Situation, das Situiertsein des Benutzers in seiner Umgebung, berücksichtigen. Dazu muß sich eine Computeranwendung in die natürliche Umgebung integrieren, Raum, Bewegung, und Realobjekte einbeziehen, um uns mehr als nur die Vorstellung von einem Ding zu geben, kurz: der Mensch muß gegenüber der computergenerierten Welt situiert sein. Eine situierte Umgebung bietet bei der Anwesenheit Dritter einen weiteren Vorteil: *„Mein Blick fällt auf einen lebendigen, in Tätigkeit begriffenen Leib, und sogleich empfangen die ihn umgebenden Objekte eine neue Bedeutungsschicht: sie sind nicht mehr nur, was ich selbst mit ihnen tun könnte, sie sind auch das, was jenes Verhalten mit ihnen zu tun im Begriff ist.“ (Merleau-Ponty 66, S. 405).* Meine Welt ist also nicht mehr nur mir gegenwärtig, sondern auch für die andere Person gegenwärtig.

2.2 Forschungsstand

Bei einer Betrachtung der Vielzahl von Computeranwendungen läßt sich feststellen, daß der Mensch in zwei Welten arbeitet, die relativ wenig gemeinsam haben, der Computerwelt und der natürlichen Umgebung. Der Wunsch liegt also nahe, nach Ansätzen zu suchen, die nicht wie bisher den Menschen auffordern in der Welt der digitalen Maschine zu arbeiten, sondern den Computer so zu gestalten, daß er in unserer Welt arbeiten kann. Das Ergebnis wäre nicht nur einfach ein vergrößerter Interaktionsraum, sondern auch eine ergonomischere und natürlichere Arbeitsweise. Aus diesem Wunsch haben sich eine Vielzahl von interaktiven Systemen entwickelt. Systeme, die trotz verschiedener Zielrichtungen und diverser Benennungen stets das gleiche Ziel verfolgen: Eine intuitive, natürliche Interaktion mit dem Computer. Je nach Annäherung an dieses Ziel und verwendeter Technik lassen sich folgende Ansätze unterscheiden:

Virtual Reality befreit den Benutzer vom Sitzen vor dem Bildschirm und erlaubt die Interaktion des gesamten Körpers, sowie eine Vielzahl von (virtuellen) Interaktionsmöglichkeiten mit computergenerierten Welten. Hilfsmittel sind Head-Mounted-Displays, große Projektionen, Stereoprojektionen, Datenhandschuhe oder Datenanzüge, mit denen der Versuch unternommen wird, die physikalische Welt durch eine computergenerierte zu ersetzen. Der Hauptnachteil liegt darin begründet, daß keine wirklich glaubwürdige Simulation mit einem vertretbaren Kostenaufwand erreichbar ist, bzw. bis heute nicht erreicht worden ist (Azuma 97). Der Benutzer wird durch die computergenerierte Welt nicht getäuscht, und der Computer verschwindet nicht aus dem Hauptaugenmerk des Benutzers, da meist sehr aufwendige Apparaturen für den Gebrauch notwendig sind (wie etwa beim CAVE).

Augmented Reality. Bei diesem Ansatz wird genau das Gegenteil des VR-Ansatzes verfolgt: Statt den Benutzer mit Hilfe des Computers „einzuhüllen“, wird hier versucht, mit Hilfe von Computern die reale Welt zu erweitern. Als Hilfsmittel werden hier verschiedene Sensoren verwendet, damit der Computer auf die Bedürfnisse des Benutzers reagieren kann. Halbdurchlässige Head-Mounted-Displays und Projektionen erlauben elektronische Informationen darzustellen, ohne daß der Benutzer die Verbindung zu seiner natürlichen Umgebung verliert. Ein spezielles Feld der Augmented Reality integriert dreidimensionale virtuelle Objekte in die ebenfalls dreidimensionale, reale Umgebung. Dabei treten sogenannte Registrierungsprobleme auf, die bis heute nicht umfassend gelöst worden sind. Daher werden in diesem Bereich bisher auch keine kommerziellen Produkte angeboten (Azuma 97).

Responsive Environments. Hierunter sind alle Anwendungen zusammengefaßt, deren Hauptfokus auf der Reaktion des Computers auf die Benutzerbewegungen im Raum liegt. Sie lösen das Problem der Zuordnung von mehr oder weniger freien Bewegungen des Benutzers in seiner natürlichen Umgebung zu Funktionen der aktuellen Anwendung. Für diese Art der Vorgehensweise gibt es eine Vielzahl von Projektbegriffen wie *Perceptive* oder *Interactive Spaces*, *Smart Rooms*, *Immersive Environment*, *Reactive Environment*, *Intelligent Room* etc. (Wren 97, Druin 94, Cooperstock 97, Coen 98, Torrance 95). Je nach Umgebung und verwendeter Technik lassen sie sich oft auch in eine der anderen hier aufgeführten Vorgehensweisen einordnen.

Ubiquitous Computing (ubiquitous = allgegenwärtig) verfolgt ebenfalls das Gegenteil des VR-Ansatzes, indem der Computer „gezwungen“ wird, in der Welt des Benutzers zu leben (Weiser 93). Besonder Aspekt ist hier, daß die Rechenleistung sich nicht in einem

speziellen (persönlichen) Gerät manifestiert, sondern verteilt in den Dingen der natürlichen Welt, quasi überall situiert ist. Ziel ist es den Computer so anzupassen, so stark in die natürliche Welt zu integrieren, daß er benutzt wird, ohne daß überhaupt darüber nachgedacht wird. Dabei wird jede Person fortwährend mit hunderten von nahezu kabellos miteinander verbundenen Computern kommunizieren. Ein Gedanke der im Internet schon Realität geworden ist, jedoch ohne die räumliche Situation des sogenannten *Ubicomp*-Prinzips, bei dem Computer in Wänden, Stühlen, Kleidung, Lichtschaltern, Autos, ja eben in jedem Gerät vorgeschlagen werden. Zu beachten ist, daß *Ubicomp* mehr ein Prinzip ist, von dem Teilaspekte von mehreren Anwendungen verwendet werden; es läßt sich also sowohl in dem Ansatz der *Responsive Environments* als auch der *Augmented Reality* wiederfinden.

Calm Technology. Die sogenannte *stille Technologie* ist eine gedankliche Weiterentwicklung des *Ubicomp*-Prinzips, in dem eine verteilte Rechnerstruktur zu Grunde gelegt wird. Besonderes Augenmerk wird hier auf die Einbeziehung der Peripherie gelegt, d.h. die Unterscheidung des Zentrums, als das Zentrum der Aufmerksamkeit, von der Peripherie, die in der Umgebung allgegenwärtig ist und bei Bedarf, z.B. beim Eintreffen neuer Informationen, in das Zentrum der Aufmerksamkeit rücken kann. Auch hier wird argumentiert, daß, wo immer es möglich ist, digitale Information mit physikalischen Objekten verbunden werden soll (Weiser 96). Das Prinzip der *stillen Technologie* verbindet die Ansätze der *Reactive Environments*, *Ubicomp* und *Tangible Bits*.

Tangible Bits. Die *fühlbaren Bits*, ein Forschungsprojekt des MIT (Ishii 97), erweitern die physikalische Welt durch die Koppelung von alltäglichen Dingen und Umgebungen mit digitaler Information; sie sollen die Welt selbst in ein Interface transformieren. Hauptziel ist es, den sensitiven Reichtum, den physikalische Dinge mit sich bringen, mit ihren vielfältigen Interaktions- und haptischen Erfahrungsmöglichkeiten in die Welt der Mensch-Maschine-Schnittstellen zu bringen. Auch hier gibt es Ähnlichkeiten und Übereinstimmungen zu anderen Bereichen, wenngleich hier der Fokus stärker auf eine greifbare reale Welt gelegt wird, als auf eine rein visuelle Erweiterung der Realität der *Augmented Reality* oder auf die Zuordnung von Bewegungen zu Funktionen der aktuellen Anwendung, wie sie bei den *Reactive Environments* verfolgt wird.

Es läßt sich also feststellen, daß gerade in den letzten Jahren eine Vielzahl von Begriffen geprägt worden sind, die letztendlich aus einem gemeinsamen Antrieb nach natürlicher und intuitiver Kommunikation mit dem Computer herrühren. Die Begriffsvielfalt hilft hierbei leider nicht, die einzelnen Bereiche zu verbinden, da teilweise identische Ansätze unterschiedlich benannt werden. So werden beispielsweise die *Natural User Interfaces* in Anlehnung an grafische Benutzerschnittstellen (GUIs) mit NUIs abgekürzt (Rauterberg 97), die zum Teil identischen *Tangible Bits* als TUIs bezeichnet. Es bleibt abzuwarten, inwieweit sich eine bestimmte Technologie oder Denkweise in Zukunft durchsetzen wird. Im folgenden werden die verschiedenen Ansätze mit Beispielen näher erläutert. Aufgrund der angesprochenen Ähnlichkeiten wird hier eine hierarchische Einteilung nach inhaltlichen Ansätzen vorgenommen, die sich aus dem grundlegenden Wunsch nach intuitiver Interaktion über reaktive Umgebungen bis hin zu Interaktion mit Realobjekten ableitet.

2.2.1 Intuitive Interaktion

2.2.1.1 Einfache Schnittstellen und multimodale Interaktion

Ausgangspunkt für eine intuitive Interaktion mit dem Computer ist der Wunsch nach einer einfachen und multimodalen Kommunikation, die aus der Mensch-Mensch-Kommunikation abgeleitet wird. Demgegenüber steht die Beobachtung, daß Technologie,

besonders im Bereich der Computer, die Aktivitäten des Menschen mit Bedienungsanleitungen und zum Teil verwirrenden Benutzerschnittstellen verkompliziert hat. Werden die Regeln der Interaktion mit dem Computer betrachtet, befinden wir uns noch immer im Zeitalter von mit oder durch Lochkarten getriebenen Benutzerschnittstellen, d.h. der Benutzer muß sich in einen expliziten maschinenorientierten Dialog begeben. Einen Dialog, der eine Einschränkung der Bewegungsfreiheit bedingt und die Bedienung von Eingabegeräten erfordert.

Mit der Verfügbarkeit von preiswerten Framegrabber-Karten zu Beginn der 90er Jahre entstanden erste Ansätze für kabellose Techniken zur Animation von computergenerierten Szenen, wie etwa *virtuelle Holografie*, d.h. die Änderung des Blickwinkels der Szene je nach Betrachterstandpunkt (Azarbayejani 93). Damit kann sich der Benutzer zusätzlich zur herkömmlichen Interaktion aktiv bewegen und damit multimodal interagieren. Eine multimodale Interaktion bildet eine wichtige Basis für eine einfache, natürliche Schnittstelle, deren Notwendigkeit aus der Beobachtung der alltäglichen Mensch-Mensch-Interaktion abgeleitet und auf den Computer übertragen wurde. Gestützt wird diese Vorgehensweise durch Untersuchungen wie die von Hauptmann (89), die zu dem Ergebnis kommt, daß Menschen multimodale Dialoge vor anderen vorziehen und diese auch intuitiv verwenden. Konkrete multimodale Systeme sind schon in Bolt (84) zu finden, die jedoch aufgrund der fehlenden Technologie (Hard- wie Software) nicht aus einem prototypischen Stadium herauskamen. Spätere Systeme verwenden Sensoren und Eye-Tracking, um Objekte auf einem Display zu erzeugen, zu bewegen oder zu löschen (Thorisson 92). Für die Schwierigkeiten, die verschiedenen Eingabekanäle der multimodalen Interaktion zusammenzuführen, wird hier eine agentenbasierte Struktur vorgeschlagen, um beispielsweise Kommandos wie „lösche das Objekt südlich des Hauses“ zu ermöglichen. Natürlichsprachliche Interaktion wird auch in Jung (98) für die interaktive Montage von 3D-computergrafisch dargestellten Bauteilen vorgeschlagen. Für eine erste Entwicklung ist diese Beispielanwendung eingeschränkt auf eine Spielwelt (Baufix), jedoch sind Anwendungen für eine Konstruktion von „wirklichen“ Maschinen vorstellbar. Hier, wie auch in der intelligenten Umgebung VIENA (Wachsmuth 95), ist die Entwicklung einer intelligenten Kommunikation mit einer virtuellen Umgebung im Bereich der interaktiven Konstruktion bzw. des Grafikdesigns das Ziel. Die Grundüberlegung ist, daß die Anweisungen eines Designers um so besser interpretiert und ausgeführt werden können, je kompetenter das System ist. Diese Kompetenz soll durch ein agentenbasiertes, verteiltes System erreicht werden.

Einen Schritt weiter gehen die Ansätze zur sprachbegleitenden Körpergestik in multimodalen Umgebungen, bei denen eine gestische Kommunikation durch simple Spracheingaben unterstützt wird und dadurch die Interaktion einfacher gestaltet werden soll (Latoschik 97, Fröhlich 97). Hierbei entstehen bestimmte Anforderungen an die Schnittstelle, da in den dargestellten virtuellen Welten eine traditionelle WIMP-Schnittstelle¹ in Umgebungen mit Projektion, wie etwa der des CAVE oder der *Responsive Workbench* (Krüger 94), unzureichend ist (Latoschik 97). Neben der Eingabe von Anweisungen, muß also der Dialog selbst einfacher gestaltet werden. Eine Beobachtung, die in Fukumoto (94) zu der Bezeichnung *Common Interface* im Gegensatz zu einem *Professional Interface* führt. In Pentland (95) wird argumentiert, daß die Erkennung verschiedener Modalitäten wie Sprache, Gestik oder auch Gesichtserkennung lediglich den ersten Schritt für erfolgreiche Interfaces darstellen. Wichtige Voraussetzung ist das Erkennen und Verstehen der menschlichen Handlungen und Absichten. Dabei stehen wir vor der Schwierigkeit, daß wir das menschliche Wesen nicht gut genug verstehen, um ein akkurates Verhaltensmodell im Computer erstellen zu können. Die in Pentland (95) beschriebenen Anwendungen

¹ WIMP steht für Window, Icon, Menu und Pointing device

beziehen die Körperbewegungen des Benutzers im Raum in die Schnittstelle ein. Ein wirkliches Verstehen von Handlungen liegt hier jedoch gerade nicht vor, der Fokus liegt eher auf der Entwicklung einer kabellosen Interfacetechnologie, die an einigen Beispielen demonstriert wird. Darüberhinausgehende Überlegungen werden auch in Pinhanez (97) angestellt, bei denen die Aktionen des Menschen in den Vordergrund gerückt werden. Systeme sollten menschliche Aktionen verstehen und berücksichtigen können, nicht nur einfache Abbildungen zwischen Sensor und Reaktion realisieren. Bei allen Überlegungen zu kabellosen Interfacetechniken sollte trotzdem stets berücksichtigt werden, daß eine Interaktion ohne Geräte nicht immer sinnvoll ist, z.B. ist die Verwendung eines Joystick für die Steuerung eines Videorecorder sicher besser geeignet, als gestische Anweisungen mit dem ausgestreckten Arm wie es in Fukumoto (94) vorgeschlagen wird.

2.2.1.2 Zeige- und Gestiksysteme

Zur Einbeziehung des Menschen in einen multimodalen Dialog sind eine Vielzahl von Systemen zur Erkennung von Zeigegestiken bis hin zu komplexerer Gestik entwickelt worden. Aufgrund der Einfachheit der sensorischen Koppelung wurden oft Datenhandschuhe oder elektromagnetische Sensoren verwendet. In Bolt (84) wird zur Erkennung einer Zeigegestik lediglich ein Sensor verwendet. Mit der Entwicklung des Datenhandschuhs (Zimmermann 87) entstanden erste Anwendungen zur Manipulation von computergenerierten Objekten. Motiviert wurden diese Anwendungen aus der Beobachtung, daß Menschen normalerweise nicht bloß einen Finger für eine Interaktion benutzen (Hauptmann 89). So wurde die gesamte Hand zur Steuerung einer virtuellen Hand via Datenhandschuh in Weimer (89) verwendet, um u.a. mit Hilfe eines virtuellen Kontrollfeldes Modellierungsaufgaben im Rechner zu ermöglichen. Eine Datenhandschuh-basierte Zeigegestik vor einem Computerdisplay wird in Thorisson (92) beschrieben, eine Präsentationsanwendung mit Zeigegestiken verwirklicht das System CHARADE von Baudel (93).

Um kabellose Interaktionen mit dem Computer zu ermöglichen, sind mit Hilfe von Kameras und Verfahren der Mustererkennung eine Reihe von Anwendungen unter Verwendung von Zeigegestiken entstanden: Das System *Perseus* wird verwendet, um Objekte im Raum zu lokalisieren, auf die der Benutzer zeigt (Kahn 95/96). In Mase (97) und Russel (95) werden richtungsweisende Zeigegestiken zur Navigation in einer virtuellen Welt verwendet. M. C. Torrance beschreibt einen *intelligenten Raum* zur Unterstützung der alltäglichen Arbeit in einer Büroumgebung (Torrance 95, Coen 97/98). Der Benutzer kann auf Objekte (farbig gekennzeichnete Klebezettel) oder auf eine Projektion zeigen und dadurch ein zuvor festgelegtes Kommando ausführen. Das System besteht aus mehreren Kameras, zwei großen Projektionsflächen in einem kleinem Raum. Einfache Zeigegesten in Verbindung mit einem Sprachkommando werden verwendet, um beispielsweise die Wettervorhersage eines bestimmten Ortes, dargestellt auf einer Landkarte, zu erhalten. Für Präsentationen können mit dem System Gliederungsebenen ein- und ausgeblendet werden, oder es kann zur nächsten Folie geblättert werden. In ihrer Art der Benutzung dienen die gestischen Anweisungen hierbei lediglich als eine Art Mausersatz. Das Zeigen auf Schaltflächen, dargestellt auf einer Projektion, wird in dem System *Finger-Pointer* beschrieben (Fukumoto 94). Hier ist die Steuerung eines Cursors in Echtzeit möglich. Ein Schwerpunkt des Ansatzes konzentriert sich auf eine Synchronisation von Zeigegestik und Spracherkennung, ein Problem, daß durch unterschiedliche Verzögerungszeiten einzelner Erkennungsmodule entsteht. So ist es beispielsweise wichtig die genaue Zeigerichtung zum Zeitpunkt eines Kommandos zu kennen, obwohl das Kommando erst 0,5 Sekunden später erkannt werden kann. Das *Finger-Pointer*-System arbeitet zwar kabellos, hat aber bestimmte Anforderungen an die Beleuchtung im Raum und erlaubt nach einer notwendigen Kalibrierungsphase keine Bewegung des Benutzers im Raum.

Die Erweiterung eines Systems um die Erkennung auch komplexer Handgesten kann unter anderem durch die Untersuchungen wie die von Hauptmann (89) motiviert werden. Dort zeigte sich eine erstaunliche Einheitlichkeit der verwendeten Gesten von verschiedenen Testpersonen. Dabei handelte es sich hier um sehr eingeschränkte Manipulationsmöglichkeiten, die außerdem keine statistische Signifikanz besitzen, sondern lediglich als beschreibende Unterstützung dienen können (Hauptmann 89). Trotz diverser Ansätze hat sich diese Art der Interaktion, d.h. eine Interaktion mit eher komplexen Gesten noch nicht durchgesetzt. In Weimer (89) werden für die Modellierungsaufgaben Gestiken mit Daumen und Zeigefinger verwendet. Dazu zählen das Positionieren von Objekten wie auch die Oberflächenverfeinerung für CAD-Anwendung. In dem *Finger-Pointer*-System wird der Daumen als Selektionsschalter verwendet (Fukumoto 94). In Onishi (93) werden bestimmte Handhaltungen zur Manipulation von virtuellen Objekten auf einer Projektion verwendet. Ansätze mit komplexer Gestik in Form von verschiedener Handgesten finden sich in (Baudel 93, Bröckl-Fox 95, Maggioni 95, Stark 95). Das System ARGUS dient zur Steuerung von Geräten im Haushaltsbereich, wie etwa die Stereoanlage, durch Gesten (Kohler 96).

In der Regel werden die verschiedenen Gesten zur Steuerung von Funktionen verwendet, die traditionell über Kommandos, Menüs oder die Maus eingegeben werden. Oft fehlt in den Ansätzen eine Nutzung des Interaktionsraums selbst: Der Benutzer hat zwar einen gewissen Aktionsradius, dieser ist aber nicht Teil der Schnittstelle. Zusätzlich lassen viele Anwendungen die explizite Einbeziehung von Mensch und Umgebung außer Acht. Der Schwerpunkt der in dieser Arbeit beschriebenen Intuitiven Schnittstelle soll analog der meisten Ansätze ebenfalls auf Echtzeit gelegt werden, wobei hier eine Bewegung im Raum explizit als Teil der Schnittstelle berücksichtigt werden soll.

2.2.2 Reaktive Umgebungen

Unter reaktiven Umgebungen werden hier sowohl Präsentationssysteme, die auf die Anwesenheit und Bedienungen von einer oder mehreren Personen reagieren, als auch sogenannte intelligente oder situierte Umgebungen zusammengefaßt.

2.2.2.1 Präsentationssysteme

Neben dem oben erwähnten, auch für Präsentationen verwendeten *intelligenten Raum*, wird mit dem sogenannten *Reactive Environment* von Cooperstock (95/97) ein Prototyp eines computerunterstützten Konferenzraums beschrieben. Der Konferenzraum ist mit Technik für Videokonferenzen und elektronischen Präsentationen ausgestattet, im Einzelnen also mit Dokumentenkamera, Macintosh-Computer, Videorecorder, Projektionswand, mehreren Kamera- und Monitor- bzw. Lautsprecherkombinationen, die als elektronische Sitze für zugeschaltete Sprecher oder Gäste fungieren. Die komplexe Kontrolle dieser Vielzahl von Geräten geschieht normalerweise mittels einer graphischen Benutzeroberfläche auf dem Macintosh-Computer. Dieses Verfahren ist jedoch sehr zeitaufwendig und bedarf der eingehenden Schulung des Benutzers. Das *Reactive Environment* übernimmt nun die meisten Aufgaben mit Hilfe von Sensoren und versteckten Computern: Wenn der Benutzer den Raum betritt schalten sich Licht und Geräte automatisch ein. Der Benutzer trägt ein als *Active Badge* bezeichnetes Gerät (Hopper 93), durch das er identifiziert wird und die Verbindung mit dem Konferenzteilnehmer aufgrund eines elektronischen Buchungssystems automatisch hergestellt wird. Werden Vorlagen unter die Dokumentenkamera gelegt, wird automatisch das entsprechende Bild übermittelt, mit einer zusätzlichen Einblendung der normalen Kameraansicht in einem kleinen Fenster des Videobildes. Ähnlich verhält es sich beim Abspielen von Videokassetten. Zusätzlich kann jeder Konferenzteilnehmer durch Kopfbewegung vor seinem Monitor einen Kameraschwenk der zugehörigen Kamera initiieren und sich so im zugeschalteten Raum umsehen. Es entsteht

sozusagen ein virtuelles Fenster in den entfernten Raum. Bis auf diese Kopfbewegung wird hier jedoch die Körperbewegung des Benutzers nicht einbezogen. Eine Bewegung im Raum, um beispielsweise Verbindungen zwischen Geräten herzustellen (indem Schalter an den Geräten gedrückt werden) wird explizit als zu aufwendig angesehen. Bei der intuitiven Schnittstelle wird im Gegensatz dazu der Realraum zum Datenraum, und die Körperbewegungen werden Teil des Interfaces.

2.2.2.2 *Situierte Umgebungen*

Die Motivation für situierte Umgebungen leitet sich aus den oben erwähnten Überlegungen ab, daß der Computer, um uns nützlich zu sein, an unserer perzeptiven Umgebung Anteil haben oder eben allgegenwärtig sein muß. Die Entwicklung von situierten Umgebungen ist dabei nicht die einer grafischen Benutzeroberfläche: Im Gegensatz zu Multimedia, das stets versucht Aufmerksamkeit zu erzeugen, ist beispielsweise das Ideal des *UbiComp*-Prinzips die Unsichtbarkeit. Die bisher (nicht ganz unsichtbare) verwendete Technologie, bei Xerox Parc in den Jahren 1988-94 entwickelt, besteht aus elektronischen Tafeln (boards), Handheld-Computern (pads) und kleinen, lediglich mit Display und einem Knopf ausgestatteten Interaktionsgeräten (tabs). Einige Anwendungen werden in den *situierten Informationsräumen* von Fitzmaurice (93) beschrieben, in denen die Palmtop-Computer als ein Fenster in einen Informationsraum verwendet werden, der im physikalischen Raum situiert ist. Auf einer aktiven Landkarte können so Informationen über bestimmte Städte abgerufen werden oder in einer Bibliothek Bücher lokalisiert werden. Ein sogenanntes portables Büro gestattet es, vom heimischen Arbeitsplatz Informationen aus dem Büro abzurufen, einfach durch eine räumliche Positionierung des Palmtop-Computers bzgl. des Körperraums. Ebenso können Markierungen und Nachrichten auch räumlich plziert werden und zu einem späteren Zeitpunkt im Büro abgerufen werden. Eine weitere Anwendung von *UbiComp* findet sich in Elrod (93). Das dort beschriebene *Responsive Office Environment* verfolgt ein ähnliches Ziel wie das in Hodges (94) beschriebene *Interactive Office*: Die automatische Steuerung verschiedener Geräte, wie Licht, Heizung, Klimaanlage etc. aufgrund von detektierten Aktivitäten in Büroräumen. Einerseits soll es die alltägliche Büroarbeit erleichtern, indem beispielsweise eine automatische Rufumleitung realisiert wird oder ein Telefon bei einem Meeting lediglich einmal klingelt, andererseits gibt es auch einen ökonomischen Aspekt, wenn beispielsweise Konferenzräume nur bei Bedarf geheizt werden. Beide Ansätze versuchen den Computer in den Hintergrund zu setzen und mit der physikalischen Welt zu verbinden, so daß der Mensch sich normal verhalten kann, ohne in direkten Kontakt mit dem Computer treten zu müssen.

Die *Responsive Workbench* ist ein Beispiel für eine situierte Umgebung, bei der virtuelle Objekte und Kontrollelemente auf einer realen „Werkbank“ erscheinen (Krüger 94). Die Objekte werden als dreidimensionale Bilder auf die Oberfläche der *Workbench* projiziert. Mittels Shutter-Brillen werden die entsprechenden Bilder für das linke und rechte Auge getrennt. Als Anwendungen werden Architektursimulationen, Landschaftsplanungen und Operationsplanung angegeben. Die Arbeitssituation entspricht hierbei der natürlichen Arbeitssituation eines Architekten im Büro oder Chirurgen am Operationstisch. Die *Workbench* selbst, als physikalisches Objekt, integriert sich in den normalen Raum und verbindet so physikalische Welt mit virtuellen Objekten. Sie ist im Realraum situiert und gestattet ein Umschreiten und die Anwesenheit Dritter. Allerdings wird die korrekte Veränderung der Perspektive bezüglich der aktiven Leibbewegung lediglich für eine Person wiedergegeben, die dazu zusätzlich mit einem Sensor verkabelt sein muß². Damit wird der Benutzer auf die Arbeitssituation festgelegt, ein freies Wechseln zwischen Peripherie und Arbeitssituation ist hier jedoch nicht ohne weiteres möglich.

² Aktuelle Forschungen erlauben dieses Erlebnis für zwei Personen gleichzeitig (Agrawala 97).

2.2.3 Reale Interaktion

Die Ansätze, die eine Interaktion in oder mit der Realwelt ermöglichen, lassen sich einteilen in Konzepte einer einfachen Raumnutzung, d.h. die Unterstützung einer gewissen Beweglichkeit des Benutzers, die Overlay-Techniken der *Augmented Reality* und eine echte Interaktion mit physikalischen Realobjekten.

2.2.3.1 Raumnutzung

In Bolt (84) wird argumentiert, daß die Interaktion zwischen Benutzer und den zu lösenden Aufgaben wesentlich mehr einbezieht als Tastatur und Monitor allein, den Tisch, die Umgebung, eben den gesamten Raum. Das dort entwickelte prototypische Szenario des *MediaRooms* beschreibt ein zukünftiges Wohnzimmer mit Projektionswand, in dem der Benutzer in einem Sessel oder Stuhl sitzt und mit dem Computer durch Zeigegesten und Sprachkommandos kommuniziert. Die Technologie zum damaligen Zeitpunkt erforderte den Einsatz von am Körper getragenen Geräten um eine Gestik zu erkennen. In verschiedenen Anwendungen (*Put-that-there*, *Dataland*) wird die Verwendung der zweidimensionalen grafischen Fläche des Bildschirms für ein Datenmanagement beschrieben und als Ausblick die Verwendung von dreidimensionalen virtuellen Räumen auf Projektionswänden zur Ablage von Daten vorgeschlagen. Die Verwendung von Raum bezieht hier jedoch die Verwendung des Realraums in Verbindung mit dem Leib nicht explizit ein. Einen Schritt weiter gehen die Forschungsarbeiten von Feiner (93-2), die den Realraum für das Ablegen von Fenstern eines Standard-Fenstersystems, X-Windows, verwenden. Hierzu müssen allerdings eine halbdurchlässige Brille und zwei elektromagnetische Sender an Handgelenk und Körper getragen werden, die den Aktionsradius und die Bewegungsfreiheit schon durch das Tragen der Geräte und eine notwendige Verkabelung einschränken. Die in Krueger (91) beschriebene Umgebung *Videoplace* kommt hier ohne eine beschwerliche Verkabelung des Körpers mit Sensoren aus. In mehreren speziell gestalteten Installationen können verschiedene Arten der Interaktion erforscht werden: In einer kann der Benutzer mit seiner eigenen Silhouette interagieren, in einer anderen wird das Abbild des Benutzers durch die Hand eines zweiten beeinflußt, eine weitere zeigt einen computergenerierten Charakter, der auf spielerische Weise mit der Kontur des Benutzers auf der Projektion in Verbindung tritt.

Eine Interaktion mit Körperbewegung zur Navigation in virtuellen Welten wird in Mase (97) beschrieben. Hier werden u.a. Bewegungen des Benutzers vor einer Projektionswand nach links oder rechts verwendet um eine virtuelle Kamera zu steuern. In ihrer Art der Interaktion erinnert dieser Ansatz an die Anwendungen *SURVIVE* und *ALIVE* der interaktiven Videoumgebung *IVE*³ des MIT. Zur direkten Steuerung der spielähnlichen Anwendungen werden hier die Extremitäten der Silhouette, einfache Gesten und Bewegungen verwendet. In der Anwendung *SURVIVE* (Russel 95) steuert der Benutzer das Spiel *Doom* mit seinen Körperbewegungen statt mit der Tastatur, so geschieht die prinzipielle Navigation durch eine Bewegung nach links, rechts bzw. vorne und hinten des Benutzers. Weiterhin lassen sich beispielsweise Türen durch Anheben der Arme öffnen. In *ALIVE* tritt er in Interaktion mit einem virtuellen Hund und anderen Kreaturen, indem er durch seine Bewegungen im Raum sein Spiegelbild auf einer Projektionswand steuert (Darrell 94, Wren 97). Diese als *magic-mirror* bezeichnete Metapher situiert den Benutzer zwar in gewisser Weise gegenüber einer virtuellen Welt, eine explizite Nutzung des umgebenden Raums beschränkt sich jedoch auf eine relative Bewegung innerhalb sehr kleiner Grenzen. Andererseits besitzt diese Technik, die an die Arbeiten von Krueger (91) erinnert, den Vorteil, daß zum einen die Komplexität der Umgebung genügend eingeschränkt und damit einfache Bildverarbeitungsroutinen erfolgreich arbeiten können, zum anderen kann

³ Diese und weitere Anwendung sind inzwischen in der Projektbezeichnung *SmartRooms* zusammengefaßt worden (<http://vismod.www.media.mit.edu/vismod/demos/smartroom/>).

das Interface ohne ein spezielles Training oder Apparaturen verwendet werden (Wren 97). Der Nachteil liegt in dem Verlust des Körperraums und eines Leibgefühls des Benutzers. In Wren (97) werden die erwähnten Umgebungen unter dem Sammelbegriff *Perceptive Spaces* beschrieben und sehr wohl die Notwendigkeit zur Einbeziehung des Raums erkannt: Es wird argumentiert, daß wir in einem dreidimensionalen Raum leben und die wichtigste Interaktionserfahrung die mit anderen Personen ist. Weiterhin sind wir es gewohnt uns in Räumen zu bewegen, am Schreibtisch zu arbeiten und unsere Umgebung räumlich zu organisieren und daher sollte wir bestrebt sein, einen möglichst großen Teil der realen Umgebung reaktiv zu gestalten. Trotz dieser Überlegungen wird bei allen der dort beschriebenen Anwendungen auf die explizite Einbeziehung mehrerer Benutzer verzichtet und Realobjekte und Raumbewegung nicht als Teil der Schnittstelle eingesetzt. Die Körperbewegungen dienen meist nur einer Art Mausersatz, obwohl die Verfolgung des Benutzers durch Kameras (inzwischen) auf dem *Pfinder*-System basiert (Wren 96), welches in seinen Möglichkeiten über einfache 2D-Systeme zur Einbeziehung von Körperbewegungen, wie in *Videoplace* von Krueger (91) und dem *Mandala*-System (Mandala 93), hinausgeht.

Ebenfalls in Wren (97) wird eine Anwendung beschrieben, die die menschliche Fähigkeit des räumlichen Erinnerns ausnutzt um in einer virtuellen Welt Internet-Verweise anzuordnen und zu einem späteren Zeitpunkt wieder abzurufen. In dieser *Netspace* genannten Anwendung orientiert sich die Nutzung von Raum jedoch nicht an dem Realraum, sondern an dem in Bolt (84) vorgeschlagenen Möglichkeiten zur Verwendung vom Raum. Internet-Verweise werden in eine dreidimensionale grafische Welt abgebildet, die auf einer Projektionswand sichtbar werden. Die Argumentation, der Benutzer würde dadurch den Eindruck gewinnen, die Verweise befänden sich in dem umschließenden Raum, erscheint aufgrund der Überlegungen im Abschnitt *Körper und Raum* fraglich. Eine ganz andere Erfahrung bietet dagegen die wahrhaftige Anordnung von Informationseinheiten im dreidimensionalen Realraum, wie sie beispielsweise in der interaktiven Installation *smdk* realisiert wurde (Fleischmann 94-2). Es handelt sich um eine begehbare interaktive Datenbank bestehend aus mehr als 300 verschiedenen Klangdaten, die von einem Besucher mittels eines kabelloses Ultraschall-Sensorsystems im Realraum aktiviert werden können. Hier sind Orte im Raum tatsächlich mit virtuellen Informationen verknüpft.

Aus obigen Überlegungen findet die beschriebene *magic-mirror*-Metapher in der Intuitiven Schnittstelle keine Verwendung. Auch soll der Realraum nicht nur als relatives Referenzsystem verwendet werden, um in einem virtuellen Raum navigieren zu können, sondern Teil der Schnittstelle selbst werden.

2.2.3.2 *Augmented Reality*

Anwendungen der *Augmented Reality* umfassen medizinische, militärische und Konstruktionsanwendungen (Azuma 97), die Unterstützung der Wartung von Geräten (Feiner 93-1), sowie Systeme zum Hinterlassen von Anmerkungen an realen Orten (Fitzmaurice 93). In Zukunft sind Navigationsanwendungen denkbar, sofern eine Nutzer- und Objektverfolgung in freier Außenumgebung realisiert werden kann. Weiterhin könnten zukünftige oder vergangene Ereignisse oder Gebäude in reale Umgebungen eingeblendet und so direkt erfahrbar gemacht werden. Die notwendige Genauigkeit bei der Verfolgung der Nutzerbewegungen ist jedoch bis heute nicht zufriedenstellend gelöst. Ein weiteres Problem stellt die Latenz eines Systems dar: Damit der Fehler bei der Registrierung virtueller Objekte mit der Realwelt bei einer Kopfbewegung von ca. 50 Grad pro Sekunde kleiner

als 0,5 Grad beträgt, darf die Latenzzeit nicht mehr als 10 ms betragen⁴, typische Systeme besitzen jedoch eine Latenzzeit von etwa 100 ms, was schon in einer Entfernung von nur 70 cm zu einem Registrierungsfehler von 6 cm führt (Azuma 97).

Eine andere Anwendung der *Augmented Reality*, jedoch ohne Registrierungsprobleme, wird in Gibbs (96) beschrieben. Die Verwendung eines wandgroßen Displays für eine Telekommunikation soll in *TelePort* bestimmte Einschränkungen üblicher Videokonferenzumgebungen bewältigen. Die synthetische Szene verändert den Betrachterstandpunkt analog zu den Benutzerbewegungen. Durch die Integration des Displays in die Wand eines realen Raums, erscheint der virtuelle Raum als eine Erweiterung des realen und gibt einen Eindruck eines einzigen, kontinuierlichen Raums wieder. Ein weiterer Vorteil ist hier die Möglichkeit, die Körpersprache des Konferenzteilnehmers einzubeziehen, also eine gewisse Peripherie, die bei traditionellen Systemen nicht vorhanden ist. Das System erlaubt durch sein spezielles Setup eine natürlichere Sicht auf eine virtuelle Welt und verwendet die Körperbewegungen des Nutzers im Raum, die Realumgebung wird jedoch nicht explizit einbezogen. Anders verhält es sich beim *DigitalDesk* (Wellner 93), bei dem virtueller und physikalischer Raum durch eine Overlay-Technik integriert werden. Motivation ist die Feststellung, daß wir bei der Interaktion mit Dokumenten in zwei verschiedenen Welten, der elektronischen und der physikalischen, arbeiten. Die Interaktionsarten beider Welten sind dabei nicht aufeinander abgestimmt, Funktionen sind unterschiedlich, eine Integration beider Welten ist beschränkt. Die Art und Weise einer physikalischen Interaktion mit elektronischen Dokumenten ist im Vergleich zu der mit Papier, Bleistift, Lineal und anderen traditionellen Hilfsmitteln sehr eingeschränkt. Trotzdem glaubten Enthusiasten der VR-Technologie, im Virtuellen alle Grenzen der physikalischen Welt durch flexiblere, virtuelle Alternativen aufheben zu können, sie sagten beispielsweise das papierlose Büro voraus (Wellner 93). Der *DigitalDesk* projiziert elektronische Bilder auf einen normalen Schreibtisch und reagiert auf Interaktionen mit Bleistift und den bloßen Fingern. Weiterhin können Papierdokumente eingelesen werden. Statt eine Computerumgebung also ähnlich einem physikalischen Schreibtisch zu gestalten, wird hier das Gegenteil, der Schreibtisch wird dem Computer ähnlicher, verfolgt. Diese Anwendung, wie auch ein ähnliches System (*MetaDESK*, Ishii 97), erweitern die physikalische Umgebung durch eine Betrachtung in einem doch zweidimensionalen Aufbau. Für die Ergänzung um dreidimensionale Funktionen sind zusätzliche Techniken erforderlich, wie etwa am Beispiel der Workbench von Krüger (94) oder einige der im nächsten Abschnitt beschriebenen Techniken.

2.2.3.3 Interaktion mit Realobjekten

Einen anderen Weg als den der *Augmented Reality* gehen einige Anwendungen, die konkrete, „normale“, Realobjekte einbeziehen. Die Grundidee ist die Verwendung des großen Interaktionspotentials der Hände bzw. des Körpers mit physikalischen Objekten. Eine Idee, der auch die Entwicklung des Datenhandschuhs und Datenanzugs zugrunde lag, hier jedoch ohne eine notwendige Verkabelung, durch eine Koppelung von digitaler Information mit realen Artefakten verfolgt wird. Ein Beispiel für derartige Ansätze ist die in Ishii (97) beschriebene *Marble Answering Machine* von Durell Bishop, die bei jedem Anruf eine reale Kugel instanzierte, die der Benutzer bei Bedarf mitnehmen oder auf einem Telefon plazieren konnte, um die Nummer des Anrufenden zu wählen, oder die enthaltene Nachricht abzuspielen. Ein weiteres Beispiel ist das Projekt *LEGO/Logo*, in dem LEGO-Bausteine zu programmierten Maschinen zusammengesetzt werden können, während die Bausteine selbst in ihrer Ursprungsform verbleiben (dieser Ansatz hat inzwischen Produktreife erlangt und wird kommerziell vertrieben). Ebenfalls finden sich einige

⁴ Allein das Auslesen des Bildspeichers für ein 60 Hz Display benötigt zur Zeit noch 16.67 ms (Azuma 93).

künstlerische Anwendungen: In der Installation *Legible City* wird ein reales Fahrrad zur Navigation und Fortbewegung in einer virtuellen Stadt verwendet (Shaw 90), die Installation *Interactive Plant Growing* verwendet reale Pflanzen, deren Ströme bei Berührung das Wachstum virtueller Pflanzen anregen (Sommerer 93); in der Installation *A-Volve* wird eine Wasserlandschaft mit virtuellen Kreaturen durch ein als Interface fungierendes Wasserbecken erfahrbar gemacht (Sommerer 94). Eine weitere Installation verwendet einen Tisch und einige Realobjekte, zur Integration eines virtuellen Buches in die Umgebung (*Beyond Pages*, Fujihata 96).

Ein Experiment einer narrativen Umgebung, in der das Aufnehmen von Objekten, das Berühren der Wand oder eine Bewegung in eine bestimmte Richtung eine Veränderung der Beleuchtung verursacht oder Videosequenzen und Töne abspielen läßt, wird in Druin (94) beschrieben. Die Verknüpfungen werden mit Hilfe einer visuellen Programmieranwendung, einiger Sensoren und einer Schnittstelle zur Digital-Analog-Wandlung realisiert. Eine mit Kameras ausgestattete, erweiterte narrative Umgebung wird in Bobick (97) beschrieben. Ziel war die Entwicklung eines narrativen Spielraums für Kinder, der die Anwesenheit mehrerer Personen gleichzeitig erlaubt und vollständig automatisiert auf die Aktionen der „Benutzer“ reagiert. Im Gegensatz zu der oben erwähnten einfachen Zuordnung von Sensordaten zu Systemreaktionen, werden in dieser *KidsRoom* genannten Umgebung bestimmte Aktionen durch Meßergebnisse und die Einbeziehung des Kontextes erkannt. Der Raum besteht aus mehreren sensitiven Zonen, zwei Projektionswänden und einem realen Bett (zur Zeit das einzig bewegliche reale Artefakt). Im Verlauf einer Geschichte, die durch einfache 2D-Animation und Sprache im Raum erzählt wird, werden bestimmte, zur Geschichte passende Ereignisse erkannt (z.B. ob alle Personen sich auf dem Bett befinden). Dabei soll sich die Aufmerksamkeit der Zuhörer nicht primär auf die Leinwand beziehen, sondern im physikalischen Raum verteilen. Hauptfokus liegt hier in der sinnvollen Erzeugung eines Kontextes, durch den die verwendeten Mustererkennungsverfahren einfach gestaltet werden können. Auch gibt es starke Einschränkung bzgl. der Beleuchtung (sie muß zur kamerabasierten Erkennung konstant sein), die teilweise durch narrative Mittel überbrückt werden können (indem Beleuchtungsveränderungen zu festgelegten Zeitpunkten erfolgen, zu denen keine Erkennung notwendig ist). Eine Schwierigkeit dieser Umgebung ist, daß je mehr physikalische Objekte verwendet werden, desto unvorhersehbarer gestalten sich die möglichen Verhaltensweisen der verfolgten Personen. Ein Vorteil scheint zu sein, daß die Personen bei dem Vorhandensein einer Geschichte leichter mit ihren Handlungen kooperieren (Bobick 97).

Die Verwendung von Realobjekten als Teil einer Benutzerschnittstelle wird in dem System *Build-it* als *Natural User Interface* (NUI) bezeichnet (Rauterberg 97). Es basiert auf den in Fitzmaurice (95) vorgestellten *Bricks*, kleinen klötzchengroßen Bausteinen, die zur Interaktion mit einer digitalen Umgebung zur Konstruktionsplanung von Industrieanlagen auf einem Tisch eingesetzt werden. Auf einer horizontalen Arbeitsfläche, ähnlich der des *DigitalDesk*, wird der Grundriß einer geplanten Anlage dargestellt. Die *Bricks*, in Rauterberg (97) auch *Universal Interaction Handler* genannt, dienen dazu, Objekte aus einem Menü auszuwählen, sie auf der Grundfläche zu positionieren oder zu rotieren. Die Objekte lassen sich von dem *Brick* lösen und später erneut selektieren oder auch löschen. Bestimmte Operationen, wie das Skalieren oder die Veränderung der vertikalen Lage der Objekte, lassen sich über ein weiteres Menü am Rand des Tisches aufrufen. Eine dreidimensionale Ansicht der Szene wird auf einer Projektionswand zusätzlich dargestellt. Durch diese Art der Interaktion werden die oben angesprochenen Probleme von *Augmented Reality* und gleichzeitig auch viele Beschränkungen der digitalen Welt bewältigt: *Bricks* erlauben ein taktiles Feedback, also eine Berührungsinformation, und liefern eine Tiefeninformation. Dieses „greifbare“ Interface erleichtert nicht nur die Interaktion durch eine direktere und manipulierbarere Arbeitsweise, sondern schafft eine externe Manife-

station von traditionell internen Repräsentationen von Computermodellen und -abläufen. Zusätzlich können mehrere Personen auf natürliche Weise gleichzeitig mit dem System arbeiten.

Ebenfalls eine Erweiterung des DigitalDesk findet sich mit dem *MetaDESK* (Ishii 97): Hier werden gleich mehrere verschiedene reale Objekte auf einem Tisch integriert, um mit der virtuellen Umgebung zu interagieren. Derartige reale Objekte werden als *phicons* (physical icons) bezeichnet. In einer Anwendung, die eine Landkarte der Umgebung zeigt, können Modelle von dort befindlichen Gebäuden auf dem *MetaDESK* platziert werden. Die Darstellung der Landkarte wird entsprechend der Position und Lage eines *phicons* dargestellt. Wird ein zweites Gebäude hinzugefügt, so wird die Karte zusätzlich entsprechend der relativen Lage der Modelle skaliert. Eine weitere Interaktionsmöglichkeit ist das Setzen von virtuellem Licht oder die globale Veränderung der Darstellungsart mittels einer realen Taschenlampe. Bei Ishii (97) ist diese Anwendung in den größeren Kontext der *Tangible Bits* eingebettet. Basierend auf dem *Ubicomp*-Prinzip und mit ähnlichen Überlegungen wie die in Weisers Artikel zur *Calm Technology* werden neben den greifbaren physikalischen Objekten des *MetaDESKs* zwei weitere Forschungsbereiche konkretisiert: Interaktive Oberflächen (*transBOARD*) und sogenannte ambiente Medien in dem Projekt *ambientROOM*. Die ambienten Medien sollen analog dem Gedanken von Zentrum und Peripherie dem Benutzer die Möglichkeit geben, digitale Hintergrundinformation in der Peripherie der realen Umgebung aufzunehmen, und zwar auf die gleiche Weise, wie dem Menschen ein Wetterwechsel durch die Veränderung des ambienten Lichtes bewußt wird. So kann ein Modellauto des *ambientROOM* dazu verwendet werden, die Zugriffsrate auf eine bestimmte Internet-Seite anzuzeigen: Wird dieses Auto in die Nähe eines Lautsprechers gestellt, so kann ein Audiosignal im Hintergrund, etwa Regentropfen, die Aktivität anzeigen. Der Benutzer nimmt die Audioinformation meist als Hintergrundgeräusch wahr, ändert sich jedoch das Signal abrupt, so wird er darauf aufmerksam und die Information gerät aus der Peripherie wieder in das Zentrum der Aufmerksamkeit. Ein Umstand, der nicht nur durch die Veränderung des Audiosignals selbst möglich wird, sondern auch durch die Situation des Benutzers zu dem physikalischen Objekt verstärkt wird, das während seiner Hintergrundaktivität gleichsam als Teil des in Kapitel *Körper und Raum* beschriebenen Horizonts bestehen bleibt.

An dieser Stelle ist die Verbindung zum Ansatz der Intuitiven Schnittstelle zu sehen, die unter Berücksichtigung des Situiertsein des Benutzers die in Ishii (97) angesprochenen Konzepte berücksichtigt. Das Konzept der Intuitiven Schnittstelle geht dabei das Problem nicht von einer Trennung der Konzepte von *Augmented Reality*, *Responsive Environments* oder auch der *Tangible Bits* an, sondern rückt den Körper des Menschen mit seiner Leibbewegung in den Vordergrund. Hiermit wird der Raum, in den sich sowohl „fühlbare Bits“ als auch Overlay-Techniken integrieren lassen, selbst zum Interface.

2.3 Konzept der Intuitiven Schnittstelle

Die Beobachtungen des letzten Abschnittes zeigen eine große Anzahl von Forschungsprojekten, denen, das sei noch einmal betont, der gemeinsame Wunsch nach intuitiver Kommunikation mit dem Computer zugrunde liegt. Das Konzept der Intuitiven Schnittstelle greift diesen Wunsch ebenso auf, vollzieht dabei jedoch nicht die erwähnte Trennung einzelner Bereiche, sondern rückt den Benutzer in den Mittelpunkt der Überlegungen, indem seine Bewegungen im Raum explizit als Teil der Schnittstelle berücksichtigt werden. Der Realraum wird dabei nicht nur relatives Referenzsystem, er wird vielmehr Teil der Schnittstelle selbst, zu der der Benutzer sich in Situation befindet. In Bezug auf Gestikererkennung werden hier nicht komplexe Gesten erkannt, sondern sich auf einfache und intuitive Gesten beschränkt.

2.3.1 Überblick

Das Szenario für die Intuitive Schnittstelle ist ein einzelner Raum mit einer Projektions-einrichtung und verschiedenen alltäglichen Objekten aus der Umgebung des Benutzers (Stuhl, Tisch, Regal u.ä.). Als Benutzer setzen wir für das Szenario präziser eine Person vor, die aus dem Arbeitsbereich künstlerischer/gestalterischer Bildproduktion stammt (z.B. einen Filmregisseur), eine Person also, die gewohnt ist, eher mit der Stimme, mit Körperbewegungen und Gestik im Produktionsraum zu agieren als in ruhiggestellter sitzender Haltung vor der programmierten Maschine. Wir konstruieren also einen Benutzertypus, dessen Arbeitsweise ganz spezielle Anforderungen an das Medium Computer stellt und eine multimodale Schnittstelle mit unterschiedlichen Kommunikationskanälen erfordert. Für den eher intuitiven und den ganzen Körper einbeziehenden Umgang mit dem Computer sollen Positionen des Benutzers, seine Konzentrationsrichtung, die Blickrichtung, aber auch die Unruhe im Raum erkannt werden. Zusätzlich sollen zur Manipulation von Objekten und zur Eingabe einfacher Befehle neben der verbalen Befehl-seingabe einfache Gestiken wie demonstrative Verweise oder Anweisungen erkannt und algorithmisch umgesetzt werden. Im folgenden werden als Beispielanwendungen der Intuitiven Schnittstelle das Verwalten von Informationseinheiten im Raum und ein *inter-aktives Filmplanungssystem* verwendet, wie es in einem Projekt der Kunsthochschule für Medien Köln bereits als Idee entwickelt worden ist (Fleischmann 94-1/95). Ziel dieser Anwendung ist es u.a., in der Planungsphase eines Films unterschiedliche Kameraein-stellungen an einer computergenerierten Szene testen zu können.



Abbildung 2-1: Szenario

Auf der Projektionsfläche wird eine computergenerierte Szene mit verschiedenen Objekten präsentiert. Der Benutzer kann die Objekte bewegen, indem er auf sie zeigt, sie mit einer sprachlichen Anweisung auswählt und anschließend auf der Projektion durch er-

neutes Zeigen an eine andere Stelle verschiebt. Als weitere Möglichkeit, Objekte an einem anderen Ort zu platzieren, kann sich der Benutzer analog im Realraum bewegen; die relative Distanz zwischen der Ausgangsposition bei der Auswahl des Objekts und der Endposition wird verwendet, um das simulierte Objekt in der 3D-Szene entsprechend zu positionieren. Zur weiteren Vereinfachung können im Realraum Objekte in Form von Platzhaltern verwendet werden, um die Position von Objekten der 3D-Szene und auch ihre relative Positionierung zueinander anzuzeigen. Dadurch können räumliche Abhängigkeiten der computergenerierten Objekte im Realraum diskutiert und beurteilt werden, ohne auf Hilfsmittel wie orthographische Ansichten der 3D-Szene angewiesen zu sein, wie sie bei 3D-Animationssystemen verwendet werden (Hierbei wird eine 3D-Aktion in eine Folge von zweidimensionalen Aktionen aufgeteilt). Zur Platzierung neuer Objekte werden diese aus einer Art Requisite, einem digitalen Archiv, ausgewählt. Hierbei wird davon ausgegangen, daß diese Objekte in einer Modellierungsphase an einem herkömmlichen Arbeitsplatz schon erstellt worden sind. Zur Auswahl von Objekten können durch einfache Zeigegestik in Richtung des Projektionsrandes Menüs in die Projektion eingeblendet werden, aus denen dann ein passendes Objekt gewählt werden kann. Eine weitere Möglichkeit ist das Einblenden der Menüs in Abhängigkeit von der Position des Benutzers im Raum. So können beispielsweise (virtuelle) Möbelstücke links hinten im Realraum positioniert werden, (virtuelle) Fahrzeuge rechts hinten. Zum Platzieren eines Stuhls in der 3D-Szene würde sich der Benutzer dann in den hinteren, linken Bereich des Realraums bewegen, wodurch ein Auswahlmenü mit Möbelstücken auf der Projektion sichtbar wird. Mit einem gestischen Hinweis kann dann ein Stuhl ausgewählt werden. Diese nach dem Prinzip der *Ars Memoriae* (Yates 66) aufgebaute *Gedächtnisfunktion* ermöglicht einerseits das Aufrufen von Menüs auf intuitive Weise und ist zusätzlich einfacher zu erinnern als eine Funktion in einem hierarchisch aufgebauten Menüsystem.

2.3.2 Technischer Aufbau

In Abbildung 2-1 ist die räumliche Situation der Schnittstelle für den Prototyp in einem Labor der Kunsthochschule für Medien dargestellt. Der Benutzer interagiert typischerweise in einer Entfernung zwischen drei und fünf Metern vor der Projektionswand, die in einer Wand des Labors eingelassen ist. Zur Zeit wird ein Stereoaufbau mit zwei Kameras, die sich oberhalb der Projektionswand befinden, verwendet, für zukünftige Anwendungen ist die Verwendung von weiteren Kameras denkbar, wodurch der gesamte Raum als Interaktionsraum zur Verfügung stehen würde. Für die meisten Operationen wird ein Sprachkommando im Zusammenhang mit der Zeigeoperation verwendet, das Bewegen von Objekten ist alternativ auch durch die Verwendung von realen Platzhalterfiguren möglich. Für die Spracherkennung konnte auf ein kommerzielles Produkt zurückgegriffen werden (vgl. Kapitel 3). Für den in Kapitel 6 beschriebenen Prototyp zur Filmplanung werden zwei PCs verwendet: Einer für das Bildverarbeitungssystem und die Sprachverarbeitung, ein anderer für die Darstellung der Benutzerschnittstelle. Die Softwarearchitektur der intuitiven Schnittstelle (siehe Kapitel 5) erlaubt jedoch die Anbindung beliebiger Rechner zur Darstellung der Benutzerschnittstelle. So ist auch die Anbindung einer Hochleistungs-Grafikworkstation zur Darstellung von realistischen 3D-Welten in Echtzeit möglich (in der unten beschriebenen Anwendung *Trace/Pattern* wurde beispielsweise eine SiliconGraphics Workstation O² verwendet).

2.3.3 Anwendungen

Im folgenden werden zwei Anwendungen beschrieben: Die Verwaltung von Informationseinheiten im Raum verdeutlicht das Prinzip der *Ars Memoriae*. Der Prototyp zur intuitiven Filmplanung dient als Basis der in Kapitel 6 beschriebenen empirischen Untersuchung der Konzepte der intuitiven Schnittstelle. Anschließend werden drei interaktive

Installationen beschrieben, die zeigen, daß sich die Softwarearchitektur der Intuitiven Schnittstelle auch für diesen künstlerischen Kontext als nützlich erwiesen hat.

2.3.3.1 Verwalten von Informationseinheiten

In dieser Anwendung interagiert der Benutzer durch seine Körperbewegungen und -positionen im Raum. Die Bewegungen werden zur Navigation in und Manipulation von Daten- und Informationseinheiten verwendet, Kommandos werden durch Sprache oder eine Zeigegestik ausgelöst. Ähnlich der Idee der *situierten Informationsräume* von Fitzmaurice (93) kann der Benutzer Informationseinheiten, wie etwa Internet-Verweise, Daten oder Klangdateien, an einer beliebigen Stelle des Raums vor der Projektionswand ablegen und sie zu einem späteren Zeitpunkt abrufen, indem er sich einfach an die Raumposition begibt. Abbildung 2-2a zeigt die Situation der Beispielanwendung. Auf der Projektion werden im linken Teil ein Netscape-Fenster, im rechten eine Kontrollszene dargestellt. Die Kontrollszene (Abbildung 2-2b) zeigt die aktuelle Auswahl der gespeicherten Informationen in Abhängigkeit von der Benutzerposition (hier als Liste von Pictogrammen mit dem Netscape-Logo und dem Titel der entsprechenden Internetseite als erklärenden Text). Zusätzlich wird zu Kontrollzwecken während der Entwicklungsphase die Benutzerposition innerhalb der 3D-Szene des Labors angezeigt. Die einzelnen Verweise lassen sich durch ein Sprachkommando aktivieren, die entsprechende Internet-Seite erscheint anschließend in dem Netscape-Fenster auf der linken Projektionsseite. Eine weitere Funktion dieser Anwendung erlaubt das Zeigen auf Verweise im Netscape-Fenster mittels eines Zeigestabs (Abbildung 2-2a) und Auswählen durch ein Sprachkommando. Die Anwendung macht sich das Prinzip der *Ars Memoriae* (Yates 66) bzw. der Gedächtnisfunktion der Intuitiven Schnittstelle zu Nutze und basiert auf der Idee, daß ein realer Ort zum Ablegen von Daten einfacher zu erinnern ist als die Auswahlmöglichkeit in einem hierarchischen Dateisystem.

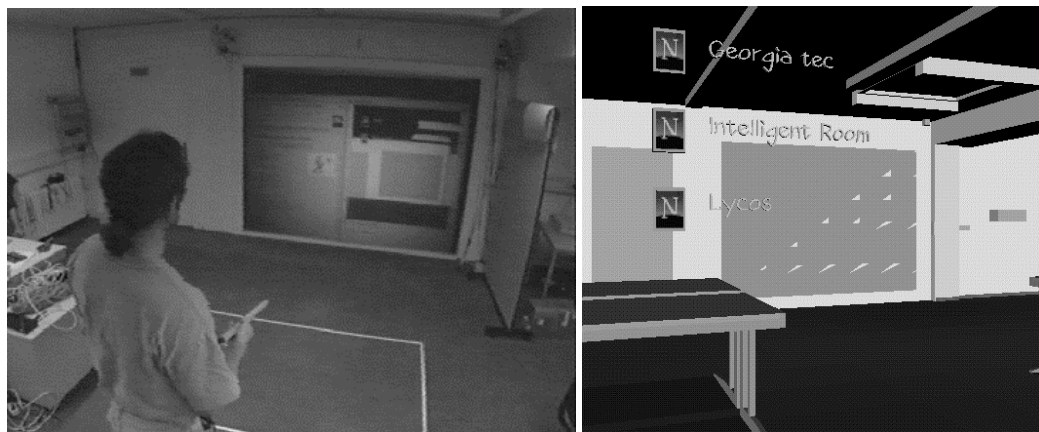


Abbildung 2-2: Internet-Verweise können im Raum aufgerufen werden: a) Situation der Anwendung, b) Kontrollbild mit aktueller Auswahl

Als Orte, die der Benutzer zum Speichern der Daten auswählt, kommen vorzugsweise Positionen in der Nähe von Realobjekten in Betracht, die einfach zu erinnern sind, wie etwa ein Tisch, ein Regal, ein Fenster oder auch Orte wie die Mitte oder Seite des Raums. Ebenso könnte die Höhe verwendet werden, um Daten zu organisieren, indem beispielsweise weniger wichtige Informationen in der Nähe des Bodens positioniert werden, sehr wichtige Informationen in Augenhöhe. Die hier beschriebene Beispielanwendung realisiert zum jetzigen Zeitpunkt das Aufrufen von Informationseinheiten an bestimmten

Raumpositionen (sowie die beschriebene Zeigegestik). Mit Erweiterungen des Systems sind aber auch die folgenden Interaktionsarten leicht vorstellbar:

- Um eine Informationseinheit an einem bestimmten Ort zu plazieren, wird sie vom Benutzer auf der Projektion selektiert. Anschließend begibt er sich an den gewünschten Ort im Realraum und legt sie dort z. B. mittels eines weiteren Sprachkommandos ab.
- Um die Informationseinheit wieder abzurufen, begibt sich der Benutzer in die Nähe des Ortes, an dem er die Einheit erinnert. Die Anwendung zeigt die aktuelle Auswahl der hier abgelegten Einheiten, von der eine selektiert werden kann. Verschiedene zwei- oder dreidimensionale Ansichten sind hier vorstellbar, um auf die verschiedenen Benutzerpräferenzen einzugehen.
- Ein Umschalten zwischen verschiedenen Sets von Informationseinheiten ist durch ein Sprachkommando möglich und läßt den Benutzer verschiedene Datensets im Raum verwalten, so daß der gleiche Ort zum Speichern und Abrufen verschiedener Informationen verwendet werden kann (Yates 66).
- Informationseinheiten können auch an festen Positionen im Körperraum des Benutzers abgelegt werden, ganz analog zu den sogenannten *umgebungsfixierten Fenstern* (surround-fixed-windows), wie sie in Feiner (93-2) beschrieben werden. Selektierte Informationseinheiten werden hierbei durch ein Sprachkommando „in Reichweite“ des Benutzers abgelegt (links oben, rechts unten etc.). Diese Daten werden fortan relativ zu der Position des Benutzers fixiert und sind somit an der gleichen Stelle abrufbar, auch wenn sich der Benutzer im Raum bewegt. Diese Art der Interaktion kann verwendet werden, um eine Gruppe von Einheiten, etwa eine Menge von Internet-Verweisen oder ein Set von Funktionen zur Kamerakontrolle, bei freier Beweglichkeit im Raum stets verfügbar zu halten.

Eine weitere Möglichkeit, die durch die Verbindung von Daten mit Orten des Realraums ermöglicht wird, ist eine Verwendung von selbstorganisierenden Techniken, wie sie in Fleischmann (94-2) beschrieben werden. Dort kann eine selbstorganisierende Klangdatenbank in einem realen Aktionsraum erforscht werden. Klänge bilden Gruppen mit ähnlichen Eigenschaften, die sich kontinuierlich in vorgegebenen Grenzen „im Realraum“ fortbewegen. In Bezug auf die Verwaltung von Informationseinheiten könnten Verknüpfungen zwischen Daten geschaffen werden, die es ermöglichen, einfacher auf Informationen zuzugreifen. Eine Methode, die auch bei den sogenannten agentenbasierten Methoden Anwendung findet, dort jedoch ohne die zusätzliche Hilfe einer Manifestation der Information in der natürlichen Umgebung eines Benutzers, dem Realraum. Die Verwendung des Realraums könnte diese Methoden im wahrsten Sinne des Wortes leibhaftig erfahrbar machen.

2.3.3.2 Intuitive Filmp lanung

Für die Planung und Vorbereitung von Filmen werden üblicherweise Storyboards verwendet. Storyboard sowie auch Aufriß sind Strichzeichnungen oder Skribbles der Schlüsselszenen eines Films (Abbildung 2-3). Die Zeichnungen des Storyboards zeigen einen ungefähren Eindruck der Kameraeinstellungen und geben eine gute Beschreibung, oft durch ergänzenden Text, des Szenenaufbaus und der Aktionen in der Szene. Im Aufriß werden die wesentlichen Objekte, Personen und die Kamera z. B. aus der Vogelperspektive gezeichnet, um die Einstellungen und Abläufe der Szene im ganzen darzustellen.

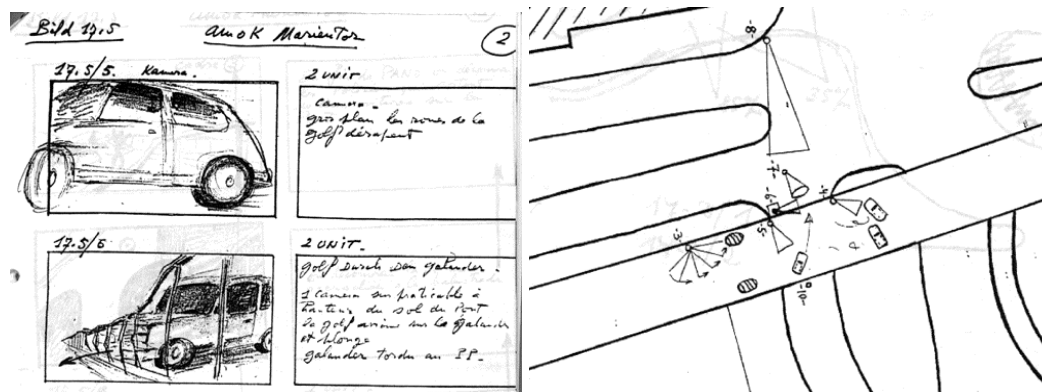


Abbildung 2-3: Traditionelles Storyboard und Aufriß

Diese Hilfsmittel haben sich als Gesprächsgrundlage für Regisseur, Kameramann, Ausstatter sowie Produktionsleiter bewährt und sind ein wichtiges Kommunikationsmittel. Der Nachteil liegt darin, daß Bewegungseindrücke nur ungenügend wiedergegeben werden, Kamerafahrten beispielsweise lassen sich gar nicht darstellen. Besonders für die Vorbereitung komplizierter Szenen ist eine genaue Planung jedoch sehr wichtig, da es am Set aus Kostengründen in der Regel nicht möglich ist, verschiedene Varianten durchzuspielen. Auch kommt es trotz Storyboards immer wieder zu Mißverständnissen zwischen Kameramann und Regisseur, die u.U. eine Wiederholung der Szene notwendig machen. Der Prototyp zur intuitiven Filmp lanung realisiert unter Verwendung der Intuitiven Schnittstelle einige wesentliche Komponenten eines interaktiven Filmp lanungssystems. Eine detaillierte Beschreibung dieser Anwendung zusammen mit den Ergebnissen der empirischen Untersuchung findet sich in Kapitel 6.

2.3.3.3 *VbodyRbody*

VbodyRbody ist eine interaktive Installation von Akke Wagenaar, Masahiro Miwa, Michael Hoch und Matthias Melcher, ausgestellt in Köln 1996. Die Idee für dieses Projekt entstand 1995, 50 Jahre nach dem Abwurf der Atombomben auf Hiroshima und Nagasaki. Die Installation behandelt dieses Thema in einer sehr abstrakten Weise: Die Geschichte wird auf der Ebene der Einzelschicksale in Erinnerung gebracht durch die Interaktion des Betrachters mit einer virtuellen menschlichen Gestalt. Das Benutzerinterface besteht aus einer Infrarotkamera, die die Bewegungen des Betrachters vor der Projektion registriert (Abbildung 2-4). Das Kamerasignal mit den Betrachterbewegungen wird analysiert und steuert die Bewegungen der Figur. Hierbei werden die Körperbewegungen von Kopf, Händen und Beinen registriert, um die Figur entsprechend zu bewegen (Abbildung 2-4a). Wird eine starke Bewegung über einen gewissen Zeitraum festgestellt, imitiert die Figur nicht mehr die Bewegungen, sondern führt ein „Eigenleben“ (Abbildung 2-4b).



Abbildung 2-4: Interaktive Installation *VbodyRbody*: a) räumliche Situation mit Projektion, b) Besucher und Figur nach starker Bewegung durch den Besucher

2.3.3.4 *Trace/Pattern*

Dieses Projekt ist eine interaktive Installation von Ursula Damm, gezeigt auf der ArtCologne 1997 und auf dem Medienforum NRW 1998. Das Softwaresystem der Intuitiven Schnittstelle wird hier zur Verfolgung der Bewegungsbahnen von Personen in einem Kamerabild verwendet. Die so extrahierten Daten werden über Netzwerk von einem Grafikprogramm angefordert, das aufgrund der Bewegungsdaten ein Liniengeflecht erzeugt, das die aktuellen Aufenthaltsorte der Menschen als Territorien anzeigt und Nachbarn grafisch miteinander verbindet (Abbildung 2-5).

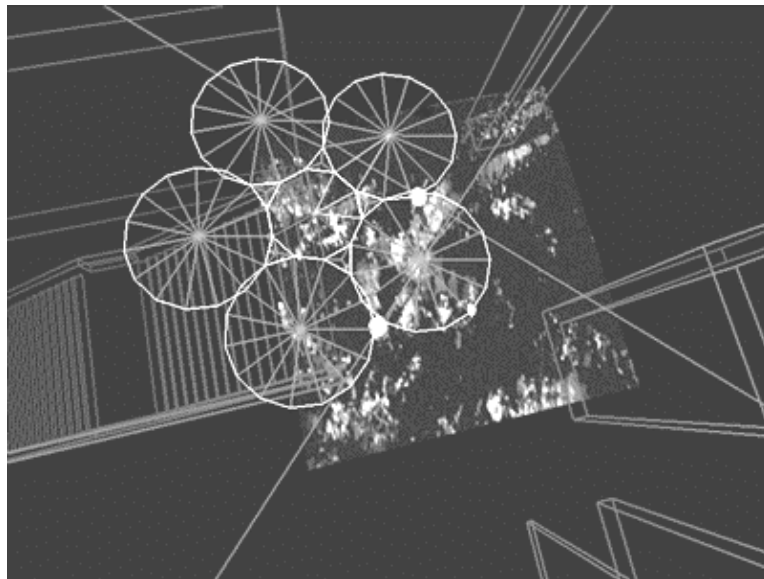


Abbildung 2-5: Generierung von Mustern ausgelöst durch die Bewegung von zwei Personen (Installation *Trace/Pattern*)

Aus den Positionsdaten werden die Hauptgerichtungen und Schwerpunkte berechnet und das Abstandsverhalten der Personen auf Proportionen hin geprüft (beispielsweise auf Basis des goldenen Schnitts). Die dadurch entstehenden lokalen Geometrien werden so miteinander in Verbindung gesetzt, daß ein globales, den Ort überspannendes Netz von Mustern entsteht. Diese Muster werden als Tonhöhen hörbar gemacht und werfen die Frage auf, ob in unserem Verhalten im Raum „harmonische“ Verhältnisse (im Sinne der althergebrachten Proportionslehre) einen besonderen Stellenwert haben. In dieser Installation sind bisher die Positionen von zwei Menschen und deren Bewegungsrichtung berechnet worden, geplante Erweiterungen sollen es erlauben, auch mehrere Personen zu verfolgen und auch deren Interaktion untereinander in Beziehung zu setzen. Der Mensch wird somit nicht nur in seiner Position, sondern auch in seiner motorischen Ausdrucksweise erfaßt. Das System erlaubt damit gleichermaßen die Thematisierung von Mensch-Maschine- wie auch Mensch-Mensch-Kommunikation.

2.3.3.5 *LightSpace - Experimente*

Dieses Projekt zeigt die Ergebnisse erster Experimente zur Erzeugung eines interaktiven Lichtraums. Der Künstler Seth Riskin verwendet in seinen Performances Licht, um die Bewegungen seines Körpers zu erweitern und sie im ganzen Raum zu verteilen, bisher unter Benutzung fast ausschließlich analoger Techniken. Die Schaffung eines interaktiven Lichtraums erweitert diese Möglichkeiten durch die Erzeugung von räumlichen Strukturen aus den Bewegungen des mit Kameras verfolgten Akteurs.

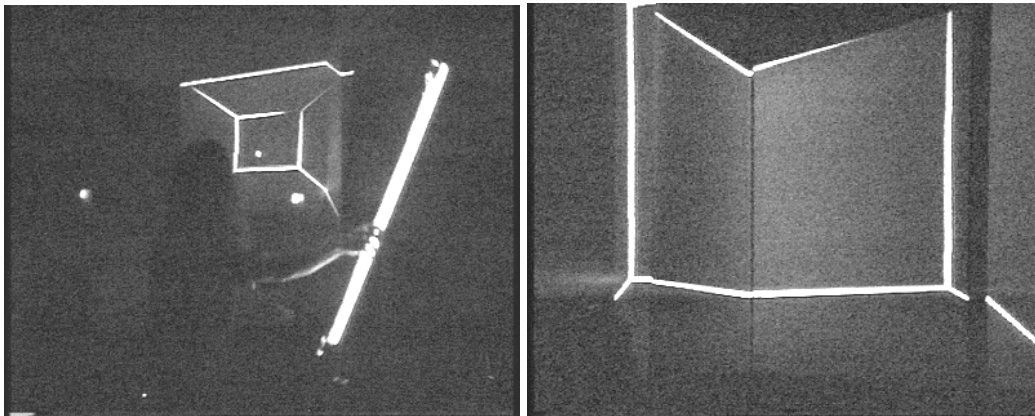


Abbildung 2-6: Interaktiver Lichtraum, gesteuert durch einen Lichtstab

In den ersten Experimenten wurde das Softwaresystem der Intuitiven Schnittstelle verwendet, um ein Lichtstab mit einem Stereosystem zu verfolgen (Abbildung 2-6a). Im Gegensatz zur farbbasierten Personenverfolgung kamen hier sehr preiswerte Einplatinen-Schwarzweiß-Kameras zum Einsatz. Die Konfiguration des Lichtstabs im Raum kontrollierte einen als Drahtgittermodell definierten virtuellen Raum, der durch eine Projektion neue Strukturen im Realraum erzeugt (Abbildung 2-6b). Für dieses System wurde ein einziger PC verwendet, d.h. in diesem Fall wurde für das Anwendungsprogramm der gleiche PC verwendet wie für das Softwaresystem der Schnittstelle.

3 Interaktionstechniken

Die Gestaltung von interaktiven Systemen beschränkt sich nicht allein auf die Hard- und Software der Systeme, sondern bezieht insbesondere wissenschaftliche Elemente der Mensch-Maschine-Kommunikation und Interaktionskonzepte, die Hard- und Software verbinden, mit ein. Für grafische interaktive Systeme haben sich im Laufe der Zeit eine Reihe von Interaktionstechniken, wie etwa die *direkte Manipulation* oder das *WYSIWYG-Prinzip* (what you see is what you get), bewährt. Meist ist die Kombination zwei oder mehrerer Interaktionstechniken besser geeignet, die Bedienung zu erleichtern, als eine strikt durchgängige Verwendung einer Technik allein. Auch die Verwendung von Raum erleichtert durch das räumliche Erinnerungsvermögen des Menschen den Umgang mit dem Computer, jedoch bezieht sich die Nennung von Raum meist auf den Raum, der auf dem Bildschirm bzw. der Projektionswand zur Verfügung steht. Dieser Raum ist jedoch auf die Ausmaße des Bildschirms beschränkt, so daß sehr oft eine neue Information die alte ersetzt. Der Benutzer muß daher allein den Inhalt erinnern, der Bezug der Information zum Ort und zum Zusammenhang geht verloren. Um trotzdem eine visuelle Kontinuität zu erreichen, wird in (Bolt 84) u.a. vorgeschlagen, sich Analogien des Apparates Kinos zu bedienen, aber es wird nicht erwähnt, daß sich der Besucher des Kinos freiwillig der Situation unterwirft, dagegen ein Benutzer einer Computerschnittstelle stets erwartet, die Kontrolle über die angebotenen Vorgänge zu haben. Wird zusätzlich der Raum, der den Benutzer umgibt, also der Realraum, als Interaktionsraum in die Schnittstelle integriert, so entsteht eine völlig neue Situation und es bedarf auch neuer Interaktionsformen. In manchen Fällen können sprachliche Anweisungen zusammen mit einer richtungsweisenden Gestik für eine einfache präzise Interaktion verwendet werden. Eine andere Möglichkeit ist das Einblenden von großflächigen Menüs bei der Erkennung einer Zeigegestik, die eine einfache, eher grobe Auswahl ermöglichen. Auch können szenenabhängige Hinweise auf die Zeigerichtung, etwa das Aufleuchten von Objekten, auf die gezeigt wird, hilfreich sein, dem Benutzer jene Form von Aufmerksamkeit zu signalisieren, wie sie bei der Mensch-Mensch-Kommunikation notwendig ist.

Die Anwendbarkeit solcher bildlicher Bezüge für die intuitive Schnittstelle bedarf eingehender Untersuchungen und wurden an einem Anwendungsbeispiel getestet. Im Rahmen dieser Arbeit kann dabei keine umfassende Untersuchung der einzelnen Interaktionsmethoden durchgeführt werden, auch werden keine völlig neuartigen Konzepte betreffend der graphische Gestaltung der Benutzeroberfläche vorgestellt. Hauptaugenmerk liegt in der Exploration der Möglichkeiten, die eine Interaktion im Raum bietet, und die Verwendung von Menüs bzw. Alternativen zur Auswahl von Objekten in dieser Situation. Zu diesem Zweck wird im folgenden erst auf die Gestaltungsprinzipien interaktiver Systeme eingegangen, um anschließend eine Einordnung der intuitiven Schnittstelle vorzunehmen. Nach der Benennung der Basisinteraktionen (der intuitiven Schnittstelle) folgt im Abschnitt Dialoggestaltung die Umsetzung der Funktionalitäten mit Hilfe dieser Basisinteraktionen sowie eine Beschreibung des Systemverhaltens der Beispielanwendung.

3.1 Gestaltung interaktiver Systeme

Ein gutes interaktives System zeichnet sich u.a. durch *leichte Erlernbarkeit*, *schnelle Bedienbarkeit*, *geringe Fehlerrate*, *einfache Erinnerung* und durch ein *ansprechendes visuelles Design* aus (Foley 91). Klarheit, Konsequenz und attraktive Erscheinung beeinflussen sowohl den ersten Benutzereindruck, als auch die längerfristige Verwendbarkeit. Bei der Gestaltung einer Benutzerschnittstelle kann man zwischen seiner *Funktion*, d.h.

das prinzipielle Konzept der Anwendung sowie seiner detaillierten Funktionalität, und der *Form*, dazu gehören elementare Interaktionsformen, Dialoge und Systemverhalten, unterscheiden.

3.1.1 Elementare Interaktionsformen

Als elementare Interaktion wird der Prozeß bezeichnet, bei dem der Benutzer eine Informationseinheit eingibt, die eine Bedeutung im Sinne des interaktiven Systems besitzt. Dazu zählen die Angabe einer Position durch Lokalisierungsmedien wie Maus, Joystick oder Trackball, die Selektion von Objekten, die Auswahl eines Menüeintrages, die Eingabe von Text und die Angabe einer bestimmten Größe.

Lokalisierungsmedien. Unter den Lokalisierungsmedien ist die Maus am weitesten verbreitet. Sie wird verwendet für die Auswahl von Menüeinträgen, zum Bewegen innerhalb einer Tabelle, zum Öffnen von Fenstern oder Starten von Anwendungen über eine Selektion der zugehörigen Pictogramme. Zur Steuerbarkeit der Maus bedarf es einer Aktualisierungsrate, die nicht unter 0,2 Sekunden liegen sollte, da es zu einer ruckartigen Visualisierung auf dem Bildschirm käme, wenn die Stellung zu selten erfaßt würde. Die visuelle Rückmeldung sollte für den Benutzer unmittelbar erfolgen, da er sonst keine unmittelbare Kontrolle über das Interaktionsmedium erfährt und damit auch nicht über die Anwendung. Diese Echtzeitverarbeitung ist ein allgemeiner Gestaltungsaspekt von Multimediasystemen und bedeutet, daß das System so rasch reagieren sollte, daß weder die Darstellung von Information noch die Verarbeitung von Vorgängen der beobachtbaren Realität verzerrt werden (Stry 96). In einigen Fällen, etwa bei Kiosksystemen, werden Touchscreens als Lokalisierungsmedien verwendet, mit dem Nachteil, daß auch Dateneingaben hierüber erfolgen müssen, wodurch einerseits die Eingabe umständlich wird und andererseits ein Teil des Bildschirms nicht mehr für die Visualisierung zur Verfügung steht.

Selektion. Die Selektion von Objekten oder Dateneinheiten ist durch Angabe eines Namens, durch Zeigen, durch Auswahl eines Feldes einer Liste oder auch durch eine spezielle, mit dem Lokalisierungsmedium durchgeführte, Bewegung möglich. Bei der Selektion ist eine visuelle Rückmeldung, die ihren Status dem Benutzer anzeigt, entscheidend. Werden anschließend weitere Manipulationen gewünscht, wie etwa beim Selektieren und anschließendem Skalieren von Objekten, hat sich das Konzept der „Handles“ bewährt, die einen eindeutigen visuellen Code für die Selektion eines Objekts darstellen und gleichzeitig, z.B. beim Skalieren, eine Manipulation ermöglichen, ohne daß der Benutzer darüber nachdenken muß, wo sich das Zentrum befindet (Foley 91). Im Falle einer 3D-Interaktion gestaltet sich ein Positionieren und Selektieren deutlich komplizierter. Bei 3D-Animationsprogrammen wird versucht, dieses Problem durch die Verwendung von mehreren orthographischen Ansichten der 3D-Szene zu lösen.

Menüauswahl. Menüauswahl ist eine der mächtigsten Techniken für die Selektion aus einer relativ begrenzten Auswahl. Menüs zeigen den Funktionsumfang einer Anwendung auf einen Blick, erfordern dabei wenig Eingabe über die Tastatur und sind robust gegenüber Fehlbedienungen (Stry 96). Aufgrund dieser Faktoren und einer einfachen Erlernbarkeit sind die meisten bekannten interaktiven Systeme Menüsysteme. Bei komplexen Anwendungen kann jedoch auch die Struktur der Menüs sehr komplex werden und dadurch der Weg zur eigentlichen Operation sich sehr zeitaufwendig gestalten. So bedarf es bei vielen 3D-Animationsprogrammen aufgrund der Vielfalt der Kommandos, die über eine Menüauswahl aufgerufen werden können, einer grundlegenden und längeren Einarbeitungsphase durch den Benutzer, in der das Erlernen des Umgangs mit dem System und nicht die Lösung der Aufgabe im Vordergrund steht. Lösungsmöglichkeiten sind unter

Anderem eine ständige Anzeige der Menüs oder das Auflösen hierarchischer Strukturen. Eine weitere Möglichkeit Menüs darzustellen, ist die Verwendung eines zweiten Bildschirms, eines Hilfsbildschirms, mit dem Nachteil, daß vom Benutzer verlangt wird, den Blick von der Darstellung der Anwendung abzuwenden, und dadurch die visuelle Kontinuität der Anwendung zu zerstören.

3.1.2 Dialoge

In der Regel ist die Form einer Benutzerschnittstelle durch die Verwendung einer bestimmten Dialogart vorgegeben. Zu den grafisch-orientierten zählen hierbei das *WYSIWYG-Prinzip*, *direkte Manipulation* und *ikonische* oder *Pictogramm-basierte Dialogarten*. Nicht ausschließlich grafisch orientiert sind *Menüsysteme*, *Kommandosprachen*, *natürlichsprachliche Dialoge* sowie sogenannte *Frage-und-Antwort-Dialoge*. Jede dieser Varianten bietet sowohl Vor- als auch Nachteile. So reduzieren Menüsysteme die Anforderungen an das Erinnerungsvermögen des Benutzers, wie es beispielsweise bei Kommandosprachen gefordert ist, erlauben jedoch stets nur eine begrenzte Anzahl von darstellbaren Alternativen. Häufig ist die Kombination von zwei oder mehrerer Interaktionstechniken besser geeignet als eine strikt durchgängige Verwendung einer Technik allein.

Menüsysteme. In Stry (96) wird, basierend auf empirischen Studien aus dem Jahre 1985, argumentiert, daß Menüsysteme von unerfahrenen Benutzern schwer erlernbar sind und zum Teil die Eingabe von Kommandos erfolgreicher ist. Demgegenüber steht, daß durch die inzwischen weite Verbreitung von Computern mit grafischen Benutzerschnittstellen, viele Menschen schon auf die Verwendung eines Menüsystems über Maus und Tastatur sozialisiert sind. Als Alternative zu den textuellen Menüeinträgen können Pictogramme verwendet werden, bei denen die Reaktion auf Information rascher als bei textueller Information erfolgt. Zusätzlich nehmen Pictogramme unerfahrenen Benutzern die Scheu den Umgang mit komplexen Schnittstellen zu erlernen, eine Assoziation mit Bekanntem ist leichter möglich als bei der Auswahlmöglichkeit textueller Befehle. Unerfahrene Benutzer besitzen oft eine Schwelle vor unbekannten Systemen, da sie „nichts falsch machen wollen“. Die Verwendung von Pictogrammen als Ersatz für textuelle Menüeinträge erweist sich jedoch nicht als so erfolgreich, wie es intuitiv scheinen mag. Daher sollten Pictogramme stets zusammen mit Text verwendet werden (Stry 96). Weiterhin kann die Verwendung von Pictogrammen auch kontraproduktiv sein, falls diese nicht adäquat abstrahiert oder ausreichend detailliert sind.

Direkte Manipulation. Direkte Manipulation ist ein Konzept, bei dem ein Objekt vor der Ausführung einer Operation selektiert wird. Es orientiert sich in erster Linie an den Datenobjekten und in zweiter Hinsicht an den Funktionen. Das bekannteste Beispiel ist das Löschen einer Datei durch ein Bewegen des Dateipictogramms auf den Papierkorb der Macintosh- bzw. Windows-Benutzeroberfläche. Der Lernaufwand zur Bedienung eines Systems wird dadurch reduziert und die Geschwindigkeit erhöht. Zusätzlich erlaubt direkte Manipulation ein rasches visuelles Feedback für den Benutzer, jedoch ist diese Technik auf sichtbare Objekte beschränkt und daher nicht immer anwendbar. Wenn das betreffende Objekt nicht sichtbar ist, ist es oft effektiver den Namen einzutippen anstatt seine Anzeige zu selektieren. Es gibt auch Fälle, bei denen direkte Manipulation langsamer ist als die Verwendung von Kommandos. Direkte Manipulation zur Dialoggestaltung eignet sich besonders gut, wenn Benutzer gewöhnt sind, das Objekt auch physisch zu manipulieren.

Multimodale Dialoge. Multimodale Dialoge sind Kombinationen mehrerer elementarer Interaktionsformen (z.B. Menüs und Sprache), die es dem Benutzer ermöglichen, eine Anforderung an das System über verschiedene parallele Kanäle zu stellen, beispielsweise

die Verbindung einer Menüauswahl mit Sprache oder die Mischung von Zeigegesten mit Sprachkommandos. So kann eine Zeigegestik zusammen mit einem Sprachkommando u.U. eine lange sprachliche Anweisung ersetzen. Multimodale Dialoge erleichtern die Interaktion, indem dem Benutzer ermöglicht wird, eine Aufgabe auf eine natürlichere Art und Weise und mit weniger Aufwand auszuführen, als es bei der Verwendung nur eines Kommunikationskanals nötig wäre. Schlußfolgerungen aus empirischen Untersuchungen zeigen, daß die Verwendung von Sprache eine Option sein sollte, d.h. daß es stets möglich sein sollte, über andere Interaktionsformen zu derselben Wirkung zu gelangen (Stary 96). Weiterhin empfiehlt sich, für Sprachkommandos ein kleines eingeschränktes Vokabular zu verwenden, zum Einen aufgrund der bis heute bestehenden Schwierigkeiten bei der automatischen Spracherkennung, zum Anderen damit der Benutzer nicht damit belastet wird, eine Vielzahl von Kommandos lernen und erinnern zu müssen. Multimodale Dialoge sind nicht auf verschiedene Eingabemodalitäten beschränkt. Ebenso kann Information auch über verschiedene Ausgabemodalitäten angeboten werden, um eine Information auf die für den Benutzer und die Situation adäquateste Art und Weise darzustellen.

3.1.3 Systemverhalten

Das Systemverhalten ist entscheidend für die Akzeptanz einer Benutzerschnittstelle. Die nach Foley (91) wichtigsten Punkte hierbei sind: die *Durchgängigkeit der Bedienung*, die *Rückmeldungen des Systems* (Feedback), die *Minimierung von Fehlermöglichkeiten*, die *Möglichkeit der Fehlerbehebung*, die *Unterstützung verschiedener Bedienungsfertigkeiten* und die *Minimierung der Erinnerung* durch den Benutzer. So werden durch die Verwendung von Namen statt Nummern für Dateien gleichermaßen Fehlermöglichkeiten verringert sowie ein leichtes Wiederauffinden durch den Benutzer ermöglicht, d.h. der Benutzer braucht sich nicht abstrakte Zahlen zu merken.

Rückmeldungen des Systems sind sehr wichtig während und nach einem Bedienablauf: Wer schon einmal versucht hat sich mit jemandem zu unterhalten, der dabei keinerlei Regung zeigt, weder körperlicher Art, noch mimischer oder artikulatorischer Art, der versteht, welche wichtige Rolle eine Rückmeldung als Teil der Kommunikation zukommt. Ähnliches gilt für Mensch-Maschine-Kommunikation, jedoch mit dem Unterschied, daß der Computer wenig automatische Rückmeldungen an den Benutzer geben kann und es daher die Aufgabe des Anwendungsprogramms ist, dieses zu gewährleisten. Ort und Zeitpunkt von Rückmeldungen sind entscheidend. Dabei scheint es die natürliche Tendenz zu geben, einen festen Bildausschnitt der Benutzerschnittstelle für Rückmeldungen zu reservieren, was jedoch zur Beeinträchtigung der visuellen Kontinuität führt, da der Benutzer seinen Blick zwischen Arbeitsbereich und Meldungsbereich wechseln muß und deshalb oft Meldungen übersehen werden (Foley 91). Wird Audio-Feedback anstelle eines Meldungsfensters oder anderen grafischen Feedbacks verwendet, besteht dieses Problem nicht. Bei Rückmeldungen ist außerdem zwischen dem Anwendungsbereich und dem Kontrollbereich zu unterscheiden, d.h. ob die Rückmeldung sich auf die Erledigung der Aufgabe bezieht oder auf die Bedienung des Systems.

Bedienebenen. Hilfreich für die Verwendung eines Systems ist die Möglichkeit verschiedene Entwicklungsstufen der Bedienung (z.B. Anfänger oder Experte) zu unterstützen. Anfänger werden durch einen systemgesteuerten Frage-und-Antwort-Dialog leicht in die Bedienung eines Systems eingeführt, dagegen bevorzugen es versierte Benutzer, ganze Arbeitsvorgänge mit einem Tastendruck zu erledigen.

Zustandsdiagramme. Zur Überprüfung der Konsistenz einer Benutzerschnittstelle empfiehlt sich die Darstellung der Interaktionsmöglichkeiten in einem *Zustandsdiagramm*,

durch *rekursive Transitionsnetzwerke* oder sogenannte *Ereignissprachen*. Zustandsdiagramme sind eine bewährte Technik zur Darstellung des Systemverhaltens von interaktiven Systemen (Stary 96). Verschiedenen Zustände und die Zustandsübergänge lassen sich grafisch darstellen. Zusätzlich können Inkonsistenzen in der Bedienung durch solche Diagramme erkannt werden, die beispielsweise auftreten, wenn zwei vom Typ her ähnliche Operationen auf verschiedene Weise aufgerufen werden müssen. Solche Inkonsistenzen führen meist dazu, den Benutzer zu verwirren, da das System in bestimmten Situation nicht so reagiert wie erwartet, bzw. nicht so reagiert, wie es der Benutzer an anderer Stelle kennengelernt hat. Durch derartige Formalismen können in der Entwicklungsphase einer Schnittstelle die Interaktionsmöglichkeiten definiert oder die Bedienungsabläufe fertiggestellter Systeme beschrieben und überprüft werden. Eine andere Möglichkeit der Definition verschiedener Bedienungsabläufe bietet die Demonstration eines Beispiels und dem Lernen des Systems.

3.2 Einordnung der Intuitiven Schnittstelle

Das Szenario der Intuitiven Schnittstelle plziert den Benutzer in die unbeschwerliche Umgebung eines Realraumes vor einer Projektionswand. Die Gestikinteraktion in der speziellen Situation beim Arbeiten vor einer Rückprojektionswand und unter Verwendung von Raum und Körperbewegung stellt bestimmte Anforderung an die Software-schnittstelle eines Systems. Soll das System dazu noch intuitiv bedienbar sein, und soll es nicht notwendig sein, seine Interaktionsmuster zu erlernen, sind implizit weitere Anforderungen definiert. Interaktionsmuster, die sich bei menübasierten Systemen bewährt haben, müssen nicht notwendigerweise in dieser Umgebung funktionieren, d.h. sie können nicht einfach übernommen werden. Ausgehend von den notwendigen Funktionalitäten der Schnittstelle wird auf die Besonderheiten des Szenarios eingegangen, die bei einer Umsetzung zu beachten sind.

3.2.1 Geforderte Funktionalitäten

Die für die Intuitive Schnittstelle geforderten Funktionalitäten werden hier auf die notwendigen Interaktionsformen der Beispielanwendung beschränkt.

Lokalisierung	Zur Auswahl von Objekten oder Alternativen muß der Benutzer die Möglichkeit haben, seine Auswahl dem System anzuzeigen.
Selektion	Nach erfolgter Lokalisierung sollen Objekte der Szene selektiert werden können, um nachfolgende Manipulationen zu ermöglichen.
Manipulation	Zum Arrangieren der Szene sind verschiedene Manipulationsmöglichkeiten zur Verfügung zu stellen, um Objekte beispielsweise zu bewegen, zu rotieren oder auch zu entfernen.
Menüauswahl	Um die Szene mit Objekten zu ergänzen, muß eine Art Menüauswahl zur Verfügung gestellt werden, die es dem Benutzer gestattet, aus einer Anzahl von Objekten ein gewünschtes auszuwählen.

3.2.2 Besonderheiten des Szenarios

Bei traditionellen Desktop-Systemen geschieht die Lokalisierung eines Punktes auf dem Bildschirm entweder durch das Bewegen des Cursors an die gewünschte Position und Drücken eines Knopfes, durch die Eingabe eines Wertes über die Tastatur oder durch ein

Sprachkommando, sowie kombinierte Techniken z.B. mit Maus und Tastatur. Hiermit vergleichbar ist die Verwendung von Zeigegestiken des Benutzers im Raum für die Lokalisierung auf der Projektionswand, beispielsweise durch die Steuerung eines Cursors. So ergibt sich zusätzlich zu der bekannten Methode einer Positionsanwahl mittels einer Cursorsteuerung eine weitere Dimension: Die Position des Benutzers im Raum. Diese Situation kann so interpretiert werden, daß der Benutzer im Raum entweder als ein weiteres 3D-Eingabegerät, neben 3D-Maus, 3D-Joystick, Trackball, Datenhandschuh etc. betrachtet wird, oder als ein völlig neues Eingabemedium. Im folgenden wird die zweite Alternative vorgezogen, d.h. die Körperbewegungen des Benutzers werden hier nicht als ein Eingabegerät für den Computer gesehen, sondern als etwas, das einer speziellen Aufmerksamkeit bedarf.

Im Gegensatz zu einem Eingabegerät ergibt sich aus der Position des Benutzers zu seiner Umwelt, dem Realraum, eine neue Ebene, die in die Schnittstelle eingeht als eigenständiger Input neben einer Lokalisierung, einer Texteingabe, einer Auswahl, einer Mengenangabe oder numerischen Wertangabe. Aus dieser Betrachtungsweise ergeben sich für die Schnittstelle zusätzliche elementare Interaktionsformen: Die Bewegung im Raum, die Verwendung sensibler Bereiche und ein, die Körperbewegung einbeziehender, modifizierter Umgang mit Menüs. Diese Interaktionsformen unterscheiden sich von ähnlich erscheinenden Beispielen der Interaktionen in virtuellen Welten, da hier auch die psychoanalytische Beziehung zwischen Realumgebung und dem Benutzer als Subjekt mit seiner ganzen Körperempfindung einfließt (vgl. Kapitel 2, Abschnitt *Körper und Raum*). Eine weitere Interaktionsform ist durch die Verwendung von Realobjekten zur Steuerung von virtuellen Objekten gegeben, die eine Art der direkten Manipulation darstellt und hier als *reale Manipulation* bezeichnet wird.

3.3 Basisinteraktionen

Die Basisinteraktionen der Intuitiven Schnittstelle sind die Zeigeoperation, Sprachkommandos, die Interaktion im Raum und die Verwendung von Realobjekten.

3.3.1 Zeigegestik

Eine Zeigegestik - ausgeführt vor einer Projektionswand - unterscheidet sich in vielen Dingen von einer Lokalisierung mit der Maus bei Desktop-Systemen. Mit Verwendung der Maus liegt die erreichte Genauigkeit innerhalb weniger Pixel, da die visuelle Rückmeldung von grafischen Benutzerschnittstellen genügend Information für eine Hand-Auge-Koordination liefert. Dagegen ist das Zeigen auf Objekte, die auf der Projektionswand dargestellt werden, eine Aufgabe, die eher einer Zeigegestik bei Mensch-Mensch-Kommunikation ähnlich ist. Während einer Unterhaltung verwenden Menschen häufig richtungsweisende Gesten, wie das gelegentliche Zeigen mit dem Zeigefinger, parallel zu einer sprachlichen Äußerung, um den Bezug zu Objekten zu verdeutlichen. Im Gegensatz zu der üblichen Bedeutung von Mausklicks bei Umgebungen mit direkter Manipulation, ist bei Konversationen die Region auf die der Mensch gerade zeigt nicht notwendigerweise identisch mit der Region auf die er sich bezieht (Sullivan 91). Die natürliche Zeigegestik ist oft mehrdeutig und vage. Daher kann eine Desktop-Anwendung nicht direkt in die Umgebung der Intuitiven Schnittstelle übersetzt werden. Interaktionsmuster, die sich für die Desktop-Metapher bewährt haben, müssen nicht unbedingt funktionieren, wenn Gesten und eine Projektionswand verwendet werden.

Sollen Objekte durch eine Zeigegestik selektiert werden, entscheiden die Art des Menüs, die Größe und die Form der Menüpunkte über Geschwindigkeit und Benutzerfreundlichkeit der Anwendung. Dabei können ikonische Menüs räumlich flexibler angeordnet

werden als textuelle Menüs, da bildliche Einträge nicht lang und schmal wie Textfelder sein müssen. Zeigegenauigkeit und Geschwindigkeit der Bedienung werden von der Größe der einzelnen Menüpunkte beeinflusst: Größere Einträge sind schnell anzuwählen, kleine Einträge benötigen weniger Platz und erlauben mehr Punkte darzustellen. Es gibt also einen Konflikt zwischen der Verwendung kleiner Menüpunkte, um Platz zu sparen und mehr Einträge auf einem begrenztem Platz darstellen zu können, und der Herabsetzung von Selektionszeit und Fehlerwahrscheinlichkeit. Um die Zeigegestik zu testen, wurde eine Beispielanwendung entwickelt, mit der die erzielbare Genauigkeit ermittelt wurde. In dieser Anwendung wird auf der Projektionswand eine Netscape-Anwendung dargestellt (siehe Kapitel 2, Seite 25). Mausbewegungen werden durch die Bestimmung der Zeigerichtung des Benutzers und dem Setzen der Mausposition auf die so ermittelte Position simuliert. Mausklicks werden durch ein Sprachkommando ausgelöst. Der Anwender agiert typischerweise in einer Entfernung zwischen drei und vier Metern vor der Projektionswand mit den Abmessungen 2,10 x 2,60 Meter (vgl. Abbildung 6-2, Seite 115). Das Selektieren von Internet-Verweisen in einem Netscape-Fenster, die mit einer Höhe von ca. 5 cm dargestellt wurden, gestaltete sich als schwierig und führte zu mehreren fehlerhaften Anläufen, begründet durch ein „Zittern“ des Systems in etwa in der gleichen Größenordnung. Andererseits war das Zeigen und Selektieren von Bildern oder Pictogrammen, die in einer Größe von ca. 20 qcm dargestellt wurden, einfach und sicher zu bedienen. Dadurch wird auch offenbar, daß ein solches Auswählen nicht für diese Art der Interaktion geeignet ist, da eine zu präzise Positionierung des durch die Zeigegestik gesteuerten Mauszeigers notwendig ist. Zeigegestiken sind auch im täglichen Leben eher richtungsweisend als präzise, und so muß die Intuitive Schnittstelle diese Ungenauigkeit berücksichtigen.

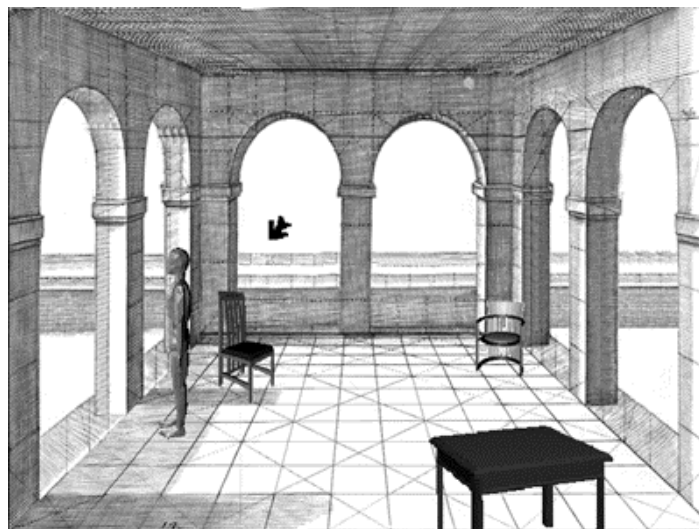


Abbildung 3-1: Typische Szene der Beispielanwendung zur Filmplanung mit mehreren Objekten und Darstellung des Cursors.

Aufgrund dieser Erfahrung wurde für die Entwicklung des Prototyps zur Filmplanung (siehe Kapitel 6) die Zeigegestik zur Steuerung eines Cursors verwendet, der deutlich größer als sonst üblich dargestellt wurde (Abbildung 3-1). Auf eine Anwahl von kleinen Objekten oder Menüeinträgen wurde verzichtet. Die Zeigegestik wird verwendet, um Objekte der dargestellten 3D-Szene zu selektieren und anschließend zu verschieben. Alternativ kann das Verschieben mit Hilfe von Realobjekten im Raum durchgeführt werden.

3.3.2 Sprachkommandos

Für die Spracherkennung wird ein kommerzielles, PC-basiertes System (Lernout & Hauspie 96) verwendet, um einfache Kommandos in Form von Einzelwörtern eingeben zu können. Das System erlaubt die sprecherunabhängige Erkennung kontinuierlich gesprochener Sprache, wobei die zu erkennenden Sätze oder Befehle in einer standardisierten Form (Backus-Naur-Grammatik) festgelegt werden. Dadurch erkennt das System nur die festgelegten Wortfolgen, und der Benutzer kann sich auch bei aktiver Spracherkennung frei unterhalten. Beispiele der hier zum Einsatz kommenden Kommandos sind „select“ zum Selektieren, „rotate“ zum Rotieren und „delete“ zum Löschen von Objekten.

3.3.3 Menüauswahl

Menüauswahl ist eine der Hauptinteraktionsformen bei der Bedienung von Anwendungen über ein WIMP-Interface⁵. Diese Interaktionsform kann sicherlich nicht direkt in eine Computerumgebung mit Projektionswand übertragen werden, nicht nur, weil die Menüeinträge schwer lesbar wären, sondern auch aufgrund der Bemerkungen im Abschnitt *Besonderheiten des Szenarios*. Zur Lösung des Problems, wurden große sensitive Flächen an den Seiten der Projektionswand untersucht, die ein Menü zur Anzeige bringen, wenn der Benutzer auf die entsprechende Stelle zeigt (Abbildung 3-2). Diese Art von Menü wird auch *pull-out-Menü* genannt. Beim Test verschiedener Menügrößen zeigte sich, daß ein Menü eine bestimmte minimale Größe besitzen muß, damit es in dem Szenario verwendbar ist, d.h. es muß - je nach Anzahl der Einträge - ca. ein Viertel des Bildes überdecken. Eine maximale Größe darf jedoch auch nicht überschritten werden: In dem in Abbildung 3-3a dargestellten Beispiel beträgt die Größe des Menüs mehr als die Hälfte des projizierten Bildes, wodurch ein Kontextwechsel für den Benutzer hervorgerufen wird, da die Szene verdeckt wird. Die für den Benutzer wichtige visuelle Kontinuität der Schnittstelle wird dadurch durchbrochen.

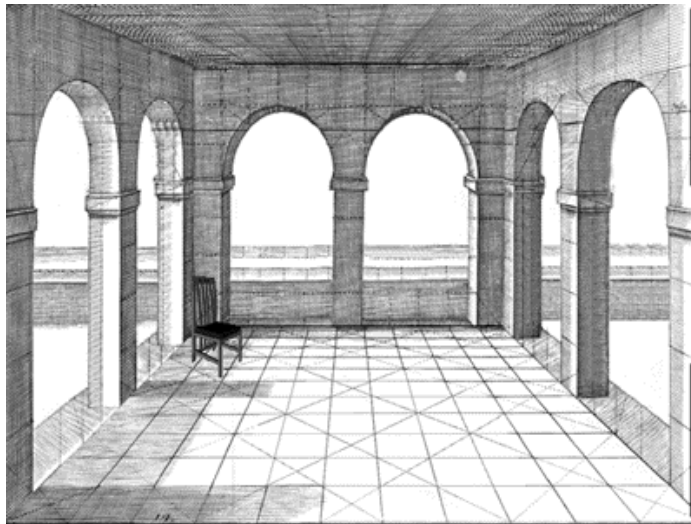


Abbildung 3-2: Szene mit sensitiven Feldern, markiert durch schwarze Balken am rechten Rand

Bei Desktop-Systemen erscheint ein Menü normalerweise an der Cursorposition, also dort wo sich meist die Aufmerksamkeit des Benutzers befindet. Das Einblenden von *Pop-*

⁵ WIMP steht für *Window, Icon, Menu, Pointing device*.

up- oder *Pulldown*-Menüs hat sich dort bewährt, da so die visuelle Kontinuität eher erhalten bleibt, als daß ein ausgelöster Kontextwechsel sich negativ auswirkt. Für die Projektionswand wird dagegen eher das Gegenteil erreicht, da Menüs hier den Kontext verdecken, der dem Benutzer unter Umständen helfen würde, eine Auswahl zu treffen. Ein Menü nimmt hier zusätzlich einen großen physikalischen Raum und dadurch auch Platz des Interaktionsraums des Benutzers in Anspruch. Die Anzeige eines Menüs wirkt daher nicht angenehm. Abbildung 3-3b zeigt den besten Kompromiß zwischen einer minimalen Größe zur Selektion der einzelnen Menüpunkte und dem Sichtbarbleiben der Szene. Wird das Menü weiter verkleinert, so wird eine Auswahl erschwert und das Menü ist nicht mehr deutlich erkennbar.

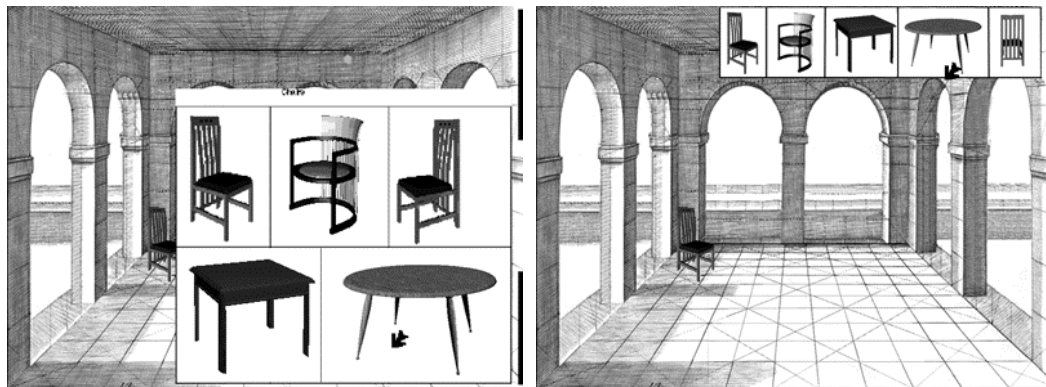


Abbildung 3-3: Darstellung eines Menüs mit Stühlen und Tischen durch Zeigen auf den unteren rechten Rand der Projektion (a), bzw. auf den oberen (b)

Ein weiteres Problem der Menüs ist die Skalierbarkeit: Stehen viele Objekte zur Auswahl, können diese nicht mehr in einem Menü untergebracht werden, ohne daß doch die Szene verdeckt werden würde. Auch können bei komplexen Anwendungen ebenso komplexe Menüs entstehen, die oft zu Navigationsproblemen führen, bevor der gewünschte Eintrag gefunden wird. Bei der Arbeit vor einer Projektionswand ist dies auf Grund der weniger präzisen Interaktion noch eher zu erwarten, da die Handhabung von verschachtelten Menüs hier sich noch schwieriger gestalten würde. Zusätzlich würden derartige Menüs erneut die visuelle Kontinuität beeinträchtigen. Aus diesen Gründen erscheinen derartige Menüs in der Computerumgebung der intuitiven Schnittstelle nicht anwendbar. Eine Lösung des Problems wurde durch die im nächsten Abschnitt beschriebene Einbeziehung des Realraums und der Integration einer ikonischen Auswahl in die Szene erreicht.

3.3.4 Rauminteraktion

Im letzten Abschnitt wurde das Problem des Kontextwechsels der Verwendung von Menüs für den Auswahlprozeß erläutert. Ein Lösungsansatz bietet die Einbeziehung des Realraums der intuitiven Schnittstelle in die Mensch-Maschine-Interaktion. Hierbei werden Kontextwechsel durch Raumpositionen ausgelöst und die räumliche Erinnerung des Benutzers für das Aufrufen von Objekten verwendet. Besonders für eine Arbeitssituation mit mehreren Personen vor der Projektionswand ergeben sich Vorteile, da die Betrachter der Interaktion durch die aktive Bewegung des Benutzers im Raum ein Feedback bekommen. Diese Situation ist vergleichbar mit einer Präsentation, bei der vom Vortragenden Körperbewegungen zur Verdeutlichung eingesetzt werden, und der Inhalt so leichter rezipierbar wird. Nachteile ergeben sich unter Umständen aus einer verringerten Bedienungsgeschwindigkeit im Vergleich zu Menüsystemen und der Notwendigkeit aktiv zu

gehen, so daß diese Interaktionsformen nicht für jede Anwendung gleich gut geeignet sind. Für die intuitive Schnittstelle unterscheiden wir zwei Interaktionsformen:

Intuitiver Kontextwechsel. Hierunter ist die Änderung des Anwendungsmodus durch eine aktive Leibbewegung des Benutzers zu verstehen, bei dem z.B. das Gehen nach hinten, also den hinteren Teil des Raumes, eine Funktion aufruft. In der Beispielanwendung zur Filmplanung wird hierdurch die Requisite zum Hinzufügen von Objekten aufgerufen. Durch diese Art des Kontextwechsels bleibt dem Benutzer die aktuelle Umgebung bzw. Szenerie bewußt, d.h. er verliert nicht den „Kontakt“ zu ihr. Durch die aktive Leibbewegung *fühlt* der Benutzer außerdem, daß er den Kontextwechsel hervorruft. Dadurch bleibt auch eher die visuelle Kontinuität gewahrt als durch das Einblenden von (notwendigerweise) großflächigen Menüs, wie sie im letzten Abschnitt erläutert wurden.

Bei der Gestaltung von Benutzerschnittstellen ist die Minimierung von Fehlermöglichkeiten durch den Benutzer ein wichtiger Aspekt. Die Notwendigkeit der aktiven Bewegung - zum Aufrufen der Requisite beispielsweise - bietet den deutlichen Vorteil, daß hier Fehlbedienungen selten möglich sind. Alternativen, die verschiedene Modi mit einem Gestikkommando aufzurufen, könnten aufgrund der Eigenbewegung des Benutzers leicht mißverstanden werden. So ist es ein schwieriges Problem, automatisch zu unterscheiden, wann eine Handbewegung eine Intention besitzt und wann es sich nur um eine eher zufällige Gestik handelt. Auch Sprachkommandos könnten, wegen der noch immer nicht perfekten Technik auf dem Gebiet der Spracherkennung, evtl. falsch verstanden werden. Die Bedienbarkeit einer Anwendung und eine Benutzerakzeptanz würden sinken, wenn Kommandos oft wiederholt oder aufgrund falscher Interpretation rückgängig gemacht werden müßten. Zusätzlich ist die Erinnerung vieler Kommandos ein Problem, welches die Interaktion erschweren kann.

Gedächtnisfunktion. Die Gedächtnisfunktion stellt eine Lösungsmöglichkeit dar, um auf traditionelle Menüs zu verzichten. Sie verwendet den Realraum als Positionen mit inhärentem Kontext. Dazu werden bestimmte Bereiche des Realraums sensitiven Zonen zugewiesen und dadurch als Selektionsbereiche definiert. Ein interner Kontextwechsel vollzieht sich in dem Moment, indem der Benutzer sich in diesen Bereich bewegt. Ähnlich dem in Kapitel 2, Abschnitt *Anwendungen* beschriebenen Aufrufen von Datenlinks im Realraum, kann der Benutzer je nach Raumposition verschiedene Objektgruppen anwählen. Diese auf dem Prinzip der „Ars Memoriae“ aufgebaute Funktion entspricht dem Aufrufen eines Kommandos, jedoch mit einem anderen Gesicht (*look-and-feel*) der Schnittstelle. Erste Untersuchungen mit einer Beispielanwendung haben gezeigt, daß das Ablegen von Daten an Orten im Realraum, intuitiv verwendbar ist und von Benutzern des Systems direkt verwendet werden kann. In dieser Beispielanwendung können Internet-Hyperlinks im Raum aufgerufen werden, indem sich der Benutzer an den Ort begibt und einen Hyperlink durch ein Sprachkommando auswählt. Die so angewählten Seiten erscheinen in einem Netscape-Fenster auf der Projektionswand (vgl. Kapitel 2, Abschnitt *Anwendungen*).

Diese Art der Selektion von Objektgruppen hat auch für anwesende Dritte Vorteile, da sie die Person sich aktiv bewegen sehen und der Effekt schneller Menüwechsel - der je nach Anwendung einem „Menüflimmern“ gleichen kann - sich hier nicht negativ auswirkt. Bei traditionellen Anwendungen genügt oft ein für Dritte unbemerkter Tastendruck, um ein Menü anzuzeigen und so die visuelle Kontinuität für diese Betrachter zu durchbrechen. Nachteile ergeben sich, wie auch beim intuitiven Kontextwechsel, durch die Notwendigkeit sich aktiv bewegen zu müssen und der damit verbundenen geringeren Bediengeschwindigkeit. Die Eignung dieser Interaktionsformen muß daher je nach Anwendung

überprüft werden (diese Aspekte werden bei der Überprüfung des Prototyps in Kapitel 6 aufgegriffen).

3.3.5 Verwendung von Realobjekten

Zusätzlich zur Verfolgung des Benutzers werden Realobjekte, die sich mit virtuellen Objekten verbinden lassen, in die Computerumgebung der Intuitiven Schnittstelle einbezogen. Ist diese Verknüpfung vollzogen, werden die virtuellen Objekte der Szene analog der Modifikation der Objekte im Realraum bewegt. Dadurch ist die Positionierung und das Arrangieren von Objekten und die Diskussion eines virtuellen Sets im Realraum möglich. Die räumlichen Bezüge lassen sich im Realraum besonders gut beurteilen, dagegen gestaltet sich die Positionierung und Beurteilung von 3D-Objekten bzw. Szenen über eine zweidimensionale Bildschirmoberfläche sich als schwierig und kann oft nur durch zusätzliche Interaktionshilfsmittel, wie orthografische Ansichten, erfolgen.

In der Beispielanwendung zur Filmplanung werden Schachfiguren-ähnliche Platzhalterfiguren zur Platzierung von virtuellen Objekten oder Figuren der Szene verwendet (siehe Abbildung 6-8, Seite 120). Besonders in Bezug auf die Minimierung von Fehlermöglichkeiten und die Einrichtung einer Undo-Funktion für den Benutzer sind diese Realobjekte optimal, da Korrekturen intuitiv und direkt durchführbar sind. Weiterhin lassen sich die Realobjekte auch alternierend von mehreren Personen bedienen, eine Option, die in den meisten Desktop-basierten Systemen nur durch eine Übergabe des Interaktionsgerät möglich ist, das zur Folge hat, das der Benutzer aufgrund der Anordnung von Bildschirm und Interaktionsgerät oft auch den Platz mit dem Vorbenutzer tauschen muß. Die zur Positionierung der Platzhalterfiguren notwendige aktive Leibbewegung hat hier durch den Erfahrungsschatz des Menschen ganz klare Vorteile.

3.4 Dialoggestaltung

3.4.1 Umsetzung der Funktionalitäten

Im folgenden wird die Umsetzung der im Abschnitt *Einordnung der Intuitiven Schnittstelle* geforderten Funktionalitäten für die Beispielanwendung zur Filmplanung beschrieben.

Lokalisierung. Bei der Auswahl von einzelnen Objekten wird eine Lokalisierung mit Hilfe der Zeigegestik durchgeführt. Die Auswahl von Objektgruppen der Requisite geschieht mittels der Gedächtnisfunktion, d.h. eine Lokalisierung wird hierzu durch eine Körperbewegung im Raum realisiert.

Selektion. Das Selektieren von Objekten erfolgt über den auf der Projektion dargestellten Cursor. Als Rückmeldung für den Benutzer werden verschiedene Pictogramme angezeigt, je nach dem, ob der Cursor frei ist, sich über einem selektierbaren Objekt befindet, oder ob das Objekt selektiert worden ist. Zusätzlich kann bei erfolgter Anwahl ein Klick-Geräusch ausgegeben werden. Das Selektieren kann prinzipiell auf zwei alternativ zu verwendende Weisen erfolgen:

Sprachkommando Mit Hilfe eines Sprachkommandos („select“) wird das Objekt selektiert, das sich unter dem Cursor befindet. Es bleibt solange selektiert, bis es durch das Sprachkommando „ok“ deselektiert wird.

Auto-Klick Diese Operation selektiert ein Objekt, wenn sich der Cursor länger als ca. eine Sekunde über einem Objekt befindet. Sie basiert auf der Annahme, daß ein Benutzer normalerweise nicht ruhig auf eine Stelle zeigen würde (wenn nicht zum Selektieren von Objekten). Eine Deselektion ist durch ruhiges Zeigen auf eine Zielposition möglich: das Objekt rastet nach etwa einer Sekunde wieder aus. Alternativ kann auch ein Sprachkommando („ok“) verwendet werden.

Manipulation. Eine Möglichkeit der Manipulation ist es, das Selektieren von der eigentlichen Manipulation zu trennen. In manchen Systemen wird daher ein Objekt erst selektiert oder referenziert und anschließend die Operation, wie etwa „bewegen“, genannt. Im Gegensatz dazu wird hier eine Technik verwendet, die beim Selektieren implizit Kommandos ausführt wie etwa das Bewegen des Objekts (*click-and-drag*-Interaktion), oder das Objekt bei Eingabe eines Kommandos implizit selektiert. So kann ein Objekt automatisch selektiert werden, wenn beispielsweise das Kommando „rotate“ eingegeben wurde, um es erst nach einem erneuten Sprachkommando („ok“) wieder zu deselektieren und damit die Operation zu beenden. Für die Beispielanwendung zur Filmplanung wurden einige grundlegende Manipulationen implementiert. Je nach Anwendung können auch andere Formen sinnvoll sein, d.h. daß nicht notwendigerweise das Bewegen durch Zeigen für jedes Objekt anwendbar oder wünschenswert ist. Die Manipulationsmöglichkeiten der Beispielanwendung sind hier daher exemplarisch aufgeführt:

bewegen	Geschieht entweder über direkte Manipulation, d.h. nach Selektion wird das Objekt direkt bewegt, solange es selektiert ist, oder durch reale Manipulation, d.h. die Verwendung von Realobjekten im Raum, wodurch sich auch die Definition von Bewegungsbahnen sehr einfach gestalten würde.
rotieren	Ausgelöst durch ein Sprachkommando wird durch eine horizontale Handbewegung des Benutzers das selektierte Objekt um die Y-Achse rotiert. Dies erfolgt analog zu den üblichen Schieberegler in einigen 3D-Anwendungen grafischer Systeme.
entfernen	Das Löschen von Objekten erfolgt über ein Sprachkommando, hierbei muß das Objekt zuvor selektiert sein.
aktivieren	Durch Zeigen auf einen Bildbereich wird eine bestimmte Aktion ausgelöst, z.B. wird in der Requisite der Beispielanwendung eine Gruppe von Objekten durch Zeigen aktiviert.

Menüauswahl. Eine Menüauswahl ist dann notwendig, wenn der Benutzer neue Objekte der Szene hinzufügen möchte. Dazu wird zunächst der intuitive Kontextwechsel verwendet, d.h. eine Bewegung im Raum nach hinten, um die Requisite aufzurufen. Die Szene fährt zurück und eine Art Bühne mit Auswahlfeldern für verschiedene Objektgruppen wird sichtbar. Die einzelnen Objektgruppen werden mittels der Gedächtnisfunktion aufgerufen und mit einer Zeigegestik aktiviert, d.h. je nach Raumposition des Benutzers innerhalb einer sensitiven Zone werden andere Objektgruppen als Pictogramme dargestellt. Zeigt der Benutzer auf ein Pictogramm erscheint die entsprechende Gruppe von Objekten auf der Bühne. Anschließend können einzelne Objekte der Gruppe auf der Bühne mit einer Zeigegestik selektiert werden. Die eigentliche Menüauswahl erfolgt also durch Aktivieren der gewünschten Objektgruppe und anschließender Selektion eines Objekts.

3.4.2 Ergonomie

Dialoge der Intuitiven Schnittstelle können auf verschiedene Weisen und über mehrere Kommunikationskanäle erfolgen. Auch ist nicht eine bestimmte Dialogart vorgeschrieben, so können Objekte durch ein Sprachkommando oder eine Auto-Klick-Operation selektiert werden bzw. durch direkte Manipulation auf der Projektionswand bewegt oder durch die Manipulation von realen Objekten im Raum plazierte werden. Systemrückmeldungen werden stets auf die für die Situation am besten geeignete Art ausgegeben. So wird bei der Beispielanwendung ein mechanisches Geräusch als Audio-Feedback beim Wechsel zwischen Szene und Requisite ausgegeben. Bei der Auswahl eines Objektes der Requisite wird das Objekt über eine Animationssequenz auf der Bühne verkleinert und anschließend in der Szene vergrößert. Für die prototypische Beispielanwendung wurden exemplarisch eine kleine Anzahl von Manipulationsmöglichkeiten implementiert. Dementsprechend einfach gestalten sich die Dialoge. Die Manipulationsmöglichkeiten können jedoch als Basis für weitere Anwendungen verwendet werden. Zusätzlich lassen sich weitere Formen, die eine Interaktion vereinfachen oder in bestimmten Situationen beschleunigen, integrieren.

Folgende Grundmuster der Dialoggestaltung lassen sich benennen:

- In Bezug auf die Projektionswand handelt es sich um eine grafisch-orientierte Dialogform unter Verwendung von ikonischen Menüs.
- Es gibt multimodale Dialoge mit Einbeziehung von Sprache und Zeigegestiken, die Sprachkommandos sind auf ein kleines Vokabular beschränkt.
- Objekte der computergenerierten Szene können durch direkte Manipulation auf der Projektion und durch reale Manipulation im Raum verändert werden.
- Durch die Echtzeitverarbeitung gibt das System ein rasches Feedback, bestimmte Aktionen werden durch Audio-Feedback unterstützt.
- Bestimmte Dialoge, wie die reale Manipulation, minimieren Fehlermöglichkeiten und erlauben eine intuitive, direkte Fehlerbehebung.
- Die Verwendung des Realraums verringert die Erinnerungsanforderungen an den Benutzer.

3.4.3 Dialogablauf

Für eine Beschreibung des Dialogablaufs soll hier ein Zustandsdiagramm angegeben werden. Da derartige Diagramme je nach Anwendung verschieden sind, wird hier exemplarisch das Zustandsdiagramm der Beispielanwendung für die Filmplanung angegeben. Bei der Bewertung dieses Diagramms hinsichtlich der Konsistenz ist zu berücksichtigen, daß es sich um eine prototypische Anwendung handelt, die der generellen Überprüfung dieser Schnittstelle dient. Insbesondere sollte mit dieser Anwendung eher die Adäquatheit der Schnittstelle unter Einbeziehung des Realraumes überprüft werden, als ein durchgängiges Bedienkonzept für diese Situation erstellt werden. Ein derartiges Konzept könnte auf Basis der in Kapitel 6 gemachten Untersuchungen realisiert werden. Für die Darstellung wird analog der Klassendarstellung in Kapitel *Systemarchitektur* die UML-Notation verwendet (Fowler 97). Abbildung 3-4 zeigt zwei übergeordnete Zustände, *scene* und *stage*, sogenannte *Superstates*, die erreicht werden, je nachdem ob der Benutzer im vorderen oder im hinteren Teil des Raumes arbeitet. Die einzelnen Zustände enthalten jeweils den Namen und die Aktivität, die in diesem Zustand ausgeführt wird. Zustandsübergänge sind mit Kanten markiert, deren Beschriftung nach der Syntax *Ereignis [Bedingung] / Aktion* erfolgt. Die Bedingung muß beim Eintreten des Ereignisses erfüllt sein, damit der Zustandswechsel und damit die Aktion ausgeführt wird. Hierbei sind die einzelnen Kom-

ponenten optional, d.h. wenn für eine Aktion keine Bedingung notwendig ist, kann sie weggelassen werden.

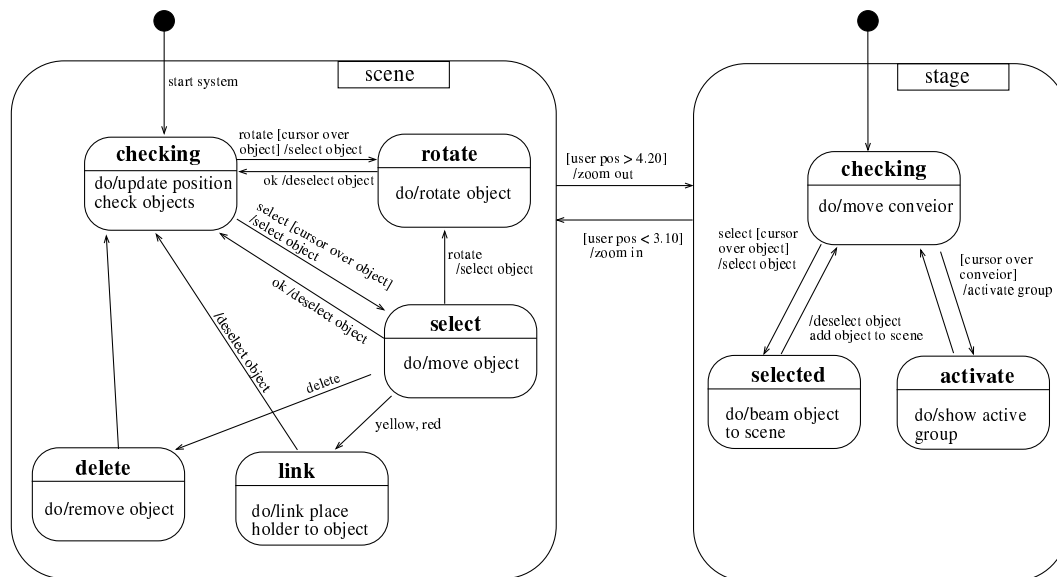


Abbildung 3-4: Zustandsdiagramm der Beispielanwendung für die Filmplanung

Nach Abbildung 3-4 gestaltet sich ein Beispielablauf wie folgt: Nach dem Start wartet das System auf die Eingabe eines Kommandos (Zustand *checking*). Dabei werden die Benutzerposition und die Position des Cursors ständig aktualisiert. Wird ein Selektionskommando aufgerufen, wird das entsprechende Objekt unter dem Cursor selektiert und bewegt. Befindet sich kein Objekt unter dem Cursor verbleibt das System im Zustand *checking*. Wird dagegen im *select*-Zustand das Kommando zum Entfernen von Objekten eingegeben, so wird das selektierte Objekt entfernt (Zustand *delete*) und in den Zustand *checking* zurückgekehrt. Überschreitet der Benutzer die Schwelle von 4,20 Metern im Realraum, wechselt das System vom Superstate *scene* zu *stage*. Hier wird ebenso die Benutzerposition aktualisiert und es werden die Objektgruppen durch die Steuerung eines Fließbandes ausgewählt (Zustand *checking*, *do/move conveyior*). Die Selektion erfolgt analog, jedoch wird das Objekt der Szene hinzugefügt und der Zustand *select* sofort wieder verlassen. Bewegt sich der Benutzer nun wieder nach vorne, d.h. seine Position unterschreitet die Schwelle bei 3,10 Metern, so wechselt das System wieder in den Superstate *scene*. (Für eine genauere Beschreibung der räumlichen Situation und Interaktionsmöglichkeiten siehe die Beschreibung des Prototyps zur Filmplanung in Kapitel 6, Abschnitt 6.2).

4 Personenverfolgung

Der Ansatz der Intuitiven Schnittstelle erlaubt die Verwendung des Realraums zur Speicherung von Information jeglicher Art und ist frei von speziellen Bewegungserkennungsgeräten. Es kommt daher eine videobasierte Personenverfolgungs-Software zum Einsatz, genauer ein Stereosystem, das aus den Kamerabildern die Position und Zeigerichtung einer Person ermittelt. In diesem Kapitel werden neben dem Forschungsstand und der Definition der notwendigen Verfahren zur 3D-Rekonstruktion vor allem die Probleme der Segmentierung und einer stabilen Verfolgung beschrieben. Eine stabile Segmentierung, hier speziell Farbsegmentierung, ist ein nicht ganz triviales Problem, da sich trotz einer Laborumgebung die Beleuchtungsverhältnisse verändern können, wenn z.B. Tageslicht einfällt oder auch nur der Benutzer durch seine Bewegungsfreiheit seinen Standort verändert. Besondere Probleme entstehen, wenn die Person sich dreht oder auch nur eine Hand hinter dem Körper – und damit unsichtbar für die Kameras – positioniert. Die der Segmentierung folgenden Verfahren zur Verfolgung müssen daher neben den nicht perfekten oder auch fehlerhaften Ergebnissen der Segmentierung auch Verdeckungen einbeziehen, um ein stabiles System zu erhalten. In diesem Fall kann ein Modell helfen, indem die Bewegungen von Körper und Händen vorhergesagt werden und beim Verschwinden einer Region die Bewegung extrapoliert, d.h. fortgeführt wird, bis wieder neue Meßdaten vorliegen. Im vorliegenden Szenario ist hierbei nicht so sehr eine absolute Meßgenauigkeit sondern die Erkennung der Absicht des Benutzers und eine Kontinuität in der Verfolgung notwendig. Bei der Auswahl der Verfahren muß ein Kompromiß zwischen der Leistungsfähigkeit und der Echtzeitfähigkeit des Systems gefunden werden, um die in Kapitel 3 beschriebene unmittelbare Rückkoppelung des Systems zu ermöglichen. Insbesondere soll das System auf Standardhardware realisierbar sein.

Im folgenden wird der Forschungsstand im Bereich Personenverfolgung näher erläutert. Anschließend werden die verwendeten Verfahren zur Farbsegmentierung (Schwellwertverfahren und statistisches Verfahren mit Lookup-Tabelle) und ein Verfahren zur Farbkalibrierung im Raum vorgestellt. Nach den Techniken zur 3D-Rekonstruktion und zur Verfolgung werden im Abschnitt *Modellbasierte Verfahren* zwei Verfahren für eine stabile Verfolgung vorgestellt. Einen guten Kompromiß im obigen Sinne stellt insbesondere das *Körpermodell mit Wissensbasis und Aktionen* dar. Die Einbeziehung von Realgegenständen in das System sowie eine Beschreibung der Leistungsdaten schließen das Kapitel ab.

4.1 Forschungsstand

Vorhandene Ansätze der computergestützten, videobasierten Personenverfolgung (zum Teil mit Gestikererkennung) sind oft für spezielle Anwendungsfälle entwickelt worden (Industrie, Medizin, Robotik) und ersetzen in der Regel vorhandene Eingabegeräte durch die Möglichkeit, menschliche Gestik zur Manipulation von Objekten im Rechner zu verwenden. Hierbei steht die Lösung der gewünschten Teilaufgabe, die Manipulation von Objekten, im Vordergrund. So wird zum Beispiel die Hand als 3D-Eingabegerät verwendet, um einen Datenhandschuh zu ersetzen (Maggioni 95) oder es werden mit Hilfe von Handzeichen z.B. Diavorträge gesteuert (Bröckl-Fox 95). Der Vorteil dieser Anwendungen liegt darin, den Benutzer von herkömmlichen Eingabegeräten wie Datenhandschuh oder Fernbedienung zu befreien. Trotzdem mag es sinnvolle Dinge geben, die eine Bedienung erleichtern und dadurch ein „herumtragen“ rechtfertigen. Teilweise kann eine Eingabe von komplexen Gestiken durch das Fehlen eines physikalischen Eingabegeräts noch erschwert werden. Bei der Intuitiven Schnittstelle werden nicht komplexe Gestiken erkannt, die zwar eine hohe Funktionalität bereitstellen, jedoch auch eine große Lernkur-

ve besitzen, sondern es wird der Benutzer nach Möglichkeit ohne Hilfsmittel im ganzen Raum verfolgt.

Zahlreiche Ansätze zur Erkennung und Verfolgung menschlicher Bewegungen sind in der Literatur bekannt: In der Medizin und Orthopädie werden menschliche Bewegungsabläufe analysiert. Dabei geht es vor allem um sehr präzise Messung von Bewegungsabläufen und um die Erkennung von Abweichungen von idealen Bewegungsmustern. Diese Verfahren können in der Regel nur mit einer Bedienung und Off-Line angewendet werden und liefern eine hier nicht geforderte Präzision. Bewegungsmuster werden auch in Ishii (93) unterschieden, dienen dort jedoch lediglich der Klassifikation von Bewegungszuständen, die *Positur* kann nicht segmentiert oder verfolgt werden. In Kurakake (92) wird ein *linienbasierter Ansatz* vorgeschlagen, der jedoch davon ausgeht, daß die relevanten Linien sichtbar sind und der Hintergrund wenig strukturiert ist. Besonders wenn zum Beispiel die Hand vor dem Körper positioniert wird, werden hier die Grenzen erreicht, da sich notwendige Konturen nicht mehr deutlich genug abzeichnen. In Hogg (84) und Rohr (93) werden ebenfalls linienbasierte Ansätze gewählt, bei denen davon ausgegangen wird, daß die Person sich horizontal durch das Kamerabild bewegt. *Konturbasierte Verfahren* werden in Darrell (94), Johnson (94) und Kakadiaris (94/95) beschrieben. Diese Verfahren können jedoch bei typischen Zeigeoperationen nicht arbeiten, da hier die Hand des Benutzers nicht in der Kontur sichtbar ist. Eine Verbesserung bringen modellbasierte Verfahren: Die in Akita (84) und Leung (94) vorgestellten Verfahren basieren wiederum auf Konturbildern, mit den oben genannten Problemen. Andere Verfahren konzentrieren sich eher auf die Erkennung von einigen Handgesten (Maggioni 95, Bröckl-Fox 95). Außerdem sind viele bekannte Ansätze nicht echtzeitfähig, z.B. Gavrilu (96). Ein vielversprechender Ansatz zur Personenverfolgung ist in Wren (96) zu finden: Das im Quellcode verfügbare System *Pfinder* hat jedoch seine Grenzen in der Verarbeitungsgeschwindigkeit von 10 Bildern pro Sekunde für ein einzelnes Bild und läßt nur eine einzige Person als Benutzer zu. Ein Beispiel für eine Anwendung einfacher Bildverarbeitungsroutinen zur Verfolgung mehrerer Personen ist in Bobick (97) beschrieben. Hier wird eine kontextsensitive Auswahl der entsprechenden Algorithmen vorgenommen, bei der wiederum von einer konstanten Beleuchtung ausgegangen wird, und der Hauptfokus in der Erkennung von bestimmten Aktionen innerhalb einer narrativen Umgebung liegt und eher nicht auf der Erkennung von Körperhaltungen. Einer der bei diesem Verfahren verwendeten Ansätze, der des sogenannten *Closed-World Tracking*, ist ein Beispiel für eine (zweidimensionale) Verfolgung mehrerer unabhängiger Personen im Raum (Intille 97), jedoch mit der erwähnten Einschränkung einer konstanten Beleuchtung und eines statischen Hintergrundbildes.

Arbeiten, die in ihrem Anwendungsszenario mit der vorliegenden Arbeit verwandt sind, finden sich in dem *Finger-Pointer*- und *Perseus*-System (Fukumoto 94, Kahn 95/96). Das System *Finger-Pointer* erkennt Zeigegesten des Benutzers auf eine Projektionswand. Dazu muß das System auf den Benutzer kalibriert werden. Der Benutzer darf sich nach der Kalibrierung jedoch nicht im Raum bewegen. Weiterhin bestehen hohe Anforderungen an die Beleuchtungssituation, um die Hand bzw. Zeigefingerspitze in den Kamerabildern eines Stereosystems zu finden. Das System *Perseus* erlaubt das Zeigen einer Person auf Objekte im Raum. Hierbei handelt es sich jedoch lediglich um die Bestimmung einer sehr groben Zeigerichtung, um Objekte auf dem Boden des Raumes zu lokalisieren. Mehrere Personen und eine genauere Zeigerichtung im dreidimensionalen Raum, die sich z.B. für eine Cursorinteraktion eignen würde, werden nicht unterstützt. Im Gegensatz dazu beinhaltet die Dialoggestaltung der Intuitive Schnittstelle vor allem eine freie Beweglichkeit und erlaubt auch die Anwesenheit Dritter.

Es ist festzustellen, daß keines der bekannten Verfahren alle Anforderungen der Intuitiven Schnittstelle erfüllt. Diese sind die freie Beweglichkeit des Benutzers, eine Unterstützung der Anwesenheit Dritter und die Echtzeitfähigkeit auf Standardhardware. Die Problemstellung, die sich hieraus ergibt, ist die Entwicklung einer Personenverfolgung, die mit dem Stand der Technik die Anforderungen erfüllt. Bei der Auswahl eines geeigneten Verfahren muß dabei ein Kompromiß zwischen Leistungsfähigkeit und Einfachheit auf der Bildverarbeitungsebene gefunden werden. In dieser Arbeit wird ein einfaches farbbasiertes Segmentierungsverfahren vorgeschlagen, das zusammen mit einem modellbasiertes Verfolgungsverfahren ein stabiles System liefert.

4.2 Segmentierung

Die meisten in der Literatur bekannten Verfahren der digitalen Bildverarbeitung auf dem Gebiet der Bewegungs- und Gestenerkennung verwenden Grauwertbilder als Basis. Die Nachteile von grauwertbasierten Verfahren sind die hohe Anzahl an Merkmalen, die meist zusätzlich zur Intensitätsinformation benötigt werden, um eine zuverlässige Segmentierung oder Verfolgung von Objekten zu ermöglichen. Eine einfache Möglichkeit ein gesuchtes Objekt vom Hintergrund zu trennen ist die Verwendung von Differenzbildern mit einem statischen Hintergrundbild. Das Differenzbild liefert die veränderten Pixel, eine *Blobanalyse* liefert zusammenhängende Regionen, d.h. es werden alle Pixel mit einem ähnlichen Merkmal (z.B. Farbe) und einer Nachbarschaftsbeziehung (üblicherweise die vier oder acht angrenzenden Pixel) zu einer Region zusammengefaßt (Jähne 95). Die Vorteile des Ansatzes mit Differenzbildern sind die einfache Implementierung und die Möglichkeit stabile Segmentierungsalgorithmen zu implementieren ohne Berücksichtigung des Hintergrundes. Ein Beispiel hierfür ist das *Pfinder*-System (Wren 96). Ein weiterer Vorteil ist die Möglichkeit der effizienten Codierung durch eine Runlength-Codierung oder einen Kettencode. Anschließende Verfahren können entlang der Konturlinie einer Person auf einfache Weise Kopf und Hände detektieren (siehe Abschnitt Verfolgung). Der Hauptnachteil von Differenzbildverfahren ist die Prämisse des statischen Hintergrundes, die in vielen Situationen nicht eingehalten werden kann: Sind mehrere Personen anwesend, oder verändert sich gar der Hintergrund, liefern die Verfahren falsche oder gar keine Ergebnisse. Besonders anfällig sind derartige Verfahren gegenüber wechselnder Beleuchtung. Schon das Ein- oder Ausschalten einer Lampe führt oft zu einer unbrauchbaren Segmentierung. Fällt Tageslicht ein, gibt es oft einen Wechsel von Sonnenlicht und bedeckter Beleuchtung, und derartige Verfahren können nicht arbeiten. Für bestimmte Situationen erscheinen diese Verfahren trotzdem praktikabel.

Erst in den letzten Jahren, auch durch die Verfügbarkeit von preiswerten Farbframegrabbern, sind mehrere farbbasierte Segmentierungsansätze vorgestellt worden (Dai 95, Kjeldsen 96, Rehrmann 94, Saxe 96, Schiele 95, Schuster 96, Schröter 96/97, Wren 96). Die Hauptmängel der vorgeschlagenen Verfahren werden in Schröter (96) genannt: Ein meist hoher Kalibrierungsaufwand, die Beschränkung auf Objektfarben ohne Berücksichtigung des Hintergrundes, das Verwenden einer festen Tabelle zur Segmentierung, die keine Nachregulierung erlaubt, oder adaptive Systeme, die in bestimmten Situationen jedoch auch auf den Hintergrund adaptieren. Für eine stabile Personenverfolgung im Raum muß außerdem gewährleistet sein, daß Farbveränderungen des Objekts an verschiedenen Raumpositionen sowie zu verschiedenen Tageszeiten in das Verfahren integriert werden. Die Ursachen für Farbveränderungen sind die Beleuchtung, eine Kamera- oder Objektbewegung, eine Modifikation der Szene und damit der Interreflexion zwischen Objekten und im starken Maße auch der Lichteinfall der Lichtquelle in die Kamera. In Kjeldsen (96) ist bei wechselnden Beleuchtungen ein erneutes Trainieren notwendig. Ein Verfahren mit relativ hohem Trainingsaufwand für ein neuronales Netz wird in Schiele (95) vorgestellt. Dieses Verfahren verfolgt die hautfarbene Gesichtsregion auch

an verschiedenen Raumpositionen, jedoch fehlen Aussagen über die Auswirkungen von Beleuchtungsschwankungen und Raumpositionen. In Schuster (96) wird die Auswirkung der Beleuchtungsveränderung im RGB-Farbraum untersucht, der stark helligkeitsabhängig ist und daher auch starke Schwankungen aufweist. Das Ergebnis ist eine Begründung für die Verwendung des helligkeitsunabhängigen HUV-Farbraums. Eine Untersuchung der Auswirkungen auf den Farbwert wird nicht durchgeführt. Außerdem wird die Messung lediglich auf einigen Beispielbildern durchgeführt. Die hier beschriebenen Anwendungen zeigen jedoch, daß Änderungen gerade im Verlauf einer Interaktion, z.B. durch die Veränderung der Position des Benutzers, auftreten. Daß die Raumposition trotz konstanter Beleuchtung Einflüsse auf die Farbe und Helligkeit von Objekten haben kann, wird beispielsweise auch in Bobick (97) angemerkt. Auf verschiedene Beleuchtungen wird auch in Saxe (96) eingegangen, mit dem Hinweis, daß die Kombination der Meßergebnisse aus verschiedenen Beleuchtungen für eine spezielle Situation gute Ergebnisse liefert. Eine umfassende Untersuchung wird hier jedoch nicht durchgeführt. Neben der nur unzureichenden Berücksichtigung von Beleuchtungsschwankungen sind einige der vorgeschlagenen Verfahren für eine Stereobildverarbeitung in Echtzeit, bzw. einer Bildrate von mindestens 12 Bildern pro Sekunde auf einem Standard-PC, zu komplex (Rehrmann 96, Schuster 96, Wren 96). Viele Ansätze gehen davon aus, daß der Benutzer vor dem Computermonitor sitzt, sich also nicht im Raum bewegt (Kjeldsen 96, Saxe 96, Schröter 96), oder als einzige Person vor einem statischen Hintergrund anwesend ist (Wren 96).

Im folgenden wird ein für die Personenverfolgung geeigneter Farbraum gewählt, in dem farbige Regionen zu segmentieren sind. Die Regionen bilden die Primitive des Verfolgungssystems und werden von den modellbasierten Verfahren als Input verwendet. Zusätzlich werden verschiedene Farbveränderungen auf Grund von Beleuchtungsschwankungen und Positionsveränderungen untersucht und ein automatisches Kalibrierungssystem entwickelt.

4.2.1 Farbräume

Für die Farbbildverarbeitung gibt es eine Reihe unterschiedlicher Farbräume mit je nach Aufgabe unterschiedlicher Eignung. Der wichtigste Vorteil von hardwareorientierten Farbräumen, wie etwa RGB und YUV, ist die effektive Implementierung, bei wahrnehmungsorientierten Farbräumen, wie etwa HSV, liegt der Vorteil im direkten Bezug zum intuitiven Farbempfinden des Menschen. An dieser Stelle sollen die Vor- und Nachteile nicht diskutiert werden, für eine Übersicht mit Diskussion siehe Schröter (96) und Rehrmann (94). Es ist jedoch festzustellen, daß es keinen idealen Farbraum gibt, der sich für alle Anwendungen eignet, daher ist jeweils der geeignetste Farbraum zu wählen. Für die hier benötigte Anwendung stehen neben einer stabilen Segmentierung vor allem auch die Geschwindigkeit der Berechnung, d.h. die Bildrate, im Vordergrund. Aufgrund der einfachen Berechenbarkeit und der helligkeitsunabhängigen Darstellung wird hier der YUV-Farbraum gewählt (Einige Framegrabber-Karten stellen die Bildinformation im YUV-Farbraum ohne Zeitverlust zur Verfügung). Werden weiterhin Eingangs-Lookup-Tabellen unterstützt, kann auch eine Segmentierung (Binarisierung) ohne Zeitverlust durchgeführt werden. Der YUV-Farbraum ergibt sich aus dem RGB-Farbraum durch folgende Umrechnung (siehe auch Ford <http://www.wmin.ac.uk/ITRG/docs/colorreq>):

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ 0.493 \cdot (B - Y) \\ 0.877 \cdot (R - Y) \end{pmatrix} = \begin{pmatrix} .299 & .587 & .114 \\ -.147 & -.289 & .436 \\ .615 & -.515 & -.1 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

4.2.2 Regionensegmentierung

Bei der Verarbeitung von Farbbildfolgen bieten sich regionenbasierte Verfahren an, da im Gegensatz zu kantenorientierten Verfahren dadurch die Vorteile der Farbe gegenüber Grauwertbildern ausgenutzt werden können (Rehrmann 94). Für die Segmentierung sind eine Reihe von Verfahren bekannt, dazu zählen *Regionenwachstumsverfahren*, *Split-and-Merge-Verfahren* und *Cluster-Analyseverfahren*. Diese Verfahren, wie auch das hierarchische *CSC-Verfahren* (Rehrmann 94), sind für die oben genannten Anforderungen an die Bildrate zu komplex. Es wird daher eine sehr einfache Segmentierung auf Basis einer Farbsegmentierung und anschließender Blobanalyse gewählt. Die Farbsegmentierung, genauer die Trennung von Objektpixeln von Hintergrundpixeln, wird hier mit Schwellwerten oder mit Lookup-Tabellen durchgeführt. Für die Blobanalyse wird eine Funktion der Bildverarbeitungsbibliothek des Framegrabbers verwendet (MIL 96). Die Problemstellung, die sich hieraus ergibt, besteht in der Wahl eines geeigneten Farbintervalls für ein Schwellwertverfahren bzw. der Bestimmung einer geeigneten Lookup-Tabelle.

4.2.2.1 Schwellwertverfahren

In einem ersten Schritt kann die Segmentierung mit einfachen Schwellwerten für den U- und V-Kanal durchgeführt werden. Die Schwellwerte für die zu erkennende Farbregion werden dabei interaktiv mit Hilfe von Schiebereglern eingestellt. Durch die Rückkopplung und die Bewertung des Segmentierungsergebnis durch den Benutzer können diese Schwellen relativ schnell auf die aktuellen Begebenheiten (Farbe, Beleuchtung, etc.) eingestellt werden. Für die Schwellwertoperation zur Binarisierung des Bildes wird eine Funktion der Bildverarbeitungsbibliothek des Framegrabbers verwendet (MIL 96). Ein Segmentierungsverfahren zum Auffinden von Primitiven im Bild gestaltet sich denkbar einfach:

```
PROCEDURE segment(primitive)
// Binarisierung
uv_threshold = primitive->get_thresholds()
binarize (current_image, uv_thresholds)

// Extrahiere Merkmale (z.B. Schwerpunkt, bounding box):
found = features->extract(current_image)
END
```

In der Praxis hat sich gezeigt, daß die Segmentierung mit Schwellwerten allein nicht ausreicht: Auch bei konstanter Beleuchtung wird die Segmentierung fehlerhaft, wenn sich der Benutzer im Raum bewegt. Trotzdem konnten mit dem einfachen Schwellwertverfahren eine grüne Markerregion und die rechte Hand des Benutzers für eine Beispielanwendung erfolgreich verwendet werden. Aufgrund der verschiedenen Farben „Grün“ und „Haut“, die im Hintergrund nicht vorkamen, konnten die Schwellwerte großzügig gewählt werden, so daß auch verschiedene Personen ohne Neukalibrierung das Systems benutzen konnten. Bei wechselnder Beleuchtung oder nicht extrem gut unterscheidbaren Farben ist jedoch mit einer fehlerhaften Segmentierung zu rechnen. Ein weiteres Problem stellen die unterschiedlichen Beleuchtungssituationen in verschiedenen Räumen dar. In Tabelle 4-1 sind die Schwellwerte für die grüne Markerregion und Haut für zwei verschiedene Räume aufgeführt. Die Werte wurden durch Experimente interaktiv ermittelt. Obwohl hier nur zwei verschiedene Situationen dargestellt sind, ist die Größenordnung des Unterschieds deutlich, die eine manuelle Kalibrierung hervorruft.

Tabelle 4-1: Beispielschwellwerte für Farbregionen: Grüner Kalibrierungsmarker, grüner Marker für Benutzer und Hautfarbe für eine Laborumgebung und eine Büror Umgebung.

	min_u	max_u	min_v	max_v
calib lab	0	90	21	116
calib office	87	110	105	128
green lab	0	90	21	104
green office	60	118	13	109
skin lab	0	255	147	219
skin office	101	126	153	191

Die Auswirkung auf eine Segmentierung bei wechselnder Beleuchtung ist leicht vorstellbar. Eine Verbesserung bringt eine statistische Analyse von Objekt- und Hintergrundpixeln und die Berechnung einer Lookup-Tabelle.

4.2.2.2 Statistisches Verfahren

In Schröter (97) wird ein Segmentierungsverfahren mit Lookup-Tabellen auf Basis einer statistischen Analyse von Objekt- und Hintergrundpixeln beschrieben. Dieses Verfahren wird hier im wesentlichen übernommen, als Farbraum wird jedoch der YUV-Farbraum verwendet. Ausgangspunkt für die Berechnung sind zwei normalisierte Histogramme $H_O(u, v)$ und $H_B(u, v)$ für Objekt und Hintergrund, die durch interaktive Auswahl von Beispielregionen im Bild ermittelt werden. Aus den Histogrammen und der jeweiligen Anzahl beteiligter Pixel werden die bedingten Wahrscheinlichkeiten $P(B_O | A)$ und $P(B_B | A)$ für das Ereignis A berechnet, d.h. die Wahrscheinlichkeiten dafür, daß die Farbe zum Objekt bzw. Hintergrund gehört unter der Bedingung, daß die Farbe $[u, v]$ aufgetreten ist. Damit ergibt sich die gesuchte Lookup-Tabelle (Schröter 97):

$$lut(u, v) = \begin{cases} object & \text{if } \frac{P(B_O | A)}{P(B_B | A)} > \varepsilon \\ background & \text{else} \end{cases}$$

Dabei ist ε ein gegebener Schwellwert, der jeweils von Hand eingestellt wurde und die Möglichkeit der Nachregulierung bietet. Das Ergebnis ist eine 2D-Lookup-Tabelle, die auf folgende Weise in zwei 1D-Lookup-Tabellen aufgespalten werden kann (* steht für *don't-care*):

$$lut(u) = \begin{cases} object & \text{if } lut(u, *) = object \\ background & \text{else} \end{cases}$$

$$lut(v) = \begin{cases} object & \text{if } lut(*, v) = object \\ background & \text{else} \end{cases}$$

Damit kann ein Segmentierungsverfahren für Primitive angegeben werden:

```
PROCEDURE segment(primitive)
// Binarisierung
lut = primitive->get_lut()
lut_map (current_image, lut)

// Extrahiere Merkmale (z.B. Schwerpunkt, bounding box):
found = features->extract(current_image)
END
```

Werden Eingangs-Lookup-Tabellen vom Framegrabber unterstützt, läßt sich Binarisierung ohne Zeitverlust durchführen. Tests haben jedoch ergeben, daß die 1D-Lookup-Tabelle deutlich schlechtere Ergebnisse liefert als die 2D-Lookup-Tabelle. Im folgenden wird daher mit der 2D-Lookup-Tabelle gearbeitet.

4.2.3 Einfluß von Beleuchtung und Raumposition

Ausgangspunkt für die Untersuchung der Einflüsse von Beleuchtung und Raumposition auf die Histogramme ist eine Laborumgebung mit relativ konstanter Beleuchtung. Aufgrund der vielen kamerabedingten Störungsursachen, Beleuchtungsschwankungen und das Einfallen von Tageslicht wurden oft starke Schwankungen in der Segmentierungsqualität beobachtet. Die Hauptursachen sind:

- Verschiedene Beleuchtungszustände durch Ein- und Ausschalten von Lampen
- Einfallen von Tageslicht über zwei (verdunkelbare) Deckenlichter und seitliche, in 10 Metern Entfernung befindliche, Fenster (Unterschiede zwischen Tag und Nacht, oder sogar direktes Sonnenlicht)
- Unterschiedliche Kameraeinstellungen oder -bedingungen
- Andere kamerabedingte Störungsursachen (Schuster 96)

Zur Untersuchung wurde ein grüner Ball unter verschiedenen Beleuchtungsbedingungen an zwei verschiedenen Position segmentiert. Die Qualität der Segmentierung wurde durch die Kompaktheit c der segmentierten Region ermittelt:

$$c = \frac{p^2}{4\pi A}$$

Hierbei steht A für die Fläche der segmentierten Region (ein Pixel entspricht einer Fläche von 1), p für den Umfang, der sich aus den Längen der Pixelkanten inklusive der Kanten von Löcher berechnet. Hierbei entspricht ein Pixel einem Umfang von 4 und (zur Kompensation des Treppenstufeneffekts) eine diagonale, aus zwei Pixeln bestehende Kante einer Länge von 1.414. Im Idealfall würde das Bild eines Balls einen Kreis ergeben und damit eine Kompaktheit von $c = 1$. Außerdem wurde die Qualität der Segmentierung nach Summieren der Histogramme beider Positionen und Beleuchtungen ermittelt. Tabelle 4-2 zeigt die zugehörigen Histogramme, Tabelle 4-3 die Segmentierungsergebnisse bzw. die gemessene Kompaktheit.

Tabelle 4-2: Histogramme eines Testobjektes (grüner Ball) im UV-Bereich

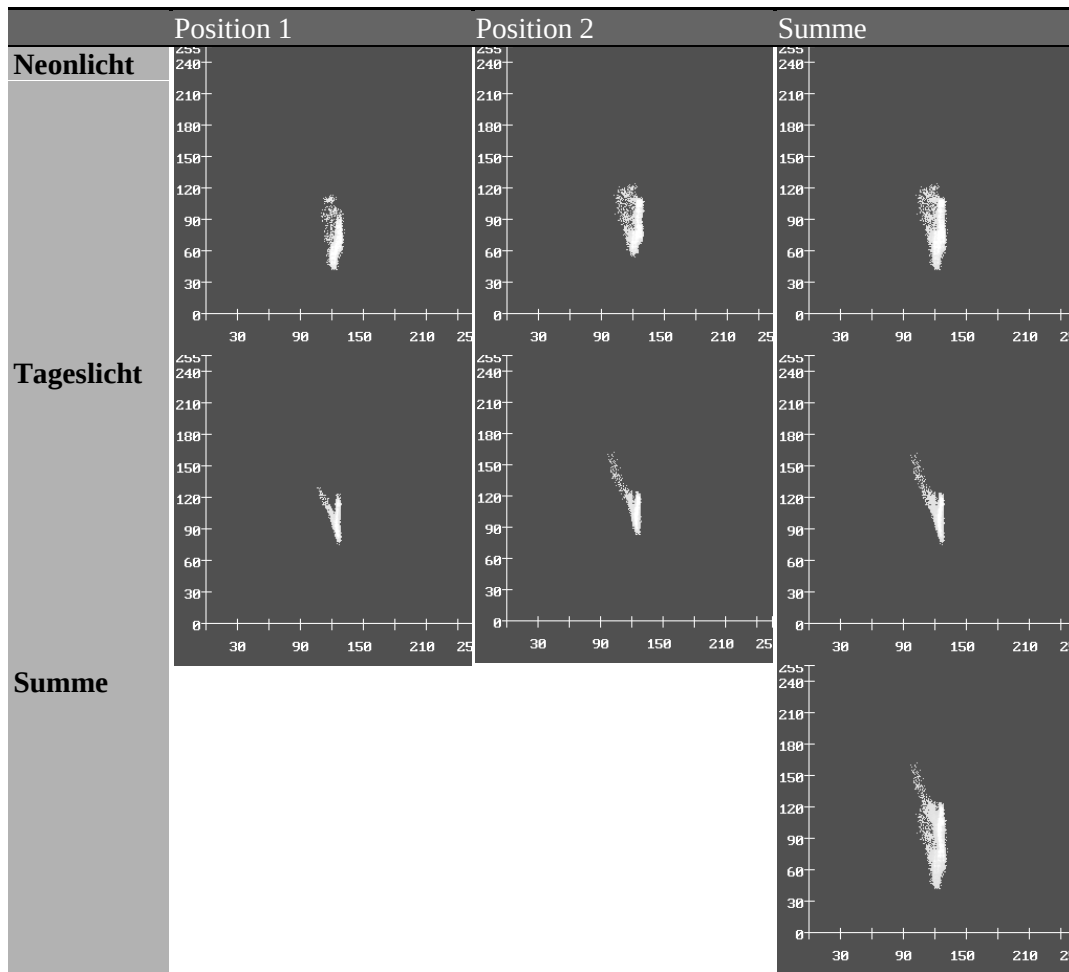


Tabelle 4-3: Meßergebnisse Kompaktheit grüner Ball für Histogramme aus Tabelle 4-2 (1 = ideal), die Werte in Klammern entsprechen der Summe der Histogramme aus beiden Positionen und der Summe der Histogramme auch von beiden Beleuchtungen

	Position 1	Position 2
Neonlicht	5.9 (4.1) (4.3)	2.6 (2.4) (2.2)
Tageslicht	1.8 (1.8) (1.7)	2.7 (2.1) (2.5)
Mischlicht	6.1 (6.1) (-)	4.7 (8.3) (-)

Aus Tabelle 4-3 ist zu entnehmen, daß die Segmentierung durch die Summierung der Histogramme verschiedener Messungen annähernd für beide Meßsituationen gute Ergebnisse liefert. So ergibt die Segmentierung des Balls unter Neonlicht an Position 1 mit einer Summierung der Histogramme beider Positionen eine Kompaktheit von 4,1. Eine weitere Summierung auch der anderen Beleuchtungssituation ergibt eine Kompaktheit von 4,3. Für Position 2 ergeben sich analog die Werte 2,4 und 2,2. Weiterhin zeigt die Kontrolle der Segmentierung an einer Position mit der Lookup-Tabelle einer anderen Position oder Beleuchtung, daß die Summierung tatsächlich eine Verbesserung liefert bzw. eine Segmentierung überhaupt erst ermöglicht (für Details der Untersuchungen sie-

he Kulesa 98-1). Die Schlußfolgerung hieraus ist, daß die Addition der Histogramme unterschiedlicher Positionen und Beleuchtungen zu einer kombinierten Lookup-Tabelle führt, die unter allen genannten Bedingungen befriedigende Ergebnisse liefert.

4.2.4 Kalibrierung

Die Ergebnisse aus dem letzten Abschnitt liefern ein praktisches Verfahren zur interaktiven Kalibrierung.

```
PROCEDURE calibrate
histogramm_type HIST_SUMME = EMPTY

Für n unterschiedliche Raumpositionen
  BEGIN
  Für n verschiedene Beleuchtungsbedingungen
    BEGIN
    1) Wähle einen Objektbereich im Bild
    2) Wähle einen Hintergrundbereich im Bild
    2) Bestimme Histogramm
    3) HIST_SUMME += Histogramm
    END
  END
END
```

Weiterhin kann ein Verfahren zur automatischen Kalibrierung angegeben werden, das für die Regionen eines zu kalibrierenden Objekts (in diesem Falle Hände und T-Shirt des Benutzers) die Summierung der Histogramme vornimmt. Hierzu werden die Schwerpunkte der betreffenden Regionen beispielsweise mittels Kalman-Filterung verfolgt und immer dann, wenn sich der Benutzer nicht bewegt, also an einer neuen Position angekommen ist, die Histogramme der segmentierten Regionen zu den bisher gespeicherten aufaddiert. Dadurch kann eine Kalibrierung für den gesamten Raum auf einfache Weise vorgenommen werden.

```
PROCEDURE calibrate
histogramm_type HIST_SUMME[#regionen] = EMPTY

Warte bis Objekt gefunden
Für alle i = Objektregion
  Bestimme Histogramm(i)
  HIST_SUMME[i] = Histogramm(i)

WHILE not ( ende )
  BEGIN
  Verfolge Objekt
  IF Objekt bewegt sich nicht
    Für alle i = Objektregion
      Bestimme Histogramm(i)
      HIST_SUMME[i] += Histogramm(i)
    END
  END
END
```

Erste Ergebnisse dieses Verfahren zeigen intuitiv eine Segmentierungsqualität, die durchaus vergleichbar ist mit der eines Schwellwertverfahrens, das von Hand erstellte und mehrmals optimierte Schwellwerte verwendet (Kulesa 98-1). Damit liefert dieses Verfahren eine einfache und schnelle Möglichkeit einer Kalibrierung der Farbregionen für eine Verfolgung im Raum.

4.3 3D Rekonstruktion

In dem hier beschriebenen System werden aus den Ergebnissen der Segmentierung die 3D-Koordinaten korrespondierender Bildinformationen bestimmt. Hierzu werden zwei oder mehrere Kameras benötigt, deren Abbildungsparameter (Öffnungswinkel und Rotationsparameter) in einem Kalibrierungsschritt bestimmt werden müssen. Im folgenden werden zunächst die verwendete Schreibweise für geometrische Transformationen, das verwendete Lochkameramodell und die Verfahren zur Kamerakalibrierung erläutert. Anschließend werden die Verfahren zur Rücktransformation eines Punktes und eines Stabes sowie seiner Zeigerichtung dargestellt.

4.3.1 Geometrische Transformationen

Für die Darstellung der Abbildungsverhältnisse und der Transformation zwischen den verschiedenen Koordinatensystemen wird die Notation aus (Foley 91) bzw. (Klette 96) verwendet, für eine Darstellung der trigonometrischen Abhängigkeiten siehe (Stöcker 95). Die Beziehung zwischen einem Punkt im Weltkoordinatensystem $p_w = (x_w, y_w, z_w, 1)$ und einem Bildpunkt $p_p = (x_p, y_p, f, 1)$ im Kamerakoordinatensystem ist

$$p_p = N P R_z R_x R_y T p_w \quad (4-1)$$

mit der Translationsmatrix

$$T = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und den Matrizen für die Rotation um die x-, y- und z-Achse:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Die Translationsmatrix, T , verschiebt den Ursprung des Weltkoordinatensystems in den Ursprung des Kamerakoordinatensystems. Die Rotationsmatrizen ändern die Orientierung des Koordinatensystems entsprechend der Orientierung der Kamera. Die Matrizen P und N schließlich sind für die perspektivische Projektion und Transformierung der zweidimensionalen Bildebene verantwortlich. Im folgenden wird stets von einem rechtshändigen Koordinatensystem ausgegangen, in dem positive Rotationen gegen den Uhrzeigersinn ausgeführt werden, wenn von der positiven Achse auf den Ursprung geblickt wird (Foley 91).

4.3.2 Lochkameramodell

Eine Kamera liefert ein zweidimensionales Bild einer dreidimensionalen Szene. Um diesen Vorgang der Bildaufnahme zu beschreiben, bedient man sich eines Modells für das Abbildungssystem. Das einfachste Modell hierbei ist das Lochkameramodell, das eine perspektivische Projektion modelliert. Im folgenden wird das Lochkameramodell aus (Niemann 90) verwendet.

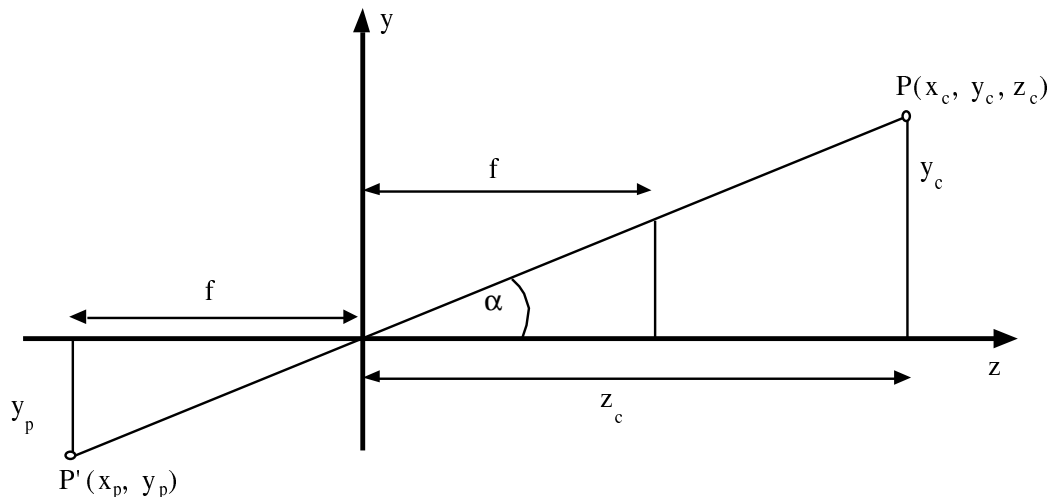


Abbildung 4-1: Lochkameramodell: Projektion eines Punktes auf die Bildebene

Das Lochkameramodell projiziert einen Punkt $P = (x_c, y_c, z_c)$ im Kamerakoordinatensystem auf einen zweidimensionalen Bildpunkt mit den Koordinaten $P' = (x_p, y_p)$.

Ist f die Brennweite der Kamera, so gilt nach Strahlensatz:

$$x_p = \frac{f \cdot x_c}{z_c}, \quad y_p = \frac{f \cdot y_c}{z_c} \quad (4-2)$$

Ein reales Linsensystem weicht von dem Model der Lochkamera ab. Handelsübliche Linsen weisen in der Regel Verzeichnungen auf, die dazu führen, daß Linien nicht als Linien dargestellt werden. Diese Verzeichnungen können am Rand des Bildes mehrere Pixel betragen (Jähne 95), so daß sie für Präzisionsmessungen korrigiert werden müssen. Im folgenden werden wir auf diese Korrektur verzichten, da es sich bei dem vorgestellten System nicht um eine Präzisionsmessung handelt. Die Verfahren zur Kalibrierung werden dadurch vereinfacht. Im folgenden wird unterschieden zwischen Punkten $P_w = (x_w, y_w, z_w)$ im Weltkoordinatensystem (WKS), den Punkten

$P_c = (x_c, y_c, z_c)$ im Kamerakoordinatensystem (KKS) und den Punkten der Bildebene $P_p = (x_p, y_p)$ im Bildkoordinatensystem (IKS). Die Abbildungsgeometrie folgt der Darstellung in Niemann (90).

4.3.3 Kamerakalibrierung

Kamerakalibrierung ist der Prozeß, die unbekannten Parameter der Kamera eines bestimmten Kameramodells durch ein geeignetes Kalibrierungsbild zu bestimmen (Niemann 90). Es wird zwischen internen und externen Kameraparametern unterschieden. Zu den internen gehören die Brennweite, Verzeichnungskoeffizient, Skalierungsfaktor, Bildmittelpunktkoordinaten. Die externen Kameraparameter bestehen aus Rotation und Translation des Kamerakoordinatensystems (Klette 96). Im folgenden sollen lediglich die Brennweite bzw. der horizontale und vertikale Öffnungswinkel als interne Parameter der Kamera berechnet werden. Dadurch vereinfacht und reduziert sich der Prozeß der Kalibrierung auf die horizontale und vertikale Vermessung eines Rechtecks mit bekannter Größe im Kamerabild. Für eine Kalibrierung auch der übrigen Parameter sind mindestens 5 Meßpunkte (Tsai 86, Klette 96) bzw. 9 Meßpunkte (Niemann 90) eines Kalibrierungsmusters erforderlich. Im folgenden wird eine einfache Methode zur Bestimmung der Öffnungswinkel erläutert. Diese Methode ist ähnlich der in Kohler (96) beschriebenen, wobei hier die Öffnungswinkel in x- und in y-Richtung aus zwei getrennten Meßwerten berechnet werden.

4.3.3.1 Bestimmung der Öffnungswinkel

Aufgabe: Bestimmung der Kameraöffnungswinkel α und β sowie der Brennweite und Pixelbreite und -höhe $pixel_width$, $pixel_height$ mittels eines geeigneten Kalibrierungsaufbaus.

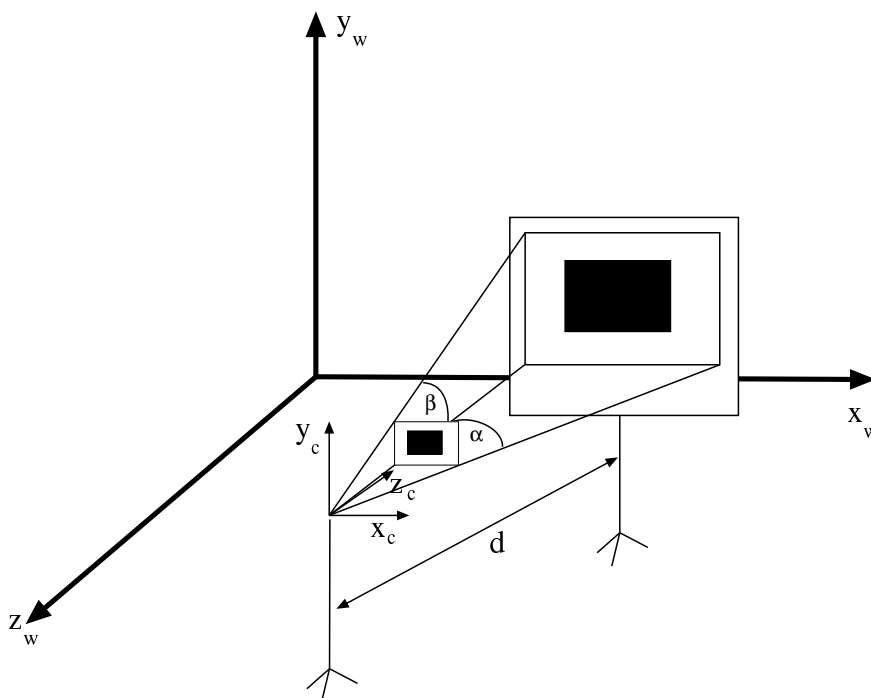


Abbildung 4-2: Aufnahme des Kalibrierungsfeldes (DINA4-Blatt)

Zur Kalibrierung wird ein helles DIN A4-Blatt (29.7 x 21.0 cm) vor dunklem Hintergrund aufgenommen (Abbildung 4-2). Alternativ kann auch ein farbiges Blatt vor andersfarbigem Hintergrund verwendet werden, solange ein starker Kontrast zwischen Vorder- und Hintergrund bestehen bleibt, der eine einfache Segmentierung des Blattes im Kamerabild mittels Schwellwerte ermöglicht. Anschließend sind der Abstand d vom Fokuspunkt zu vermessen und die Größe des DIN A4-Blattes im Kamerabild zu bestimmen. Die Öffnungswinkel in horizontaler Richtung α und in vertikaler Richtung β lassen sich nun aus dem Strahlensatz und einfacher Trigonometrie berechnen:

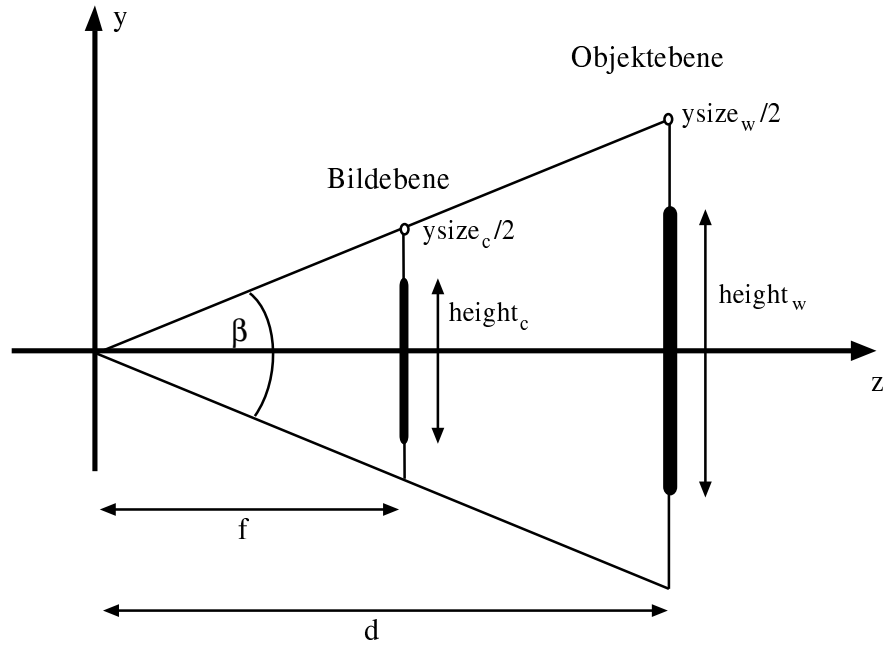


Abbildung 4-3: Bestimmung der Öffnungswinkel

Nach Abbildung 4-3 ergibt sich die y-Größe des Bildes in der Objektebene $ysize_w$ aus der Anzahl Pixel in y-Richtung $ysize_c$ geteilt durch die Anzahl Pixel des Kalibrierungsfeldes im Bild $height_c$ multipliziert mit der Höhe des Kalibrierungsfeldes $height_w$ (analog ergibt sich die x-Größe des Bildes):

$$ysize_w = \frac{ysize_c}{height_c} \cdot height_w, \text{ mit } height_w = 21.0 \text{ cm}$$

$$xsize_w = \frac{xsize_c}{width_c} \cdot width_w, \text{ mit } width_w = 29.7 \text{ cm}$$
(4-3)

Hieraus lassen sich die Öffnungswinkel α und β mittels der Winkelbeziehungen der rechtwinkligen Dreiecke mit den Katheten d und $ysize_w$ bzw. $xsize_w$ berechnen:

$$\alpha = 2 \cdot \arctan\left(\frac{0.5 \cdot xsize_w}{d}\right)$$

$$\beta = 2 \cdot \arctan\left(\frac{0.5 \cdot ysize_w}{d}\right)$$
(4-4)

Weiterhin ergeben sich die Brennweite und die Konstanten $pixel_width$ für die Breite und $pixel_height$ für die Höhe eines Pixels, die für die spätere Berechnung verwendet werden:

$$\begin{aligned}
 f_x &= 0.5 \cdot \frac{xsize_c}{\tan(0.5 \cdot \alpha)}, & f_y &= 0.5 \cdot \frac{ysize_c}{\tan(0.5 \cdot \beta)} \\
 pixel_width &= \frac{2 \cdot \tan\left(\frac{\alpha}{2}\right)}{xsize_c} \cdot f_x = \frac{xsize_w}{xsize_c \cdot d} \cdot f_x \\
 pixel_height &= \frac{2 \cdot \tan\left(\frac{\beta}{2}\right)}{ysize_c} \cdot f_y = \frac{ysize_w}{ysize_c \cdot d} \cdot f_y
 \end{aligned} \tag{4-5}$$

Aufgrund der Meßungenauigkeit und Diskretisierung ergeben sich aus den Öffnungswinkeln unterschiedliche Werte f_x , f_y für die Kamerakonstante f .

4.3.3.2 Bestimmung der Rotationsparameter

Aufgabe: Bestimmung der Winkel ϕ , θ (Rotation um die x- bzw. die y-Achse) der Kamera mittels eines bekannten Markerpunktes im Weltkoordinatensystem $M_w = (mx_w, my_w, mz_w, 1)$.

In diesem Schritt sind die externen Kameraparameter, d.h. die Rotationen der Kamera um die x- und die y-Achse, zu bestimmen. Hierbei wird davon ausgegangen, daß die Kamera nicht seitlich geneigt ist, d.h. der Rotationswinkel um die z-Achse gleich Null ist.

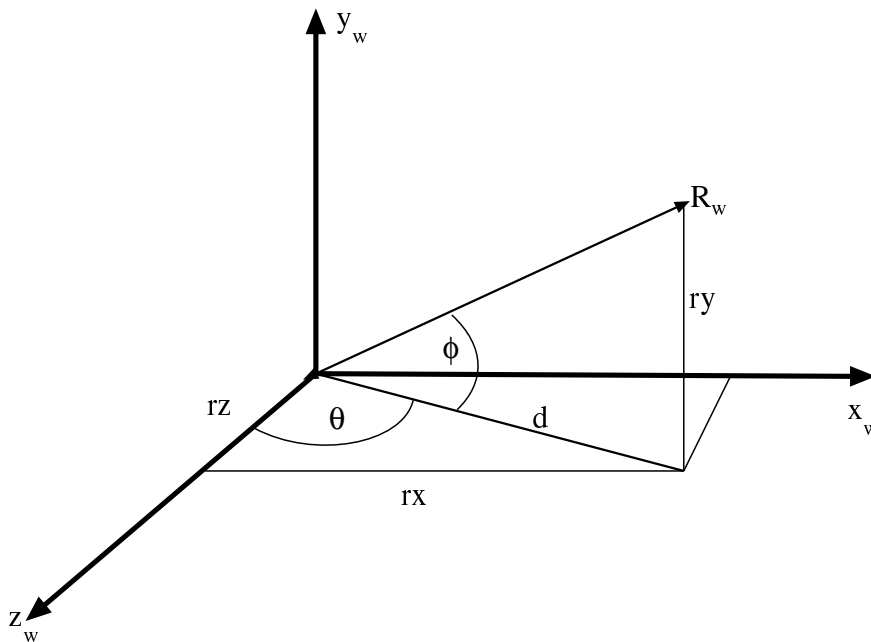


Abbildung 4-4: Bestimmung der Rotationswinkel

Für die Bestimmung der Rotationen werden zunächst die Winkel des Richtungsvektors von der Kamera $R_w = (rx_w, ry_w, rz_w, 1)$ zu einem bekannten Markerpunkt im Weltkoordinatensystem $M_w = (mx_w, my_w, mz_w, 1)$ bestimmt. Mit der Kameraposition $C_w = (cx_w, cy_w, cz_w, 1)$ ergibt sich:

$$R_w = M_w - C_w$$

Für R_w werden jetzt die Winkel θ (Rotation um die y-Achse) und ϕ (Rotation um die x-Achse) bestimmt:

$$\begin{aligned}\phi &= \arctan\left(\frac{ry}{d}\right) \\ \theta &= \arctan\left(\frac{rx}{rz}\right) \\ \text{mit } d &= \sqrt{rx^2 + rz^2}\end{aligned}$$

Anschließend wird die Position $p = (x_p, y_p)$ des Markerpunktes im Bild bestimmt. Die gesuchten Rotationswinkel ergeben sich erneut durch einfache Trigonometrie wie folgt:

$$\begin{aligned}rot_x &= \phi + \arctan\left(\frac{x_p \cdot pixel_width}{f_x}\right) = \phi + \arctan\left(x_p \cdot \frac{2 \cdot \tan\left(\frac{\alpha}{2}\right)}{xsize_c}\right) \\ rot_y &= \theta + \arctan\left(\frac{y_p \cdot pixel_height}{f_y}\right) = \theta + \arctan\left(y_p \cdot \frac{2 \cdot \tan\left(\frac{\beta}{2}\right)}{ysize_c}\right)\end{aligned} \quad (4-6)$$

Wiederum geben $xsize_c$ und $ysize_w$ jeweils die Anzahl Pixel in x- bzw. y-Richtung an. Hier wird davon ausgegangen, daß die Kamera erst um die x-Achse gedreht und anschließend um die y-Achse geschwenkt wird. Diese Reihenfolge läßt die Rotation der Kamera in z-Richtung unbeeinflusst und muß auch bei der Bestimmung der Rotationen eingehalten werden. Obiges Verfahren folgt der in Foley (91) dargestellten Reihenfolge für die Bestimmung der Rotationswinkel eines Richtungsvektors.

4.3.4 Rücktransformation

Zur Bestimmung der 3D-Konfiguration von Objekten, die im Stereobild verfolgt werden, müssen die in Pixelkoordinaten gegebenen Ergebnisse der Bildverarbeitung entsprechend den zuvor bestimmten Abbildungsverhältnissen der Kameras transformiert werden. Für das hier beschriebene System werden die Rekonstruktion eines Punktes im Raum und die Lage und Zeigerichtung eines Zeigestabs verwendet.

4.3.4.1 Koordinatentransformation

Aufgabe: Gegeben sei ein Punkt $p_p = (x_p, y_p)$ im Kamerabild. Gesucht ist der Vektor $p_w = (x_w, y_w, z_w, 1)$, der sich aus p_p durch Koordinatentransformation in das Weltkoordinatensystem ergibt.

Nach der oben eingeführten Notation und unter der Annahme, daß die Kamera erst um die x-Achse, anschließend um die y-Achse und nicht in z-Richtung gedreht wurde, ergibt sich ein Bildpunkt aus folgender Beziehung:

$$p_p = NPR_x R_y T p_w = M p_w \quad (4-7)$$

Die Umkehrung lautet:

$$p_w = M^{-1} p_p = T^{-1} R_y^{-1} R_x^{-1} (NP)^{-1} p_p \quad (4-8)$$

Im ersten Schritt werden die Bildkoordinaten $p_p = (x_p, y_p)$ in das Kamerakoordinatensystem transformiert. Zur Vereinfachung wird die Kamerakonstante $f = 1$ gesetzt, da zur Rekonstruktion für jede Kamera lediglich der Richtungsvektor durch den Punkt p berechnet werden muß. Seien (c_x, c_y) die Bildkoordinaten des Kameramittelpunktes, dann ergibt sich mit Formel (4-5) der Punkt $p_c = (x_c, y_c, z_c)$ aus:

$$\begin{aligned} x_c &= (x_p - c_x) \cdot \text{pixel_width} \\ y_c &= (y_p - c_y) \cdot \text{pixel_height} \\ z_c &= 1 \end{aligned} \quad (4-9)$$

Anschließend werden die Rotationen um die x- und um die y-Achse angewandt und die Translation durchgeführt:

$$p_w = T^{-1} R_y^{-1} R_x^{-1} p_c \quad (4-10)$$

4.3.4.2 Punkt im Raum

Aufgabe: Gegeben seien die Punkte $p_1 = (x_1, y_1)$ und $p_2 = (x_2, y_2)$ im linken bzw. rechten Kamerabild. Gesucht ist der Punkt $p_w = (x_w, y_w, z_w)$ des Weltkoordinatensystems, der diese Bilder erzeugt.

Zur Bestimmung der 3D-Koordinaten eines Objekts im Raum, werden die Pixelkoordinaten $p = (x_p, y_p)$ seines Schwerpunkts in beiden Kamerabildern bestimmt. Aufgrund der Abbildungseigenschaften der Kamera und der Diskretisierung sind diese Punkte nicht exakt das Bild eines Punktes im Weltkoordinatensystem, d.h. die Strahlen g_1 und g_2 vom Kameraursprung C_1 bzw. C_2 durch p_1 bzw. p_2 schneiden sich nicht (Abbildung 4-5). Als Näherung für den gesuchten Punkt p_w wird der Mittelpunkt P_s der kürzesten Verbindungsstrecke der beiden Strahlen g_1 und g_2 berechnet.

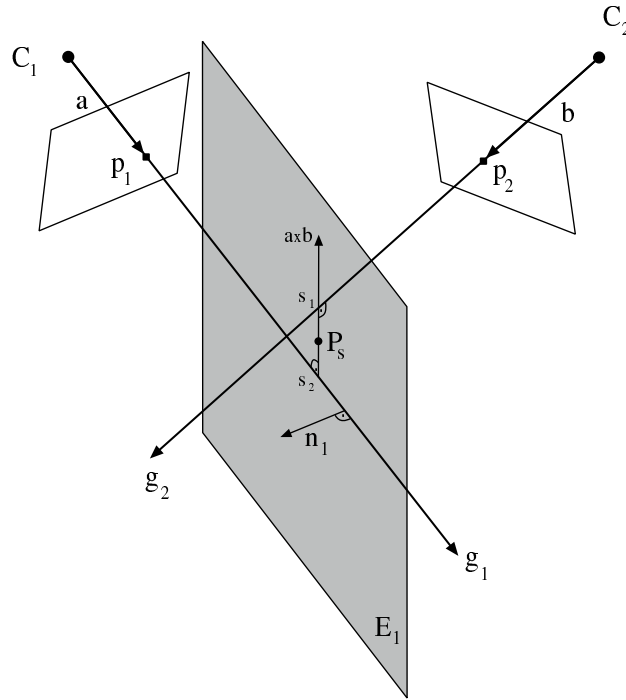


Abbildung 4-5: Schnittpunkt zweier windschiefer Geraden

Der gesuchte Punkt P_s ist der Mittelpunkt zwischen den beiden Punkten s_1 und s_2 . Eine Koordinatentransformation der Punkte p_1 und p_2 nach den Formeln (4-9) und (4-10) liefert zunächst die Richtungsvektoren a und b . Nach Abbildung 4-5 ergibt sich der Punkt s_1 aus dem Schnittpunkt von g_2 mit der Ebene E_1 , die parallel zu a verläuft. Der Normalenvektor von E_1 berechnet sich aus $n = a \times (a \times b)$, wobei $(a \times b)$ der Vektor ist, der senkrecht zu a und b steht. Damit können wir die Parameterformen von g_2 und E_1 angeben:

$$g_2 : x = C_2 + t \cdot b$$

$$E_1 : n \bullet x = d, \quad \text{mit } d = n \bullet C_1$$

Damit ergibt sich der Schnittpunkt s_1 (Stöcker 95):

$$s_1 = C_2 + t_0 \cdot b, \quad \text{mit } t_0 = \frac{d - n \bullet C_2}{n \bullet b}$$

Für s_2 wird analog verfahren. Damit ergibt sich P_s als Näherung für den gesuchten Punkt p_w aus:

$$P_s = s_2 + \frac{1}{2}(s_1 - s_2)$$

$$= \frac{1}{2}(s_1 + s_2) \quad (4-11)$$

4.3.4.3 Stab

Aufgabe: Gegeben seien die Schwerpunkte zweier Pixelwolken $s_1 = (x_1, y_1)$, $s_2 = (x_2, y_2)$, die das Abbild eines Stabs im linken und rechten Kamerabild darstellen. Weiterhin seien jeweils der Winkel γ der Hauptachse gegeben. Gesucht sind der Richtungsvektor r des Stabes und sein Schnittpunkt mit der xy -Ebene $p_s = (x_s, y_s, 0)$ sowie die 3D-Position des Stabes $p = (x_w, y_w, z_w)$.

Für die Verfolgung eines Zeigestabs sollen die 3D-Position und die Zeigerichtung bestimmt werden. In Hoch (96) wird ein Verfahren beschrieben, daß die gesuchte Zeigerichtung durch den Schnittpunkt von drei Ebenen ($E_1, E_2, z = 0$) berechnet (Abbildung 4-6). Dieses Verfahren liefert ausschließlich den Schnittpunkt mit der xy -Ebene. Unter Ausnutzung des speziellen Aufbaus, können die Normalenvektoren der Ebenen effektiver berechnet werden und zusätzlich stehen die 3D-Position und der Richtungsvektor des Stabs zur Verfügung.

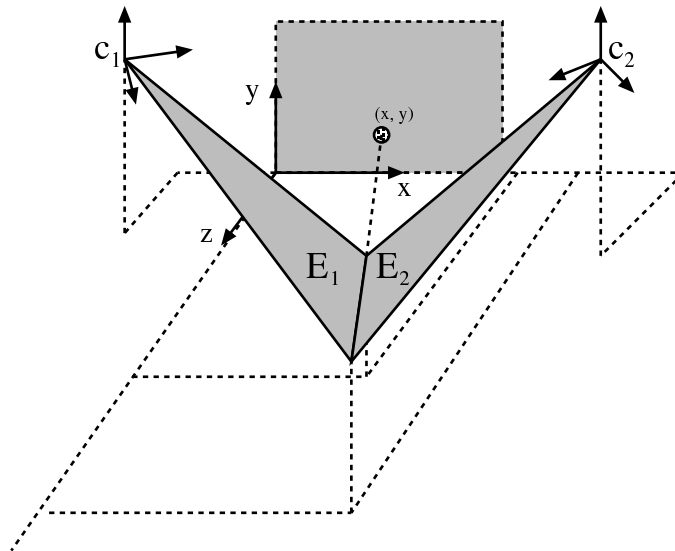
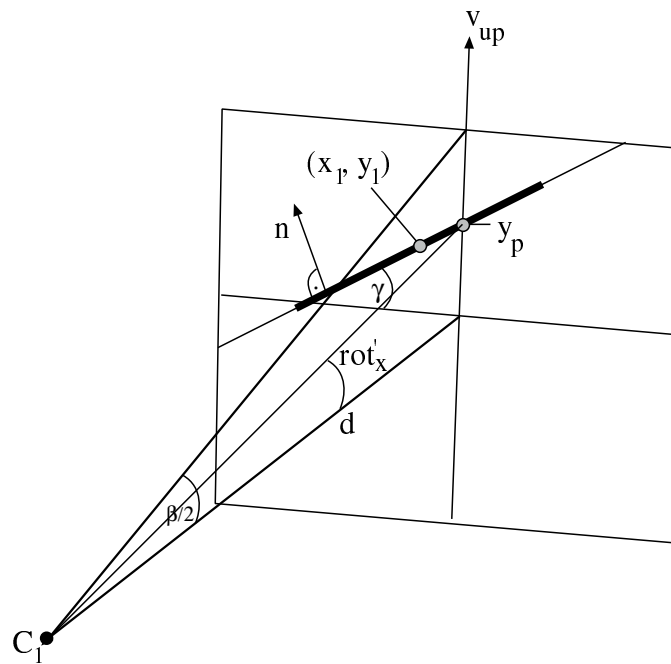


Abbildung 4-6: Situation bei der Verfolgung eines Stabs mit zwei Kameras

Der Richtungsvektor ergibt sich aus dem Kreuzprodukt der Normalenvektoren von E_1 und E_2 , die 3D-Position kann aus dem Schwerpunkten berechnet werden. So die Bildverarbeitung die Schwerpunkte des Stabs im Kamerabild und den jeweiligen Winkel der Hauptachse liefert, können die Normalenvektoren effektiv berechnet werden (im folgenden exemplarisch für Kamera 1):

Abbildung 4-7: Berechnung der Normalenvektoren der Stabebene E_1

Im ersten Schritt wird der Normalenvektor n der Ebene E_1 in Abhängigkeit des Kameranormalenvektors v_{up} berechnet (siehe Abbildung 4-7). Mit den bekannten Rotationsmatrizen R_x, R_y der Kameras ergibt sich v_{up} aus:

$$v_{up} = R_y R_x \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Der Schnittpunkt y_p der Hauptachse mit der y-Achse des Bildkoordinatensystems ergibt sich aus der Geradengleichung wie folgt:

$$y_p = y_1 - x_1 \cdot \tan(\gamma)$$

Für die Bestimmung des gesuchten Normalenvektors sind nun die zwei Rotationswinkel zu bestimmen, die v_{up} mit dem gesuchten Normalenvektor zur Deckung bringen, d.h. es ist eine Rotation um die x-Achse der Kamera notwendig, die y_p in den Bildmittelpunkt rückt und eine Rotation um die z-Achse, so daß die Hauptachse des Stabs mit der x-Achse zusammenfällt.

Mit

$$d = \frac{ysize}{2 \cdot \tan\left(\frac{\beta}{2}\right)}$$

$$rot'_x = \arctan\left(\frac{y_p}{d}\right)$$

erhalten wir:

$$rot'_x = \arctan\left(y_p \cdot \frac{2 \cdot \tan\left(\frac{\beta}{2}\right)}{ysize}\right) = \arctan(y_p \cdot pixel_height) \quad (4-12)$$

$$rot'_z = \gamma$$

Die Rotation um die z-Achse ergibt sich direkt aus γ . Damit errechnet sich der Normalenvektor der Ebene aus:

$$n_1 = R_y R_x R'_x R'_z \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \quad (4-13)$$

mit $R'_x = R_x(-rot'_x)$, $R'_z = R_z(-\gamma)$

Im nächsten Schritt werden aus den Normalenvektoren n_1, n_2 der Stabebene beider Kameras und der 3D-Position des Schwerpunkts $p = (x_w, y_w, z_w)$ der Richtungsvektor des Stabs und die Zeigeposition auf der xy-Ebene berechnet. Die 3D-Position wird dabei nach dem im vorherigen Abschnitt beschriebenen Verfahren berechnet. Der Richtungsvektor lässt sich aus dem Kreuzprodukt der beiden Normalenvektoren ermitteln, der Schnittpunkt wird aus der Parameterform der Geradengleichung $g: \vec{x} = \vec{a} + t \cdot \vec{b}$ mit Aufpunkt p und Richtungsvektor r und Lösen nach t für $z_s = 0$ gewonnen:

$$r = n_1 \times n_2$$

$$p_s = (x_s, y_s, 0) = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} + \frac{-z_w}{r_z} \cdot \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \quad (4-14)$$

4.4 Verfolgung

In den vorangehenden Abschnitten wurden Techniken zum Auffinden von Primitiven in Bildern und deren 3D-Rekonstruktion im Falle von Stereokameras behandelt. Beim Auffinden der Primitive ergeben sich oft Mehrdeutigkeiten aufgrund der Szene oder einer nicht perfekten Segmentierung. Für die Echtzeitbildverarbeitung ist es weiterhin wichtig, den Berechnungsaufwand möglichst klein zu halten und die Redundanz zwischen den Bildern einer Sequenz auszunutzen. Die Aufgabe der Verfolgung ist es unter diesen Voraussetzungen kontinuierliche und auch verlässliche Daten von im Bild erkannten Primitiven oder Bildteilen zu generieren. Im vorliegenden Fall soll die Verfolgung ein stabiles System liefern trotz Verwendung von Standardkameras, die ein verrauschtes Signal liefern können, und einer nicht perfekter Segmentierung z.B. aufgrund von Beleuchtungsschwankungen, die auch mit einer Kalibrierung auf die Beleuchtungssituation oft die Segmentierungsergebnisse negativ beeinflussen. Zu diesem Zweck bedient man sich oft sogenannter *Heuristiken*, die beispielsweise eine initiale Suche nach den relevanten Regionen bei der Personenverfolgung erleichtern und damit den Suchraum auf einen für den Anwendungsfall vertretbaren Rechenaufwand reduzieren. Heuristiken sind einfache Hypothesen oder Annahmen über einen Sachverhalt zum Zweck des besseren Verständnis. In der Bildverarbeitung werden Heuristiken als Hilfsmittel zur Vereinfachung eines Prozesses oder zum Auflösen von Mehrdeutigkeiten verwendet. Im Gegensatz zu *modellbasierten Verfahren* liegen hierbei keinerlei expliziten Modelle eines Sachverhaltes im Rechner vor, vielmehr wird die Verarbeitungsreihenfolge in eine den Annahmen entsprechende Richtung gelenkt. Derartige Verfahren können jedoch mit modellbasierten Verfahren kombiniert werden.

Im folgenden wird zunächst der *Kalman-Filter* vorgestellt als wichtiges Werkzeug zur Filterung verrauschter Daten und somit als Basis für eine stabile Verfolgung. Anschließend wird am Beispiel der *Verfolgung über Silhouette* die Leistungsfähigkeit von Heuristiken in Form von Regeln zur Lösung eines Verfolgungsproblems erläutert. Diese Ergebnisse werden später in den modellbasierten Verfahren eingesetzt und in Form einer Wissensbasis konsequent weiterentwickelt.

4.4.1 Kalman-Filterung

Ein Kalman-Filter gestattet es, Vorhersagen über ein dynamisches System mit Hilfe von statistischen Annahmen über den Meßfehler einer Meßwertreihe zu machen. Die Schätzung des Filters basiert auf der Maximum-Likelihood-Schätzmethode und ist im statistischen Sinne optimal (du Plessis 67). Dieser Filter ist besonders für die Echtzeitverarbeitung geeignet, da er die Vorhersage des zukünftigen Systemzustandes erlaubt, ohne die vorherigen Meßwerte speichern zu müssen. In der Bildverarbeitung kann der Filter verwendet werden, um die Bewegung eines Objekts im Bild vorherzusagen und Segmentierungsergebnisse zu glätten. In dem hier vorgestellten System wird der Kalman-Filter verwendet, um den Schwerpunkt der segmentierten Region zu filtern und die Position der Suchregion im nächsten Bild vorherzusagen. Die Position eines Punktes zum Zeitpunkt t_k ergibt sich aus der vorherigen Position s_{k-1} , seiner Geschwindigkeit v_{k-1} und der Beschleunigung a_{k-1} :

$$s_k = s_{k-1} + v_{k-1} \cdot \Delta t + \frac{1}{2} a_{k-1} \cdot \Delta t^2 \quad (4-15)$$

Für den Kalman-Filter wird die Bewegung analog der Notation aus Gelb (74) durch folgendes System modelliert:

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1} \quad (4-16)$$

Der Zustand des Systems zum Zeitpunkt t_k resultiert aus der Verknüpfung der Übergangsmatrix Φ_{k-1} mit dem vorherigen Systemzustand überlagert durch weißes Rauschen w_{k-1} . Der Zustandsvektor $x_k = (s_k, v_k)$ enthält die Position und Geschwindigkeit des Punktes. Experimente zeigen, daß die Modellierung der Beschleunigung durch das weiße Rauschen hinreichend berücksichtigt wird (Kohler 97). Diese Beobachtung konnte mit dem vorliegenden System bestätigt werden. Die aktuellen Meßwerte für die Position ist durch folgendes System gegeben:

$$z_k = H_k x_k + e_k \quad (4-17)$$

Dabei stellt z_k eine Menge von Meßwerten der Bildverarbeitung zum Zeitpunkt t_k dar, H_k beinhaltet die Linearkombinationen des Zustandsvektors, die den Meßwert ergeben, wenn Rauschen vernachlässigt wird. Der Summand e_k schließlich beinhaltet das Rauschen, das die Meßwerte verfälscht. Ist ein Schätzwert $\hat{x}_k^-$ für x_k gegeben, so läßt sich aus dem aktuellen Meßergebnis ein neuer Schätzwert berechnen:

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \\ &= K_k' \hat{x}_k^- + K_k z_k, \text{ mit } K_k' = I - K_k H_k \end{aligned} \quad (4-18)$$

Hierbei symbolisiert das Minuszeichen von $\hat{x}_k^-$ den *a priori* Schätzwert zum Zeitpunkt t_k vor Berücksichtigung des Meßwertes. K_k' und K_k bezeichnen die Gewichtsmatrizen für den alten Schätzwert bzw. den aktuellen Meßwert. K_k wird basierend auf der Fehler-Kovarianzmatrix P_k berechnet und ist so gewählt, daß die relative Genauigkeit der Messung berücksichtigt wird und ein im statistischen Sinne optimaler Schätzwert berechnet wird. Damit ist ein Rekursionsschema definiert, das für fortlaufende Meßwerte einen Schätzwert für den nächst folgenden Meßwert liefert. (Für eine Berechnung von K_k und P_k siehe Gelb (74), für eine Implementierung der Kalman-Rekursion siehe <http://www.taygeta.com/kalman.html>.)

Bei der Vorhersage der Suchregionen für die Personenverfolgung hat sich der Kalman-Filter als nützlich erwiesen. Zusammen mit dem oben erwähnten Heuristiken ergibt sich oft ein stabiles System. Ebenso kann beispielsweise bei der Personenverfolgung die Zeigerichtung gefiltert werden, wodurch eine wesentlich bessere Hand-Cursor-Koordination beim Arbeiten vor der Projektionswand möglich ist. Die Verfolgung von Regionen im Bild durch Kalman-Filterung erfolgt im zweidimensionalen Kamerakoordinatensystem, d.h. es werden die x- und y-Koordinaten der Schwerpunkte gefiltert. Die Modellierung der Bewegung eines Punktes wird ebenfalls in diesem Koordinatensystem formuliert, obwohl eine Bewegung im Raum dreidimensionale Meßwerte hervorbringt. Eine bessere Vorhersage der Suchregionen für das nächste Bild kann erreicht werden, indem die Bewegungsbahnen im dreidimensionalen Koordinatensystem durch Kalman-Filterung verfolgt werden, also x-, y- und z-Koordinaten. Die Kalman-Rekursion liefert dabei einen Schätzwert für die 3D-Position des Punktes im nächsten Bild. Durch eine Projektion dieses Punktes auf die Bildebene kann eine neue Suchregion ermittelt werden. Erste Versu-

che zeigen intuitiv eine bessere Vorhersage der Region. Besonders bei modellbasierten Verfahren (siehe unten) ist dieses Verfahren anzuwenden, da es eine direkte Vorhersage der Suchregion aus den Modellparametern gestattet.

4.4.2 Verfolgen über Silhouette

Werden Differenzbilder zur Segmentierung verwendet, kann aus der größten zusammenhängenden Region die Konturlinie extrahiert werden. Anschließend lassen sich Kopf und Hände mit einfachen Annahmen über die Lage und unter Verwendung von Krümmungsmerkmalen entlang der Konturlinie extrahieren. Für ihre Position werden einfache Heuristiken, ähnlich den in Darrell (94) und Johnson (94) beschriebenen, verwendet, um den Suchprozeß zu vereinfachen und fehlende Daten zu ersetzen, falls diese nicht sicher segmentiert werden konnten. So wird davon ausgegangen, daß der Benutzer mit dem Gesicht zur Kamera schaut, wodurch die Hände auf der linken und rechten Seite der Kontur zu finden sind. Die Suchregionen für die Hände befinden sich am linken und rechten Rand des umschließenden Rechtecks der Kontur und werden auf 10% der Breite gesetzt (Abbildung 4-8). Können keine Krümmungsmerkmale in einer der Regionen detektiert werden, wird eine Ruhelage der Hand angenommen, d.h. sie wird auf 45% der Höhe der Kontur gesetzt. Wird die Hand gefunden, wird die Position ständig aktualisiert, solange der Abstand zur vorherigen Position nicht 30% der Konturbreite überschreitet.

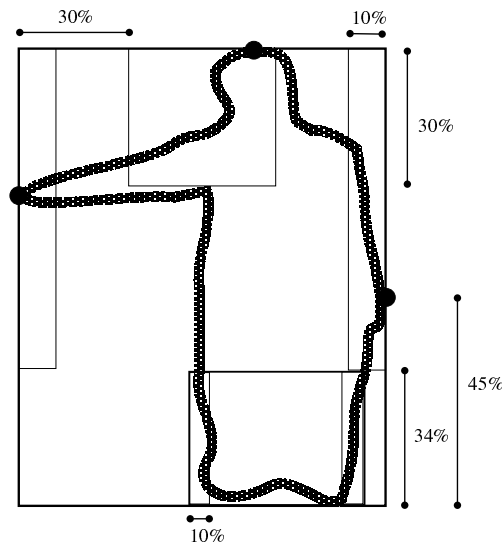


Abbildung 4-8: Silhouette einer Person mit Suchregionen und detektierten Händen und Kopf

Das gleiche Schema wird für den Kopf mit der in Abbildung 4-8 dargestellten Suchregion verwendet: Gegeben sei das umschließende Rechteck der Kontur $bbox = (min_x, min_y, max_x, max_y)$ mit Breite $width$ und Höhe $height$. Dann ergibt sich beispielsweise für den Kopf und ein gefundenes Krümmungsmerkmal $fc = (x_{fc}, y_{fc})$ folgende Regel:

$$head_found = \begin{cases} true, & \text{if } (bbox.min_x + 0.3 \cdot width < x_{fc} < bbox.max_x - 0.3 \cdot width) \\ & \text{and } (bbox.max_y - 0.3 \cdot height < y_{fc} < bbox.max_y) \\ false, & \text{else} \end{cases} \quad (4-19)$$

Für die Füße wird ein anderes umschließendes Rechteck gewählt, das 34% der unteren Kontur einschließt. Die Suchregionen werden analog der der Hände gesetzt, die Ruheposition ist hierbei in der unteren linken bzw. rechten Ecke der Kontur. Kann eine Position im aktuellen Bild durch Anwendung der oben beschriebenen Regeln nicht gefunden werden, wird die relative Lage innerhalb der Kontur des letzten Bildes verwendet, um die aktuelle Lage zu erhalten. Die relative Lage, angewandt auf die aktuellen Werte des umschließenden Rechtecks, ergibt dabei die neue Position. Kann eine Position in mehr als fünf hintereinanderfolgenden Bildern nicht gefunden werden, wird davon ausgegangen, daß das System die Hand verloren hat und die Hand wird zur Ruhelage durch Interpolation über die Zeit bewegt. Die Größe der Suchregionen und die Heuristiken wurden im Experiment ermittelt. Zusammen mit der Kalman-Filterung von Positionsdaten ergibt sich hieraus ein stabiles System indem die zu suchenden Körperteile analog zu Regel (4-19) sequentiell bestimmt werden. Im folgenden werden wir sehen, wie ähnliche Heuristiken in einem modellbasierten Verfahren verwendet werden, um eine stabilere Verfolgung zu erhalten.

4.5 Modellbasierte Verfahren

Die menschliche Bewegung und damit die zu segmentierenden Merkmale sind oftmals hochgradig komplex. Oft können Merkmale aufgrund von Verdeckung gar nicht segmentiert werden. In der Bildverarbeitung werden daher Modelle verwendet, deren Eigenschaften mit den Merkmalen verglichen werden. Sind in dem Bild die Merkmale des Modells oder einer bestimmten Modellkonfiguration wiederzufinden, wird auf das Vorhandensein des Objekts geschlossen. Für die menschliche Personenverfolgung sind verschiedene Personenmodelle bekannt. Sie unterscheiden sich dahingehend, ob die Positur der Person, ein bestimmter Bewegungszyklus, oder nur Teile der Person erkannt werden sollen. Auch unterscheiden sich die Verfahren je nachdem, ob sie zum Verfolgen oder *Labeling* (Markieren) verwendet werden. Strichmännchen und Volumenmodelle werden meist zur Verfolgung im Dreidimensionalen verwendet, zum Labeling bei Leung (94) werden nur Projektionen von Modellen verwendet. Für die Wahl eines Modells muß ein Kompromiß zwischen der Komplexität des Modells und der Genauigkeit der Erkennung eingegangen werden. Das Modell sollte jede gewünschte Konfiguration annehmen können und trotzdem einfach und schnell zu berechnen sein. Daher liegen die meisten Modelle im Bereich zwischen Strichmännchen (Akita 84) und Volumenmodellen (Hogg 84, Rohr 93). Im folgenden wird besonders auf die Einfachheit des Modells und seine schnelle Berechenbarkeit Wert gelegt.

Die in der Literatur bekannten Ansätze verfolgen sehr verschiedene Ziele: Zum einen werden vorgegebene Bewegungssequenzen erkannt (Ishii 93), so daß keine echte Verfolgung der realen Positur der Person möglich ist. Andere Verfahren konzentrieren sich eher auf die Erkennung von einigen Handgesten (Maggioni 95, Bröckl-Fox 95). Zusätzlich erscheinen einige bekannte Ansätze für die hier gestellten Anforderungen an Echtzeitfähigkeit nicht geeignet, z.B. (Lee 85, Chen 92 und Gavrilu 96). Ein vielversprechender Ansatz zur Personenverfolgung ist in Wren (96) zu finden, der jedoch seine Grenzen in der Verarbeitungsgeschwindigkeit von 10 Bildern pro Sekunde für ein Einzelkamerasystem hat und nur eine einzige Person als Benutzer zuläßt. Ein echtzeitfähiges linienbasiertes und modellbasiertes Verfahren zur Verfolgung eines Buches wird in Lowe (92) beschrieben. Hier wird davon ausgegangen, daß die Linienmerkmale im Bild sichtbar sind. Im Bereich „Active Vision“ sind Verfahren vorgestellt worden, die das Verfolgen von starren Körpern mit Hilfe von 3D-Modellen erlauben (Blake 92). Ein weiteres Verfahren zeigt das Verfolgen von flexiblen 3D-Objekten für spärliche, verrauschte Daten (Blake 92). Auch hierbei ist die Berechnung relativ aufwendig, da die Gesetzmäßigkeiten der nicht starren

Körper berücksichtigt werden. Weiterhin wird davon ausgegangen, daß genügend Merkmale im Bild vorhanden sind. Eine einfache Methode zur Bestimmung der Zeigerichtung wird in Fukumoto (94) beschrieben. Das System *Finger-Pointer* verwendet die Verbindungslinie zwischen einem als VPO (virtual pointing origin) genannten Punkt und der Spitze des Zeigefingers, um die Zeigerichtung zu bestimmen. Dazu muß das System auf den Benutzer kalibriert werden, jedoch sind anschließend keine Nutzerbewegungen im Raum erlaubt. Es wird zwar argumentiert, daß das System unabhängig von der Art des Zeigens sei, dies scheint sich aber auf die verschiedenen Vorlieben eines Benutzers zu beziehen, ob er mit ausgestrecktem oder eher angewinkeltem Arm auf die Projektion zeigt. Stets wird eine Zeigegestik angenommen, bei der die Fingerspitze in die Nähe der Sichtlinie geführt wird. Gestiken mit herabhängendem Arm werden so nicht erfaßt.

Für die Problematik der Personenverfolgung, insbesondere die Verfolgung der Positur im Raum und die Bestimmung der Zeigerichtung im Kontext der hier beschriebenen Anwendung, erscheint keines der genannten Verfahren direkt geeignet. Im folgenden werden zwei modellbasierte Verfahren zur Unterstützung der Bildverarbeitung auf Basis von Regionen entwickelt, die eine Verfolgung in einer komplexen Laborumgebung unter Anwesenheit mehrerer Personen erlauben. Die genannten Verfahren sollten hierbei auf einen Standard-PC in Echtzeit - genauer mindestens 12 Bilder pro Sekunde - ausführbar sein. Das erste, sehr einfache Verfahren zeigt hierbei das Prinzip der Kombination von Heuristiken mit einem modellbasierten Ansatz, kann jedoch aufgrund seiner Einfachheit nicht in jeder Situation befriedigende Ergebnisse liefern. Eine Erweiterung um eine flexible, erweiterbare Wissensbasis zur Einbindung von Heuristiken wird im zweiten Verfahren beschrieben.

4.5.1 Einfaches Armmodell

Aufgabe: Gegeben seien die 3D-Koordinaten einer Person, genauer die Körperposition $p_b = (x_b, y_b, z_b)$ und die Position der rechten Hand $p_h = (x_h, y_h, z_h)$, die aus dem Bild mittels einfacher Regionen-segmentierung bestimmt worden sind und daher anfällig gegenüber den im Abschnitt Verfolgung genannten Störungen sind. Zur Vereinfachung trägt die Person eine grüne Markierung, deren Farbe sonst nicht in der Szene vorkommt, die Handregion wird mittels Hautfarbe segmentiert. Weiterhin können andere Personen anwesend sein. Gesucht ist ein stabiles Verfahren, das durch den Einsatz von Heuristiken trotz dieser einfachen Segmentierung und Anwesenheit Dritter eine stabile Segmentierung der Person mit dem Marker und unter Verwendung eines Modells die Bestimmung der Zeigerichtung, d.h. des Schnittpunkts mit der xy-Ebene $p_p = (x_p, y_p, 0)$, erlaubt.

Besteht ein zu verfolgendes Objekt oder eine Person aus mehreren leicht zu segmentierenden Regionen, so lassen sich Heuristiken in Form von Nebenbedingungen angeben, die eine Suche vereinfachen und zu einem stabilen System führen, auch wenn mehrere ähnliche Regionen im Bild vorhanden sind. In Abbildung 4-9 ist dies für die Personenverfolgung mit Hilfe einer grünen Markierung dargestellt. Es wird davon ausgegangen, daß die Farbe der Markierung in der Szene sonst nicht vorkommt und somit eine stabile Segmentierung sowie eine Verfolgung mittels eines Kalman-Filters erreicht werden kann. Nachdem nun die Position der grünen Markierung $p_m = (x_m, y_m)$ im Bild bestimmt worden ist, wird die rechte Hand relativ zu dieser Position gesucht. Die Suchregion der hautfarbenen Region mit den Eckpunkten p_1, p_2 kann durch eine Menge von Nebenbe-

dingungen in Abhängigkeit von p_m und der Breite dx und Höhe dy der Markerregion definiert werden. Abbildung 4-9 zeigt die Position der grünen Markierung $p_m = (x_m, y_m)$ und die Suchregion für die hautfarbene Handregion. Weiterhin sind die Suchregionen für Hand- und Markerregionen dargestellt, die verwendet werden um die Regionen im nächsten Bild zu finden.

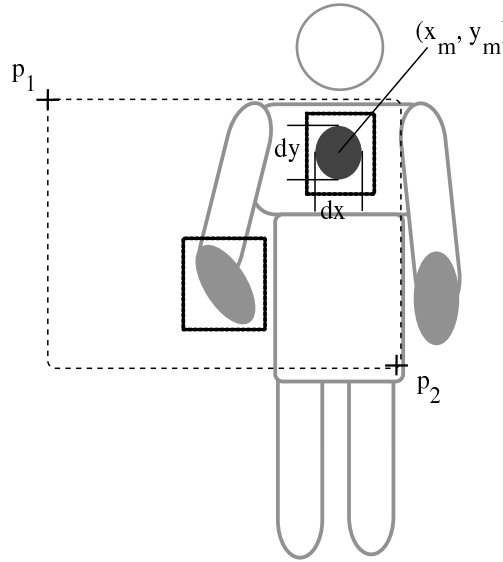


Abbildung 4-9: Schematische Ansicht einer Person mit Nebenbedingungen für Regionen

Die Suchregion kann durch die beiden Eckpunkte eines Rechtecks angegeben werden. So wird die obere linke Ecke der Suchregion beispielsweise definiert als x_m minus der sechsfachen Breite der Markerregion und y_m minus der 2,8-fachen Höhe der Markerregion. Für die rechte untere Ecke wird analog verfahren, so daß sich die Eckpunkte p_1, p_2 wie folgt ergeben:

$$p_1 = (x_b - 6 \cdot dx, y_b - 2.8 \cdot dy)$$

$$p_2 = (x_b + 2.5 \cdot dx, y_b - 5 \cdot dy)$$

Während der Verfolgung wird für die Handregion stets die Erfüllung der Nebenbedingungen geprüft. Aufgrund dieser einfachen Heuristik, ergänzt durch eine Kalman-Filterung der Suchregionen für Markerregion und Hand, ist es möglich die Person sicher zu segmentieren, auch wenn andere Personen anwesend sind oder hautfarbene Objekte im Raum vorhanden sind. Lediglich in der Initialisierungsphase dürfen keine hautfarbenen Objekt in der in Abbildung 4-9 gezeigten Suchregion vorhanden sein. Sofern sich Regionen während der Verfolgung überschneiden, z.B. durch eine Hand einer anderen Person, wird die Segmentierung jedoch gestört und kann erst wieder korrekt arbeiten, wenn die falsch detektierte Hand die Nebenbedingungen nicht erfüllt. Für Marker- und Handposition wird anschließend eine 3D-Rekonstruktion nach Abschnitt 4.3.4.2 durchgeführt, die Körperposition $p_b = (x_b, y_b, z_b)$ und Position der rechten Hand $p_h = (x_h, y_h, z_h)$ liefert.

Im nächsten Schritt wird für die Bestimmung der Zeigerichtung der verfolgten Person allein aus Hand- und Körperposition ein einfaches Armmodell, bestehend aus Hand,

Ellbogen und Schulter unter Anwendung von *Inverser Kinematik* verwendet. Das Lösen der inversen Kinematik ist keine einfache Aufgabe, da die zugrundeliegende Funktion nicht linear ist und im allgemeinen keine eindeutige Zuordnung zur Lösung des Problems existiert. Allerdings kann für eine einfache zweigliedrige Kette, bestehend aus Ober- und Unterarm, eine geschlossene Lösung angegeben werden. Zunächst seien die Position der Hand $p_h = (x_h, y_h, z_h)$ und eine Näherung für die der Schulter $p_s = (x_s, y_s, z_s)$ in Abhängigkeit der Körperposition sowie die Längen für Ober- und Unterarm l_1, l_2 durch folgende Definition gegeben (Angaben in Metern):

$$p_s = \begin{pmatrix} x_b + 0.2 \\ 1.46 \\ z_b \end{pmatrix}$$

$$l_1 = l_2 = 0.3$$

Als erstes soll die Ellbogenposition bestimmt werden, aus der anschließend die Zeigerichtung berechnet wird. Die Ellbogenposition p_e ist eindeutig definiert durch die Angabe von l_1 und den zwei Winkeln θ_1 und γ (Rotation des Oberarms um die x- bzw. z-Achse), sowie den Rotationswinkeln α, β der Handposition (vgl. Abbildung 4-10). Diese Darstellungsweise wird gewählt, da sich die Winkel α und β einfach aus den 3D-Koordinaten der Handposition bestimmen lassen. Damit erhalten wir die Ellbogenposition im Schulterkoordinatensystem. Hand- und Schulterposition liefern jedoch keine eindeutige Lösung für die gesuchten Winkel θ_1 und γ . Mit Hilfe einer einfachen Heuristik zur Steuerung des Winkels γ (Drehung des Ellbogens um die Achse Schulter-Hand, siehe Abbildung 4-10), wird das Problem zweidimensional und damit lösbar (Abbildung 4-11).

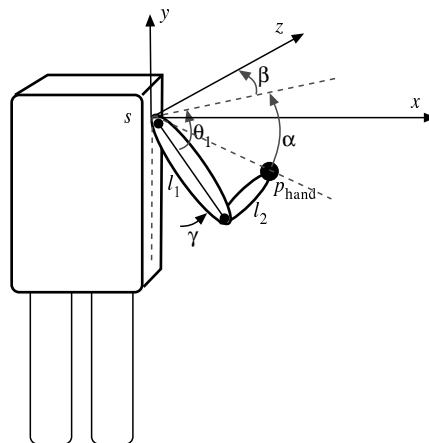


Abbildung 4-10: Einfaches Armmodell: Ober- und Unterarmlänge (l_1, l_2) sind vorgegeben, die Schulterposition (s) wird relativ zu Körperposition berechnet.

Durch einfache Annahmen über die Armhaltung des Benutzers beim Ausführen einer Zeigoperation, kann dieser Winkel automatisch bestimmt werden. Unter Annahme, daß eine Drehung des Ellbogens zur Decke einem Winkel γ von 0 Grad entspricht und eine Drehung gerade nach unten 180 Grad, kann folgende Beobachtung gemacht werden: Zeigt der Benutzer zur, von ihm aus gesehenen, rechten Seite, wird der Arm näher am Körper gehalten (ca. 175 Grad), während er beim Zeigen zur linken Seite um etwa 45 Grad angehoben wird, was zu einem Winkel von 130 Grad führt. Steht der Benutzer mit

Blickrichtung zu den Kameras, gibt die x-Position der Hand relativ zur Schulterposition an, ob nach links oder nach rechts gezeigt wird. Dieser Wert wird verwendet, um den Winkel zu setzen:

$$\gamma = 135 + 40 \cdot \frac{(x_h - x_s + 0.35)}{0.7}$$

Zur Bestimmung von θ_1 und θ_2 wird zunächst der Vektor der Handposition p_h durch Drehung um die y-Achse (Winkel β) in die yz-Ebene rotiert. Es ergibt sich die in Abbildung 4-11 gezeigte Situation. Anschließend wird die Lösung für die Gliederelemente des Modells berechnet.

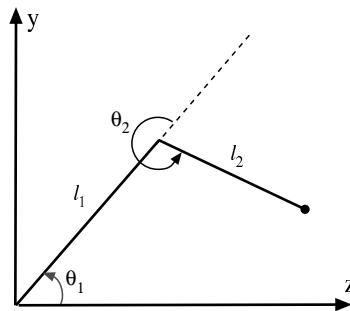


Abbildung 4-11: Einfache zweigliedrige Kette

Ist der Positionsvektor einer zweigliedrigen Kette $X = (y, z)$ gegeben, liefert die inverse Kinematik die Lösung für:

$$\Theta = f^{-1}(X)$$

Hierbei bezeichnet $\Theta = (\theta_1, \theta_2)$ die Winkel der beiden Glieder nach Abbildung 4-11. Aus den Seitenverhältnissen sowie durch einfache trigonometrische Umformungen aus der Vorwärtsgleichung

$$X = (l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2), l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2))$$

und Auflösen nach θ_1 ergibt sich die Lösung zu (Watt 92):

$$\theta_2 = 180^\circ + \arccos \frac{(z^2 + y^2 - l_1^2 - l_2^2)}{-2l_1 l_2}$$

$$\theta_1 = \arctan \left(\frac{-(l_2 \sin \theta_2)z + (l_1 + l_2 \cos \theta_2)y}{(l_2 \sin \theta_2)y + (l_1 + l_2 \cos \theta_2)z} \right)$$

Nun wird der Oberarm-Vektor $v = (0, 0, l_1)$ um die x-, z-, und y-Achse um θ_1 , γ und die zuvor bestimmten x- und y-Rotationswinkel gedreht:

$$p_e = R_y(\beta) \cdot R_x(\alpha) \cdot R_z(\gamma) \cdot R_x(\theta_1 - \alpha) \cdot \begin{pmatrix} 0 \\ 0 \\ l_1 \\ 1 \end{pmatrix}$$

Trotz dieser recht einfachen Annahme über den Winkel γ , ergibt sich eine korrekte Zeige-
geposition auf der Projektionswand. Weiterhin kann eine natürlichere Armbewegung des
Modells beobachtet werden. Wird der Winkel γ auf einen festen Wert gesetzt, führt dies
zu einer falschen Zeigerichtung und das Modell weist ein eher steifes Verhalten auf. Mit
diesen Ergebnissen kann ein Verfolgungsverfahren angegeben werden:

```
PROCEDURE track_person
// definiere person als Ding der Realwelt
thing person

// ergänze Objekt um Segmentierungsinfo und Modellinfo
person->add_primitive(marker)
person->add_primitive(hand)
person->set_model(simple_arm_model)

//process scene
while (true)
  FOR ALL primitive = Primitive von person
    primitive->segment()

    // bestimme Position mittels 3D-Rekonstruktion
    person->get_position()

    // bestimme Zeigerichtung mittels Armmodell
    person->model->get_pointing_position()

    // sage Position von Objekt im nächsten Bild voraus
    FOR ALL primitive = Primitive von person
      primitive->update_search_box()
    ENDWHILE
  END
```

Das in diesem Abschnitt beschriebene einfache Armmodell liefert für bestimmte Situa-
tionen befriedigende Ergebnisse, d.h. es erlaubt eine Zeigeoperation innerhalb des Rau-
mes nur, wenn der Benutzer in etwa parallel zur Projektionswand steht, also auf diese
blickt. Im Zuge einer Interaktion kommt es häufig vor, daß sich der Benutzer etwas
schräg stellt, um beispielsweise einen Gesprächspartner anzublicken, oder sich sogar
umdreht bzw. kurz die Aufmerksamkeit auf einen Schreibtisch richtet, um anschließend
in gewohnter Weise die Zeigeoperation fortzusetzen. In jedem dieser Fälle, verliert das
Verfolgungssystem, aufgrund unvollständiger Daten, kurzzeitig die Person, und muß neu
mit einer Initialisierung der Verfolgung beginnen. Dies hat zur Folge, daß der auf der
Projektion dargestellte Cursor verschwindet und erneut wieder erscheint, wenn die neuen
Daten gültig sind. Für den Benutzer hat dies ein „Springen“ zur Folge, eine visuelle Dis-
kontinuität. Ein weiteres Problem ist die Bildverarbeitung selbst, die aufgrund vieler
Faktoren fehlerhafte Daten liefern kann, die die gleichen Effekte hervorrufen, wie die
eben beschriebene Benutzerinteraktion: Dazu gehören das Teilen eines Blobs aufgrund
von Beleuchtungsschwankungen, das anschließende „Zusammenfallen“ eines Blobs auf-
grund von Abschattungen, das „Verschwinden“ eines Blobs aufgrund von Verdeckungen,
das Auffinden eines falschen Blobs aufgrund falscher Zuordnung oder der Anwesenheit
Dritter.

4.5.2 Körpermodell

Aufgabe: Gegeben seien, wie im letzten Abschnitt, die 3D-Koordinaten einer Person, genauer die Körperposition $p_b = (x_b, y_b, z_b)$ und die Position der rechten Hand $p_h = (x_h, y_h, z_h)$ (sowie analog der linken). Neben den Problemen der letzten Aufgabe, ist es zusätzlich erlaubt, daß die Person sich dreht, die Markierung durch eine Hand verdeckt oder eine Hand hinter dem Körper und damit unsichtbar für die Kamera plazierte wird. Gesucht ist ein stabiles Verfahren, das durch den Einsatz von Heuristiken und eines Modells eine stabile und kontinuierliche Verfolgung der Person mit dem Marker erlaubt.

Zur Lösung der am Ende des letzten Abschnitts angesprochenen Probleme reicht das einfache Armmodell nicht aus. Daher wird in diesem Abschnitt ein Körpermodell beschrieben, das es gestattet, die aktuellen Daten der Bildverarbeitung mit Nebenbedingungen des Modells zu kombinieren. Aus der Bildverarbeitung fließen hierbei die aktuellen Positionen von Händen und Körper (so diese ermittelt werden konnten) und die Sichtbarkeit des grünen Markers, genauer die Sichtbarkeit in beiden Kamerabildern, ein. Das Modell gibt zum einen durch seine Struktur und die Begrenzung der Auslenkung der Gliederelemente Nebenbedingungen vor, die durch die Angabe von eher wahrscheinlichen oder „angenehmeren“ Körperhaltungen ergänzt werden. Die passende Kombination dieser Variablen könnte durch Methoden der nichtlinearen Programmierung (NLP), Optimierungsverfahren wie Newton-Verfahren oder Pseudoinverse realisiert werden, die jedoch zum Teil aufgrund der Komplexität und damit der nicht Echtzeitfähigkeit ausscheiden. Konkrete Lösungen finden sich in Badler (93), wo der Begriff „comfort“ geprägt wird, der die Gliederstellungen eines menschlichen Modells beeinflusst, das einen Gegenstand heben muß. Hierbei wird aus den Nebenbedingungen eine Zielfunktion minimiert. Eine andere Möglichkeit ist die Bestimmung einer Bewegungsrichtung aus einer Summe von Kraftvektoren (jede Nebenbedingung würde einen Kraftvektor erzeugen), wobei die Aktualisierung durch eine Euler-Gleichung realisiert wird (Fleischmann 94-2). Die Kombination mehrerer gewichteter Aktionen zur Animation von computergenerierten Charakteren wird in Perlin (96) beschrieben. Hier werden vordefinierte Bewegungen interpoliert, um ein kontinuierliches Verhalten zu erhalten. Im folgenden wird eine Fuzzy-basierte Wissensbasis in Kombination mit einem Set von Regeln und Aktionen vorgeschlagen, wobei die Summation der „fuzzy-ähnlichen“-Regeln als eine Summe von Kraftvektoren aufgefaßt werden kann, die mit Aktionen, ähnlich wie den in Perlin (96) beschriebenen, kombiniert werden. Das hier verwendete Modell wird anhand eines Körpers mit zwei Armen und Beinen in Form von Gelenkketten beschrieben.

4.5.2.1 Kinematisches Modell

Das Modell besteht aus vier Gelenkketten, den Armen und den Beinen, die an einem Körper angebracht sind. Die aktuelle Konfiguration wird durch die Position des Körpers und der Stellung der Arme durch Angabe der Längen und Gelenkwinkel angegeben. Die Stellung der Beine wird durch die detektierte Körperhöhe bestimmt. Für die Darstellung der Kinematik beschränken wir uns im folgenden auf den rechten Arm. Die anderen Glieder werden analog behandelt.

Die Formulierung einer Vorwärtstransformation, d.h. die Bestimmung des Endeffektors Hand aufgrund der Position des Körpers und bestimmter Gelenkwinkel erfolgt analog der Darstellung lokaler Koordinatentransformationen in Foley (91), bzw. von Koordinatensystemen in einer Hierarchie von Gelenken (Watt 92), wobei hier eine etwas andere Darstellungsweise als beim einfachen Armmodell gewählt wurde (Abbildung 4-12). Die Winkel sind jeweils auf die lokalen Koordinatensysteme der Gliederkette bezogen. Die

verwendeten Transformationsmatrizen entsprechen der Notation aus dem Abschnitt 3D-Rekonstruktion in diesem Kapitel.

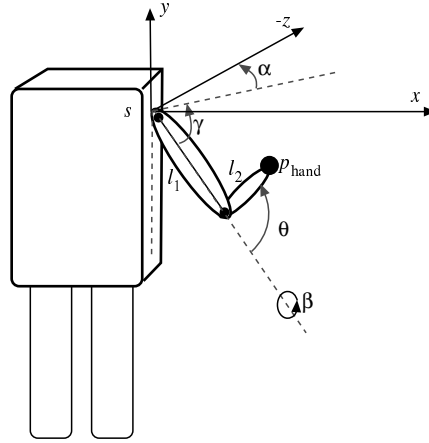


Abbildung 4-12: Winkelbeziehungen des Körpermodells exemplarisch für den rechten Arm

Vorwärtstransformation. Ausgehend von einem rechtsorientierten Koordinatensystem und einer Lage des Oberarms entlang der z-Achse, mit α als Rotation des Oberarms um die y-Achse, also die horizontale Richtung des Oberarms, und γ als Rotation um die x-Achse (vertikale Richtung), sowie β und θ analog für eine Rotation des Unterarms um die z- und x-Achse, ergibt sich die Bestimmung der Handposition im Weltkoordinatensystem wie folgt:

$$P_{hand_w} = M_{s \rightarrow w} R_y(\alpha) \cdot R_x(\gamma) \cdot T_z(l_1) \cdot R_z(\beta) \cdot R_x(\theta) \cdot \begin{pmatrix} 0 \\ 0 \\ l_2 \end{pmatrix} = M_{s \rightarrow w} M_1 \vec{l}_2 \quad (4-20)$$

wobei l_1 und l_2 die Längen von Ober- und Unterarm angeben und $M_{s \rightarrow w}$ die Transformation von Schulterkoordinaten zu Weltkoordinaten angibt. M_1 faßt die Rotations-Translationsmatrizen zusammen. Ohne Berücksichtigung der Transformation ins Weltkoordinatensystem, ergibt sich für die Komponenten der Handposition folgende Beziehung:

$$P_{hand} = \begin{pmatrix} x_{hand} \\ y_{hand} \\ z_{hand} \end{pmatrix} = \begin{pmatrix} A \cdot l_2 + (\sin(\alpha) \cos(\gamma)) l_1 \\ B \cdot l_2 - \sin(\gamma) l_1 \\ C \cdot l_2 + (\cos(\alpha) \cos(\gamma)) l_1 \end{pmatrix} \quad (4-21)$$

mit

$$A = \cos(\alpha) \sin(\beta) \sin(\theta) - \sin(\alpha) \sin(\gamma) \cos(\beta) \sin(\theta) + \sin(\alpha) \cos(\gamma) \cos(\theta)$$

$$B = -\cos(\gamma) \cos(\beta) \sin(\theta) - \sin(\gamma) \cos(\theta)$$

$$C = -\sin(\alpha) \sin(\beta) \sin(\theta) - \cos(\alpha) \sin(\gamma) \cos(\beta) \sin(\theta) + \cos(\alpha) \cos(\gamma) \cos(\theta)$$

Rückwärtstransformation. Das Ziel der Rückwärtstransformation ist die Bestimmung der Gelenkwinkel von Ober- und Unterarm aus der Position des Endeffektors Hand. Analog der Erläuterungen beim einfachen Armmodell im letzten Abschnitt kann hier mit Hilfe einer Heuristik für einen der Gelenkwinkel eine geschlossene Lösung angegeben werden, indem die Gleichung der Vorwärtstransformation nach den gesuchten Größen aufgelöst wird, d.h. die Komponenten der Matrix M_1^{-1} in folgender Gleichung bestimmt werden:

$$\vec{l}_2 = M_1^{-1} p_{hand}$$

Weiterhin werden der Rotationswinkel des Oberarms in x-Richtung, γ , durch eine Heuristik vorgegeben und der Winkel zwischen Ober- und Unterarm, θ , mit Hilfe des Cosinussatzes aus den Längenverhältnissen des Dreiecks (l_1, l_2, L) bestimmt. Dabei gibt $L = |p_{shoulder} - p_{hand}|$ die Länge zwischen Hand und Schulter an. Dadurch ergibt sich aus Gleichung 4-21 durch Auflösen auch β :

$$\gamma = \arcsin\left(\frac{|y_{shoulder} - y_{hand}|}{\|p_{hand}\|}\right), \quad \theta = \arccos\left(\frac{l_1^2 + l_2^2 - L^2}{2l_1l_2}\right) - 180^\circ$$

$$\beta = -\arccos\left(\frac{\frac{-y_{hand} - \sin\gamma \cdot l_1}{l_2} - \sin\gamma \cdot \cos\theta}{\cos\gamma \cdot \sin\theta}\right)$$

Aus Gleichung (4-21) ergibt sich für die x- und z-Koordinate der Handposition folgende Beziehung:

$$x_{hand} = A \cdot \sin\alpha + B \cdot \cos\alpha$$

$$z_{hand} = -B \cdot \sin\alpha + A \cdot \cos\alpha$$

mit

$$A = \cos\gamma \cdot l_1 - \sin\gamma \cdot \cos\beta \cdot \sin\theta \cdot l_2$$

$$B = \sin\beta \cdot \sin\theta \cdot l_2$$

Hieraus läßt sich schließlich auch α bestimmen:

$$\alpha = \begin{cases} \arccos\left(\frac{A \cdot z_{hand} + B \cdot x_{hand}}{A^2 + B^2}\right), & \text{wenn } x_{hand} > z_{hand} \\ -\arccos\left(\frac{A \cdot z_{hand} + B \cdot x_{hand}}{A^2 + B^2}\right) + 360^\circ, & \text{sonst} \end{cases}$$

4.5.2.2 Wissensbasis und Aktionen

Das hier verwendete Körpermodell soll mit wenigen, auch unvollständigen Informationen aus der Bildverarbeitung gesteuert werden, um die Positionen von Marker, rechter und linker Hand im Bild verfolgen zu können und weiterhin stets „brauchbare“ Positionsdaten dem aktualisierten Modell entnehmen zu können. Folgende Probleme sind dabei von dem Modell zu lösen:

Initialisierung. In dieser Phase sind die zu verfolgenden Regionen im Kamerabild zu finden, bei dem die Konfiguration des Modells verwendet wird, um die am Anfang dieses Abschnitts erwähnten Heuristiken zu ersetzen.

Verfolgung. In der Initialisierungsphase als auch bei der Verfolgung können mehr als ein Merkmal in der Suchregion gefunden werden. Hier besteht die Aufgabe darin, die wahrscheinlichste Region für die weitere Verfolgung auszuwählen und damit Mehrdeutigkeiten aufzulösen.

Verdeckungen. Während der Verfolgung können die Daten aufgrund von Verdeckungen unvollständig werden, beispielsweise, wenn der Marker durch eine Zeigeoperation des Benutzers oder die Hand durch den Körper verdeckt wird. Trotz dieser fehlenden Daten soll eine kontinuierliche Interaktion ermöglicht werden. Dazu reichen einfache Heuristiken im 2D-Kamerabild nicht aus, sondern es muß je nach aktueller 3D-Konfiguration des Modells, d.h. Körperhaltung des Benutzers im Raum, ein gezieltes Auffüllen der fehlerhaften Daten und eine sinnvolle Vorhersage erfolgen, damit die Verfolgung bei Wiederauftreten des Merkmals fortgesetzt werden kann.

Datenverlust. Während der Verfolgung können Daten auch durch das Verschwinden eines verfolgten Merkmals unvollständig werden, obwohl das zugehörige Objekt im Kamerabild sichtbar ist. Auch in diesem Fall liegen für bestimmte Regionen keine Daten vor und es muß ein Auffüllen der Daten und eine Vorhersage unterstützt werden. Dieses tritt beispielsweise auf, wenn sich eine Region durch Abschattungen in zwei Regionen teilt, von denen eine nach kurzer Zeit vollständig verschwindet. Hat sich die Verfolgungssoftware für die kleinere Region entschieden, ist das Merkmal in diesem Moment verloren. Dies führt in der Regel zu einer erneuten Initialisierungsphase.

Ereignisse. Für bestimmte Konfigurationen kann es hilfreich sein, Aktionen auszulösen, die als vollständige Handlungsabläufe unabhängig von der Bildverarbeitung durchgeführt werden. Derartige Aktionen können beispielsweise die Ausrichtung und Position des Modells jeweils dann anpassen, wenn die Daten der Bildverarbeitung einen Schwellwert überschreiten oder eine Situation erkannt worden ist, die eine Aktion zum weiteren Verfolgen notwendig macht.

Die hier verwendete Wissensbasis löst diese Probleme durch eine Bewertung von Regionen zum Auflösen von Mehrdeutigkeiten, eine sinnvolle Fortführung von Bewegungen und eine Vorhersage von möglichen Konfigurationen des Benutzers in Form von einem Regelsatz. Die einzelnen Regeln werden jeweils durch eine Funktion angegeben, so daß in einem bestimmten Bereich eine kontinuierliche Bewertung der Regel vorliegt. In Anlehnung an die Fuzzy-Set-Theorie werden diese Funktionen hier als „fuzzy-ähnliche“-Funktionen bezeichnet. Sie wurden auf intuitive Weise im Experiment bestimmt und repräsentieren gewissermaßen das Expertenwissen des Entwicklers. Im folgenden wird kurz auf die Einbettung des Regelwerks in die Fuzzy-Set-Theorie und anschließend auf den verwendeten Regelsatz eingegangen.

4.5.2.3 Fuzzy-Inferenzsysteme

Die Fuzzy-Set-Theorie wurde von Zadeh (65) als Erweiterung klassischer zweiwertiger Mengen eingeführt (hierzu sowie für die folgenden Darstellungen siehe Tizhoosh 98). Eine klassische (scharfe) Menge A einer Grundmenge X kann durch die Auflistung ihrer Elemente

$$A = \{a, b, c, \dots\}$$

oder durch Angabe einer Eigenschaft Δ ihrer Elemente

$$A = \{x \mid x \text{ besitzt die Eigenschaft } \Delta\}$$

angegeben werden. Eine Fuzzy-Menge wird durch ihre Zugehörigkeitsfunktion μ_A charakterisiert, die jedem Element x der Grundmenge X eine reelle Zahl aus dem Intervall $[0, 1]$ zuordnet. Die Menge A wird dann als Menge von Paaren definiert:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

Die Zugehörigkeit berücksichtigt eine gewisse Unschärfe von Eigenschaften bzw. Aussagen und liefert damit die Grundlage einer Verbindung der komplexen, überwiegend sprachlichen Kommunikationswelt des Menschen und der abstrakten Welt der Mathematik und Logik (Tizhoosh 98). Als Beispiel ist in Abbildung 4-13 die Menge der "dunklen Grauwerte" als klassische und auch als Fuzzy-Menge angegeben.

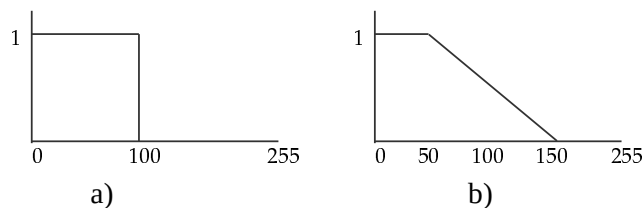


Abbildung 4-13: Menge der dunklen Grauwerte angegeben durch a) eine klassische (scharfe) Menge und durch b) eine Fuzzy-Menge.

Die in dieser Arbeit verwendete Methode lehnt sich an das Konzept der Fuzzy-Inferenzsysteme an. Im folgenden wird auf die Unterschiede bzw. Vereinfachungen des hier verwendeten Systems eingegangen (für Details der Fuzzy-Inferenzsysteme siehe Tizhoosh 98). Ein Fuzzy-Inferenzsystem arbeitet nach Tizhoosh (98) in drei Stufen:

1) Codierung der Eingänge (Fuzzyfizierung). Es wird für jeden Eingang die Zugehörigkeit zu allen Termen der entsprechenden Grundmenge ermittelt. Für das Beispiel der Grauwerte ergeben sich beispielsweise die in Abbildung 4-14a dargestellten Funktionen.

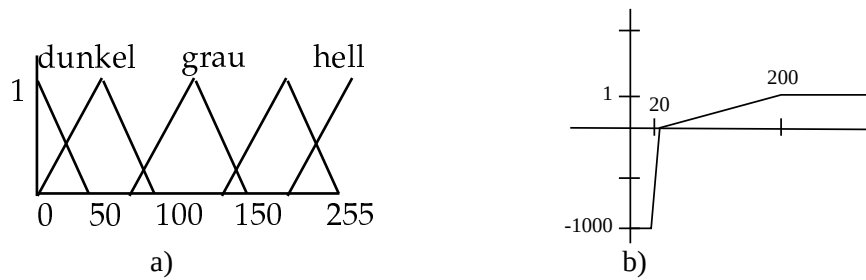


Abbildung 4-14: a) Codierung von Grauwerten durch Zugehörigkeitsfunktionen, b) Beispiel einer „fuzzy-ähnlichen“-Funktion. Die x-Achse entspricht dem Abstand einer Region vom Kopf des Modells in cm, die y-Achse ergibt die Bewertung Regel

Für das hier verwendete Regelwerk werden Zugehörigkeitsfunktionen zugelassen, die nicht auf den Wertebereich $[0, 1]$ beschränkt sind, sie werden daher als „fuzzy-ähnliche“-Funktionen bezeichnet. Abbildung 4-14b zeigt ein Beispiel einer Funktion, die eine Handregion der Bildverarbeitung in Abhängigkeit vom Abstand des Kopfes des Modells bewertet und entfernte Regionen bevorzugt. Der stark negative Wert der Funktion deutet an, daß es sehr unwahrscheinlich ist, daß eine Handregion in der Nähe des Kopfs gefunden wird. Diese Regel verhindert eine fälschliche Erkennung der Kopfregion als Handregion. Der nicht beschränkte Wertebereich gestattet es hier, eine sprachliche Formulierung auf intuitive Art und Weise zu codieren, mit dem Nachteil, daß hier die Fuzzy-Set-Theorie "verlassen" wird. Ein Vorteil dieser Methode ist die leichtere Handhabbarkeit bei der Einstellung der Grenzen. Würden nur Funktionen aus dem Intervall $[0, 1]$ zugelassen werden, müßte obige Regel u.U. in mehrere Regeln aufgeteilt werden.

2) Schlußfolgerung (Inferenz). Liegt das Expertenwissen in der Regelbasis in Form von *Wenn-Dann-Regeln* vor, die ihrerseits aus Teilbedingungen zusammengesetzt sein können, so werden diese in den Schritten *Aggregation* (Erfüllungsgrade von Teilprämissen), *Implikation* (Erfüllungsgrade der Konklusionen) und *Akkumulation* (der Teilergebnisse) verknüpft (Für eine Darstellung der Teilschritte siehe Tizhoosh 98). Das Ergebnis der Inferenz ist eine zusammengesetzte Fuzzy-Menge, die beispielsweise eine Form wie in Abbildung 4-15a dargestellt haben kann.

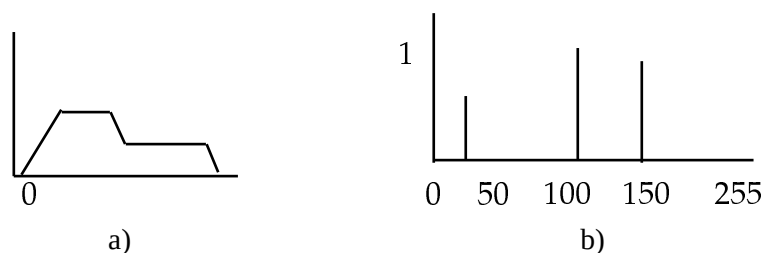


Abbildung 4-15: Ergebnis der Inferenz: a) zusammengesetzte Fuzzy-Menge, b) Singletons als Darstellung der Ausgangsmenge

Zur Beschleunigung der anschließenden Defuzzifizierung können zur Darstellung der Ausgangsmengen auch sogenannte Singletons (Fuzzy-Mengen mit nur einem Stützpunkt) verwendet werden (Abbildung 4-15b). In dem in dieser Arbeit beschriebenen System wird diese Methode verwendet: Jede Teilregel liefert hier direkt ein Singleton, somit beschränkt sich die Inferenz auf die Ermittlung der Zugehörigkeiten.

3) Decodierung (Defuzzifizierung). Da das Ergebnis der Inferenz eine Fuzzy-Menge ist, wird in diesem Schritt ein scharfer, konkreter Zahlenwert ermittelt, der die unscharfe Ausgangsmenge am besten repräsentiert. Im Falle einer zusammengesetzten Fuzzy-Menge kann der Schwerpunkt der Fläche berechnet werden. Für eine diskrete Grundmenge ist die Berechnung in Abbildung 4-16a angegeben. Dabei bezeichnet $\mu(z)$ die Zugehörigkeitsfunktion des Inferenzergebnisses in den Grenzen a und b .

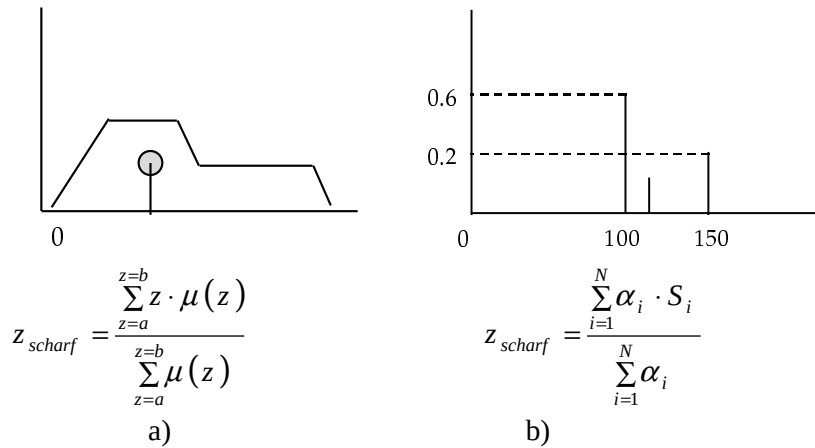


Abbildung 4-16: Decodierung des Ausgangs. a) COA-Methode (Center of Area),
b) COA-Methode für Singletons

Im Falle von Singletons vereinfacht sich diese Methode auf die Summierung der Singletons (Abbildung 4-16b). S_i bezeichnet die Singletons der Regeln R_i mit dem Erfüllungsgrad α_i . Durch die Einstellung des Gewichtungsfaktors α_i kann das System in Bezug auf ein bestimmtes Verhalten hin optimiert werden.

Damit ergibt sich ein effektives Verfahren für die Berechnung der Ausgangswerte aus den "fuzzy-ähnlichen"-Funktionen. Im folgenden werden auf dieser Basis nicht nur eine Bewertung von Regionen durchgeführt, sondern auch Winkelgeschwindigkeiten bestimmt und eine Auswahl von Aktionen getroffen.

4.5.2.4 Regelwerk

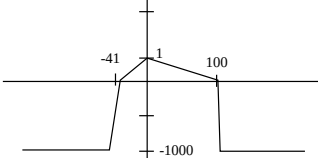
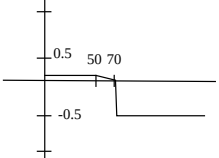
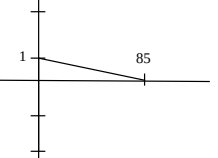
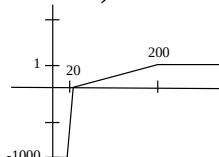
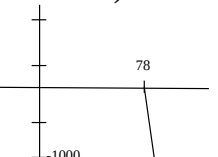
Im folgenden wird der Ablauf der Verfolgung mit den verwendeten Regeln beschrieben: Ist die Markerposition gefunden, werden in der Initialisierungsphase für die linke und rechte Hand jeweils eine große Suchregion um eine Ruheposition neben dem Körper verwendet. Diese Regionen werden auch periodisch verwendet, um ein Wiederaufsetzen des Modells zu ermöglichen, wenn eine fehlerhafte Verfolgung durchgeführt wurde. Aus der Suchregion wird mittels der in Tabelle 4-1 aufgeführten Regeln die jeweils beste Region aus der Menge der detektierten gewählt, indem diese Regionen durch Auswertung der Regeln bewertet werden und anschließend diejenige mit dem größten Wert verwendet wird. Die Werte der einzelnen Teilregeln (a, b, c etc.) werden dazu analog Abbildung 4-16b summiert. Dieses führt zu einer Mischung ihres Einflusses. Eine Gewichtung erlaubt die Verstärkung oder Abschwächung jeder einzelnen Regel, je nach Anwendungsfall oder gewünschtem Systemverhalten:

$$y = \frac{\sum_{i=1}^n (g_i \cdot R_i(x_i))}{\sum_{i=1}^n g_i} \quad (4-22)$$

Für die Regeln R_i mit Gewichtungsfaktoren g_i und den Variablen x_i wird das Ergebnis y berechnet. Der Wertebereich der Regeln und damit auch der Wertebereich des Ergebnisses, der für die weitere Auswertung berücksichtigt wird, entspricht in dieser Darstellung aus Gründen der Anschauung jeweils dem Bildbereich. Für das Beispiel der rechten Hand liegt er zwischen 0 und 1 und gibt damit eine Bewertung der Regionen vor (ein Wert größer 1 ist in diesem Fall zugelassen und unkritisch). Für die weiter unten beschriebenen Regeln zur Bestimmung der Standrichtung liefert das Ergebnis einen Winkel zwischen -90 und 180 Grad und für den Oberkörper eine Winkelgeschwindigkeit.

In Tabelle 4-1 ist für den Marker eine Regel angegeben, die eine aufrechte Haltung des Modells bevorzugt, sowie bestimmte Grenzen vorgibt. Der Wert von -1000 ist hierbei willkürlich gewählt und bezeichnet eine sehr unwahrscheinliche Region, die nicht weiter berücksichtigt wird (Ablehnung). Für eine Hand fließen die Abstände zwischen einer detektierten Region und a) der Hand des Modells, b) der Zentralposition, c) des Kopfes und d) der Schulter in die Berechnung ein.

Tabelle 4-1: Regeln zur Auswahl der wahrscheinlichsten Region (Regel für linke Hand analog)

Primitiv	Parameter	Funktion
Marker	Abstand Standardhöhe (140 cm) Einheiten: X : Abstand in cm Y : Bewertung (-1000 = Ablehnung)	
rechte Hand	a) Abstand vom Modell b) Abstand Zentralposition c) Abstand Kopf d) Abstand Schulter Einheiten: X : Abstand in cm Y : Bewertung (-1000 = Ablehnung)	   

Zur Aktualisierung des Modells während der Verfolgung und zur Fortführung von Bewegungen bei Verdeckungen oder Datenverlust werden für Oberkörper und Oberarm Winkelgeschwindigkeiten berechnet, die zur Bestimmung der neuen Position verwendet werden. So die Bildverarbeitung Daten liefert, wird der Oberkörper gemäß den ersten beiden Regeln in Tabelle 4-2 aktualisiert. Der Standrichtungs-Offset der ersten Regel fließt hierbei in Regel c) für den Oberkörper ein und dreht beispielsweise den Oberkörper stärker

nach links, wenn keine Daten im rechten Kamerabild vorliegen. Die Standrichtung selbst, d.h. die globale Ausrichtung des Modells, wird durch die unten beschriebenen Aktionen gesteuert. Die Armposition ergibt sich aus der Handposition mittels des kinematischen Modells. Bei Datenverlust oder Verdeckung wird die Armhaltung durch die Vorwärtstransformation nach Gleichung 4-20 ermittelt, wobei der Winkel α des Oberarms durch die in Tabelle 4-2 aufgeführten Regeln aktualisiert wird. Erneut werden die Werte der Teilregeln nach Gleichung (4-22) summiert. Durch Anwendung der Regel wird der Winkel des Oberarms der Rotation des Oberkörpers angepaßt und in Richtung der Verdeckung fortgeführt. Die Regel c) für den Oberarm schließlich, verhindert ein Überdrehen des Arms.

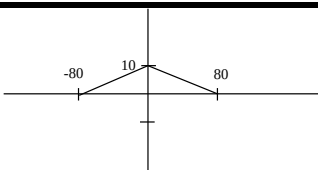
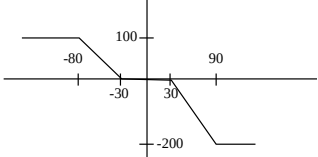
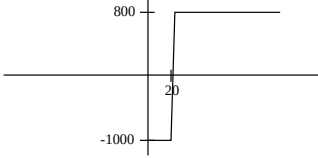
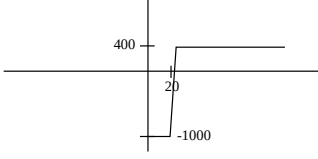
Tabelle 4-2: Regeln zur Berechnung der Winkelgeschwindigkeiten von Oberkörper und rechtem Oberarm (links analog)

Körperteil	Parameter	Funktion
Standrichtungs-Offset	Sichtbarkeit des Markers Einheiten: X : 0 = nicht sichtbar 1 = rechte Kamera 2 = linke Kamera 3 = beide Kameras Y : Winkel in Grad	
Oberkörper	a) Rotation Torso b) Oberarmwinkel rechts (links analog) c) Standrichtung + Rotation Torso + Standrichtungs-Offset Einheiten: X : Winkel in Grad Y : Winkelgeschwindigkeit in Grad/sek	
Oberarm rechts	a) Geschwindigkeit Rotation Torso X : Winkelgeschwindigkeit b) Verdeckung X : Bewertung Verdeckung [0, 1] c) Oberarmrotation X : Winkel in Grad Einheiten: X : siehe a), b) und c) Y : Winkelgeschwindigkeit in Grad/sek	

Für bestimmte Kombinationen von Werten bzw. bestimmte Modellkonfigurationen können Aktionen ausgelöst werden. Aktionen sind vollständige Handlungsabläufe, wie etwa eine Körperdrehung, die unabhängig von der Bildverarbeitung durchgeführt werden. Es sind also Animationen des Modells oder von Teilen des Modells, die es erleichtern sollen

eine kontinuierliche Verfolgung zu ermöglichen. Diese Aktionen sind ähnlich denen, die in Perlin (96) beschrieben werden, sie werden hier jedoch mit der Bildverarbeitung verschränkt eingesetzt, um diese zu unterstützen. Tabelle 4-3 zeigt die Regeln für die verschiedenen Aktionen. So wird beispielsweise eine Drehung des Modells nach rechts dann ausgelöst, wenn der Oberkörper des Modells sich stark in diese Richtung dreht (negativer Winkel) oder eine Gehbewegung, falls die Differenz in der horizontalen Ebene von Modellposition und aktueller Markerposition eine Schwelle überschreitet (20 cm). Nach Berechnung aller Regeln wird die wahrscheinlichste Aktion aktiviert. Aus Tabelle 4-3 ist zu entnehmen, daß der Zustand *Neutral* nur angenommen wird, wenn keine der anderen Regeln zutrifft (schwache Bewertung der Regel). Ebenso wird der Zustand *Gehen* sofort ausgeführt, falls die Abweichung von der Position mehr als 20 cm beträgt.

Tabelle 4-3: Aktionsauslösende Regeln (Linksrotation analog)

Aktion	Parameter	Funktion
Neutral	Oberkörperrotation Einheiten: X : Winkel in Grad Y : Bewertung	
Rechtsdrehung	Oberkörperrotation Einheiten: X : Winkel in Grad Y : Bewertung	
Gehen	Offset Position horizontal Einheiten: X : Abstand in cm Y : Bewertung	
Kniebeuge	Differenz der Position zur Standardhöhe Einheiten: X : Abstand in cm Y : Bewertung	

Durch Anwendung der oben aufgeführten Regeln für die Winkelgeschwindigkeiten bewegt beispielsweise das Modell die Hand hinter den Körper, falls eine Verdeckung und vorherige seitliche Körperhaltung anzeigen, daß dieses Verhalten für den Benutzer sehr wahrscheinlich ist. Dreht sich die Person anschließend weiter, so kann durch das Modell die Hand leichter wiedergefunden werden, wenn sie auf der anderen Körperseite im Kamerabild erscheint. Ein Beispiel für eine Aktion ist das Gehen des Modells in eine bestimmte Richtung, angezeigt durch die 3D-Koordinate des Markers. Diese Aktion liefert ein seinem Verhalten nach natürlicher aussehendes Modell. Ebenso wird in bestimmten Situationen eine Körperdrehung durchgeführt. Weitere Beispiele und Details dieser Technik finden sich in Tillmann (98), eine Evaluation für eine Reihe von Testfällen findet sich am Ende dieses Kapitels.

Unter Verwendung dieses Modells erhalten wir ein modifiziertes Verfolgungsverfahren:

```
PROCEDURE track_person
// definiere person als Ding der Realwelt
thing person

// ergänze Objekt um Segmentierungsinfo und Modellinfo
person->add_primitive(marker)
person->add_primitive(hand)
person->set_model(body_model)

//process scene
while (true)
  FOR ALL primitive = Primitive von person
    primitive->segment()

    // aktualisiere Modell unterstützt durch Wissensbasis
    person->model->update()

    // bestimme Position und Zeigerichtung mittels Modell
    person->model->get_position()
    person->model->get_pointing_position()

    // sage Position von Objekt mittels Modell voraus
    person->model->predict()

  ENDWHILE
END
```

Der Aktualisierung des Modells in obigen Verfolgungsverfahren liegt eine dreistufige Struktur zugrunde: Die 3D-Positionen werden durch *Kalman-Filter* geglättet, eine Aktualisierung des Modells geschieht mittels *Winkelgeschwindigkeiten* und bei bestimmten Konstellationen werden *Aktionen* ausgelöst. Damit stellt sich die Funktion *update()* des Modells wie folgt dar:

```
PROCEDURE update(model)
// wähle wahrscheinlichste Region
FOR ALL part = Part von model
  part->get_best_primitive()
  part->kalman_filter_position()

// Berechnung aktueller Aktion
updateActions()

// Berechnung der Winkelgeschwindigkeiten
updateVelocities()

// neue Konfiguration
update_model()
END
```

4.6 Einbeziehung realer Gegenstände

Durch das Einbeziehen von Realgegenständen ist es möglich, virtuelle Objekte im Realraum durch aktives Bewegen von Platzhalterfiguren zu bewegen. Dadurch ist eine direkte Beurteilung der Position im 3D-Raum und der Lage von Objekten relativ zu anderen möglich. Die Objekte werden hierbei durch aktive Körperbewegung und Kraft bewegt, durch das Gewicht der Platzhalterfiguren ist zusätzlich ein implizites Force-Feedback gegeben. Die Szene kann so im Realraum arrangiert und diskutiert werden, und auf der Projektionswand kontrolliert werden (siehe hierzu Abbildung 6-8, Seite 120). Eine Ver-

folgung dieser Objekte beschränkt sich auf die 3D-Position von zwei verschiedenen Platzhalterfiguren. Durch eine farbliche Kennzeichnung der Figuren mit einer gelben und einer roten Markierung und der Annahme, daß diese Farben sonst nicht in der Szene vorkommen, gestaltet sich das Verfolgungsverfahren sehr einfach. Folgender Pseudocode implementiert das Verfahren:

```
PROCEDURE track_figure(color)
// definiere figure als Ding der Realwelt
thing figure

// ergänze Objekt um Segmentierungsinfo
figure->add_primitives(color)
primitive = Primitiv von figure

//process scene
while (true)
  // segmentiere Region im Bild
  primitive->segment()

  // bestimme Position durch 3D-Rekonstruktion
  figure->get_position()

  // sage Position von Objekt im nächsten Bild voraus
  primitive->update_search_box()

ENDWHILE
END
```

Eine weitere Verwendungsmöglichkeit der Realfiguren besteht im hinteren Teil des Raumes, wo die Realobjekte als modifizierbare Markierungen für Orte und Funktionen verwendet werden. Beim Prototyp zur Filmplanung werden sie verwendet, um die Orte für das Hinzufügen bestimmter Objekte der Requisite zu markieren. So kann der Benutzer den Ort, an dem sich beispielsweise Stühle befinden, in der dafür vorgesehenen sensitiven Zone im Raum variieren. Auch hier wird die 3D-Position der Figuren wie oben bestimmt und hierdurch die Positionen der Requisite vorgegeben. Die Requisite ihrerseits wird dann durch die Position des Benutzers aufgerufen. Die Idee hierbei ist, die Verwendung einer sichtbaren, realen Markierung von virtuellen Objekten (hier Requisiten).

4.7 Leistungsdaten

Zur Beurteilung der Erkennungsleistung der Personenverfolgung ist zum Einen die Genauigkeit zu bestimmen, d.h. die Genauigkeit der 3D-Rekonstruktion und der Zeigegeistik, zum Anderen die erzielte Geschwindigkeit sowie die Zuverlässigkeit bei der Verfolgung. Zu berücksichtigen ist, daß bei der Auswahl der Verfahren, wie Eingangs erwähnt, ein Kompromiß zwischen der Leistungsfähigkeit und der Echtzeitfähigkeit des Systems gefunden werden mußte. Die im folgenden aufgeführten Meßwerte müssen also auch unter diesem Gesichtspunkt und unter Berücksichtigung des Gesamtsystems bewertet werden (Für genaue Aufstellung der verwendeten Hardware siehe Tabelle 6-1, Seite 116).

4.7.1 3D-Rekonstruktion

Tabelle 4-3 zeigt die mit der Bildverarbeitung erzielte Genauigkeit der 3D-Rekonstruktion. Berechnungsgrundlage ist ein YUV-Bild mit halber PAL-Größe, d.h. eine Auflösung von 192x144 Pixeln für jeweils den U- und den V-Kanal (die Auflösung des Y-Kanals beträgt 384x288). Die absolute Genauigkeit von 9 cm ist für das Anwendungsszenario durch die visuelle Rückmeldung durch den Cursor auf der Projektionswand ausreichend. Ähnlich ist es bei der Hand-Auge Koordination mit Maus und Maus-

zeiger, bzw. Cursor auf dem Bildschirm, bei der die *Auflösung*, d.h. die kleinste meßbare Änderung, wesentlich wichtiger ist als eine absolute Positionsgenauigkeit. Die Auflösung des Systems beträgt 2 cm. Die Genauigkeit wurde an 30 verschiedenen Positionen im Raum in einer Entfernung zwischen 2,70 und 5 Metern gemessen. Tabelle 4-4 zeigt hierbei jeweils den schlechtesten Fall.

Tabelle 4-4: Genauigkeit der 3D-Rekonstruktion

absolute 3D-Position	9 cm
Auflösung	2 cm

4.7.2 Zeigeoperation

Zum Test der Zeigeoperation mittels Stab und mittels Armmodell wurde auf der Projektionswand jeweils eine Anwendung mit Cursor dargestellt. Die Cursorbewegung wurde durch die aus der Bildverarbeitung bestimmte Zeigerichtung vorgegeben. Die Abmessungen der Projektionswand betrugen 2 mal 2,60 Meter, und der Benutzer arbeitete in einer Entfernung zwischen 3 und 4 Metern. Zur Bestimmung der Genauigkeit wurde ein Laserpointer auf dem Stab so montiert, daß er die Zeigerichtung durch einen Punkt auf der Projektionswand markiert. Für das Armmodell wurde der Stab unterhalb des Unterarms montiert, so daß der Stab mit der Zeigerichtung des Unterarms zusammenfällt. Die Genauigkeit wurde an vier Stellen im Bereich der Ecken und an zwei Stellen im Bereich der Mitte der Projektionswand gemessen. Der Wert für die Abweichung, d.h. das Springen des Cursors aufgrund des einfachen Segmentierungsverfahrens, wurde hierbei abgeschätzt.

Tabelle 4-5: Genauigkeit der Zeigeoperation mit Stab

Zeigeposition (Regelfall)	5 cm
Zeigeposition (alle Fälle)	10 cm
Abweichung (Springen)	10 cm

Tabelle 4-6: Genauigkeit der Zeigeoperation mit Armmodell

Zeigeposition (Regelfall)	10 cm
Zeigeposition (alle Fälle)	20 cm
Abweichung (Springen)	15 cm

Für das Armmodell liegt die Genauigkeit der Zeigeposition meist innerhalb von 10 cm (Regelfall), nur selten ist eine Abweichung von 20 cm zu beobachten. Die stärksten Probleme mit dem System entstanden bei Stab sowie Armmodell aufgrund eines Springens der Zeigerichtung. Dies liegt an dem einfachen Segmentierungsverfahren, das nicht alle verschiedenen Schattensituationen ausgleichen kann und beispielsweise sensitiv gegenüber einer Handbewegung reagiert.

4.7.3 Geschwindigkeit der Verfolgung

Die erzielte Bildrate des Bildverarbeitungssystems, zusammen mit der Spracherkennung und der Anwendung zur Filmplanung, auf einem 200 MHz Pentium/Pro PC mit zwei Matrox Meteor Grabber-Karten beträgt 12 Bilder pro Sekunde (Tabelle 4-7). Wird ein zweiter PC für das Applikationsprogramm verwendet, so beträgt die Bildrate 20 Bilder für eine Nutzerverfolgung und 16 Bilder pro Sekunde für eine Nutzerverfolgung und zwei

reale Platzhalterfiguren, die durch je eine farbige Markierung gekennzeichnet sind und mit halber Bildrate verfolgt werden.

Tabelle 4-7: Geschwindigkeit des Bildverarbeitungssystems und des Gesamtsystems (200 MHz Pentium/Pro)

Personenverfolgung	20 fps
Personenverfolgung + 2 Platzhalterfiguren (halbe Bildrate)	16 fps
Verfolgung + Spracherkennung + Anwendungsprogramm	12 fps

Die Tabelle zeigt, daß die in Kapitel 3 geforderte unmittelbare Rückkoppelung des Systems mit einer Aktualisierungsrate von nicht unter 0,2 Sekunden sogar dann erreicht wird, wenn Bildverarbeitung, Spracherkennung und Anwendungsprogramm auf einem einzigen Standard-PC ausgeführt werden. Eine Lösung unter der Verwendung von zwei PC verbessert die Aktualisierungsrate nochmals und ist auch aufgrund der Systemstabilität (Windows) vorzuziehen.⁶ Weiterhin gestattet ein zweiter PC auch die visuelle Kontrolle der Bildverarbeitung auf einem separaten Monitor, was sich besonders in der Entwicklungsphase als vorteilhaft herausgestellt hat.

4.7.4 Zuverlässigkeit

Für das vorgestellte Szenario der Intuitiven Schnittstelle ist die Zuverlässigkeit für eine Benutzerakzeptanz des Systems entscheidend. Für die prototypische Anwendung sind hierbei die sichere Erkennung der Zeigegestik und eine kontinuierliche Verfolgung von Bedeutung. Eine sichere Erkennung bedeutet, daß der Cursor auf der Projektionswand in Zeigerichtung sichtbar sein muß, wenn der Benutzer auf diese zeigt. Eine kontinuierliche Verfolgung bedeutet, daß der Cursor nach Möglichkeit keine plötzlichen größeren Abweichungen aufweisen oder gar ganz von der Projektionswand verschwinden sollte. Eine quantitative Bestimmung der Zuverlässigkeit ist im vorliegenden Fall nicht einfach, da entweder die Komplexität der möglichen Rahmenbedingungen (Benutzerposition und -größe, Kleidung, Beleuchtung etc.) zu groß für eine Messung mit vertretbarem Aufwand wäre oder die Rahmenbedingungen derart eingeschränkt werden müßten, daß die Aussagekraft der gemessenen Größen nicht auf eine größere Gesamtheit extrapoliert werden kann und damit speziell für das Szenario dieser Arbeit unbrauchbar wäre. Aus diesem Grund wurde hier eine qualitative Beurteilung der Zuverlässigkeit anhand von Testfällen durchgeführt, die sich beim Umgang mit dem System einerseits als typisch und andererseits als kritisch herausgestellt haben. Die Testfälle zeigen Interaktionsmuster, wie sie typischerweise bei einer Bedienung auftreten. Für alle Testfälle arbeitet der Benutzer in einer Entfernung zwischen 3 und 4 Metern im Bereich der Mitte vor der Projektion:

⁶ Besonders die Kombination eines Grafikprogramms mit einem kontinuierlichen Bildeinzug unter Verwendung des Matrox-Treibers führt öfter zu einem Abbruch oder Ausfall des Systems.

Tabelle 4-8: Testfälle für die Bestimmung der Zuverlässigkeit

Nr.	Testfall	Erläuterung
1	Gerade Körperhaltung	Der Benutzer steht mit seiner Schulterlinie parallel zur Projektionswand, eine Zeigegestik wird in Richtung der Projektionswand ausgeführt ohne den Oberarm stark zu bewegen, d.h. er verbleibt etwa in seiner Ruheposition.
2	Schräge Körperhaltung	Wie Testfall 1, jedoch mit schräger Körperhaltung, Marker nur in einem Kamerabild sichtbar. Hierzu stellt sich der Benutzer zunächst gerade, und dreht sich anschließend ca. 45 Grad in eine Richtung.
3	Anwesenheit Dritter im Aktionsraum	Bei gerader Körperhaltung (Testfall 1) nähert sich eine Person dem Benutzer bis auf minimal 1 Meter.
4	Anwesenheit Dritter im Hintergrund	Bei gerader Körperhaltung (Testfall 1) befindet sich eine Person oder eine hautfarbene Region im Hintergrund.
5	Drehung	Aus einer schrägen Körperhaltung (Testfall 2) vollzieht der Benutzer eine vollständige Drehung.
6	Fehlende Daten	Hier wird davon ausgegangen, daß während der Verfolgung im Testfall 1 die Bildverarbeitung die Hand verliert und über mehrere Bilder (ca. 10 frames) keine Daten liefert.
7	Marker verdeckt	Aus gerader Körperhaltung (Testfall 1) wird der Arm so angehoben, daß die Hand den Marker in einem Kamerabild verdeckt.
8	Hand verdeckt	Aus gerader Körperhaltung (Testfall 1) wird die Zeigehand hinter den Körper geführt.

Tabelle 4-9: Verhalten der Verfahren für die Testfälle

Verfahren	1	2	3	4	5	6	7	8
einfaches Armmodell	+	-	o ^a	o ^a	-	-	-	-
Körpermodell	+	+	+	+	o ^b	o ^c	+	+

- + Verfolgung erfolgreich
- Verfolgung nicht erfolgreich, Cursor „springt“ weg oder zu fehlerhafter Position
- o Verfolgung erfolgreich jedoch nicht in jedem Fall:
 - a) Aufgrund der großen Suchregion kommt es häufiger zu Fehlern, wenn dritte Personen nahe stehen oder sich im Hintergrund bewegen.
 - b) Hier ist ein Kompromiß zwischen stabiler Zeigegestik und der Erkennung von Drehungen im Regelwerk der Wissensbasis notwendig.
 - c) Durch die langsame Annahme einer Ruheposition wird ein Springen reduziert und gleichzeitig ein Wiederfinden der Hand erleichtert.

Tabelle 4-9 gibt einen Überblick über die Zuverlässigkeit auch bei kritischen Testfällen. Hierbei bedeutet eine erfolgreiche Verfolgung (+) nicht in jedem Fall perfekte Ergebnisse, je nach Umgebung, Beleuchtung und Kleidung des Benutzers können die Ergebnisse natürlich auch schlechter ausfallen. Für die vorgestellten Testfälle und die Testsituation sind diese Ergebnisse jedoch nachvollziehbar und eine qualitative Aussage ist daher

möglich. Insbesondere wird die Zuverlässigkeit des Körpermodells im Vergleich zu dem einfachen Armmodell deutlich. Trotz der Unzulänglichkeiten des einfachen Armmodells ist eine stabile Verfolgung auch bei Anwesenheit Dritter möglich, vorausgesetzt die folgenden Punkte sind erfüllt:

- In der Initialisierungsphase der Personenverfolgung sind keine hautfarbenen Regionen in der Suchregion für die Hand vorhanden.
- Während der Verfolgung gibt es keine Überschneidung von hautfarbenen Regionen mit der verfolgten Hand.
- Beim Verlieren der Hand (z.B. durch Verdeckung) befindet sich keine hautfarbene Region in der Suchregion für die Hand.

Sind diese Punkte nicht erfüllt, so können sie durch Positionswechsel des Benutzers relativ zu den dritten Personen behoben werden.

5 Systemarchitektur

Den Kern der Intuitiven Schnittstelle bildet ein Musteranalysesystem für die Rauminteraktion im Zusammenspiel mit der realen Umgebung und dem Anwendungsprogramm. In diesem Kapitel wird, ausgehend von einer allgemeinen Struktur von Musteranalysesystemen, die Systemarchitektur des Gesamtsystems vorgestellt. Abweichungen oder Vereinfachungen des Bildverarbeitungssystems von dieser allgemeinen Struktur, die aufgrund des Anwendungsszenarios und der geforderten Echtzeitfähigkeit des Systems notwendig waren, werden verdeutlicht. Im Abschnitt *Objektorientierter Entwurf* werden die Klassenhierarchien und Abhängigkeiten erläutert. Das Kapitel schließt mit einer Darstellung der Erweiterungsmöglichkeiten des Systems durch den Systemprogrammierer.

5.1 Allgemeine Struktur eines Musteranalysesystems

Das zu lösende Problem bei der Musteranalyse, besteht darin, entsprechend dem aktuellen Problemkreis die beste Beschreibung des Eingangssignals zu finden, d.h. eine Interpretation der Segmentierungsergebnisse. In der Bildverarbeitung wird dazu nach einer Vorverarbeitungsphase das Signal in der Regel in aussagekräftige Teile segmentiert. Hierzu sind eine Vielzahl von Segmentierungsverfahren bekannt, die das Bild in sogenannte Segmentierungsobjekte zerlegen (etwa Linien oder Regionen). Diese Objekte werden mit den Modellen einer Wissensbasis des Systems verglichen, in der bestimmte Annahmen über die möglichen Szenen des Problemkreises gespeichert sind. Wird der Datenfluß bei der Musteranalyse betrachtet, ergeben sich die in Abbildung 5-1 dargestellten Komponenten (Paulus 94). Das Problem, die beste Übereinstimmung zu den Modellen und eine gute Segmentierung zu erreichen, kann als Optimierungsproblem angesehen werden. Hierbei muß zwischen dem Aufwand für die Berechnung einer optimalen Übereinstimmung und der Verarbeitungsgeschwindigkeit abgewogen werden, die besonders bei Echtzeitanwendungen kritisch ist.

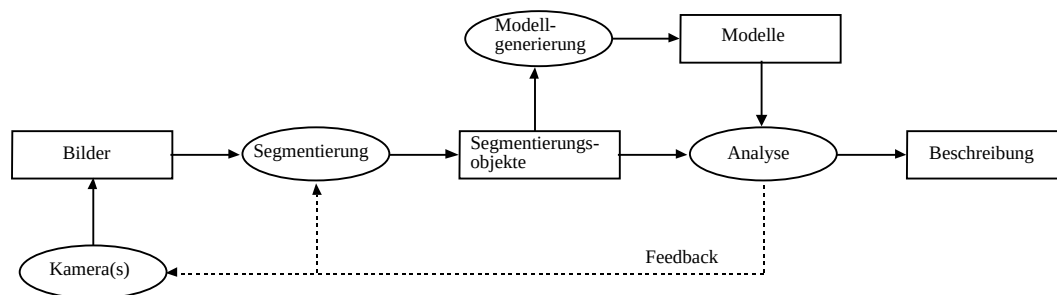


Abbildung 5-1: Datenfluß in einem Bildanalyse-System nach Paulus (94) (Bildverarbeitungsschritte sind in ovalen Boxen, Daten in Rechtecken dargestellt)

Für eine wissensbasierte Musteranalyse reicht das einfache Datenflußmodell nicht aus. Die Architektur, die im Kern den meisten Musteranalysesystemen zugrunde liegt, ist in Abbildung 5-2 dargestellt (Niemann 90). Sie besteht aus verschiedenen Modulen: Im Modul *Methoden* werden die Funktionen und Verfahren zur Vorverarbeitung, Extraktion von Merkmalen und zur Segmentierung des Bildes zusammengefaßt. In der Regel sollten diese Verfahren problemunabhängig sein und kein explizit repräsentiertes Wissen über den Problemkreis erfordern (Sagerer 90). Explizit repräsentiertes Wissen und anwen-

dungsabhängige Verfahren werden in einem separaten Modul *Wissen* angegeben. Die Methoden können von dem Modul *Kontrolle* beeinflusst werden, das die Steuerung der Verfahren durch das Wissen des Systems durchführt. Die Aufgabe besteht darin, aus dem aktuellen Zustand der Analyse und der Information aus dem Modul *Wissen*, neue Informationen zu generieren und Funktionen der Module Methoden oder Wissen zu aktivieren. In der *Ergebnisdatenbank* werden die Zwischen- und Endergebnisse während der Analyse abgelegt. Im Modul *Lernen* wird beschrieben, wie die Wissensbasis automatisch angereichert werden kann. Im Modul *Erklärung* befindet sich ggf. eine Komponente zur Interpretation der Endergebnisse für den Benutzer (Lernen und Erklärung sind optionale Module).

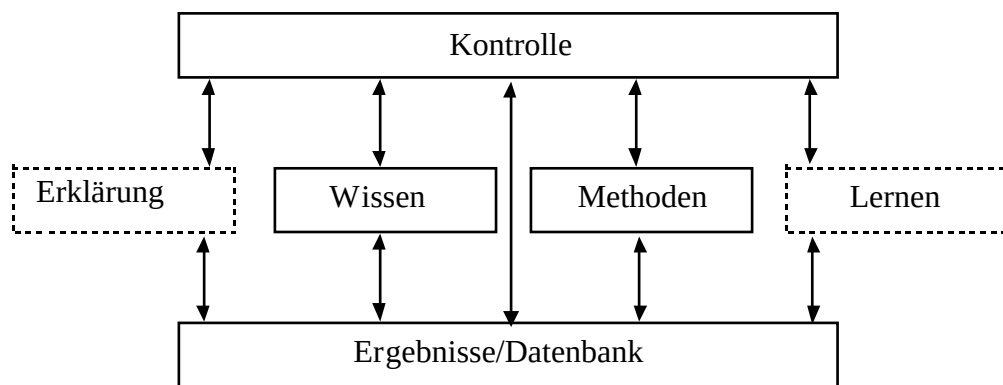


Abbildung 5-2: Struktur von Musteranalysesystemen nach Niemann (90)

Viele der beschriebenen Komponenten sind sowohl in kommerziellen als auch Public-Domain Bildverarbeitungssystemen wiederzufinden. Der Vorteil einer derartigen Strukturierung und Modularisierung liegt in der Allgemeingültigkeit eines Ansatzes zur Bildverarbeitung. Durch die Verwendung von objektorientierten Programmiersprachen lassen sich die einzelnen Module, sowie Daten und Methoden in Klassenhierarchien gruppieren, durch die das Gesamtsystem überschaubar bleibt und Einzelteile leichter wiederverwendbar sind. Die Hauptnachteile eines allgemeingültigen Systems zur Bildverarbeitung sind der interne Verwaltungsaufwand für die Aktualisierung der Datenstrukturen und die notwendige Kommunikation zwischen den Modulen. Dadurch sind besonders bei Anwendungen der Echtzeitverarbeitung Grenzen in der Bildrate gesetzt. Eine Übersicht über verschiedene Bildverarbeitungssysteme findet sich in Paulus (92/94). Nur einige der vorgestellten Systeme sind echtzeitfähig bzw. bedingt echtzeitfähig. Bei der Implementierung auf Standard-PCs kann ein systembedingter Verwaltungsaufwand die Interaktivität eines System stark herabsetzen, so daß eine Anwendung im Bereich Mensch-Maschine-Kommunikation nicht möglich ist (minimale Bildrate 10 Bilder pro Sekunde). Bei der Verwendung von Wissen wird oft versucht, eine Lösung der Aufgabenstellung zu finden, die unabhängig von der Anwendung zu korrekten Ergebnissen führt. Auch hier kann die notwendigerweise komplexere Struktur der Verfahren zu längeren Verarbeitungszeiten führen.

Im folgenden wird ein System vorgestellt, das speziell für eine Interaktion im Raum und für eine Echtzeitverarbeitung entwickelt wurde. Besonderen Wert wurde dabei nicht auf die Einhaltung einer Struktur wie in Abbildung 5-2 dargestellt gelegt, auch wenn einzelne Komponenten sich in dem System wiederfinden. Die Kriterien bei der Entwicklung des Systems waren Einfachheit, Echtzeitfähigkeit und eine einfache Erweiterbarkeit durch den Benutzer.

5.2 Systemüberblick

Dieser Abschnitt gibt einen Überblick über das Gesamtsystem für die Rauminteraktion im Zusammenspiel mit der realen Umgebung und dem Anwendungsprogramm. Das System ist unterteilt in einen Musteranalyse-Teil, bestehend aus dem Bildverarbeitungssystem und einer Spracherkennungssoftware, ein Serverprogramm und ein sogenanntes Frontend mit dem Anwendungsprogramm (Abbildung 5-3). Für die Spracherkennung wird ein kommerzielles, PC-basiertes System verwendet, um einfache Kommandos zu erkennen (Lernout & Hauspie 96). Eine Objektdatenbank mit einer für die Anwendung passenden Auswahl von virtuellen Objekten ist vom Anwendungsprogramm zur Verfügung zu stellen und kann ggf. auch in das Anwendungsprogramm integriert werden.

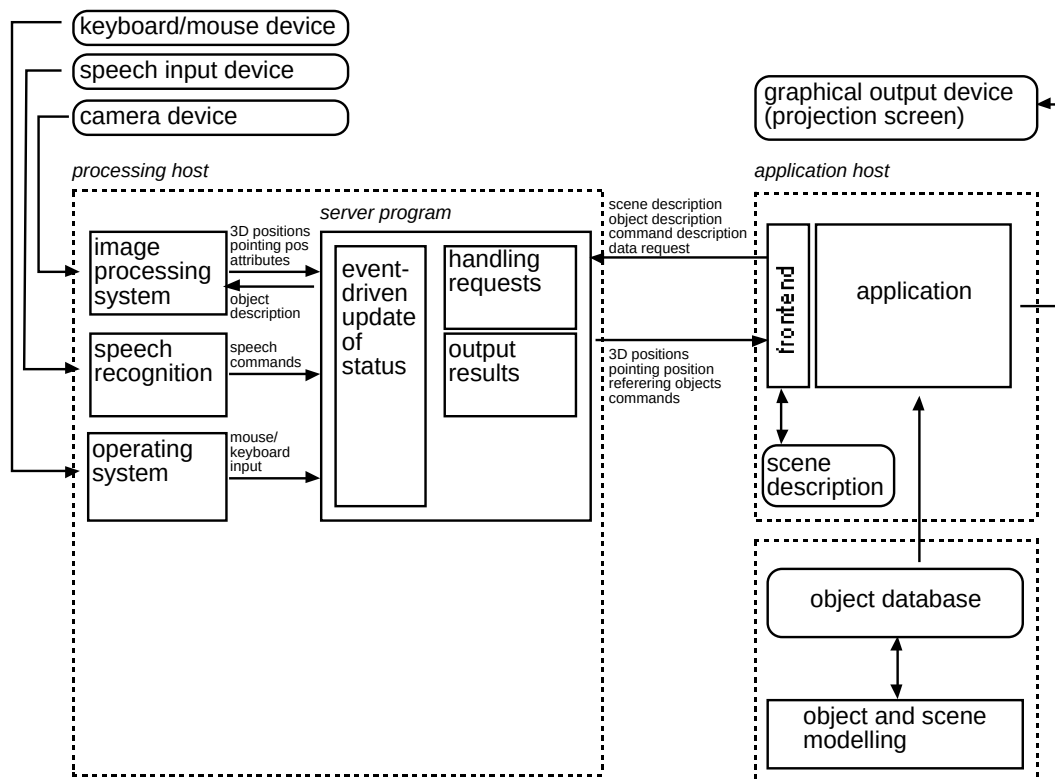


Abbildung 5-3: Systemarchitektur

Bildverarbeitungssystem. Das Bildverarbeitungssystem verfolgt den Benutzer über ein oder zwei Videokameras. In einer Initialisierungsphase erhält es eine Beschreibung der im Kamerabild zu verfolgenden (realen) Objekte über das Serverprogramm. Anschließend werden kontinuierlich die Segmentierungsergebnisse (z.B. 3D-Positionsdaten des Benutzers und die Zeigerichtung) an das Serverprogramm übermittelt. Die Segmentierung basiert auf Farbbildverarbeitung und Blobanalyse und enthält die im Kapitel *Personenverfolgung* erläuterten Verfahren. Aufgrund der Erweiterungsmöglichkeiten ist eine Verarbeitung auf Basis von Linien oder anderen Merkmalen möglich.

Serverprogramm. Das Serverprogramm verbindet die Anwendung mit dem Bildverarbeitungssystem, der Spracherkennung und ggf. Benutzereingaben über Maus und Tastatur. Der aktuelle Zustand der zu übertragenden Werte wird in einer nachrichtenorientierten Verarbeitung aktualisiert.

Frontend. Das Frontend ist eine Bibliothek von Methoden für die Kommunikation mit dem Serverprogramm und auch für grundlegende Interaktionsmuster der Intuitiven Schnittstelle. In der Initialisierungsphase wird sowohl eine Beschreibung der zu verfolgenden Objekte, als auch eine Beschreibung der Abhängigkeiten mit Objekten der virtuellen Szene des Anwendungsprogramms eingelesen. Dadurch können Verbindungen zwischen den Segmentierungsergebnissen und Aktionen bzw. Modifikationen in der virtuellen Szene definiert werden. Die Szenenbeschreibung muß von der Anwendung in einer Textdatei zur Verfügung gestellt werden.

Anwendungsprogramm. Das Anwendungsprogramm implementiert die Methoden für die zu lösende Aufgabe und ist für eine ggf. grafische Repräsentation virtueller Objekte verantwortlich. Es benutzt die Szenenbeschreibung und das Frontend um mit dem Serverprogramm zu kommunizieren. Das Anwendungsprogramm kann hierbei auf einem anderen Rechner als das Serverprogramm gestartet werden.

5.2.1 Datenfluß

Abweichend von dem in Abbildung 5-1 dargestellten Datenflußmodell, das einen datengetriebenen Ansatz beschreibt, wird hier ein modellgetriebener Ansatz verfolgt. Bei einem datengetriebenen Ansatz wird versucht, aus den Daten bzw. den Segmentierungsergebnissen ein Modell einer Modelldatenbank zu instanziiieren. Beim modellgetriebenen Ansatz wird versucht, ausgehend von einem Modell, dieses im Bild zu finden. Der modifizierte Datenfluß ist in Abbildung 5-4 dargestellt.

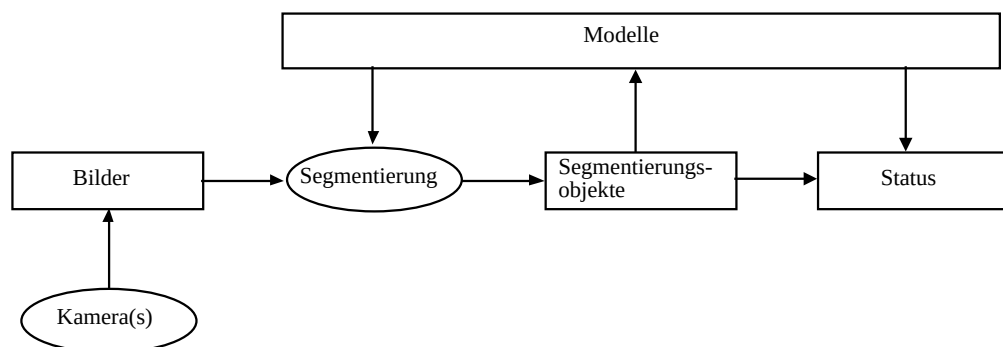


Abbildung 5-4: Modellgetriebener Datenfluß

Bei Echtzeitsystemen ist es oft nicht möglich aus den Segmentierungsergebnissen Hypothesen zu generieren und diese zu bewerten, um schließlich ein Modell zu instanziiieren. Meist muß sich für eine Hypothese entschieden werden und gegebenenfalls neu begonnen werden. Durch die in Abbildung 5-4 dargestellte direkte Beeinflussung der Segmentierung durch die Modelle wird dieser Verarbeitungsschritt auf Kosten der Allgemeingültigkeit eingespart. Die zu erkennenden Objekte sind durch das ausgewählte Modell vorgegeben. Die Musteranalyseaufgabe besteht in der Erkennung des aktuellen Zustandes des Objekts und einer Verfolgung mit einer möglichst hohen Aktualisierungsrate.

5.2.2 Wissensbasis

Klassisch wird zwischen deklarativen und prozeduralen Wissen unterschieden (Paulus 92). Die Formulierung von Modellen gehört dabei zur erstgenannten Klasse, der Vergleich der Modelle mit den Daten erfordert dann prozedurales Wissen. Bei einer Struktur eines Musteranalysesystems nach Abbildung 5-2 ergibt sich im Allgemeinen eine datengetriebene Segmentierungsphase mit einer anschließenden Abbildung der Segmentierungsergebnisse auf die Modelle, wobei gegebenenfalls auch die Segmentierungsphase wissensbasiert erfolgen kann. Oft ist es jedoch nicht sinnvoll, eine problemunabhängige Segmentierung vom Wissen über die Welt des Anwendungsgebietes zu trennen. Besonders bei der Erkennung bewegter Objekte in Bildfolgen erscheint es sinnvoll, problemabhängige Segmentierungsverfahren mit direkter Koppelung an die Wissensbasis aufzunehmen (Paulus 92). In dem hier vorgestellten System liegt eine weitere Vereinfachung dieser Vorgehensweise vor, indem die Wissensbasis implizit im Modell definiert wird. Durch die Verwendung des Modells als Bestandteil des Segmentierungsverfahrens fließt dieses Wissen direkt in den Verarbeitungszyklus ein. Auf eine gesonderte Wissensdatenbank kann dadurch verzichtet werden und der Verwaltungsaufwand wird minimiert. Die Verarbeitungsgeschwindigkeit des Systems hängt lediglich von den in den Modellen verwendeten Methoden ab.

5.2.3 Kontrollstruktur

Nach der in Abbildung 5-2 dargestellten allgemeinen Struktur eines Musteranalysesystems erfolgt die Steuerung und Auswahl der Verfahren durch das Wissen über das Modul Kontrolle. Es wählt und beeinflusst je nach den aktuellen Segmentierungsergebnissen die weitere Verarbeitung in der Weise, daß die Lösung der Analyseaufgabe gefunden wird. Die verwendete Kontrollstruktur ist dabei stark anwendungsabhängig und läßt sich unterscheiden in *interaktive* Verfahren mit Benutzerkontrolle, *hierarchische* Verfahren (bottom-up), *modellbasierte* Verfahren (top-down) oder *gemischte* und *datenbankorientierte* Verfahren (Niemann 90). Oft ist auch eine Bewertung der Ergebnisse nach den Kriterien Wichtigkeit, Zuverlässigkeit oder Genauigkeit erforderlich sowie effektive Suchalgorithmen, wie etwa der A^* -Algorithmus, wenn beispielsweise die beste Übereinstimmung mit den Modellen einer Wissensbasis gesucht wird. Für den Fall, daß stets die gleiche Abfolge von Aktionen auf alle Muster anwendbar ist, kann die Kontrollstruktur einmal festgelegt werden. Ein separates Kontrollmodul entfällt. Das hier vorgestellte System besitzt einen festgelegten Kontrollstruktur-Rahmen, der durch die verwendeten Modelle beeinflusst und um zusätzliche Methoden ergänzt werden kann. Neben dem Wissen wird damit auch ein Teil der Kontrollstruktur des Systems implizit in den Modellen definiert. Ein separates Kontrollmodul ist nicht vorhanden. Dem Nachteil einer nicht problemunabhängigen Formulierung einer Kontrollstruktur steht der Vorteil einer einfachen und effizienten Kontrolle über die Segmentierung gegenüber. Durch die Erweiterbarkeit um neue Modelle lassen sich auch neue Kontrollstrukturen in das System integrieren.

Der vorgegebene Kontrollstruktur-Rahmen geht davon aus, daß in den Bildfolgen ein oder mehrerer Kameras bewegte Objekte verfolgt werden sollen. Die Kameras liefern Ansichten einer realen Szene, deren projizierte Objekte aus einer Anzahl primitiver Elemente (z.B. Regionen) bestehen. Zur begrifflichen Unterscheidung der Objekte der realen Welt von der Projektion dieser auf ein Kamerabild werden Objekte der Szene, d.h. reale Objekte, als *Ding* bezeichnet, Objekte im Bild als *Objekt*. Die Primitive sowie die Abhängigkeit zwischen einzelnen Primitiven eines Objekts werden durch die Modelle definiert. Ein Primitiv besteht wiederum aus einer Anzahl von Merkmalen, die im Bild zu segmentieren sind. Nach erfolgter Segmentierung wird die Position aller Dinge in der Szene bestimmt. Am Ende der Verarbeitung kann vom Modell eine Vorhersage über die Positionen der Primitive im nächsten Bild getroffen werden. Eine Pseudocode-

Darstellung der Kontrollstruktur *mTrack* ist im folgenden dargestellt. Sie ersetzt damit die Verfolgungsprozeduren aus dem letzten Kapitel. Im Falle eines modellbasierten Verfahrens geschieht die Aktualisierung des Modells innerhalb von *thing->get_position()*, wodurch sich eine einheitliche Kontrollstruktur unabhängig von der Verwendung eines Modells ergibt. Die Verfahren zur Segmentierung von Primitiven, *primitive->segment()*, werden ebenfalls durch eine generische Funktion ersetzt, die Segmentierungsdetails (z.B. Schwellwert oder LUT-Verfahren) finden sich nun in *primitive->preprocess()* wieder.

PROCEDURE mTrack

```
// warte auf Szenenbeschreibung vom Server
get_scene_description_from_server()

// initialisiere verfolgte Dinge (und Objekte)
FOR ALL thing = zu verfolgendes THING
    scene->add_object(thing)

WHILE(true)
    FOR ALL view = Ansicht der Szene
        stelle aktuelles Bild zur Verfügung
        // find()
        FOR ALL object = zu verfolgendes Objekt im Bild
            FOR ALL primitive = Primitive von object
                primitive->segment()

            // bestimme Position aus zugehörigen Objekt im Bild
            // und Ergebnisdarstellung
            FOR ALL thing = Ding der Szene
                thing->get_position()
                thing->draw()

            // sage Position der Primitiven im nächsten Bild voraus
            thing->model->predict()

        send_data_to_server()
    ENDWHILE
END
```

PROCEDURE segment(primitive)

```
// Vorverarbeitung: z.B. Binarisierung
primitive->preprocess()

// Extrahiere Merkmale:
features->extract()
FOR ALL feature = extrahiertes Merkmal
    // schließe mehrfache Zuordnung zu Primitiven aus
    IF Schwerpunkt(feature) schon belegt NEXT

    // prüfe Nebenbedingungen zwischen Primitiven
    IF NOT primitive->check_constraints(feature) NEXT
    ELSE found = TRUE; BREAK

// Entscheidung über gefunden/verloren
IF NOT found für mehr als 4 frames
    lost = TRUE
ELSE
    lost = FALSE
END
```

5.3 Objektorientierter Entwurf

Für die Darstellung des Designs eines Softwaresystems gibt es eine Vielzahl Notations-sprachen, die jeweils mit einer bestimmten Analyse- und Entwurfsmethode einhergehen. Die bekanntesten objektorientierten Analyse- und Entwurfsmethoden sind *Booch*, *OOSE* und *OMT* (Rumbaugh 91). Erst 1995 einigten sich die führenden Wissenschaftler im Bereich Softwareengineering, Grady Booch, Ivar Jacobson und James Rumbaugh auf den gemeinsamen Standard *UML* (Fowler 97). UML steht für *Unified Modeling Language* und ist hauptsächlich eine graphische Notation zur Darstellung von Entwürfen und damit ein wichtiges Kommunikationsmittel bei der Softwareentwicklung. Im folgenden wird die UML-Notation zur formalen Darstellung der Klassenhierarchie und deren Abhängigkeiten verwendet. Die Abbildungen, die diese Notation verwenden, sind entsprechend gekennzeichnet.

5.3.1 Struktur des Entwurfs

Der objektorientierte Entwurf unterteilt sich in das Verfolgungssystem (tracking system), das eine Szene in Form von Objekten, Primitiven und Merkmalen (*features*) modelliert, die Modellierung der zu verfolgenden Dinge der realen Welt (physical environment), das Frontend, das die Verbindung zu einem Anwendungsprogramm realisiert, und die Szenenbeschreibung (scene description), in der das Anwendungsprogramm die zu verfolgenden Objekte definiert. Abbildung 5-5 zeigt die logischen Bestandteile des Entwurfs in Form eines sogenannten *Package-Diagramms*. Ebenso sind die Bestandteile des Verfolgungssystems zur Ergebnisdarstellung (results) und zur Behandlung des Bildeinzugs (camera) dargestellt.

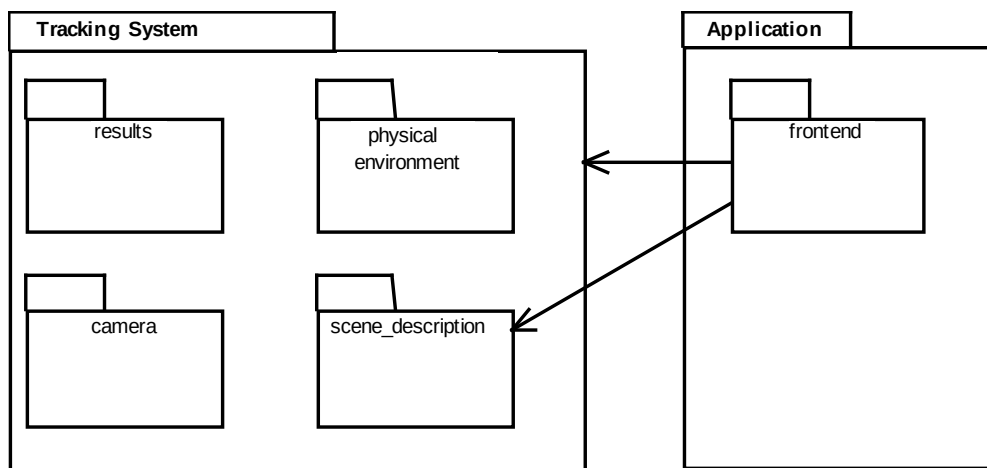


Abbildung 5-5: UML-Darstellung der Struktur nach logischen Bestandteilen

5.3.2 Modellierung des Verfolgungssystems

Die Grundidee des Verfolgungssystems ist in Abbildung 5-6 dargestellt, die die Abhängigkeiten zwischen den einzelnen Bestandteilen bzw. Klassen für eine Stereobildverarbeitung zeigt. Eine Szene besteht aus ein oder mehreren Ansichten (views), die jeweils mit einer Kamera verbunden sind. Die Szene enthält auch eine Möglichkeit einer Ergebnisdarstellung (results), die eine grafische Repräsentation der verfolgten Objekte zeigt.

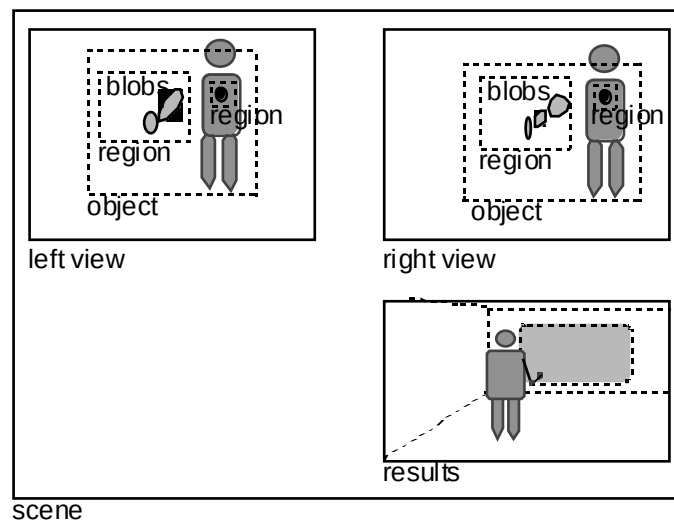


Abbildung 5-6: Zuordnung und Abhängigkeiten der Klassen bei der Personenverfolgung mit einem Stereoaufbau. Die im Bild zu findenden Primitive sind Regionen, die aus Blobs bestehen.

scene. Der Hauptbestandteil des Systems ist die Klasse *scene*, die die Verwaltungsinformation über die Anzahl der Ansichten (views) einer Szene, die Anzahl der Fenster für die Darstellung des Bildverarbeitungsprozesses und die Ergebnisdarstellungen enthält.

view. Die Klasse *view* repräsentiert die abstrakte Idee, daß Kameras zur Betrachtung einer Szene verwendet werden. Jeder *view* enthält als Bestandteil eine Kamera und eine Beschreibung der zu verfolgenden Objekte im Bild.

object. Die Klasse *object* beschreibt ein zu verfolgendes Objekt in Form von Primitiven. Das Objekt verwaltet auch die Querbezüge zwischen Primitiven im Falle von Abhängigkeiten zwischen den Primitiven eines Objekts. Abhängigkeiten definieren Nebenbedingungen für das Auffinden und Verfolgen der Primitive im Bild, so kann beispielsweise angegeben werden, daß ein Primitiv sich stets links von einem anderen befindet.

primitives. Die Klasse *primitives* beinhaltet die Primitive. Das sind die Grundbestandteile zum Auffinden von Objekten. Sie können aus einem oder mehreren Merkmalen (*features*) bestehen. Für die beschriebene Personenverfolgung werden zusammenhängende Regionen (*blobs*) verwendet (Abbildung 5-6).

features. Die Klasse *features* beschreibt die Merkmale eines Bildes oder Bildausschnitts mit den notwendigen Attributen, wie etwa Schwerpunkt, Bounding-Box, Hauptachse, Anzahl Pixel. Für die Personenverfolgung werden Blobs als Merkmale verwendet.

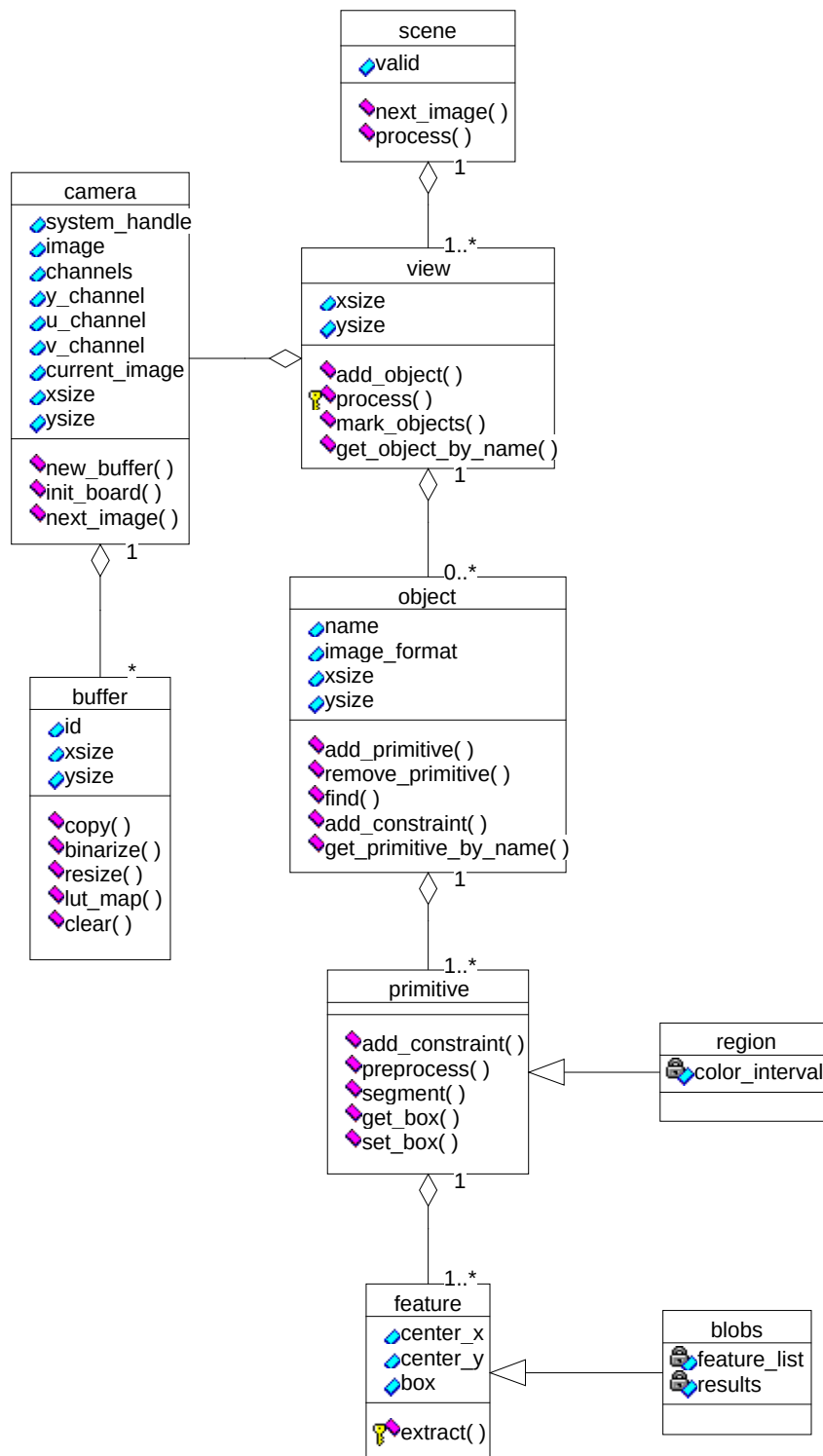


Abbildung 5-7: UML-Klassendarstellung des Verfolgungssystems

Abbildung 5-7 zeigt die Klassenhierarchie des Verfolgungssystems. Nach der UML-Notation bezeichnen Kanten mit Raute sogenannte Aggregationen, d.h. Bestandsbeziehungen, mit Angabe der Kardinalitäten. Ein Objekt besteht also aus ein oder mehreren Primitiven und jedes Primitiv ist eindeutig einem Objekt zugeordnet. Kanten mit Pfeil zeigen Vererbungen an, so ist eine Region eine Spezialisierung eines Primitivs. Im oberen

Teil des Klassenrumpfs sind die wichtigsten Variablen der einzelnen Klasse dargestellt, im unteren Teil die Methoden.

5.3.3 Modellierung der physikalischen Umgebung

Zur Modellierung der realen Welt müssen Begriffe oder Konzepte aus der Welt des Anwendungsgebiets in Form von Elementen und Beziehungen zwischen den Elementen modelliert werden. In dem beschriebenen System werden Objekte der realen Welt, durch sogenannte *Dinge* modelliert. Ein *Ding* kann hierbei als einfaches Modell eines zu verfolgenden Objekts der realen Welt angesehen werden. Es beschreibt die Objekte, die im Kamerabild zu verfolgen sind, mit Hilfe von Primitiven und ggf. zusätzlichen Nebenbedingungen. Bei komplexen Objekten kann ein *Ding* durch ein Modell ergänzt werden, das zusätzlich Wissen und Methoden enthält, die das Auffinden und Verfolgen des Objekts im Kamerabild erleichtern. Die Klasse *scene* stellt eine Methode *add_object()* zur Verfügung, die ein Objekt der realen Welt (*thing*) zu allen *views* der Szene hinzufügt. Hierbei wird für jedes *Ding* die Methode *add_primitives()* aufgerufen, in der die zu verfolgenden Primitive beschrieben sind. Nach der Analyse eines Bildes wird die Funktion *draw()* für jedes *Ding* aufgerufen, in der eine Folge von Funktionsaufrufen für die grafische Repräsentation der Analyseergebnisse für dieses Objekt definiert sind (vgl. auch den Kontrollstruktur-Rahmen *mTrack*, Seite 95).

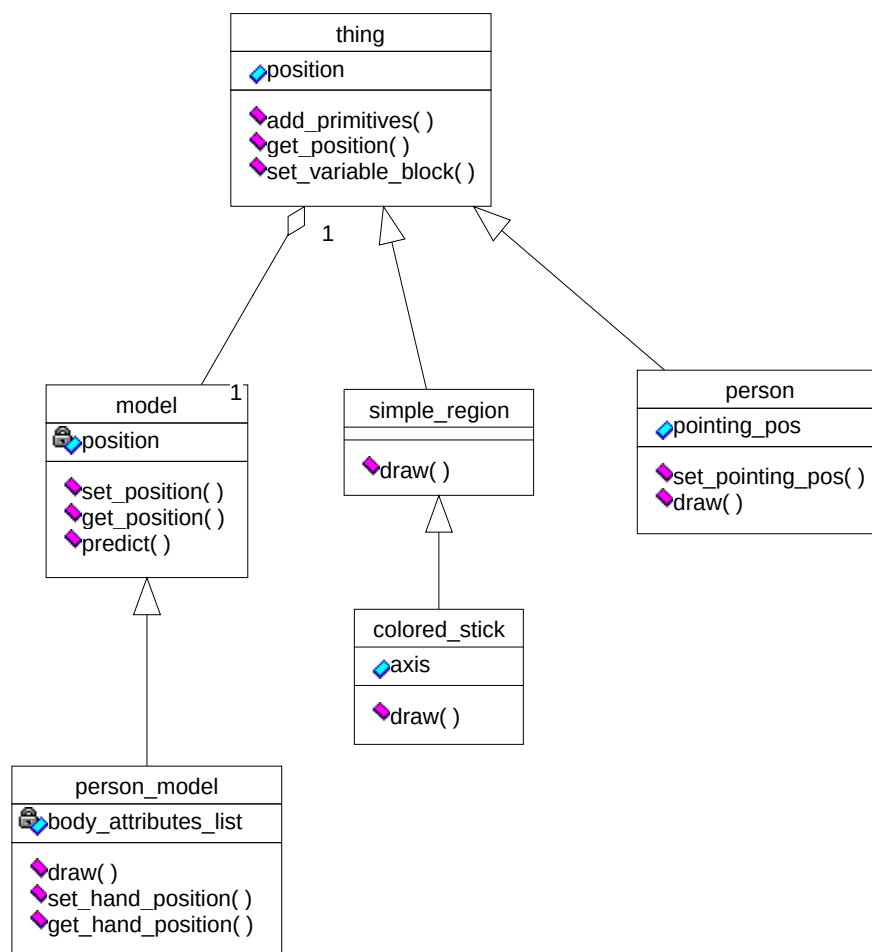


Abbildung 5-8: UML-Klassendarstellung der physikalischen Welt

thing. Die Klasse *thing* ist die abstrakte Basisklasse aller Dinge der realen Welt. Eine abgeleitete Klasse muß die Methoden zum Hinzufügen von Primitiven (*add_primitives()*) und eine Zeichenfunktion für die grafische Repräsentation des Dinges im Ergebnisfenster implementieren.

simple_region. Die Klasse *simple_region* ist von der Klasse *thing* abgeleitet und modelliert ein einfaches Objekt, das durch eine einzige Region im Kamerabild zu segmentieren ist.

model. Die Klasse *model* erlaubt die Verwendung eines elaborierten Modells zur Steuerung der Bildverarbeitung und die Integration von anwendungsbezogenem Wissen. Dazu wird das Modell mit den Daten der Bildverarbeitung aktualisiert und trifft jeweils am Ende der Verarbeitung eines Bildes eine Vorhersage der Objekte im nächsten Bild in Abhängigkeit seines Systemzustandes.

stick. Die Klasse *stick* beschreibt einen farbigen Zeigestock der für die Erkennung einer Zeigegestik verwendet werden kann. In diesem Fall ist *stick* eine *simple_region* mit der Ergänzung, daß die Hauptachse des segmentierten Objekts berechnet wird. Aus dieser Information wird direkt die Zeigerichtung berechnet (siehe Kapitel *Personenverfolgung*, Seite 62).

person. Die Klasse *person* beschreibt ein Objekt, daß mindestens aus zwei Regionen besteht, eine für den Körper einer Person und eine für die Hand. Für die Handposition werden dabei Nebenbedingungen formuliert, so daß die Hand nur auf einer Seite des Körpers gesucht wird (siehe Abschnitt *Einfaches Armmodell*, Seite 69).

5.3.4 Frontend

Das Frontend schafft für das Anwendungsprogramm eine einfache Möglichkeit zur Kommunikation mit dem Verfolgungssystem. Über das Frontend kann das Anwendungsprogramm die zu verfolgenden Objekte definieren und die Variablen der Objekte festlegen, die vom Serverprogramm übertragen werden sollen. Weiterhin lassen sich Abhängigkeiten zwischen den verfolgten Objekten der realen Welt und den vom Anwendungsprogramm verwalteten virtuellen Objekten angeben. So können beispielsweise sensitive Zonen im Raum definiert werden, in denen zuvor festgelegt Realobjekte virtuelle Objekte steuern: Befindet sich das Realobjekt in dieser Zone, so reagiert das virtuelle auf die zuvor festgelegte Weise. Eine andere Möglichkeit sind Bereiche des Realraums, die beim Eintritt und Verlassen durch ein zuvor festgelegte Realobjekt Funktionen des Anwendungsprogramms auslösen (sogenannte ENTER- und EXIT-Funktionen). Innerhalb dieser Bereiche lassen sich weiterhin eine Selektions- und eine Gedächtnisfunktion definieren. Die Selektionsfunktion wird dann aufgerufen, wenn eine (zuvor festgelegte) Variable anzeigt, daß ein Objekt für eine gewisse Zeit aktiviert worden ist. Dazu wird dem Frontend die Variable mitgeteilt, von der die Aktivierung abhängig ist, und zusätzlich eine Funktion des Anwendungsprogramms, die für jeden Wert der Variablen eine Liste der aktivierten Objekte zurückgibt. Ein Beispiel soll dies verdeutlichen:

Auf einer Projektionswand wird ein Cursor durch die Zeigerichtung des Benutzers gesteuert. Zeigt der Benutzer länger als eine Sekunde auf ein virtuelles Objekt, soll es selektiert werden. Dazu wird dem Frontend mitgeteilt, daß die Aktivierung von der x- und y-Position der Zeigerichtung abhängt. Weiterhin wird eine sogenannte Rollover-Funktion dem Frontend übermittelt, die bei jeder Änderung der Variablen aufgerufen wird und eine Liste der

aktivierten Objekte liefert. Das Frontend ruft die Selektionsfunktion für die Objekte auf, die für länger als eine Sekunde aktiviert worden sind.⁷

Eine weitere Funktion des Frontend ist die Gedächtnisfunktion, die es gestattet, Objekte vom Typ *item* im Frontend an bestimmten Raumpositionen zu registrieren. Weiterhin wird ein Objekt der Realwelt benannt, von dem diese Funktion abhängt: Befindet sich das Realobjekt in der Nähe von derart registrierten Realobjekten, wird dem Anwendungsprogramm eine Liste der an dieser Raumposition befindlichen Objekte mitgeteilt. Dadurch lassen sich beispielsweise die Funktionen der Netscape-Anwendung (siehe Abschnitt *Szenario*, Seite 25) realisieren. Hierbei werden die Methoden für die zu lösende Aufgabe und die graphische Repräsentation der virtuellen Objekte vom Anwendungsprogramm implementiert, die Interaktionsmuster der Intuitiven Schnittstelle vom Frontend zur Verfügung gestellt. Als Schnittstelle dient die sogenannte Szenenbeschreibung, eine Textdatei, die die Definition der erwähnten Abhängigkeiten und Funktionalitäten erlaubt. Der Aufbau dieser Datei wird im nächsten Abschnitt beschrieben.

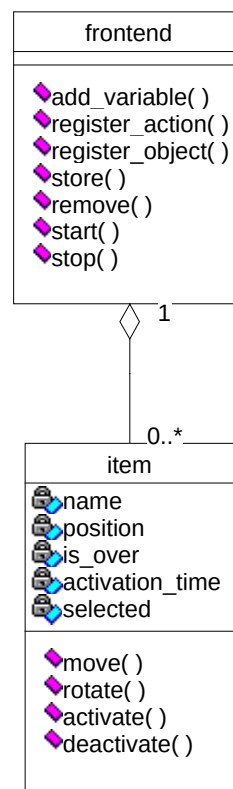


Abbildung 5-9: UML-Klassendarstellung des Frontends

Abbildung 5-8 zeigt das aus einer Klasse bestehende Frontend mit den Funktionen zum Registrieren von Variablen, Aktionen und Objekten. Die Funktionen *store()* und *remove()* werden für die Gedächtnisfunktion benötigt, *start()* und *stop()* schließlich starten bzw. beenden den Ablauf. Weiterhin ist in Abbildung 5-9 das Basisobjekt für virtuelle Objekte dargestellt. Virtuelle Objekte des Anwendungsprogramms, die im Frontend registriert werden sollen, müssen von diesem Typ sein, insbesondere müssen sie die Funktionen *move()* und *rotate()* implementieren, die als reine virtuelle Funktionen realisiert sind.

⁷ Die Rollover-Funktion ist Teil des Anwendungsprogramms, da nur hier entschieden werden kann, ob ein Objekt der Anwendung aktiviert worden ist, oder nicht.

5.3.5 Szenenbeschreibung

Für den Datenaustausch zwischen Serverprogramm und Anwendungsprogramm bzw. Frontend wird die sogenannte Szenenbeschreibung (scene description) verwendet. Sie enthält sowohl eine Liste der zu verfolgenden Objekte der realen Welt, als auch Abhängigkeiten mit der virtuellen Szene, die beispielsweise auf einer Projektionswand dargestellt wird. Durch die Szenenbeschreibung wird auch die enge Kopplung zwischen den verfolgten Objekten der realen Welt und den Interaktionsmethoden des Anwendungsprogramms vereinfacht. Bestimmte Archetypen der Rauminteraktion und Bedienung werden durch das Frontend zur Verfügung gestellt und lassen sich durch die Szenenbeschreibung mit virtuellen Objekten verbinden. Die Szenenbeschreibung ist vom Anwendungsprogramm zur Verfügung zu stellen und wird vom Frontend eingelesen und anschließend zur Initialisierung des Serverprogramms verwendet (siehe Abbildung 5-3). Sie teilt sich in die Abschnitte reale Welt (physical environment), eine Beschreibung des zu sendenden Datenblocks (data block) und die virtuelle Szene mit den Abhängigkeiten zu den Realobjekten (virtual scene).

PHYSICAL_ENVIRONMENT. Dieser Abschnitt beschreibt die Objekte der realen Welt durch die im letzten Abschnitt definierten *Dinge*. Er enthält eine Liste der zu verfolgenden Objekte, d.h. Objekte der Klasse *thing*.

Beispiel:

```
# -----
#
PHYSICAL_ENVIRONMENT
  THING person person1 NIX 1
  THING simple_region person1 gruen 1
  THING simple_region red_pole rot 2
  THING simple_region yellow_pole gelb 2
END
```

DATA_BLOCK. In diesem Abschnitt wird der Aufbau des Datenblocks mit den Analyseergebnissen beschrieben, den das Serverprogramm an das Anwendungsprogramm schickt. In dem Beispiel besteht der Datenblock aus den Positionsdaten einer Person, der Zeigerichtung der Person und der Position der beiden Platzhalterfiguren. Zusätzlich wird der Code des Sprachkommandos übermittelt.

Beispiel:

```
# -----
#
DATA_BLOCK
  SPEACH COMMAND_CODE
  ENTRY person1 pos_x
  ENTRY person1 pos_y
  ENTRY person1 pos_z
  ENTRY person1 pointing_pos_x
  ENTRY person1 pointing_pos_y
  ENTRY red_pole pos_x
  ENTRY red_pole pos_y
  ENTRY red_pole pos_z
  ENTRY yellow_pole pos_x
  ENTRY yellow_pole pos_y
  ENTRY yellow_pole pos_z
END
```

VIRTUAL_SCENE. Dieser Abschnitt beschreibt die virtuelle Szene, die durch den Benutzer manipuliert wird. Er enthält Abbildungen von den vom System zur Verfügung gestellten Kommandos auf Modifikationen von Objekten. So können beispielsweise sen-

sitive Zonen im Realraum definiert und die Bewegung des Benutzers auf eine Aktion abgebildet werden, solange er sich in dieser Zone befindet. Die Interaktionsmuster werden als Rahmen durch Bibliotheksfunktionen des Frontends zur Verfügung gestellt, die dann entsprechende Methoden der Komponenten der virtuellen Szene aufrufen. Das Anwendungsprogramm muß die grafische Repräsentation der Objekte und die Methoden für Aktionen implementieren. Beispielsweise müssen für jedes vom Frontend zu manipulierende Objekt die Funktionen *move()* und *rotate()* implementiert werden. Weitere Funktionen können über den Funktionsnamen referenziert werden, indem hier der Name angegeben wird und die Funktion vom Anwendungsprogramm über den textuellen Namen im Frontend registriert wird.

Beispiel:

```
# -----
#
VIRTUAL_SCENE
ACTION ROLLOVER person1 pointing_pos_x pointing_pos_y ANY cursor_over

AREA scene_area ANY ANY 0.0 TO ANY ANY 4.20
  ACTION ENTER person1 pos_x pos_y pos_z enter_area
  ACTION EXIT person1 pos_x pos_y pos_z exit_area
  ACTION SELECT select_object
  ACTION DESELECT deselect_object

  ZONE room_zone 0.50 ANY 1.50 TO 2.00 ANY 3.50
    ACTIVITY LINK red_pole pos_x pos_y pos_z TO MOVE place_holder_1
    ACTIVITY LINK yellow_pole pos_x pos_y pos_z TO MOVE place_holder_2
  END_ZONE
END_AREA

AREA stage_area ANY ANY 3.10 TO ANY ANY 6.00
  ACTION ENTER person1 pos_x pos_y pos_z enter_area
  ACTION EXIT person1 pos_x pos_y pos_z exit_area
  ACTION SELECT add_object_to_scene

  ZONE conveyor_zone 0.0 ANY ANY TO 2.8 ANY ANY
    ACTIVITY LINK person1 pos_x pos_y pos_z TO MOVE conveyor
  END_ZONE
END_AREA
END
```

Das Beispiel zeigt die virtuelle Szenenbeschreibung für eine Anwendung, wie sie im Prototyp zur Filmplanung realisiert wurde. Die Berührungsfunktion (ROLLOVER) wird abhängig von der x- und z-Position von person1 definiert. Sie ist dazu vom Anwendungsprogramm zu implementieren und im Frontend zu registrieren. Die Berührungsfunktion muß eine Liste der Objektnamen zurückgeben. Für jedes „berührte“ Objekt wird dann die Funktion *activate()* aufgerufen. Es folgen zwei Bereiche (AREA), in denen sowohl *ENTER*-, als auch *EXIT*-Funktionen und verschiedene Select-Funktionen definiert sind. Nach einer gewissen Aktivierungszeit, angezeigt über die „globale“ Rollover-Funktion, wird im ersten Bereich eine Select-Funktion, die über den textuellen Eintrag „select_object“ im Frontend registriert ist, aufgerufen. Ein weiterer Bereich mit ähnlichen Funktionen folgt. In beiden Bereichen sind sensitive Zonen im Realraum definiert, wobei die erste die Bewegung von zwei Platzhalterfiguren (siehe Abbildung 6-2, Seite 115) mit den im Frontend unter den Namen *place_holder_1* und *place_holder_2* registrierten Objekten verbindet. Die zweite sensitive Zone steuert ein unter dem Namen *conveyor* registriertes Objekt in Abhängigkeit der x-Position von dem *Ding person1*. Die Registrierung von Objekten im Frontend nach den angegebenen Namen kann dynamisch während der Laufzeit erfolgen.

Im folgenden ist die Syntax der Szenenbeschreibung in Backus-Naur-Notation angegeben.

```

<scene_description> ::=      SCENE_DESCRIPTION
                             <physical_environment>
                             <data_block>
                             <virtual_scene>
                             END_SCENE_DESCRIPTION

<physical_environment> ::=  PHYSICAL_ENVIRONMENT
                             {THING <thing> <update_rate>}
                             END

<thing> ::=                  thing_class_in_system <thing_name> <parameter>

<parameter> ::=              parameter dependent on thing class

<update_rate> ::=            1 to n frames

<data_block> ::=             DATA_BLOCK
                             {ENTRY <thing_name> <variable>}
                             END

<virtual_scene> ::=          VIRTUAL_SCENE
                             {<AREA> | <ZONE> | <ACTION> | <ACTIVITY>}
                             END

<AREA> ::=                   AREA <location> TO <location>
                             {<ZONE> | <ACTION> | <ACTIVITY>}
                             END_AREA

<ZONE> ::=                   ZONE <location> TO <location>
                             {<ACTIVITY>}
                             END_ZONE

<ACTION> ::=                 ACTION
                             ENTER <depend_spec> <user_function> |
                             EXIT  <depend_spec> <user_function> |
                             MEMORY <depend_spec> <user_function> |
                             SELECT <user_function> |
                             DESELECT <user_function> |

<ACTIVITY> ::=               LINK <depend_spec> TO <item_method> <object>

<location> ::=               <any_or_var> <any_or_var> <any_or_var>

<any_or_var> ::=             <variable> | ANY

<variable> ::=               thing_variable as defined in thing

<depend_spec> ::=            <thing_name> <variable> <variable> <variable>

<thing_name> ::=             user defined thing_name

<user_function> ::=          user define function name

```

5.3.6 Zusammenspiel der Komponenten

Abbildung 5-10 zeigt das Zusammenspiel des Verfolgungssystem mit den Klassen der realen Welt. Querbezüge zwischen Klassen werden durch einfache Kanten (Assoziationen) dargestellt. Die Darstellung von Funktionen und Variablen wurde auf die in diesem Zusammenhang relevanten beschränkt. Die Reihenfolge der Verarbeitung ergibt sich durch den Kontrollstruktur-Rahmen aus Abschnitt 5.2.3.

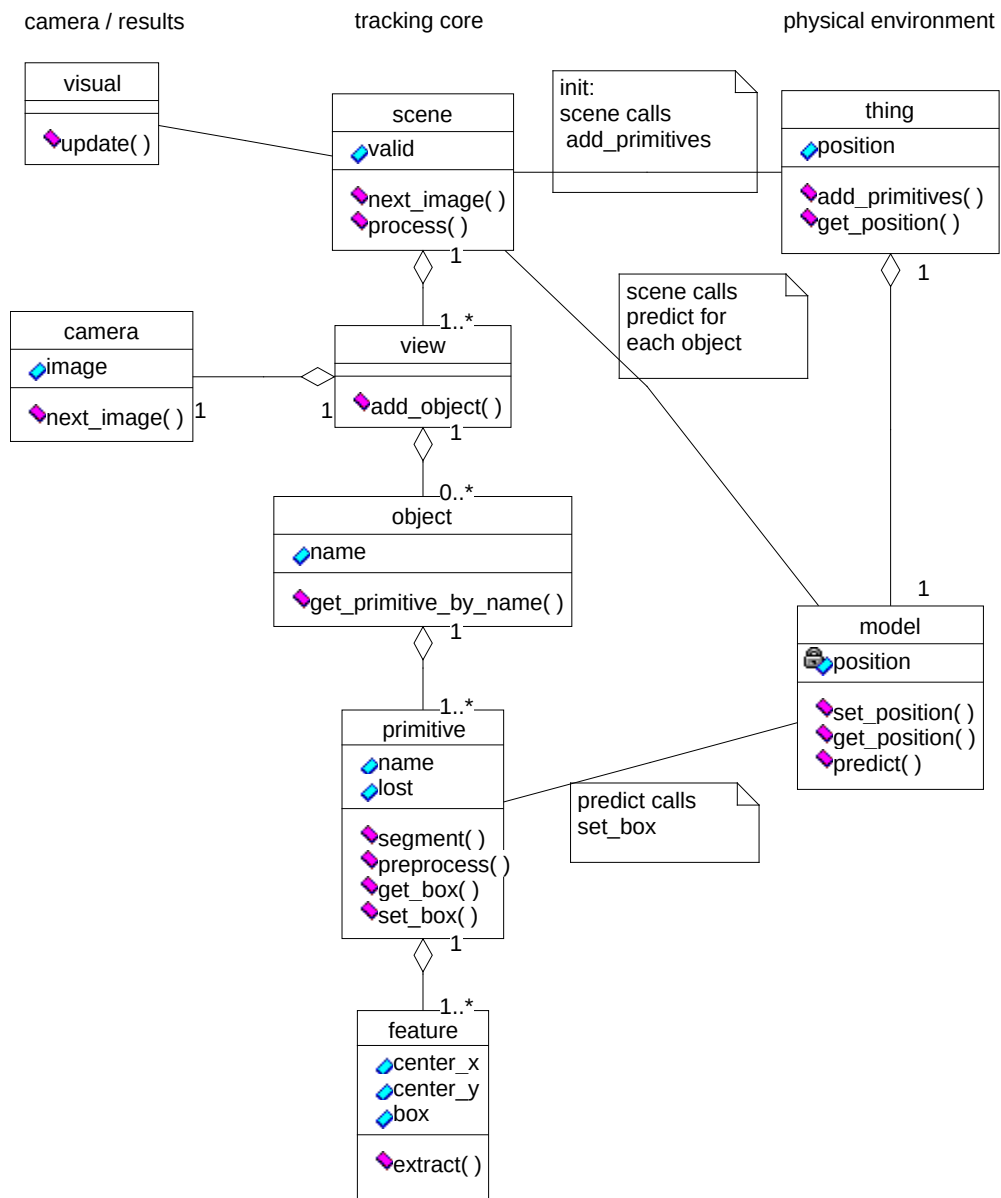


Abbildung 5-10: UML-Darstellung der Abhängigkeiten der einzelnen Module (beschränkt auf relevante Details)

5.4 Entwicklerschnittstelle

Das Verfolgungssystem ist so konzipiert, daß ein Systemprogrammierer es mit neuen Komponenten erweitern kann, ohne das Gesamtsystem neu übersetzen zu müssen. Dazu steht eine Schnittstelle über eine DLL-Bibliothek (dynamic link library) der Windows-Umgebung zur Verfügung. Diese Bibliothek enthält die vom Programmierer gewünschten Erweiterungen, d.h. Definitionen neuer Realobjekte mit Modellen (Klassen *thing* und *model*) und neuer im Bild zu suchenden Primitiven und Merkmalen (Klassen *primitive* und *feature*). Die so definierten Klassen werden zur Laufzeit des Systems geladen und können damit über die Szenenbeschreibung verwendet werden. Abbildung 5-11 zeigt die Übersicht über die relevanten Klassen mit neuen, d.h. erweiterten, Klassen (hervorgeho-

ben durch das Präfix *user_*). Bei den erweiterten Klassen sind die zu implementierenden Methoden herausgestellt.

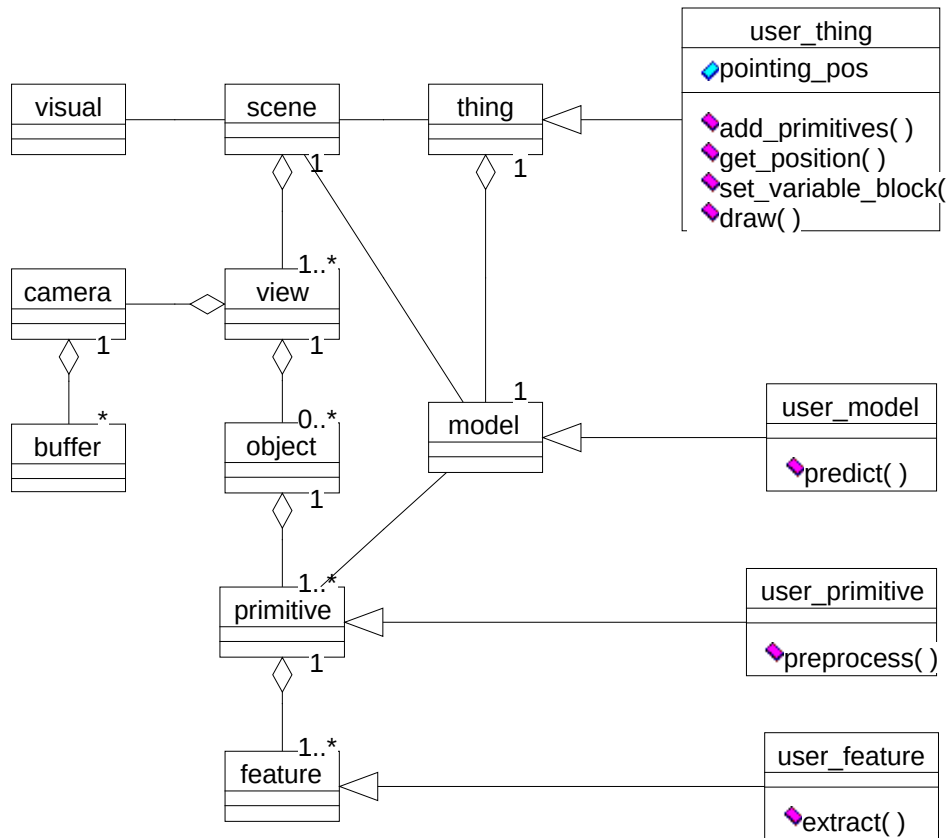


Abbildung 5-11: UML-Darstellung des Verfolgungssystems mit den vom Systemprogrammierer erweiterten Klassen, gekennzeichnet durch das Präfix *user_*

Im folgenden sind zur Verdeutlichung Teile einer Definition eines neuen Realobjekts *user_thing* unter Verwendung von *user_model* und *user_primitive* dargestellt. Die Definition von *user_model* und *user_primitive* geschieht auf analoge Weise. Sie sind hier aus Gründen der Übersichtlichkeit nicht aufgeführt. Ebenso wurde auf die Klassendefinitionen und auf Beispiele für die Funktionen *draw()* und *get_position()* verzichtet, da hier lediglich der dem System zugrundeliegende Mechanismus verdeutlicht werden soll.

Konstruktor. Im Konstruktor wird ein Objekt der Klasse *user_thing* inklusive seiner Basisklasse *thing* initialisiert. Der als Text angegebene Wert des Arguments *parameter* kann hier verwendet werden, um das Objekt zu parametrieren. Dieser Parameter kann dann in der Szenenbeschreibung dem Objekt entsprechend gewählt werden.

```

//-----
//
user_thing::user_thing(char* name, char* parameter,
                      int update_rate)
    : m_thing ( name, new user_model(), update_rate )
{
    color = select_color_from_name ( parameter ) ;
}
  
```


Primitive. Diese Funktion definiert die Anzahl und Art der Primitive, aus der dieses Realobjekt besteht. Die ersten drei Parameter sind von der Basisklasse *primitive* vorgegeben, die folgenden sind Bestandteil der neu definierten Klasse *user_primitive*. Die Funktion *add_primitives()* wird vom System aufgerufen. Dabei wird ein Objekt der Szene übergeben, in das das Primitiv mit Hilfe der Funktion *add_primitive()* eingetragen werden muß.

```
// -----
//
void user_thing::add_primitives ( object * object )
{
    user_primitive * region1 = new user_primitive ( "region",
        update_rate, object->primitive_info, color,
        center_area_box_8_30, 40, "mylookuptab.lut" ) ;

    object->add_primitive ( region1 ) ;
}
```

Variablendefinition. Für jedes Realobjekt müssen textuelle Namen für die Variablen definiert werden, die über das Serverprogramm dem Anwendungsprogramm übermittelt werden sollen. Anschließend können diese Variablen in der Szenenbeschreibung über die definierten textuellen Namen referenziert werden. Eine Variablendefinition geschieht über die Funktion *add_variable()*.

```
// -----
//
void m_user_thing::set_variable_block ( )
{
    add_variable ( "pos_x", &norm_position.x ) ;
    add_variable ( "pos_y", &norm_position.y ) ;
    add_variable ( "slope", &slope ) ;
}
```

DLL-Schnittstelle. Damit das System ein Objekt der Klasse *user_thing* erzeugen kann, muß eine Funktion in die DLL-Bibliothek exportiert werden, die ein solches Ding zurückliefert. Der Funktionsname muß dabei die Konvention *new_<thing_class_name>* einhalten, wobei *<thing_class_name>* in diesem Fall *user_thing* entspricht.

```
extern "C"
{
    // -----
    //
    DllExport thing*
    new_user_thing(char*name,char*parameter,int rate)
    {
        user_thing * t ;
        t = new user_thing ( name, parameter, rate ) ;
        return t ;
    }
}
```

Applikation. Das Anwendungsprogramm unter Verwendung des Frontends stellt sich schließlich wie folgt dar:

```
#include "frontend.h"

// global variables
//
float      x, y, slope ;

// -----
//
void main ( int argc, char *argv [] )
{
    // initialize connection to PC server program
    //
    m_frontend frontend ( "193.197.170.118" ) ;

    // register variables in frontend
    // the order must match order in the data_block of the
    // scene_description, variables must be floats
    //
    frontend.add_variable ( &x ) ;
    frontend.add_variable ( &y ) ;
    frontend.add_variable ( &slope ) ;

    // start transmission of values
    //
    frontend.start () ;

    // use received values
    //
    while( !ende )
    {
        do_main_application();
    }
}
```

6 Empirische Untersuchung: Tests eines Prototyps

Die problem- und benutzerorientierte Entwicklung interaktiver Systeme gliedert sich in der Regel in mehrere Phasen von der Analyse über Design und Implementierung bis hin zur Weiterentwicklung. Oft ist hierbei eine durchgehend sequentielle Abarbeitung der Phasen nicht möglich, so daß zum einen Iterationen der einzelnen Phasen notwendig sind, zum anderen weitere Phasen wie Prototyping und Evaluierung hinzukommen. Gründe hierfür liegen vor allem in ungenügend motivierten und vorbereiteten Analysen sowie Designentscheidungen. Aus dem Ergebnis empirischer Studien geht weiterhin hervor, daß die Mehrheit von Entwicklern sich eher nicht mit der Erfassung und Umsetzung von Aufgaben sowie Benutzeranforderungen beschäftigt (Stry 96). Unter einem *Prototyp* versteht man eine Version einer Anwendung im Rahmen einer Entwicklung, die zur Klärstellung von Aufgaben, Benutzeranforderungen, Design- und Implementierungsvorschlägen sowie zur Erprobung neuer Interaktions- und Implementierungstechniken erstellt wird. Entwickelte Interaktionsformen eines Prototyps müssen also nicht notwendigerweise in die endgültige Anwendung übernommen werden, wenn beispielsweise die Evaluierung, d.h. die Bewertung des Prototyps oder eines Entwurfs ergibt, daß diese den Benutzeranforderungen nicht vollständig gerecht werden. Nach Stry (96) sind bei der Verwendung von Prototypen folgende Vorteile zu erwarten:

- Testen von Bedingungen, die durch bestehende Designprinzipien nicht abgedeckt werden,
- Evaluierung eines ersten Konzepts für eine Benutzerschnittstelle,
- frühzeitiges Feedback durch den Benutzer,
- erhöhte Benutzerakzeptanz,
- Kostenreduktion im Gesamtentwicklungsprozeß.

Die Nachteile sind die eingeschränkten Bedingungen, unter denen die Anwendung zum Einsatz kommt, eine Überbewertung der Ergebnisse bei der Evaluierung und die Schwierigkeit, Kriterien für den Prototyping-Prozeß komplexer Anwendungen zu erstellen. Anhand eines Prototyps lassen sich jedoch die Gestaltungsmerkmale und Bewertungskriterien interaktiver Systeme einer ersten Prüfung unterziehen. Zu den Merkmalen und Kriterien, die im Zusammenhang mit dem hier beschriebenen System überprüft werden sollen, gehören eine *Aufgabenangemessenheit* in Bezug auf Brauchbarkeit und Funktionalität, die *Bedienbarkeit* und die *Erlernbarkeit* eines Systems. Aufgabenangemessenheit bedeutet, daß der Benutzer bei der Durchführung von Arbeitsaufgaben unterstützt wird, ohne unnötige Belastung durch die Eigenschaften der Interaktionsmittel. Die Bedienbarkeit bezieht sich auf die Beeinflussung der Geschwindigkeit und der Auswahl und Reihenfolge von Interaktionshilfsmitteln. Die Erlernbarkeit schließlich bedeutet, daß die Aufgabenbewältigung mit dem System in einer angemessenen Zeitspanne erlernbar ist. Für die Entwicklung der intuitiven Schnittstelle wurde ein Prototyp entwickelt und bewertet. Von den zur Bewertung üblicherweise herangezogenen Techniken, wie direkte oder indirekte *Befragungen* oder *Aufzeichnungen*, direkte oder indirekte *Beobachtungen* und *Experimente*, wurde eine Interviewform als Art der Befragung ausgewählt. Zusätzlich wurde dem Benutzer in einem Experiment die Möglichkeit der Bedienung des Systems gegeben. Das gesamte Interview inklusive Benutzertest wurde durch Videoaufzeichnung protokolliert. Im folgenden werden zuerst die verschiedenen Methoden der empirischen Sozialfor-

schung genannt, von denen die gewählten Methoden *Intensivinterview* und *Experiment* sowie die *teilnehmende Beobachtung* genauer erläutert werden. Anschließend werden der Prototyp und seine Funktionalität vorgestellt. Der Forschungsplan für die empirische Untersuchung und eine Auswertung schließen dieses Kapitel ab.

6.1 Empirische Sozialforschung

In der Wissenschaft haben wir es häufig mit einer mangelnden Problemorientierung, einem gebrochenem Verhältnis zwischen Theorie, Empirie und Praxis zu tun. „Die einen denken theoretisch und machen «Theorie», die anderen entwickeln Methoden und korrigieren Fragebögen, weitere entwickeln Computer-Programme, wieder andere stellen politische Praxis über die Erfordernisse einer empirischen Basis oder einer methodologisch-empirischen Kontrolle der eigenen Arbeit, weil sie ihren Ansatz nicht als Annahmen, sondern als Gewißheit betrachten, zu der es nur einiger «Daten» bedarf“ (Friedrichs 90, S. 112). Die empirische Sozialforschung versteht sich hierbei als ein auf Problemlösung orientiertes Handeln, das die genaue Formulierung eines Problems und die Kenntnis einer angemessenen Methode zu einer Lösung voraussetzt. Um den genannten Konflikt zu lösen, ist es notwendig, Methoden der empirischen Sozialforschung in die Forschungsarbeit mit einzubeziehen. Besonders für die Entwicklung und Bewertung interaktiver Systeme, eines der am wenigsten empirisch abgesicherten Gebiete (Stary 96), erscheint die Durchführung einer Untersuchung zum Zwecke der Erlangung neuer Erkenntnisse über den Problembereich als zwingend erforderlich. Je nachdem, welche Kenntnis dabei über einen Objektbereich bereits vorliegt, wird eine empirische Untersuchung eher beschreibend (deskriptiv) oder analytisch (Hypothesen prüfend) sein. Analytische Studien setzen beschreibende voraus. In einer beschreibenden Studie will man Kenntnisse über einen bisher unbekannten oder nur wenig bekannten Gegenstandsbereich gewinnen, die Auswahl von Variablen ist mehr oder weniger intuitiv, der Studie liegen nur vage Annahmen zugrunde. Sind die Beschreibungen hinreichend ausführlich und nach gleichen Merkmalen vorgenommen worden, so stellen solche Studien eine außerordentlich wichtige Grundlage für weitere Studien und für die Formulierung von Hypothesen dar.

6.1.1 Methoden

Zur Planung einer Untersuchung ist die Kenntnis aller möglicher Methoden wichtig, da keine der Methoden universell anwendbar ist und es zwischen den Methoden fließende Übergänge gibt bis hin zu kombinierten Methoden. Eine gute Übersicht über die verschiedenen Methoden der empirischen Sozialforschung findet sich in (Friedrichs 90). Eine Besprechung von kombinierten, standardisierten Verfahren, die sich mit der Bewertung von Benutzerschnittstellen beschäftigen, findet sich in (Stary 96). Oft haben solche standardisierten Verfahren den Nachteil, daß sie nur nach einer gründlichen Schulung anwendbar sind und daher einen großen Verwaltungs- und Zeitaufwand mit sich ziehen. Nach Stary (96) lassen sich die zur Bewertung verwendeten Methoden in folgende Kategorien einteilen:

Erhebungen. Hierunter fallen Befragungen und Aufzeichnungen inklusive Interviews und die Verteilung und Auswertung von Fragebögen oder Prüflisten mit dem Ziel, den Erfahrungsschatz von Benutzern zu erfassen. Hierbei fallen die Ergebnisse um so präziser aus, je genauer die möglichen Fragen und Antworten dem Untersuchten mitgeteilt werden. Das Interview ist die mündliche Form der Befragung, die ebenso durch eine schriftlichen Befragung durchgeführt werden kann. Die mündliche Form wird dann angewandt, wenn nur wenig über den untersuchten Problemkreis bekannt ist, die Verwendung von Fragebögen oder Prüflisten bedarf einer genauen Kenntnis über Problemkreis und be-

fragte Personen. Es wird zwischen einem *strukturierten Interview*, bei dem Fragethema und Frageanordnung festgelegt sind, und einem *standardisierten Interview*, bei dem die Frageformulierung festgelegt ist, unterschieden. Interviews ohne Standardisierung und mit nur geringer Strukturierung lassen sich als explorative oder Intensivinterviews bezeichnen.

Beobachtungen erlauben die direkte oder indirekte Partizipation des Untersuchenden bei der Aufgabenausführung durch die Benutzer. Im ersten Fall spricht man von einer teilnehmenden Beobachtung, wenn der Beobachtende dem Benutzer sichtbar gegenübertritt, im zweiten Fall werden Kameras oder Spiegelwände verwendet, um den Beobachter vom Untersuchenden zu verbergen.

Experimente untersuchen die Abhängigkeit und wechselseitige Beeinflussung von Testgrößen bzw. Variablen durch eine kontrollierte Durchführung von Testaufgaben durch den Benutzer. Meist finden sie mit einer zufälligen Auswahl von Aufgaben, Benutzern und Computersystemen statt, da es keine empirisch fundierte Theorien zur Mensch-Maschine-Kommunikation gibt (Stary (96)).

6.1.2 Intensivinterview

Das Intensivinterview ist eine mündliche Form der Befragung mit nicht-standardisierten Fragen und mit einem geringen Maß an Strukturierung der Fragenanordnung. In der Literatur wird diese Form u.a. auch „offene Befragung“, „Gespräch“, „zentriertes Interview“ oder „Tiefeninterview“ genannt. Ziel ist hierbei, durch eine den spezifischen Problemen und Bedürfnissen des Befragten angepaßte Befragung, genauere Informationen vom Befragten zu erlangen. Das Interview wird anhand eines *Leitfadens*, das ist ein grob strukturiertes Schema, durchgeführt, wodurch der Interviewer stärker auf den Befragten eingehen kann und die Möglichkeit des Nachfragens gegeben ist. Durch die relativ offene Gestaltung ist das Intensivinterview eine wichtige Methode, um Einsichten zu gewinnen in das Denken der ausgewählten Personen, in die Struktur von dem Forscher noch unbekannten Problemen (Exploration), sowie zur Vertiefung von aus standardisierten Interviews erzielten Ergebnissen. Dem explorativen Charakter stehen als Nachteile der Einfluß des Interviewers in der Erhebungssituation, die Dauer des Interviews, der zeitliche Aufwand bei der Auswertung und die geringe Vergleichbarkeit der Ergebnisse gegenüber (Friedrichs 90). Der Einfluß des Interviewers muß bei der Auswertung des Interviews mit berücksichtigt werden. Oft brauchen solche Interviews sowie deren Auswertung so viel Zeit, daß nur jeweils wenige Interviews durchgeführt werden können. Statt einer Stichprobe wird meist eine gezielte Auswahl von Personen praktiziert. Das Ergebnis ist eine Datenmatrix mit verschiedenen und ungleich umfangreichen Informationen. Auch die als vergleichbar erscheinenden Informationen können in unterschiedlichen Interaktionsprozessen gewonnen worden sein. Trotzdem liefert das Intensivinterview einen wichtigen Beitrag zur Exploration eines Problems. Häufig wird es auch als erster Schritt im Hinblick auf eine spätere standardisierte Befragungsform benutzt.

Neben dem Leitfaden ist zur Vorbereitung eines Intensivinterviews ein *Forschungsplan* mit einer Begründung von Stellenwert, Zielen und Inhalten des Vorgehens zu erstellen. Nach Friedrichs (90) empfiehlt es sich, vorab zwischen Hypothesen und weniger explizierten Fragestellungen zu unterscheiden. Aus den Hypothesen und Fragestellungen werden dann die Fragen des Interviews abgeleitet, die sich unterteilen in *Schlüsselfragen*, die in jedem Falle zu stellen sind, und *Eventualfragen*, die gestellt werden können, wenn es der Gesprächsverlauf erlaubt. Dabei sollten die Fragen durchweg als sogenannte offene Fragen gestellt werden, d.h. dem Befragten sollten die möglichen Antworten, etwa im Sinne einfacher Ja-/Nein-Entscheidungen, nicht vorgegeben werden. Bei Nachfragen

können jedoch auch geschlossene Fragen gestellt werden, wobei sich ein Nachfragen bei unklaren Formulierungen des Befragten nicht empfiehlt, sondern eher ein Wechseln auf die „Beziehungsebene“ sinnvoll ist, d.h. beispielsweise eine Formulierung wiederholen, schweigen, oder den Befragten zum weitererzählen ermuntern. Ist der Befragte unsicher, sind Zusammenfassungen und Interpretationen sinnvoll, um sich zu vergewissern, ob der Befragte verstanden worden ist. Vor dem eigentlichen Interview kann ein Pretest, d.h. das Abhalten einiger „Testinterviews“, durchgeführt werden. Dies ist sinnvoll, wenn viele Interviews durchgeführt werden sollen oder wenn ein Problem allein mit dieser Methode untersucht werden soll. Der Pretest liefert Aufschluß über die Brauchbarkeit des Leitfadens und dient der Schulung des Interviewers. Bei der Auswertung ist der Wert für die Hypothesenprüfung aus den oben genannten Gründen eingeschränkt, was dazu führt, daß ein Intensivinterview meist eher dem Ziel dient, über Einzelprobleme genauere Informationen zu erhalten als der Sicherung statistischer Schlüsse.

6.1.3 Teilnehmende Beobachtung

Eine teilnehmende Beobachtung liegt dann vor, wenn der Forscher das Verhalten von Personen in ihrer natürlichen Umgebung auf geplante Art und Weise beobachtet. Dabei integriert sich der Beobachter in das Handlungsfeld der untersuchten Personen und nimmt an den Interaktionen teil. Geplant ist die Beobachtung insofern, als nur die zu untersuchenden Handlungen protokolliert werden. Der Vorteil derartiger Untersuchungen liegt besonders darin, komplexe Interaktionsstrukturen beobachten und zahlreiche Variablen mit möglichen Veränderungen in der Zeit untersuchen zu können. Oft können solche Beobachtungen nicht einfach mit einer Kamera aufgenommen werden, da dadurch die Situation verzerrt werden würde. Jeder Untersuchung geht die Erstellung eines Untersuchungsplans voraus, wobei je nach Problemkreis abzuwägen ist, ob eine exakte Beobachtung mit einer Beschränkung auf wenige Punkte einer weniger exakten Beobachtung mit umfangreicherem Material vorzuziehen ist. Im ersten Fall kann nur ein eingeschränkter Aspekt untersucht werden, wohingegen im zweiten Fall eine Aussage über Häufigkeitsverteilungen und Stellenwert schwierig wird. In der Regel sind teilnehmende Beobachtungen standardisiert, in einigen Fällen kann es jedoch notwendig sein, die Hypothesen erst während der Untersuchung selbst zu formulieren, um anschließend systematischer vorgehen zu können. Zur Auswertung werden sogenannte Beobachtungsschema verwendet, ein die Perzeption lenkender Protokollbogen. Dieser kann entweder sehr exakt auf bestimmte Interaktionen gerichtet sein, oder dem Beobachter mit etwas komplexeren Fragen mehr Spielraum für Interpretationen lassen.

Teilnehmende Beobachtungen eignen sich besonders zur Untersuchung sozialer Verhaltensweisen in der natürlichen Umgebung (beispielsweise das Verhalten Jugendlicher in Freizeitheimen). Hierbei gibt es eine Reihe von Fehlerquellen, die nach Friedrichs (90) meist nicht in der Veränderung des Feldes begründet sind, sondern eher in einer Veränderung des Beobachters selbst (wenn dieser sich beispielsweise zu sehr einzelnen Personen des Feldes zuwendet). Varianten der teilnehmenden Beobachtung ähneln, bei einer geringen Teilnahme von seiten des Beobachters, einem Experiment. Auch werden teilnehmende Beobachtungen oft durch Interviews zur Überprüfung der Validität ergänzt (Friedrichs 90, S. 303).

6.1.4 Experiment

Das (wissenschaftliche) Experiment ist eine wiederholbare Beobachtung unter kontrollierten Bedingungen mit der Veränderung einer oder mehrerer Variablen zur Überprüfung von Hypothesen. Nach Friedrichs (90) gilt das Experiment als die exakteste Form wissenschaftlicher Forschung mit den Vorteilen der genauen Hypothesenprüfung und der Mög-

lichkeit, die Versuchsbedingungen zu manipulieren, um Einfluß auf die Variablen nehmen zu können. Weiterhin können Zusammenhänge zwischen zwei Variablen gemessen werden. Um bestimmte Beziehungen unter kontrollierten Bedingungen prüfen zu können, kann der Forscher die Situation selbst herstellen oder sich in eine entsprechende Situation begeben. Im ersten Falle spricht man von einem Laborexperiment, im zweiten von einem Feldexperiment. Der Vorteil eines Laborexperiments liegt im Ausschluß möglicher Störeinflüsse. Demgegenüber ist das Feldexperiment von der Erhebungssituation abhängig, da es sich um eine Untersuchung in einer realistischen Situation handelt, die damit anfälliger gegenüber nicht antizipierten Situationen ist. Eine weitere Variante des Experiments ist die (computergestützte) Simulation, die es gestattet, Abhängigkeiten zwischen Variablen an Hand eines Modells oder auch das Modell selbst zu verifizieren.

Grundvoraussetzung für das Experiment sind eine präzise Formulierung der Hypothesen und ein Versuchsplan. Während der Untersuchung kann die Erwartung des Forschers das Ergebnis beeinflussen: Bei einem Experiment wurden von verschiedenen Forschergruppen das Lernverhalten von Ratten untersucht. Der einen Gruppe wurde dazu gesagt, die Ratten seien durch Züchtung besonders klug, der anderen, sie seien besonders dumm. Die Forschergruppe, die die „klugen“ Ratten untersuchte, erzielte bessere Lernerfolge als die andere (Friedrichs 90, S. 349). Ebenfalls kann das Verhalten des Forscher, also die Art und Weise, wie er seine Anweisungen formuliert, zu einer Verfälschung des Ergebnisses führen. Weiterhin ist es möglich, daß die Teilnehmer sich an die vermutete Erwartungen des Versuchsleiters anpassen. Hier wird von einer gewissen Künstlichkeit der Erhebungssituation gesprochen. Ein weiterer Einflußfaktor ist die Auswahl der am Experiment beteiligten Personen, die zum einen aufgrund ihrer freiwilligen Mitarbeit, zum anderen dadurch, daß sie in der Regel durch eine bewußte Auswahl bestimmt worden sind, eine Generalisierung der Ergebnisse nicht rechtfertigen. *„Wenn nur bestimmte Personen an Experimenten teilnehmen - für welche Gruppe der Bevölkerung stehen sie ein? Wenn das Experiment in einer weitestgehend kontrollierten Situation stattfindet - auf welche Situationen oder Variablen-Strukturen außerhalb des Labors sind die Ergebnisse übertragbar?“* (Friedrichs 90, S. 352). Zur Kontrolle der Künstlichkeit einer Erhebungssituation, der Reaktionen der beteiligten Personen und zur Abgrenzung von Störeinflüssen kann ein Pretest vor dem eigentlichen Experiment durchgeführt werden.

6.2 Prototyp zur intuitiven Filmplanung

Der Prototyp zur intuitiven Filmplanung ist eine Beispielanwendung der Intuitiven Schnittstelle und realisiert einige wesentliche Komponenten eines interaktiven Filmplanungssystems, wie es in einem Projekt der Kunsthochschule für Medien Köln bereits als Idee entwickelt worden ist (Fleischmann 94-1/95). Ziel dieser Anwendung ist es u.a., in der Planungsphase eines Films unterschiedliche Kameraeinstellungen an einer computer-generierten Szene testen zu können und damit die traditionelle Arbeitsweise mit Storyboard zu ergänzen (siehe auch Abbildung 2-5, Seite 27). Für den Prototyp wurde eine Beispielszene mit mehreren Objekten entwickelt (Abbildung 6-1). Der Benutzer hat die Möglichkeit, die Objekte in der Szene zu arrangieren, d.h. es ist ein Bewegen, Rotieren oder Entfernen von Objekten möglich. Dabei interagiert er typischerweise in einer Entfernung zwischen drei und fünf Metern. Die Umgebung der Intuitiven Schnittstelle, in der sich der Benutzer bewegt, ist hierbei nicht der Raum um den Computer, sondern der Realraum selbst, in den der Computer integriert ist, d.h. die gesamte Interaktionsumgebung wird zum Interface.



Abbildung 6-1: Intuitive Schnittstelle mit Beispielanwendung zur Filmplanung

6.2.1 Systemaufbau

In Abbildung 6-2 ist die räumliche Situation der Schnittstelle für den Prototyp in einem Labor der Kunsthochschule für Medien dargestellt. In einer Wand des Labors ist eine 2,60 Meter breite und 2,10 Meter hohe Rückprojektionswand eingelassen. Die Kameras des Stereoaufbaus befinden sich oberhalb der Projektionswand an den Positionen $cam1 = (0, 2.41, 0.177)$ und $cam2 = (2.90, 2.41, 0.177)$ (Angaben in Metern).

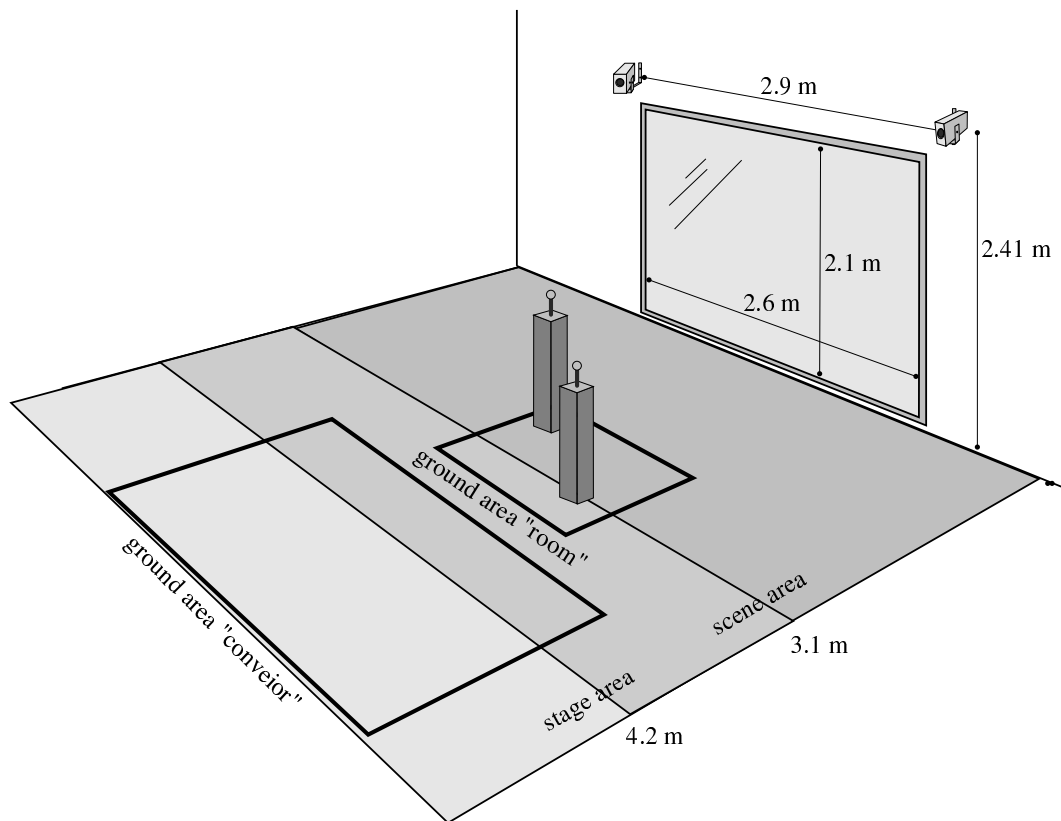


Abbildung 6-2: Räumliche Anordnung der Schnittstelle für den Prototyp

Für die meisten Operationen wird ein Sprachkommando im Zusammenhang mit der Zeigeoperation mittels des einfachen Armmodells (vgl. Kapitel *Personenverfolgung*) verwendet. Das Bewegen von Objekten ist alternativ auch durch die Verwendung von realen Platzhalterfiguren möglich. Abbildung 6-2 zeigt die zugehörige sensitive Zone für das Arbeiten mit den Platzhalterfiguren (*ground area „room“*). Weiterhin können neue Objekte aus einer Art Requisite der Szene hinzugefügt werden, die entsprechende Funktion wird beim überschreiten der Schwelle bei 4,20 Metern aufgerufen (*stage area* in Abbildung 6-2). Ein Zurückschalten in die Szene geschieht erst beim erneuten Überschreiten der Schwelle bei 3,10 Metern. So ist sichergestellt, daß der Benutzer sich frei bewegen kann, ohne die Gefahr ein ständiges Umschalten zwischen Szene und Requisite auszulösen. Für die Anwahl der Objektgruppen ist eine weitere sensitive Zone im hinteren Teil des Raumes definiert (*ground area „conveyor“* in Abbildung 6-2).

Hardware

Für den Prototyp werden zwei PCs verwendet: Einen für das Bildverarbeitungssystem und die Sprachverarbeitung, den anderen für die Darstellung der Benutzerschnittstelle. Beide PC sind über eine Netzwerkverbindung miteinander verbunden. Die Standard-Videosignale (FBAS) der beiden CCD-Kameras werden über zwei Grabber-Karten, das Sprachsignal über einen Funkempfänger und eine Soundkarte in den PC geführt. Die Darstellung der Benutzerschnittstelle übernimmt ein 3-Linsen-Projektor. Vor dem Arbeiten mit dem System muß der Benutzer ein Head-Set-Mikrofon aufsetzen und sich eine grüne Markierung mit einem Durchmesser von 11 cm anstecken. Abbildung 6-3 zeigt die Verbindung der Komponenten des Systems, eine Auflistung dieser findet sich in Tabelle 6-1.

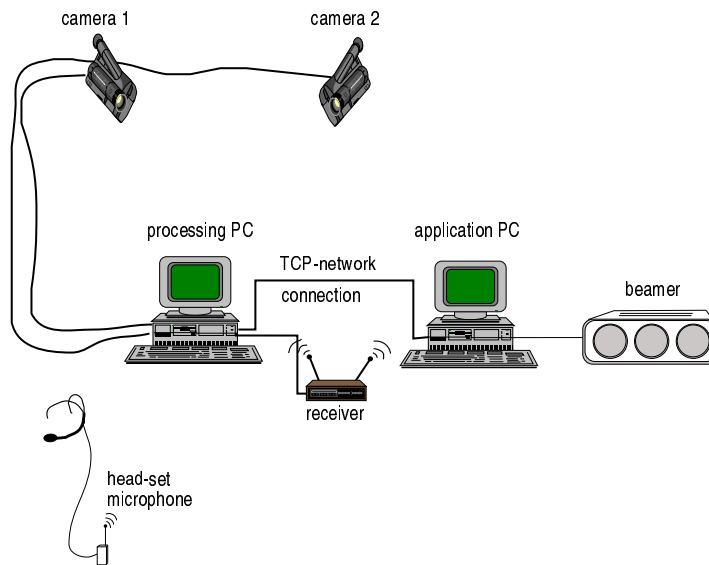


Abbildung 6-3: Technischer Aufbau des Prototyps

Tabelle 6-1: Liste der Hardwarekomponenten

Kameras	2 handelsübliche 1/3" CCD-Kameras 440.000 Pixel, 1 Lux, 6,5 mm Öffnungswinkel: 42,8 Grad in x-, 32 Grad in y-Richtung
Bildverarbeitungs-PC	200 MHz Pentium/Pro 128 MB Hauptspeicher Grafikkarte mit GL-Unterstützung (FireGL 1000) 2 x Matrox Meteor Grabber-Karten Soundkarte für Spracherkennung (Soundblaster) Netzwerkkarte
Anwendungs-PC	200 MHz Pentium 96 MB Hauptspeicher Grafikkarte (FireGL 1000) Netzwerkkarte
Mikrofon	drahtloses Head-Set-Mikrofon (Bayerdynamik)
Beamer	Großbildprojektor (Sony VPH-1271 QM)

Software

Neben dem Bildverarbeitungssystem, das auf die Bildverarbeitungsbibliothek von Matrox zurückgreift, werden das Sprachverarbeitungssystem und das Autorensystem *Director* von Macromedia benötigt. Als universelles Autorensystem, meist zur Produktion von CD-ROMs verwendet, eignet sich Director auch als Prototyping-Werkzeug zur Darstellung von Benutzerschnittstellen. Abläufe, Menüs und Interaktionen brauchen nicht programmiert zu werden, sondern können visuell definiert und ausgetestet werden. Zur Kontrolle von komplizierteren Abläufen steht eine Skriptsprache zur Verfügung. Der Nachteil dieses Autorensystems besteht darin, daß es auf der Verarbeitung von zweidimensionalen Bildern und Pictogrammen basiert. Echte dreidimensionale Szenen können daher nicht dargestellt werden, sie müssen simuliert werden. Bestimmte Erweiterungen (sogenannte Plug-Ins) erlauben zwar die Darstellung von dreidimensionalen Szenen, jedoch ist die Bildrate für bildschirmfüllende Darstellungen weit unterhalb der hier geforderten 12 Bilder pro Sekunde. Für den Prototyp wurde ein dreidimensionales Verschieben und ein Rotieren durch entsprechende Größenanpassung und Austausch der dargestellten Bilder

simuliert. Dem Nachteil fehlender Dreidimensionalität, beispielsweise zur Darstellung verschiedener Kameraeinstellungen, stand der Vorteil der schnellen Änderbarkeit und einem Testen bestimmter Interaktionen gegenüber. So konnten u.a. die im Kapitel *Interaktionstechniken* erläuterten Untersuchungen über die Darstellung von Menüs auf der Projektionswand durchgeführt werden und der Prototyp ohne die sonst üblichen Auswahlmenüs entwickelt werden. In Tabelle 6-2 sind die verwendeten Softwarekomponenten zusammengefaßt:

Tabelle 6-2: Liste der Softwarekomponenten

Bildverarbeitungs-Bibliothek	Matrox MIL 4.0
Sprachverarbeitungssystem	ASR1500 Recognition Engine (Lernout & Hauspie 96) ASR Software Development Kit 3.0
Autorensystem	Macromedia Director Version 5 für Windows

6.2.2 Beispielszene

Abbildung 6-4 zeigt die Beispielszene mit drei Figuren und zwei Objekten, die arrangiert worden sind. Das Hintergrundbild wurde nach einer Zeichnung des niederländischen Architekten und Malers Hans Vredemann de Vries (1527-1606) erstellt und zeigt einen offenen Raum in einem südländischen Ambiente. Diese Szene ist beim Prototyp vorgegeben, jedoch ist ein Austauschen der Szene leicht möglich und für eine spätere Anwendung erforderlich.

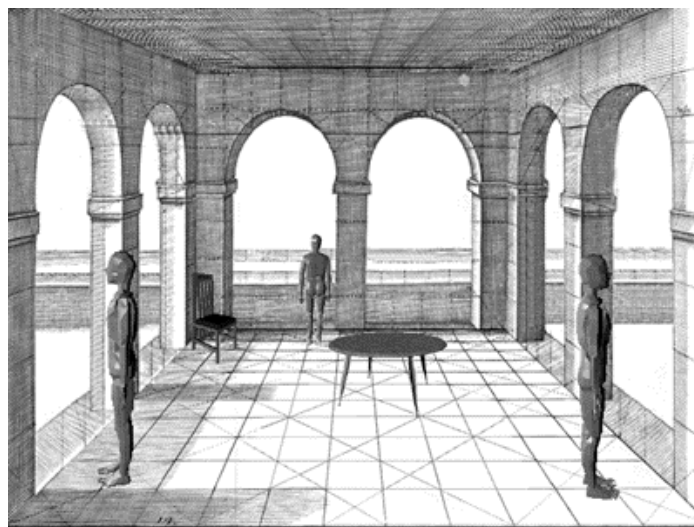


Abbildung 6-4: Beispielszene des Prototypen

6.2.3 Manipulationsmöglichkeiten

Der Prototyp ermöglicht das Selektieren und Bewegen, das Rotieren und Entfernen von Objekten der Szene und das Hinzufügen von Objekten aus einer Art Requisite. Zusätzlich lassen sich reale Gegenstände einbeziehen, um virtuelle Objekte zu bewegen. Im folgenden werden die verschiedenen Manipulationsmöglichkeiten erläutert.

Selektieren von Objekten. Durch eine Zeigeoperation des Benutzers wird ein Cursor auf der Projektionswand gesteuert. Das Selektieren von Objekten kann auf zwei alternativ zu verwendende Weisen erfolgen: Durch ein Sprachkommando und eine sogenannte Auto-Klick-Operation. Mit Hilfe des Sprachkommandos wird das Objekt selektiert, das sich unter dem Cursor befindet. Beim Testen des Prototypen hat sich jedoch gezeigt, daß es oft lästig ist, den Sprachbefehl immer zu wiederholen, besonders wenn weitere Personen im Laborbereich arbeiten. Daher wurde die Auto-Klick-Operation als Alternative angeboten. Diese Operation selektiert ein Objekt, wenn der Benutzer eine gewisse Zeit (länger als ca. eine Sekunde) auf ein Objekt zeigt und basiert auf der Annahme, daß der Benutzer normalerweise nicht ruhig auf eine Stelle zeigen würde (wenn nicht zum Selektieren von Objekten). Als Rückmeldung für den Benutzer werden verschiedene Pictogramme angezeigt, je nach dem, ob der Cursor frei ist, sich über einem selektierbaren Objekt befindet, oder das Objekt selektiert worden ist (Abbildung 6-5). Beim Selektieren wird zusätzlich ein Klick-Geräusch ausgegeben. Ein Objekt wird deselektiert durch das Sprachkommando „ok“. Falls das Objekt über die Auto-Klick-Operation selektiert wurde, ist eine Deselektion auch durch ruhiges Zeigen auf eine Zielposition möglich: Das Objekt rastet nach etwa einer Sekunde wieder aus.

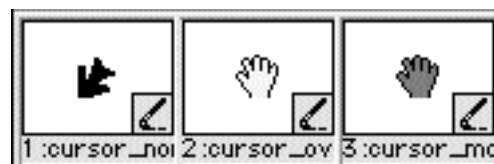


Abbildung 6-5: Verschiedene Pictogramme für Zustände des Cursors (normal, über Objekt, Objekt selektiert)

Requisite. Zu Beginn der Interaktion wird die Beispielszene mit lediglich einem Stuhl gezeigt (Abbildung 6-6). An diesem Stuhl ist es leicht möglich, die Zeigeoperation, das Verschieben und Rotieren von Objekten zu testen. Dazu arbeitet der Benutzer in dem in Abbildung 6-2 mit „scene area“ bezeichneten Bereich, typischerweise vor der Projektion in einer Entfernung von etwa 3,50 Metern.

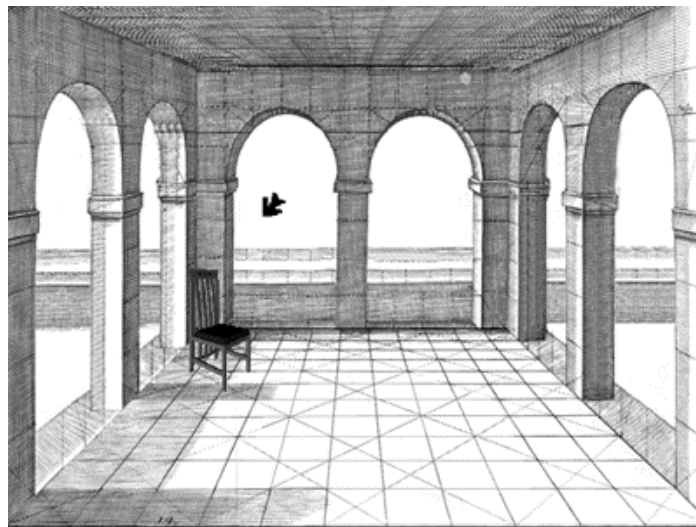


Abbildung 6-6: Anfangskonfiguration der Szene

Hat sich der Benutzer mit dem System vertraut gemacht, können weitere Objekte aus der Requisite hinzugefügt werden. Dazu bewegt der Benutzer sich aktiv in den hinteren Teil des Raumes. Beim Überschreiten der Schwelle zum Umschalten in die Requisite (4,20 Meter), fährt die Szene heraus und eine Art Bühne wird sichtbar (Abbildung 6-7a). Zusätzlich wird ein metallisch, quietschendes Geräusch ausgegeben, daß den Eindruck eines mechanischen Verschiebevorgangs beim Zurückfahren verstärkt. Um eine größere Auswahl von Objekten zu ermöglichen, werden die Objekte in Gruppen aufgeteilt (Tische, Stühle, Figuren), die dann über ein Fließband im unteren Teil der Szene dargestellt werden. Jede Gruppe wird an einer bestimmten Raumposition im Realraum abgelegt und ist abrufbar, indem der Benutzer sich an die entsprechende Raumposition begibt (Gedächtnisfunktion). In diesem Fall wird die seitliche Bewegung des Fließbandes durch die Benutzerbewegung in x-Richtung, d.h. parallel zur Projektionswand, gesteuert. Durch Zeigen auf das Fließband, wird die aktuelle Gruppe aktiviert und auf die Bühne „gefahren“ (Abbildung 6-7b). Anschließend kann ein Objekt der Gruppe wie oben beschrieben selektiert werden.

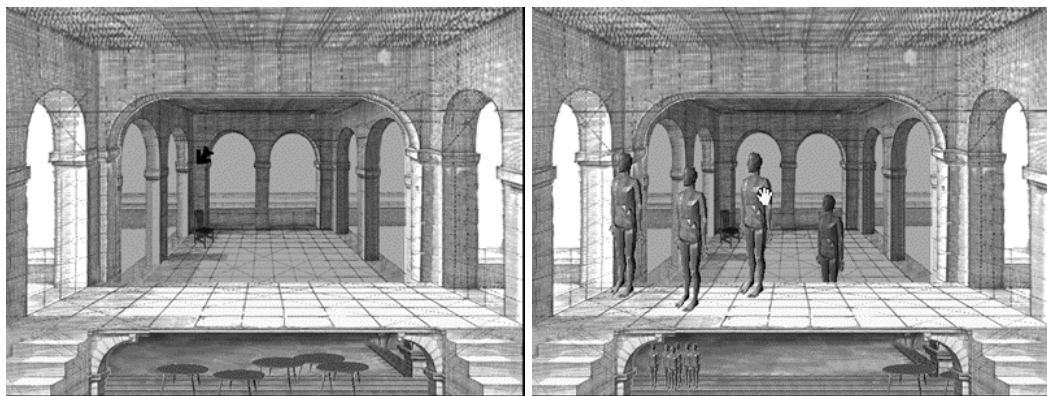


Abbildung 6-7: a) Bühnenmetapher der Requisite für das Hinzufügen von Objekten zu der Szene. Verschiedene Objektgruppen sind über das unten verlaufende Fließband aufrufbar. b) Selektierte Objekte verschwinden von der Bühne und erscheinen in der Szene.

Nachdem genügend Objekte ausgewählt und der Szene hinzugefügt worden sind, bewegt sich der Benutzer wieder nach vorne. Beim Erreichen der Schwelle für das Umschalten von der Requisite auf die Szene (3,10 Meter, vgl. Abbildung 6-2), fährt die Szene wieder zurück und der Benutzer kann die zugefügten Objekte in der Szene arrangieren. Hierbei kann der Benutzer sich frei im Raum bewegen, solange er vor der Schwelle bei 4,20 Metern bleibt.

Rotieren von Objekten. Durch Angabe eines Sprachkommandos wird das aktuell selektierte Objekt rotiert. Ist kein Objekt selektiert, jedoch der Cursor über einem Objekt, wird dieses Objekt selektiert und anschließend rotiert. Zur Rotation bewegt der Benutzer seine Hand in horizontaler Richtung nach links bzw. rechts, um eine gewünschte Stellung des Objekts auszuwählen. Das Sprachkommando „ok“ beendet die Rotationsfunktion.

Entfernen von Objekten. Durch Angabe eines Sprachkommandos wird das aktuell selektierte Objekt aus der Szene entfernt.

Einbeziehen realer Gegenstände. Neben der Verfolgung des Benutzers, werden zwei Platzhalterfiguren aus Holz, jeweils mit einer farbigen Markierung versehen, im Raum verfolgt. Die Platzhalterfiguren lassen sich über ein Sprachkommando an virtuelle Ob-

jekte koppeln. Anschließend werden die virtuellen Objekte entsprechend der Bewegung der Platzhalterfiguren im Realraum bewegt (Abbildung 6-8). Die virtuellen Objekte lassen sich hierbei innerhalb einer sensitiven Zone, die im Realraum markiert ist, steuern (siehe *ground area „room“* in Abbildung 6-2). Hierdurch ist das Arrangieren und Diskutieren der virtuellen Szene im Realraum möglich und der Benutzer erhält zusätzlich, durch das Gewicht der Holzfiguren, eine haptische Rückmeldung beim Bewegen der virtuellen Objekte.



Abbildung 6-8: Steuerung virtueller Objekte durch das Bewegen von Realobjekten

6.2.4 Sprachkommandos

Sprachkommandos unterstützen die Interaktion mit dem Prototypen und ergänzen die Zeigeoperation. Die Sprachkommandos sind in Tabelle 6-3 angegeben.

Tabelle 6-3: Liste der verwendeten Sprachkommandos

Sprachkommando	Verwendung
select	Selektieren von Objekten
rotate	Rotation von Objekten
ok	Beenden von Selektier- oder Rotationsfunktion
delete	Entfernen von Objekten aus der Szene
yellow	Verknüpfung des aktuell selektierten Objekts mit der gelb markierten Platzhalterfigur
red	dito mit rot markierten Platzhalterfigur

6.3 Forschungsplan

Jede empirische Studie ist ein aus verschiedenen Phasen bestehender Prozeß, der von der Problemidentifikation über seine Untersuchung bis zur Verwertung der Ergebnisse reicht (Friedrichs 90). Zur Planung dieses Prozesses wird ein sogenannter *Forschungsplan* aufgestellt. Er besteht aus den Arbeitsschritten *Konzeptualisierung* und der Festlegung des *Untersuchungsplan*. Die Konzeptualisierung dient u.a. dazu, das Problem zu strukturieren, Variablen zu isolieren und Hypothesen zu formulieren, der Untersuchungsplan nennt u.a. die Art der Untersuchung und beschreibt die Erhebungssituation (Friedrichs 90, S. 158). Im folgenden ist der zur Durchführung der Intensivinterviews und Experimente erstellte Forschungsplan abgedruckt. Er beschreibt das für diese Untersuchung vorliegende Forschungsproblem, formuliert die Hypothesen und zeigt den Untersuchungsplan mit dem Leitfaden für das Interview.

6.3.1 Konzeptualisierung

6.3.1.1 Forschungsproblem

In dieser Arbeit werden Methoden skizziert, die es gestatten, den Computer aus der herkömmlichen, wenig flexiblen Anordnung rund um den Monitor loszulösen und in die alltägliche Umgebung des Nutzers zu integrieren. Die intuitive Schnittstelle definiert dabei ein für die Informatik klares Forschungsproblem, für das ein Personenverfolgungssystem und eine prototypische Anwendung mit einer klaren Benutzergruppe entwickelt worden sind. Zusätzlich zur Informatik liegt ein zweites Forschungsproblem vor, bei dem es um die Frage geht, ob die ausgesuchte Benutzergruppe die Vermutungen der Anwendbarkeit des Szenarios bestätigen kann. Dazu wird eine empirische Studie anhand einer ausgewählten Gruppe von Nutzern durchgeführt, um zu überprüfen, inwieweit die vorgestellte Zielgruppe paßt und um Aufschlüsse und Anregungen zu erhalten, welche Aspekte der intuitiven Schnittstelle funktionieren und welche anders gestaltet oder verbessert werden können. Kurz, inwieweit die Schnittstelle auch empirisch bestätigt werden kann und dadurch Hinweise für die weitere Forschung ableitbar sind. Aus diesem Forschungsproblem leiten sich die Hypothesen ab, die vor dem Hintergrund mehrerer vorangegangener Gespräche mit potentiellen Anwendern entstanden sind.

6.3.1.2 Hypothesen

Im folgenden werden mehrere Hypothesen zu verschiedenen Themengebieten angegeben. Die Auflistung teilt sich in allgemeine Hypothesen und untergeordnete oder abgeleitete Hypothesen.

Kernaussage Beweglichkeit:**Die Arbeitsweise des Filmemachers ist körperbetont mit Menschen umgehend**

Hypothese	abgeleitete Hypothesen
<i>Für den Filmemacher ist es wichtig in der Entwicklungsphase auch sinnlich zu kommunizieren.</i>	- Durch eine aktive Leibbewegung wird eine adäquate Positionierung des Benutzers relativ zur virtuellen Welt erreicht und gleichzeitig die Repräsentation des Benutzers in der virtuellen Welt definiert. - Die Gedächtnisfunktion ist zum Aufrufen von Funktionen und Objekten im Raum einfach anzuwenden.
<i>Je mehr sich auch im Realraum abspielt, desto eher wird ein planerischer Umgang mit dem Medium Computer erleichtert.</i>	- Die Verwendung von Raum erleichtert den Umgang mit dem System und der virtuellen Szene. Weiterhin entsteht durch die aktive Leibbewegung ein natürliches Verhältnis zur virtuellen Szene. - Die starke Einbeziehung von Ort und Raum ist für das „Sichdraufeinlassen“ (Funktionieren der Schnittstelle) ausschlaggebend. - Die Diskussion von Szenen mit und im Realraum ist gut geeignet filmische Szenen zu besprechen.

Kernaussage Ähnlichkeit:**Ein System für Filmemacher, das auf dem Ähnlichkeitsprinzip basiert ist sinnvoll**

Hypothese	abgeleitete Hypothesen
<i>Die Arbeitsumgebung der intuitiven Schnittstelle ist für den Filmemacher adäquat.</i>	- Durch eine multimodale Schnittstelle mit Gestikinteraktion und freier Beweglichkeit wird die Interaktion mit dem Computer nicht mehr als Hindernis wahrgenommen. Eine demonstrative Gestik ist zur Lösung des Problems (Arrangieren von Szenen) geeignet. - Die Bedienung erfordert keine Aufmerksamkeit, d.h. ein ständiges Fokussieren/Defokussieren zwischen der Lösung der Aufgabe und einer symbolischen Interaktion mit dem System ist nicht notwendig. Eine Konzentration auf die eigentliche Arbeit, d.h. auf Raum und Bewegung, Ausstattung und Licht, ist möglich. - Eine Verbesserung der Filmplanung kann durch Storyboardsysteme erreicht werden, die auch Bewegungseindrücke wiedergeben (Klassische Storyboards sind ein wichtiges Kommunikationsmittel, geben jedoch Bewegungseindrücke ungenügend bzw. gar nicht wieder).

Kernaussage praktischer Nutzen für den Filmemacher:**Das System schafft Arbeitserleichterung im Sinne effektiverer Ökonomie**

Hypothese	abgeleitete Hypothesen
<i>Das System hat praktischen Nutzen.</i>	- Das vorliegende System ist (nach Erweiterungen) zum Planen von Filmszenen vorstellbar: z.B. als Storyboardsystem.

6.3.1.3 Variablen

Da es sich hier um eine explorative Untersuchung handelt, lassen sich an dieser Stelle keine Variablen angeben, die beispielsweise im Hinblick auf eine statistische Auswertung verwendet werden können. Jedoch sollen einige Variablen genannt werden, die das Funktionieren der Schnittstelle und auch die Ergebnisse des Interviews beeinflussen.

Bei der Auswahl des Benutzertypus wurde versucht Regisseure verschiedenen Alters (31 - 57 Jahre), Geschlechts und aus verschiedenen Tätigkeitsbereichen (Autorenfilm bis Experimentalfilm) auszuwählen. Weiterhin unterschied sich die Vorbildung hinsichtlich eines früheren Kontakts mit dem Computer erheblich. Bei einer Auswertung ist trotz dieser Streuung zu berücksichtigen, daß die Auswahl der Teilnehmer in starkem Maße die Ergebnisse determiniert. Rückschlüsse auf allgemeingültige Aussagen sind hier, auch aufgrund des geringen Umfangs der Stichprobe, nicht möglich, die Untersuchung gestattet jedoch eine erste Exploration der Intuitiven Schnittstelle für einen Einsatz zur Filmplanung. Ein weiterer Einflußfaktor auf mögliche Antworten ist die generelle Einstellung der Person gegenüber dem experimentellen Problem oder auch eine Schwellenangst. Neben diesen personbedingten Faktoren beeinflusst auch die Erhebungssituation selbst die möglichen Antworten, d.h. die Art des Raums, der Umfang der physischen Bewegungsmöglichkeit, technisches Gerät, wie Bildschirm, Projektion und Mikrofon, und die Sichtbarkeit der Schnittstelle selbst (etwa durch die Verwendung von Kameras).

6.3.2 Untersuchungsplan

6.3.2.1 Art der Untersuchung

Ziel der Studie ist eine explorative Untersuchung der Intuitiven Schnittstelle. Als Untersuchungsmethode wird das Intensivinterview ausgewählt, da über den gewählten Objektbereich in seiner speziellen Kombination keine vergleichbare Untersuchungen vorliegen. Die Studie dient in erster Linie der Erlangung von Kenntnissen über den Objektbereich und ist daher eher beschreibender als analytischer, d.h. Hypothesen prüfender, Natur. Die Auswahl der Variablen wurde mehr oder weniger intuitiv vorgenommen, und es liegen nur wenige eher vage Annahmen zugrunde, die in Form von Hypothesen formuliert worden sind. In das Interview wurde ein Experiment integriert, um den Befragten die Möglichkeit zu geben, das System kennenzulernen. Hierbei wurden die im letzten Abschnitt beschriebenen Manipulationsmöglichkeiten anhand der Beispielszene demonstriert und anschließend von den Teilnehmern selbst durchgeführt. Wichtiger als hieraus erhoffte Aufschlüsse über Probleme einer gewählten Interaktionsart zu erlangen, ist die Notwendigkeit der Teilnehmer, die Schnittstelle leibhaftig erfahren zu können, um so die anschließenden Fragen beantworten zu können.

Das Ergebnis der Studie kann als Grundlage für weitere Studien und für eine präzisere Hypothesenformulierung herangezogen werden. Die Konsequenz aus dieser Art der qualitativen Untersuchung ist eine sehr geringe Stichprobe⁸ von $n = 6$ und einer Durchführung des Interviews ohne straffen Rahmen und Vorgaben. Lediglich ein Leitfaden mit Schlüsselfragen und Eventualfragen ist vorgegeben. Bei der Beantwortung der Fragen ist jedoch mehr Wert auf den Gesprächsverlauf und die Beziehungsebene zu legen, als auf eine umfassende Beantwortung aller Fragen. Dementsprechend ist keinerlei Exaktheit bei der Prüfung der Hypothesen vorgegeben.

⁸ Die Stichprobe bezeichnet hier die gezielte Auswahl von Elementen (n) aus der Gesamtheit aller Elemente (N).

6.3.2.2 Datenerhebung

Die Durchführung der Interviews erfolgt in einem Computerlabor der KHM, in dem die Intuitive Schnittstelle in einem hinteren Bereich installiert ist. Der Zeitpunkt ist nach Möglichkeit so zu wählen, daß die Frequentierung des Labors gering ausfällt. Zur Unterstützung und Kontrolle soll das gesamte Interview auf Tonband (DAT-Rekorder) und ein Übersichtsbild auf Video aufgezeichnet werden, nach Möglichkeit sollte ein Beisitzer anwesend sein. Die Erhebungssituation sollte auch durch Vorgespräche mit dem Befragten möglichst angenehm gestaltet werden und eher Gesprächscharakter haben. Ein Interesse der Befragten liegt insofern vor, als die konkrete Anwendung der Intuitiven Schnittstelle als Filmplanungssystem direkt aus dem Berufsumfeld der Befragten kommt.

6.3.2.3 Stichprobe

Die Stichprobe besteht aus $n = 4$ Regisseuren aus der KHM und $n = 2$ außerhalb der KHM. Diese Zahl ist für ein Intensivinterview, das hauptsächlich explorativen Zwecken dient, adäquat.

6.3.2.4 Pretest

Ein Pretest wird nur insoweit durchgeführt, daß die Interviewsituation mit einigen Studenten oder Mitarbeitern ($n = 2$ bis $n = 5$) der KHM aus dem Bereich Film/Fernsehen durchgespielt wird. Dieser Pretest dient in erster Linie der Überprüfung der Verständlichkeit der Fragestellungen, der Adäquatheit von Interviewsituation und Dauer sowie der Schulung des Interviewers.

6.3.2.5 Schlußfolgerungen

Ziel ist die präzisere Formulierung von Hypothesen und/oder die Ableitung von Richtlinien für den Einsatz der Intuitiven Schnittstelle im Arbeitsbereich künstlerischer/gestalterischer Bildproduktion. Auch das Aufzeigen falscher Annahmen und Hypothesen kann als Input für weitere Untersuchungen dienen.

6.3.3 Leitfaden

Interview-Einführung:

- Begründung für das Interview.
- Ausdruck der Bitte, zu jeder Frage die persönliche Erfahrungen und Ansichten ganz offen anzugeben.
- Geplante Dauer des Interviews (ca. 1 Std.)
- Erlaubnis der Tonband-/Filmaufzeichnung.

Fragen (Teil 1):

Die Fragen sind nach Themengebieten sortiert. Links steht jeweils die aus den Hypothesen allgemein zu klärenden Frage, rechts die Liste der konkreten Fragen an den Befragten.

allgemeine Angaben

<i>Daten des Befragten</i>	- Alter - Können Sie mir kurz als Stichworte die Betätigungsfelder nennen in denen Sie als Regisseur arbeiten oder gearbeitet haben.
<i>Wie ist der Kenntnis- bzw. Erfahrungsstand des Befragten?</i>	- Bitte beschreiben Sie, mit welchen technischen Geräten Sie beruflich bisher zu tun hatten, d.h. mechanische sowie z.B. professionelle Videotechnik oder Computer? - Benutzen Sie Computer, Seit wann? Für welche Tätigkeiten?

zur Arbeitssituation

<i>Was sind die Hauptmerkmale des Arbeitsprozesses eines Regisseurs?</i>	- Beschreiben Sie kurz wie Sie selber arbeiten bei der Realisierung eines Films, also den Ablauf vom Exposé über Drehbuch zur Produktion selbst, d.h. Modifikationen am Set.
<i>Entspricht das Szenario der intuitiven Schnittstelle der natürlichen Arbeitssituation eines Regisseurs?</i>	- Beschreiben Sie Ihre Arbeitssituation an einem Set beim Arrangieren einer Szene. Nennen Sie Örtlichkeiten, Formen der Kommunikation, Gesten und Körperhaltungen. - (Eventualfrage: wie verhält es sich bei Proben?)

Demonstration des Systems:

Hier wird kurz die intuitive Schnittstelle und das Beispielsystem zum Arrangieren von Szenen erklärt und demonstriert. Anschließend hat der Befragte die Möglichkeit das System selbst zu testen. Der Pretest zeigte, daß folgende Punkte für den Ablauf der Demonstration und das Verständnis durch den Benutzer wichtig sind:

- Die Interviewstühle sind aus dem Interaktionsbereich zu entfernen.
- Wichtig ist eine präzise Einleitung mit Verdeutlichung, daß es sich um einen Prototyp handelt.
- Beide Möglichkeiten der Selektion sind genau zu erklären.
- Die Funktion des Fließbandes in der Requisite ist klar zu erläutern.

Fragen (Teil 2)*zum System*

Ist die körperbetonte Kommunikation in der Planungsphase wichtig?

- Beschreiben Sie, wie Sie das Wechseln zwischen Szene und Requisite durch Ihre Körperbewegung empfunden haben?
- Fanden Sie es einfach, die realen Orte aufzusuchen, um virtuelle Objekte anzuwählen? Beschreiben Sie Ihren Eindruck.
- Wie haben Sie die Einbeziehung von Realgegenständen zur Manipulation von virtuellen empfunden?

Inwieweit entspricht die Arbeitsweise mit dem System der gewohnten Arbeitsweise eines Regisseurs

- Beschreiben Sie, ob Sie sich mehr auf die Gestaltung der Szene konzentriert haben oder auf die Bedienung des Systems?
- Inwieweit entspricht die Arbeitsweise mit dem System Ihrer gewohnten Arbeitsweise als Regisseur bzw. inwieweit nicht?
- *(Eventualfrage: Beschreiben Sie den Unterschied der Positionierung eines Regisseurs gegenüber der virtuellen Szene im Vergleich zu einem realen Set.)*

Praktischer Nutzen

- Ist das vorliegende System zur Filmplanung geeignet. Wenn nein, warum nicht. Wenn ja, was könnte verbessert werden.
- Welche Vor- und Nachteile hat für Sie ein interaktives Storyboardsystem?

Fragen (Teil 3)*ergänzende Fragen*

Weiterentwicklungen

- Ist die vorliegende Szene für Sie glaubwürdig?
- Bei einer Weiterentwicklung des Systems, sollte man Ihrer Meinung nach die Darstellungen von Objekten konkretisieren oder eher abstraktere Darstellungen wählen?
- Inwieweit können Sie sich vorstellen, daß man auch die Arbeit mit Schauspielern mit einem solchen System vorbereiten kann?

weiterhin ist zu notieren:

- Datum, Ort, Tageszeit, Dauer, Beisitzer (wenn vorhanden)
- Anwesenheit Dritter
- Beurteilung des Verhalten des Befragten: Interessiertheit, verbale Flüssigkeit oder Aufgeregtheit.
- Beurteilung des Verhalten des Interviewers (Selbsteinschätzung)

6.4 Durchführung

6.4.1 Pretest

Für die Untersuchung wurde ein Pretest mit $n=2$ Mitarbeitern des Bereichs Film/Fernsehen der Kunsthochschule für Medien durchgeführt. Es zeigte sich, daß keine wesentlichen Änderungen am Leitfaden notwendig waren und auch die angestrebte Dauer von ca. einer Stunde eingehalten werden konnte. Aufgrund des Pretests konnten kleine Details bei der Interaktion mit dem Prototyp angepaßt und der Interviewablauf optimiert werden. Aufgrund der Videoaufzeichnung und der Erfahrung aus dem Pretest wurde auf einen Beisitzer bei den Interviews verzichtet.

6.4.2 Befragte Personen

Es wurden fünf Regisseure und eine Regisseurin aus verschiedenen Altersgruppen (31 bis 57 Jahre), und verschiedenen Berufszweigen der Filmregie befragt. Die von ihnen entwickelten Filme decken einen weiten Bereich von Autorenfilm und Spielfilm, über Dokumentarfilm und Musikverfilmungen bis hin zu experimentellen Filmen, die teilweise ohne Drehbuch entstanden sind, ab. Entsprechend unterschiedlich fällt die Arbeitsweise der einzelnen Regisseure aus, wobei sich bestimmte Grundmuster bei der Planungsphase eines Films wiederholen und sich in die verschiedenen Filmkategorien einordnen lassen. So ist die Planung mit Storyboards und/oder eine genaue Besprechung mit Kameramann und Ausstatter bei (nicht improvisierten) Spielfilmen absolut notwendig, hingegen bei Dokumentarfilmen oft gar nicht möglich, da bestimmte Szenen, wie Interviews nicht geplant werden können und sollen. Die vorhandenen Vorkenntnisse im Bereich Computer waren ebenfalls stark unterschiedlich und variierten zwischen dem Benutzen des Computers als Schreibmaschine bis hin zur Erfahrung mit computergenerierten Szenen mit Hilfe von 3D-Animationsprogrammen. Es folgt eine tabellarische Aufzählung der Personen in Form von kurzen Arbeitsprofilen:

Profil Person A

Alter	45 Jahre
Beruf	Film- und Fernsehregisseur, Professor für Regie
Filme	fast ausschließlich Spielfilm, Werbung, Serie
Technik	alle Geräte zur Filmaufnahme (z.T. auch Video), Schneidetisch bis Avid
Computer	seit 1989, hauptsächlich für Texte, Schnitt, gering zum Zeichnen und Recherche
Arbeitsweise	1) Regie am Rechner überarbeiten (in der Regel fremde Drehbücher) 2) Für die Vorbereitungsphase Arbeit mit Storyboard (eher selten, da schlechter Zeichner), oder mit Grundrißzeichnungen, da schneller. Hier werden Kamera mit Beschreibung der Einstellgröße und Kamerabewegung, Darsteller evtl. mit Hinweis auf Dialogstelle, und Bewegungen eingezeichnet. Ergänzende Verwendung von Fotos der Orte. 3) Inszenieren der Szene, Arbeit mit Schauspielern, Arrangieren mit Worten
Zitat	„Die Vorbereitungsphase für den Dreh ist die wichtigste Phase“.

Profil Person B

Alter	31 Jahre
Beruf	Kurzfilmregisseur, Regieassistent Spielfilmregisseur
Filme	Kurzfilm, Dokumentarfilm, Experimente
Technik	3-Maschinenschnittplatz, Avid, Harry, alle Arten von Kameras, Videosysteme: Hi8, Tonbereich (kein Motion Control)
Computer	Zum Drehbuchschreiben (mit Word), für Email, etwas DTP und Musikprogramm.

Arbeitsweise	1) Storyboard mit Kameramann und Regieassistent optisch durcharbeiten, d.h. für jede Szene eine Idee entwickeln (noch nicht fix). 2) Casting der Hauptdarsteller, dann Nebendarsteller (die Gruppe muß funktionieren) 3) Ausstattungsplanung, Auswahl der Sets, der Orte im Gespräch mit der Produktionsleitung und dem Ausstatter. 4) Probe der Schlüsselszenen mit Schauspielern und Kameramann, jedoch ohne Kamera, um die Auflösung zu korrigieren. 5) Stellprobe fürs Licht, Anlaufprobe, Dreh
Zitat	„Der Regisseur hält den roten Faden, an dem gezogen werden darf, aber er darf Ihn sich nicht aus der Hand nehmen lassen“.

Profil Person C

Alter	54 Jahre
Beruf	Regisseurin, Professorin für Filmregie
Filme	8 Dokumentar- und Spielfilme (Autorenfilm), teilweise selbst gedreht und produziert
Technik	Erster Kontakt vor 20 Jahren mit der Schreibmaschine, sonst Kameraführung, Hi8, aber auch sehr viel Handarbeit (schreiben),
Computer	schon früh für die Textverarbeitung, Internet ist zu langweilig und zeitraubend, die kreative Arbeit ohne Computer
Arbeitsweise	1) Erstellung des Drehbuchs allein oder mit einem Coauthor 2) Zusammenstellen eines Teams (oft gibt es vorher schon Gespräche) 3) Reise zu Drehorten mit dem Ausstatter (Panoramafotos), Erstellung von Zeichnungen und Plänen. 4) Einbauen von Kameraeinstellungen (Storyboard im besten Fall), durchsprechen mit Kameramann, Regieassistent und Produktionsleiter. 5) Proben mit Schauspielern kurz vor dem Dreh, evtl. auf einer Bühne, ohne Kamera, ohne Licht 6) Bei der Dreharbeit ist viel Klarheit und Geduld notwendig. Es ist der Versuch, die Abteilungen zu kombinieren, Stimmung machen.
Zitat	„Regisseure bewegen sich nicht gerne“

Profil Person D

Alter	31 Jahre
Beruf	Dokumentar- u. Kurzfilmregisseur
Filme	Dokumentarfilme, Kurzspielfilme
Technik	SVHS-Kamera, 3-Maschinenschnittplatz, Avid, Dolly-Schienen, Lampen, Tonmischer, Stopptrick, Spezialeffekte mit Lastenaufzug und zwei Kränen.
Computer	Verwendung seit 87/88 als Schreibmaschine, dann für Grafikprogramme (Designstudio), kein Internet, eigener Rechner jedoch ohne Modem und CD-ROM.
Arbeitsweise	1) Drehbuch wird, aus einer Geschichte, mit visuellen Keys über Exposé und Treatment erstellt. 2) Skizzen für die Auflösung anfertigen und mit dem Kameramann durchsprechen, von wichtigen Stellen ein Storyboard anfertigen (dies muß bei Spezialeffekten genau sein). Arbeit auch mit Floorplan, Modellen und Modellfiguren. Wenn die Orte bekannt sind, wird dort mit dem Kameramann gesprochen oder auch 360-Grad-Aufnahmen und ein Video erstellt. 4) Nach dem Casting: Proben mit Schauspielern (oft kurz vor Dreh, obwohl eigene Probenzeit besser wäre). Es wird darauf geachtet, was die Schauspieler einbringen können, damit es natürlich wird. 5) Beim Dreh Konzentration darauf, daß alles läuft, daß Schauspieler in guter Verfassung sind.
Zitat	„Der Dreh ist das Feld, in dem sich die Schauspieler bewegen“

Profil **Person E**

Alter	57 Jahre
Beruf	Experimentalfilm-Regisseur und Produzent, oft Kameraarbeit, stets Schnitt.
Filme	Spielfilmähnliche Dokumentarfilme, experimentelle Spielfilme, Filme ohne Drehbuch
Technik	Mechanische Filmkameras, Arbeit mit mehreren (Timecode-gesteuerten) Kameras, Videokameras und Trickkameras.
Computer	Verwendung seit Ende 70er, erst nur im Büro, jetzt als Office-Computer zu Hause, Entwurf eines Solarkraftwerk mit Auto-CAD, etwas Erfahrung mit 3D-Animationsprogrammen.
Arbeitsweise	Prinzipiell keine durchgängige Arbeitsweise, da lebenslang improvisiert und oft improvisierte Dialoge verwendet wurden: 1) Der Film ist mit einer Idee am Morgen schon vollständig gegenwärtig. 2) Geeignete Personen suchen, Recherchereise zum Drehort durchführen, ein paar Dialoge oder ein Drehbuch schreiben, jedoch mit häufigen Änderungen. 3) Nun entwickelt sich der Film durch umherreisen und Improvisation. 4) Der Dreh wird durch Gespräche geplant, ansonsten improvisiert: Arbeit geschieht direkt mit den Schauspielern, dazu den Kameramann einbeziehen und kurz über die Auflösung sprechen. Der Rest ergibt sich aus der Situation. Geplant werden die optimalen Kamerastandpunkte, und die Entscheidung ob z.B. eine Schnitt-Gegenschnitt-Montage durchgeführt wird, d.h. die Syntax des Films wird besprochen. 5) Der Film verfestigt sich anschließend im Schnitt
Zitat	„Das Irre am Drehen ist, daß man es mit Realpersonen zu tun hat“

Profil **Person F**

Alter	38 Jahre
Beruf	Kurzfilm- und Musikfilm-Regisseur
Filme	Kurzfilme (experimentell), Dokumentarfilme, Videoclips, kurze Spielfilme, hauptsächlich aber Musikverfilmungen.
Technik	Alle Kameras, Umatic- und Beta-Schneidetisch, Computerschnitt nur mit Cutter, 16/35mm Schnitt, HighSpeed-Kameras, 6 Kameras gleichzeitig.
Computer	Notebook wird als Schreibmaschine und für Tabellenkalkulation verwendet, sonst keine Computer („ich habe die bis heute nicht begriffen“)
Arbeitsweise	1a) Der Dokumentarfilm ist oft nicht festgelegt, es gibt kein Drehbuch, d.h. es werden eine praktische Recherche durchgeführt, die Drehorte festgelegt und die Personen, die man treffen will, aufgesucht. 1b) Bei einer Ballett-Verfilmung oder einem Musikfilm sind Bilder und Situationen relativ fest, d.h. es gibt eine Art Drehbuch: mit Stoppuhr (Ton ist schon vorhanden), das Bild wird aufgemalt, die zu verwendende Optik bestimmt. 2) Zusammensetzen mit Kameramann und Ausstatter (inzwischen geht das von selbst und der Kameramann hat große Freiheiten). 3) Am Drehort entwickelt sich immer mehr, da Ideen dazukommen und die Schauspieler auch Ideen einbringen. Für Licht am liebsten natürliches verwenden. 4) Die Montage rettet oft Dinge, die sich beim Dreh verschoben haben.
Zitat	„... meist wird's anders als geplant, aber trotzdem gut“

6.4.3 Interviewsituation

Für die Durchführung der Interviews wurde der hintere Bereich des Labors für Dritte durch fahrbare Trennwände provisorisch gesperrt, die Anwesenheit weiterer Personen im Laborbereich hatte dadurch keinerlei Auswirkungen auf den Verlauf der Interviews. Direkte Störungen waren nur punktuell und haben den Ablauf nicht beeinträchtigt. Die Dauer der Interviews variierte bei vier Interviews zwischen einer und 1:15 Stunden, zwei Interviews dauerten ca. 1:30 Stunden aufgrund der Erzählfreude der Gesprächspartner. Bei der Demonstration des Systems, wie auch bei der anschließenden Möglichkeit für die Regisseure, das System zu testen, war der Interviewer stets anwesend und hat erklärende

und unterstützende Hilfestellung geleistet. Dieses war besonders dann notwendig, wenn das System aufgrund von falscher Arm- oder Körperhaltung nicht funktionieren konnte. Die hieraus entstehende Beeinflussung der Erhebungssituation ist nicht zu unterschätzen und muß auch bei der Auswertung berücksichtigt werden. So kann z.B. eine beobachtete relativ steife Körperhaltung des Benutzers bei der Bedienung des Systems auch auf diese Faktoren zurückgeführt werden (der Benutzer möchte nichts falsch machen).

6.4.4 Auswertung

Die Auswertung orientiert sich an den Kernaussagen und den Hypothesen des Forschungsplans. Bei der Bewertung der einzelnen Aussagen ist stets zu berücksichtigen, daß sowohl die Teilnehmer, der Beobachter bzw. Interviewer als auch das Angebot des Prototypen die Ergebnisse determinieren. So entscheidet die Qualität der Teilnehmer über die möglichen Schlußfolgerungen einer empirischen Untersuchung. Beispielsweise wird ein analytisch arbeitender Dokumentarfilm-Regisseur eine Raumsituation völlig anders interpretieren und empfinden als ein Actionfilm-Regisseur, wodurch negative Aussagen zum Teil relativiert werden und daher auch dementsprechend gewichtet werden müssen. An dieser Stelle lassen sich also lediglich Hinweise und Vermutungen für bestimmte Richtungen der Systementwicklung formulieren, die gegebenenfalls in weiteren Studien verifiziert werden müßten.

Zur Auswertung von Interview und Experiment wurden die Videobänder betrachtet und eine Protokollierung der Aussagen vorgenommen. Beim Experiment wurden zusätzlich Auffälligkeiten bezüglich der Bedienung, der Körperhaltung oder sonstige Schwierigkeiten protokolliert. Diese Aussagen wurden in einer Tabelle gesammelt, um sie besser vergleichen und Übereinstimmungen leichter herausarbeiten zu können. Es folgt eine Zusammenfassung der Ergebnisse dieser Auswertung.

6.4.4.1 *Beweglichkeit: Sinnliche Kommunikation und Realraum*

Die Nutzung von Raum als Teil der Schnittstelle wurde unterschiedlich empfunden: Beim Aufrufen von Funktionen im Raum, wie etwa das Heraustreten aus der Szene zum Aufruf der Requisite, ist die Meinung nicht einheitlich, sie schwankt zwischen einer Ablenkung (Person A) bis hin zu einem Wohlfühlen (Person E). Einige Personen sind der Meinung, daß die Bewegung eine zusätzliche Konzentration erfordere und daher von der eigentlichen Aufgabe eher ablenken würde, andere empfanden den Raumeindruck eher verstärkt (Personen C, D, F). Der hier empfundene organische Eindruck könnte verstärkt werden, wenn verschiedene Kameraperspektiven möglich und diese durch Benutzerbewegung steuerbar sind. Beim Aufrufen von Requisiten im Raum (Gedächtnisfunktion) waren alle Befragten sich einig, daß dies eher mühsam sei und sie ein Menü, eine Auswahlleiste auf dem Bildschirm bevorzugen würden, das dann durch Sprache oder auch Bewegung aufrufbar wäre. Von den meisten wurde das Stehen und physische Gehen als nicht angenehm betrachtet („*Regisseure bewegen sich nicht gerne*“, Person C). Bei der Beurteilung dieser Aussage muß allerdings der befragte Personenkreis berücksichtigt werden, so gibt es auch Regisseure, die vorwiegend körperbetont arbeiten. Auch könnten sich vollständig andere Eindrücke und Antworten ergeben, wenn beispielsweise ein Regisseur für Actionfilme befragt werden würde. Der negative Eindruck kann auch in der bestimmten Art und Weise, wie hier Objektgruppen im Raum auszuwählen waren, begründet sein und kann ebenfalls anders ausfallen, wenn andere oder verschiedene Interaktionsformen im Raum möglich sind oder allein sich die Benutzerschnittstelle anders präsentiert. Die hier dargestellte Szene hat trotz der simulierten dreidimensionalen Struktur einen eher flachen 2D-Charakter, so daß eine Assoziation mit einem bekannten menübasierten Desktop-System naheliegt. Sind etwa Kamerabewegungen in einer echten 3D-Szene, mit Blickwinkeländerungen durch Wechsel des Betrachterstandortes möglich, kann sich unter Umständen

eine viel stärkere sinnliche Bindung zum Realraum ergeben. Auch die Wahl der Szene kann dies beeinflussen: So ist die Situation bei dem hier verwendeten halboffenen Raum mit einem südländischen Ambiente eine andere als bei einer komplexen Außenszene mit Spezialeffekten, bei der sich die räumliche Frage auf ganz andere Weise stellen würde. Die Erwähnung einer Auswahlleiste oder eines Menüs kann auch so interpretiert werden, daß eine diskrete Auswahlmöglichkeit bevorzugt wird vor der Notwendigkeit, Einfluß auf einen Prozeß zu nehmen, wie er sich bei dem Fließband mit den Objektgruppen präsentiert. Trotz der negativen Aussage über die körperbetonte Bewegung, äußerten sich bis auf Person A alle Personen positiv über das erfahrene Raumgefühl durch die Leinwand und die Positionierung des Benutzers.

Trotz der genannten Einschränkungen aufgrund des befragten Personenkreises und der spezifischen Art der Rauminteraktion ist hier die Schlußfolgerung, daß eine sinnliche, körperliche Kommunikation möglicherweise für einen Regisseur nicht zwingend notwendig ist, da Regie oft eine eher geistige Arbeit ist. Planen ist ein rationaler, analytischer Prozeß, der auch im Zweidimensionalen stattfinden kann, eine Integration des Raumerlebnisses in die Simulation erscheint daher für erlebnishafte Prozesse, in denen Emotionen eine Rolle spielen, adäquater. Ob das Raumerlebnis für bestimmte Personengruppen hilfreich wäre, konnte mit dem System nicht geklärt werden. Dazu wäre eine weitere Untersuchung unter Einbeziehung von verschiedenen „echten“ 3D-Szenen und mit einem erweiterten Personenkreis notwendig. In Bezug auf die Bedienbarkeit konnte festgestellt werden, daß der Realraum allein den Umgang mit dem Computer nicht erleichtert. Dabei scheint es, daß viele Menschen inzwischen schon auf die Arbeit mit Bildschirm und Maus sozialisiert sind, da auch Nicht-Computerleute durchaus Anwendungen mit Mausbedienung vorschlagen. Die Konsequenz hieraus ist eine mögliche Vereinfachung des Systems: Für bestimmte Bereiche der Filmplanung genügt unter Umständen eine traditionelle Desktop-Anwendung mit Maus und Bildschirm. Das System wird dadurch leichter aufbaubar und transportabel und ist nicht an eine stationäre Laborumgebung gebunden. Für Situationen, in denen das Raumgefühl oder eine große Darstellung zur Diskussion mit mehreren Personen erwünscht ist, kann ein stationärer Aufbau mit Projektionswand verwendet werden.

6.4.4.2 Ähnlichkeit: Adäquate Arbeitsumgebung

Die Frage der Ähnlichkeit der Arbeitsumgebung wurde von allen Befragten negativ beantwortet, je nach Betätigungsfeld waren die Aussagen von „nicht so ähnlich“ (Personen A, B, D) bis hin zu „überhaupt nicht ähnlich“ (Personen C, E, F). Dies mag zum einen an der oben erwähnten Determinierung der Ergebnisse durch die Teilnehmer liegen (als Begründung stellte Person C heraus, daß sie sich zur Planung hinsetzen und reden würde, Person F meinte, für den Dokumentarfilm sei es nicht brauchbar), zum anderen auch an der eher theaterhaften Darstellung der Szene und Bühne. Der Prototyp liefert lediglich eine sogenannte *Totale* auf die Szene, die als szenisches Element im Film eher selten vorkommt, wodurch vier der Befragten die Assoziation mit einer Anwendung für das Theater vollzogen (Personen A, B, D, E). Person D begründete die Aussage damit, daß sie eher auf die Frage konzentriert gewesen wäre, wo die Kamera stand, und bei der Planung eines Filmes mehr in Aufsichten und Kameraeinstellungen denken würde. Zur endgültigen Bewertung dieser Frage müßte das System also um Kameraansichten und die Möglichkeit, den Blickwinkel zu variieren, erweitert und einer erneuten Prüfung unterzogen werden.

Die Verwendung von Realgegenständen wurde durchweg als sehr gut empfunden, da sie schnell handhabbar, und praktisch wären (Person A). Objekte lassen sich sehr schnell durch die Schachfiguren-ähnlichen Platzhalter bewegen. Auch würde der sinnliche Eindruck durch die Schwere der Platzhalter verstärkt (Person C). Vier Personen sagten, sie

könnten sich diese Technik als Modell vorstellen, d.h. in der Größe eines Schachspiels oder einer Puppenstube (Personen A, B, D, F). Zwei Personen gaben den Hinweis, daß wenn das System feinfühlig genug und einfach zu bedienen wäre, eine virtuelle Manipulationsmöglichkeit ausreichen würde (Person E), z.B. über die Maus, denn dann würde man nicht noch Schachfiguren bewegen wollen (Person C). Alle anderen Personen konnten sich ein derartiges System als Ersatz der Arbeit mit Modellen gut vorstellen. Auch äußerte sich Person A hier positiv, ebenso zu den Wortbefehlen wie „rotate“, als Entsprechung zur Arbeitssituation eines Regisseurs.

Die Arbeit am Set besteht zu einem großen Teil aus der Arbeit mit den Schauspielern. Hierbei werden Schauspielerbewegungen ab drei Personen schon sehr komplex. Oft fehlt die Vorstellung von der Verteilung der Personen im Raum, daher ist dies schwierig, exakt vor auszuplanen und es werden Proben gemacht. Eine Besonderheit ist, daß sich derartige Bewegungen nicht schrittweise aufbauen lassen, da die vielen Interaktionsmöglichkeiten hochgradig komplex sind, was gegen eine Planbarkeit solcher Bewegungen mit einem Filmplanungssystem spricht. Würde man nämlich eine Szene durchplanen, würden sich durch den Faktor Mensch schon bei der Probe Konstellationen erneut verändern. Auch spricht gegen eine Anwendung des Systems, daß Schauspieler sich nicht so gerne vom Regisseur festlegen lassen. Schauspieler entwickeln eine Aktion und daraus ein Gefühl, sie müssen aus sich heraus versuchen in der Szene zu funktionieren (Person D). Schauspieler, das sei das besondere am Dreh, sind eben Menschen, die stelle man nicht einfach irgendwo hin (Person C). Anders verhalte es sich mit Schauspielern beim Theater, die eher noch chiffrierte Personen seien (Person E). Trotzdem ist für alle Befragten die Planung der eigenen Arbeit mit den Schauspielern, beziehungsweise insbesondere die Planung der Szenenauflösung mit dem Kameramann, vorstellbar. Für Auflösungen allein vor der Kamera könnten die einzelnen Bewegungen der Schauspieler simuliert werden, die Produktion würde dadurch durchsichtiger werden, Kosten könnten besser kalkuliert werden (Person F). Auch bei komplizierten Szenen ist ein Planungswerkzeug hierfür denkbar. Zusätzlich zu den Bewegungen könnte man auch Dialoge sprechen lassen und würde eine Idee für das Timing der Szene bekommen (Person A). Dagegen spreche erneut, daß sich durch den Faktor Mensch nie alles vorausplanen lasse und dies auch nicht sinnvoll sei, weil die Gefahr bestehe, daß Personen ohne Selbstvertrauen die leblose Planungsszene in den Film übersetzen (Person F). Andererseits sei es auch nicht sinnvoll so viel vorgeben, wenn man Ideen von den Schauspielern haben möchte. Je präziser die Vorgabe ist, desto weniger können sich die Schauspieler selbst einbringen. Ein Schauspieler bringe stets selbst noch etwas in die Szene ein (Personen B, D, E, F). Eine Möglichkeit wäre das Aufnehmen (Rotoskopie) der Bewegungen der Schauspieler bei Proben und ein anschließendes Ausprobieren von Kameraeinstellungen am virtuellen Set mit den Schauspielerbewegungen als Animation (Person B). Auf keinen Fall sollte man den Schauspielern dies zeigen, denn sie müßten aus sich heraus funktionieren und sich eher nicht von außen sehen, sagte Person D. Eine Verschärfung äußerte Person C: „*die gehen Ihnen an die Gurgel*“. Die allgemeine Meinung war dennoch, daß die eigene Arbeit mit Schauspielern vorbereitet werden könnte, um bestimmte Ideen visualisieren oder sie sich selbst (in Zusammenarbeit mit Kameramann und Ausstatter) klar machen zu können.

Als Schlußfolgerung läßt sich feststellen, daß die Schnittstelle im jetzigen Stadium noch erhebliche Aufmerksamkeit erfordert und lediglich die Schachfiguren-ähnlichen Platzhalterfiguren uneingeschränkt funktionieren. Die Auswertung der Videoaufzeichnungen ergab außerdem, daß die Personen durchweg relativ steif stehen, was in der noch nicht perfekten Verfolgungssoftware begründet ist, die nicht bei jeder Körperhaltung befriedigende Ergebnisse liefert. Der Benutzer trägt beispielsweise eine grüne Markierung zur einfachen Lokalisierung. Dreht sich der Benutzer um ca. 30 Grad seitwärts, ist diese Markierung in bestimmten Situationen in einer der beiden Kameras nicht mehr sichtbar

und die Verfolgung wird unterbrochen, was zur Folge hat, daß der Cursor kurzzeitig aus dem Bild verschwindet. Für eine Benutzerakzeptanz muß aber eine freie Beweglichkeit im Raum als solche auch gewährleistet sein, d.h. das alleinige Im-Raum-stehen reicht für eine Anwendung mit gestischer Interaktion nicht aus. Zusätzlich erscheinen das Stehen, das „Sichbewegen“ und eine Zeigegestik als Mausersatz der Arbeitssituation eines Regisseurs eher nicht ähnlich. Weiterhin wird, wie auch im letzten Abschnitt, deutlich, daß die Darstellung von Bewegungseindrücken sehr wichtig ist. Der Einsatz eines Planungswerkzeuges, wie es mit diesem Prototypen skizziert wurde, erscheint jedoch bei komplizierten Szenen sowie zur Vorbereitung der Arbeit mit den Schauspielern sehr hilfreich. Besonders einfach würde sich die Interaktion mit einem modellhaften Set von Figuren und Objekten, ähnlich einem Schachspiel, gestalten.

6.4.4.3 Praktischer Nutzen

Der Hauptnachteil, der von vier Personen gesehen wurde, ist die generelle Verwendung des Computers für die Regietätigkeit. Einerseits würde man sich durch ein Planungswerkzeug von dem prozeßhaften Arbeiten am Set entfernen, da sich stets Dinge aus den Menschen und der Gruppe ergeben würden, die nicht in den Einzelpersonen begründet sind, sondern erst aus der Zusammenarbeit entstehen (Person B). Weiterhin bestünde beim Computer das Problem der Abstraktion: Bei sparsamen Zeichnungen ließen sich Lücken sehr viel einfacher durch Phantasie auffüllen. Hier bestehe die Gefahr, in ein Schema zu kommen (Person D), dadurch, daß die Leblosigkeit des Computers in den Film übersetzt werden würde (Person F). Ein derartiges Planungssystem suggeriere jedem potentiellen Benutzer, daß er die Regiearbeit beherrsche, obwohl es sich eher um eine Arbeitsweise nach dem *trial-and-error*-Prinzip handeln würde. Ein Effekt, der auch bei computergestützten Schnittsystemen wie Avid zu beobachten sei (Person A). Trotzdem konnten sich alle Befragten ein derartiges System besonders bei der Planung der Ausstattung und bei der Planung von Szenenauflösungen mit Ausstatter und Kameramann als sehr nützlich vorstellen. Teilweise wurde dies auf komplizierte bzw. auf nicht vorstellbare Szenen oder auf die Visualisierung einer Vision eingeschränkt (Personen B, E, F). Bei komplizierten Szenen könnte die Raumgestaltung inklusive Licht und Kameraeinstellung dargestellt werden. Der Beleuchter könnte dann direkt die Situation erkennen. Außerdem ließen sich, im Gegensatz zum Storyboard, Kameraeinstellungen überprüfen, denn oft, so meinte Person D, würden Zeichner zuwenig darüber wissen. Auch lassen sich Mißverständnisse zwischen Regisseur und Kameramann, zu denen es nach Person F immer wieder kommt, besonders bei Kamerafahrten und Schwenks im Vorfeld verhindern.

Die Nützlichkeit eines Planungssystems ist also bei komplexen Szenen und als Vorbereitung und Diskussionsgrundlage vorstellbar. In der vorliegenden Version ist mangels Kameraeinstellungsmöglichkeiten und der Totale als einzige Sicht auf die Szene eher eine Theateranwendung praktikabel. Das Abwenden vom prozeßhaften Arbeiten sowie die Gefahr der Illusion einer kinderleichten Regie durch den Computer sind bei der Überprüfung des praktischen Nutzens zu berücksichtigen.

6.4.4.4 Diskussion

Zusammenfassend läßt sich feststellen, daß der untersuchte Prototyp der Arbeitssituation zumindest der hier befragten Regisseure eher nicht ähnlich ist. Die Verwendung von Raum zur Schaffung einer sinnlichen, körperlichen Kommunikation erscheint nicht zwingend notwendig, obgleich eine Projektion für ein Raumgefühl hilfreich ist und die Nützlichkeit nach Einbeziehung von Perspektiven und Kamerabewegungen erneut überprüft werden sollte. Für bestimmte Anwendungen der Filmplanung bietet es sich an, das System in Richtung einer traditionellen Desktop-Anwendung mit Maus und Bildschirm zu vereinfachen. Für bestimmte Situationen kann ein stationärer Aufbau mit Projektionswand verwendet werden. Diesbezügliche Aussagen der befragten Regisseure könnten

einen Hinweis darauf sein, daß mit der Auswahl der Personen nicht die richtige Zielgruppe getroffen wurde. Die Interviews bestätigen, daß ein Regisseur ein Teil eines Teams ist, in dem besonders Kameramann und Ausstatter eine wichtige Rolle spielen. Alle Befragten konnten sich ein derartiges System besonders bei der Planung der Ausstattung und bei der Planung von Szenenauflösungen mit Ausstatter und Kameramann als sehr nützlich vorstellen. Die Befragung wurde somit nur mit einem Teil der Zielgruppe durchgeführt. Die Arbeit mit einem derartigen System dient eher der Vorbereitung und als Gesprächsgrundlage für Regisseur, Kameramann, Ausstatter und Beleuchter, weniger für die direkte Arbeit mit den Schauspielern. Besonders bei komplexen Szenen erscheint dies als sehr hilfreich. Szenen also, die eher bei großen Spielfilmen oder Actionfilmen die Regel sind, als beispielsweise bei einem Dokumentarfilm oder einem klassischen Autorenfilm. Somit kann der praktische Nutzen der Intuitiven Schnittstelle nunmehr hypothetisch auf ein engeres Berufsfeld innerhalb der Filmplanung eingeschränkt werden. Eine erneute Untersuchung mit entsprechend gezielt ausgewählten Personen könnte nähere Aufschlüsse geben. Als absolut notwendig kann zuvor die Ergänzung um Kameraperspektiven und Bewegungseindrücke angesehen werden. Als Variante ist ein modellhaftes System mit kleinen Figuren sehr gut vorstellbar, das besonders einfach zu bedienen wäre und nicht die Aufmerksamkeit des Benutzers erfordern würde. Die vorgeschlagene Aufzeichnung von Schauspielerbewegungen zur Vorbereitung der Arbeit mit Schauspielern erscheint beim jetzigen Stand der Technik als zu aufwendig.

Zur generellen Gültigkeit des Systems ist noch einmal zu betonen, daß hier ein Vorbereitungswerkzeug untersucht wurde. Es geht also nicht um die Ersetzung der Filmproduktion mit ihrer komplexen Prozeßhaftigkeit, sondern um die Planung von Vorgängen vor der Produktion, bei denen die Verwendung von Standards unter Umständen durchaus sinnvoll sein kann. Gerade bei solchen Filmproduktionen, die hochkomplex sind, wie beispielsweise „Independence Day“, stellt sich die Frage der Ökonomie besonders, so daß ein derartiger Einsatz enorm sinnvoll sein kann. Hingegen könnte bei einer Autorenproduktion eine Desktop-Variante zum Einsatz kommen.

7 Ausblick

An dieser Stelle soll aus den Erfahrungen der empirischen Untersuchung ein Vorschlag für einen funktionalen Prototypen gemacht werden. Das vorgeschlagene System besteht aus einem Tisch mit einer Projektion von unten analog dem *MetaDESK* von Ishii (97). Auf diesem „Spieltisch“ werden verschiedene Figuren durch im Tisch angebrachte Kameras erkannt. Zur Bedienung des Systems wird auf dem Tisch ein Floorplan (Grundriß) angezeigt. Wird eine Figur auf dem Tisch platziert, erscheint eine entsprechende virtuelle Figur in der Szene. Je nach Ausbau des Systems kann die Szene auf einer Projektionswand oder auf einem nebenstehenden Computermonitor dargestellt werden. Im ersten Fall würde der Tisch die Abmessungen eines Arbeitstisches ähnlich der *Workbench* von Krüger (94) besitzen (Abbildung 7-1).

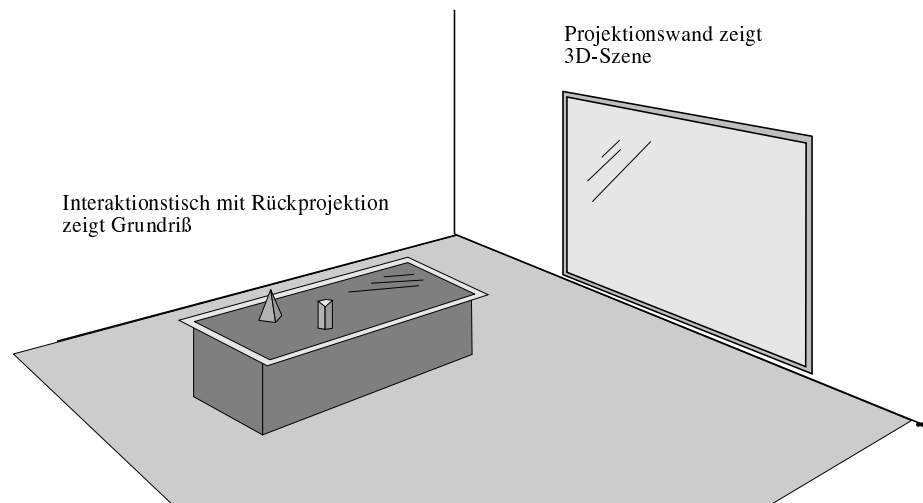


Abbildung 7-1: Darstellung als stationäres System mit Interaktionstisch

Im zweiten Fall kann ein System verwendet werden, das neben einem Monitor Platz findet (Abbildung 7-2). Auf der Projektion bzw. auf dem Monitor wird die Szene dreidimensional mit der gewählten Kameraeinstellung dargestellt. Ein Hinzufügen von Objekten erfolgt durch einfaches Plazieren einer entsprechenden Figur auf das Spielfeld. Dazu müßte zumindest eine kleinere Anzahl unterscheidbarer Figuren vorhanden sein. Der Kamerastandpunkt kann durch einfaches Bewegen einer Kamerafigur erreicht werden.

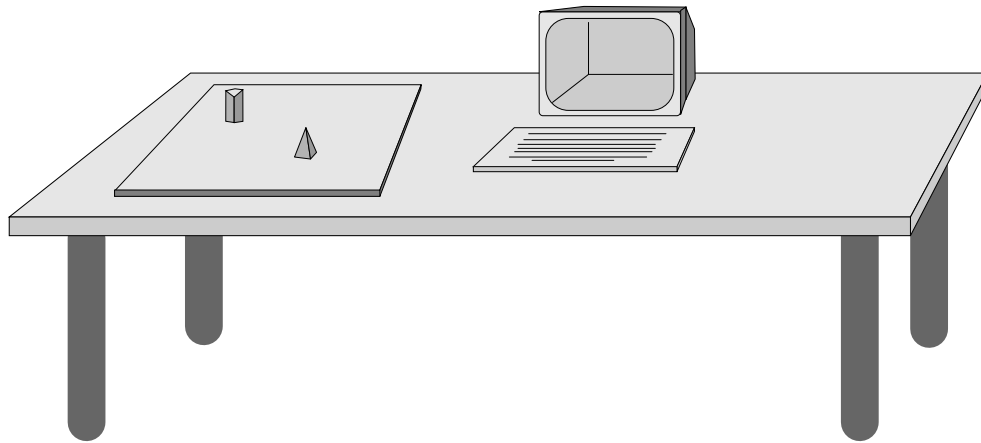


Abbildung 7-2: Portables System mit Interaktionsfeld

Die durchgeführten Manipulationen der Szene werden bei Bedarf über Internet-Verbindungen, z. B. in einer Netscape-Umgebung, an beliebigen Orten dargestellt, so daß sich Regisseur und Kameramann gegebenenfalls auch über Entfernungen austauschen können. Die Java-basierte Internet-Applikation zeigt dabei sowohl den Grundriß mit den Figuren als zweidimensionale Icons, als auch die gewählte dreidimensionale Ansicht der Szene (Abbildung 7-3). Manipulationen im Grundriß über diese Java-Schnittstelle werden auf dem stationären System durch projizierte Symbole auf den Tisch dargestellt und können vom Benutzer bei Bedarf durch die Figuren aktualisiert werden (indem die Figuren mit den angezeigten „neuen“ Positionen zur Deckung gebracht werden). Eine Diskussion der Szene erfolgt dann zusätzlich über eine Sprachverbindung per Telefon oder Internet.

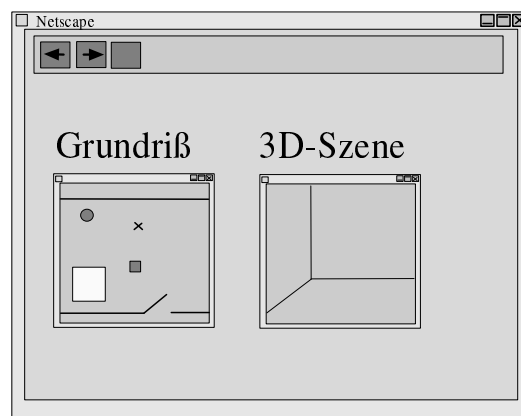


Abbildung 7-3: Internet-Anwendung mit Grundriß und 3D-Szene

Dieser Setup eines Systems erlaubt sowohl die Einbeziehung des Realraumes vor der Projektion für eine weitere Untersuchung als auch die Verbindung von virtuellen mit realen Objekten (Spielfiguren), die sich als nützlich erwiesen hat. Zusätzlich ist das System bei Bedarf (in der kleineren Version) transportabel und kann sogar über weltweite Netzwerke verwendet werden. Dies kann bei Reisen zu den Drehorten hilfreich sein aber auch auf Grund von Terminschwierigkeiten der beteiligten Personen eine Diskussion zu einem bestimmten Zeitpunkt überhaupt erst ermöglichen.

Konkret ergeben sich hieraus folgende Forschungsprobleme:

Entwicklung Tischinterface. Hier sind vor allem Fragen der Gestaltung und Ergonomie für eine Akzeptanz beim Benutzer ausschlaggebend. Zu den zu lösenden Problemen gehören sowohl eine genaue Untersuchung welche Art von Spielfiguren sich für ein derartiges Interface eignet, als auch die Frage nach dem darzustellenden Abstraktionsgrad (sowohl der Figuren, als auch der Projektion auf dem Tisch). Weiterhin muß ein Softwareinterface entwickelt werden, das einen guten Kompromiß zwischen einfacher, intuitiver Bedienung und Flexibilität und Funktionalität darstellt.

Verfolgungssoftware für Spielfiguren. Die Verfolgung von Spielfiguren stellt allein durch die Größe der Objekte andere Anforderungen an eine Verfolgungssoftware als sie in dieser Arbeit skizziert wurden. Sollen zudem mehrere Objekte verwendet werden, so reicht eine rein farbliche Markierung der Figuren nicht mehr aus, und es müssen weitere Merkmale einbezogen werden. Ebenso ergeben sich neue Herausforderungen aufgrund neuer Funktionalitäten, wie etwa eine Kamerasteuerung über eine entsprechende Figur, oder durch die Verdeckungsmöglichkeiten durch Benutzerinteraktionen.

Internet-Anwendung. Die Entwicklung einer Internet-Anwendung ist mit Hilfe einer Java-Schnittstelle leicht vorstellbar. Zu klären sind jedoch diverse Aspekte der Online-Kommunikation, wie beispielsweise ein Multicasting, das notwendig wäre, wenn mehrere Personen gleichzeitig am System arbeiten wollen. Weiterhin ist hier noch unklar, welche Art von Abstraktion oder Detailreichtum für eine derartige Anwendung adäquat ist. Durch den fehlenden persönlichen Kontakt kann eine Strichzeichnung oder eine einfache 3D-Szene u.U. nicht ausreichend sein, die Szene zu begreifen, andererseits jedoch gerade die richtige Abstraktionsstufe für die geplante Szene darstellen. Eine Untersuchung dieser Problemstellung könnte jedoch wertvolle Information auch für andere Bereiche der CSCW (Computer Supported Cooperative Work) liefern.

Unter Berücksichtigung der Ergebnisse aus Kapitel 6 ergeben sich für eine weitergehende Untersuchung von Körper und Raum als Teil der Schnittstelle folgende Aufgabenstellungen:

- Entwicklung eines Prototypen mit Kamerabewegung und komplizierteren Szenen
- Erweiterung der Verfolgungssoftware um Möglichkeiten der Kamerasteuerung. Dabei ist eine freie Beweglichkeit im Raum auch während der Bedienung zu ermöglichen.
- Durchführung einer empirische Untersuchung mit einem erweiterten und konkretisierten Personenkreis wie in Kapitel 6 definiert.

8 Literatur

- Agrawala M, Beers A, Fröhlich B, Hanrahan P, et. al. (1997), *The Two-User Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space*, Proceedings SIGGRAPH '97, pp 327-331.
- Akita K (1984), *Image Sequence Analysis of Real World Human Motion*, Pattern Recognition, vol. 17, 1984, pp. 73-83.
- Azarbayejani A, Starner T, Horowitz B, and Pentland A (1993), *Visually Controlled Graphics*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No 6, June 1993, pp 602-605.
- Azarbayejani A, Wren C, Pentland A (1996), *Real-Time 3-D Tracking of the Human Body*, Proceedings of IMAGE'COM 96, Bordeaux, Mai 1996.
- Azuma R (1993), *Tracking Requirements for Augmented Reality*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 50-51.
- Azuma R (1997), *A Survey of Augmented Reality*, Presence: Special Issue on Augmented Environments, Vol 6, Issue 4, August 1997, pp 355-385.
<http://www.cs.unc.edu/~azuma>
- Badler N, Phillips C, Webber B (1993), *Simulating Humans*, Oxford University Press, 1993.
- Baudel T, Beaudouin-Lafon M (1993), *CHARADE: Remote Control of Objects using Free-Hand Gestures*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 28-34.
- Blake A, Yuille A eds. (1992), *Active Vision*, MIT Press.
- Bobick A, Intille S, Davis J, et.al. (1997), *The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment*, , M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 398, http://vismod.www.media.mit.edu/cgi-bin/tr_pagemaker.
- Bolt RA (1984), *The Human Interface*, Lifetime Learning Publications, Belmont, California.
- Bröckl-Fox U (1995), *Untersuchung neuer, gestenbasierter Methoden für die 3D-Interaktion*, Dissertation im Fach Informatik, Universität Karlsruhe, Verlag Shaker 1995.
- Bröckl-Fox U (1995), *Real-Time 3D Interaction with up to 16 Degrees of Freedom from Monocular Video Image Flows*, IEEE International Workshop on Automatic Face- and Gesture-Recognition, June 26-28, Zürich, Switzerland, pp. 172-178, 1995.
- Cha T eds. (1980), *Apparatus*, Cinematographic Apparatus: Selected Writings, Tanam Press, New York 1980.

- Chen Z, Lee H-J (1992), *Knowledge-Guided Visual Perception of 3-D Human Gait from a Single Image Sequence*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 22, No. 2, 1992, pp 336-342.
- Coen M (1997), *Building Brains for Rooms: Designing Distributed Software Agents*, In Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence. (IAAI97). Providence, R.I. 1997. <http://www.ai.mit.edu/people/mhcoen/>.
- Coen M (1998), *Design Principles for Intelligent Environments*, In Proceedings of AAAI 1998 Spring Symposium on Intelligent Environments. <http://www.ai.mit.edu/people/mhcoen/>
- Cooperstock J (1995), *Evolution of a Reactive Environment*, Proceedings CHI'95, Denver, Colorado. <http://www.dgp.toronto.edu/people/rroom/research>.
- Cooperstock J, Fels S, Buxton W, Smith K (1997), *Reactive Environments*, Communications of the ACM, Vol. 40, No. 9, September 1997, pp 65-73.
- Dai Y, Nakano Y (1995), *Extraction of Facial Images from Complex Background Using Color Information and SGLD Matrices*, IEEE International Workshop on Automatic Face- and Gesture-Recognition, June 26-28, Zürich, Switzerland, pp. 238-242, 1995.
- Darrell T, et. al. (1994), *A Novel Environment for Situated Vision and Behavior*, Proc. of CVPR-94 Workshop for Visual Behaviors, pp. 68-72, Seattle, Washington, June 1994.
- Druin A, Perlin K (1994), *Immersive Environments: A Physical Approach to the Computer Interface*, CHI'94 Conference Companion, pp 325-326.
- du Plessis RM (1967), *Poor Man's Explanation of Kalman Filtering or how I stopped worrying and learned to love Matrix Inversion*, Technical report, Rockwell International, California, June 1967 (available via Taygeta Scientific Inc., 1340 Munras Ave., Suite 314 Monterey, CA 93940, <http://www.taygeta.com/kalman.html>).
- Elrod S, Hall G, Costanza R, et.al. (1993), *Responsive Office Environments*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 75-84.
- Faugeras O (1993), *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, Cambridge, 1993.
- Feiner S, MacIntyre B, Seligmann D (1993-1), *Knowledge-Based Augmented Reality*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 53-62.
- Feiner S, et al. (1993-2), *Windows on the World: 2D Windows for 3D Augmented Reality*, Proc. UIST '93: ACM Symp. On User Interface Software and Technology, Atlanta, Georgia, 3.-5. November 1993, S. 145-155
- Fleischmann G, Hoch M, Schwabe D, Barg W, Lampe G (1994-1), *FilmPlan: ein interaktives Filmplanungssystem*, Lab 1, Magazin der Kunsthochschule für Medien Köln, 1994, pp 34-39.
- Fleischmann G, Hoch M, Schwabe D, et. al. (1994-2), *smdk: An Interactive Self-Organizing Sound Environment*, AAAI-94 Workshop, Artificial Intelligence, Artificial Life, & Entertainment, Working Notes, Seattle, Washington, 1994.

- Fleischmann G, Schwabe D (1995), *Interaktive Simulationsumgebung für die Filmplanung*, Tagungsband des 40. internationalen wissenschaftlichen Kolloquiums der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität Ilmenau, 1995.
- Fitzmaurice G (1993), *Situated Information Spaces and Spatially Aware Palmtop Computers*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 39-49.
- Fitzmaurice G, Ishii H, Buxton W (1995), *Bricks: Laying the Foundations for Graspable User Interfaces*, ACM Proceedings CHI'95, Denver, Colorado, 1995.
- Foley J, van Dam A, Feiner S, Hughes J (1991), *Computer Graphics. Principles and Practice. Second Edition*. Addison-Wesley, 1991.
- Ford A (http), *The Colour (color) Equations Document*, available at <http://www.wmin.ac.uk/ITRG/docs/colorreq>
- Fowler M, Scott K (1997), *UML Distilled, Applying the Standard Object Modeling Language*, Object Technology Series, Addison-Wesley.
- Friedrichs J (1990), *Methoden empirischer Sozialforschung*, WV Studium Sozialwissenschaft, Westdeutscher Verlag, Opladen 1990.
- Fröhlich M, Wachsmuth I (1997), *Gesture Recognition of the Upper Limbs - from Signal to Symbol*, Proc. of Gesture Workshop Bielefeld 1997, Lecture Notes in Artificial Intelligence, Springer 1998.
- Fujihata M (1996), *Beyond Pages*, SIGGRAPH '96 Visual Proceedings, pp 24-25.
- Fukomoto M, Suenaga Y, Mase K (1994), „Finger-Pointer“: *Pointing Interface by Image Processing*, Computer & Graphics, Vol. 18, No. 5, pp 633-642.
- Gavrila DM (1996), *3-D model-based tracking of human upper body movement: a multi-view approach*, IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco, USA, June 1996.
- Gelb A (1974), *Applied Optimal Estimation*, MIT Press, Cambridge.
- Gibbs S, Arapis C, Breiteneder C, et. al. (1996), *TELEPORT - An Immersive 3D Teleconferencing System*, ACM MM'96. <http://viswiz.gmd.de/projects/cwall.html>
- Graffmann B (1996), *Modellbasierte 3D-Bewegungsverfolgung*, Diplomarbeit Universität Dortmund, Fachbereich Informatik, Dezember 1996.
- Halbach WR (1994), *Interfaces, Medien- und Kommunikationstheoretische Elemente einer Interface-Theorie*, Wilhelm Fink Verlag, München 1994.
- Hammes G (1995), *Bestimmung der Zeigerichtung aus Stereobildfolgen zur Entwicklung einer neuen Mensch-Maschine-Schnittstelle*, Diplomarbeit im Fachbereich Nachrichtentechnik, Vertiefungsfachrichtung Informationsverarbeitung, Fachhochschule Köln.

- Hauptmann A (1989), *Speech and Gestures for Graphic Image Manipulation*, Proceedings ACM CHI'89, pp 241-245.
- Hoch M, Fleischmann G (1996), *Social Environment: Towards an Intuitive User Interface*, Proc. of 3D Image Analysis and Synthesis, Infix 18.-19. November 1996, pp 155-161
- Hoch M (1997), *Intuitive Schnittstelle*, Lab, Jahrbuch 1996/97 für Künste und Apparate, Verlag Walther König, Köln 1997.
- Hoch M (1997), *Human Body Tracking for the Intuitive Interface*, Proc. of 3. Workshop Farbbildverarbeitung, IRB-Verlag Fraunhofer Gesellschaft, Stuttgart 1997, pp. 73-79.
- Hoch M (1997), *Object Oriented Design of the Intuitive Interface*, Proc. of 3D Image Analysis and Synthesis, Erlangen November 17-19, Infix 1997, pp 161-167.
- Hoch M (1998), *A Prototype System for Intuitive Film Planning*, Third IEEE International Conference on Automatic Face and Gesture Recognition (FG'98), April 14-16, Nara, Japan 1998, pp 504-509.
- Hoch M (1998), *Interaktion, Körper und Realraum*, Informatik '98, Informatik zwischen Bild und Sprache, 28. Jahrestagung der Gesellschaft für Informatik, 21.-25. September 1998, Magdeburg, Springer 1998.
- Hodges S, Louie G (1994), *Towards the Interactive Office*, Conference Companion CHI'94, Boston, MA, April 1994, pp 305-306.
- Hogg D (1984), *Interpreting Images of a Known Moving Object*, PhD dissertation, University of Sussex, Brighton/UK 1984.
- Hopper A, Harter A, Blackie T (1993), *The Active Badge System*, ACM CHI'93 Proceedings of INTERCHI '93, pp 533-534.
- Intille S, Davis J, Bobick A (1997), *Real-Time Closed-World Tracking*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '97), 1997.
- Ishii H, Mochizuki K, Kishino F (1993), *A Human Motion Image Synthesizing by Model Based Recognition from Stereo Images*, Proceedings Imagina '93, pp 69-74.
- Ishii H, Ullmer B (1997), *Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms*, ACM Proceedings of CHI'97, pp. 234-241.
<http://www.media.mit.edu/groups/tangible>.
- Jähne B (1995), *Digital Image Processing: Concepts, Algorithms, and Scientific Applications*, Springer-Verlag, Berlin 1995.
- Johnson MP, Maes P, Darrel T (1994), *Evolving Visual Routines*, Artificial Life, 1(4), 1994.
- Jung B, Wachsmuth I (1998), *Integration of Geometric and Conceptual reasoning for Interacting with Virtual Environments*, to appear in Proceedings of 98' AAAI Spring Symposium on Multimodal Reasoning 1998.

- Kahn R, Swain M (1995), *Understanding People Pointing: the Perseus System*, ISCV, November 1995. <http://www.cs.uchicago.edu/groups/ai/perseus/>.
- Kahn R, Swain M, Prokopowicz P, and Firby R (1996), *Gesture Recognition Using the Perseus Architecture*, Computer Vision and Pattern Recognition, San Francisco CA, June 1996.
- Kakadiaris I, Metaxas D, Bajcsy R (1994), *Active Motion-Based Segmentation of Human Body Outlines*, Proc. of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin TX, November 11-12, 1994, pp 50-56.
- Kakadiaris I, Metaxas D (1995), *3D Human Body Model Acquisition from Multiple Views*, 5th IEEE International Conference on Computer Vision ICCV, June 20-23, Boston, 1994, pp 618-623.
- Kjeldsen R, Kender J (1996), *Finding Skin in Color Images*, IEEE International Conference on Automatic Face and Gesture Recognition, Oct 96, Killington, Vermont, 1996, pp 312-317
- Klette R, Koschan A, Schlüns K (1996), *Computer Vision - Räumliche Information aus digitalen Bildern*, Vieweg.
- Kohler M (1996), *Vision Based Remote Control in Intelligent Home Environments*, Proc. of 3D Image Analysis and Synthesis, Infix 18.-19. November 1996, pp 147-154
- Kohler M (1997), *Using the Kalman Filter to track Human Interactive Motion - Modeling and Initialization of the Kalman Filter for Translational Motion*, Technical Report 629, Informatik VII, University of Dortmund, February 1997 <http://ls7-www.informatik.uni-dortmund.de/>.
- Krueger M (1991), *Artificial Reality II*, Addison-Wesley 1991.
- Krüger W, Fröhlich B (1994), *The Responsive Workbench*, IEEE Computer Graphics and Applications, May 1994, Volume 14, Issue 3, pp 12-15.
- Kulesa T (1998-1), *Automatische Kalibrierung zur Personenverfolgung in Farbbildfolgen*, Diplomarbeit Universität Dortmund, Fachbereich Informatik, Februar 1998.
- Kulesa T, Hoch M (1998-2), *Efficient Color Segmentation under Varying Illumination Conditions*, Tenth IEEE Image and Multidimensional Digital Signal Processing (IMDSP) Workshop, July 12-16, 1998, Alpbach, Austria.
- Kurakake S, Nevatia R (1992), *Description and Tracking of Moving Articulated Objects*, Proc. 11th Intern. Conf. on Pattern Recogn. ICPR, Den Hague, The Netherlands, Aug. 30 - Sep 3, Vol. 1, pp 491-495, 1992.
- Latoschik M, Wachsmuth I (1997), *Exploiting distant pointing gestures for object selection in a virtual environment*, Proceedings of Gesture Workshop Bielefeld 1997, to appear in Lecture Notes in Artificial Intelligence, Springer 1998.
- Lee H-J, Chen Z (1985), *Determination of 3D Human Body Postures from a Single View*, Computer Vision, Graphics, and Image Processing, 30, 1985, pp 148-168.

- Leung M, Yang Y (1994), *A Model Based Approach to Labeling Human Body Outlines*, IEEE Workshop on Motion of Non-Rigid and Articulated Objects, November 11-12, Austin, TX, 1994, pp 57-62.
- Lernout & Hauspie Speech Products (1996), *User's Reference and Programming Guide*, Sint-Krispijnstraat 7, 8900 Ieper, Belgium, V3.010, July 1996.
- Lowe D (1992), *Robust model-based motion tracking through the integration of search and estimation*, International Journal of Computer Vision, 8, 2 (August 1992), pp. 113-122.
- Maggioni C (1995), *GestureComputer – New Ways of Operating a Computer*, IEEE International Workshop on Automatic Face- and Gesture-Recognition, Zürich, 26.-28. Juni 1995, pp 166-171.
- Mandala (1993), *Mandala: Virtual Village*, ACM SIGGRAPH Visual Proceedings 1993.
- Mase K, Kadobayashi R (1997), *Gesture Interface of a Virtual Walk-through*, Workshop on Perceptual User Interface 97 (PUI97), 1997.
- Merleau-Ponty M (1966), *Phänomenologie der Wahrnehmung*, Walter de Gruyter & Co.
- Merleau-Ponty M (1984), *Das Auge und der Geist : Philosophische Essays*, Felix Meiner Verlag GmbH, Hamburg 1984.
- MIL (1996), *Matrox Imaging Library - Users Guide*, Matrox Electronic Systems Ltd, Canada. <http://www.matrox.com>.
- Niemann H (1990), *Pattern analysis and understanding*, 2nd edn. Springer, Berlin Heidelberg New York.
- Onishi T, Takemura H, Kishino F (1993), *The Manipulation of Graphics Objects by Using Hand Gestures*, Proceedings Imagina '93, pp 129-134.
- Paulus D (1992), *Objektorientierte und wissensbasierte Bildverarbeitung*, Vieweg.
- Paulus D, Niemann H (1994), *Object-Oriented Programming for image analysis*, in J. Mennon (ed.), Current Topics of Pattern Recognition Research, vol 1 of Reserach Trends, India 1994, pp 185-204.
<http://www5.informatik.uni-erlangen.de/HTML/English/Mitarbeiter/pa/pa.html>
- Pentland A (1995), *Machine Understanding of Human Action*, M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 350, Sept. 1995, short version available in: 7th Int'l Forum on Frontier of Telecom Technology Nov. 1995, Tokyo, Japan
- Perlin K, Goldberg A (1996), *Improv: A System for Scripting Interactive Actors in Virtual Worlds*, Proceedings SIGGRAPH'96, 1996, ACM.
<http://mrl.nyu.edu/improv/sig96-paper/>
- Pinhanez C (1997), *Computer-Theater*, Proceedings of the Eighth International Symposium on Electronic Arts (ISEA'97), Chicago, Illinois, September 1997.

- Rauterberg M, Bichsel M, Fjeld M (1997), *Computer vision based interaction with a planning tool for construction and design*, Proc. of Gesture Workshop Bielefeld 1997, to appear in Lecture Notes in Artificial Intelligence, Springer 1998.
- Rehrmann V (1994), *Stabile, Echtzeitfähige Farbbildauswertung*, Koblenzer Schriften zur Informatik, Fölbach 1994.
- Resnick M (1993), *Behavior Construction Kits*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 64-71.
- Rohr K (1993), *Incremental Recognition of Pedestrians from Image Sequences*, Proc. IEEE Conf. on Computer Vision & Pattern Recognition (CVPR '93), 1993, pp 8-13.
- Ropohl G (1979), *Eine Systemtheorie der Technik*, Hanser Verlag, München 1979.
- Rumbaugh, Blaha, Premerlani et. al. (1991), *Object-Oriented Modeling and Design*, Prentice Hall, 1991.
- Russel K, Starner T, Pentland A (1995), *Unencumbered Virtual Environments*, IJCAI-95 Workshop on Entertainment and AI/Alife.
- Sagerer G (1990), *Automatisches Verstehen gesprochener Sprache*, Reihe Informatik, BI Wissenschaftsverlag, Mannheim.
- Saxe D, Foulds R, (1996), *Towards Robust Skin Identification in Video Images*, IEEE International Conference on Automatic Face and Gesture Recognition, Oct 96, Killington, Vermont, 1996, pp 379-384
- Schiele B, Waibel A (1995), *Gaze Tracking Based on Face-Color*, IEEE International Workshop on Automatic Face- and Gesture-Recognition, June 26-28, Zürich, Switzerland, pp. 344-349, 1995.
- Schröter S (1996), *Entwicklung von Verfahren zur automatischen Kalibration eines farb-basierten Objekterkennungssystems*, Diplomarbeit Universität Dortmund, Fachbereich Informatik, Oktober 1996.
- Schröter S (1997), *Automatic Calibration of Lookup-Tables for Color Image Segmentation*, Proc. of 3D Image Analysis and Synthesis, 17.-18. November 1997, pp 123-129, Infix 1997.
- Schuster R (1996), *Objektverfolgung in Farbbildfolgen*, Dissertation, Technische Universität München, Infix 1996.
- Shaw J (1990), *The Legible City*, Prix Ars Electronica 1990. <http://web.aec.at/prix/1990/90azI-legible.html>.
- Sommerer C, Mignonneau L (1993), *Interactive Plant Growing*, Genetische Kunst - Künstliches Leben, Ars Electronica 1993, PVS Verleger, Wien. <http://web.aec.at/fest/fest93/somm.html>.
- Sommerer C, Mignonneau L (1994), *A-Volve, A real-time interactive environment*, Visual Proceedings SIGGRAPH'94, ACM. <http://web.aec.at/prix/1994/94gnI-avolve.html>.

- Stark M, Kohler M, PG Zyklus (1995), *Video based gesture recognition for human computer interaction*, Technical Report 593, Informatik VII, University of Dortmund, November 1995.
- Stary C (1996), *Interaktive Systeme. Software-Entwicklung und Software-Ergonomie*, Vieweg Informatik/Wirtschaftsinformatik 1996.
- Stöcker H (1995), *Taschenbuch Mathematischer Formeln und moderner Verfahren*, Verlag Harri Deutsch, Frankfurt am Main 1995.
- Sullivan JW, Tyler SW (Ed.) (1991), *Intelligent User Interfaces*, ACM press, New York 1991.
- Thorisson K, Koons D, Bolt R (1992), *Multi-Modal Natural Dialogue*, Proceedings ACM CHI'92, pp 653-654.
- Tillmann H (1998), *Ein modellbasiertes Konzept zur Personenverfolgung*, Diplomarbeit Universität Dortmund, Fachbereich Informatik, Juli 1998.
- Tizhoosh H (1998), *Fuzzy-Bildverarbeitung: Einführung in Theorie und Praxis*, Springer 1998.
- Torrance M (1995), *Advances in Human-Computer Interaction: The Intelligent Room*, Working Notes of CHI '95 Research Symposium, Denver, Colorado, 6.-7. Mai 1995.
- Tsai RY (1986), *An Efficient and Accurate Camera Calibration Techznique for 3D Machine Vision*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986, pp 364-374.
- Wachsmuth I, Lenzmann B, Cao Y (1995), *VIENA: A Multiagent Interface to a Virtual Environment*, Proceedings First International Conference on Multi-Agent Systems (ICMAS-95), 1995.
- Watt A (1992), *Animating articulated structures*, in "Advanced Animation and Reendering Techniques : theorie and practice, ACM Press, New York, 1992
- Weimer D (1989), *A Synthetic Visual Environment with Hand Gesturing and Voice Input*, Proceedings ACM CHI'89 pp 235-240.
- Weiser M (1993), *Some Computer Science Issues in Ubiquitous Computing*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 75-84.
- Weiser M (1996), *Designing Calm Technology*, PowerGrid Journal, v101, July 1996, <http://powergrid.electriciti.com/1.01>.
- Wellner P (1993), *Interacting with Paper on the DigitalDesk*, Communications of the ACM, Vol. 36, No. 7, July 1993, pp 87-96.
- Wren C, Azarbayejani A, Darrell T, Pentland A (1996), *Pfinder: Real-Time Tracking of the Human Body*, IEEE International Conference on Automatic Face and Gesture Recognition, Oct 96, Killington, Vermont, 1996, pp 51-56.

Wren C, Sparacino F, Azarbayejani A et. al. (1997), *Perceptive Spaces for Performance and Entertainment: Untethered Interaction using Computer Vision and Audition*, Applied Artificial Intelligence, June 1997, Vol. 11, No. 4, pp. 267-284.

Yates FA (1966), *The Art of Memory*, Routledge & Kegan Paul/PLC, London 1966.

Zadeh LA (1965), *Fuzzy sets*, Information and Control 8, 1965, pp. 338-353.

Zimmermann T, Lanier J, Blanchard, Bryson S, Harvill Y (1987), *A Hand Gesture Interface Device*, CHI+GI '87 Proceedings, ACM Press, 1987 pp 189-192.