Lehrstuhl für Mensch-Maschine-Kommunikation
der Technischen Universität München

Stochastische Modellierung von Bildsequenzen zur Segmentierung und Erkennung dynamischer Gesten

Peter Morguet

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. J. Eberspächer
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. rer. nat. M. Lang
	2.	Univ.-Prof. Dr.-Ing. J. Hagenauer

Die Dissertation wurde am 13.06.2000 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 06.11.2000 angenommen.

Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Assistent am Lehrstuhl für Mensch-Maschine-Kommunikation der Technischen Universität München.

Meinem Doktorvater Herrn Prof. Dr. rer. nat. Manfred Lang, ohne dessen Betreuung diese Arbeit nicht möglich gewesen wäre, bin ich zu großen Dank verpflichtet. Die von ihm geförderte freie Arbeitsweise verbunden mit der stetigen Bereitschaft, bei auftauchenden Problemen für Diskussionen zur Verfügung zu stehen, ermöglichten mir eine interessante und produktive wissenschaftliche Betätigung.

Ebenso gilt an dieser Stelle mein Dank Herrn Prof. Dr.-Ing. Joachim Hagenauer vom Lehrstuhl für Nachrichtentechnik der Technischen Universität München für die Übernahme des Zweitgutachtens.

Meinem Kollegen Herrn Dr. Hans-Jürgen Winkler danke ich für die zur Verfügung gestellte Software, die mir in der Anfangsphase meiner wissenschaftlichen Tätigkeit einen schnelleren Start ermöglichte. Ihm und allen anderen Kolleginnen und Kollegen am Lehrstuhl für Mensch-Maschine-Kommunikation verdanke ich neben vielen fruchtbaren fachlichen Diskussionen auch nicht zuletzt die offene und freundschaftliche Atmosphäre, die wesentlich mit zur Freude an der Arbeit am Lehrstuhl beigetragen hat.

Weiterhin danke ich allen Studentinnen und Studenten, die sich in Form von Diplomarbeiten und Werkstudententätigkeiten aktiv an der Realisierung dieser Arbeit beteiligt haben.

Den Herren Peter Brand und Heiner Hundhammer gebührt Dank für ihren Einsatz bei der Verwaltung des Rechnernetzes und ihrer Geduld hinsichtlich der vielen rechnertechnischen „Sonderwünsche“.

Schließlich sei Frau Christine Reischer gedankt für ihre hilfsbereite, tatkräftige und offene Art, die die vielen kleinen Probleme des Arbeitsalltags erst gar nicht auftauchen ließ.

Nicht zuletzt danke ich meiner Frau Annette Springer-Morguet für ihre Geduld und ihre Unterstützung während der gesamten Entstehungszeit dieser Arbeit sowie meinen Kindern Clara und Luis für die vielen Stunden, die sie ohne ihren Vater verbringen mußten.

München, im Juni 2000 Peter Morguet

Zusammenfassung

In der vorliegenden Arbeit wird die Entwicklung eines Systems für den mit Handgesten gesteuerten Mensch-Maschine-Dialog beschrieben. Das System arbeitet bildverarbeitungsgestützt und ist somit nicht-intrusiv. Damit das Ziel einer natürlichen und intuitiven Interaktion erreicht werden konnte, wurde das an *dynamische Gesten* angepaßte Dialogkonzept der *indirekten Manipulation* eingeführt. Dieses neuartige Konzept erlaubt es, auch komplexe Aktionen mit einfachen, allgemein bekannten Gesten zu steuern. Es wurde in Usability-Experimenten validiert, wobei gleichzeitig ein sinnvoller Gestenkatalog und realistische Dialog-Videosequenzen gewonnen werden konnten.

Im Zentrum der automatischen Erkennung der aus den dynamischen Gesten resultierenden Bildsequenzen stehen teilweise modifizierte *Hidden-Markov-Modelle*, die sich zur Darstellung zeitlicher Merkmalssequenzen bewährt haben. Zur Anpassung der räumlich-zeitlichen Bildsequenzen an diese stochastische Modellierung wurden über eine räumliche Segmentierung sowie eine anschließende Merkmalsextraktion die zeitlich-seriellen Nutzdaten gewonnen.

Für die *räumliche Segmentierung*, mit der die Hände des Benutzers ohne weitere Hilfsmittel auf robuste Weise vom Hintergrund eines typischen Büro-Arbeitsplatzes getrennt werden können, wurde ein echtzeitfähiges, farbhistogrammbasiertes Verfahren entwickelt, das schnell an wechselnde Benutzer adaptiert werden kann.

Weiterhin wurden verschiedene modell- und pixelbasierte *Merkmalsextraktionsverfahren* mit jeweils spezifischen Vor- und Nachteilen teilweise neu konzipiert und implementiert. Die vergleichende Evaluierung dieser Verfahren im Verbund mit an die Bedingungen der Bildsequenzerkennung angepaßten Hidden-Markov-Modellen lieferte zunächst ein leistungsstarkes, echtzeitfähiges System für die personenunabhängige Erkennung *isolierter* Gesten.

Der Einsatz der Gestikerkennung in einem realen Dialogsystem erfordert für den *kontinuierlichen* Betrieb zusätzlich eine *zeitliche Segmentierung*, mit der die bedeutungstragenden gestischen Bewegungen aus dem Videobildstrom herausgelöst werden können. Dazu wurden im Rahmen dieser Arbeit zwei alternative Verfahren entwickelt und in ihrer Leistungsfähigkeit verglichen.

Beim ersten Ansatz handelt es sich um ein neuartiges einstufiges *Spotting-Verfahren*, mit dem Segmentierung und Klassifikation auf der Basis von Hidden-Markov-Modellen in einem integralen Arbeitsschritt durchgeführt werden können. Dieses Verfahren ist sehr leistungsfähig, da es insbesondere in der Lage ist, Gesten von bedeutungslosen Bewegungen zu trennen, wie dies für eine Dialoganwendung erforderlich ist. Für eine Echtzeitanwendung ist diese Methode allerdings zu rechenaufwendig.

Der zweite, in der vorliegenden Ausprägung neue Ansatz behandelt Segmentierung und Klassifikation getrennt in zwei Verarbeitungsstufen. Zuerst werden mit einer regelbasierten *Bewegungsdetektion* Kandidaten für mögliche Gesten im Videostrom gefunden. In einem anschließenden, *isolierten Erkennungschritt* werden die Bewegungsintervalle wiederum mit Hidden-Markov-Modellen klassifiziert. Dieser Ansatz ist im Vergleich zum

einstufigen Verfahren weniger leistungsfähig, da keine bedeutungslosen Bewegungen unterdrückt werden können.

Setzt man jedoch kontextfreie Gesten ein, die ein entsprechend kooperatives Benutzerverhalten voraussetzen, so lassen sich mit dem zweistufigen Ansatz Echtzeitfähigkeit mit sehr guten Erkennungsleistungen verbinden. Auf diese Weise konnte erstmalig ein *Demonstrator* in Form eines mit dynamischen Gesten gesteuerten, dreidimensionalen Szenen-Editors verwirklicht werden.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation und Anwendungen

Wenn Menschen miteinander kommunizieren, tauschen sie Informationen auf visueller, akustischer und taktiler Ebene aus. Jede dieser *Modalitäten* hat spezifische Vor- und Nachteile, die vom Inhalt der zu übermittelnden Nachricht und der jeweiligen Situation abhängen. In der Regel werden sowohl auf der Eingabe- als auch auf der Ausgabeseite mehrere dieser Modalitäten gleichzeitig verwendet, so daß sich diese gegenseitig ergänzen.

Im heute üblichen Dialog zwischen Mensch und Maschine wird jedoch nur ein kleiner Teil dieses kommunikativen Spektrums eingesetzt: die Eingabe — aus Sicht des Computersystems — läuft über den taktilen Kanal (Tastatur und Maus), während die Ausgabe auf visueller und akustischer Ebene erfolgt (Monitor, Lautsprecher und Drucker). Durch diese Asymmetrie und das Fehlen des so wichtigen visuellen und akustischen Eingabekanals ist die Mensch-Maschine-Kommunikation in hohem Maß gewöhnungsbedürftig, unnatürlich und damit insbesondere für einen Anwender, der sich nur auf seine Aufgabe und nicht auf die Bedienung des Arbeitsgerätes konzentrieren möchte, sehr ineffizient [Lan94b].

Damit der Mensch-Maschine-Dialog entscheidend verbessert und natürlicher gestaltet werden kann, müssen daher die visuellen und akustischen Eingabemodalitäten nutzbar gemacht und die taktilen Modalitäten entsprechend weiterentwickelt werden [Lan94a, Lan99a]. Die Spracherkennung und -interpretation blickt auf eine lange Forschungstradition zurück und wurde in den letzten Jahren wesentlich weiterentwickelt [Lan94c, Sta97, Mül97]. Mit der Verfügbarkeit der entsprechenden Eingabegeräte konnte auch die Handschrift- und Formelerkennung erfolgreich vorangetrieben werden [Lan94b, Win96].

Die bildverarbeitungsgestützte Gestikerkennung als visuelle Eingabekomponente rückte dagegen erst vor wenigen Jahren in das Interesse der Forschung [Lan94b, Hua95, Ekm95]. Dies liegt sicherlich nicht unwesentlich daran, daß erst in letzter Zeit die erforderliche Rechnerleistung zum Verarbeiten der großen anfallenden Informationsmengen auf breiter Basis zur Verfügung steht [Lan99b]: auch beim Menschen dominiert der optische Sinn mit 87 % Anteil an der empfangenen Information alle anderen Sinne [Lan94a]. Aus dem Fehlen einer visuellen Eingabe in der Mensch-Maschine-Kommunikation ergibt sich darüberhinaus ein Widerspruch zum Vorherrschen der in üblichen Systemen verwendeten visuellen Ausgabe, die sich zudem im Bereich der dreidimensionalen Visualisierung weiter vervollkommnet und an die kognitiven Fähigkeiten des Menschen anpaßt [Fol90, Kru91, Spe95].

Motivation für die vorliegende Arbeit war daher der Wunsch, durch die Gestaltung einer bildverarbeitungsbasierten und damit berührungsfreien und ungehinderten Gestikerkennung im Zusammenspiel mit einer visuellen Interaktion die gestische Modalität gewinnbringend für einen natürlichen und intuitiven Umgang mit dem Computer einzusetzen.

Die Anwendungsfelder einer gestischen Steuerung sind sehr vielfältig. Zunächst erscheint sie prädestiniert für die „virtuelle Realität“, die sich schon seit ihren Anfängen immer mit hinderlichen Datenhandschuhen beholfen hat [Kru91]. Auch ein CAD- (*computer aided design*) und jedes dreidimensionale Visualisierungssystem profitieren von der gestischen Eingabe: bisher wird im allgemeinen mit gewöhnungsbedürftigen, funktional erweiterten taktilen Eingabegeräten gearbeitet [Lan94b]. Weitere Anwendungen liegen beispielsweise in der Robotersteuerung (intuitive Eingabemöglichkeit), im medizinischen Bereich (besondere hygienische Anforderungen), in der Fertigungs- und Schwerindustrie (große Anforderungen an Schmutzresistenz), in der Raumfahrt und in gefährlichen Umgebungen (Behinderung durch Schutzkleidung) und bei öffentlich zugänglichen Systemen wie Informationsterminals, Verkaufs- und Bankautomaten (Schutz gegen Vandalismus).

Viele Haushaltsgeräte und Geräte aus dem Bereich der Konsum- und Unterhaltungselektronik enthalten „versteckte“ Computer, ohne mit eigentlich notwendigen teueren und unhandlichen Eingabeterminals versehen zu sein. Hier ließe sich sicherlich die oft schwierige Handhabbarkeit durch eine Komponente der visuellen Interaktion wesentlich vereinfachen. Dies ist umso wichtiger in Umgebungen wie dem Automobil, in denen die Aufmerksamkeit nicht durch die Bedienung von Informations- und Navigationsgeräten gebunden werden darf. Auch ließe sich damit der Nutzerkreis von Geräten erweitern, die bisher von Menschen aufgrund einer spezifischen Behinderung nicht bedient werden können.

Nicht zuletzt kann die visuelle Interaktion als zusätzlicher und wesentlicher Baustein in einem alle natürlichen Modalitäten umfassenden Dialog zwischen Mensch und Maschine betrachtet werden (s. Kap. 12).

1.2 Aufgabenstellung

1.2.1 Randbedingungen und Folgerungen für das System

Ziel dieser Arbeit war die Konzeption und Umsetzung eines Systems für die gestisch-visuelle Interaktion. Eine visuelle Interaktion ist dabei gekennzeichnet durch die Verbindung der üblichen visuellen Ausgabe mit einer visuellen Eingabe. Zwei wesentliche Randbedingungen waren dabei vorgegeben:

1. Als Eingabemodalität wurden *ausschließlich Gesten* betrachtet, was ihrer großen Bedeutung in der zwischenmenschlichen Kommunikation Rechnung trägt (vgl. Kap. 1.1). Damit wurde das Repertoire möglicher Gesten implizit auf die sog. *autonomen* Gesten eingeschränkt: das sind solche, die nicht auf weitere Modalitäten wie beispielsweise die Sprache angewiesen sind. Es zeigte sich, daß die Verwendung autonomer Gesten keine Einschränkung darstellt (s. Kap. 2.4).

   Mit dieser Vorgabe wurde es möglich, das Potential der gestischen Modalität unabhängig von anderen Modalitäten zu untersuchen und voll auszuschöpfen. Es konnte gezeigt werden, daß — in Verbindung mit einem geeigneten Dialogkonzept — ei-

ne graphische Anwendung prinzipiell ausschließlich durch Gesten gesteuert werden kann (s. Kap. 3 und 4 sowie [Mor98a]).

2. Trotz der Beschränkung auf die gestische Modalität sollte das System für andere Modalitäten offen bleiben. Aus diesem Grund wurde ein Systemaufbau gewählt, der in seiner räumlichen Anordnung einem typischen *Computerarbeitsplatz* entspricht: der Benutzer sitzt am Schreibtisch vor Monitor, Tastatur und Maus. Durch die sitzende Haltung beschränken sich die gestischen Äußerungen des Benutzers gezwungenermaßen auf die obere Körperhälfte. Außerdem ist nur der Oberkörper des Benutzers für das System uneingeschränkt sichtbar (s. Kap. 4.1).

   Diese Anordnung erleichtert für zukünftige Systeme die Verbindung mit weiteren Modalitäten erheblich: neben den konventionellen haptischen Eingabemedien wie Tastatur und Maus können auch sehr leicht Modalitäten wie beispielsweise Sprache, Handschrift, Skizzen und auch Mimik optimal integriert werden.

   Die typischen Aufgaben an einem Computerarbeitsplatz erfordern den Einsatz bewußter oder sog. *primärer* Gesten, mit denen sich Anwendungen gezielt steuern lassen. Nur solche Gesten wurden in dieser Arbeit untersucht (vgl. Kap. 2.2 und 2.4).

Es wird die These vertreten, daß sich mit den in dieser Arbeit vorgestellten Verfahren auch unbewußte oder gefühlsgesteuerte gestische Äußerungen verarbeiten und erkennen lassen. Für diese Art von Gesten ergeben sich allerdings völlig andere als die in dieser Arbeit vorgestellen Anwendungen.

Um das System offen zu halten für andere Modalitäten, aber auch um eine natürliche Art der Interaktion zu ermöglichen, darf es *nicht intrusiv* sein. Somit entfallen zur Erfassung von Gesten Datenhandschuhe oder gar Sensoranzüge, wie sie teilweise in der virtuellen Realität oder zur Steuerung von künstlichen Schauspielern eingesetzt werden [Fol90, Spe95]. Jede Art von Bewegungs- und Positionssensoren, die an Händen und Kopf befestigt und verkabelt werden müssen, behindern den Benutzer in seiner Bewegungsfreiheit, erfordern teilweise große Vorbereitungen und machen somit die Vorteile, die mit dem Zugewinn von Modalitäten gewonnen werden, wieder zunichte.

Aus diesen Überlegungen folgt, daß für einen visuellen Dialog nur *Videokameras* als berührungsfreies, nicht-intrusives und unscheinbares Eingabemedium verwendet werden dürfen. Die Beobachtungen müssen dann mit Mitteln der Bildinterpretation ausgewertet werden, so daß eine *automatische Erkennung* der Gesten ermöglicht wird. Auf eine stereoskopische Bildauswertung wurde von vornherein verzichtet, um die erforderliche Rechenleistung in Grenzen zu halten. Die guten Erkennungsergebnisse in Kap. 8 bestätigen, daß eine monokulare Gestenmodellierung ausreicht.

1.2.2 Dynamische Gesten und gestenspezifisches Dialogkonzept

Damit ein natürlicher und intuitiver Dialog entstehen kann, müssen die Stärken und Schwächen einer gestischen Interaktion identifiziert, analysiert und zu einem *Dialogkonzept* verarbeitet werden, mit dem die Möglichkeiten einer gestischen Interaktion optimal ausgenutzt werden können. Dazu mußte zunächst geklärt werden, worum es sich bei dem Phänomen der Gestik eigentlich handelt (s. Kap. 2). Anschließend wurde ein am gestischen Dialog zwischen Menschen orientiertes Dialogkonzept entwickelt (s. Kap. 3). Dieses Konzept wurde in umfangreichen Usability-Experimenten optimiert und validiert (s. Kap. 4).

Eine wichtige Erkenntnis aus den in Kap. 2 zusammengestellten Überlegungen soll hier vorweggenommen werden: bei Gesten handelt es sich grundsätzlich um *Bewegungen*. Da wesentliche Teile der Information verloren gehen, wenn Gesten lediglich als statische Erscheinung betrachtet werden, stellt diese Arbeit ausschließlich Verfahren vor, die *dynamischen Gesten* gerecht werden. Damit muß ein System für die Erkennung von Gesten bedeutungstragende Bewegungen in Bild*sequenzen* erkennen können. Darüberhinaus müssen auch selbständig die Zeitabschnitte detektiert werden können, in denen solche bedeutungstragenden Bewegungen auftreten.

Das in dieser Arbeit vorgestellte System grenzt sich unter anderem genau in diesem *gestenspezifischen Dialogkonzept* im Verbund mit der *bildverarbeitungsgestützten* Erkennung *dynamischer* Gesten von den meisten bestehenden gestischen Dialogsystemen deutlich ab (vgl. Betrachtungen und Literatur in Kap. 3.1 und Überblick in [Hua95]).

1.2.3 Aufgaben für die Bildinterpretation

In der Sprache der Bildinterpretation galt es daher folgende drei Hauptaufgaben zu lösen:

1. Die *räumliche Segmentierung* dient der Trennung von interessierenden Bildobjekten vom Hintergrund. Für die räumliche Segmentierung wird die im Bild enthaltene Farbinformation ausgewertet. Das genaue Vorgehen ist in Kap. 5 dargestellt.

2. Die *Klassifikation von Bildsequenzen* weist einer Bewegung, deren zeitliche Grenzen zunächst als bekannt vorausgesetzt werden, ihre eigentliche Bedeutung zu. Die Klassifikation beruht auf der stochastischen Modellierung von Bildsequenzen mit speziell angepaßten Hidden-Markov-Modellen, die in Kap. 6 näher beschrieben werden. Der Einsatz dieser Modelle für die Bildsequenzerkennung erfordert spezifische Verfahren der Merkmalsextraktion, die in Kap. 7 vorgestellt werden. Die endgültige Auswahl eines Merkmalsextraktionsverfahrens erfolgt nach der zu erzielenden Erkennungsrate sowie nach weiteren, anwendungsabhängigen Kriterien. Die Ergebnisse dieser Betrachtungen finden sich in Kap. 8.

3. Die *zeitliche Segmentierung von Bildsequenzen* liefert die Grenzen der bedeutungstragenden Bewegungen. Die zeitliche Segmentierung ist Teil der kontinuierlichen Erkennung und wird in Kap. 9 dargestellt. Die Bewertung der beiden untersuchten Verfahren (s. Kap. 1.4) und die zu erzielenden kontinuierlichen Erkennungsergebnisse sind in Kap. 10 zusammengestellt.

1.2.4 Stochastische Modellierung von Bildsequenzen mit Hidden-Markov-Modellen

Hidden-Markov-Modelle (HMMs) werden schon seit den sechziger Jahren sehr erfolgreich in den unterschiedlichsten Disziplinen eingesetzt [Bau67]. Im Verlauf der achtziger Jahre verdrängten sie in der Spracherkennung aufgrund ihrer überlegenen Erkennungsleistung [Rab89, Hua90] die bis dahin üblicherweise verwendeten Verfahren der dynamischen Programmierung [Dav80, Fel84] zunehmend. Dazu trug auch bei, daß die Struktur der Modelle zunächst um ein Sprachmodell (s. beispielsweise [Rab89, Pla95]) und in jüngerer Zeit noch zusätzlich um ein semantisches Modell [Mül97, Sta97] konsistent erweitert wurde, so daß mit einer integralen Erkennung über alle diese Bereiche ein stochastischer Ansatz zum Sprachverstehen verwirklicht werden konnte.

Mit Beginn der neunziger Jahre wurden HMMs erfolgreich in weiteren Disziplinen wie der Erkennung kontinuierlicher Handschrift (vgl. z. B. [Nat95]), der Formelerkennung [Win96] oder der Gesichtserkennung (z. B. [Nef98]) angewandt. Ebenso hielten HMMs durch die Pionierleistung von [Yam92] Einzug in die Bild*sequenz*erkennung. Daß HMMs für die Erkennung von Bildsequenzen im allgemeinen und für die bildbasierte Erkennung dynamischer Gesten im speziellen bis zur zweiten Hälfte der neunziger Jahre dennoch nur in Einzelfällen eingesetzt wurden, hat unter anderem folgende Gründe:

1. Da die Verarbeitung von Bildsequenzen viel Rechenleistung benötigt, werden traditionell heuristische Algorithmen eingesetzt, die in der Regel sehr schnell oder sogar echtzeitfähig sind. Allerdings sind die resultierenden Algorithmen nicht sehr leistungsfähig und meist nur für einen genau definierten Einsatzfall anwendbar (s. beispielsweise [Dar95, Hie96, Jo98]). Dieses Problem verliert immer mehr an Bedeutung, weil inzwischen schon Standard-PCs für viele Bildverarbeitungsoperationen genug Prozessorleistung zur Verfügung stellen, so daß teuere Spezial-Hardware überflüssig wird.

2. Im Gegensatz zu vielen heuristischen Verfahren wird bei den HMM-basierten Verfahren viel Trainingsmaterial benötigt. Es ist daher sehr aufwendig, die großen benötigten Datenmengen für das Training zu akquirieren und zu speichern. Dies gilt aber auch für andere stochastische Verfahren zur Bewegungsklassifikation (z. B. [Wil95]) sowie für die prinzipiell ebenfalls einsetzbaren künstlichen neuronalen Netze (z. B. [Rei96]). Mit der in jüngerer Zeit zur Verfügung stehenden Kompressions- und Speicherhardware verschwindet allerdings auch dieses Problem des hohen Datenaufkommens.

Die ersten HMM-basierten Systeme zur Bildsequenzerkennung verwenden heuristische Vorverarbeitungs- bzw. Trainingsmethoden, so daß nur die prinzipielle Funktionsfähigkeit gezeigt werden kann (Klassifikation von 6 Tennisschlägen [Yam92] und 3 Handgesten [Sch94a]). Auch die später erscheinenden Systeme sind nicht sehr leistungsfähig, da nur einfache Merkmalsextraktionsverfahren in Kombination mit wenig leistungsfähigen HMMs eingesetzt werden (Erkennung von typischerweise 10 unterschiedlichen Gesten in [Rig96, Sch96b]; Erkennung von 10 Lippenbewegungen in [Chi96]; Erkennung der Bewegung von 11 auf die Eckpunkte reduzierten einfachen geometrischen Körpern (*moving light displays*) in [Fie95]). Trotz willkürlicher, an keine reale Anwendung gebundene Definition extrem ausladender und damit leicht diskriminierbarer Ganzkörpergesten gelingt daher auch in [Rig97, Rig98] die Erkennung eines etwas größeren Vokabulars von 24 Gesten nur mit mäßiger Erkennungsleistung. Größere Gestenkataloge mit realistischen Gesten können dagegen aufgrund der einfachen Vorverarbeitung und Modellierung nur mit Hilfsmitteln erkannt werden (Erkennung von 40 Gesten der amerikanischen Zeichensprache mit zwei einfarbigen Handschuhen in [Sta95]; Erkennung von 35 gestischen Bewegungsmustern mit einem sechsfarbigen Handschuh und zwei Farbmarkern an Ellbogen und Schultern in [Hie96]; Erkennung von 262 Gesten der niederländischen Zeichensprache mit einem einfarbigen und einem siebenfarbigen Handschuh in [Ass98]).

Die in dieser Arbeit vorgestellte und aufeinander abgestimmte Kombination aus Vorverarbeitung (s. Kap. 5), Merkmalsextraktion (s. Kap. 7) und einer Modellierung basierend auf semikontinuierlichen HMMs (s. Kap. 6) ist dagegen in der Lage, ein großes Vokabular realer Dialogdaten mit sehr guten Erkennungsraten ohne weitere Hilfmittel zur Vereinfachung der Bildvorverarbeitung zu klassifizieren (s. Kap. 8 basierend auf [Mor97a,

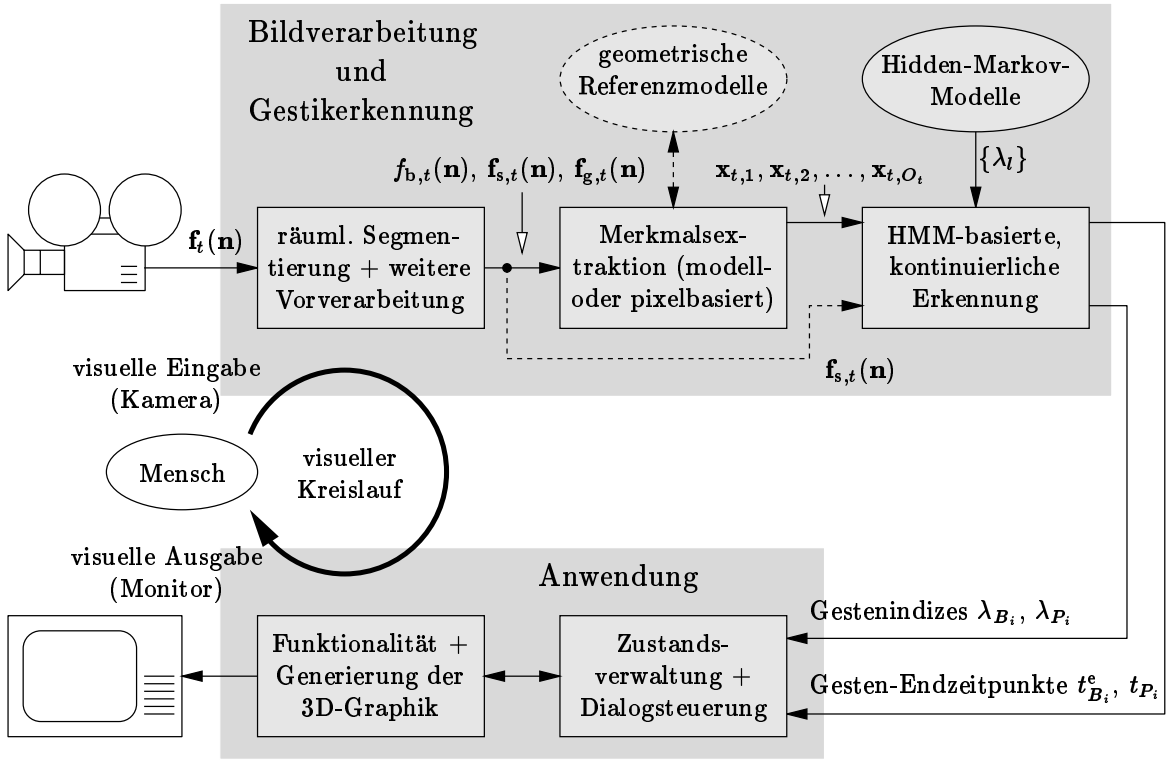


Bild 1.1: Systemüberblick

Mor97b]). Darüberhinaus werden auch zwei auf HMMs basierende Ansätze zur *kontinuierlichen* Erkennung präsentiert, wie sie für ein realistisches Dialogszenario unbedingt erforderlich ist (s. Kap. 1.4).

1.3 Systemüberblick

Im Systemüberblick in Bild 1.1 sind die notwendigen Komponenten im Zusammenhang dargestellt. Die Kamera auf der Eingabeseite liefert einen permanenten Bildstrom; der Einfachheit halber sei angenommen, daß dieser Bildstrom schon in digitalisierter Form vorliegt. Das digitalisierte Bild $\mathbf{f}_t(\mathbf{n})$ mit den Ortskoordinaten $\mathbf{n} = (n_1, n_2)$ zum Zeitpunkt t wird der räumlichen Segmentierung und eventuell einer weiteren Vorverarbeitung unterzogen (s. Kap. 5). Für die Weiterverarbeitung steht somit alternativ die binäre Segmentierungsmaske $f_{b,t}(\mathbf{n})$, das segmentierte Bild $\mathbf{f}_{s,t}(\mathbf{n})$ oder das auf dem segmentierten Bild berechnete Gradientenbild $\mathbf{f}_{g,t}(\mathbf{n})$ zur Verfügung. Es folgt die Merkmalsextraktion, die mit verschiedenen Methoden realisiert werden kann (s. Kap. 7). Wird ein modellbasierter Ansatz verwendet, so sind die gestrichelt dargestellten Referenzmodelle erforderlich. Die Merkmalsextraktion liefert pro Bild einen oder mehrere Merkmalsvektoren der Form $\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,O_t}$. Zeitliche Segmentierung und Klassifikation sind unter der HMM-basierten, kontinuierlichen Erkennung zusammengefaßt: in Kap. 1.4 wird darauf näher eingegangen. Die kontinuierliche Erkennung liefert am Ausgang Gestenindizes λ_{B_i} bzw. λ_{P_i} und Zeitpunkte $t^e_{B_i}$ bzw. t_{P_i}, die das Ende der entsprechenden Gesten angeben (s. Kap. 9). Damit ist die Erkennungsseite abgeschlossen.

Im unteren Zweig sind die Komponenten dargestellt, welche der Anwendung zuzuordnen sind, die von den gefundenen Gesten gesteuert werden soll. Von der Anwendung wird zunächst nur gefordert, daß sie eine visuelle Ausgabe liefert; ansonsten ist jede Art

von Anwendung denkbar, die räumlich manipulierbar ist (genaueres s. Kap. 3.3). Als typische Anwendung wurde in dieser Arbeit ein dreidimensionaler Szenen-Editor implementiert, der sich grob aufspalten läßt in einen speziell an die gestische Steuerung angepaßten Verwaltungsteil (Zustandsverwaltung und Dialogsteuerung) und die eigentliche Editor-Funktionalität zusammen mit der Graphikgenerierung (s. Kap. 3.5).

Aus Sicht des Systems reagiert der Mensch auf die visuelle Ausgabe mit neuen visuellen Eingaben: der Mensch stellt eine visuelle Rückkopplung her. Aus menschlicher Sicht reagiert das System auf eine visuelle Eingabe mit einer visuellen Ausgabe, die ebenfalls als visuelle Rückkopplung betrachtet werden kann. Diese gegenseitige Beeinflussung führt daher zu einem *visuellen Kreislauf*, auch wenn beide, Mensch und System, zur „Berechnung" der Reaktionen den visuellen Bereich verlassen.

1.4 Strukturen für die kontinuierliche Erkennung

In der realen Dialoganwendung muß die Erkennung in der Lage sein, Gesten im kontinuierlichen, nicht abreißenden Videostrom zu finden und zu klassifizieren (vgl. Bild 1.1), was hier *kontinuierliche Erkennung* genannt wird. Diese Aufgabenstellung ist *nicht* mit der ebenfalls als kontinuierlich bezeichneten *Satz*erkennung in der Spracherkennung zu vergleichen, da dort nicht von einem endlosen Eingangsstrom, sondern von vorsegmentierten Satzeinheiten ausgegangen wird [Rab89, Pla95]. Auch in der Gestikerkennung gibt es solche Ansätze, die sich dann analog mit der Erkennung von Zeichensprache in vorsegmentierten Satzeinheiten beschäftigen (s. beispielsweise [Ass98], allerdings sind farbkodierte Handschuhe erforderlich, vgl. Kap. 1.2.4).

Systeme für die kontinuierliche Gestikerkennung sind in der Forschungsliteratur nur selten anzutreffen, obwohl eine kontinuierliche Erkennung für die praktische Anwendung in einem visuellen Dialog unabdingbar ist. In dieser Arbeit werden zwei Alternativen zur kontinuierlichen Erkennung untersucht und verglichen [Mor98b, Mor98c, Mor99]. Beide Ansätze lassen sich auf identische Weise in das in Bild 1.1 gezeigte Gesamtsystem einbetten:

1. Der erste Ansatz ist zweistufig und beinhaltet im Kern einen HMM-basierten Erkenner für eine isolierte Erkennung (s. Bild 1.2). Die zeitliche Segmentierung wird durch einen vorgeschalteten, regelbasierten Bewegungsdetektor erreicht. Dieser Ansatz wird in Kap. 9.2 näher beschrieben und in Kap. 10.6 mit den realen Dialogdaten evaluiert.

   Vergleichbare zweistufige Ansätze finden sich in [Hof98, Lia98]: hier sind ebenfalls Segmentierung und HMM-basierte isolierte Erkennung getrennt. Allerdings arbeiten beide Systeme mit Datenhandschuhen und sind deshalb nicht mit dem Ansatz dieser Arbeit vergleichbar.

   Zwei weitere zweistufige Ansätze aus der Literatur sind bildverarbeitungsbasiert. In [Wat98] werden allerdings *zwei* Kameras benötigt, so daß Tiefeninformation über die Gesten gewonnen werden kann. Die Bildsequenzen werden in Symbolsequenzen umgeformt und mit einem einfachem Symbolvergleich erkannt. Bei acht verschiedenen Gesten ergibt sich jedoch nur eine schwache Erkennungsleistung. Im zweiten Ansatz wird nur eine Kamera benötigt [Yea99]. Die Erkennung der regelbasiert segmentierten Daten erfolgt mit einem endlichen Zustandsautomaten. Statt einer echten Evaluierung werden jedoch nur Plausibilitätstests durchgeführt.

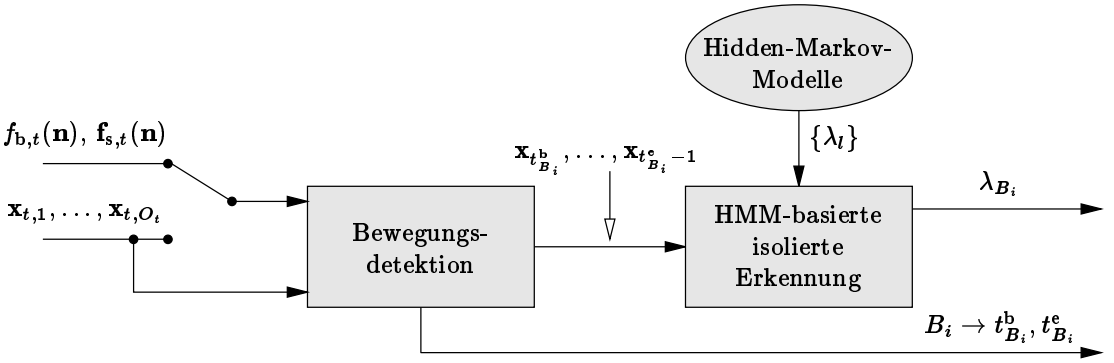


Bild 1.2: Zweistufige kontinuierliche Erkennung

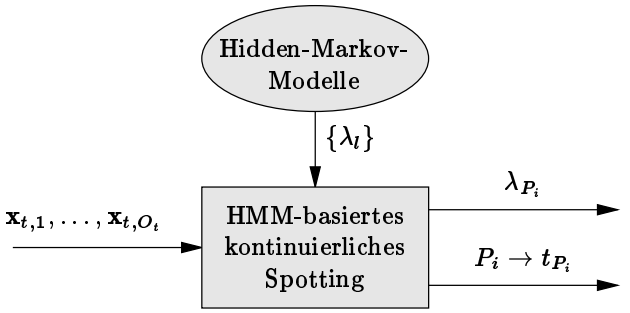


Bild 1.3: Einstufige kontinuierliche Erkennung (Spotting)

2. Der zweite Ansatz ist ein einstufiger HMM-basierter *Spotter*, der Segmentierung und Klassifikation in einem integralen Schritt ausführen kann (s. Bild 1.3). Der Spotting-Algorithmus ist eine erweiterte Form des Algorithmus für die isolierte Erkennung und wird in Kap. 9.3 besprochen. Die Evaluierung erfolgt unter anderem mit denselben Daten wie beim zweistufigen Ansatz in Kap. 10.7.

   Die beiden vergleichbaren einstufigen Ansätze in der Literatur sind ebenfalls bildverarbeitungsbasiert und arbeiten beide mit einer kontinuierlichen Version der dynamischen Programmierung (vgl. z. B. [Fel84]). Während in [Nag96] keine Evaluierung durchgeführt wird, ergibt sich in [Nis96] für acht Gesten nur eine geringe Erkennungsleistung. Die Erfahrungen aus der Spracherkennung lassen von vornherein vermuten, daß ein *HMM*-basierter Ansatz im Vergleich zur dynamischen Programmierung eine bessere Erkennungsleistung zeigen wird (vgl. Bemerkungen in Kap. 1.2.4), was die Evaluierung des Spotting-Ansatzes dieser Arbeit bestätigt.

1.5 Übersicht über die Arbeit

In Kap. 2 wird zunächst auf die Gestik aus geisteswissenschaftlicher Sicht eingegangen, so daß eine Charakterisierung ermöglicht wird. Systeme zur kategorialen Klassifikation von Gesten erlauben dann eine Einschätzung, wie sinnvoll es ist, Gesten für einen visuellen Dialog einzusetzen. Mit diesen Erkenntnissen und der Verallgemeinerung von Beispielen gelingt es in Kap. 3, ein allgemeines Konzept für den visuellen Dialog aufzustellen und als Beispielanwendung einen 3D-Szenen-Editor zu konzipieren. Mit diesem Konzept wurden Usability-Untersuchungen durchgeführt, die in Kap. 4 zusammen mit den dort gewonnenen Erkenntnissen vorgestellt werden. Durch die Versuche konnten insbesondere ein

sinnvoller Katalog von benötigten Gesten aufgestellt sowie Daten für das Training und die Evaluierung einer automatischen Erkennung gewonnen werden.

Die weiteren Kapitel beschäftigen sich mit der algorithmischen Umsetzung der Erkennungskomponenten eines Systems für den visuellen Dialog. Zunächst werden dazu in Kap. 5 die nötigen Schritte für die räumliche Segmentierung und eine eventuell notwendige weitere Vorverarbeitung vorgestellt. Als zentrale Mustererkennungs-Instanz kommen Hidden-Markov-Modelle (HMMs) zum Einsatz. Die in dieser Arbeit verwendeten Modelle werden zusammen mit den Besonderheiten, die zu beachten sind, wenn man HMMs für die Erkennung von Bildsequenzen einsetzt, in Kap. 6 vorgestellt. Kap. 7 beschäftigt sich mit dem zentralen Thema der Merkmalsextraktion, die das Bindeglied zwischen dem zu verarbeitenden Videobildstrom und der gewählten stochastischen Modellierung darstellt. Zur Auswahl eines Merkmalsextraktionsverfahrens und zur Evaluierung der Leistungsfähigkeit der Modellierung werden in Kap. 8 Ergebnisse für die isolierte Erkennung von Bildsequenzen präsentiert.

Kap. 9 stellt zwei alternative Verfahren vor, mit denen eine kontinuierliche Erkennung durchgeführt werden kann, wie sie für den praktischen Einsatz eines visuellen Dialogsystems unbedingt notwendig ist. Die Ergebnisse in Kap. 10 gestatten dann einen Vergleich der beiden Ansätze und eine realistische Einschätzung der Leistung des Gesamtsystems.

Die Erkenntnisse aus den Evaluierungen der isolierten und kontinuierlichen Erkennung erlaubten es, einen Demonstrator für die visuelle Interaktion am Beispiel des gestengesteuerten, dreidimensionalen Szenen-Editors zu implementieren. Die Konzeption dieses Demonstrators wird in Kap. 11 vorgestellt. Kap. 12 zeigt schließlich auf, an welchen Stellen das System weiterentwickelt werden kann.

Kapitel 2

Grundlagen der Gestik

2.1 Geschichtlicher Abriß

Die Erforschung von Gesten hat eine lange Tradition, die mindestens bis in das 17. Jahrhundert zurückreicht. Man erhoffte sich, durch die Untersuchung von Gesten Aufschluß über den Ursprung der Sprache und die Natur des Denkens zu erhalten. Mit Beginn des 20. Jahrhunderts erlahmte jedoch das Interesse an Gestik, wie auch die Forschung nach dem Ursprung der Sprache in den Hintergrund trat. Für die Linguistik stellte die Beschäftigung mit Gesten vermeintlich keine Bereicherung dar, weil man Gesten zu sehr auf individuelle Äußerungen reduzierte, die sich nicht in ein festes grammatikalisches Regelwerk einbetten ließen. Obwohl man sich in der Psychologie zunehmend für die nonverbale Kommunikation zu interessieren begann, hielt man Gesten für wenig relevant, weil man mit ihnen bewußte Aktionen verband, die zudem zu stark durch gesellschaftliche Konvention eingeschränkt und zu eng mit dem Sprachlichen verbunden waren.

Das Wiedererwachen des Interesses an der Evolution der Sprache, der Beginn der linguistischen Untersuchung von Zeichensprachen und die überraschende Tatsache, daß es gelang, Schimpansen eine Zeichensprache beizubringen, ließen die Gestik mit Beginn der siebziger Jahre unseres Jahrhunderts wieder in einem neuen Licht erscheinen. Es zeigte sich in der Linguistik weiterhin, daß gesprochene Äußerungen oft nur deshalb „funktionieren“, weil sie in einen Kontext anderer Verhaltensformen — wozu auch die Gestik gehört — eingebettet sind. Außerdem wurden Gesten in der Psychologie als eine symbolische Ausdrucksform interessant, da man sich mit höheren mentalen Prozessen zu beschäftigen begann [Ken86].

Erst in den letzten Jahren begann man auch in der Disziplin der Mensch-Maschine-Kommunikation Gestik als Kommunikationsmittel zu entdecken. Die Forschung auf diesem Gebiet wird dabei stets vom Wunsch getrieben, die Natürlichkeit der Schnittstelle zwischen Mensch und Computer zu steigern [Hua95].

2.2 Definition und Kennzeichen

Der Begriff „Gestik“ ist in der Literatur sehr unterschiedlich gefaßt, da es schwierig ist, eine Definition zu finden, die für jede wissenschaftliche Disziplin und Situation passend ist. Es scheint lediglich eine Einigung darüber zu bestehen, daß Änderungen in der Erscheinung des Körpers einer Person eine Wirkung bei einer anderen Person hervorrufen

[Ekm95]. Eine der restriktiveren Definitionen, die zudem gut zur Anwendung der Gestik im visuellen Dialog paßt, ist in einer Formulierung von Kendon enthalten [Ken86]:

> Das Wort „Geste“ dient als Bezeichnung für denjenigen Bereich sichtbarer Handlung, den Kommunikationsteilnehmer routinemäßig „herausfiltern“ und bei dem davon ausgegangen wird, daß er von einer offen eingestandenen, kommunikativen Absicht geleitet wird[1].

Hervorzuheben an dieser Definition sind nun folgende Aspekte (diese Begriffe werden im Umfeld der obigen Formulierung in [Ken86] näher erläutert):

1. Gesten sind sichtbare Handlungen, also *Bewegungen*.
2. Jeder Mensch hat die Fähigkeit, Gesten leicht als solche zu erkennen und sie von anderen Arten der Bewegung abzugrenzen.
3. Bewegungen gelten nur dann als Gesten, wenn sie in kommunikativer Absicht vorgebracht werden.

Punkt 2 ist nur implizit formuliert und muß noch näher spezifiziert werden. Kendon fand dazu in einem Experiment heraus, daß Menschen mit sehr großer Übereinstimmung Bewegungen als Gesten identifizieren konnten, auch wenn ihnen die Bedeutung der Gesten unbekannt war. Folgende Charakteristika wurden für die Identifikation von Bewegungen als Gesten herangezogen:

- Werden *Gliedmaßen* abrupt vom Körper wegbewegt und kehren sie anschließend wieder in ihre Ausgangsposition zurück, so wird eine Geste wahrgenommen.
- Rotations- oder Nickbewegungen des *Kopfes*, die schnell oder wiederholt ausgeführt werden, sind Gesten, wenn sie nicht in eine neue Kopfposition münden oder wenn sie nicht in Verbindung mit Augenbewegungen ausgeführt werden.
- Bewegungen des gesamten *Körpers* gelten als Gesten, wenn sie zur Ausgangsposition zurückkehren und nicht in eine permanent veränderte Körperhaltung oder -position münden.

Gesten sind also erkennbar an einem scharfen Einsatz der Bewegung. Außerdem haben sie eher den Charakter eines symmetrischen „Exkurses“ als den einer bleibenden Positionsänderung.

Allgemein wird davon ausgegangen (s. [Ken86] und Überblick in [Wex94]), daß der Bewegungsablauf von Gesten in 3 Phasen eingeteilt werden kann (*Drei-Phasen-Modell*): Der Gesten*kern* (engl. *nucleus* oder auch *stroke* genannt) enthält das Bewegungssegment, das die eigentliche Bedeutung trägt. Dem Kern geht eine *Vorbereitungsbewegung* voraus, die dazu dient, den an der Geste beteiligten Körperteil in eine definierte Position zu bewegen, von der aus die Kernbewegung ausgeführt wird. Eine *Nachbereitungsbewegung* führt schließlich vom Endpunkt der Kernbewegung wieder in die Ruheposition.

Im oben genannten Punkt 3 ist Kendons Definition wesentlich enger gefaßt als andere Definitionen. Damit Bewegungen als Gesten gelten, muß mit ihnen *bewußt* etwas *ausgedrückt* werden (*deliberately expressive movements*). *Keine* Gesten nach dieser Definition sind:

[1] Die Formulierung lautet im Original: *The word 'gesture' serves as a label for that domain of visible action that participants routinely separate out and treat as governed by an openly acknowledged communicative intent.*

- *praktische* Bewegungen, mit denen beispielsweise ein Objekt manipuliert oder mit denen ein Dialogvorgang hergestellt und aufrechterhalten wird (beispielsweise durch Anpassung und Änderung der Distanz und Orientierung zum Dialogpartner);
- meist unbewußte *nervöse*, *komfortbezogene* Bewegungen, wie das Spielen an Ringen und Armbändern, das Herumstreichen im Haar, das Ordnen der Kleider;
- unwillkürliche Bewegungen in Verbindung mit *Gefühlsäußerungen* (wären solche Gefühlsäußerungen aufgesetzt, so würde man allerdings von Gesten sprechen).

Es ist wichtig festzuhalten, daß auch Bewegungen, die keine Gesten sind, eine Bedeutung haben und etwas über den „Sender“ aussagen können. Diese Aussagen sind allerdings nicht gewollt und nicht in kommunikativer Absicht übermittelt worden.

Es gibt Indizien dafür, daß für Gesten einerseits und für unbewußte und praktische Bewegungen andererseits verschiedene „Erkennungssysteme“ im menschlichen Gehirn existieren, die bewirken, daß Gesten bewußt und andere Bewegungen in der Regel unbewußt vom Rezipienten aufgenommen werden. Dementsprechend konnte experimentell bestätigt werden, daß die beiden Bewegungsklassen verschiedenen Erzeugungszentren entspringen: so ist die Fähigkeit zum Erzeugen von Gesten an das Sprachzentrum gebunden, während die nicht-gestischen Bewegungen auch bei gestörtem Sprachzentrum hervorgebracht werden können [Ken86].

2.3 Gestik und Sprache

Gemäß der oben aufgeführten Definition ist der prinzipielle Zweck von Gestik und Sprache derselbe: beide erfüllen die Aufgabe, Aktionsmuster zu produzieren, die für andere eine Bedeutung repräsentieren. Die Art und Weise, wie die beiden Modalitäten ihre Aufgabe erfüllen, ist jedoch sehr unterschiedlich: Gesten sind ein visuelles Medium, dem sowohl Raum als auch Zeit zur Verfügung stehen, während sich das auditive Medium Sprache nur auf die zeitliche Dimension stützen kann.

In vielen Untersuchungen wurde gezeigt, wie Gestik und Sprache zusammen auftreten. Gesten können dabei je nach der Situation und dem auszudrückenden Sachverhalt (visuelle) Ergänzungen zur Sprache liefern, sie können sprachliche Mehrdeutigkeiten auflösen und sie können im Wechsel mit Sprache auftreten. Beide Modalitäten können natürlich auch völlig getrennt verwendet werden. Dies mag für Sprache selbstverständlich sein, führt aber bei Gesten zu der Erkenntnis, daß sie in vielen Situationen auch *autonom* sind. Durch diese Symmetrie wird deutlich, daß Gesten nicht etwa primitiver oder weniger vielseitig als Sprache sind. Es handelt sich bei Gestik auch nicht einfach nur um ein Anhängsel oder eine Dekoration der Sprache. Schließlich ist Gestik schon gar nicht redundant, so daß man sie einfach weglassen könnte [Ken86].

Kendon führt weiterhin aus, daß in der menschlichen Entwicklungsgeschichte entsprechend den vorherrschenden Bedingungen die Entscheidung für die gesprochene Sprache gefallen war, so daß sie sich zu dem mächtigen Medium entwickelte, das sie heute darstellt. An der weit gediehenen Entwicklung der verschiedenen Zeichensprachen ist jedoch erkennbar, daß entsprechende äußere Umstände auch dazu hätten führen können, daß sich die Gestik zu einem ebenso mächtigen Kommunikationsinstrument wie die Sprache weiterentwickelt. Kendon stellt daher die These auf, daß Gestik und Sprache als Kommunikationsmedien in ihrer Flexibilität, Allgemeingültigkeit und Ausdruckskraft gleichwertig sind [Ken86].

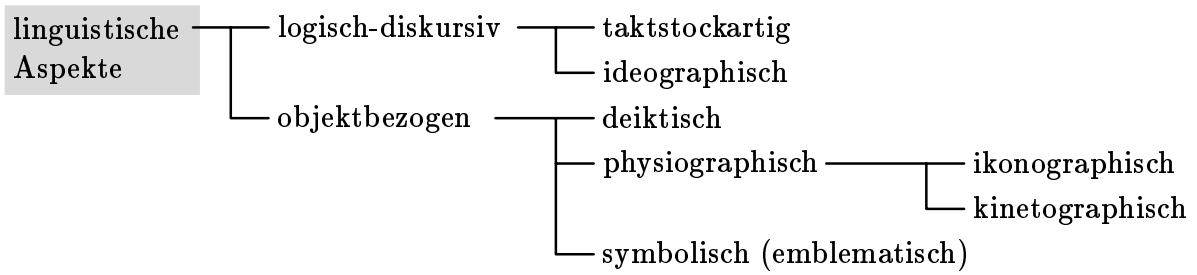


Bild 2.1: Ausschnitt aus dem Klassifikationssystem von Efron (aus [Efr72])

2.4 Klassifikationssysteme und Beispiele

2.4.1 Klassifikationssystem von Efron

Efron war der Pionier in der quantitativen empirischen Untersuchung von Gestik und Mimik als Teil der sozialen Interaktion (s. [Efr72] mit der Neuauflage seines Werkes von 1941). Das Klassifikationssystem bildet für Efron das Gerüst seiner Beobachtungen, mit denen er nachweisen kann, daß ein typisch gestisches Verhalten durch das soziale Umfeld und nicht durch die „Rasse“ geprägt wird. Kendon gliedert nach drei unterschiedlichen Gesichtspunkten: dem räumlich-zeitlichen, dem kommunikativ-interaktiven und dem linguistischen Aspekt.

Die linguistische Gliederungsart ist für die Anwendung von Gesten in dieser Arbeit am aufschlußreichsten (s. Bild 2.1). Während die *logisch-diskursiven* Gesten immer nur in Verbindung mit Sprache sinnvoll sind, sind die *objektbezogenen* Gesten prinzipiell von der Sprache unabhängig, wobei sie auch in Verbindung mit Sprache auftreten können. Kendons Untersuchungen bezogen sich allerdings immer nur auf sprachbegleitende Gesten. Die logisch-diskursiven Gesten werden in *taktstockartige* Gesten unterteilt, die die aufeinanderfolgenden Bedeutungseinheiten der Sprache betonen, und in die *ideographischen* Gesten, die den Entstehungsprozeß und die „Richtung“ der Gedanken visualisieren.

Die objektbezogenen Gesten unterteilen sich in *deiktische* Gesten, mit denen auf ein Objekt direkt gezeigt wird, *physiographische* Gesten, die das gemeinte Objekt visualisieren, und *symbolische* oder *emblematische* Gesten, die ein visuelles oder logisches Objekt in einer abstrahierten und stilisierten Weise repräsentieren und die damit einer Konvention bedürfen. Die physiographischen Gesten werden nochmals unterteilt in *ikonographische* Gesten, die die Form eines Objektes beschreiben, und *kinetographische* Gesten, durch die eine Bewegung nachgeahmt wird.

Es ist zu erwarten, daß die objektbezogenen Gesten in gestengesteuerten Anwendungen die größte Rolle spielen.

2.4.2 Klassifikationssystem von Morris

Bei vielen Gliederungen fällt auf, daß mehrheitlich von sprachbegleitenden Gesten ausgegangen wird (s. beispielsweise [Efr72, Nes86, Wex94]). Da sich diese Arbeit mit einem rein gestischen Dialog auseinandersetzt, scheinen viele der Gliederungen zunächst nicht anwendbar zu sein. Allerdings wird bei vielen sprachbegleitenden Gesten immer wieder betont, daß diese zwar häufig in Verbindung mit Sprache auftreten, daß sie allerdings auch unabhängig von Sprache gebraucht werden und somit eine eigenständige Bedeutung haben.

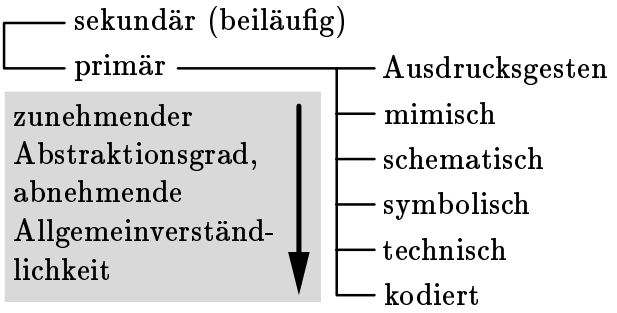


Bild 2.2: Ausschnitt aus dem Klassifikationssystem nach Morris (aus [Mor81])

Morris war der erste, der sich überwiegend mit autonomen Gesten befaßte (s. beispielsweise [Mor81]). Ein wichtiger Ausschnitt aus seiner pragmatischen Einteilung autonomer Gesten ist in Bild 2.2 dargestellt. Morris unterscheidet zunächst zwischen *sekundären* oder beiläufigen und *primären* Gesten. Bei den beiläufigen Gesten handelt es sich um mechanische oder unbewußte Handlungen mit unbeabsichtigter Aussagekraft (die nach Kendon also gar keine Gesten darstellen). Die primären, bewußt zur Informationsübermittlung eingesetzten Gesten werden wiederum in sechs Gruppen unterteilt, von der sich die erste Gruppe der *Ausdrucksgesten* wiederum von den anderen Gruppen absetzt. Die Ausdrucksgesten dienen der Mitteilung des Gefühlszustandes und werden insbesondere auch durch die Gesichtsmimik übermittelt.

Die restlichen fünf Kategorien sind so angeordnet, daß sich ein zunehmender Abstraktionsgrad bei gleichzeitig abnehmender Allgemeinverständlichkeit ergibt. Am wenigsten abstrahiert sind dabei die *mimischen* Gesten, die Personen, Dinge oder Vorgänge imitieren. Die *schematischen* Gesten sind standardisierte Kürzel imitierender Gesten, die schon gewisser Konventionen bedürfen. Durch *symbolische* Gesten werden Stimmungen und Gedanken wiedergegeben, also abstrakte Konzepte, die sich nicht durch Objekte repräsentieren lassen. Die *technischen* Gesten werden zur effektiven Verständigung innerhalb von Spezialistengruppen verwendet, wozu sie genau standardisiert sein müssen. *Kodierte* Gesten bilden Zeichensprachen auf der Grundlage formaler Systeme.

2.4.3 Beispielsituationen für die Verwendung autonomer Gesten

Es gibt sehr viele Situationen, in denen autonome Gesten verwendet werden. Um die Wichtigkeit und Verbreitung autonomer Gesten zu untermauern, sind einige typische Beispielsituationen in Tabelle 2.1 aufgelistet (teilweise aus [Mor81]). Die Situationen lassen sich in drei Gruppen einteilen, die mit der Motivation zusammenhängen, aus der heraus autonome Gesten zum Einsatz kommen [Mor98a]:

1. Die Verständigung durch Sprache ist aus physikalischen oder physiologischen Gründen nicht oder nur schlecht möglich.

2. Die Sprache als akustisches Signal ist störend.

3. Es wird bewußt auf Sprache verzichtet.

Wenn auch nicht alle Situationen zum Alltag gehören, so sind sie doch fast alle allgemein bekannt, was nicht heißt, daß alle diese Gesten von jedermann beherrscht werden.

Gr.	Situation	Kategorie	Motivation	Bemerkungen
1	Baustelle (Verständigung mit Kranführer)	technisch	große Entfernung	Gesten sind standardisiert
	Flughafen (Einweiser auf Rollfeld)	symbolisch–technisch	Lärm, Trennscheibe	vereinbarte Gesten, teilweise allgemeinverständlich, Stäbe als Hilfsmittel
	Feuerwehr	technisch	Lärm, große Entfernung	vereinbarte Gesten
	Fußball (Schiedsrichter, Linienrichter)	symbolisch–technisch	Lärm, große Entfernung	vereinbarte Gesten, teilweise mit Flaggen als Hilfmittel
	Börse (Parkett)	technisch	Lärm	streng vereinbart, ermöglichen quantifizierte Kaufs- und Verkaufsaktionen
	Stummfilm	mimisch–symbolisch	Medium ohne Ton	allgemeinverständliche Gesten; fast vollständiger Ersatz von Sprache möglich
	Verständigung beim Tauchen	technisch	Sprechen unter Wasser nicht möglich	vereinbarte Zeichen, müssen erlernt werden
	Auto zum Parken einweisen	mimisch–symbolisch	Trennscheibe, große Entfernung	bekannte Alltagssituation
	Flaggenalphabet	technisch	große Entfernung	sehr großes Vokabular, Flaggen als Hilfsmittel
	Taubstummensprache	kodiert	Behinderung	vollständiger Ersatz der Sprache, national unterschiedlich
2	Fernseh- und Tonstudio	symbolisch–technisch	Trennscheibe, Sprache störend	vereinbarte Verständigung zwischen Toningenieur und Künstlern
	Orchester	mimisch–technisch	Sprache störend	Verständigung zwischen Dirigent und Musikern, stark individuell mit vereinbarten Elementen
3	Pantomime	mimisch–symbolisch	künstlerisches Ausdrucksmittel	Allgemeinverständlichkeit gefordert
	Verständigung im Ausland	mimisch–symbolisch	mangelnde Sprachkenntnisse	kulturübergreifende Allgemeinverständlichkeit gefordert

Tabelle 2.1: Beispielsituationen für die Verwendung autonomer Gesten (Gr. = Gruppe, Kategorie nach Morris, s. Text)

2.4.4 Klassifikationssystem für den visuellen Dialog

Aus der Einteilung in Bild 2.2 geht hervor, daß sich Abstraktionsgrad und Allgemeinverständlichkeit gegenläufig verhalten. Da ein visueller Dialog ohne Lernphase verlaufen

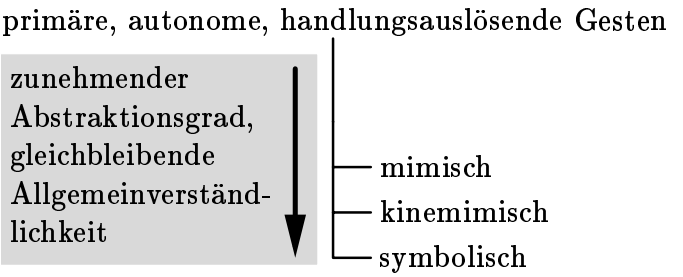


Bild 2.3: Klassifikationssystem für den visuellen Dialog: Einteilung nach Abstraktionsgrad

soll, ist insbesondere der Allgemeinverständlichkeit der Gesten eine große Bedeutung beizumessen. Daher kommen hochgradig spezialisierte technische oder kodierte Gesten für einen visuellen Dialog nicht in Frage.

Im Rahmen der Entwicklung eines Konzeptes für den visuellen Dialog (s. Kap. 3) stellte sich heraus, daß einerseits viele der nach Bild 2.2 klassifizierten Gesten nicht für einen Dialog verwendbar sind. Andererseits ist das Raster für die tatsächlich verwertbaren Dialoggesten zu grob.

Wie im Dialogkonzept im nächsten Kapitel noch ersichtlich wird, sind für den gestischen Mensch-Maschine-Dialog *handlungsauslösende* Gesten oder *gestische Direktiven* erforderlich. Selbstverständlich werden für die gestische Steuerung auch bewußte (primäre) und autonome Gesten vorausgesetzt. In den Klassifikationssystemen entsprechen die *objektbezogenen* Gesten noch am meisten denen, die im visuellen Dialog zu erwarten sind (s. [Efr72, Nes86]). Die im Dialog vorkommenden Gesten lassen sich in drei Gruppen einteilen (s. Bild 2.3):

1. **Mimische Gesten:** Das sind Gesten, die Objekte zusammen mit ihrer Bewegung direkt nachahmen (Begriff allgemein üblich; Verwendung beispielsweise in [Mor81] und [Nes86]).

2. **Kinemimische Gesten:** Sie lösen Bewegungen aus, in dem sie den Bewegungsvorgang oder die Bewegungstendenz nachahmen. Dazu zählt beispielsweise die große Klasse der Winkbewegungen (sie entsprechen den kinetographischen Gesten bei [Efr72]; der Begriff „kinemimisch“ wird synonym dazu bei [Nes86] gebraucht).

3. **Symbolische Gesten:** Sie ahmen Vorgänge auf abstrakterer Ebene nach, wobei der konkrete Bezug im Verlauf der gestischen Entwicklungsgeschichte immer mehr verloren gegangen ist (Verwendung wie in [Mor81]; entspricht den emblematischen Gesten bei [Efr72]). Es sollen allerdings nur Gesten zur Anwendung kommen, die so verbreitet sind, daß sie keiner Erklärung bedürfen.

Die Kategorien in Bild 2.3 sind so angeordnet, daß sich ein steigender Abstraktionsgrad (wie in Bild 2.2) ergibt. Allerdings wurde über die Durchführung von Usability-Experimenten (s. Kap. 4) sichergestellt, daß unabhängig von der Kategorie nur allgemein bekannte Gesten verwendet werden. Damit ergibt sich in Bild 2.3 eine gleichbleibende Allgemeinverständlichkeit.

2.5 Analyse von Gesten im Dialog

Bei den Usability-Versuchen (s. Kap. 4) stellte sich heraus, daß es sehr wichtig ist, von einem *Klassifikationssystem* auszugehen, damit die Vielzahl der vorkommenden Bewe-

gungen strukturiert beobachtet werden kann. Die Strukturierung hilft einerseits bei der Interpretation, andererseits lassen sich damit Ausprägungen von Gesten vorhersagen, die zwar bis zu einem gewissen Beobachtungszeitpunkt noch nicht aufgetreten sind, deren Kategorie aber aufgrund des Klassifikationssystems schon bekannt ist.

Dagegen erschien es weder notwendig, ein bestehendes *Gestennotationssystem* zu verwenden, noch ein eigenes System einzuführen. Ein solches Notationssystem wurde beispielsweise schon in [Efr72] konzipiert und angewendet. Dort machte es aufgrund der vielfältigen Situationen, die untersucht wurden und die eine enorm große Anzahl unterschiedlichster Gesten auftreten ließen, sowie der weniger weit entwickelten technischen Möglichkeiten zur Darstellung von Bewegungen sicherlich Sinn. Durch die Möglichkeit, Gesten mit Hilfe von Videoaufnahmen genauestens zu reproduzieren, war ein Notationssystem im Rahmen dieser Arbeit nicht notwendig, zumal die betrachtete Domäne eine im Vergleich zu den Untersuchungen in [Efr72] geringe Anzahl von Gesten „hervorbrachte". Bei den Usability-Experimenten hat sich gezeigt (s. Kap. 4), daß es tatsächlich ausreicht, die Gesten verbal zu skizzieren und in Zweifelsfällen typische Momentaufnahmen von Gestensequenzen darzustellen (s. Tabellen B.4 und B.5 in Anh. B.3.1). Außerdem war es jederzeit möglich, Gesten vom Videoband oder in digitalisierter Form direkt auf dem Rechner in der originalen Bewegung wiederzugeben.

Ein Notationssystem kann auch dabei behilflich sein, Gesten in Bewegungsprimitive zu zerlegen. Da es durch die gewählte Art der Modellierung möglich ist, die Gesten als Ganzes zu repräsentieren, stellt sich die Zerlegungsaufgabe in dieser Arbeit nicht: die Modelle führen beim Training diese Zerlegung vielmehr selbsttätig durch (s. Kap. 6). Durch die Analyse der Modellzustände wäre man somit sogar umgekehrt in der Lage, Bewegungsprimitive aus der Modellierung abzuleiten. Dies könnte dann notwendig werden, wenn ein *wesentlich* größeres Gestenvokabluar als für den visuellen Dialog benötigt (s. Kap. 4) hierarchisch modelliert werden müßte, wie dies bei aktuellen Systemen zur Spracherkennung üblich ist (s. beispielsweise [Pla95, Rei96, Sta97, Mül97]).

Kapitel 3

Konzept einer visuellen Interaktion

3.1 Bestehende gestische Dialogsysteme

Die meisten bisher existierenden Forschungssysteme, in denen ausschließlich Gesten zur Steuerung eingesetzt werden, orientieren sich an der Steuerung graphischer Benutzeroberflächen: die Gesten werden als Mausersatz verwendet (s. beispielsweise [Que95, Fre95, Nes95, Fre96, Kje96, Que96a, Que96b, Bow97]).

Das gilt prinzipiell auch für viele gestengesteuerte Systeme der *virtuellen Realität*: zwar kann man hier mitunter in die zu beeinflussenden Anwendungen „eintauchen"; allerdings wird meist am Bedienkonzept graphischer Benutzungsoberflächen — der *direkten Manipulation* — festgehalten. Auch bei der zusätzlich erforderlichen dreidimensionalen Navigation durch den virtuellen Raum wird die Handorientierung zusammen mit einer vereinbarten Menge von Handkonfigurationen *direkt* in einen Bewegungsvorgang umgesetzt (z. B. in [Kru91, Mag93, Mag94, Brö95, Pav96, Jo98]). Allerdings ist bei dreidimensionalen Strukturen die direkte Manipulation mit Gesten im Vergleich zur Mausbedienung (wie z. B. in [Hou92, Ven93]) wesentlich einfacher zu handhaben.

Im Konzept der *augmented reality* werden reale Objekte und Umgebungen auf innovative Art mit virtuellen Objekten angereichert. Oft spielen auch hier Gesten als intuitive Eingabemodalität eine große Rolle. Trotzdem wird auf konventionelle Metaphern der direkten Manipulation zurückgegriffen (beispielsweise [Wel91, New92, Wel93a, Wel93b, Cro95]).

Multimodale Systeme, in denen Gesten im Verbund mit anderen Modalitäten wie z. B. Sprache eingesetzt werden, sind weit seltener anzutreffen. Bei solchen Systemen werden die Aktionen meist durch die Sprache initiiert und durch einfache Gestik ergänzt, so daß kein tragfähiges visuelles Dialogkonzept abgeleitet werden kann (beispielsweise das *Put-That-There*-System [Bol80] und andere [Wei89, Bol92]). Andere Systeme wiederum behandeln Gestik und Sprache gleichberechtigt, greifen jedoch für die Gestik auf die oben skizzierten Bedienkonzepte der virtuellen Realität zurück (z. B. [Pav97]).

Um mit Gesten einen sinnvollen visuellen Dialog gestalten zu können, muß ein Dialogkonzept eingesetzt werden, das sich an den spezifischen Eigenschaften und Einsatzmöglichkeiten von Gesten orientiert [Mor98a]. Für die Erarbeitung eines allgemeingültigen Konzeptes ist es hilfreich, sich an allgemein bekannten zwischenmenschlichen Situationen zu orientieren, in denen ebenfalls ausschließlich Gesten verwendet werden. Viele solcher Beispielsituationen wurden schon in Kap. 2.4.3 vorgestellt. Es handelt sich beispielsweise

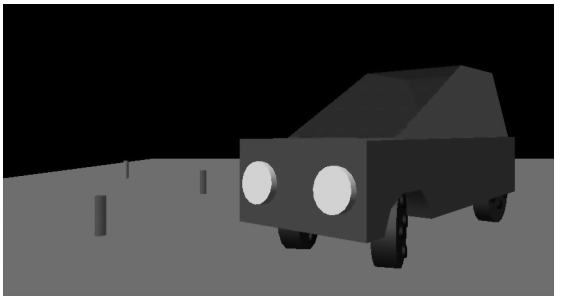


Bild 3.1: Beispielszene aus dem Auto-Einparksimulator (Auto von vorne)

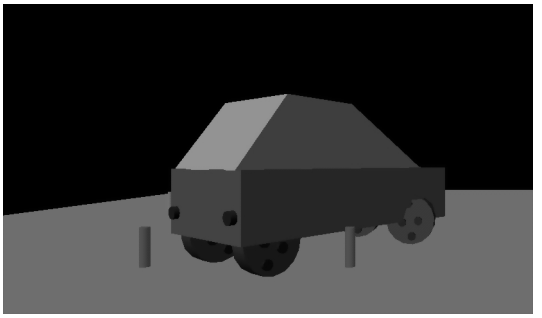


Bild 3.2: Beispielszene aus dem Auto-Einparksimulator (Auto beim rückwärts Einparken)

um eine solche typische Situation, wenn eine Person einen Autofahrer mit Gesten in eine Parklücke dirigiert.

3.2 Anwendung 1: Auto-Einparksimulator

Als erste Anwendung im Rahmen dieser Arbeit wurde daher ein *Auto-Einparksimulator* implementiert [Vol97]. Dabei handelt es sich um eine Studie für den gestischen Dialog und weniger um eine sinnvolle Anwendung. Nur wenige Kommandos sind in dieser Simulation zulässig, um ein stilisiert dargestelltes Auto von einer Ausgangsposition heraus in eine durch Pfosten markierte Parklücke zu dirigieren (s. Bild 3.1 und 3.2): (1) vorwärts beschleunigen, (2) rückwärts beschleunigen, (3) Lenkungsinkrement rechts, (4) Lenkungsinkrement links, (5) stopp.

3.3 Grundsätze eines gestischen Dialogs

Diese Einparksituation enthüllt eine Vielzahl der Grundsätze, die vom Menschen intuitiv im gestischen Dialog eingehalten bzw. vorausgesetzt werden:

Verwendung dynamischer Gesten (G1): Im allgemeinen werden im gestischen Dialog *dynamische Gesten* verwendet. So werden bei den ersten vier oben genannten Befehlen für die Bedienung des Einparksimulators üblicherweise Wink*bewegungen* eingesetzt. Auch bei der Stoppgeste handelt es sich um einen Bewegungsvorgang (etwa das Hochreißen beider Hände) und nicht nur um eine statische Handhaltung.

Dies wird auch schon in der Gestendefinition in Kap. 2.2 deutlich: hier wird ausschließlich von Bewegungen ausgegangen. Selbst vermeintlich statische Gesten, wie beispielsweise das Zeigen, folgen dem Drei-Phasen-Modell und werden von Vor- und Nachbereitungs*bewegungen* begleitet. Außerdem wird die Gültigkeit der Zeigerichtung im Kern der Zeigegeste in der Regel von einem kleinen „Ruck“ angezeigt.

Obwohl statische Gesten im zwischenmenschlichen Dialog nicht verwendet werden, bilden sie die Grundlage fast aller in Kap. 3.1 erwähnten Forschungssysteme für den gestischen Dialog (vgl. [Hua95]).

Verwendung weniger, allgemeingültiger Gesten (G2): Situationen wie die Hilfe beim Einparken des Autos lassen sich ohne vorherige Einigung auf ein Gestenvo-

kabular meistern. Um dies auch für den gestischen Mensch-Maschine-Dialog zu gewährleisten, sollten nur *wenige, allgemein übliche Gesten* verwendet werden.

Sind komplexere Handlungen als beim Auto-Einparksimulator erforderlich, so besteht die Möglichkeit, diese in der Anwendung räumlich zu repräsentieren (siehe zweite Anwendung in Abschnitt 3.5.3). Damit kann eine eventuell spezielle und nicht mehr allgemeinverständliche Geste vermieden und in eine Serie einfacher Gesten aufgelöst werden.

Vergegenständlichung und indirekte Manipulation (G3): Die Umsetzung der Anwendungen sollte sich an bekannten (Alltags-)Situationen orientieren. Dazu ist es erforderlich,

- die Funktionalität einer Anwendung räumlich zu repräsentieren (zu „vergegenständlichen“) und
- eine Anwendung nur indirekt über sogenannte gestische Direktiven zu manipulieren (Prinzip der *indirekten Manipulation*).

Jede Anwendung, die von sich aus schon räumlicher Natur ist (etwa ein CAD-System oder der in Kap. 3.5 vorgestellte Szenen-Editor) oder die sich mit einer räumlichen Metapher repräsentieren läßt (zwei- oder dreidimensional), ist grundsätzlich für den indirekt wirkenden gestischen Dialog geeignet.

3.4 Kennzeichen der indirekten Manipulation

Die indirekte Manipulation steht im Gegensatz zur heute üblichen Technik der direkten Manipulation bei mausbasierten graphischen Benutzeroberflächen. Davon ausgehend basieren auch viele der bestehenden gestischen Dialogsysteme auf einer direkten Beeinflussung des Geschehens, die oft proportional zur Handbewegung erfolgt (vgl. Beispiele in Kap. 3.1). Im gestischen Dialog zwischen Menschen ist eine solche direkte Beeinflussung nicht üblich. Kennzeichen der indirekten Manipulation sind:

Simulation eines Kommunikationspartners (K1): Es muß einen Kommunikationspartner oder *Agenten* geben, der als Teil der Anwendung graphisch repräsentiert wird.

Im Auto-Einparksimulator ist der Kommunikationspartner der (gedachte) Fahrer des in der Szene dargestellten Autos.

Eigenständig ablaufende Aktionen (K2): Dieser Agent nimmt die gestischen Direktiven des Benutzers entgegen, interpretiert sie und führt daraufhin weitgehend *eigenständige Aktionen* aus, die in Form von Animationen dargestellt werden.

Im Einparksimulator ist als Reaktion auf eine Direktive lediglich implizit eine Änderung der Bewegung des Autos sichtbar. In anderen Anwendungen können solche Animationen allerdings explizit notwendig sein.

Zustandsabhängigkeit (K3): Eine gestische Direktive bewirkt immer nur eine *Änderung* der vorhandenen Situation. Die Anwendung benötigt also eine *Zustandsverwaltung* als Gedächtnis für die zuletzt eingegangene Direktive.

Eine Zustandsverwaltung ermöglicht es darüberhinaus, verschiedene Dialogmodi bereitzustellen, so daß identische Gesten kontextabhängig verschiedene Wirkungen hervorrufen können. Der graphisch repräsentierte Dialogpartner kann in diesem Zusammenhang dazu verwendet werden, diese inneren Zustände zu visualisieren.

Eine Geste bewirkt im Einparksimulator eine Änderung des Bewegungszustandes des Autos: es wird schneller oder langsamer, der Lenkungseinschlag erhöht sich oder wird geringer. Der Bewegungszustand wird beibehalten, bis eine neue Direktive eintrifft.

3.5 Anwendung 2: 3D-Szenen-Editor

3.5.1 Allgemeine Beschreibung und Befehlsumfang

In einer zweiten Anwendung wurden nun alle oben genannten Grundsätze umgesetzt. Hierbei handelt es sich um einen gestengesteuerten *3D-Szenen-Editor*, dessen Bedienung wesentlich komplexere Aktionen als der Einparksimulator erfordert [Vol97]. Im Editor können Objekte im dreidimensionalen Raum *erzeugt*, *frei positioniert*, *in ihrer Orientierung verändert* und auch wieder *gelöscht* werden. Außerdem ist es möglich, Eigenschaften der Objekte wie *Größe* und *Farbe* zu ändern. Schließlich kann *die Orientierung* und *die Beobachtungsdistanz* der gesamten Szene variiert werden. Dazu stehen insgesamt 16 Kommandos zur Verfügung (s. Tabelle B.6 in Anh. B.3.1: die Kommandos entsprechen den Gesten*kategorien*).

In den nachfolgend beschriebenen Usability-Versuchen (s. Kap. 4) hatten die Objekte die Form von Möbelstücken, was die Anschauung erleichtern sollte. Man konnte sich also die Aufgabe vorstellen, einen Raum einzurichten (s. Bild 3.3). Das Programm kann leicht so modifiziert werden, daß auch andere Objekte für die Generierung zur Verfügung stehen; somit können prinzipiell beliebige dreidimensionale Szenen interaktiv entworfen werden.

Alle Aktionen im Szenen-Editor werden indirekt vorgenommen (siehe Grundsatz G3 für den gestischen Dialog in Abschnitt 3.3). Der Kommunikationspartner (siehe Kennzeichen K1 der indirekten Manipulation in Abschnitt 3.4) wird durch eine Hand symbolisiert. Die Hand als handelndes Element (K2) hat gleichzeitig den Vorteil, daß sich mit ihr die inneren Zustände des Editors sehr leicht visualisieren lassen (K3).

3.5.2 Visualisierung des Programmzustandes und einfacher Handlungen

Im Neutralmodus (s. Bild 3.4) — dem Grundzustand — läßt sich die Hand in sechs aufeinander senkrecht stehende Richtungen dirigieren (drei Dimensionen, jeweils zwei Richtungen). In diesem wie auch in allen anderen Zuständen beendet sich die durch eine gestische Direktive initiierte Bewegung nach einer gewissen Zeit von selbst. Zum Zurückzulegen größerer Strecken muß also in regelmäßigen Abständen dieselbe Geste immer wieder dargeboten werden. Kommen die Gesten sehr dicht hintereinander, so wird dies als „Drängen“ interpretiert und die Bewegung der Hand wird schneller. Mit einer Stoppgeste kann die Bewegung jederzeit angehalten werden.

Nähert sich die Hand einem Objekt, so wird die Farbe dieses Objektes etwas heller, was signalisiert, daß dieses Objekt nun bei Bedarf gegriffen werden kann (s. Übergang von Bild 3.6 nach 3.7). Wird die Hand vom Objekt weg dirigiert, normalisiert sich die Farbe

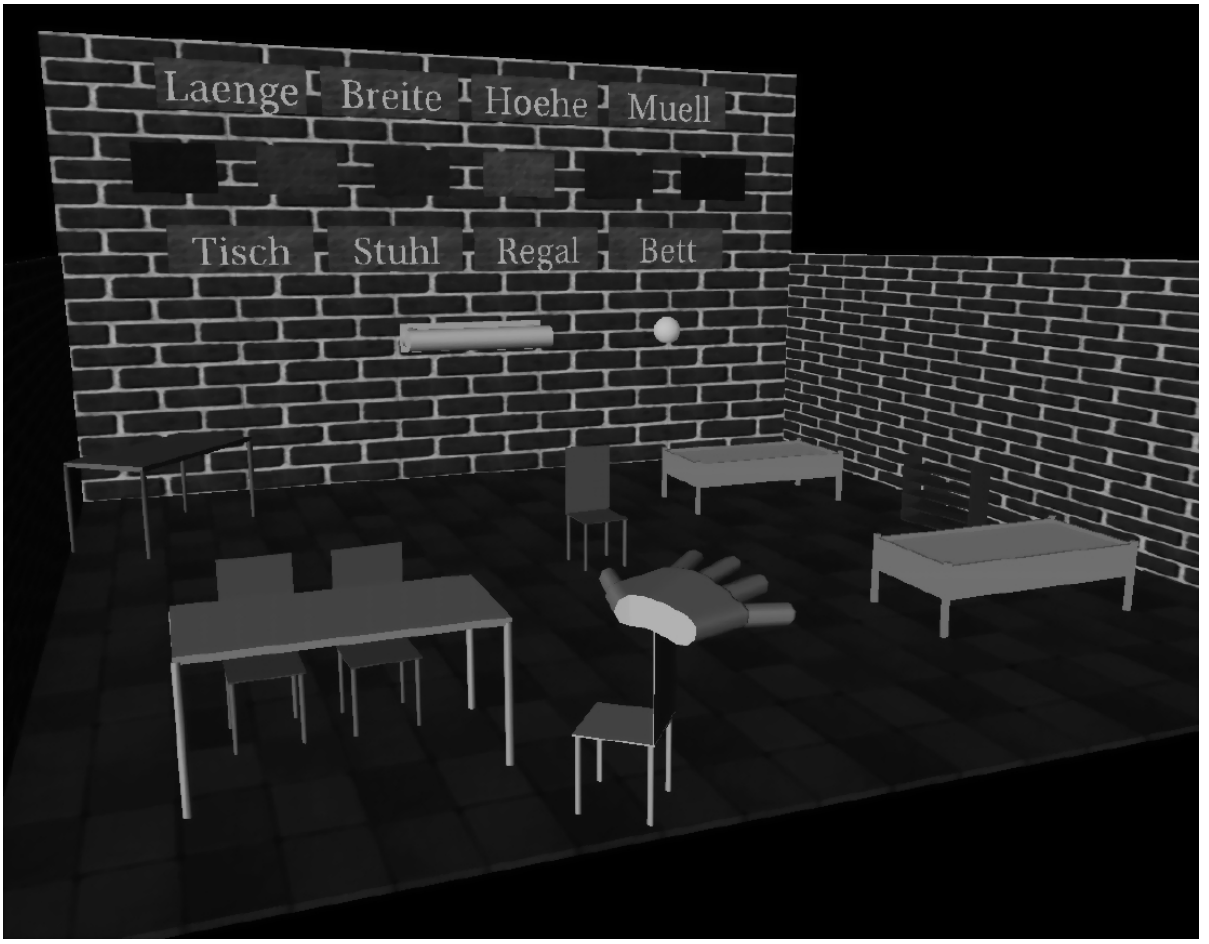


Bild 3.3: Beispielszene aus dem 3D-Szenen-Editor

des Objektes wieder. Erfolgt aber eine Greifgeste, während ein Objekt heller dargestellt ist, so wandert die Hand auf direktem Weg zum Objekt und greift es (s. Bild 3.8). Ist keine eindeutige Entscheidung möglich, weil sich mehrere Objekte im Einflußbereich der Hand befinden, wird ein Auswahldialog eingeblendet, der mit entsprechenden Gesten zugunsten des gewünschten Objektes beendet werden kann.

Ein gegriffenes Objekt kann nun zusammen mit der Hand in die sechs Raumrichtungen bewegt werden. Außerdem läßt sich das Objekt mit entsprechenden Gesten auch in seiner Orientierung ändern (drehen). Eine Loslaß- oder Annulierungsgeste bewirkt, daß sich die Hand wieder vom Objekt löst. Eventuell fällt das Objekt dann unter dem Einfluß der „Schwerkraft“ nach unten.

Um die Selektion eines Objektes abzukürzen, kann die Hand durch eine entsprechende Geste auch in einen Zeigemodus versetzt werden. Hierin ist ein Objekt greifbar, wenn es von einem Selektionsstrahl getroffen wird, der vom Zeigefinger ausgeht (s. Bild 3.5). Im Zeigemodus kann die Hand nur noch um Hoch- und Querachse gedreht werden. Teilweise führen Gesten, die im Neutralmodus eine Translation bewirken, nun zu einer Rotation: abhängig vom Kontext werden identische Gesten also unter Umständen verschieden interpretiert (K3). Soll ein vom Selektionsstrahl getroffenes Objekt gegriffen werden, so bewegt sich die Hand zum Objekt hin und wird zur Faust, was dann nicht mehr vom Greifvorgang aus dem Neutralmodus heraus unterscheidbar ist (s. Bild 3.8). Zwischen dem Zeige- und Neutralmodus kann beliebig hin und her gewechselt werden.

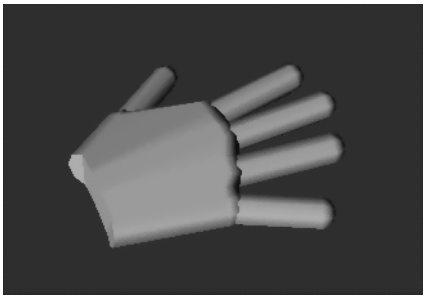


Bild 3.4: Neutralmodus

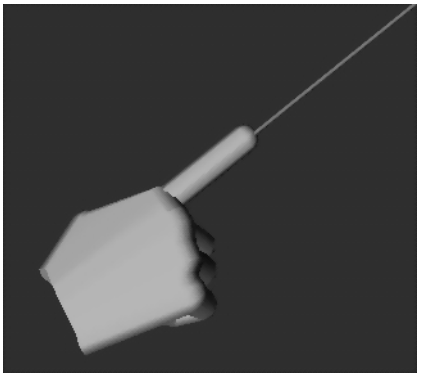


Bild 3.5: Zeigemodus mit Selektionsstrahl

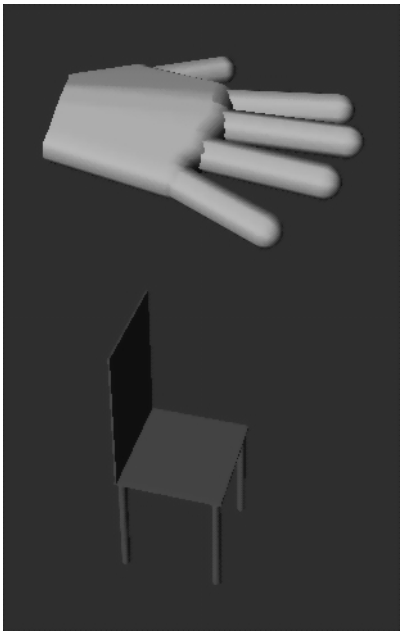


Bild 3.6: Annäherung an ein Objekt

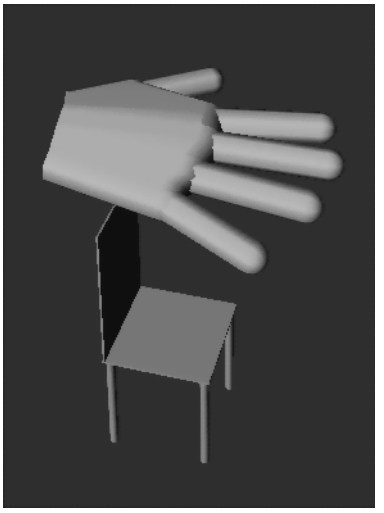


Bild 3.7: Objekt greifbar im Neutralmodus

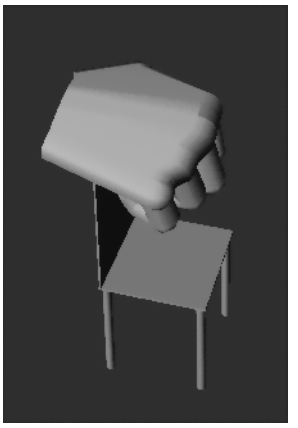


Bild 3.8: Objekt gegriffen

3.5.3 Umsetzung und Visualisierung komplexer Handlungen

Komplexe Handlungen wie das Erzeugen und Löschen von Objekten sowie das Ändern ihrer Farbe und Größe werden räumlich durch Tasten an der hinteren Begrenzungsebene des dreidimensionalen Raumes repräsentiert (s. Bild 3.3). Diese Tasten sind damit zwar fest im Raum positioniert, in Bezug auf ihre Selektierbarkeit werden sie aber wie alle anderen Objekte behandelt: eine solche Taste wird durch Greifen oder Drücken aktiv.

Eine Gegenstandstaste erzeugt ein ihr zugeordnetes neues Objekt, das im Raum unter die entsprechende Taste gestellt wird.

Alle anderen Tasten bewirken, daß die Hand nun zur stets vorhandenen Selektionsfähigkeit zusätzliche Fähigkeiten erhält, die graphisch durch Symbole unter der Hand symbolisiert werden:

- Ein Hammer zeigt an, daß das nächste gegriffene Objekt zerstört (d. h. gelöscht) wird (s. Bild 3.9).

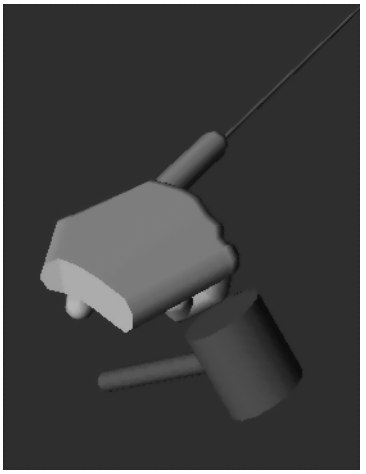


Bild 3.9: Löschmodus

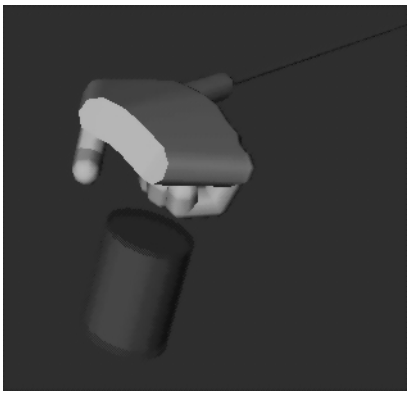


Bild 3.10: Färbemodus

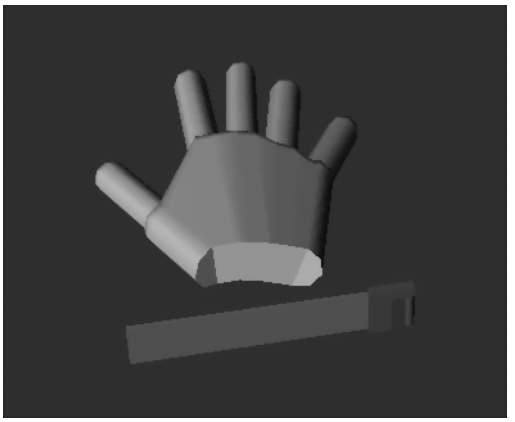


Bild 3.11: Skalierfähigkeit (führt in den Skaliermodus)

- Ein Farbeimer steht für die Fähigkeit der Hand, ein Objekt neu zu färben (s. Bild 3.10).
- Eine Säge bedeutet, daß die Hand in der Lage ist, Objekte in ihrer Größe zu ändern (s. Bild 3.11). Wird ein Objekt aus diesem Modus heraus gegriffen, so wird es gewissermaßen elastisch und kann mit entsprechenden Gesten gestaucht oder gestreckt werden.

Greift die Hand den „Weltgriff“, der unterhalb der Tasten angebracht ist (s. Bild 3.3), wird die Bewegung der Hand auf die gesamte Szene umgelenkt. Damit kann der Raum um seine Hoch- und Querachse gedreht sowie insgesamt nach vorne oder hinten geschoben werden.

Alle komplexen Handlungen erfordern also keine speziellen und komplizierten Gesten. Sie werden, wie unter Grundsatz G2 gefordert, in eine Abfolge von einfachen Gesten aufgelöst. Mit dieser Methode können in einer Anwendung beliebige Funktionen konsistent durch einfache Gesten gesteuert werden.

Kapitel 4

Usability-Experimente und Datengewinnung

4.1 Aufbau der Wizard-of-Oz-Versuche

Das Dialogkonzept wurde im Rahmen von sogenannten *Wizard-of-Oz*-Versuchen im Usability-Labor getestet (s. beispielsweise [Hau89]). Die Probanden und der Versuchsleiter (oder *Wizard*) saßen dabei in getrennten Räumen. Die schematische Struktur des Versuches ist in Bild 4.1 verdeutlicht.

Die Tatsache, daß das System für einen *Arbeitsplatzdialog* (s. Kap. 1.2.1) geeignet sein soll, spielte für den Versuchsaufbau eine große Rolle. Da der Benutzer bei dieser Art des Dialogs am Schreibtisch vor dem Computermonitor sitzt, ist in der Regel nur etwa die obere Körperhälfte sichtbar. Im Hinblick auf Gestik bedeutet dies, daß es ausreicht, lediglich die Hände und den Kopf des Benutzers zu beobachten. Es ist nicht nötig, die Arme zu betrachten, weil sich die Armbewegungen — soweit für die Gestik wichtig — implizit aus den Handbewegungen ergeben.

In der sitzenden Haltung und konzentriert auf die visuelle Rückmeldung auf dem Monitor (s. Kap. 3.5.2) sind auch keine sehr ausladenden Hand- und Kopfbewegungen zu erwarten. Es reicht also aus, als mögliche Aufenthaltszonen von Kopf und Händen nur relativ kleine räumliche Bereiche zu beobachten. Weiterhin ist zu erwarten, daß die Gesten nicht nur über die Handtrajektorie, sondern auch durch Verändern der Handform gebildet werden, weil es sich um „Nahdistanzgesten“ handelt. Im Unterschied dazu stehen „Weitdistanzgesten“, die insbesondere aus ausladenden Armbewegungen bestehen (beispielsweise verwendet in [Rig97]). Hier sind Veränderungen der Hand von untergeordneter Bedeutung, da sie auf die Distanz gar nicht wahrgenommen werden können.

Im *Probandenraum* stand der Simulationsrechner, auf dem die graphische Anwendung lief, die mit Gesten gesteuert werden sollte. Der Proband wurde mit drei Kameras beobachtet: eine Mimik- und Kopfgestenkamera (Gesicht von vorne), eine Handgestenkamera (Arbeitsfläche von oben) und eine Überblickskamera (Oberkörper von schräg hinten, Arbeitsfläche und Monitor mit Graphik-Anwendung von vorne). Die Bilder der Kameras wurden im *Beobachtungsraum* auf zwei Monitoren angezeigt (Kopf- und Überblickskamera waren auf einen Monitor zusammengemischt). Ebenso konnte der Proband über Monitore im Probandenraum seine Kamerabilder kontrollieren. Im Beobachtungsraum befand sich außer den Kontrollmonitoren noch der Steuerrechner. Simulations- und Steuerrechner

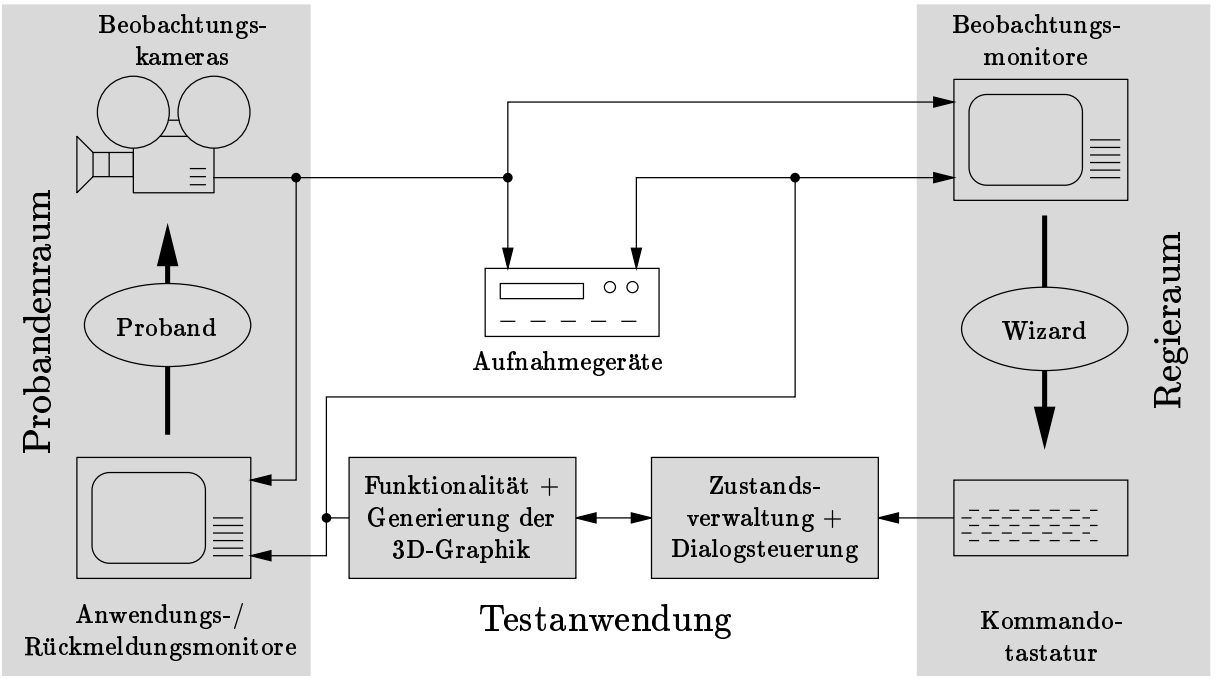


Bild 4.1: Schematische Struktur der Wizard-of-Oz-Versuche

waren miteinander vernetzt. Auf beiden Rechnern lief dieselbe Anwendung; über das Netzwerk waren beide Anwendungen synchronisiert.

Der Proband hatte die Aufgabe, die auf seinem Bildschirm sichtbare graphische Anwendung mit Gesten zu steuern, wobei er über die Kameras beobachtet wurde. Der Versuchsleiter verfolgte das Geschehen auf seinen Beobachtungsmonitoren, interpretierte die auftretenden Gesten und gab entsprechende Kommandos über die Tastatur ein. Der Wizard simulierte also an dieser Stelle einen maschinellen Erkenner. Die Anwendung enthielt die Komponenten der Zustandsverwaltung und der Dialogsteuerung, so daß der Wizard wirklich nur die Gesten erkennen und für jede Gestenkategorie ein entsprechendes Kommando eintippen mußte (s. Tabelle B.6 auf Seite 167). Das anwendungsabhängige Kontextwissen und die speziell an die gestische Modalität angepaßte Dialogstruktur waren also implementiert und liefen automatisch ab. Zwar wurde die Ausgabe der Anwendung auch dem Wizard angezeigt; diese Ausgabe wurde jedoch nicht direkt zur Anwendungssteuerung benötigt. Es hatte sich im Gegenteil als optimal herausgestellt, wenn der Versuchsleiter nur die Monitore beobachtete, die die Gesten des Probanden zeigten, um diese möglichst schnell zu interpretieren, ohne zu wissen, was der Proband eigentlich mit seinen Aktionen erreichen wollte: dies hätte nur zu kontextabhängigen Interpretationen geführt, die ein automatischer Erkenner nicht leisten kann.

Die Reaktion auf die eingetippten Kommandos beeinflußte die Darstellung auf dem Simulationsrechner, so daß es für den Probanden den Anschein hatte, daß das System seine Gesten erkannt hatte.

Wie die Pfeile in Bild 4.1 zeigen, reagieren sowohl der Proband als auch der Versuchsleiter auf visuelle Reize. Beim Probanden äußert sich die Reaktion in einer visuellen, beim Versuchsleiter jedoch in einer taktilen Reaktion. Der Kreislauf wird im fertigen System auf der Seite des Versuchsleiters durch eine automatische Erkennung geschlossen.

Für die nachträgliche Auswertung des Versuchs und zur Gewinnung von Datenmaterial für die automatische Erkennung wurden die Bilder der Beobachtungsmonitore auch auf Videoband aufgezeichnet. Außerdem wurden alle Aktionen des Wizards — die ja den beobachteten Gesten entsprechen — und die automatisch ablaufenden Vorgänge im Szenen-Editor in Protokoll-Files mit Zeitstempeln abgespeichert, so daß der Versuch synchron mit den Videoaufnahmen exakt rekonstruiert werden kann. Außerdem lassen sich auf diese Weise leicht Statistiken über die Verwendung der Gesten erstellen.

4.2 Ergebnisse der Evaluierung

Zur Evaluierung wurden zwei Versuchsreihen mit jeweils unterschiedlichen Zielsetzungen durchgeführt [Vol97]. Dabei diente der Auto-Einparksimulator nur der Eingewöhnung zu Beginn der ersten Versuchsreihe. Alle im folgenden beschriebenen Ergebnisse wurden mit dem 3D-Szenen-Editor gewonnen.

4.2.1 Versuchsreihe 1: „freie Gesten“

Die erste Versuchsreihe (VR 1) wird am besten durch das Stichwort „freie Gesten“ gekennzeichnet. Es sollten hier insbesondere Erkenntnisse über die Verwendung ungelernter Gesten gewonnen werden. Außerdem wurde VR 1 genutzt, um die gestische Benutzerschnittstelle des Szenen-Editors zu evaluieren und in Einzelheiten weiter zu verbessern.

An dieser Versuchsreihe waren 14 Versuchspersonen (VPs) beteiligt. Dabei wurden knapp 11 Stunden Videomaterial aufgezeichnet. Die erste Versuchsreihe war durch folgende Bedingungen gekennzeichnet:

- Die Beleuchtung und die Einrichtung des Probandenraumes vermittelte eine normale Büroatmosphäre.

- Es wurde vorher bekanntgegeben, daß ein menschlicher Versuchsleiter die Gesten interpretierte.

  Damit sollte sichergestellt werden, daß durch gewollte oder ungewollte Rücksichtnahme auf eine vermeintlich begrenzte maschinelle Erkennungsfähigkeit nicht etwa besonders einfache oder „gekünstelte“ Gesten verwendet würden. Streng genommen handelte es sich bei der ersten Versuchsreihe damit nicht um einen „echten“ Wizard-of-Oz-Versuch.

- Die Probanden hatten völlige Freiheit in der Wahl und der Ausführungsgeschwindigkeit ihrer Gesten. Es wurde mitgeteilt, daß als Gesten *sowohl Hand- als auch Kopfbewegungen* zulässig wären. Für die Handgesten konnten ein- und zweihändige Gesten verwendet werden.

- In einem Einführungsvideo wurde gezeigt, welche *Aktionen* im Szenen-Editor durchgeführt werden können. Es wurde aber nicht gezeigt, welche *Gesten* zum Auslösen dieser Aktionen verwendet werden sollten.

- Es bestand die Aufgabe, mit dem Editor eine auf einem Bild gezeigte Szene nachzubilden.

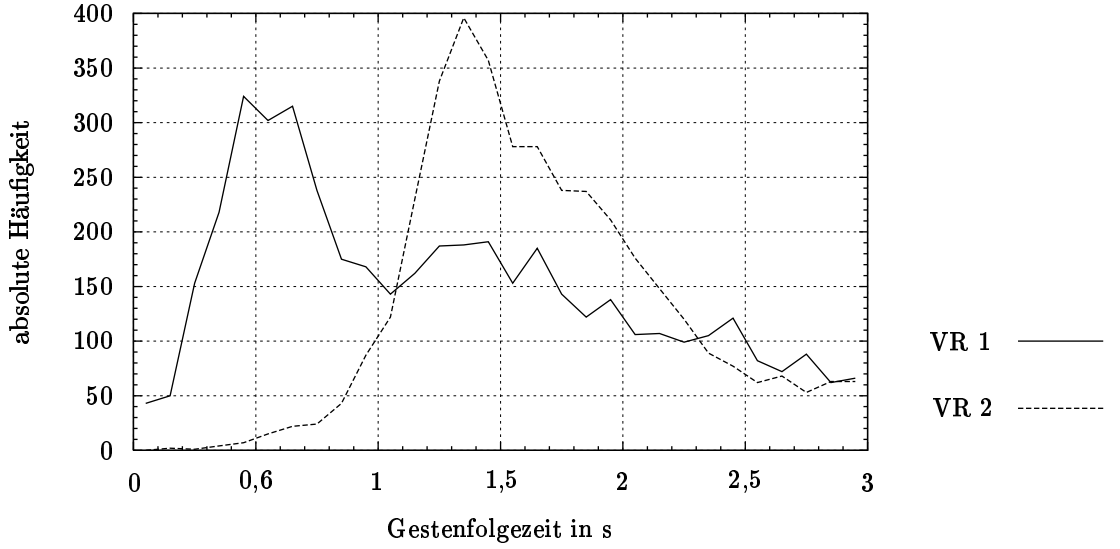


Bild 4.2: Absolute Häufigkeit der Gestenfolgezeiten für Versuchsreihe 1 und 2 (VR 1, 2)

Abschließend wurde über Fragebogen (s. Anh. B.1.1) die Meinung der Probanden zum Versuch und zum Dialogkonzept erfaßt. Die meisten Fragen erforderten eine Antwort auf einer quantitativen Skala von 1 (bester Wert, ja) bis 5 (schlechtester Wert, nein). Die genauen Ergebnisse der Befragung sind in Tabelle B.1 auf Seite 157 aufgeführt.

Obwohl den Probanden keine Gesten vorgeführt wurden, ließ sich jeweils mit deutlicher Tendenz belegen, daß der Szenen-Editor mit Alltagsgesten bedienbar ist (mittlerer Wert 2,2; Wert für Alltagsgesten 1) und daß der „Erfindungsprozeß“ der Gesten für die Anwendung intuitiv vonstatten ging (4,0 bei optimalem Wert von 5). Außerdem wurde angegeben, daß der Dialogpartner ansprechend war (2,0) und die Reaktion des Systems weitgehend der Absicht der Benutzer entsprach (2,5). Weitere Auswertungen werden im Vergleich mit den Auswertungen der zweiten Versuchsreihe in Kap. 4.2.2 vorgestellt.

Es ist wichtig festzustellen, daß alle Versuchspersonen auf Anhieb mit der Bedienung des Szenen-Editor zurechtkamen. Dabei waren sie sehr kreativ und „erfanden“ etwa 70 verschiedene Gesten (für eine komplette Auflistung siehe [Vol97]). Über eine Häufigkeitsanalyse konnte ein Katalog von 41 Gesten zusammengestellt werden (s. die systematische Tabelle B.4 mit einer verbalen Beschreibung und Tabelle B.5 in Anh. B.3.1 illustriert mit typischen Bildern der Gesten-Bildsequenzen). Im Gestenkatalog sind *keine Kopfgesten* enthalten, da die Probanden Handgesten sehr stark bevorzugten. Im Katalog gibt es nur noch drei beidhändige Gesten, da einhändige Gesten wesentlich häufiger verwendet wurden.

Ein Problem der VR 1 war, daß die Gesten der Benutzer relativ schnell aufeinander folgten (s. absolute Häufigkeiten in Bild 4.2). Die mittlere Gestenfolgezeit liegt zwar nur bei 1,32 s, allerdings gibt es ein ausgeprägtes Maximum im Intervall zwischen 0,4 und 0,5 s. Weitergehende statistische Analysen zeigen, daß es oft *gleiche* Gesten sind, die schnell hintereinander ausgeführt werden.

4.2.2 Versuchsreihe 2: „festgelegte Gesten“

Für eine maschinelle Erkennung gibt es einige Einschränkungen zu beachten. In der zweiten Versuchsreihe (VR 2) mit der Überschrift „festgelegte Gesten“ sollte untersucht

werden, wie die Benutzer mit diesen Einschränkungen im Vergleich mit VR 1 zurecht kommen und ob sich Unterschiede im Benutzerverhalten gegenüber einer menschlichen bzw. maschinellen Erkennung ergeben.

An VR 2 nahmen ebenfalls 14 Versuchspersonen teil, wobei 11 VPs mit denen aus der VR 1 übereinstimmten. Es wurden unter Zuhilfenahme der Erkenntnisse der ersten Versuchsreihe folgende Versuchsbedingungen hergestellt:

- Es wurde angekündigt, daß der Rechner nun mit dem Material aus der ersten Versuchsreihe darauf trainiert worden war, Gesten *automatisch* zu erkennen. Damit handelte es sich dieses Mal um einen „echten" Wizard-of-Oz-Versuch.

- Die Versuchsumgebung wurde mit Strahlern bildverarbeitungsgerecht ausgeleuchtet. Ebenso wurde unter dem Handbereich eine schwarze Hintergrundpappe montiert.

  Der Szenen-Editor wurde mit einer „Erkennungslampe" ausgestattet (s. Bild 3.3 rechts unter den Tasten). Ein grünes Leuchten signalisierte die Bereitschaft des Rechners, neue Gesten entgegenzunehmen. Ein rotes Leuchten zeigte an, daß der Rechner (vermeintlich) beschäftigt war. Damit sollte die hohe Wiederholfrequenz der Gesten, wie sie in der ersten Versuchsreihe auftrat, reduziert werden.

- Der Katalog von 41 Gesten, der sich aus der ersten Versuchsreihe ergab, wurde verbindlich vorgegeben. Somit waren für diesen Versuch keine Kopfgesten mehr zugelassen.

  Die Probanden wurden dazu aufgefordert, sich bei den einhändigen Gesten für eine Hand zu entscheiden und diese auch für die Dauer des Versuchs beizubehalten.

  Die Gesten aus dem Katalog wurden in einem Einführungsvideo vorgeführt. Es wurde ein Trainingslauf durchgeführt, in dem die Probanden die Gesten aus dem Katalog systematisch üben konnten. Es wurde betont, daß dies notwendig wäre, um die automatische Erkennung an den jeweiligen Benutzer anzupassen.

- Dieses Mal fanden die Probanden vorgefertigte Szenen auf dem Bildschirm vor, die gezielt verändert werden sollten.

Der Fragebogen zur VR 2 findet sich in Anh. B.2.1 auf. Die Ergebnisse sind in Tabelle B.2 auf Seite 160 zusammengetragen. Dabei ist es insbesondere interessant, welche quantitativen Veränderungen gegenüber VR 1 zu beobachten sind. Dazu sind in Tabelle B.3 auf Seite 160 die Ergebnisse der übereinstimmenden bzw. zusammenhängenden Fragen aufgeführt.

Die Auswertung der Fragebogen zeigt, daß die Vorgabe der Gesten *keine* Probleme bereitete (4,1). Die Versuchsleiter beobachteten auch kaum „verbotene" Gesten (auf sie wurde nicht reagiert). Die vorgegebenen Gesten wurden mit einem Wert von 3,8 als Alltagsgesten empfunden (was exakt dem Wert von VR 1 mit 2,2 bei entgegengesetzt angeordneter Skala entspricht).

Demgegenüber entsprach die Reaktion des Systems mit 1,9 deutlich mehr der Absicht des Benutzers als in der vorherigen Versuchsreihe (2,8). Interessanterweise wurde die Reaktion des System nun auch menschlicher eingestuft (2,1) als in VR 1 (2,5). Dies lag daran, daß die vorgegebenen Gesten vom Wizard teilweise wesentlich einfacher als die freien Gesten zu interpretieren waren, was Mißverständnisse in der Kommunikation reduzierte.

Es ergab sich eine deutlich ruhigere Gestenfolge: in der ersten Versuchsreihe lag die durchschnittliche Gestenfolgezeit bei 1,32 s, in der zweiten Versuchsreihe bei 1,73 s (der Maximalwert der Abfolgehäufigkeit verschob sich sogar von 0,5 s auf 1,4 s; s. Bild 4.2). Weiterführende Analysen in [Vol97] zeigen, daß sich in VR 2 die Gestenfolgezeiten von gleichen und unterschiedlichen Gesten einander annäherten. Die Reaktionszeit des Systems wurde mit 2,6 als ausgeglichen eingestuft (optimaler Wert bei 3), wobei sie mit 2,3 bei VR 1 als etwas zu langsam angegeben wurde. Offenbar nahmen die Probanden die vorgeschriebene ruhigere Gestenfolge an, denn sie fühlten sich durch die Kontrollampe nicht unter Druck gesetzt (4,5).

Die während der zweiten Versuchsreihe gewonnen Bilddaten (s. Anh. C.1) bilden die Basis für die Evaluierungen der automatischen Erkennung. Die Einschränkungen der zweiten Versuchsreihe waren dabei für eine praktikable maschinelle Erkennung unbedingt notwendig.

Als Ergebnis der Usability-Versuche läßt sich festhalten, daß das vorgestellte Konzept für die gestische Interaktion als intuitiv empfunden wurde und daß es für die Benutzer möglich war, das System mit allgemeinverständlichen Alltagsgesten zu bedienen, die keiner vorherigen Erklärung bedürfen. Auch die Einschränkungen, die von einer automatischen Erkennung diktiert werden, führten nicht zu einer Verschlechterung der Benutzbarkeit und Akzeptanz.

Nachdem nun das Konzept für eine visuelle Interaktion konzipiert und evaluiert wurde, steht aus VR 1 ein Gestenkatalog und aus VR 2 realistisches Trainings- und Testmaterial für eine automatische Erkennung zur Verfügung. In den folgenden Kapiteln werden die einzelnen Komponenten, die für diese automatische Erkennung notwendig sind, vorgestellt und evaluiert.

Kapitel 5

Räumliche Segmentierung und weitere Vorverarbeitungsschritte

Mit diesem Kapitel beginnt der algorithmische Teil dieser Arbeit. Wie im Systemüberblick in Bild 1.1 auf Seite 6 ersichtlich, ist als erster Vorverarbeitungsschritt die räumliche Segmentierung erforderlich, die direkt anschließend in Kap. 5.1 besprochen wird. Abhängig vom verwendeten Merkmalsextraktionsverfahren (s. Kap. 7) wird eventuell noch als letzter Vorverarbeitungsschritt ein Gradientenbild benötigt, was in Kap. 5.2 kurz skizziert wird.

Alle anderen Vorverarbeitungsschritte wie Bildskalierung, Farbraumwandlung, Kontrasteinstellung und Weißabgleich werden durch die Kamera- oder die Framegrabberhardware in Echtzeit geleistet und sind daher nicht Teil der hier beschriebenen Algorithmen.

5.1 Räumliche Segmentierung

5.1.1 Begriffsdefinitionen

Der räumlichen Segmentierung kommt im allgemeinen die Aufgabe zu, das Bild in Gebiete mit ähnlichen Eigenschaften aufzuteilen [Har91, Pra91], die sich gegenseitig nicht überlappen dürfen [Pal93]. Die Segmentierung wird in dieser Arbeit eingesetzt, um das für eine Klassifikation benötigte Objekt (oder den *Vordergrund* $\mathcal{V}$) vom Bild*hintergrund* zu trennen.

Bei der Gestikerkennung handelt es sich beim Vordergrund um die *Hände* des Benutzers. Auch wenn beide Hände im Bild sind, wird immer noch von *einem* Vordergrund geredet, obwohl damit unter Umständen zwei Bildobjekte gemeint sind. Zum Hintergrund zählen alle anderen Körperteile des Benutzers und die Gegenstände der Arbeitsumgebung, wie sie typischerweise bei der gewählten Kameraperspektive sichtbar sind. Dies sind also insbesondere die Arme und eventuell sichbare Teile des Oberkörpers, die Schreibtischoberfläche, Tastatur und Maus zur konventionellen Bedienung des Rechners, die Mausunterlage und weitere Gegenstände, die auf einem Schreibtisch zu finden sein könnten, wie z. B. Schreibstifte, Bücher und Papier (s. Beispiele in Bild 5.1).

Als Zwischenschritt der Segmentierung wird zu einer Bildfunktion $\mathbf{f}(\mathbf{n})$ eine *Segmentierungsmaske* $f_\mathrm{b}(\mathbf{n})$ erzeugt (s. Beispiel in Bild 5.2). Diese Maske hat für Werte des

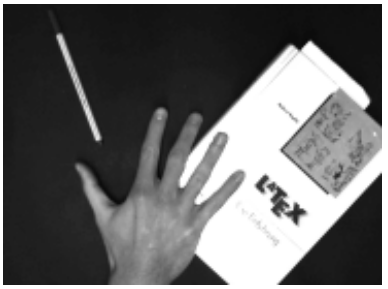


Bild 5.1: Beispiel für Originalbild

Bild 5.2: Segmentierungsmaske erzeugt aus Bild 5.1

Bild 5.3: Mit Maske 5.2 segmentiertes Bild 5.1

Vordergrundes $\mathcal{V}$ den Wert 1, sonst den Wert 0:

$$f_\mathrm{b}(\mathbf{n}) = \begin{cases} 1 & \text{für} \quad \mathbf{n} \in \mathcal{V} \\ 0 & \text{für} \quad \mathbf{n} \notin \mathcal{V} \end{cases} . \tag{5.1}$$

Das segmentierte Bild $\mathbf{f}_\mathrm{s}(\mathbf{n})$ entsteht dann durch Multiplikation der originalen Bildfunktion mit der Maske:

$$\mathbf{f}_\mathrm{s}(\mathbf{n}) = \mathbf{f}(\mathbf{n}) \cdot f_\mathrm{b}(\mathbf{n}) = \begin{cases} \mathbf{f}(\mathbf{n}) & \text{für} \quad \mathbf{n} \in \mathcal{V} \\ \mathbf{0} & \text{für} \quad \mathbf{n} \notin \mathcal{V} \end{cases} . \tag{5.2}$$

Es enthält also die Werte der ursprünglichen Bildfunktion für den Vordergrund und den Wert $\mathbf{0}$ für den Hintergrund (s. Beispiel in Bild 5.3). Abhängig von den nachfolgenden Verarbeitungsschritten wird direkt mit der Segmentierungsmaske $f_\mathrm{b}(\mathbf{n})$, mit dem segmentierten Bild $\mathbf{f}_{YUV,\mathrm{s}}(\mathbf{n})$ oder mit Teilkomponenten des segmentierten Bildes weitergearbeitet.

5.1.2 Evaluierungskriterien und Vorauswahl der Farbsegmentierung

Während sich ein Vordergrundbereich sprachlich oft sehr einfach beschreiben läßt, ist es schwierig, algorithmisch definierte, einfache Kriterien zu finden. Welche Kriterien sich für die Segmentierung eignen, ist immer von der speziellen Anwendung abhängig; allgemeingültige Verfahren, die *das* Segmentierungsproblem lösen, sind noch nicht gefunden worden [Pra91, Pal93].

Die Verfahrenswahl wird dadurch erleichtert, daß für die weiteren Vorverarbeitungsschritte (s. Kap. 5.2) und die Merkmalsextraktion (s. Kap. 7) *nicht* gefordert wird, daß der Vordergrund $\mathcal{V}$ ein einfach zusammenhängendes Gebiet im mathematischen Sinne (Definition beispielsweise in [Bro81]) sein muß. So sind bis zu einem gewissen Grad auch „Löcher“ im Vordergrund zugelassen. Unter Umständen bedeuten solche Segmentierungsfehler allerdings eine Verlängerung der benötigten Rechenzeit (beispielsweise bei der Berechnung von inneren Konturen für die Verfahren in Kap. 7.3.1.2).

Zur Auswahl eines geeigneten Verfahrens und zur Beurteilung der Güte der Verfahren werden die folgenden Kriterien S1–S5 herangezogen:

Echtzeitfähigkeit (S1): Die Segmentierung muß schritthaltend mit jedem Bild des Bildstroms durchgeführt werden können. Außerdem müssen zusätzlich noch weitere Vorverarbeitungsoperationen und die Merkmalsextraktion für jedes Bild ausgeführt werden. Die Segmentierung sollte also schneller als in Echtzeit erfolgen.

Damit werden von vornherein Verfahren ausgeschlossen, die *zur Laufzeit* komplexe arithmetische Berechnungen mit jedem einzelnen Bildpixel durchführen oder die es sogar erfordern, daß mehrere Pixel miteinander verrechnet werden[1]. Zu komplex sind unter diesem Gesichtspunkt unter anderem Verfahren, die Texturen auswerten, alle iterativen und morphologischen Verfahren sowie wissensbasierte Verfahren, die abstraktere Konzepte, wie beispielsweise die Form von Objekten, mit in die Auswertung integrieren (s. in [Har91, Pal93]).

Verfahren, die den optischen Fluß oder seine Betragsnäherung in Form von Pixeldifferenzen aufeinanderfolgender Bilder benutzen, sind ebenfalls zu rechenaufwendig, außer es werden wie beispielsweise in [Sch96b] und [Rig97] stark verkleinerte Bilder verwendet. Darüberhinaus sind diese Verfahren nicht robust genug, da sich auch Hintergrundobjekte bewegen können (s. Anmerkungen unter Kriterium S2).

Möglich sind dagegen Verfahren, die zur Laufzeit die Segmentierungsentscheidung für ein bestimmtes Pixel abhängig von seinem Inhalt lediglich „nachschlagen" müssen. Dies kann man erreichen, indem man den Inhalt eines Pixels als Adresse für einen Tabelleneintrag benutzt, in dem dann das Segmentierungsergebnis zu finden ist. Eine solche Tabelle wird allgemein üblich *Look-up-Tabelle* (LUT) genannt.

Damit bleibt als Grundlage der Segmentierung lediglich der Inhalt eines isolierten Pixels in Form eines Grau- oder Farbwertes übrig.

Hintergrundunabhängigkeit (S2): Wie in Kap. 5.1.1 beschrieben, zählen neben der Bekleidung des Benutzers und der Schreibtischoberfläche alle möglichen Gegenstände, die sich an einem Arbeitsplatz befinden können, zum Hintergrund. Die Segmentierung der Hände sollte an einem beliebigen Arbeitsplatz und unabhängig von zufällig im Aufnahmefeld der Kamera liegenden Gegenständen zuverlässig funktionieren. Insbesondere muß es möglich sein, daß sich der Hintergrund dynamisch ändert, indem beispielsweise Gegenstände bewegt werden oder neue Gegenstände in den Blickwinkel der Kamera geraten.

Bei der möglichen Komplexität der Hintergründe und der durch das Kriterium S1 vorgegebenen Beschränkung auf den isolierten Pixelinhalt ist klar, daß der *Grauwert* als Segmentierungskriterium nicht ausreichen kann. Dies haben die in Rahmen dieser Arbeit in [Bru95] durchgeführten ausführlichen Untersuchungen auch bestätigt. Es bleibt als Segmentierungskriterium also die Auswertung der *Farbinformation* übrig.

Um systematische Aussagen über die Unabhängigkeit der Verfahren vom Hintergrund treffen zu können, werden sie im folgenden mit fünf einfarbigen Hintergründen (schwarz, blau, grün, grau, rot) und einem schwarzen Hintergrund mit aufliegender Computereingabetastatur (Tastatur) getestet.

Benutzerunabhängigkeit (S3): Soweit der Benutzer mit seiner Kleidung zum Hintergrund zu rechnen ist, gilt dasselbe wie für die Unabhängigkeit vom Hintergrund unter Punkt S2.

Die Eigenschaften der Hände als Vordergrundobjekte können natürlich bei verschiedenen Benutzern stark variieren, wenn insbesondere berücksichtigt wird, daß

[1] *Nach* der Segmentierung sind Berechnungsschritte, die mehrere Pixel kombinieren, durchaus denkbar, weil dann große Bildbereiche als Hintergrund deklariert sind. Diese Bereiche können von weiteren Berechnungen durch einen einfachen logischen Vergleich ausgeschlossen werden.

Menschen verschiedener Hautfarbe als Benutzer in Frage kommen. Da somit eine Unabhängigkeit vom Benutzer mit schnellen pixelbasierten LUT-Verfahren kaum erreichbar sein wird, wird hier nur gefordert, daß die Segmentierung schnell (annähernd in Echtzeit) an einen neuen Benutzer *anpaßbar* ist.

Diese Anpassung könnte dann auch mit einfachen Verfahren unbemerkt automatisch vonstatten gehen, wenn postuliert wird, daß die Hand eines Benutzers erst den Bildbereich verlassen muß, bevor die Hand eines neuen Benutzers auftaucht.

Zusätzlich zur Echtzeitforderung im Betrieb aus Kriterium S1 ergibt sich somit eine Echtzeitforderung für das Training des Segmentierungsalgorithmus.

Unabhängigkeit von Beleuchtung und Kameraeinstellungen (S4): Die Beleuchtung ist charakterisiert durch Farbtemperatur und Helligkeit. Da Beleuchtungsschwankungen gleichzeitig in Vorder- und Hintergrund auftreten, können sie nur sehr schwer detektiert werden, so daß entsprechende Adaptionsverfahren schwierig zu steuern sind.

Farbtemperaturschwankungen sind dabei das geringere Problem, da man in der Regel in einer Arbeitsplatzatmosphäre von einer konstanten künstlichen Beleuchtung ausgehen kann, die zusätzlich vorhandenes Tageslicht meist dominiert.

Helligkeitsschwankungen dagegen können auch insbesondere bei Abschattungen der Hände untereinander und durch Schattenwurf der Hände auf den Hintergrund entstehen. Trotz dieser Schatten müssen Vorder- und Hintergrund zuverlässig getrennt werden.

Als weiteres Kriterium für die Auswahl eines Segmentierungsverfahrens bleibt daher die Robustheit gegenüber Helligkeitsschwankungen. Diese werden bei konstanter Beleuchtungsstärke simuliert durch Bildaufnahmen bei verschiedenen Blendenöffnungen (Blendenreihe 1,4; 2,0; 2,8; 4,0; 5,6).

Möglichst geringer Segmentierungsfehler (S5): Alle Pixel, die zu den Händen gehören, sollten durch die Segmentierungsmaske $f_\mathrm{b}(\mathbf{n})$ als Vordergrund klassifiziert werden, alle anderen Pixel als Hindergrund. Dementsprechend wurden in dieser Arbeit zwei Arten von Güte- bzw. Fehlermaßen definiert, die beide für jedes zu untersuchende Bild eine *Referenzmaske* $f_\mathrm{b}^\mathrm{Ref}(\mathbf{n})$ voraussetzen, die manuell mit Hilfe eines Graphikprogrammes erzeugt werden muß. Das erste Maß ist der *Vordergrund-Segmentierungsgrad* g_S^Vg, der als Anzahl der korrekt klassifizierten Vordergrundpixel bezogen auf die Vordergrundfläche $|\mathcal{V}|$ (in Pixel) gebildet wird:

$$g_\mathrm{S}^\mathrm{Vg} = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{n}\in\mathcal{V}} \delta\left(f_\mathrm{b}(\mathbf{n}) - f_\mathrm{b}^\mathrm{Ref}(\mathbf{n})\right). \tag{5.3}$$

Das zweite Maß ist der *Hintergrund-Segmentierungsfehler* e_S^Hg als Verhältnis der falsch klassifizierten Hintergrundpixel ebenfalls bezogen auf die Vordergrundfläche:

$$e_\mathrm{S}^\mathrm{Hg} = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{n}\notin\mathcal{V}} \delta\left(f_\mathrm{b}(\mathbf{n}) - f_\mathrm{b}^\mathrm{Ref}(\mathbf{n})\right). \tag{5.4}$$

Der Gesamtsegmentierungsfehler e_S wird dann mit

$$e_\mathrm{S} = \left(1 - g_\mathrm{S}^\mathrm{Vg}\right) + e_\mathrm{S}^\mathrm{Hg} \tag{5.5}$$

gebildet. Alle Maße werden in % angegeben. Obwohl dieses Segmentierungsmaß nur ein pauschales Maß darstellt und damit beispielsweise nicht unterschieden wird, ob ein falsch segmentiertes Vordergrundpixel vom Rand oder vom Inneren eines Vordergrundgebietes stammt, ist es für die Zwecke dieser Arbeit ausreichend und wird zur Begutachtung aller untersuchten Segmentierungsverfahren verwendet. Es hat sich als heuristischer Erfahrungswert bewährt, daß die Segmentierung noch als gut bezeichnet werden kann, wenn der Segmentierungsfehler e_S nicht größer als 10 % wird [Bru95].

Zusammenfassend gilt es also, ein LUT-basiertes Segmentierungsverfahren zu konzipieren, das die Farbinformation der Pixel auswertet. Das Verfahren muß annähernd in Echtzeit trainierbar sein, damit leicht eine Adaption an verschiedene Benutzer und an sich eventuell verändernde Beleuchtungsbedingungen vorgenommen werden kann. Die verschiedenen Verfahren können mit dem Fehlermaß aus Gl. (5.5) objektiv bewertet werden. Der Fehler muß mit Variation der unter Punkt S2 angegebenen Hintergründe und der Variation der Blende nach Punkt S4 möglichst gering bleiben.

5.1.3 Grundlagen der Farbsegmentierungsverfahren

Im Zusammenhang mit bildverarbeitungsgestützter statischer Gestikerkennung sind farbbasierte Segmentierungsverfahren mit LUTs relativ weit verbreitet. Die existierenden Verfahren haben aber entweder den Nachteil, daß sie sehr aufwendig zu trainieren und damit nur bedingt adaptierbar sind (beispielsweise [Mag94] abgeleitet von [Sch93]) oder daß sie schnell trainierbar sind, dann aber Bildregionen in Hautfarbe nur sehr ungenau darstellen (z. B. [Que95]), so daß eventuell eine rechenaufwendige, morphologische Nachbearbeitung des Vordergrundes erforderlich wird (z. B. [Kje96]).

Das im Rahmen dieser Arbeit entwickelte Verfahren ist dagegen in Echtzeit adaptierbar und erfordert keine Nachbearbeitung. Bis das endgültige Verfahren in Kap. 5.1.6 dargestellt werden kann, müssen zunächst noch einige Zwischenstadien präsentiert und evaluiert werden, mit denen Notwendigkeit und Effizienz des neuen Verfahrens quantitativ nachgewiesen werden kann.

Da die Verfahren in Echtzeit arbeiten sollen, stehen als Ausgangsbasis der *RGB*- und der *YUV*-Farbraum zur Verfügung: nur diese Farbdarstellungen werden von der verwendeten Videohardware in Echtzeit geliefert [Sch94b]. Bei der *RGB*-Darstellung handelt es sich um die Farbkomponenten *Rot*, *Grün* und *Blau*. Die *YUV*-Darstellung besteht aus der *Luminanz* Y, die nur die Helligkeitsinformation beinhaltet, und den *Chrominanz*komponenten U und V, die somit die eigentliche Farbinformation tragen. Beide Farbdarstellungen können über eine lineare Beziehung ineinander umgerechnet werden [Pra91].

Die folgenden Verfahren werden daher stets im dreidimensionalen *RGB*- und im zweidimensionalen *UV*-Raum getestet, da auf die Y-Komponente der *YUV*-Darstellung bei der Farbsegmentierung verzichtet werden kann. Die Ergebnisse beruhen auf einer Testmenge von 350 Bildern für alle Hintergründe [Bru95].

5.1.4 Verwendung eindimensionaler Komponentenhistogramme

Grundlage für die Bildung von LUTs für die Segmentierung sind *Histogramme*, die über die Häufigkeit der Farbwerte der Pixel im Bild Buch führen [Gon87]. Im einfachsten

Hintergrund	R-Kanal			G-Kanal			B-Kanal		
	g_S^{Vg}	e_S^{Hg}	e_S	g_S^{Vg}	e_S^{Hg}	e_S	g_S^{Vg}	e_S^{Hg}	e_S
schwarz	96,44	2,40	5,96	81,57	0,46	18,89	52,80	0,64	47,84
blau	97,06	4,11	7,05	72,33	0,44	28,11	0,85	0,43	99,58
grün	97,82	3,09	5,27	0,93	1,64	100,71	22,91	0,46	77,55
grau	0,40	0,00	99,60	46,98	0,09	53,11	74,19	0,14	25,95
rot	0,08	0,03	99,95	85,24	2,78	17,54	54,67	1,90	47,23
Tastatur	0,00	0,00	100,00	0,00	0,14	100,14	0,01	0,18	100,17

Tabelle 5.1: Ergebnisse für RGB-Hintergrundsegmentierung mit einkanaliger LUT (VG-Segmentierungsgrad g_S^{Vg}, HG-Segmentierungsfehler e_S^{Hg} und Gesamtsegmentierungsfehler e_S in %)

Fall werden getrennte, eindimensionale Histogramme $N^{Hg}(x)$ über die Farb*komponenten* $f_x^{Hg}(\mathbf{n})$ eines Bildes, das nur den Hintergrund enthält, gebildet:

$$N^{Hg}(x) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \delta\left(x - f_x^{Hg}(n_1, n_2)\right). \tag{5.6}$$

$N^{Hg}(x)$ enthält dann die *absoluten* Häufigkeiten der Pixelwerte des jeweiligen Bildkanals x. Aus jedem dieser Histogramme kann auf einfache Art eine LUT $L_x^{Hg}(x)$ gebildet werden, indem alle Werte, die im Histogramm von mindestens einem Pixel belegt sind, als Hintergrundwerte definiert werden, alle anderen als Vordergrundwerte:

$$L_x^{Hg}(x) = \begin{cases} 1 & \text{für} \quad N^{Hg}(x) = 0 \\ 0 & \text{für} \quad N^{Hg}(x) > 0 \end{cases}. \tag{5.7}$$

Im laufenden Betrieb wird mit der so erzeugten LUT $L_x^{Hg}(x)$ die Segmentierungsmaske $f_b(\mathbf{n})$ einfach dadurch erzeugt, daß jedes Pixel als Adresse für die Auswahl eines Tabellenelementes benutzt wird:

$$f_b(\mathbf{n}) = L_x^{Hg}(f_x(\mathbf{n})). \tag{5.8}$$

Die Tabelle 5.1 zeigt nun die Segmentierungsergebnisse für die verschiedenen Hintergründe, die unter Punkt S2 in Kap. 5.1.2 festgelegt wurden, im RGB-Farbraum (HG: Hintergrund, VG: Vordergrund). Die LUT wurde jeweils für denselben Hintergrund trainiert, für den auch die Segmentierung durchgeführt wurde. Man erkennt, daß die verschiedenen Farbkanäle bei jeweils bestimmten Hintergründen ihre Stärken haben. Nur der Tastatur-Hintergrund wird von allen Verfahren gleichermaßen schlecht segmentiert. Für bestimmte Hintergründe (schwarz, blau, grün) funktioniert die R-LUT sehr gut, für alle anderen Hintergründe und für alle Hintergründe bei den G- und B-LUTs liegen die Ergebnisse jedoch sehr deutlich über der 10 % Gesamtfehlerrate.

Verwendet man die LUTs der U- und V-Kanäle der YUV-Farbraumdarstellung, so ergeben sich die Resultate, wie sie in Tabelle 5.2 wiedergegeben sind. Man erkennt, daß jeder der beiden Chrominanzkanäle deutlich besser für die Segmentierung geeignet ist, als die RGB-Kanäle. Der V-Kanal liefert bis auf den grauen Hintergrund einen gerade noch akzeptablen Hintergrund-Segmentierungsfehler. Der Vordergrund-Segmentierungsgrad ist bei allen Hintergründen bis auf den roten sogar sehr gut. Beim roten Hintergrund wird aber praktisch kein Vordergrund erkannt.

Hintergrund	U-Kanal			V-Kanal		
	g_S^{Vg}	e_S^{Hg}	e_S	g_S^{Vg}	e_S^{Hg}	e_S
schwarz	100,00	10,30	10,30	100,00	8,81	8,81
blau	99,05	3,48	4,43	100,00	9,47	9,47
grün	67,34	0,01	32,67	99,95	9,58	9,63
grau	99,18	33,94	34,76	100,00	21,36	21,36
rot	56,18	0,00	43,82	0,00	0,15	100,15
Tastatur	94,94	0,09	5,06	99,49	0,82	1,33

Tabelle 5.2: Ergebnisse für UV-Hintergrundsegmentierung mit einkanaliger LUT (VG-Segmentierungsgrad g_S^{Vg}, HG-Segmentierungsfehler e_S^{Hg} und Gesamtsegmentierungsfehler e_S in %)

Man erkennt, daß keines der einkanaligen Verfahren für eine ausreichend gute Segmentierung geeignet ist. Es zeichnet sich aber auch ab, daß eine Segmentierung mit UV-Komponenten prinzipiell besser arbeitet als mit RGB-Komponenten.

5.1.5 Verwendung mehrdimensionaler Verbundhistogramme

5.1.5.1 Direkte Berechnung der Hintergrund-LUTs

Es ist naheliegend, die Farbinformation direkt im mehrdimensionalen Farbraum auszuwerten. Dadurch, daß die Farbkanäle dann im Verbund ausgewertet werden, ist die Korrektheit der Klassifikation sichergestellt, wenn mindestens noch ein Kanal die Trennbarkeit gewährleistet. Dies führt auf ein mehrdimensionales Histogramm, das sich formal wie die eindimensionalen Histogramme in Gl. (5.6) schreiben läßt:

$$N^{Hg}(\mathbf{x}) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \delta\left(|\mathbf{x} - \mathbf{f}_{\mathbf{x}}^{Hg}(n_1, n_2)|\right). \tag{5.9}$$

Dieses Histogramm ist im RGB-Farbraum dreidimensional und im UV-Farbraum zweidimensional. Analog zu den Gln. (5.7) und (5.8) ergibt sich dann eine mehrdimensionale Look-up-Tabelle

$$L_{\mathbf{x}}^{Hg}(\mathbf{x}) = \begin{cases} 1 & \text{für} \quad N^{Hg}(\mathbf{x}) = 0 \\ 0 & \text{für} \quad N^{Hg}(\mathbf{x}) > 0 \end{cases} \tag{5.10}$$

und eine mehrdimensionale Segmentierungsvorschrift:

$$f_b(\mathbf{n}) = L_{\mathbf{x}}^{Hg}(\mathbf{f}_{\mathbf{x}}(\mathbf{n})). \tag{5.11}$$

Die Ergebnisse in Tabelle 5.3 zeigen, daß die Zusammenfassung der Farbkanäle und die Bildung von mehrdimensionalen Verbundhistogrammen den Vordergrund-Segmentierungsgrad in *allen* Farbräumen im Vergleich zur eindimensionalen Histogrammbildung drastisch verbessern. Im UV-Farbraum ergibt sich für jeden Hintergrund ein e_S^{Vg} von 100 %; im RGB-Farbraum ist der Segmentierungsgrad in jedem Fall größer als 97 %.

Der Hintergrund-Segmentierungsfehler e_S^{Hg} ist jedoch in beiden Farbräumen stark verbesserungswürdig, da er meist deutlich größer ist als 10 %. Lediglich beim roten Hintergrund und beim Hintergrund mit der Tastatur ergeben sich im UV-Farbraum tolerierbare Fehler von ca. 6 % bzw. 5 %.

Hintergrund	RGB-Farbraum			UV-Farbraum		
	g_S^{Vg}	e_S^{Hg}	e_S	g_S^{Vg}	e_S^{Hg}	e_S
schwarz	99,99	23,47	23,48	100,00	15,59	15,59
blau	99,99	36,60	36,61	100,00	31,46	31,46
grün	99,98	41,18	41,20	100,00	39,08	39,08
grau	99,93	39,54	39,61	100,00	86,01	86,01
rot	97,29	17,29	30,00	100,00	6,06	6,06
Tastatur	100,00	61,16	61,16	100,00	5,23	5,23

Tabelle 5.3: Ergebnisse für RGB- und UV-Hintergrundsegmentierung mit unbearbeiteter, mehrkanaliger LUT (VG-Segmentierungsgrad g_S^{Vg}, HG-Segmentierungsfehler e_S^{Hg} und Gesamtsegmentierungsfehler e_S in %)

5.1.5.2 Mögliche Maßnahmen zur Verringerung des Hintergrund-Segmentierungsfehlers

Der große Hintergrund-Segmentierungsfehler läßt sich dadurch erklären, daß durch das Training auf den Hintergrund gerade bei den einfarbigen Hintergründen nur ein sehr kleiner Bereich der LUT $L_{\mathbf{x}}^{HG}(\mathbf{x})$ mit dem Hintergrundwert 0 belegt ist. *Alle anderen* Einträge werden implizit dem Vordergrund mit dem Wert 1 zugerechnet. Bei nur minimalen Veränderungen der Lichtverhältnisse (z. B. beim langsamen Erwärmen von Leuchtstoffröhren, wechselndem Schattenwurf, Änderungen des Tageslichtes, das durch die Fenster dringt) kann die Farbcharakteristik des aktuellen Hintergrundes vom trainierten Hintergrund schon so stark abweichen, daß viele Pixelwerte des Hintergrundes dem viel größeren Vordergrundbereich zugeordnet werden. Der Problematik der zu eng begrenzten Hintergrundbereiche in der LUT kann man mit drei Strategien begegnen:

1. Die Hintergrundbereiche in der LUT können durch einfache geometrische Körper bzw. Flächen parametrisch modelliert werden. Dadurch werden Lücken geschlossen und die Bereiche können durch Ändern der Parameter beliebig skaliert werden. Der Rechenaufwand beim Training wird dadurch erheblich erhöht. Im laufenden Betrieb ändert sich aber nichts, da die modellierten Raumbereiche wiederum in einer LUT abgespeichert werden können. Über dieses Verfahren wird in [Sch93] berichtet.

2. In dem im Rahmen dieser Arbeit neu entwickelten sog. *Radiusverfahren* werden die einzelnen Punkte der Hintergrund-LUT durch Kugeln bzw. Kreise mit einstellbarem Radius aufgeweitet. Dadurch wird die prinzipielle Gestalt des Hintergrund-Bereiches beibehalten, während innere und äußere Randbereiche wachsen. Mit dieser Maßnahme wird die LUT tolerant gegenüber Farbschwankungen. Der Rechenaufwand erhöht sich durch dieses Verfahren nur beim Training geringfügig, im Betrieb ändert sich jedoch nichts.

3. Alternativ kann man auch die Auflösung der LUT-Koordinaten verringern, so daß sich mit der gröberen Rasterung auch die Hintergrundbereiche in der LUT aufweiten. Sowohl beim Training als auch im Betrieb müssen die LUT-Koordinaten der Pixel allerdings in das neue Koordinatensystem umgerechnet werden. Da dies für jedes Bildpixel erforderlich ist, erhöht sich der Rechenaufwand auch im Betrieb erheblich. Diese Verfahren scheidet also von vornherein aus.

In [Bru95] wurden alle drei Verfahren getestet. Zur parametrischen Beschreibung der LUTs nach Punkt 1 wurden Ellipsoide bzw. Ellipsen verwendet. Das 3. Verfahren funk-

Hintergrund	RGB-Farbraum				UV-Farbraum			
	$r_{\text{opt}}^{\text{Hg}}$	g_{S}^{Vg}	e_{S}^{Hg}	e_{S}	$r_{\text{opt}}^{\text{Hg}}$	g_{S}^{Vg}	e_{S}^{Hg}	e_{S}
schwarz	15	99,11	9,98	10,87	4	100,00	8,33	8,33
blau	18	99,16	8,83	9,67	11	100,00	7,85	7,85
grün	23	99,74	9,85	10,11	7	100,00	8,72	8,72
grau	21	99,01	7,95	8,94	13	99,53	7,94	8,41
rot	8	92,00	6,60	14,60	1	98,28	1,59	3,31
Tastatur	16	99,64	7,21	7,57	1	99,88	1,41	1,53

Tabelle 5.4: Ergebnisse für RGB- und UV-Hintergrundsegmentierung mit aufgeweiteter LUT nach dem Radiusverfahren (VG-Segmentierungsgrad g_{S}^{Vg}, HG-Segmentierungsfehler e_{S}^{Hg} und Gesamtsegmentierungsfehler e_{S} in % bei jeweils optimalem Aufweitungsradius $r_{\text{opt}}^{\text{Hg}}$)

tionierte im Vergleich dazu etwas besser: es kann aber aufgrund des Rechenbedarfs nicht verwendet werden. Das Radiusverfahren nach Punkt 2 zeigte die besten Segmentierungsergebnisse, wobei es gleichzeitig auch am schnellsten trainierbar ist. Es wird im folgenden näher besprochen.

5.1.5.3 Aufweitung der LUT mit dem Radiusverfahren

Um eine Aufweitung der LUT zu erreichen, werden um jeden Punkt der LUT Kugeln bzw. Kreise mit einem fest vorgegebenen Radius r^{Hg} gelegt. Betrachtet man einen Punkt $\mathbf{x}'$ in der Hintergrund-LUT, der nach Gl. (5.10) zum Hintergrund gerechnet wird, so wird dieser Punkt gemäß der impliziten Bedingung

$$|\mathbf{x} - \mathbf{x}'| \leq r^{\text{Hg}} \tag{5.12}$$

um alle Punkte $\mathbf{x}$ erweitert, die kleiner oder gleich einem Radius r^{Hg} vom Zentralpunkt $\mathbf{x}'$ entfernt sind. Für die neue, aufgeweitete LUT $\tilde{L}_{\mathbf{x}}^{\text{Hg}}(\mathbf{x})$ gilt also:

$$\tilde{L}_{\mathbf{x}}^{\text{Hg}}(\mathbf{x}) = \begin{cases} 1 & \text{wenn} \quad |\mathbf{x} - \mathbf{x}'| \leq r^{\text{Hg}} \quad \text{für alle} \quad \mathbf{x}' \quad \text{mit} \quad L_{\mathbf{x}}^{\text{Hg}}(\mathbf{x}') = 1 \\ 0 & \text{sonst} \end{cases} . \tag{5.13}$$

Der Vorteil des Aufweitungsverfahrens im Vergleich zur parametrischen Modellierung nach Verfahren 1 in Kap. 5.1.5.2 besteht darin, daß sich die Aufweitung automatisch um den tatsächlichen Verlauf des Hintergrundbereiches in der LUT schmiegt und daß somit auch kompliziertere Strukturen nicht durch vereinfachende parametrische Modellierungsformen zerstört werden können. Liegen die Hintergrundpunkte der Ausgangs-LUT dicht genug, so kann man wie bei den parametrischen Verfahren davon ausgehen, daß Lücken innerhalb des Hintergrundbereiches geschlossen werden. Der Grad der Aufweitung wird durch den Radius r^{Hg} bestimmt.

Tabelle 5.4 zeigt die Segmentierungsergebnisse für die beiden Farbräume YUV und UV mit jeweils optimal eingestellten Aufweitungsradien $r_{\text{opt}}^{\text{Hg}}$. Die optimalen Radien wurden durch Testreihen ermittelt: es sind die Ergebnisse für den minimalen Gesamtsegmentierungsfehler gezeigt. Man erkennt, daß sich die Ergebnisse im Vergleich zur unbehandelten LUT in Tabelle 5.3 drastisch verbessert haben. Wiederum erzielt man im UV- kleinere Segmentierungsfehler als im RGB-Farbraum: sie liegen nun unabhängig vom Hintergrund unter 10 %; für den Tastaturhintergrund werden sogar 1,5 % erreicht.

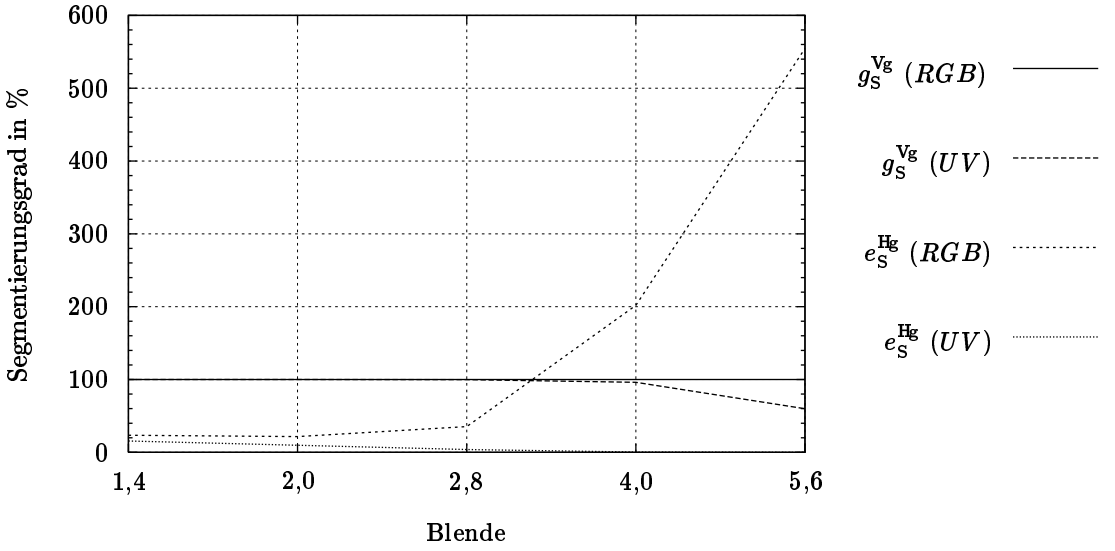


Bild 5.4: Blendenreihe für *RGB*- und *UV*-Hintergrundsegmentierung nach dem Radiusverfahren zur Begutachtung der Luminanzstabilität (VG-Segmentierungsgrad g_S^{Vg} und HG-Segmentierungsfehler e_S^{Hg} in % bei jeweils $r^{Hg} = 0$)

5.1.5.4 Evaluierung der Beleuchtungsunabhängigkeit

Daß der *UV*-Farbraum für die Segmentierung am geeignetsten ist, läßt sich auch anhand der Stabilität gegenüber Helligkeitsschwankungen nachweisen. Die Helligkeitsschwankungen wurden durch systematische Änderungen der Blendeneinstellung am Kameraobjektiv simuliert. Dazu wurde der Hintergrund zur Bildung der Hintergrund-LUT jeweils bei Blende 1,4 aufgenommen; die Segmentierungergebnisse wurden über die Blendenreihe aus Punkt S4 in Kap. 5.1.2 ermittelt. Bild 5.4 zeigt die Ergebnisse repräsentativ für den schwarzen Hintergrund ohne eine Aufweitung ($r^{Hg} = 0$). Bei den anderen Hintergründen ergibt sich eine ähnliche Tendenz. Man erkennt deutlich, daß zwar die Vordergrund-Segmentierungsrate im *RGB*-Farbraum immer über 99 % bleibt, gleichzeitig steigt aber der Hintergrund-Segmentierungsfehler mit kleiner werdender Blendenöffnung ab Blende 4,0 dramatisch an und erreicht bei Blende 5,6 über 550 %. Im *UV*-Farbraum fällt die Vordergrund-Segmentierungsrate bei Blende 4,0 leicht auf 96 % und dann auf rund 60 % bei Blende 5,6. Dafür bleibt der Hintergrund-Segmentierungsfehler immer deutlich unter dem im *RGB*-Farbraum um mit kleiner werdender Blendenöffnung schließlich auf 0 abzufallen.

Insgesamt erkennt man also, daß die Segmentierung im *UV*-Farbraum eindeutig ein wesentlich stabileres und ausgewogeneres Verhalten zeigt. Erhöht man den Radius r^{Hg}, so läßt sich der Gesamtfehler im *UV*-Farbraum noch deutlich reduzieren. Ein weiterer Vorteil des *UV*-Farbraumes ist der geringere Speicherbedarf der zweidimensionalen LUT im Vergleich zur dreidimensionalen LUT bei *RGB*-Komponenten. Außerdem kann auf die zweidimensionale LUT schneller zugegriffen werden.

Die Stabilität im *UV*-System ist der Tatsache zu verdanken, daß die *U*- und *V*-Anteile per Definition luminanznormiert sind und daß die gesamte Luminanz im *Y*-Anteil enthalten ist, der nicht verwendet wird. Prinzipiell ließen sich auch die *RGB*-Koordinaten in ein luminanznormiertes System umrechnen [Pra91], jedoch wird diese Umrechnung nicht von der Hardware unterstützt, so daß damit die Echtzeitbedingung verletzt würde.

5.1.6 Vordergrundsegmentierung

Da sich der Hintergrund im laufenden Betrieb ständig ändern kann (beispielsweise können neue Hintergrundgegenstände sichtbar werden oder Gegenstände werden bewegt und dadurch anders beleuchtet) ist es sinnvoll, auf eine *Vordergrund*-Segmentierung mit einer Vordergrund-LUT $L_{\mathbf{x}}^{\mathrm{Vg}}(\mathbf{x})$ überzugehen. Diese Vordergrund-LUT soll bei Bedarf (beispielsweise bei sich dramatisch ändernden Beleuchtungsverhältnissen, bei Wechsel des Benutzers) jederzeit neu und schnell trainiert werden können. Dies ist mit einer zweistufigen Vorgehensweise und dem Radiusverfahren im UV-Farbraum leicht möglich. Dazu wird, wie oben beschrieben, die Hintergrund-LUT $\tilde{L}_{\mathbf{x}}^{\mathrm{Hg}}(\mathbf{x})$ durch Aufnahme eines Hintergrund-Bildes und Aufweitung des Hintergrund-Bereiches nach Gl. (5.13) bestimmt. Dann hält der Benutzer eine Hand (ohne Ärmel) in das Bild, so daß eine LUT $L_{\mathbf{x}}^{\mathrm{Hg+Vg}}(\mathbf{x})$ aus Vorder- und Hintergrund zusammen berechnet werden kann. Diese LUT wird *nicht* aufgeweitet. Dann wird die Vordergrund LUT $L_{\mathbf{x}}^{\mathrm{Vg}}(\mathbf{x})$ durch „Subtrahieren" der Hintergrund-LUT von der Gesamt-LUT bestimmt:

$$L_{\mathbf{x}}^{\mathrm{Vg}}(\mathbf{x}) = \begin{cases} 1 & \text{wenn} \quad L_{\mathbf{x}}^{\mathrm{Hg+Vg}}(\mathbf{x}) = 1 \quad \text{und} \quad \tilde{L}_{\mathbf{x}}^{\mathrm{Hg}}(\mathbf{x}) = 0 \\ 0 & \text{sonst} \end{cases} . \tag{5.14}$$

Diese LUT kann nun analog zu Gl. (5.13) mit einem Vordergrundradius r^{Vg} zu $\tilde{L}_{\mathbf{x}}^{\mathrm{Vg}}(\mathbf{x})$ aufgeweitet werden.

In der Anwendungsphase wird dann nur noch die Vordergrund-LUT $\tilde{L}_{\mathbf{x}}^{\mathrm{Vg}}(\mathbf{x})$ benutzt. Kommen jetzt neue Gegenstände in den Bildausschnitt oder erscheint ein schon vorhandener Gegenstand aufgrund einer Lageänderung mit veränderten Farben, so werden diese Gegenstände so lange korrekt zum Hintergrund gerechnet, wie ihre Farbwerte nicht in den Bereich der Vordergrund-LUT fallen.

Tabelle 5.5 zeigt die Ergebnisse, die sich mit einer solchen Vordergrundsegmentierung bei verschiedenen Kreisradien $r^{\mathrm{Hg}} = r^{\mathrm{Vg}}$ im UV-Farbraum erreichen lassen. Zum Trainieren der LUTs wurde der Tastatur-Hintergrund gewählt. Bei unveränderter Vordergrund-LUT wurden dann die verschiedenen Hintergründe variiert. Man erkennt, daß sich mit steigendem Radius zwar die Vordergrund-Segmentierungsrate verbessert, gleichzeitig wird aber der Hintergrund-Segmentierungsfehler größer. Sieht man vom roten Hintergrund ab, so arbeitet das Verfahren bei beiden Radien akzeptabel innerhalb der vorgegebenen Toleranzschwelle von ca. 10 %. Durch Optimieren der beiden Kreisradien lassen sich die Ergebnisse noch weiter verbessern; dies kann leicht interaktiv in Abhängigkeit von den Lichtverhältnissen geschehen.

Es war absehbar, daß die Hand vor dem roten Hintergrund am schwierigsten zu segmentieren ist, da die Farbe Rot der Hautfarbe am ähnlichsten ist. Der rote Hintergrund kann nur durch sehr genau abgestimmte LUTs ausgeblendet werden, wie dies in Tabelle 5.4 demonstriert wurde. Das Verfahren liefert also im allgemeinen eine sehr gute Hintergrundstabilität. Erscheint nach dem Training der LUTs eine Farbe im Hintergrund, die der Hautfarbe sehr ähnlich ist, so kann durch eine erneute Abstimmung auf diesen Hintergrund wiederum eine sehr gute Segmentierung erreicht werden.

Die Anpassung an einen neuen Benutzer ist leicht automatisierbar, wenn postuliert wird, daß der alte Benutzer zuerst seine Hand aus dem Bildausschnitt entfernen muß, bevor der neue Benutzer seine Hand unter die Kamera hält. Über die Bestimmung des segmentierten Flächenäquivalentes unter Zuhilfenahme des Momentes nullter Ordnung (vgl. Kap. 7.3.1.2) und der Definition einer Schwelle läßt sich das Training der Hintergrund- und der Vordergrund-LUT im geeigneten Moment anstoßen.

Hintergrund	$r^{\mathrm{Hg}} = r^{\mathrm{Vg}} = 1$			$r^{\mathrm{Hg}} = r^{\mathrm{Vg}} = 8$		
	$g_{\mathrm{S}}^{\mathrm{Hg}}$	$e_{\mathrm{S}}^{\mathrm{Vg}}$	e_{S}	$g_{\mathrm{S}}^{\mathrm{Hg}}$	$e_{\mathrm{S}}^{\mathrm{Vg}}$	e_{S}
schwarz	93,31	0,00	6,69	99,59	0,00	0.41
blau	93,33	0,00	6,67	97,13	0,00	2,87
grün	96,12	1,35	5,23	99,19	4,25	5,06
grau	99,19	1,22	2,03	100,00	10,54	10,54
rot	94,79	11,51	16,72	96,54	29,14	32,60
Tastatur	99,89	1,25	1,36	99,98	7,42	7,44

Tabelle 5.5: Vordergrundsegmentierung mit dem Radiusverfahren mit verschiedenen Aufweitungsradien $r^{\mathrm{Hg}} = r^{\mathrm{Vg}}$ (UV-Farbraum, Hintergrund-LUT trainiert auf Tastatur, VG-Segmentierungsgrad $g_{\mathrm{S}}^{\mathrm{Vg}}$ und HG-Segmentierungsfehler $e_{\mathrm{S}}^{\mathrm{Hg}}$)

5.2 Gradientenbildberechnung

Nach Segmentierung und Vorverarbeitung folgt in der Verarbeitungskette die Merkmalsextraktion (s. Systemüberblick im Bild 1.1). Viele der Merkmalsextraktionsverfahren können direkt mit der Segmentierungsmaske $f_{\mathrm{b}}(\mathbf{n})$ oder der Y-Komponente des segmentierten Bildes $f_{Y,\mathrm{s}}(\mathbf{n}) =: f_{\mathrm{s}}(\mathbf{n})$ arbeiten, so daß keine weiteren Vorverarbeitungsschritte erforderlich sind. Einige Verfahren benötigen allerdings das *Gradientenbild*

$$\mathbf{f}_{\mathrm{g}}(\mathbf{n}) = \begin{bmatrix} f_{\mathrm{g}}(\mathbf{n}) \\ \delta_{\mathrm{g}}(\mathbf{n}) \end{bmatrix} = \begin{bmatrix} |\mathbf{f}_{\mathrm{g}}(\mathbf{n})| \\ \angle \mathbf{f}_{\mathrm{g}}(\mathbf{n}) \end{bmatrix}, \tag{5.15}$$

das in dieser Arbeit immer auf dem segmentierten Bild berechnet wird, um nur das Vordergrundobjekt zu erfassen. Das Gradientenbild besteht aus zwei Komponenten: dem *Kantenbild* $f_{\mathrm{g}}(\mathbf{n})$ — dem Betrag des Gradientenbildes — und dem *Orientierungsbild* $\delta_{\mathrm{g}}(\mathbf{n})$ [Gon87].

Das Kantenbild wird direkt nach der diskreten Näherung des mathematischen Gradientenoperators berechnet [Gon87, Bro81], da dieser für die wenig verrauschten und damit unproblematischen Bildfunktionen dieser Arbeit ausreicht:

$$f_{\mathrm{g}}(n_1, n_2) = K_1 \cdot \sqrt{[f_{\mathrm{s}}(n_1, n_2) - f_{\mathrm{s}}(n_1 + 1, n_2)]^2 - [f_{\mathrm{s}}(n_1, n_2) - f_{\mathrm{s}}(n_1, n_2 + 1)]^2}. \tag{5.16}$$

Das Orientierungsbild ist entsprechend als

$$\delta_{\mathrm{g}}(n_1, n_2) = K_2 + K_3 \cdot \tan^{-1} \frac{f_{\mathrm{s}}(n_1, n_2) - f_{\mathrm{s}}(n_1, n_2 + 1)}{f_{\mathrm{s}}(n_1, n_2) - f_{\mathrm{s}}(n_1 + 1, n_2)} \tag{5.17}$$

definiert. Durch geeignete Wahl der Konstanten K_1, K_2 und K_3 kann der mit 8 Bit dargestellte Pixel-Zahlenbereich optimal ausgenutzt werden.

Falls Merkmalsextraktionsverfahren sowohl mit dem segmentierten Grauwertbild als auch mit dem darauf aufbauenden Kantenbild verwendet werden können, so wird meist nur vom Grauwertbild gesprochen und als Formelzeichen vereinfachend $f(\mathbf{n})$ verwendet. Das Orientierungsbild wird stets gesondert erwähnt.

Kapitel 6

Hidden-Markov-Modelle zur stochastischen Modellierung isolierter Bildsequenzen

6.1 Möglichkeiten zur Bildsequenzmodellierung

Bildsequenzen sind dreidimensionale Gebilde: die einzelnen Teilbilder einer Sequenz bestehen aus zwei räumlichen Dimensionen; die Sequenz entsteht durch eine Aneinanderreihung der Teilbilder in einer zeitlichen Dimension (s. Bild 6.1). Da die Modellierung dieser gesamten Struktur zu aufwendig ist, muß eine Informationsreduktion vorgenommen werden. Dazu bestehen prinzipiell zwei Möglichkeiten:

1. Schreibt man einer Bildebene in der Darstellung 6.1 eine gewisse „Dicke“ zu, so entsteht aus der Bildsequenz ein Volumen, in dem jedem Volumenelement oder *Voxel* [Fol90] als Inhalt der Wert der betroffenen Bildfunktion zugeordnet ist. Durch diese Interpretation ist die zeitliche Dimension in eine zusätzliche räumliche Information umgewandelt worden. Das Datenaufkommen kann beispielsweise verringert werden, indem dieses Volumen auf eine Ebene projiziert wird. Liegt diese Ebene parallel zur Bildebene, so hat man die Bildsequenz um die zeitliche Dimension reduziert.

2. Das entgegengesetzte Vorgehen besteht darin, die räumliche Information nach bestimmten Vorschriften in Merkmalswerte umzurechnen und diese Merkmale zu *serialisieren*, um so unter Berücksichtigung der Bildsequenz zu einer rein zeitlichen Darstellung zu gelangen. Dabei gehen Teile der räumlichen Information verloren.

In dieser Arbeit wird die zweite Vorgehensweise gewählt, da der dynamische Aspekt zur Erkennung von Gesten so bedeutsam ist (s. Kap. 2.2), daß die Zeitinformation auf jeden Fall erhalten werden muß. Der Verlust an räumlicher Information ist dagegen vertretbar, da sich aufeinanderfolgende Bilder einer Sequenz sehr ähneln — dies wird beispielsweise zur Komprimierung von Bildsequenzen nach dem MPEG-Verfahren[1] ausgenutzt [Sch94b].

Für die Modellierung einer zeitlichen Sequenz von Merkmalen sind die *Hidden-Markov-Modelle* ideal geeignet. Sie haben sich in der Disziplin der Spracherkennung im Vergleich

[1] MPEG steht für *Moving Pictures Experts Group*.

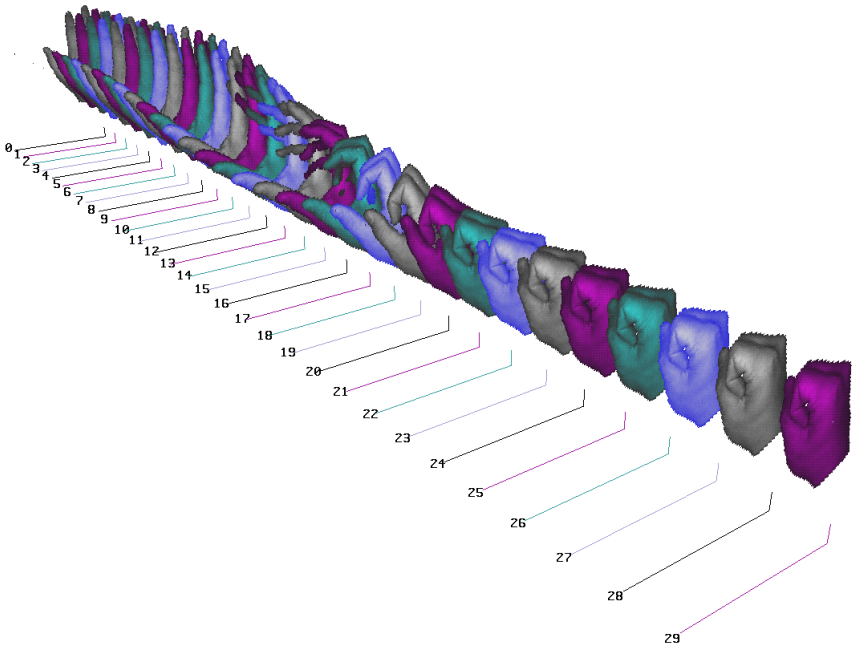


Bild 6.1: Bildsequenz als dreidimensionale räumlich-zeitliche Struktur

zur dynamischen Programmierung als besonders leistungsfähig herausgestellt [Rab89, Hua90].

Verfahren zur Extraktion zeitlich serieller Merkmale aus Bildsequenzen werden in Kap. 7 vorgestellt. Die Bewertung der möglichen Verfahren ist jedoch nur möglich, wenn als erstes die Grundlagen der Modellierung geklärt sind. Daher wird in Kap. 6.2 zunächst das für die Aufgabenstellung dieser Arbeit am besten geeignete Modell ausgewählt, das dann in Kap. 6.3 zusammen mit den grundlegenden Algorithmen näher spezifiziert wird. In Kap. 6.4 wird schließlich skizziert, wie numerischen Problemen bei der Modellierung begegnet werden kann.

Der in Kap. 6.3.5 vorgestellte Formalismus für die Erkennung behandelt zunächst den *isolierten* Fall: hierfür müssen die zeitlichen Grenzen der zu erkennenden Gesten bekannt sein. Die auf der isolierten Erkennung aufbauenden Algorithmen für die im praktischen Anwendungsfall notwendige *kontinuierliche* Erkennung werden in Kap. 9 behandelt.

6.2 Wahl des Modells

Die einfachste Art der Modellierung erfolgt mit diskreten HMMs, die zunächst einmal für die Verarbeitung von beliebigen diskreten Symbolsequenzen geeignet sind. Da man kontinuierliche Merkmale über eine Vektorquantisierung diskreten Symbolen zuordnen kann [Lin80], sind die diskreten Modelle prinzipiell auch für kontinuierliche Merkmale geeignet. Kontinuierliche Merkmale können jedoch auch direkt mit kontinuierlichen HMMs dargestellt werden, wobei diese Modellierung wesentlich leistungsfähiger als die Kombination aus Vektorquantisierung und diskreter Modellierung ist [Rab89].

Ein Problem der kontinuierlichen HMMs ist allerdings die große Zahl an freien Parametern, die nur mit sehr viel Trainingsmaterial geschätzt werden können. Da bei der Digitalisierung von Bildsequenzen sehr große Datenmengen anfallen, sind — beim jetzigen Stand der Technologie — dem möglichen Umfang der Trainingsdaten noch relativ

enge Grenzen gesetzt (vgl. Anh. C.1). Dieses prinzipielle Problem wird durch das Konzept der sog. *semikontinuierlichen* HMMs entschärft: bei ihnen ist die Anzahl der freien Parameter auf Kosten der erreichbaren Modellierungsgenauigkeit erheblich reduziert, wobei gleichzeitig immer noch direkt kontinuierliche Merkmale modelliert werden können [Hua90].

Man hat bei relativ wenig verfügbarem Trainingsmaterial also zwei Möglichkeiten: auf der einen Seite eine sehr genaue Modellierung verbunden mit einer schlechten Schätzbarkeit der einzelnen Parameter, auf der anderen Seite eine ungenauere Modellierung bei besserer Schätzbarkeit der Parameter. Es hat sich herausgestellt, daß die semikontinuierlichen HMMs, die die zweite Möglichkeit darstellen, hierbei bessere Ergebnisse liefern [Hua90]. Daher werden in dieser Arbeit ausschließlich semikontinuierliche HMMs verwendet. Zur weiteren Verringerung der Parameterzahl werden noch zusätzliche Vereinfachungen getroffen, die im folgenden dargestellt werden.

6.3 Grundlagen der semikontinuierlichen HMMs

6.3.1 Bestandteile eines HMMs und Definition der Parameter

In diesem Kapitel wird die Modellierung isolierter Gesten betrachtet. Zur Vereinfachung der Schreibweise wird ohne Beschränkung der Allgemeinheit festgelegt, daß eine Geste zum Zeitpunkt $t = 1$ beginnt und bis zum Zeitpunkt $t = T$ andauert. Die HMMs verarbeiteten also Sequenzen von Merkmalsvektoren der Form $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_T$. Bei der Zeit t handelt es sich um den *Merkmalszeittakt*, der im allgemeinen schneller ist als der Bildzeittakt (s. Kap. 7.1).

Beim *Training* sollen alle Sequenzen, die zu einer Geste gehören, auf ein Modell λ_l, $l = 1, \ldots, M$ abgebildet werden. M ist dabei die Größe des Vokabulars, d. h. die Anzahl der unterschiedlichen Gesten, die betrachtet werden. Der Trainingsalgorithmus wird in Kap. 6.3.4 beschrieben. Die Aufgabe der *isolierten Erkennung* ist es, einer unbekannten Merkmalssequenz mit bekannten Grenzen das Modell zuzuordnen, das die Sequenz am besten repräsentiert (s. Kap. 6.3.5).

Die Struktur eines Hidden-Markov-Modells ist in Bild 6.2 zu sehen. Ein Modell λ_l besteht aus N *Zuständen* $S_1, \ldots S_N$, die durch Knoten verdeutlicht werden. Die Anzahl der Zustände kann frei gewählt werden, ist dann aber für alle Modelle eines Gestenvokabulars gleich. Jeder dieser Zustände S_i enthält eine Wahrscheinlichkeitsdichtefunktion $f_{S_i}^{\lambda_l}(\mathbf{x})$ (der D-dimensionale Merkmalsvektor $\mathbf{x}$ wird zur Vereinfachung der Schreibweise ohne Zeitindex dargestellt). Die Wahrscheinlichkeitsdichtefunktionen (WDFs) beschreiben die statistischen Abhängigkeiten der Merkmale für die Zeit, während der das Modell sich in diesem Zustand befindet.

Zur Darstellung dieser WDFs benutzen alle Modelle ein gemeinsames Codebuch, dessen *Prototypen* $f_{v_k}(\mathbf{x})$, $k = 1, \ldots, L$ wiederum WDFs darstellen, die für eine „weiche" Vektorquantisierung genutzt werden [Hua90]. Aus diesem Codebuch werden nun mit *Mixturkoeffizienten* $c_{S_i v_k}^{\lambda_l}$ die modellspezifischen Zustands-WDFs gebildet:

$$f_{S_i}^{\lambda_l}(\mathbf{x}) = \sum_{k=1}^{L} c_{S_i v_k}^{\lambda_l} f_{v_k}(\mathbf{x}). \tag{6.1}$$

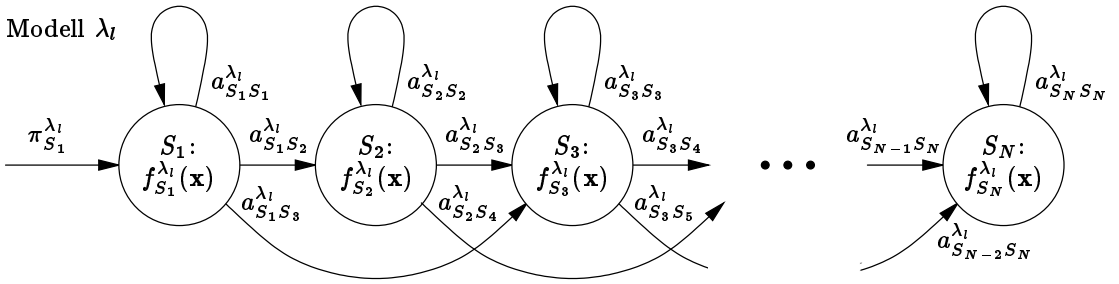


Bild 6.2: Struktur und Parameter eines Hidden-Markov-Modells λ_l

Als WDFs werden D-dimensionale Gaußverteilungen der Form

$$f_{v_k}(\mathbf{x}) = \frac{1}{\sqrt[D]{2\pi}\sqrt{|\mathbf{\Sigma}_{v_k}|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{v_k})^{\mathrm{T}}\mathbf{\Sigma}_{v_k}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{v_k})\right] \tag{6.2}$$

verwendet, die durch den Mittelwertsvektor $\boldsymbol{\mu}_{v_k}$ und die Kovarianzmatrix $\mathbf{\Sigma}_{v_k}$ vollständig bestimmt sind. Für die Modellbildung in dieser Arbeit werden schließlich nur die Diagonalelemente σ_{ii,v_k} der Kovarianzmatrix $\mathbf{\Sigma}_{v_k}$ verwertet und die Nicht-Diagonalelemente zu Null gesetzt, weil diese aufgrund des relativ geringen Umfangs der Trainingsdatenmenge nur schlecht geschätzt werden können. Von der Kovarianzmatrix bleibt also nur ein Kovarianzvektor $\boldsymbol{\sigma}_{v_k}$ der Diagonalelemente. So vereinfacht sich Gl. (6.2) zu [Bro81]:

$$f_{v_k}(\mathbf{x}) = \frac{1}{\sqrt[D]{2\pi}\sqrt{\prod_{i=1}^{D}\sigma_{ii,v_k}}} \exp\left[-\frac{1}{2}\sum_{i=1}^{D}\frac{1}{\sigma_{ii,v_k}}(x_i-\mu_{i,v_k})^2\right]. \tag{6.3}$$

Eine zusätzliche Beschränkung auf den maximalen Summanden der Zustands-WDF aus Gl. (6.1), wie sie beispielsweise in [Hua90] vorgeschlagen wird, hat sich dagegen nicht bewährt.

Die Zustände sind durch *Übergänge* miteinander verbunden, die mit Übergangswahrscheinlichkeiten $a_{S_iS_j}^{\lambda_l}$ gewichtet sind (s. Bild 6.2). Die Wahrscheinlichkeit $a_{S_iS_j}^{\lambda_l}$ beschreibt den Übergang von Zustand S_i zum Zustand S_j. Die Übergänge werden durch Kanten dargestellt. Zur Vereinfachung werden nur Übergänge von einem Zustand zurück zum selben Zustand, zum nächsten Zustand und zum übernächsten Zustand zugelassen. Eine Beaufschlagung des Modells mit einer Folge von Merkmalsvektoren $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ bewirkt, daß im Modell λ_l eine Folge von Zuständen $S = s_1, s_2, \ldots, s_T$ durchlaufen wird. Diese Zustandsfolge ist unbekannt und von außen nicht sichtbar, woher auch die Bezeichnung *Hidden*-Markov-Modell rührt. Die wahrscheinlichste Zustandsfolge kann aber nachträglich mit dem Viterbi-Algorithmus bestimmt werden (s. Kap. 6.3.3.2). Der Anfangszustand eines Modells wird durch die Einsprungwahrscheinlichkeiten $\pi_{S_i}^{\lambda_l}$ festgelegt, die sich zu einem Vektor $\boldsymbol{\pi}^{\lambda_l}$ zusammenfassen lassen.

Die Struktur eines Hidden-Markov-Modells wird implizit dadurch bestimmt, daß gewisse Einsprung- und Übergangswahrscheinlichkeiten permanent auf Null gesetzt werden. Bei der in Bild 6.2 dargestellten Struktur handelt es sich um ein Links-Rechts-Modell [Rab89], das erzwungenermaßen immer im 1. Zustand beginnt, von links nach rechts durchlaufen wird und im letzten Zustand endet. In dieser Arbeit werden ausschließlich solche Links-Rechts-Modelle verwendet.

Die verschiedenen Modellparameter lassen sich zu größeren Vektoren oder Matrizen zusammenfassen, so daß sich zusammenfassend sagen läßt, ein semikontinuierliches Hidden-Markov-Modell

$$\lambda_l = \lambda_l(\boldsymbol{\pi}^{\lambda_l}, \mathbf{A}^{\lambda_l}, \mathbf{C}^{\lambda_l}, \mathbf{M}, \boldsymbol{\Sigma}) \tag{6.4}$$

wird eindeutig bestimmt durch den Vektor der Einsprungwahrscheinlichkeiten $\boldsymbol{\pi}^{\lambda_l}$, der Matrix der Übergangswahrscheinlichkeiten $\mathbf{A}^{\lambda_l}$, der Matrix der Mixturkoeffizienten $\mathbf{C}^{\lambda_l}$, der Matrix der Mittelwertsvektoren der Prototypen $\mathbf{M}$ und der Matrix der Kovarianzvektoren der Prototypen $\boldsymbol{\Sigma}$ [Rab89, Hua90]. Die letzten beiden Parameter sind nicht modellspezifisch, sondern allen Modellen gemeinsam.

6.3.2 Bestimmung des Anfangsmodells

Vor dem eigentlichen Training der Modelle λ_l, müssen die allen Modellen gemeinsamen Parameter $\mathbf{M}$ und $\boldsymbol{\Sigma}$ des Codebuchs und die spezifischen Modellparameter $\boldsymbol{\pi}^{\lambda_l}$, $\mathbf{A}^{\lambda_l}$ und $\mathbf{C}^{\lambda_l}$ auf sinnvolle Anfangswerte gesetzt werden (s. Kap. 6.3.2.1 und 6.3.2.2). Die spätere Nachschätzung (s. Kap. 6.3.4) findet ausgehend von diesen Anfangswerten lediglich ein *lokales* Optimum, so daß das Ergebnis des Trainings stark von diesen Anfangswerten abhängt.

6.3.2.1 Initialisierung des Codebuchs

Zur Bildung des Codebuchs muß der Merkmalsraum in *Cluster* zerlegt werden; das sollten möglichst Raumbereiche sein, in denen sich die Merkmale ballen. Jedes Cluster wird dann durch einen *Prototypen* und damit nach Gl. (6.3) durch die Parameter $\boldsymbol{\mu}_{v_k}$ und $\boldsymbol{\sigma}_{v_k}$ repräsentiert. Es hat sich bewährt, die Lage der initialen Prototypen iterativ nach dem *k-means*-Algorithmus zu optimieren (s. beispielsweise [Rab89, Hua90]). Dabei werden immer abwechselnd der Schwerpunkt eines Clusters durch Mittelung der diesem Cluster zugeordneten Merkmalsvektoren gebildet und anschließend die Prototypen zu den neugebildeten Mittelpunkten hin verschoben:

1. **Initialisierung:** Die Mittelwertsvektoren $\boldsymbol{\mu}_{v_k}$ und der mittlere Quantisierungsfehler e^{vq} werden auf geeignete Anfangswerte gesetzt (näheres s. u.):

$$\boldsymbol{\mu}_{v_k} = \boldsymbol{\mu}_{v_k}^{\mathrm{init}} \quad \text{für} \quad k = 1, \ldots, L \quad \text{und} \tag{6.5}$$

$$e^{\mathrm{vq}} = \infty. \tag{6.6}$$

2. **Iterationsschritt:** Nach der Nächsten-Nachbar-Regel und unter Zuhilfenahme eines euklidischen Abstandsmaßes $\mathrm{d}[.,.]$ werden aus der Menge *aller* Trainingsvektoren $\mathbf{X}^{\mathrm{Tr}}$ Cluster C_{v_k} um die Mittelwertsvektoren $\boldsymbol{\mu}_{v_k}$ gebildet ($\mathrm{q}[\mathbf{x}|\mathbf{M}]$ bezeichnet dabei die Quantisierung eines Vektors $\mathbf{x}$, liefert also den nächstliegenden Mittelwertsvektor $\boldsymbol{\mu}_{v_k}$ zu $\mathbf{x}$). Für jedes dieser Cluster wird nun ein neuer Mittelpunktsvektor $\hat{\boldsymbol{\mu}}_{v_k}$ berechnet. $\hat{e}^{\mathrm{vq}}$ enthält dann den mittleren Quantisierungsfehler bezogen auf die aktualisierte Lage der Mittelpunktsvektoren:

$$C_{v_k} = \left\{\mathbf{x} | \mathbf{x} \in \mathbf{X}^{\mathrm{Tr}} \wedge \mathrm{q}[\mathbf{x}|\mathbf{M}] = \boldsymbol{\mu}_{v_k}\right\} \quad \text{und} \tag{6.7}$$

$$\hat{\boldsymbol{\mu}}_{v_k} = \frac{1}{|C_{v_k}|} \sum_{\mathbf{x} \in C_{v_k}} \mathbf{x} \quad \text{für} \quad k = 1, \ldots, L, \tag{6.8}$$

$$\hat{e}^{\mathrm{vq}} = \frac{1}{|\mathbf{X}^{\mathrm{Tr}}|} \sum_{\mathbf{x} \in \mathbf{X}^{\mathrm{Tr}}} \mathrm{d}\left[\mathbf{x}, \mathrm{q}\left[\mathbf{x}|\hat{\mathbf{M}}\right]\right]. \tag{6.9}$$

$|C_{v_k}|$ und $\left|\mathbf{X}^{\mathrm{Tr}}\right|$ sind die Mächtigkeiten der entsprechenden Mengen.

3. **Abbruchkriterium:** Wenn für den relativen Quantisierungsfehler gilt:

$$\frac{e^{\mathrm{vq}} - \hat{e}^{\mathrm{vq}}}{\hat{e}^{\mathrm{vq}}} < \epsilon^{\mathrm{vq}}, \tag{6.10}$$

dann weiter mit dem 4. Schritt, sonst zurück zum 2. Schritt.

4. **Resultat:** Am Ende ergeben sich die optimalen Mittelpunktsvektoren und Kovarianzvektoren; sie bilden das initiale Codebuch $\{\dot{\mathbf{M}}, \dot{\mathbf{\Sigma}}\}$:

$$\dot{\boldsymbol{\mu}}_{v_k} = \boldsymbol{\mu}_{v_k}^{\mathrm{opt}} = \hat{\boldsymbol{\mu}}_{v_k} \quad \text{und} \tag{6.11}$$

$$\dot{\boldsymbol{\sigma}}_{v_k} = \boldsymbol{\sigma}_{v_k}^{\mathrm{opt}} = \frac{1}{\left|\hat{C}_{v_k}\right| - 1} \sum_{\mathbf{x} \in \hat{C}_{v_k}} \left(\mathbf{x} - \boldsymbol{\mu}_{v_k}^{\mathrm{opt}}\right)^2 \quad \text{für} \quad k = 1, \ldots, L. \tag{6.12}$$

Da für den Iterationsschritt lediglich ein einfaches euklidisches Abstandsmaß verwendet wird, müssen während der Iteration mit Gl. (6.8) immer nur die jeweiligen Mittelwertsvektoren neu berechnet werden. Die für die Beschreibung der Prototypen noch benötigten Kovarianzvektoren werden nur einmal am Ende der Iteration bestimmt (s. Gl. (6.12)).

Da mit diesem Verfahren lediglich ein lokales Lageoptimum gefunden werden kann, ist die Initialisierung der Prototypen von entscheidender Bedeutung. Der in dieser Arbeit verwendete *splitting*- oder LBG-Algorithmus [Lin80] löst dieses Problem rekursiv in zwei sich abwechselnden Schritten. Beginnend mit einem einzelnen Prototyp, wird die Lage des oder der Protypen nach dem in den Gln. (6.5)–(6.12) beschriebenen k-means-Verfahren optimiert. In einem zweiten Schritt werden dann die Mittelwertsvektoren verdoppelt oder *gesplittet*, indem jeweils ein Vektor durch zwei Vektoren ersetzt wird, die um einen kleinen Abstand symmetrisch zum alten Vektor verschoben liegen. Diese Verschiebung erfolgt in dieser Arbeit in Richtung der Dimension, die jeweils die größte Komponente des Kovarianzvektors $\boldsymbol{\sigma}_{v_k}$ enthält [Win96]. Anschließend werden die gesplitteten Mittelwertsvektoren in ihrer Lage optimiert, wodurch die Vektoren sehr schnell auseinandergezogen werden. Ist die gewünschte Anzahl von Vektoren erreicht, wird das Verfahren abgebrochen. Man erkennt, daß die Anzahl der Prototypen, die nach dem LBG-Algorithmus erzeugt werden, immer eine *Zweierpotenz* sein muß, was aber keine Einschränkung darstellt.

Da der Clusterung ein euklidisches Abstandsmaß zugrunde liegt, die Prototypen nach Gl. (6.3) bei der Erkennung mit einem semikontinuierlichen HMM aber über ein kovarianzbasiertes Abstandsmaß verrechnet werden, passen Lage (d. h. Mittelwertsvektoren) und Kovarianzvektoren der Prototypen nach der Initialisierung nur näherungsweise zum HMM. Mit einem aufwendigeren Optimierungsverfahren läßt sich auch ein Codebuch über ein kovarianzbasiertes Abstandsmaß berechnen [Hua90]. Allerdings wird das Codebuch beim HMM-Training zusammen mit den anderen Modellparametern exakt nachgeschätzt, so daß für die Initialisierung die oben beschriebene Näherung ausreicht.

6.3.2.2 Initialisierung der Modellparameter

Die Einsprungwahrscheinlichkeiten sind durch die Vorgabe, daß ein Modell λ_l immer im ersten Zustand beginnen muß, festgelegt und müssen daher nicht nachtrainiert werden:

$$\dot{\boldsymbol{\pi}}^{\lambda_l} = \boldsymbol{\pi}^{\lambda_l} = [1, 0, 0, 0, \ldots, 0]^{\mathrm{T}}. \tag{6.13}$$

Die Selbstübergangswahrscheinlichkeit $a_{S_i S_i}^{\lambda_l}$ und die mittlere Verweildauer $\bar{\tau}_{S_i}^{\lambda_l}$ in einem Zustand S_i hängen über die Beziehung

$$\bar{\tau}_{S_i}^{\lambda_l} = \frac{1}{1 - a_{S_i S_i}^{\lambda_l}} \quad \text{für} \quad i = 1, \ldots, N-1 \tag{6.14}$$

zusammen [Rab89]. Möchte man daher Modelle mit N Zuständen mit einer jeweils gleichen mittleren Verweildauer initialisieren und haben die Trainingssequenzen für ein Modell λ_l eine mittlere zeitliche Länge von $\bar{T}^{\lambda_l}$, so ergibt sich als Initialisierungsbedingung:

$$\frac{\bar{T}^{\lambda_l}}{N} \stackrel{!}{=} \bar{\tau}_{S_i}^{\lambda_l} = \frac{1}{1 - a_{S_i S_i}^{\lambda_l}} \Longleftrightarrow \dot{a}_{S_i S_i}^{\lambda_l} = 1 - \frac{N}{\bar{T}^{\lambda_l}}. \tag{6.15}$$

Mit der Normierungsbedingung, daß die Summe aller Übergangswahrscheinlichkeiten, die von einem Zustand ausgehen, 1 ergeben muß, wird die Restwahrscheinlichkeit $1 - \dot{a}_{S_i S_i}^{\lambda_l}$ auf die Übergänge $\dot{a}_{S_i S_{i+1}}^{\lambda_l}$ und $\dot{a}_{S_i S_{i+2}}^{\lambda_l}$ aufgeteilt (in den letzten beiden Zuständen des Modells kommen entsprechend weniger Folgezustände in Frage). Die Aufteilung wird nach der empirisch gefundenen Bedingung $\dot{a}_{S_i S_{i+1}}^{\lambda_l} = 599 \cdot \dot{a}_{S_i S_{i+2}}^{\lambda_l}$ gestaltet, so daß das Überspringen eines Zustandes zwar erschwert wird, aber nicht unmöglich ist. Alle anderen Übergänge werden auf 0 gesetzt. Dadurch können sie auch beim Nachtraining keinen von Null verschiedenen Wert mehr annehmen, so daß die Struktur des Modells wie in Bild 6.2 festliegt.

Zur Initialisierung der Mixturkoeffizienten werden die Trainingssequenzen gleichmäßig auf alle Zustände verteilt. Es läßt sich daher die Menge $\mathbf{X}_{S_i}^{\mathrm{Tr},\lambda_l}$ aller Merkmalsvektoren bilden, die zu einem bestimmten Zustand S_i eines Modells λ_l gehören. Die jeweilige Anzahl der Merkmalsvektoren, auf die dies zutrifft, sei $N_{S_i}^{\mathrm{Tr},\lambda_l} = |\mathbf{X}_{S_i}^{\mathrm{Tr},\lambda_l}|$. Damit ergeben sich die Mixturkoeffizienten durch Aufsummieren der initialen Rückschlußwahrscheinlichkeiten $\dot{f}(v_k|\mathbf{x})$ zu (abgeleitet aus [Hua90]):

$$\dot{c}_{S_i v_k}^{\lambda_l} = \frac{1}{N_{S_i}^{\mathrm{Tr},\lambda_l}} \sum_{\mathbf{x} \in \mathbf{X}_{S_i}^{\mathrm{Tr},\lambda_l}} \dot{f}(v_k|\mathbf{x}) = \frac{1}{N_{S_i}^{\mathrm{Tr},\lambda_l}} \sum_{\mathbf{x} \in \mathbf{X}_{S_i}^{\mathrm{Tr},\lambda_l}} \frac{\dot{f}_{v_k}(\mathbf{x})}{\dot{f}(\mathbf{x})} = \frac{1}{N_{S_i}^{\mathrm{Tr},\lambda_l}} \sum_{\mathbf{x} \in \mathbf{X}_{S_i}^{\mathrm{Tr},\lambda_l}} \frac{\dot{f}_{v_k}(\mathbf{x})}{\sum_{k=1}^{L} \dot{f}_{v_k}(\mathbf{x})}. \tag{6.16}$$

Die initialen Prototypen $\dot{f}_{v_k}(\mathbf{x})$ sind durch die Anfangswerte der Codebuchvektoren mit den Gln. (6.11) und (6.12) festgelegt. Damit ist die Initialisierung der Modellparameter abgeschlossen.

6.3.3 Der Viterbi-Algorithmus

6.3.3.1 Gründe für die Verwendung des Viterbi-Algorithmus

Der Viterbi-Algorithmus (VA) wird in dieser Arbeit sowohl zum Training als auch für die Erkennung eingesetzt. Dabei werden jedoch unterschiedliche Resultate ausgenutzt: Für die Erkennung ist es wichtig, daß sich nach der Abarbeitung des VA durch eine sog. Zustandsrückverfolgung (oder *backtracking*) die optimale Zustandsfolge offenlegen läßt. Weil dadurch auch die Zuordnung der Merkmale zu den Zuständen eines Modells bekannt ist, können die zustandsabhängigen Parameter der Modelle auf direkte Weise bestimmt werden (s. Kap. 6.3.4). Für die Erkennung wird ausgenutzt, daß sich am Ende der Abarbeitung des VA die Wahrscheinlichkeitsdichte dafür ergibt, daß ein Modell

bei gegebener Merkmalsvektorfolge auf dem optimalen Zustandspfad durchlaufen wurde. Dieser Wert ist eine sehr gute Näherung für die eigentlich benötigte sog. Erzeugungswahrscheinlichkeit(sdichte) $F(\mathbf{X}|\lambda_l)$ (s. Kap. 6.3.5).

Zwei Gründe sprechen für den Viterbi-Algorithmus als Alternative zum *Baum-Welch-Algorithmus* (BWA) [Bau67, Bau70, Jua85], mit dem die für Training und Erkennung notwendigen Berechnungen *exakt* durchgeführt werden können:

1. Der VA läßt sich rechentechnisch weit schneller abarbeiten als der BWA. Das liegt daran, daß der VA lediglich den *optimalen* Pfad durch das Modell berücksichtigt, während der BWA *alle möglichen* Pfade durchläuft. Der dadurch bedingte Fehler bei der Zuordnung der Merkmale zu den einzelnen Modellzuständen für das Training und bei der Berechnung der Erzeugungswahrscheinlichkeit für die Erkennung ist jedoch sehr klein und wirkt sich nur sehr wenig auf die Erkennungsergebnisse aus (s. beispielsweise [Rab89, Pla95]).

2. Im VA treten nur multiplikative Verknüpfungen auf, im BWA dagegen werden sowohl Addition als auch Multiplikation verwendet. Der VA ist daher im Gegensatz zum BWA sehr leicht in eine *logarithmische Darstellung* überzuführen, was dann zu Additionen der einzelnen logarithmierten Elemente führt. Eine logarithmische Darstellung ist aber sehr wünschenswert, da sich damit die große numerische Dynamik der enthaltenen WDFs beherrschen läßt (s. Kap. 6.4).

6.3.3.2 Formulierung des Viterbi-Algorithmus

Für eine gegebene Folge von Merkmalsvektoren $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ findet der Viterbi-Algorithmus eine *global* optimale Abfolge von Zuständen $S^{\text{opt}} = s_1^{\text{opt}}, s_2^{\text{opt}}, \ldots, s_T^{\text{opt}}$ durch ein gegebenes Modell λ_l. Der Viterbi-Algorithmus ist dabei sehr effektiv, da er die Suche nach diesem global optimalen Pfad rekursiv durch eine *lokale* Maximierung berechnet. In jedem Rekursionsschritt wird für jeden Zustand S_j eines Modells λ_l ein sogenannter *lokaler Score* $D_{S_j,t}^{\lambda_l}$ berechnet, der die logarithmierte Wahrscheinlichkeitsdichte darstellt, bis zum Zeitpunkt t in den Zustand S_j gelangt zu sein. Die schrittweise optimalen Übergänge werden in der Matrix $\mathbf{\Psi}^{\lambda_l}$ gespeichert, so daß am *Ende* auf den optimalen Pfad rückgeschlossen werden kann.

Logarithmierte Größen werden im folgenden immer durch Großbuchstaben gekennzeichnet: $F_{S_j}^{\lambda_l}(\mathbf{x}) = \ln f_{S_j}^{\lambda_l}(\mathbf{x})$, $A_{S_i S_j}^{\lambda_l} = \ln a_{S_i S_j}^{\lambda_l}$ und $\Pi_{S_j}^{\lambda_l} = \ln \pi_{S_j}^{\lambda_l}$. Dabei müssen Werte, die durch die Logarithmierung zu klein werden können, durch Begrenzung des Zahlenbereichs abgefangen werden (vgl. Kap. 6.4). Im einzelnen werden beim Viterbi-Algorithmus folgende Schritte durchlaufen:

1. **Initialisierung** ($t = 1$): Die Anfangswerte der lokalen Scores $D_{S_j,1}^{\lambda_l}$ und der Rückverfolgungsmatrixelemente $\psi_{S_j,1}^{\lambda_l}$ werden festgelegt:

$$D_{S_j,1}^{\lambda_l} = \Pi_{S_1}^{\lambda_l} + F_{S_j}^{\lambda_l}(\mathbf{x}_1) \quad \text{und} \tag{6.17}$$

$$\psi_{S_j,1}^{\lambda_l} = 1 \quad \text{für} \quad j = 1, \ldots, N. \tag{6.18}$$

2. **Rekursionsschritt** ($t - 1 \rightarrow t$): Zur Bildung des aktuellen lokalen Scores $D_{S_j,t}^{\lambda_l}$ in einem Zustand S_j wird der Vorgängerzustand berücksichtigt, dessen lokaler Score zu-

sammen mit der Übergangswahrscheinlichkeit maximal ist. In den Rückverfolgungsmatrixelementen $\psi^{\lambda_l}_{S_j,t}$ wird dieser beste Vorgängerzustand gespeichert:

$$D^{\lambda_l}_{S_j,t} = \max_i [D^{\lambda_l}_{S_i,t-1} + A^{\lambda_l}_{S_i S_j}] + F^{\lambda_l}_{S_j}(\mathbf{x}_t) \quad \text{und} \tag{6.19}$$

$$\psi^{\lambda_l}_{S_j,t} = \operatorname*{argmax}_i [D^{\lambda_l}_{S_i,t-1} + A^{\lambda_l}_{S_i S_j}] \quad \text{für} \quad j = 1, \ldots, N. \tag{6.20}$$

3. **Resultate** ($t = T$): Die Wahrscheinlichkeitsdichte $F(\mathbf{X}|\lambda_l)$ dafür, daß die gegebene Merkmalsfolge $\mathbf{X}$ vom Modell λ_l stammt, ergibt sich zum Endzeitpunkt T näherungsweise als lokaler Score des letzten Zustandes:

$$F(\mathbf{X}|\lambda_l) \approx \tilde{F}(\mathbf{X}|\lambda_l) = F(\mathbf{X}, S^{\text{opt}}|\lambda_l) = D^{\lambda_l}_{S_N,T}. \tag{6.21}$$

Startet man im letzten Zustand S_N, so läßt sich nachträglich der optimale Zustandspfad durch das Modell λ_l durch Rückverfolgung mit Hilfe der Matrix $\mathbf{\Psi}^{\lambda_l}$ bestimmen:

$$s^{\text{opt}}_T = S_N \quad \text{und} \tag{6.22}$$

$$s^{\text{opt}}_{t-1} = \psi^{\lambda_l}_{s^{\text{opt}}_t,t} \quad \text{für} \quad t = T, \ldots, 2. \tag{6.23}$$

6.3.4 Training mit dem Viterbi-Algorithmus

Beim Training werden R^{λ_l} unterschiedliche Trainings-Merkmalssequenzen $\mathbf{X}^{\text{Tr},\lambda_l}_r$, $r = 1, \ldots, R^{\lambda_l}$ für jedes Modell λ_l unterschieden. Sie haben jeweils die Länge $T^{\lambda_l}_r$. Mit dem Viterbi-Algorithmus läßt sich nun für jede der R^{λ_l} Trainingssequenzen für ein Modell die optimale Zustandsfolge $S^{\text{opt},\lambda_l}_r$ mit den Gln. (6.22) und (6.23) rückverfolgen, so daß für jeden einzelnen Merkmalsvektor die genaue Zuordnung zu einem Modellzustand bekannt ist. Das Training läuft nun für jedes Modell λ_l iterativ folgendermaßen ab (analog zum Baum-Welch-Training abgeleitet aus [Hua90]):

1. **Initialisierung:** Die Codebuchparameter und die spezifischen Modellparameter werden wie in Kap. 6.3.2 beschrieben mit Anfangswerten belegt:

$$\dot{\lambda}_l = \lambda_l(\dot{\boldsymbol{\pi}}^{\lambda_l}, \dot{\mathbf{A}}^{\lambda_l}, \dot{\mathbf{C}}^{\lambda_l}, \dot{\mathbf{M}}, \dot{\mathbf{\Sigma}}). \tag{6.24}$$

2. **Optimierungsschritt:** Mit dem Viterbi-Algorithmus aus Kap. 6.3.3.2 kann für die momentan gegebenen Modellparameter $\boldsymbol{\pi}^{\lambda_l}$, $\mathbf{A}^{\lambda_l}$, $\mathbf{C}^{\lambda_l}$, $\mathbf{M}$ und $\mathbf{\Sigma}$ die optimale Zustandssequenz $S^{\text{opt},\lambda_l}_r$ bestimmt werden. Mit diesem Wissen können die Übergänge $N^{\text{Tr},\lambda_l}_{S_i,S_j}$ von einem Zustand S_i nach S_j und die Aufenthalte $N^{\text{Tr},\lambda_l}_{S_i}$ in einem Zustand S_i gezählt werden:

$$N^{\text{Tr},\lambda_l}_{S_i,S_j} = \sum_{r=1}^{R^{\lambda_l}} \sum_{t=1}^{T^{\lambda_l}_r - 1} \delta(S_i - s^{\text{opt},\lambda_l}_{r,t}) \delta(S_j - s^{\text{opt},\lambda_l}_{r,t+1}) \quad \text{und} \tag{6.25}$$

$$N^{\text{Tr},\lambda_l}_{S_i} = \sum_{r=1}^{R^{\lambda_l}} \sum_{t=1}^{T^{\lambda_l}_r} \delta(S_i - s^{\text{opt},\lambda_l}_{r,t}). \tag{6.26}$$

Die Rückschlußwahrscheinlichkeit $f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l})$ in einem „festgehaltenen" Zustand ist die Grundlage der meisten Nachschätzformeln. Bei der aus der Rückverfolgung gefundenen Zuordnung zwischen Merkmalen und Zuständen läßt sie sich aus den Prototypen und den Mixturen wie folgt angeben:

$$f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l}) = \frac{c^{\lambda_l}_{s_{r,t}^{\text{opt}},v_k} f_{v_k}(\mathbf{x}_t)}{f(\mathbf{x}_t)} = \frac{c^{\lambda_l}_{s_{r,t}^{\text{opt}},v_k} f_{v_k}(\mathbf{x}_t)}{\sum_{k=1}^{L} c^{\lambda_l}_{s_{r,t}^{\text{opt}},v_k} f_{v_k}(\mathbf{x}_t)}. \tag{6.27}$$

Daraus ergeben sich nun die Nachschätzformeln für die modellabhängigen Größen:

$$\hat{a}^{\lambda_l}_{S_i S_j} = \frac{N^{\text{Tr},\lambda_l}_{S_i,S_j}}{N^{\text{Tr},\lambda_l}_{S_i}} \quad \text{und} \tag{6.28}$$

$$\hat{c}^{\lambda_l}_{S_i v_k} = \frac{1}{N^{\text{Tr},\lambda_l}_{S_i}} \sum_{r=1}^{R^{\lambda_l}} \sum_{t=1}^{T_r^{\lambda_l}} f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l}) \tag{6.29}$$

und die Nachschätzformeln für die modellunabhängigen Größen (also das Codebuch):

$$\hat{\boldsymbol{\mu}}_{v_k} = \frac{\sum_{r=1}^{R^{\lambda_l}} \sum_{t=1}^{T_r^{\lambda_l}} f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l}) \cdot \mathbf{x}_t}{\sum_{r=1}^{R^{\lambda_l}} \sum_{t=1}^{T_r^{\lambda_l}} f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l})}, \tag{6.30}$$

$$\hat{\boldsymbol{\sigma}}_{v_k} = \frac{\sum_{m=1}^{R^{\lambda_l}} \sum_{t=1}^{T_r^{\lambda_l}} f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l}) \cdot \mathbf{x}_t^2}{\sum_{r=1}^{R^{\lambda_l}} \sum_{t=1}^{T_r^{\lambda_l}} f(v_k|\mathbf{x}_t, S_i = s_{r,t}^{\text{opt},\lambda_l})} - \hat{\boldsymbol{\mu}}^2_{v_k}. \tag{6.31}$$

3. **Abbruchkriterium und Resultat:** Für das Abbruchkriterium wird die Summe der genäherten Erzeugungswahrscheinlichkeitsdichten über alle Modelle benötigt:

$$K = \sum_{l=1}^{M} D^{\lambda_l}_{S_N,T}. \tag{6.32}$$

Wenn die relative Verbesserung von $\hat{K}$ im Vergleich zum letzten K unter der Schwelle ϵ^{hmm} bleibt:

$$(K - \hat{K})/\hat{K} < \epsilon^{\text{hmm}}, \tag{6.33}$$

wird das Training abgebrochen. Entsprechend der vorgegebenen Abbruchschwelle ϵ^{hmm} sind die Modelle λ_l nun optimal an die Trainingsdaten angepaßt. Ist die Abbruchbedingung nach Gl. (6.33) nicht erfüllt, wird die Iteration mit einer erneuten Viterbi-Suche über die *nachgeschätzten* Modellparameter mit Schritt 2 fortgesetzt.

Über die Abbruchschwelle ϵ^{hmm} ist einstellbar, wie genau die Modelle an die Trainingsdaten angepaßt sind. Ein Erfahrungswert von $\epsilon^{\text{hmm}} = 10^{-3}$ hat sich als günstiger Kompromiß

zwischen Genauigkeit der Modellierung und Variabilität der zu erkennenden Gestendaten herausgestellt. Um die Genauigkeit der Modellierung zu verringern, besteht auch alternativ die Möglichkeit, die Nachschätzformeln nicht auf den kompletten Parametersatz anzuwenden. Insbesondere kann man das Codebuch im Anfangszustand belassen und nur die Mixturen und die Zustandsübergänge optimieren (vgl. [Win96]). Es zeigte sich jedoch, daß bei der Erkennung von Gesten gerade das Nachschätzen des Codebuchs sehr viel zu einer guten Erkennungsleistung beiträgt.

6.3.5 Isolierte Erkennung mit dem Viterbi-Algorithmus

Bei der isolierten Erkennung liegt eine zeitliche Merkmalssequenz $\mathbf{X}$ mit bekannter Länge T vor. Die Grundlage dieser Erkennung bildet die genäherte Erzeugungswahrscheinlichkeitsdichte (*likelihood*) $\tilde{F}(\mathbf{X}|\lambda_l)$ aus Gl. (6.21), die der Viterbi-Algorithmus im T-ten Zeitschritt als lokalen Score $D^{\lambda_l}_{S_N,T}$ im letzten Zustand S_N berechnet. In einer *maximum-likelihood*-Entscheidung (ML-Entscheidung) wird der Index des Modells, das die maximale Erzeugungswahrscheinlichkeitsdichte besitzt, als Klassifikationsergebnis geliefert [Rab89]:

$$\lambda^{\mathrm{Er}}_{\mathbf{X}} = \operatorname*{argmax}_{l} \tilde{F}(\mathbf{X}|\lambda_l) \quad \text{für} \quad l = 1, \ldots, M. \tag{6.34}$$

Für die kontinuierliche Erkennung sind erweiterte Strategien notwendig, die aber prinzipiell alle auf eine ML-Entscheidung hinauslaufen. Allerdings muß zusätzlich eine automatische zeitliche Segmentierung durchgeführt werden. Verfahren, die eine kontinuierliche Erkennung erlauben, werden in Kap. 9 vorgestellt.

6.4 HMM-Vorverarbeitung

Abhängig vom verwendeten Merkmalsextraktionsverfahren (s. Kap. 7) können die einzelnen Komponenten eines Merkmalsvektors unterschiedlich großen numerischen Schwankungen unterworfen sein. Verhältnismäßig kleine Varianzen in einer Dimension der Merkmalsvektoren können aber sehr große WDF-Werte verursachen. Die Logarithmierung kann nur bedingt Abhilfe schaffen, weil sie erst nach der Summation der Teildichten gemäß Gl. (6.1) angewendet werden kann. Außerdem werden große WDF-Werte im Verlauf der Viterbi-Iteration immer weiter aufakkumuliert, was numerische Probleme unter Umständen verschlimmert. Bei solchen „spitzen“ WDFs und einer geringen Codebuchgröße kann es darüberhinaus leicht passieren, daß Merkmalsvektoren nicht mehr vom Codebuch abgedeckt werden können und daher Dichtewerte sehr nahe bei Null produziert werden, die nach der Logarithmierung sehr große negative Werte ergeben. In der Implementierung der HMMs werden solche Extremwerte auf einen minimalen und maximalen Schwellwert begrenzt. Allerdings kann es dadurch zu Ungenauigkeiten in der Modellierung kommen, die sich eventuell negativ auf die Erkennungsraten auswirken.

Um ein Ansprechen der Schwellwerte möglichst zu vermeiden, werden die Komponenten der Merkmalsvektoren daher mittelwertsbereinigt und so normiert, daß sie die Varianz 1 haben:

$$\mathbf{x} = \frac{\mathbf{x}' - \boldsymbol{\mu}^{\mathrm{Tr}}}{\sqrt{\boldsymbol{\sigma}^{\mathrm{Tr}}}}. \tag{6.35}$$

Der dazu benötigte Mittelwertsvektor $\boldsymbol{\mu}^{\mathrm{Tr}}$ und Varianzvektor $\boldsymbol{\sigma}^{\mathrm{Tr}}$ wird *vor* dem Training der Modelle aus den untransformierten, originalen Trainingsdaten $\mathbf{X}'^{\mathrm{Tr}}$ geschätzt [Bro81]:

$$\boldsymbol{\mu}^{\mathrm{Tr}} = \frac{1}{|\mathbf{X}'^{\mathrm{Tr}}|} \sum_{\mathbf{x}' \in \mathbf{X}'^{\mathrm{Tr}}} \mathbf{x}' \quad \text{und} \tag{6.36}$$

$$\boldsymbol{\sigma}^{\mathrm{Tr}} = \left[\frac{1}{|\mathbf{X}'^{\mathrm{Tr}}| - 1} \sum_{\mathbf{x}' \in \mathbf{X}'^{\mathrm{Tr}}} \mathbf{x}'^2 \right] - (\boldsymbol{\mu}^{\mathrm{Tr}})^2. \tag{6.37}$$

Wenn nicht anders angegeben, werden alle Merkmalsvektoren aus Kap. 7 auf diese Weise der HMM-Vorverarbeitung unterworfen. Um eine allgemeine Schreibweise zu ermöglichen, wird auch in den seltenen Fällen, in denen unnormierte Merkmalsvektoren verwendet werden, die ungestrichene Schreibweise $\mathbf{x}$ verwendet. Mißverständnisse sind ausgeschlossen, da im Einzelfall auf die Art der Merkmale hingewiesen wird. Die HMM-Vorverarbeitung nach Gl. (6.35) hat sich als sehr entscheidend für die Leistungsfähigkeit und Robustheit der HMMs erwiesen.

Nachdem die für diese Arbeit gewählte Modellierung vorgestellt wurde, kann in Kap. 7 auf die verschiedenen Möglichkeiten der Merkmalsextraktion eingegangen werden. Diese Verfahren sind teilweise allgemeiner Natur, teilweise sind sie speziell auf die HMM-Modellierung abgestimmt. Erst nach einer Evaluierungsphase kann festgestellt werden, welche Kombination aus Merkmalsextraktion und HMM für die Erkennung von Bildsequenzen am besten geeignet ist (s. Kap. 8).

Kapitel 7

Verfahren der Merkmalsextraktion

7.1 Überblick über die untersuchten Merkmalsextraktionsverfahren

Die Merkmalsextraktion hat im Zusammenhang mit der Klassifikation von Bildsequenzen mit HMMs mehrere Aufgaben zu erfüllen:

- Die wichtigste Aufgabe besteht in der „Serialisierung“ der Daten: die Transformation einer dreidimensionalen räumlich-zeitlichen Information in eine eindimensionale, rein zeitliche Information (vgl. Kap. 6.1).
- Allgemein sollte dabei jede Merkmalsextraktion nur die charakteristische Information extrahieren, die für die Unterscheidbarkeit notwendig ist, und gleichzeitig die redundante Information verwerfen [Rus93].
- Damit einher geht der Wunsch nach einer starken Datenreduktion: der Farbbilddatenstrom resultiert in sehr große Datenraten von bis zu 30 MByte/s [Sch94b], die vom Klassifikator kaum noch zu handhaben sind. Nach einer entsprechenden Vorverarbeitung und Merkmalsextraktion mit den hier vorgestellten Verfahren bleibt noch ein Datenstrom von ca. 1–15 kByte/s, was einer Reduktion von 30000–2000:1 entspricht. Diese Datenrate liegt etwa in derselben Größenordnung wie die Datenrate nach der Merkmalsextraktion bei Sprachsignalen [Fel84], so daß der Rechenaufwand bei der Bildsequenzerkennung mit HMMs mit dem Aufwand bei der Spracherkennung verglichen werden kann.

Es existieren zwei grundsätzlich verschiedene Prinzipien zur Merkmalsextraktion in der Bildverarbeitung [Har91, Hua95]):

1. **Modellbasierte Verfahren** setzen eine parametrisierte, abstrakte Repräsentation des gesuchten Bildobjektes voraus (ein *Modell*). Dieses Modell wird dann so mit der Bildfunktion über numerische Optimierungsverfahren in Wechselwirkung gebracht, daß sich die optimalen Modellparameter als Merkmale verwenden lassen (vgl. z. B. [Bal82, Bun85]).

   Modellbasierte Verfahren haben den Vorteil, daß mit ihnen das gesuchte Objekt sehr detailliert beschrieben werden kann. Einer der Nachteile ist allerdings, daß ein

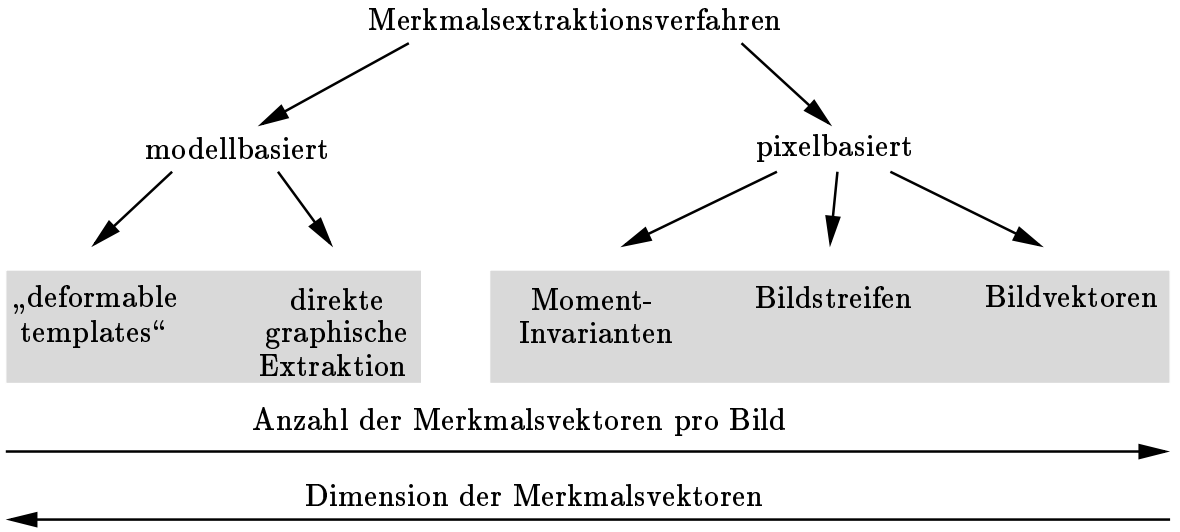


Bild 7.1: Übersicht über die untersuchten Merkmalsextraktionsverfahren

genaues Wissen über das gesuchte Objekt vorhanden sein muß, damit es mathematisch beschrieben werden kann. Für jedes gesuchte Objekt oder für verschiedene Ausprägungen eines Objektes sind neue Modellierungen erforderlich: modellbasierte Verfahren sind nicht universell einsetzbar.

Im Rahmen dieser Arbeit wurden zwei modellbasierte Verfahren entwickelt bzw. optimiert: ein zweidimensionales Verfahren mit sog. *deformable templates* zur Extraktion von Augenparametern (s. Kap. 7.2.1) und eine dreidimensionale sog. *direkte graphische Extraktion* am Beispiel der Handparameter (s. Kap. 7.2.2).

2. **Pixelbasierte Verfahren** verarbeiten direkt die Pixelinformation eines Bildes, ohne das gesuchte Objekt zu kennen. Die Verarbeitung beruht auf mathematischen Operationen und Transformationen, die Zahlenwerte liefern, die wiederum direkt als Merkmale verwendet werden können (vgl. z. B. [Gon87, Jai89, Pra91]).

   Pixelbasierte Verfahren haben prinzipiell den Vorteil der Allgemeingültigkeit, da sie keine Annahmen über ein zu suchendes Objekt treffen. Andererseits liefern sie dann auch eine vom Objekt losgelöste, abstrakte Information, in der sich das Objekt nicht mehr erkennen läßt. Wie wertvoll und diskriminativ diese Information ist, läßt sich nur im Zusammenhang mit der gewünschten Anwendung und für den Fall der Gestikerkennung nach erfolgter Klassifikation beurteilen.

   Drei pixelbasierte Verfahren wurden im Rahmen dieser Arbeit entwickelt bzw. implementiert. Die auf diesen Verfahren basierenden Merkmalsvektoren verwenden Transformationskoeffizienten auf der Basis von *Moment-Invarianten* (s. Kap. 7.3.1), sog. *Bildstreifen* (s. Kap. 7.3.5) und sog. *Bildvektoren* (s. Kap. 7.3.6).

Eine Übersicht über die untersuchten Verfahren ist in Bild 7.1 dargestellt. Dort sind die Merkmalsextraktionsverfahren so angeordnet, daß sich die Anzahl der Merkmalsvektoren pro Bild tendenziell von links nach rechts erhöht. Es wird sich bei der Vorstellung der einzelnen Verfahren zeigen, daß sich gleichzeitig die Dimension eines einzelnen Merkmalsvektors entsprechend erniedrigt. Es gilt also in grober Näherung, daß die Gesamtzahl der Merkmalsvektorkomponenten pro Bild in etwa gleichbleibt.

In Bild 7.2 ist ein Ausschnitt aus einer Bildsequenz dargestellt. Die Bilder erscheinen unter den Zeitindizes $\ldots, t, t+1, t+2, \ldots$. Jedes der Verfahren liefert pro Bild minde-

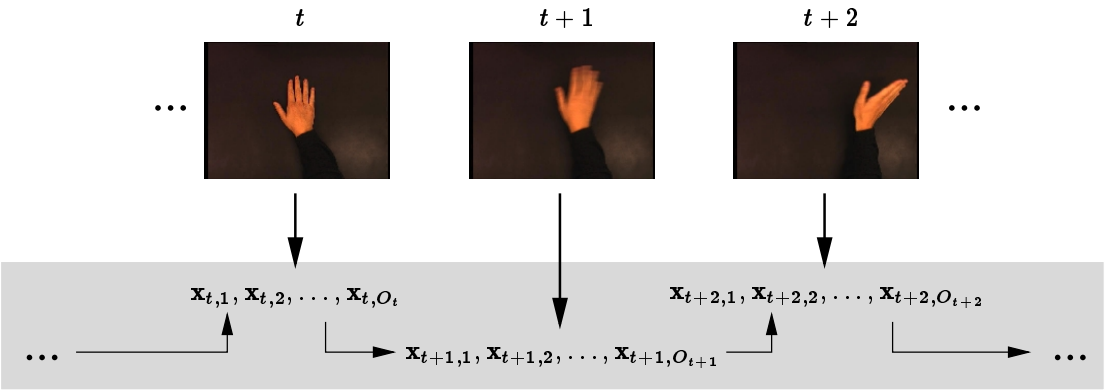


Bild 7.2: Zusammenhang zwischen Bild- und Merkmalssequenz

stens einen und bis zu O_t Merkmalsvektoren; die Anzahl der Merkmalsvektoren pro Bild kann dabei variieren. Wie im Bild 7.2 ersichtlich, werden die jeweiligen Merkmalsvektoren $\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,O_t}$ eines Bildes mit dem Zeitindex t in einer vom Merkmalsextraktionsverfahren abhängigen Reihenfolge zeitlich hintereinander angeordnet. Die Merkmalsvektoren hintereinanderliegender Bilder werden dann zu einer einzigen Zeitreihe

$$\ldots, \mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,O_t}, \mathbf{x}_{t+1,1}, \ldots, \mathbf{x}_{t+1,O_{t+1}}, \mathbf{x}_{t+2,1}, \ldots, \mathbf{x}_{t+2,O_{t+2}}, \ldots \tag{7.1}$$

verbunden. Zur Vereinfachung der Darstellung wird nun der *Merkmalszeittakt* t' eingeführt, der zusammen mit den emittierten Merkmalsvektoren hochgezählt wird. Wegen der unterschiedlichen Anzahl der Merkmalsvektoren pro Bild sind die Zeitintervalle, die durch den Merkmalszeittakt markiert werden, nicht unbedingt äquidistant. Alle Modellierungsalgorithmen in Kap. 6 und 9 arbeiten mit dem Merkmalszeittakt t'. Es kann jederzeit eine *eindeutige* Zuordnung zwischen Bildzeittakt t und Merkmalszeittakt t' hergestellt werden. Für den Spezialfall, daß *genau ein* Merkmal pro Bild extrahiert wird, gilt $t' = t$. Da stets aus dem Zusammenhang ersichtlich ist, ob Bild- oder Merkmalszeittakt gemeint sind, wird im folgenden für beide das Formelzeichen t verwendet.

7.2 Modellbasierte Verfahren

Werden die parametrischen Modell-Repräsentationen wie bei den hier vorgestellten Verfahren mit Mitteln der numerischen Optimierung an das Bildobjekt angepaßt, so sind stets Anfangswerte für die Modellparameter erforderlich, die nicht zu weit von den gesuchten Parametern entfernt sein dürfen: sie müssen im „Einzugsbereich“ des Optimierungsverfahrens liegen. Das Problem der Anfangswerte kann umgangen werden, wenn eine feste Anfangsposition des Bildobjektes vorgeschrieben wird. Damit man allerdings von der Kooperation des Benutzers unabhängig ist, wurden bei jedem Verfahren zusätzliche Maßnahmen ergriffen, mit denen sich die Anfangswerte berechnen lassen.

Im laufenden Betrieb wird davon ausgegangen, daß sich das zu modellierende Objekt von Bild zu Bild nur geringfügig verändert. Die Verfahren müssen dann nur noch in der Lage sein, die Objekte zu *verfolgen* (oder *tracken*).

Da mit den modellbasierten Verfahren ein Bildobjekt repräsentiert wird, läßt sich immer ein Maß dafür definieren, wie genau das Modell mit dem Objekt in Deckung gebracht wurde. Damit läßt sich im Unterschied zu den pixelbasierten Verfahren die *Güte* der Merkmalsextraktion angeben. Über die Güte lassen sich die modellbasierten

Verfahren unabhängig vom darauf aufbauenden Klassifikationsverfahren beurteilen, was in den Kapiteln 7.2.1, 7.2.2 und 7.4 geschieht. Die Qualität der pixelbasierten Verfahren läßt sich dagegen nur *indirekt* nach dem Klassifikationsschritt angeben (s. Kap. 8).

7.2.1 „Deformable Templates“: zweidimensionale Modellierung am Beispiel der Augen

Die Modellierung der Augen erfolgte im Rahmen dieser Arbeit weitgehend in [Sch96a] und steht *repräsentativ* für eine Klasse von Verfahren, die zweidimensionale sog. *deformable templates* verwendet. Sie wurden von [Yui89] eingeführt und auf verschiedene Gesichtsmerkmale wie Augen, Augenbrauen und Mund angewendet. Mit entsprechend angepaßten Templates kann neben der Gesichts- auch die Handkontur erfaßt werden [Cho93, Lan95]. Damit sind Templates universell zur Extraktion aller Arten von menschlichen Merkmalen geeignet, wie sie für den visuellen Dialog von Bedeutung sind.

In [Sch96a] wird zunächst die absolute Position der Augen durch eine globale Suche nach den kreisförmigen *Irisgrenzen* im Kantenbild bestimmt. Hierfür wird die *Hough-Transformation* verwendet [Bal82, Gon87].

Mit den Positionen der Iris sind die Lagen der Augen hinreichend bestimmt, so daß das eigentliche Tracken der Augen durch das Nachführen zweier Augen-Templates bewerkstelligt werden kann. Die Augenmodelle bestehen jeweils aus einem Kreis und zwei Parabelstücken, die durch insgesamt 11 Parameter in ihrer Lage und Form verändert werden können. Die Anpassung der Modelle erfolgt auf vier verschiedenen Bilddarstellungen (Grauwert-, Kanten-, *Peak*- und *Valley*-Bild). Die Wechselwirkung der Templates mit den verschiedenen Bilddarstellungen wird jeweils über eine sog. Energiefunktion hergestellt. Zusammen mit der inneren Energie eines Templates wird eine Gesamtenergiefunktion gebildet, deren Ausprägung insgesamt durch neun Koeffizienten repräsentiert wird. In insgesamt 6 Teiloptimierungsschritten (in [Yui89] *Epochen* genannt) wird jeweils ein Teil der Template-Parameter optimiert, indem die entsprechenden Teilenergien minimiert werden.

Die Evaluierung und Optimierung der Verfahren wurde anhand von 131 Portrait-Testbildern verschiedener Personen vorgenommen. Die Referenzlage der Iris-Kreise und der Augen-Templates wurde manuell festgelegt. Damit konnte eine mehrdimensionale Abstandsfunktion eingeführt werden, über die sich ein Gütemaß berechnen ließ. Durch die Überlagerung der Iris-Kreise und Augen-Templates mit den Testbildern ließ sich heuristisch ein Grenzwert des Gütemaßes festlegen, ab dem definitionsgemäß ein Iris-Kreis oder Augen-Template als korrekt positioniert bezeichnet wurde. Die *Positionierungsrate* bezeichnet dann die relative Anzahl der korrekt positionierten Modelle.

Die Positionierungsrate bei den Iris-Kreisen betrug nach dieser Evaluierungsmethode 81 %. Die Positionierungsrate der Augen-Templates betrug bei der Implementierung nach [Yui89] 25 %. Durch Optimierung der Energiefunktionskoeffizienten konnte eine Optimierungsepoche eingespart werden, und es ergab sich bei insgesamt unveränderter Anzahl der Iterationsschritte eine verbesserte Positionierungsrate von 81 %.

Die Extraktion der Template-Parameter beider Augen liefert 22 Parameter, die sich direkt als Komponenten eines Merkmalsvektors $\mathbf{x}$ verwerten lassen. Über diese Parameter sind neben der Lage und der Gestalt der Augen auch indirekt die Blickrichtung sowie die Position und die Orientierung des Kopfes festgelegt. Mit einem solchen Merkmalsvektor lassen sich also neben Kopfgesten auch Teile der Mimik erfassen. Es wird ein Merkmalsvektor pro Bild berechnet.

7.2.2 Direkte graphische Extraktion: dreidimensionale Modellierung am Beispiel der Hand

Die zweidimensionalen Templates wurden mit einer dreidimensionalen Modellierung weiterentwickelt und verfeinert. Das Verfahren wurde im Rahmen dieser Arbeit in [Lück96] entwickelt und am Beispiel der Modellierung der Hand implementiert. Es läßt sich universell auf jedes Objekt anwenden, das sich mit dreidimensionalen graphischen Renderverfahren [Fol90] realistisch darstellen läßt.

Dazu wurde ein 3D-Modell der Hand bis zum Handgelenk entworfen, das die echte menschliche Hand möglichst exakt nachbilden sollte. Die Proportionen der Hand können mit 37 statischen Parametern eingestellt werden. Um die Beweglichkeit einer realen Hand nachvollziehen zu können, sind alle Fingergelenke, die Daumengelenke — die Stellung des Daumens beeinflußt dabei auch die Ausformung des Handballens — und die Krümmung des Handrückens nachgebildet und zusammen mit der Handposition und -orientierung mit 30 dynamischen Parametern steuerbar. Dieses Handmodell ist mit dem Handagenten des 3D-Szenen-Editors identisch, so daß sich Beispielbilder in Kap. 3.5 finden.

Ein wichtiges Kennzeichen des Verfahrens ist das Ersetzen des Systems heuristischer Energiefunktionen der zweidimensionalen Templates aus Kap. 7.2.1 durch die sog. *direkte graphische Extraktion* (DGE): das gerenderte Bild wird auf *Pixelebene* direkt mit dem realen, *monokularen* Bild verglichen, wobei die *Ähnlichkeit* beider Bilder über den mittleren quadratischen Pixelfehler bewertet wird. Über numerische Optimierungsverfahren werden dann Modell und Bildobjekt möglichst exakt in Deckung gebracht. Das unterscheidet dieses Verfahren erheblich von anderen existierenden 3D-Modellverfahren, die beispielsweise Stereobilder [Del98] oder 3D-Tiefenbilder [Hea96] verwenden, farbige Marker benötigen [Hie96] oder nur für nicht-gegliederte, feste Objekte geeignet sind [Kri90].

Die DGE setzt voraus, daß das Objekt mit einem Segmentierungsverfahren vom Hintergrund getrennt wurde. Da dieses Verfahren ebenfalls nur zum Tracken geeignet ist, werden Modell und Hand in einer Initialisierungsphase über die Ermittlung ihrer Schwerpunkte in grobe Deckung gebracht. Zusätzlich wird aber eine definierte Start-Orientierung und -Fingerstellung vorausgesetzt.

Die prinzipielle Funktionsfähigkeit der DGE konnte in [Lück96] nachgewiesen werden: es ist möglich, ein Handmodell durch numerische Minimierung des mittleren quadratischen Pixelfehlers einem zweiten Handmodell folgen zu lassen, wenn die Änderungen in zwei aufeinanderfolgenden Bildern nicht zu groß werden. Allerdings muß die Anzahl der freien Parameter stark eingeschränkt werden. Im Beispiel waren es die Translations- und Rotationsparameter für die Handposition und -orientierung und ein Fingerkrümmungsparameter, mit dem die Stellung aller Fingergelenke gekoppelt war.

Je nach der zugelassenen Anzahl der freien Parameter ergeben sich für zwei Hände typischerweise 14, maximal bis zu 60 Komponenten, die zu einem Merkmalsvektor zusammengefaßt werden können. Es wird ein Merkmalsvektor pro Bild berechnet.

7.3 Pixelbasierte Verfahren

Im folgenden werden drei pixelbasierte Merkmalsextraktionsverfahren vorgestellt, auf deren Basis sich entsprechende Merkmalsvektoren bilden lassen. Dabei werden die Bilder durch Moment-Invarianten (s. Kap. 7.3.4), durch Bildstreifen (s. Kap. 7.3.5) bzw. durch Bildvektoren (s. Kap. 7.3.6) repräsentiert.

Moment-Invarianten in der Ausprägung von Hu-Moment-Invarianten [Hu62] (s. Kap. 7.3.2) und Zernike-Moment-Invarianten [Tea80] (s. Kap. 7.3.3) werden schon lange für die Erkennung starrer Objekte in der Bildverarbeitung verwendet. Es war allerdings zunächst ungewiß, ob sie sich für die Gestikerkennung eignen, da es sich hier um die Modellierung von Bewegungen nicht-starrer Objekte handelt. Außerdem reichen für die Bewegungserkennung die Moment-Invarianten alleine nicht aus, so daß zusätzliche Merkmale als Komponenten in den Merkmalsvektor aufgenommen werden mußten (s. Kap. 7.3.4). Die Bildung beider Arten von Invarianten beruht auf den allgemeinen Momenten, die in Kap. 7.3.1 vorgestellt werden.

Bildstreifen stellen einfache Merkmale dar, die zum ersten Mal von [Yam92] zur Modellierung von Bildsequenzen eingesetzt wurden. Sie haben leicht erkennbare Schwächen und werden in dieser Arbeit insbesondere zum Vergleich mit den neu entwickelten Bildvektoren [Mor97a, Mor97b] eingesetzt.

Die pixelbasierten Verfahren arbeiten prinzipiell sowohl auf segmentierten Grauwertbildern als auch auf der darauf aufbauenden Kantenbildkomponente von Gradientenbildern (zur Nomenklatur s. Kap. 5.2). Zur Vereinfachung der Darstellung werden allerdings im folgenden nur Grauwertbilder erwähnt, außer wenn spezielle Eigenschaften von Gradientenbildern ausgenutzt werden.

7.3.1 Allgemeine Momente

7.3.1.1 Kontinuierliche Momente und Invarianz

Die allgemeinen Momente der Ordnung $p+q$ sind über der kontinuierlichen Bildfunktion $f(x,y)$ definiert als [Jai89]:

$$m_{pq} = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} x^p y^q f(x,y) \mathrm{d}x\mathrm{d}y, \tag{7.2}$$

wobei vorausgesetzt wird, daß die Bildfunktion nur in einem endlichen Bildbereich größer Null ist. Momente sind somit Koeffizienten, die sich durch die Entwicklung der Bildfunktion $f(x,y)$ nach den Polynomen $x^p y^q$ ergeben. Der sog. *Schwerpunkt* $(\bar{x},\bar{y})$ einer Bildregion läßt sich dann über das Moment m_{00} (beziffert ein Flächenäquivalent A) und die beiden Momente erster Ordnung definieren:

$$A = m_{00}, \quad \bar{x} = m_{10}/A, \quad \bar{y} = m_{01}/A. \tag{7.3}$$

Berechnet man die Momente bezogen auf diesen Schwerpunkt, so ergeben sich die sog. *Zentralmomente*

$$\mu_{pq} = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \mathrm{d}x\mathrm{d}y. \tag{7.4}$$

Die Zentralmomente lassen sich auch nachträglich aus den allgemeinen Momenten bestimmen [Tea80, Bel91]:

$$\mu_{pq} = \sum_{k=0}^{p} \sum_{l=0}^{q} (-1)^{k+l} \binom{p}{k} \binom{q}{l} m_{p-k,q-l} \cdot \bar{x}^k \bar{y}^l. \tag{7.5}$$

Dieser Zusammenhang ergibt sich durch Ausmultiplizieren der Terme in Gl. (7.4) und anschließender Integration über die einzelnen Summanden. Die Berechnung der allgemeinen Momente nach Gl. (7.2) mit anschließender Zentralisierung mit Gl. (7.5) erfordert dabei wesentlich weniger Rechenoperationen als die Translation jedes einzelnen Pixels, wie sie in Gl. (7.4) gefordert wird. Die nach der Vorschrift

$$\mu_{pq}^{\mathrm{N}} = \mu_{pq} / \mu_{00}^{(p+q+2)/2} \tag{7.6}$$

normierten Zentralmomente μ_{pq}^{N} sind dann invariant gegenüber Translation und Skalierung [Tea80]. Die normierten Zentralmomente nullter und erster Ordnung sind trivial und für weiterführende Berechnungen nicht mehr verwertbar: $\mu_{00}^{\mathrm{N}} = 1$, $\mu_{01}^{\mathrm{N}} = \mu_{10}^{\mathrm{N}} = 0$.

Eine Rotationsinvarianz kann über die Berechnung der Hauptorientierung

$$\Phi = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \tag{7.7}$$

des Bildes erreicht werden. Diese Hauptorientierung ist zunächst allgemein für die Transformation von pixelorientierten Merkmalsextraktionsverfahren einsetzbar (s. Kap. 7.3.6). Darüberhinaus läßt sich, abhängig von Φ, eine sehr aufwendige Transformationsvorschrift für die Momente angeben [Tea80]. Bei der Verwendung von *Moment-Invarianten* (MI) wird allerdings durch das jeweilige Bildungsgesetz eine immanente Rotationsinvarianz hergestellt, so daß für ihre Berechnung nur die normierten Zentralmomente nach Gl. (7.6) benötigt werden. Die beiden in dieser Arbeit verwendeten MIs werden in Kap. 7.3.2 bzw. 7.3.3 vorgestellt.

7.3.1.2 Diskrete und binäre Momente

Im nächsten Schritt werden die allgemeinen Momente auf *diskreten* Bildfunktionen definiert. Dann werden verschiedene *vereinfachte* Berechnungsmethoden für allgemeine Momente auf *binären* Bildfunktionen vorgestellt. Prinzipiell sind alle Verfahren für die Merkmalsextraktion brauchbar. Welche Methode geeignet ist, hängt davon ab, ob es gewünscht ist, das segmentierte Bild oder die Segmentierungs*maske* als Grundlage der Weiterverarbeitung zu verwenden (s. Kap. 5.2). Nur durch die Beurteilung der Klassifikationsergebnisse in Kap. 8 kann festgestellt werden, welche Bildart und welche Verarbeitungsmethode die beste ist.

Für die Zentralisierung und Normierung sowie die Weiterverarbeitung zu Moment-Invarianten können alle der im folgenden vorgestellten Momentenarten verwendet werden, weshalb sie nicht durch verschiedene Formelsymbole unterschieden werden.

Momente auf diskreten Bildfunktionen

Bei diskreten Bildfunktionen $f(n_1, n_2)$ wird das Integral in Gl. (7.2) zur Summe über den endlichen Bildbereich und das Inkrement dx bzw. dy zur Differenz Δn_1 bzw. Δn_2 [Li91]:

$$m_{pq} = \sum_{n_1=0}^{N_1-1} \sum_{n_1=0}^{N_2-1} n_1^p n_2^q f(n_1, n_2) \Delta n_1 \Delta n_2 \Big|_{\Delta n_1 = \Delta n_2 := 1} = \sum_{n_1=0}^{N_1-1} \sum_{n_1=0}^{N_2-1} n_1^p n_2^q f(n_1, n_2). \tag{7.8}$$

Δn_1 und Δn_2 sorgen für eine Vergleichbarkeit der Momente bei unterschiedlich großen Diskretisierungsintervallen. Da die Diskretisierung in dieser Arbeit immer gleich bleibt, wird sie vereinbarungsgemäß auf 1 gesetzt.

Im weiteren Verlauf wird nur noch mit diskreten Bildfunktionen gerechnet, weshalb die Formelsymbole im Zusammenhang mit den diskreten Momenten nicht vom kontinuierlichen Fall unterschieden werden. Aus den Normierungsgrößen wird damit analog zu Gl. (7.3):

$$A = m_{00}, \quad \bar{n}_1 = m_{10}/A, \quad \bar{n}_2 = m_{01}/A. \tag{7.9}$$

Entsprechend können durch Anwendung der Gln. (7.5) und (7.6) auch die diskreten Momente zentralisiert und normiert werden, was auch für alle folgenden Momententypen gilt.

Binäre Flächenmomente

Auf binären Bildfunktionen wie der Segmentierungsmaske $f_\mathrm{b}(n_1, n_2)$ mit dem Vordergrundbereich $\mathcal{V}$ (s. Gl. (5.2) auf Seite 34) vereinfacht sich Gl. (7.8) zu:

$$m_{pq} = \sum_{(n_1,n_2)\in\mathcal{V}} \sum n_1^p n_2^q, \tag{7.10}$$

d. h. es werden nur noch die Polynome über dem Vordergrundbereich $\mathcal{V}$ aufsummiert.

Binäre Flächenmomente über die Kontur

Bei den binären Flächenmomenten steckt die vollständige Information über die Bildfunktion bereits in der *Kontur* $\mathcal{K}$ des Vordergrundbereichs $\mathcal{V}$. Da der Vordergrund aus mehreren Gebieten bestehen kann, die zudem nicht einfach zusammenhängend sein müssen (vgl. Kap. 5.1.2), besteht auch die Kontur im allgemeinen aus mehreren äußeren und inneren Einzelkonturen, die aber alle in $\mathcal{K}$ zusammengefaßt werden. In der Konturdarstellung lassen sich leicht Segmentierungfehler beheben, indem äußere oder innere Konturen, die zu kurz sind, einfach nicht berücksichtigt werden.

Für das Finden der Kontur wird der Kontur-Abtastungs-Algorithmus nach [Pav94] verwendet. Die Punkte $(n_{1,i}, n_{2,i})$ der Kontur sind über den Index $i = 1, \ldots, K$ für äußere Konturanteile im mathematisch positiven Sinn und für innere Konturanteile im negativen Sinn angeordnet. Über das *Greensche Theorem* läßt sich das Flächenintegral in Gl. (7.2) in ein Linienintegral über die Kontur umwandeln [Li91]. Im diskreten Fall gilt dieser Zusammenhang jedoch nur näherungsweise:

$$m_{pq} \approx \frac{1}{p+1} \sum_{i=1}^{K} n_{1,i}^{p+1} n_{2,i}^{q} \Delta n_{2,i} \quad \text{mit}$$

$$\Delta n_{2,i} = \begin{cases} 1 & \text{für} \quad n_{2,i+1} > n_{2,i} \\ 0 & \text{für} \quad n_{2,i+1} = n_{2,i} \\ -1 & \text{für} \quad n_{2,i+1} < n_{2,i} \end{cases}. \tag{7.11}$$

Durch die Unterscheidung der Laufrichtung für innere und äußere Konturanteile werden diese automatisch korrekt verrechnet. Gl. (7.11) orientiert sich an horizontalen Bild*zeilen*. Analog dazu läßt sich eine „vertikale“ Version formulieren [Li91].

Die Fläche kann nur im kontinuierlichen Fall *exakt* über die Kontur bestimmt werden. Bei diskreten Bildfunktionen müßte dazu die Kontur genau zwischen den Pixeln liegen. Berechnet man die Kontur einer Region über den inneren bzw. äußeren Pixelrand, so wird beispielsweise das Flächenäquivalent m_{00} etwas zu klein bzw. etwas zu groß ausfallen. Es

kann auch ein diskreter Algorithmus formuliert werden, der die exakte Momentenberechnung über die Kontur erlaubt [Phi93]. Die Näherung wird im folgenden allerdings in Kauf genommen, da der exakte Algorithmus zu rechenintensiv ist und die Schnelligkeitsvorteile der konturbasierten Momentenberechnung wieder teilweise zunichte macht.

Binäre Konturmomente

Anstatt bei binären Bildfunktionen die Fläche über die Kontur zu approximieren, kann auch das redundante Flächeninnere weggelassen werden, so daß man reine *Kontur*momente über $\mathcal{K}'$ erhält [Bel91]. $\mathcal{K}'$ darf dabei nur die äußeren Konturteile enthalten, da innere Anteile positiv verrechnet würden. Es ergibt sich analog zu Gl. (7.10):

$$m_{pq} = \sum_{(n_1,n_2)\in\mathcal{K}'} n_1^p n_2^q. \tag{7.12}$$

Hierbei ist die Reihenfolge der Konturpixel beliebig, weshalb der Ordnungsindex nicht benötigt wird.

7.3.2 Hu-Moment-Invarianten (HMIs)

Die Basis der Hu-Moment-Invarianten (HMIs) bilden die diskreten, zentralisierten und normalisierten Momente nach Gl. (7.6), die wiederum nach den vier unterschiedlichen Methoden entsprechend den Gln. (7.8)–(7.12) berechnet werden können.

Die HMIs werden über die Theorie der sogenannten *algebraischen Invarianten* hergeleitet, die Sonderfälle der *binären algebraischen Formen* dargestellen, die die gewünschten Invarianzen aufweisen [Hu62]. Zusätzlich wird ein Zusammenhang zwischen den algebraischen Invarianten I_{uv} und den normierten zentralen Momenten ausgenutzt (s. Anh. A.1). Für die Momente bis zur 2. Ordnung ergeben sich beispielsweise die folgenden komplexen Gleichungen:

$$I_{20} = \mu_{20}^{\mathrm{N}} - \mu_{02}^{\mathrm{N}} - 2j\mu_{11}^{\mathrm{N}} \quad \text{und} \tag{7.13}$$

$$I_{11} = \mu_{20}^{\mathrm{N}} + \mu_{02}^{\mathrm{N}}. \tag{7.14}$$

Die HMIs werden dann im allgemeinen als nichtlineare, gewichtete Kombinationen der algebraischen Invarianten I_{uv} gebildet. Die Kombinationen werden so gewählt, daß die sich ergebenden HMIs reell und unabhängig voneinander sind. Ab der 5. Ordnung läßt sich ein systematisches Bildungsgesetz angeben, mit dem HMIs beliebig großer Ordnungen erzeugt werden können (s. Anh. A.1).

Wegen der Eigenschaften der zugrundeliegenden algebraischen Invarianten sind die hier verwendeten HMIs auch ohne Zuhilfenahme der Hauptorientierung (vgl. Gl. (7.7)) translations-, skalierungs-, rotations- und spiegelungsinvariant. Die HMIs 2. Ordnung haben beispielsweise folgende Gestalt:

$$H_1 = I_{11} = \mu_{20}^{\mathrm{N}} + \mu_{02}^{\mathrm{N}} \quad \text{und} \tag{7.15}$$

$$H_2 = I_{20}I_{02} = I_{20}I_{20}^{*} = \left(\mu_{20}^{\mathrm{N}} - \mu_{02}^{\mathrm{N}}\right)^2 + 4\left(\mu_{11}^{\mathrm{N}}\right)^2. \tag{7.16}$$

Mit diesen HMIs 2. Ordnung werden Ausdehnung und Hauptorientierung der Bildfunktion charakterisiert. Es läßt sich zeigen, daß eine Rekonstruktion der Bildfunktion aus den HMIs 2. Ordnung eine Bildellipse mit konstanter „Dicke“ und einem Schwerpunkt ergibt, der mit dem Schwerpunkt der Bildfunktion übereinstimmt [Tea80].

7.3.3 Zernike-Moment-Invarianten (ZMIs)

Das Hauptproblem bei Moment-Invarianten ist immer die Rotationsinvarianz. HMIs sind rotationsinvariant, weil die Invarianzeigenschaften der zugrundeliegenden algebraischen Polynome ausgenutzt werden. Die sogenannten *Zernike-Moment-Invarianten* (ZMIs) basieren auf den aus der Optik bekannten Zernike-Polynomen $V_{nl}(x,y)$. Diese komplexen Polynome sind innerhalb des Einheitskreises zueinander paarweise orthogonal und lassen sich in Polarkoordinaten (ρ,θ) als:

$$V_{nl}(x,y) = R_{nl}(\rho)\exp(jl\theta) \tag{7.17}$$

schreiben [Tea80]: die Phase eines Zernike-Polynoms ändert sich also linear mit dem Winkel. Durch Entwicklung der Bildfunktion $f(x,y)$ nach diesen Zernike-Polynomen lassen sich im Einheitskreis *Zernike-Momente* der Form

$$A_{nl} = \frac{n+1}{\pi}\iint f(x,y)V_{nl}^{*}(x,y)\mathrm{d}x\mathrm{d}y \tag{7.18}$$

berechnen [Tea80]. Es kann ein Zusammenhang zwischen den Zernike-Momenten und den normalisierten Zentralmomenten hergestellt werden (s. Anh. A.2). Bis zur 2. Ordnung ergeben sich beispielsweise die folgenden beiden Zernike-Momente:

$$A_{22} = 3/\pi \cdot \left[\mu_{02}^{\mathrm{N}} - \mu_{20}^{\mathrm{N}} - 2j\mu_{11}^{\mathrm{N}}\right] \quad \text{und} \tag{7.19}$$

$$A_{20} = 3/\pi \cdot \left[2\left(\mu_{20}^{\mathrm{N}} + \mu_{02}^{\mathrm{N}}\right) - 1\right]. \tag{7.20}$$

Es läßt sich nachweisen, daß sich eine Rotationstransformation der Bildfunktion lediglich linear in der Phase eines Zernike-Momentes auswirkt, was eine Konsequenz der linearen Phaseneigenschaften der Zernike-Polynome ist. Diese immanente Rotationsinvarianz des Betrages der Zernike-Polynome läßt sich zum Bilden von ZMIs ausnutzen, indem zueinander konjugiert komplexe Zernike-Momente so miteinander multipliziert werden, daß sich die Phase und somit der rotationsabhängige Anteil aufhebt; gleichzeitig werden ZMIs dadurch reell. Die Ordnung der Zernike-Momente richtet sich nach der Ordnung der zugrundeliegenden normierten Zentralmomente. Die beiden ZMIs 2. Ordnung ergeben sich dann zu:

$$S_1 = A_{20} = 3/\pi \cdot \left[2\left(\mu_{20}^{\mathrm{N}} + \mu_{02}^{\mathrm{N}}\right) - 1\right] \quad \text{und} \tag{7.21}$$

$$S_2 = A_{22}A_{22}^{*} = 9/\pi^2 \cdot \left[\left(\mu_{20}^{\mathrm{N}} - \mu_{02}^{\mathrm{N}}\right)^2 + 4\left(\mu_{11}^{\mathrm{N}}\right)^2\right] \tag{7.22}$$

(weitere ZMIs s. Anh. A.2). Durch Vergleich mit den HMIs der 2. Ordnung in Gl. (7.15) und (7.16) erkennt man, daß S_1 durch eine Verschiebung und eine Skalierung aus H_1 hervorgeht. S_2 unterscheidet sich lediglich durch einen Vorfaktor von H_2. Die ersten beiden Moment-Invarianten der 3. Ordnung stimmen ebenfalls bis auf die Skalierung miteinander überein. Alle weiteren ZMIs unterscheiden sich jedoch von den HMIs wesentlich. ZMIs weisen die gleichen Invarianzen auf wie HMIs.

7.3.4 Bildung von Merkmalsvektoren mit Moment-Invarianten

Sowohl die HMIs als auch die ZMIs sind translations-, skalierungs-, rotations- und spiegelungsinvariant. Mit den Moment-Invarianten läßt sich daher die Gestalt von Handregionen sehr gut beschreiben.

Allerdings ist für eine Geste auch die *Trajektorie* des Schwerpunktes von großer Bedeutung. In [Cam95] und [Cam96] werden für die Bewegungsdarstellung beispielsweise ausschließlich Trajektoriendaten verwendet, die allerdings in drei Dimensionen durch Sensoren bzw. Stereobilder gewonnen werden. Diese Trajektorienbewegung wurde jedoch bei den MIs zur Herstellung der Translationsinvarianz herausnormiert. Da nur eine Kamera benutzt wird, kann die Bewegungskomponente in Richtung der Kameraachse nur indirekt durch Größenänderungen erfaßt werden. Durch die Skalierungsinvarianz entfällt auch diese Komponente. Damit die Trajektorie dennoch berücksichtigt werden kann, werden die Fläche A_t und die Schwerpunktskoordinaten $(\bar{n}_{1,t}, \bar{n}_{2,t})$ als zusätzliche Komponenten zur Bildung eines Merkmalsvektors herangezogen. Das sind genau die Größen, die zur Berechnung der normierten Zentralmomente verwendet werden (vgl. Gln. (7.9) und (7.6) in Kap. 7.3.1). Zusammen mit den Moment-Invarianten $M_{i,t}$ bis zur Ordnung O und der Anzahl N_O, ergibt sich dann folgender Merkmalsvektor:

$$\mathbf{x}_t = [A_t, \bar{n}_{1,t}, \bar{n}_{2,t}, M_{1,t}, M_{2,t}, \ldots, M_{N_O,t}]^{\mathrm{T}} . \tag{7.23}$$

Für $M_{i,t}$ müssen entsprechend die HMIs $H_{i,t}$ oder die ZMIs $S_{i,t}$ eingesetzt werden (s. Anh. A.1 und A.2). Die absolute Orts- und Flächeninformation ist natürlich unerwünscht: sie würde voraussetzen, daß Gesten immer absolut dieselbe Trajektorie verfolgen. Es hat sich daher als sinnvoll erwiesen, auf die zeitlichen Änderungen $\Delta A_t = A_t - A_{t-1}$ und $(\Delta\bar{n}_{1,t}, \Delta, \bar{n}_{2,t}) = (\bar{n}_{1,t} - \bar{n}_{1,t-1}, \bar{n}_{2,t} - \bar{n}_{2,t-1})$ dieser Merkmale überzugehen. Dadurch bleibt die Charakteristik der Trajektorie in Form von relativen Änderungen erhalten, die störende absolute Information ist jedoch nicht mehr vorhanden. Der Merkmalsvektor wird damit zu:

$$\mathbf{x}_t = [\Delta A_t, \Delta\bar{n}_{1,t}, \Delta\bar{n}_{2,t}, M_{1,t}, M_{2,t}, \ldots, M_{N_O,t}]^{\mathrm{T}} . \tag{7.24}$$

Unter Umständen kann es hilfreich sein, auch die zeitliche Änderung der Moment-Invarianten-Komponenten zu berücksichtigen. Damit werden die Bilder einer Sequenz über die Merkmale zusätzlich zeitlich miteinander verknüpft. Dieser Vektor hat dann die Gestalt [Mor98b, Mor98c, Mor99]:

$$\mathbf{x}_t = [\Delta A_t, \Delta\bar{n}_{1,t}, \Delta\bar{n}_{2,t}, M_{1,t}, M_{2,t}, \ldots, M_{N_O,t}, \Delta M_{1,t}, \Delta M_{2,t}, \ldots, \Delta M_{N_O,t}]^{\mathrm{T}} . \tag{7.25}$$

Ob Merkmalsvektoren nach Gl. (7.24) oder Gl. (7.25) günstiger sind und welche Dimension gewählt werden muß, kann nur durch die Evaluierungen in Kap. 8 entschieden werden. Die Merkmalsvektoren nach Gl. (7.23) werden dagegen aus den oben genannten Gründen nicht eingesetzt.

7.3.5 Merkmalsvektoren aus Bildstreifen

Es ist naheliegend, die Merkmalsinformation direkt aus der Pixelinformation zu gewinnen. Da die Anzahl der Pixel pro Bild viel zu groß ist, ist es ein gängiges Verfahren, das Bild gröber zu rastern. Dieses Verfahren wurde in der Bildsequenzmodellierung beispielsweise in [Yam92, Rig96, Sch96b] verwendet (ebenfalls in [Mor97a] zu Vergleichszwecken). Auch in der Einzelbildklassifikation ist es verbreitet (s. beispielsweise [Nef98]).

Die Darstellung der daraus resultierenden sog. *Streifenmethode* soll formal so erfolgen, daß die Parallelen zum Verfahren der *Bildvektoren* in Kap. 7.3.6 leichter ersichtlich werden: Die Verminderung der Auflösung der Bildfunktion $f(n_1, n_2)$ kann über die Mittelpunkte

$\mathbf{v}_{ij}$ eines Rasters $K \times L$ definiert werden, das äquidistant über die Bildfunktion gelegt wird

$$\mathbf{v}_{ij} = \begin{bmatrix} n_{1,i} \\ n_{2,j} \end{bmatrix} = \begin{bmatrix} \Delta n_1 \cdot (i - 1/2) \\ \Delta n_2 \cdot (j - 1/2) \end{bmatrix}, \quad i = 1, \ldots, K, \quad j = 1, \ldots, L. \tag{7.26}$$

Bei der Auflösung der Bildfunktion von $N_1 \times N_2$ werden die Rasterintervalle zu $\Delta n_1 = N_1/K$ und $\Delta n_2 = N_2/L$. Die Rastergebiete N_{ij} sind dann implizit mit der Nächsten-Nachbar-Regel und dem euklidischen Abstandsmaß d[.,.] definiert:

$$N_{ij} = \{\mathbf{n} | \mathrm{d}[\mathbf{n}, \mathbf{v}_{ij}] < \mathrm{d}[\mathbf{n}, \mathbf{v}_{kl}] \quad \text{für alle } k, l \text{ mit } k \neq i \text{ oder } l \neq j\}. \tag{7.27}$$

In jedem Gebiet N_{ij} wird dann der mittlere Grauwert

$$n_{ij} = \frac{1}{|N_{ij}|} \sum_{\mathbf{n} \in N_{ij}} f(\mathbf{n}). \tag{7.28}$$

berechnet. Da alle Rastergebiete N_{ij} gleich groß sind, kann auch auf die Normierung auf die Anzahl der Pixel $|N_{ij}|$ in einem Rastergebiet verzichtet werden (vgl. mit oberem Teilbild in Bild 7.3 Seite 71).

Merkmalsvektoren lassen sich nun dadurch bilden, daß die mittleren Grauwerte n_{ij} der Bildfunktion vertikal zu

$$\mathbf{x}_i^{\mathrm{V}} = [n_{i1}, n_{i2}, \ldots, n_{iL}]^{\mathrm{T}}, \quad i = 1, \ldots, K \tag{7.29}$$

oder horizontal zu

$$\mathbf{x}_j^{\mathrm{H}} = [n_{1j}, n_{2j}, \ldots, n_{Kj}]^{\mathrm{T}}, \quad j = 1, \ldots, L \tag{7.30}$$

formiert werden. Mit einer vertikalen oder horizontalen Serie von Merkmalsvektoren $\mathbf{x}_i^{\mathrm{V}}$ oder $\mathbf{x}_j^{\mathrm{H}}$ wird ein Bild also in neben- oder übereinanderliegende *Streifen* zerlegt. Die einzelnen Vektoren eines Bildes werden dann mit der jeweils in Gl. (7.29) oder (7.30) angegebenen Reihenfolge zeitlich nacheinander emittiert. Entsprechend Gl. (7.1) werden die Vektoren aufeinanderfolgender Bilder zu einer Gesamtsequenz zusammengesetzt. Darin ist somit eine gleichbleibende Anzahl von K vertikalen bzw. L horizontalen Streifen pro Bild enthalten.

Die Bildstreifen-Merkmale sind zwar sehr leicht zu berechnen, sie haben aber zwei grundsätzliche Nachteile:

- Um die Invarianz gegenüber Translation und Rotation zu erreichen, könnten zwar Schwerpunkt und Hauptorientierung nach den Gln. (7.9) und (7.7) berechnet werden. Allerdings müßte man mit diesen Parametern dann die gesamte Bildfunktion transformieren. Dies ist insbesondere für den Rotationsvorgang eine sehr aufwendige und damit nicht praktikable Operation, weil sie auf jedes Pixel angewendet werden muß. Eine effektive Transformation im Merkmalsraum ist dagegen nicht bekannt.

- Das Verfahren reagiert asymmetrisch auf Bewegungen in der Bildsequenz. Verwendet man beispielsweise vertikale Bildstreifen, so wirken sich horizontale Bewegungsschwankungen im wesentlichen auf die zeitliche Merkmalssequenz aus, während vertikale Schwankungen die Merkmalsvektoren selbst verändern.

7.3.6 Bildvektoren als Merkmale

7.3.6.1 Repräsentation der Bildfunktion mit Bildvektoren

Der Grundgedanke dieses neuen Verfahrens ist es, die *Ortskoordinaten* der Bildpixel *direkt* als zweidimensionale Merkmale zu verwenden [Mor97a, Mor97b]. Da die Anzahl der Pixel pro Bild dafür jedoch viel zu groß ist, muß zunächst eine datenreduzierte Version der Bildfunktion errechnet werden. Hierfür werden die Pixel bestimmter Bildbereiche durch sog. *Bildvektoren* $\mathbf{v}_{ij}$, die jeweils mit einer bestimmten Anzahl von *Attributen* verknüpft sind, repräsentiert.

Zur Initialisierung werden die Bildvektoren $\mathbf{v}_{ij}^{\text{init}}$ zunächst in einem regelmäßigen Raster $K \times L$ über die Bildfunktion $f(\mathbf{n})$ gelegt. Dieses Initialisierungsraster ist mit den Mittelpunkten der Rasterbereiche der Bildstreifen identisch und mit Gl. (7.26) beschrieben. Implizit definiert jeder Bildvektor wiederum eine sog. *Nachbarschaft* N_{ij}, die über die Nächste-Nachbar-Regel mit dem euklidischen Abstandsmaß d[.,.] mit der Gl. (7.27) gebildet wird.

Jedem Bildvektor werden bis zu zwei Attribute zugeordnet, die abhängig von der Art der Bildfunktion sind: in dieser Arbeit wurden Grauwertbildfunktionen $f(\mathbf{n})$ und Gradientenbildfunktionen — zerlegt in Betrag $f_{\text{g}}(\mathbf{n})$ und Orientierung $\delta_{\text{g}}(\mathbf{n})$ — untersucht (Nomenklatur s. Kap. 5.2). Das erste Attribut n_{ij}^{init} jedes Bildvektors $\mathbf{v}_{ij}^{\text{init}}$ enthält den mittleren Grauwert bzw. den mittleren Betrag des Gradienten der jeweiligen initialen Nachbarschaft N_{ij}^{init} und wird mit Gl. (7.28) definiert. Analog dazu wird bei der Verwendung von Gradientenbildern ein zweites Attribut d_{ij}^{init} bestimmt, das die mittlere Orientierung der jeweiligen Nachbarschaft enthält:

$$d_{ij} = \frac{1}{|N_{ij}|} \sum_{\mathbf{n} \in N_{ij}} \delta_{\text{g}}(\mathbf{n}). \tag{7.31}$$

Weitere, hier nicht untersuchte Attribute könnten beispielsweise Farb- oder Texturinformation beinhalten.

Nach der Initialisierung enthalten die Attribute der Bildvektoren fast die gesamte Bildinformation, da die Bildvektoren regelmäßig angeordnet sind. Um mehr Information in den individuellen *Ort* der Bildvektoren zu verlagern, muß deren Lage optimal an das Bild angepaßt werden. Dazu werden zuerst alle Bildvektoren $\mathbf{v}_{ij}^{\text{init}}$ gelöscht, deren Grauwert-Attribut n_{ij}^{init} gleich Null oder kleiner einer bestimmten Schwelle ist. Die Lage der übrigbleibenden Bildvektoren $\mathbf{v}_{ij}^{\text{init}'}$ wird dann iterativ mit einem abgewandelten *k-means*-Algorithmus aus der Vektorquantisierung an die Bildfunktion angepaßt (s. in [Hua90], vgl. mit Initialisierung des Codebuchs in Kap. 6.3.2.1). Während bei der Vektorquantisierung jedoch Cluster bei prinzipiell *beliebig* angeordneten Merkmalsvektoren gefunden werden sollen, besteht hier die Aufgabe, eine optimale Bildvektor-Repräsentation der *regelmäßig* angeordneten Bildpixel unter Berücksichtigung beliebig verteilter Grauwerte oder Gradienten zu finden.

Der Iterationsschritt in der Berechnung der optimalen Bildvektoren $\mathbf{v}_{ij}^{\text{opt}}$ besteht in der Berechnung neuer Bildvektoren $\hat{\mathbf{v}}_{ij}$ als die jeweiligen Schwerpunkte der alten Nachbarschaften N_{ij}. Das Orientierungsattribut aus Gl. (7.31) wird dabei nur bei Gradientenbildern eingesetzt und kann dann die Iteration unter Zuhilfenahme von Gl. (7.38) beschleunigen. Der komplette Optimierungs-Algorithmus hat folgende Gestalt [Mor97a]:

1. **Initialisierung:**

$$\mathbf{v}_{ij} = \mathbf{v}_{ij}^{\text{init}'} \quad \text{nach Gl. (7.26), wodurch implizit} \tag{7.32}$$
$$n_{ij} = n_{ij}^{\text{init}'} \quad \text{und} \tag{7.33}$$
$$d_{ij} = d_{ij}^{\text{init}'} \quad \text{aus den Gln. (7.27), (7.28) und (7.31) folgen.} \tag{7.34}$$

2. **Iterationsschritt:**

$$\hat{\mathbf{v}}_{ij} = \frac{1}{\sum_{\mathbf{n}\in N_{ij}} f(\mathbf{n})} \sum_{\mathbf{n}\in N_{ij}} \mathbf{n} \cdot f(\mathbf{n}), \tag{7.35}$$

   woraus sich wiederum neue $\hat{N}_{ij}$, $\hat{n}_{ij}$ und eventuell $\hat{d}_{ij}$ ergeben (bei Gradientenbildern ist $f(\mathbf{n})$ durch $f_{\text{g}}(\mathbf{n})$ zu ersetzen).

3. **Abbruchkriterium:** solange $\mathrm{d}\left[\hat{\mathbf{v}}_{ij}, \mathbf{v}_{ij}\right] > \epsilon$ für alle i, j, wiederhole Schritt 2, sonst gehe zu Schritt 4.

4. **Resultat:** die optimalen Bildvektoren und das optimale Grauwert-Attribut sind das Ergebnis des letzten Iterationsschrittes:

$$\mathbf{v}_{ij}^{\text{opt}} = \hat{\mathbf{v}}_{ij} \quad \text{und} \tag{7.36}$$
$$n_{ij}^{\text{opt}} = \hat{n}_{ij}. \tag{7.37}$$

Werden Gradientenbilder betrachtet, so konvergiert die Iteration schneller, wenn das Orientierungsattribut d_{ij} verwendet wird, um nur die Kantenpixel $f_{\text{g}}(\mathbf{n})$ zur Neuberechnung der Bildvektoren zuzulassen, deren Orientierung $\delta_{\text{g}}(\mathbf{n})$ weniger als eine Schwelle Δd von d_{ij} abweicht [Mor97b]. Dadurch ändern sich die Summationsgebiete in Gl. (7.35) zu:

$$\mathbf{n} \in N_{ij} \quad \text{und} \quad |\delta_{\text{g}}(\mathbf{n}) - d_{ij}| \leq \Delta d, \tag{7.38}$$

während die in den Gln. (7.28) und (7.31) unverändert bleiben. Die Differenz der Orientierungswinkel ist definitionsgemäß kleiner oder gleich ± 180 Grad.

Am Ende des Iterationsprozesses konzentrieren sich die Bildvektoren in Gebieten großer Grauwerte bzw. in der Nähe von Kanten. Der Optimierungsvorgang ist beispielhaft in Bild 7.3 an einem Grauwertbild dargestellt.

7.3.6.2 Bildung der Merkmalssequenzen aus Bildvektoren

Aus einem einzelnen Bild wird zunächst eine Merkmalssequenz, indem nacheinander alle Bildvektoren emittiert werden, deren Grauwert-Attribut nicht Null ist. Das bedeutet, daß es sich immer um *zweidimensionale* Merkmalsvektoren handelt: die Komponenten der Merkmalsvektoren werden aus Pixelkoordinaten gebildet. Die Reihenfolge der Emission ist *beliebig*, da der Trainingsalgorithmus für die HMMs — anschaulich gesprochen — die Merkmale für eines oder mehrere Bilder sammelt, um daraus in den jeweiligen Zuständen eine WDF-Repräsentation der eintreffenden Merkmalsvektoren zu schätzen: in eine WDF geht jedoch die Reihenfolge der Merkmale nicht ein. Man erhält die komplette Merkmalssequenz einer Bildsequenz, indem man die Merkmalssequenzen aufeinanderfolgender Bilder hintereinander anordnet.

Wegen der Verwendung der zweidimensionalen Bildvektoren als Merkmale, ist der Merkmalsraum ebenfalls zweidimensional und entspricht in seinen Dimensionen $N_1 \times N_2$

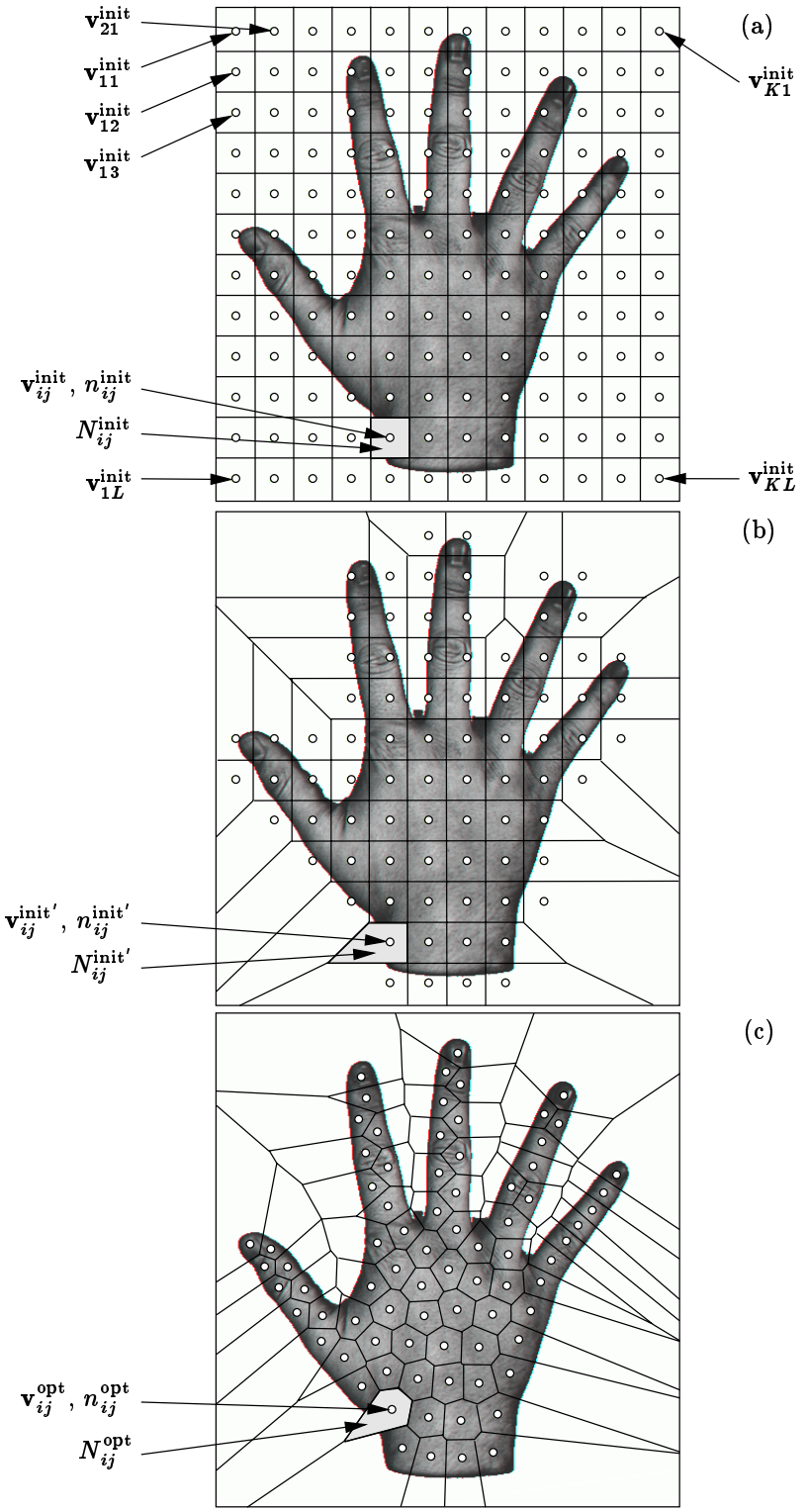


Bild 7.3: Phasen der Optimierung von Bildvektoren am Beispielbild: (a) Initialisierung, (b) Löschen der Bildvektoren mit verschwindendem Grauwert-Attribut, (c) Optimierung

den Abmessungen der Bildfunktionen. Als Besonderheit des Bildvektor-Verfahrens bilden sich somit in den Zuständen der HMMs sog. *Bilddichtefunktionen* (BDFs) aus: dies sind spezielle WDFs, die in ihrem Definitionsbereich den zugrundeliegenden Bildfunktion entsprechen, wobei gleichzeitig die Grauwerte bzw. die Gradientenbeträge der Bildfunk-

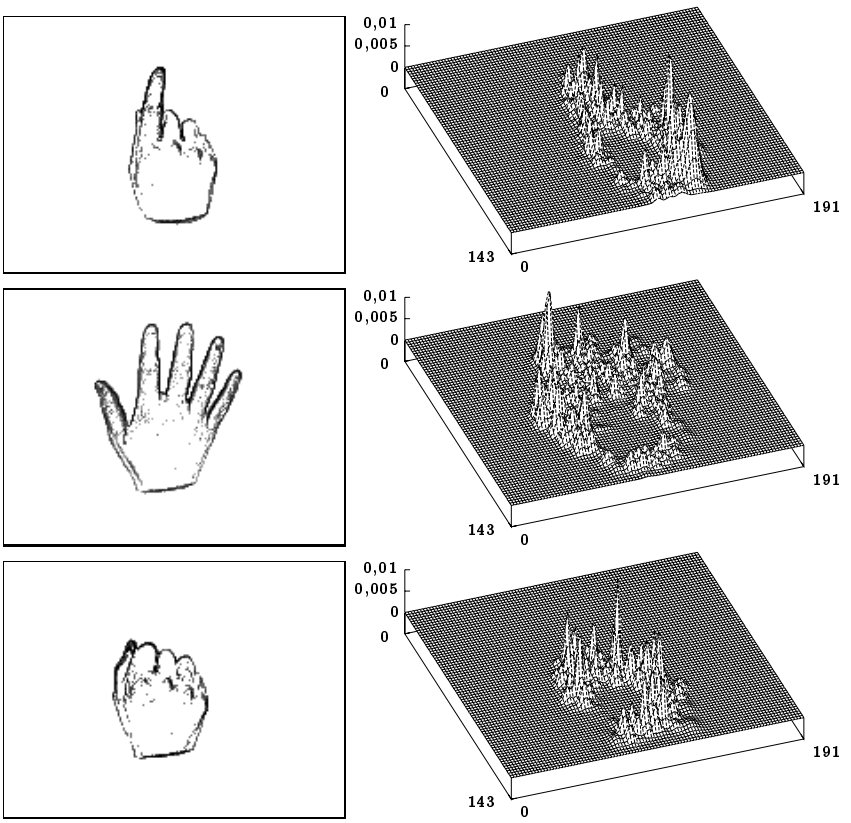


Bild 7.4: Beispiele typischer Bilder aus Gestenbildsequenzen und Gegenüberstellung mit sich ausbildenden Bilddichtefunktionen in den Zuständen der HMMs bei Modellierung mit Bildvektoren auf der Grundlage von Gradientenbildern

tionen als Wahrscheinlichkeitsdichtewerte abgebildet werden [Mor97a, Mor97b] (Beispiele s. Bild 7.4). Die genaue Ausprägung der BDFs ist abhängig von der Strategie, mit der die Bildvektoren als Merkmale emittiert werden:

1. **Wiederholte Emission der initialen Bildvektoren:** Bei dieser Wiederholsequenz werden die regelmäßig angeordneten, nicht optimierten Bildvektoren $\mathbf{v}_{ij}^{\text{init}'}$ emittiert. Da diese im initialen Stadium nur wenig Bildinformation enthalten, werden sie mit einer Anzahl c_{ij} wiederholt, die proportional zum zugeordneten Grauwert-Attribut $n_{ij}^{\text{init}'}$ ist.

   Der Trainingsalgorithmus eines HMMs „beobachtet“, daß eine Emission im Bereich eines Bildvektors $\mathbf{v}_{ij}^{\text{init}'}$ c_{ij} Mal vorkommt. Weil aber c_{ij} proportional zum mittleren Grauwert der Nachbarschaft $N_{ij}^{\text{init}'}$ ist, wird der trainierte Wahrscheinlichkeitsdichtewert ebenfalls proportional zu diesem mittleren Grauwert sein. Auf diese Art werden aufeinanderfolgende, gleichartige Bilder einer Sequenz in den Zuständen eines HMMs als BDFs abgebildet. Die Anzahl der Wiederholungen eines Bildvektors wird so normiert, daß die Gesamtanzahl der Emissionen für jedes Bild konstant bleibt.

2. **Einmalige Emission der optimierten Bildvektoren:** Für die Bildung dieser Optimalsequenzen werden die optimierten Bildvektoren $\mathbf{v}_{ij}^{\text{opt}}$ verwendet. Jeder Vektor wird genau einmal emittiert.

Weil sich die optimierten Bildvektoren in der Nähe größerer Grauwerte sammeln, wird das HMM-Training dort mehr Vektoren sammeln, was wiederum zu höheren Dichtewerten führt. Somit entstehen auch aus solchen Sequenzen BDFs.

3. **Wiederholte Emission der optimierten Bildvektoren:** Bei diesen Optimal-Wiederholsequenzen handelt es sich um eine Kombination der Strategien 1 und 2. Dadurch werden größere Grauwerte bei der Abbildung in BDFs stärker betont.

4. **Einmalige Emission der initialen Bildvektoren:** Mit diesen Kontrollsequenzen aus den Bildvektoren $\mathbf{v}_{ij}^{\text{init}'}$ soll bei der Evaluierung in Kap. 8 die Wirksamkeit der Strategien 1–3 festgestellt werden.

Da die unterschiedlichen Strategien zu unterschiedlich gearteten BDFs führen, ist es wichtig, dieselbe Strategie sowohl beim Training als auch bei der Erkennung anzuwenden.

Durch Bestimmung des Schwerpunktes $(\bar{n}_1, \bar{n}_2)$ und der Hauptorientierung Φ der Bildfunktionen nach Gln. (7.9) und (7.7) lassen sich die Bildvektoren transformieren, so daß sie invariant gegenüber Translation und Rotation werden [Bro81]:

$$\mathbf{v}_{ij}^{\text{N}} = \begin{bmatrix} v_{1,ij}^{\text{N}} \\ v_{2,ij}^{\text{N}} \end{bmatrix} = \begin{bmatrix} \cos\Phi & -\sin\Phi \\ \sin\Phi & \cos\Phi \end{bmatrix} \cdot \begin{bmatrix} v_{1,ij} - \bar{n}_1 \\ v_{2,ij} - \bar{n}_2 \end{bmatrix}. \tag{7.39}$$

Diese Normierung ist aufgrund der geringen Zahl der Bildvektoren pro Bild mit wenig Rechenaufwand durchführbar. Die Normierung erfolgt sinnvollerweise auf das erste Bild einer isolierten Sequenz. Normiert man jedes Bild einer Sequenz für sich, so geht wichtige Information über die Trajektorie einer Bewegung verloren. Mit einer solchen Normierung kann erreicht werden, daß lediglich Gestaltänderungen modelliert werden.

7.4 Bewertung und Vorauswahl der Merkmalsextraktionsverfahren

7.4.1 Vergleich modell- und pixelbasierter Verfahren

Obwohl gezeigt werden konnte, daß die untersuchten modellbasierten Verfahren funktionsfähig sind, lassen sich schon nach den ersten Evaluierungen die Nachteile gegenüber den pixelbasierten Verfahren erkennen:

- Während mit *einem* Augen-Template wenigstens noch etwa 5 Bilder pro Sekunde verarbeitet werden können — wobei aber die Algorithmen in ihrer Implementierung schon auf Geschwindigkeit hin optimiert wurden —, benötigt die Nachführung eines Hand-Modells auf demselben Rechner trotz der 3D-Graphik-Hardware für ein Bild schon mehrere Sekunden[1]. Beide Verfahren — insbesondere die direkte graphische Extraktion — sind also weit von einer Ausführung in Echtzeit entfernt. Da sie aber immer nur kleinen Bildänderungen folgen können, sind sie unbedingt auf eine schnelle Bildwiederholrate und damit auf eine schnelle Verarbeitung *angewiesen*.

  Zwar können pixelbasierte Verfahren unter Umständen ebenfalls nicht in Echtzeit ausgeführt werden. Allerdings kann hier die Bildwiederholrate reduziert werden, ohne daß die Funktionsfähigkeit der Merkmalsextraktion in Frage gestellt wird.

[1] Gemessen auf *Silicon Graphics Indigo2 High Impact* mit 250 MHz Mips R4400 CPU.

- Die modellbasierten Verfahren können prinzipiell das verfolgte Objekt verlieren und sind dann von sich aus nicht mehr in der Lage, die Verfolgung wieder aufzunehmen. Das Objekt kann auch als Folge eines sich akkumulierenden Positionierungsfehlers verloren gehen. Um die Verfahren *robust* gegenüber möglichen Aussetzern zu machen, sind zusätzliche Maßnahmen erforderlich, die aufwendige Entwicklungen voraussetzen.

  Pixelbasierte Verfahren sind dagegen robust, da sie bei jedem Bild neu mit der Extraktion beginnen, so daß sich Fehler in der Extraktion nicht akkumulieren können.

- Obwohl sich nicht genau voraussagen läßt, welche Merkmale für die eigentliche Erkennungsstufe am besten geeignet sind, ist doch anzunehmen, daß sich die zufälligen Positionierungsfehler und mangelnde Robustheit der modellbasierten Verfahren negativ auf die Erkennung auswirken werden.

  Positionierungsfehler können bei pixelbasierten Verfahren prinzipiell nicht auftreten, so daß eine mögliche Fehlerquelle in der Verarbeitungskette von vornherein ausgeschlossen ist.

- Modellbasierte Verfahren sind sehr spezialisiert, da sie sehr genau an das zu verfolgende Objekt angepaßt werden müssen. Bei einer veränderten Aufgabenstellung — wozu unter Umständen schon eine veränderte Aufnahmeperspektive zählt — muß daher stets wieder ein neues Verfahren konzipiert werden.

  Pixelbasierte Verfahren machen keine Annahmen über das zugrundeliegende Objekt, sind daher für viele Anwendungen allgemein verwendbar und müssen in der Regel keine spezielle Anpassung erfahren.

Viele der Nachteile der modellbasierten Verfahren rühren daher, daß sie in Anbetracht der Erkennungsanwendung das zugrundeliegende Bildobjekt genauer als nötig modellieren. So ist eine Parameterextraktion, die eine exakte Rekonstruktion eines dreidimensionalen Objektes ermöglichen soll, für eine „einfache“ Klassifikationsanwendung eigentlich zu aufwendig und kann durch die mangelnde Robustheit sogar zu schlechterer Erkennung führen. Außerdem nehmen die modellbasierten Verfahren einen großen Teil der Modellbildung schon vorweg, den auch die HMMs leisten könnten.

Diese Gegenüberstellung zeigt deutlich, daß die pixelbasierten Verfahren für die geplante Klassifikationsanwendung wesentlich besser als die modellbasierten Verfahren geeignet sind. Die modellbasierten Verfahren werden daher von vornherein von der Evaluierung in Kap. 8 ausgeschlossen.

7.4.2 Bewertung der pixelbasierten Verfahren

Bildvektoren scheinen auf den ersten Blick eine große Ähnlichkeit mit den Bildstreifen aus Kap. 7.3.5 zu haben. Der entscheidende Unterschied besteht aber darin, daß bei Bildstreifen die *Attribute* und bei den Bildvektoren die *Vektoren* selbst emittiert werden. Dadurch ergeben sich für die Bildvektoren im Vergleich zu den Bildstreifen mehrere Vorteile:

- Die Nachbarschaftsbereiche der Bildvektoren passen sich flexibel der Bildinformation an, während das feste Raster der Bildstreifen auch leere Hintergrundbereiche mitmodelliert.

- Bildvektoren sind nach Gl. (7.39) bei Bedarf sehr leicht normierbar, was bei Bildstreifen eine Transformation der gesamten Bildfunktion bedeutet.

- Bildvektoren sind im Unterschied zu Bildstreifen *lokale* Merkmale: sie repräsentieren immer nur einen sehr kleinen Bildbereich und sind räumlich symmetrisch. Damit werden vertikale und horizontale Bewegungskomponenten gleich behandelt: beide räumliche Dimensionen werden gleichartig serialisiert. Bei der Erkennung mit HMMs wird also die gesamte räumlich-zeitliche Struktur eine Bildsequenz konsistent modelliert.

Während es aufgrund der vielen Vorteile der Bildvektoren gegenüber den Bildstreifen gerechtfertigt erscheint, die Bildvektoren als die besseren Merkmale zu bezeichnen, ergibt sich im Vergleich mit den Merkmalsvektoren auf der Basis von Moment-Invarianten ein anderes Bild.

Bildvektoren stellen im Vergleich zu den Merkmalsvektoren auf der Basis von Moment-Invarianten zwei Extreme dar: auf der einen Seite wird ein Bild mit einer Vielzahl von lokalen und einfachen Merkmalsvektoren repräsentiert, und es wird gewissermaßen den HMMs überlassen, diese einfachen Merkmale zu einem Bild „zusammenzusetzen“ und zusammen mit dem zeitlichen Aspekt der Bildsequenz zu modellieren. Auf der anderen Seite stehen komplexere Merkmalsvektoren, die das gesamte Bild auf einmal beinhalten, so daß die HMMs nur noch den zeitlichen Aspekt der Bildsequenz modellieren können.

Welche Philosophie sich für die Klassifikation von Bildsequenzen am besten eignet, muß die nun folgende Evaluierung in Kap. 8 zeigen.

Kapitel 8

Evaluierung der Erkennung isolierter Gesten

8.1 Motivation für die Evaluierung der isolierten Erkennung

In diesem Kapitel werden Ergebnisse präsentiert, die sich aus der *isolierten Erkennung* von Gesten ergeben. Die isolierte Erkennung setzt voraus, daß die zeitlichen Grenzen der Gesten bekannt sind: es handelt sich somit um eine reine Klassifikationsaufgabe. Diese Voraussetzung ist bei einem praktisch einsetzbaren System für den visuellen Dialog (s. Bild 1.1 auf Seite 6) nicht gegeben. Ein solches System benötigt eine *kontinuierliche* Erkennung, die die Segmentgrenzen der Gesten selbsttätig bestimmt und dann innerhalb diese Segmentgrenzen die Bewegung klassifizieren kann (s. Kap. 9).

Allerdings läßt sich nur durch die isolierte Erkennung feststellen, wie leistungsfähig das Teilsystem aus Merkmalsextraktion und Klassifikation wirklich ist: eine zusätzliche zeitliche Segmentierung erhöht nur die möglichen Fehlerquellen. Fehler der räumlichen Segmentierung können dagegen praktisch ausgeschlossen werden, da für alle Aufnahmen von Trainings- und Testdaten ein homogener, schwarzer Hintergrund verwendet wurde (vgl. Segmentierungsfehler in Tabelle 5.5 auf Seite 44 und Angaben zu den Daten in Anh. C).

So soll durch die Evaluierungen in diesem Kapitel zunächst einmal geklärt werden, ob die in Kap. 6 vorgestellten semikontinuierlichen HMMs mit diagonalisierten Kovarianzmatrizen in Verbindung mit der HMM-Vorverarbeitung der Aufgabenstellung der Erkennung komplexer menschlicher Bewegungen gewachsen sind. Gleichzeitig soll festgestellt werden, welches der in Kap. 7 vorgestellten Merkmalsextraktionsverfahren am besten für die Modellierung mit HMMs geeignet ist. Ebenso lassen sich die Einflüsse weiterer Parameter, wie sie insbesondere für den praktischen Betrieb eines Gestikerkennungssystems von Bedeutung sind, auf den Erkennungsprozeß genauer untersuchen.

8.2 Trainings- und Testmaterial

Die meisten Evaluierungen werden mit dem Übungsdatensatz (s. Anh. C.1.1) durchgeführt, der im ersten Teil der zweiten Versuchsreihe der Usability-Experimente gewonnen wurde (s. Kap. 4.2.2). Der spezielle Vergleich der Bildstreifen mit den Bildvektoren muß

mit dem Analysedatensatz (s. Anh. C.1.3) geschehen, da nur er für die Evaluierung mit den nicht normierbaren Bildstreifen in Frage kommt (vgl. Kap. 8.5.2). Für den Demonstrator, der in Kap. 11 beschrieben wird, wurde ein eigener Demonstratordatensatz konzipiert und aufgenommen (s. Anh. C.1.4), der in Kap. 8.6.4 zu Vergleichszwecken herangezogen wird.

Übungs- und Demonstratordatensatz enthalten alle 41 Gesten des Hauptkataloges (s. Anh. B.3.1), der sich nach Auswertung der ersten Usability-Versuchsreihe als sinnvoll für die Steuerung des 3D-Szenen-Editors herausgestellt hat (s. Kap. 4.2.1). Für den Analysedatensatz wurde, gestützt auf den Hauptkatalog, ein Nebenkatalog mit 12 Gesten gebildet (s. Anh. B.3.2).

Während die Gesten aus dem Analysedatensatz isoliert aufgenommen wurden, liegt bei den anderen Datensätzen für jede Person eine kontinuierliche Aufnahme vor, in der alle Gesten auf einmal enthalten sind und zwar in genau der Reihenfolge, mit der sie auch bei der Aufnahme gezeigt wurden. Die kontinuierlichen Daten mußten für das Training und die isolierte Erkennung manuell *gelabelt* werden. Dieser Labelvorgang wird in Anh. C.2 näher beschrieben.

8.3 Ablauf von Training und Erkennung

Ziel der *Trainings* ist es, mit Hilfe des Trainingsdatensatzes $\mathbf{X}'^{\mathrm{Tr}}$ für jede Geste des zugrundeliegenden Katalogs *ein* Hidden-Markov-Modell zu bilden, das die jeweilige Geste repräsentiert. Die *Erkennung* wird evaluiert, indem die Modelle mit dem Erkennungsdatensatz $\mathbf{X}'^{\mathrm{Er}}$ beaufschlagt werden. Die Evaluierungskriterien werden in Kap. 8.4 besprochen. Die Daten für Training und Erkennung sind für alle Evaluierungen dieses Kapitels strikt getrennt. Genaue Angaben zu den Datensätzen und insbesondere die jeweilige Aufteilung in Trainings- und Erkennungsmengen sind in Anh. C.1 zu finden.

Das Training setzt sich aus drei einzelnen Schritten zusammen (vgl. Ausführungen in Kap. 6.3 und 6.4):

1. Mit dem gesamten originalen Trainingsdatensatz $\mathbf{X}'^{\mathrm{Tr}}$ werden Mittelwertsvektor $\boldsymbol{\mu}^{\mathrm{Tr}}$ und Varianzvektor $\boldsymbol{\sigma}^{\mathrm{Tr}}$ für die HMM-Vorverarbeitung bestimmt (s. Kap. 6.4). Mit Gl. (6.35) in Kap. 6.4 können dann die transformierten Versionen $\mathbf{X}^{\mathrm{Tr}}$ und $\mathbf{X}^{\mathrm{Er}}$ der Datensätze berechnet werden, so daß sie für die folgenden Schritte zur Verfügung stehen.

2. Mit dem kompletten Trainingsdatensatz $\mathbf{X}^{\mathrm{Tr}}$ wird dann das initiale Codebuch $\{\dot{\mathbf{M}}, \dot{\mathbf{\Sigma}}\}$ bestimmt, das allen Modellen gemeinsam ist (s. Kap. 6.3.2.1).

3. Der Trainingsdatensatz wird dann in spezifische Datensätze $\mathbf{X}^{\mathrm{Tr},\lambda_l}$ aufgeteilt, die jeweils zu einer Geste g_l gehören, so daß damit das Modell $\lambda_l(\boldsymbol{\pi}^{\lambda_l}, \mathbf{A}^{\lambda_l}, \mathbf{C}^{\lambda_l}, \mathbf{M}, \mathbf{\Sigma})$ trainiert werden kann. Zusammen mit dem Training *aller* Modelle entsteht auch die nachgeschätzte Version des Codebuchs $\{\mathbf{M}, \mathbf{\Sigma}\}$ (s. Kap. 6.3.4).

Der Erkennungsdatensatz $\mathbf{X}^{\mathrm{Er}}$ besteht aus V^{g_l} Versionen von Merkmalsvektorsequenzen $\mathbf{X}_v^{\mathrm{Er},g_l}$, $v = 1, \ldots, V^{g_l}$ der verschiedenen Gesten g_l:

$$\bigcup_{l=1}^{M} \bigcup_{v=1}^{V^{g_l}} \mathbf{X}_v^{\mathrm{Er},g_l} = \mathbf{X}^{\mathrm{Er}}. \tag{8.1}$$

Die Erkennung liefert nach Gl. (6.34) in Kap. 6.3.5 zu diesem Datensatz den Index des Modells $\lambda_v^{\mathrm{Er},g_l}$ mit dem besten Score. Dieser Index wird nun entsprechend den im nächsten Unterkapitel definierten Evaluierungsgrößen ausgewertet. Die Erkennung ist korrekt, wenn

$$\lambda_v^{\mathrm{Er},g_l} = g_l \tag{8.2}$$

ist, ansonsten ist sie falsch.

8.4 Evaluierungskriterien

Die Kriterien für die Evaluierung lassen sich in *direkte* und *indirekte* Kriterien einteilen. Für die direkten Kriterien werden quantitative Meßgrößen definiert. Die indirekten Kriterien können dagegen nur in ihrer Auswirkung auf die quantitativen Meßgrößen beurteilt werden; dabei ist es mitunter schwierig, die Auswirkung eines solchen Kriteriums getrennt von anderen zu beurteilen. Die beiden hier verwendeten direkten Kriterien sind:

Erkennungrate (E1): Die Erkennungsrate r ist das Verhältnis der nach Gl. (8.2) korrekt klassifizierten Bildsequenzen $\mathbf{X}_v^{\mathrm{Er},g_l}$ bezogen auf die Gesamtzahl der Gesten im Datensatz $|\mathbf{X}^{\mathrm{Er}}|$:

$$r = \frac{1}{|\mathbf{X}^{\mathrm{Er}}|} \sum_{l=1}^{M} \sum_{v=1}^{V^{g_l}} \delta\left(\lambda_v^{\mathrm{Er},g_l} - g_l\right). \tag{8.3}$$

Die Erkennungsrate ist das wichtigste Evaluierungskriterium. Sie wird in % angegeben.

Erkennungssicherheit (E2): Ein Maß für die Erkennungssicherheit s ist nicht so offensichtlich. In dieser Arbeit wird davon ausgegangen, daß eine Erkennung um so sicherer ist, je größer der Abstand des besten Scores $\max_l \tilde{F}(\mathbf{X}_v^{\mathrm{Er},g_l}|\lambda_l)$ vom zweitbesten Score $\max'_l \tilde{F}(\mathbf{X}_v^{\mathrm{Er},g_l}|\lambda_l)$ bei einer *korrekten* Klassifikation ist [Stö98, Ste99]. Die Einzelerkennungssicherheit $s_v^{g_l}$ für eine korrekte Klassifikation läßt sich dann als *relatives Abstandsmaß* definieren:

$$s_v^{g_l} = \frac{\exp\left[\max_l \tilde{F}(\mathbf{X}_v^{\mathrm{Er},g_l}|\lambda_l)\right] - \exp\left[\max'_l \tilde{F}(\mathbf{X}_v^{\mathrm{Er},g_l}|\lambda_l)\right]}{\exp\left[\max_l \tilde{F}(\mathbf{X}_v^{\mathrm{Er},g_l}|\lambda_l)\right]}. \tag{8.4}$$

Diese Definition wurde so gewählt, daß $s_v^{g_l}$ in einem Wertebereich zwischen 0 und 1 liegt [Ste99]. Die Erkennungssicherheit s ist dann die Summe der Einzelerkennungssicherheiten $s_v^{g_l}$ für alle korrekten Klassifikationen bezogen auf die Anzahl der korrekt erkannten Gesten $r \cdot |\mathbf{X}^{\mathrm{Er}}|$:

$$s = \frac{1}{r \cdot |\mathbf{X}^{\mathrm{Er}}|} \sum_{l=1}^{M} \sum_{v=1}^{V^{g_l}} s_v^{g_l} \cdot \delta\left(\lambda_v^{\mathrm{Er},g_l} - g_l\right). \tag{8.5}$$

Die Erkennungssicherheit s wird in % angegeben und liegt aufgrund der Eigenschaften von Gl. (8.4) im Bereich 0–100 %. Der *absolute* Wert von s läßt sich nur schwer interpretieren; es lassen sich daher immer nur Aussagen durch Vergleich zweier Erkennungssicherheiten machen. Die Erkennungssicherheit soll in den folgenden Evaluierungen nur dann als Auswahlkriterium herangezogen werden, wenn sich eine Entscheidung aufgrund der Erkennungsrate nicht eindeutig treffen läßt.

Es wurden weitere Meßgrößen, wie die Varianz der gestenspezifischen Erkennungsraten oder der maximale Abstand zwischen kleinster und größter Erkennungsrate, untersucht. Diese Größen korrelierten aber so stark mit der Erkennungsrate, daß sie als gesonderte Größen nicht benötigt wurden [Stö98].

Indirekte Evaluierungskriterien lassen sich nur als qualitative Aussagen formulieren. Ob ein indirektes Kriterium erfüllt werden kann, läßt sich entweder theoretisch voraussagen oder kann indirekt durch Vergleich der quantitativen Kriterien r und s bei speziell ausgewähltem Datenmaterial angegeben werden:

Invarianz (E3): Damit der Benutzer in seiner gestischen Interaktion nicht eingeschränkt ist, muß gewährleistet sein, daß die Gesten in einem beliebigen Bereich des Bildausschnittes mit einer beliebigen Orientierung der Hände und einer variablen Entfernung von der Kamera ausgeführt werden können. Daher muß der Verbund aus Merkmalsextraktion und Erkennung invariant gegenüber Translation, Skalierung und Rotation sein.

Wünschenswert ist auch eine Spiegelungsinvarianz, so daß das System unabhängig von rechts- oder linkshändiger Bedienung wird. Dies kann jedoch unter Umständen auch durch die Verwendung jeweils spezialisierter Modelle erreicht werden.

Echtzeitfähigkeit (E4): Um echtzeitfähig zu sein, müssen die Verarbeitungsschritte bis zur eigentlichen Erkennung schritthaltend ablaufen können. Dies wird um so einfacher, je geringer die Bildwiederholfrequenz des Videostroms für eine zuverlässige Erkennung sein darf. Der Zeitaufwand für die einzelnen Verarbeitungsschritte sinkt gleichzeitig deutlich, wenn die einzelnen Bilder des Videostroms in einer geringeren Auflösung verarbeitet werden können.

Umgebungsunabhängigkeit (E5): Dieses Kriterium soll als erfüllt gelten, wenn sich das System unter normalen Umgebungsbedingungen betreiben läßt. Während ein großer Teil dieser Bedingung im Kapitel über die räumliche Segmentierung geprüft wurde (s. Kap. 5), bleibt im Hinblick auf Merkmalsextraktion und Erkennung die Frage, ob das System auch bei einer normalen Raumbeleuchtung funktioniert: hierbei entstehen Einzelbilder mit Bewegungsunschärfe wie bei den oben erwähnten Demonstratordaten.

Personenunabhängigkeit (E6): Wenn das System in der Lage ist, gestische Eingaben von unterschiedlichen Personen zu erkennen, ohne daß die Modelle neu trainiert werden müssen, gilt es als personenunabhängig. Dabei ist eine Adaption an den jeweiligen Benutzer zulässig, wenn sie automatisch während des Erkennungsvorgangs erfolgt.

Im folgenden soll zunächst das für den Einsatzzweck am besten geeignete *Merkmalsextraktionsverfahren* ausgewählt werden. Dazu werden neben dem Hauptkriterium der Erkennungsrate auch — wenn mit der Erkennungsrate keine eindeutige Entscheidungsfindung möglich ist — die Erkennungssicherheit, die Invarianz und die prinzipielle Echtzeitfähigkeit herangezogen. Für das optimale Verfahren wird dann untersucht, inwieweit das System aus Merkmalsextraktion und Erkennung die restlichen Kriterien erfüllt.

8.5 Evaluierung der unterschiedlichen Merkmalsextraktionsverfahren

In Kap. 7.4 wurde schon festgestellt, daß die untersuchten modellbasierten Merkmalsextraktionsverfahren (MEVs) für die Anwendung in einer Gestikerkennung aus verschiedenen Gründen nicht geeignet erscheinen. Die Auswahl des optimalen MEV durch Evaluierung der Erkennung erfolgt daher aus der Gruppe der pixelbasierten Verfahren. Die Vorgehensweise ist wie folgt:

- Zunächst wird untersucht, welche der in Kap. 7.3.1 vorgestellen 12 Varianten zur Berechnung von Moment-Invarianten (MI) die besten Erkennungsleistungen liefert (s. Kap. 8.5.1). Dazu werden die Übungsdaten (s. Anh. C.1.1) *einer* Versuchsperson verwendet.
- Die Bildstreifenmethode (BS-Methode) aus Kap. 7.3.5 erfüllt von vornherein nicht das wichtige Kriterium der Invarianz. Da aber die Bildstreifen mit den Bildvektoren (BVs) verwandt sind, eignen sie sich sehr gut für eine vergleichende Betrachtung, mit der die Leistungsfähigkeit der BVs eingeschätzt werden kann. Dieser Vergleich erfolgt auf den Analysedaten (s. Anh. C.1.3), die so aufgenommen wurden, daß sie keine invarianten Merkmalsextraktionsverfahren zu ihrer Verarbeitung voraussetzen (s. Kap. 8.5.2).
- Abschließend wird das beste MI- mit dem BV-Verfahren verglichen (s. Kap. 8.5.3). Hierzu wird wieder der Übungsdatensatz einer Person verwendet.

Alle Evaluierungen in diesem Teil werden mit voller Bildgröße und bei voller Bildwiederholrate durchgeführt (Ausnahme s. Kap. 8.5.2). Angaben zu diesen Größen und Untersuchungen, wie sich deren Reduzierung auswirkt, finden sich in Kap. 8.6.3.

8.5.1 Vergleich der Moment-Invarianten-Verfahren

Sowohl die Hu-Moment-Invarianten (HMIs) als auch die Zernike-Moment-Invarianten (ZMIs) basieren auf den normalisierten Zentralmomenten, die sich jeweils auf vier verschiedene Arten berechnen lassen (vgl. Kap. 7.3.1.2). Da außerdem noch Grauwert- und Kantenbilder zugelassen sind, ergeben sich die sechs in der Tabelle 8.1 angegebenen Berechnungsverfahren. Dabei ist berücksichtigt, daß es nicht sinnvoll erscheint, die Konturdarstellung eines Kantenbildes zu berechnen.

Wie in Kap. 7.3.4 erläutert, können die Merkmalsvektoren unter Verwendung der MIs auf drei verschiedene Arten gebildet werden, die sich in der Verwendung von Differenz-Komponenten unterscheiden. Nur die beiden Merkmalsvektor-Zusammensetzungen nach den Gln. (7.24) und (7.25) erfüllen das Kriterium der Invarianz (E3 in Kap. 8.4). Sie werden im folgenden mit MVZ1 und MVZ2 abgekürzt (s. Tabelle 8.1).

MIs sind insbesondere invariant gegenüber Spiegelungen, was für die gewünschte Anwendung ideal ist: dadurch werden einhändige Gesten von *Links- und Rechtshändern* auf dieselben Merkmale abgebildet, so daß sie durch gemeinsame Modelle beschrieben werden können. Die Trajektorienkomponenten der Merkmalsvektoren nach MVZ1 und MVZ2 sind dagegen nicht spiegelungsinvariant und dürfen es auch nicht sein, da die Bewegungsrichtung einer Geste immer dieselbe sein muß, ob sie nun links- oder rechtshändig ausgeführt wird.

FB	binäres Flächenmoment
FG	gewichtetes Flächenmoment
KB	binäres Konturmoment
FBK	binäres Flächenmoment über die Kontur
FBΔ	binäres Flächenmoment auf Kantenbild
FGΔ	gewichtetes Flächenmoment auf Kantenbild
MVZ1	Merkmalsvektor-Zusammensetzung nach Gl. (7.24)
MVZ2	Merkmalsvektor-Zusammensetzung nach Gl. (7.25)

Tabelle 8.1: Abkürzungen im Zusammenhang mit der Evaluierung der Moment-Invarianten-Verfahren

	Verfahren	$D_{\max}$	L							
			2	4	8	16	32	64	128	256
r	FB	7	16,14	27,02	44,65	70,73	94,37	96,25	97,00	98,69
	FG	7	**12,38**	**25,70**	**38,84**	**76,55**	**92,31**	**97,94**	**97,94**	**99,06**
	KB	15	11,82	34,52	43,90	64,17	80,30	87,62	94,18	96,25
	FBK	7	16,89	35,27	62,48	82,36	92,12	95,68	97,37	99,06
	FBΔ	15	16,14	36,77	71,11	84,62	91,37	95,31	97,19	97,94
	FGΔ	7	15,76	39,77	45,78	80,68	87,80	90,81	95,31	97,00
s	FB	7	13,22	28,16	25,39	37,04	56,33	70,34	83,58	92,42
	FG	7	**42,93**	**29,58**	**31,80**	**37,93**	**60,26**	**77,95**	**89,59**	**96,02**
	KB	15	15,30	27,63	32,17	37,28	51,86	66,57	82,89	90,03
	FBK	7	15,83	26,70	31,58	39,87	57,64	74,40	83,32	92,07
	FBΔ	15	7,14	16,67	38,03	55,99	71,96	85,57	91,54	94,30
	FGΔ	7	20,96	28,04	32,10	42,87	52,26	61,78	73,53	83,67

Tabelle 8.2: Verlauf der Erkennungsraten r und Erkennungssicherheiten s (in %) über Codebuchgröße L für unterschiedlich berechnete HMI-Merkmalsvektoren (bestes Verfahren fett hervorgehoben; Übungsdatensatz; $D_{\max}$ jeweils für maximale Erkennungsrate; Anzahl der Zustände $N = 9$; Merkmalsvektor nach MVZ2; Abkürzungen s. Tabelle 8.1; Darstellung von r in Bild 8.1)

8.5.1.1 Ergebnisse für Merkmalsvektoren mit Hu-Moment-Invarianten

In Tabelle 8.2 sind die Erkennungsraten r und -sicherheiten s für $N = 9$ Zustände für Merkmalsvektoren aus HMIs und den sechs verschiedenen Berechnungsverfahren über eine steigende Codebuchgröße dargestellt (Darstellung von r in Bild 8.1). Die Erkennungsrate steigt für alle Verfahren sehr stark mit der Codebuchgröße an. Ihre maximale Erkennungsrate erreichen die Verfahren entweder mit der Dimension $D_{\max} = 7$ (HMIs 2. Grades) oder $D_{\max} = 15$ (HMIs 3. Grades).

Betrachtet man eine Codebuchgröße ab $L = 64$, so ist das FG-Verfahren (gewichtetes Flächenmoment) den anderen Verfahren klar überlegen, und es läßt sich mit $L = 256$ eine Erkennungsrate von knapp über 99 % erzielen. Obwohl die FBK-HMIs (binäre Flächenmomente über die Kontur) nicht die Grauwerte der Handregion auswerten, lassen sich mit ihnen die zweitbesten Erkennungsraten erzielen; für $L = 256$ stimmen die Raten sogar überein. Da die Ergebnisse mit den exakten binären Flächenmomenten (FBs) im Vergleich zu dem FBKs deutlich abfallen, obwohl die FBKs die FBs nur approximieren, muß die erhöhte Leistungsfähigkeit der FBKs darauf zurückzuführen sein, daß sie in der

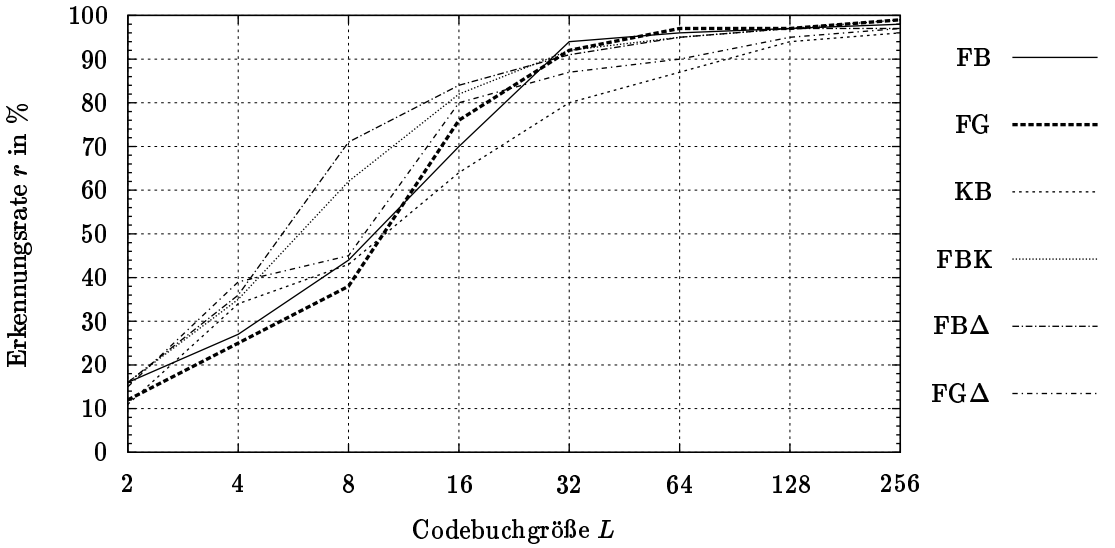


Bild 8.1: Darstellung der Erkennungsraten r zu Tabelle 8.2

Lage sind, kleinere Segmentierungsfehler zu unterdrücken (s. Kap. 7.3.1.2). Trotz dieser Fähigkeit stellen die FBs die beste Variante dar, weil sich mit ihnen im Vergleich zu den FBKs eindeutig die beste Erkennungs*sicherheit* s erreichen läßt.

Interessant ist außerdem, daß die *Kontur*momente (KB) die schlechtesten Ergebnisse liefern, obwohl sie exakt auf derselben Information wie die FBKs beruhen: allein die Verrechnung der Konturelemente ist unterschiedlich.

Die kantenbasierten Verfahren (FBΔ und FGΔ) sind deutlich schlechter als ihre verwandten flächenbasierten Verfahren (FB und FG), obwohl Kanten vom Augenschein her als Unterscheidungsmerkmal sehr geeignet erscheinen. Hier wirkt sich außerdem — im Gegensatz zu den flächenbasierten Verfahren — die Verwendung der Gewichtung (FGΔ im Vergleich zu FBΔ) *negativ* auf das Ergebnis aus.

Tabelle 8.3 und Bild 8.2 zeigen den Verlauf der Erkennungsraten r über der Dimension des Merkmalsvektors bei einer konstanten Codebuchgröße von $L = 256$; Tabelle 8.3 zeigt zusätzlich noch die Erkennungssicherheiten s. Es ist zu erkennen, daß die Erkennungsraten aller Verfahren von Dimension $D = 3$ auf $D = 7$ sehr stark ansteigen: dies zeigt, daß die Trajektorie, die in den ersten drei Merkmalsvektorkomponenten enthalten ist, allein zur Unterscheidung der Geste nicht ausreicht, denn ab der Dimension $D = 7$ kommen Komponenten zur Formbeschreibung hinzu. Diese Komponenten steigern die Erkennungsrate bis zu dem verfahrensabhängigen Maximum bei $D_{\max} = 7$ bzw. $D_{\max} = 15$ absolut um bis zu 16 %. Alle Verfahren fallen mit größer werdender Dimension mehr oder weniger stark in der Erkennungsrate ab. Offenbar sind Gesten so variabel, daß sich eine Steigerung der Merkmalsgenauigkeit negativ auf die Erkennung auswirkt.

Die Erkennungsraten fallen mit steigendem D unterschiedlich stark ab. Tendenziell gilt, daß die schlechteren Verfahren auch schwächer abfallen. Der Verlauf der Erkennungssicherheiten zeigt gewisse Parallelen: während die Sicherheit bei den schlechten Verfahren noch bis in höhere Dimensionen als $D_{\max}$ ansteigen kann, verläuft sie bei den besten Verfahren parallel zur Erkennungsrate.

	Verfahren	D						
		3	7	15	25	37	51	67
r	FB	92,68	98,69	94,00	91,93	85,18	78,05	58,16
	FG	**94,00**	**99,06**	**97,37**	**94,37**	**90,81**	**87,43**	**62,66**
	KB	79,36	95,31	96,25	90,62	92,31	88,56	89,31
	FBK	91,93	99,06	95,87	93,43	91,37	90,24	81,80
	FBΔ	90,99	97,75	97,94	97,00	97,19	96,06	95,12
	FGΔ	89,31	97,00	94,37	92,50	87,62	82,74	80,49
s	FB	67,54	92,42	92,48	91,74	86,83	85,11	84,91
	FG	**70,51**	**96,02**	**93,89**	**95,55**	**89,64**	**88,61**	**81,97**
	KB	51,53	84,94	90,03	90,86	92,98	90,72	91,86
	FBK	66,01	92,07	91,30	92,71	90,47	89,40	84,04
	FBΔ	60,59	88,85	94,30	97,15	96,78	96,16	96,71
	FGΔ	59,31	83,67	88,29	91,71	91,07	88,76	88,21

Tabelle 8.3: Verlauf der Erkennungsraten r und Erkennungssicherheiten s (in %) über Merkmalsvektordimension D für unterschiedlich berechnete HMI-Merkmalsvektoren (bestes Verfahren fett hervorgehoben; Codebuchgröße $L = 256$, sonstige Parameter s. Tabelle 8.2; Abkürzungen s. Tabelle 8.1; Darstellung von r in Bild 8.2)

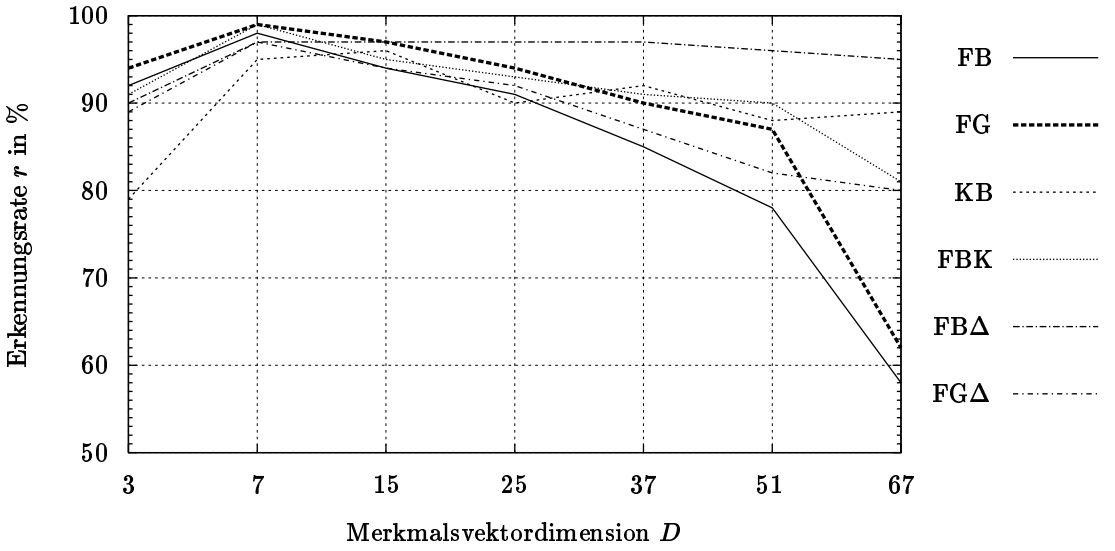


Bild 8.2: Darstellung der Erkennungsraten r zu Tabelle 8.3

8.5.1.2 Vergleich der verschiedenen Merkmalsvektor-Zusammensetzungen

Tabelle 8.4 und Bild 8.3 zeigen die Ergebnisse, wenn man statt der bisherigen Merkmalsvektor-Zusammensetzung 2 (MVZ2) die alternative MVZ1 verwendet: hier entfallen die zusätzlichen differentiellen Komponenten der Moment-Invarianten. Der direkte Vergleich der FB-Merkmale bei MVZ2 und MVZ1 zeigt, daß die Merkmalsvektoren mit MVZ1 zwar nur einen geringeren Extremwert erreichen, dafür aber mit wachsender Dimension wesentlich flacher abfallen (vgl. Bild 8.3 mit 8.2). Dies gilt tendenziell auch für die restlichen Verfahren. Im Sinne der maximal möglichen Erkennungsrate bleiben somit FB-Vektoren mit MVZ2 die optimale Wahl. Mit der Verflachung der Kennlinien geht eine Verschiebung der maximalen Erkennungssicherheit s hin zu größeren Dimensionen einher. Damit ergibt

	Verfahren		*D*						
			3	7	15	25	37	51	67
r	MVZ2	FG	94,00	99,06	97,37	94,37	90,81	87,43	62,66
r	MVZ1	FB	92,68	97,75	96,62	94,37	91,37	85,18	77,49
		FG	94,00	98,87	97,56	96,81	94,56	91,18	82,18
		KB	79,36	97,00	97,94	95,12	96,25	95,68	94,75
		FBK	91,93	98,31	96,62	94,00	95,87	94,00	91,93
		FBΔ	90,99	97,75	97,94	97,75	97,56	97,75	97,19
		FGΔ	89,31	97,75	98,31	96,62	95,68	94,56	92,31
s	MVZ2	FG	70,51	96,02	93,89	95,55	89,64	88,61	81,97
s	MVZ1	FB	67,54	92,21	94,45	96,59	93,60	91,60	85,29
		FG	70,51	93,54	96,64	95,67	94,50	94,26	89,10
		KB	51,53	88,94	92,26	94,80	96,24	95,53	95,92
		FBK	66,01	91,66	92,39	95,27	95,27	93,44	89,58
		FBΔ	60,59	92,37	96,89	97,97	98,74	98,90	98,26
		FGΔ	59,31	90,56	95,48	95,47	96,89	94,31	94,59

Tabelle 8.4: Verlauf der Erkennungsraten r und Erkennungssicherheiten s (in %) über Merkmalsvektordimension D für unterschiedlich berechnete HMI-Merkmalsvektoren bei MVZ1 im Vergleich zur MVZ2 (Codebuchgröße $L = 256$, sonstige Parameter s. Tabelle 8.2; Abkürzungen s. Tabelle 8.1; Darstellung von r in Bild 8.3)

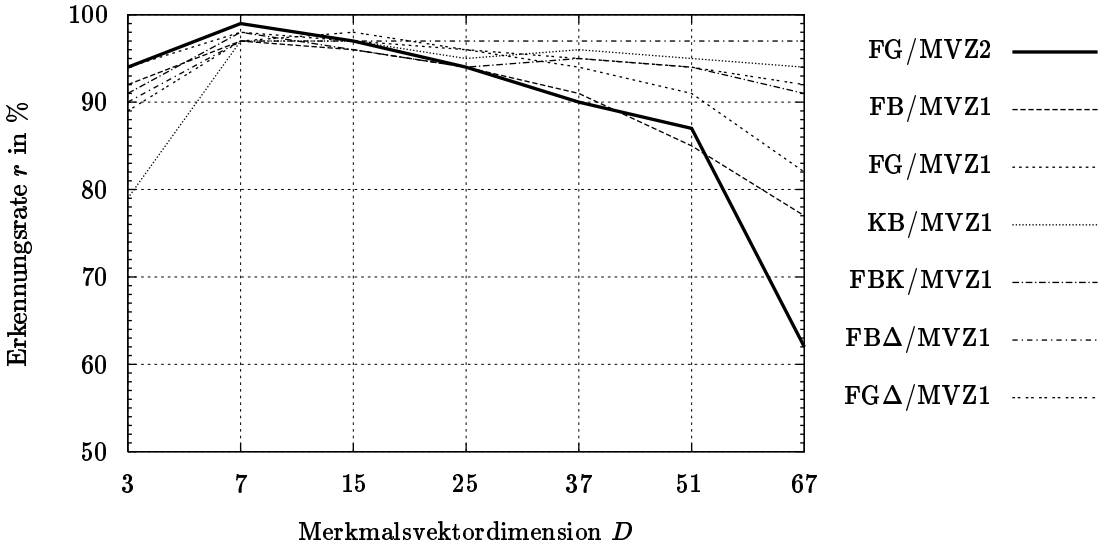


Bild 8.3: Darstellung der Erkennungsraten r zu Tabelle 8.4

sich als weiterer ungünstiger Umstand für die MVZ1, daß sich bei keinem der Verfahren mehr die Dimension der maximalen Erkennungsrate mit der der maximalen Erkennungssicherheit deckt.

	Verfahren		D						
			3	7	15	25	37	51	67
r	HMI	FG	94,00	99,06	97,37	94,37	90,81	87,43	62,66
r	ZMI	FB	92,68	98,69	95,50	90,81	82,18	71,67	47,09
		FG	94,00	99,06	98,50	97,56	91,93	90,81	72,61
		KB	79,36	95,31	95,12	92,87	92,68	88,00	89,12
		FBK	91,93	98,69	94,93	95,87	89,49	89,68	59,10
		FBΔ	90,99	97,75	97,37	96,81	97,19	95,31	94,93
		FGΔ	89,31	97,37	95,87	91,37	90,62	85,93	85,74
s	HMI	FG	70,51	96,02	93,89	95,55	89,64	88,61	81,97
s	ZMI	FB	67,54	92,42	92,52	93,71	86,81	78,23	73,78
		FG	70,51	96,02	95,71	97,43	92,04	91,23	79,13
		KB	51,53	84,94	91,28	88,97	90,00	88,01	90,27
		FBK	66,01	93,42	90,40	95,23	83,76	78,91	81,85
		FBΔ	60,59	88,85	94,51	96,48	96,55	97,01	95,84
		FGΔ	59,31	83,88	89,48	93,26	92,27	92,43	90,69

Tabelle 8.5: Verlauf der Erkennungsraten r und Erkennungssicherheiten s (in %) über Merkmalsvektordimension D bei Verwendung von HMIs im Vergleich zu ZMIs (Codebuchgröße $L = 256$, sonstige Parametereinstellungen s. Tabelle 8.2; Abkürzungen s. Tabelle 8.1; Darstellung von r zusätzlich in Bild 8.4)

8.5.1.3 Vergleich von Merkmalsvektoren mit Hu-Moment-Invarianten und Zernike-Moment-Invarianten

Der Vergleich von HMI- und ZMI-Merkmalsvektoren zeigt zunächst einmal, daß sowohl Erkennungsraten als auch Erkennungssicherheiten bis $D = 7$ bis auf kleine numerische Ungenauigkeiten identisch sind (s. Tabelle 8.5 und Bild 8.4 im Vergleich mit Tabelle 8.3 und Bild 8.2). Das ist nicht verwunderlich, da bei Dimension $D = 3$ exakt dieselben Merkmalskomponenten verwendet werden; bei $D = 7$ unterscheiden sich HMIs und ZMIs nur durch die Skalierung und einen Offset (s. Kap. 7.3.3).

Im Unterschied zu den HMI- zeigen die ZMI-Merkmalsvektoren bei *allen* Verfahren ihre maximale Erkennungsrate bei $D_{\max} = 7$. In der Regel fallen die Erkennungsraten bei den ZMI-basierten Merkmalen auch zu höheren Dimensionen hin schneller ab. Dies deutet in Analogie zu den oben angestellten Betrachtungen darauf hin, daß die ZMIs die Bildsequenzen zwar *genauer* beschreiben können (vgl. Betrachtungen von [Tea80]), daß sich dies aber aufgrund der großen Variationsbreiten gestischer Bewegungen negativ auswirkt.

Im Gegensatz zur allgemeinen Tendenz fallen für das optimale FG-Verfahren die Erkennungsraten und -sicherheiten von ZMI-Vektoren mit höherer Dimension flacher ab als die von HMI-Vektoren; das ändert allerdings nichts am identischen Optimum bei $D_{\max} = 7$. Bei dieser Dimension ist somit die Wahl der Moment-Invarianten völlig gleichgültig. Da die ZMIs bis zur zweiten Ordnung ein klein wenig rechenaufwendiger sind (s. Vergleich der Formeln in Kap. 7.3.3), sind die aus grauwertgewichteten Flächenmomenten berechneten HMIs der Dimension $D = 7$ und der Merkmalsvektor-Zusammensetzung 2 mit zusätzlichen HMI-Differenzen zunächst die optimale Wahl. Die Ergebnisse mit diesen Merkmalen werden später mit denen mit Bildvektoren erzielbaren Erkennungsleistungen verglichen.

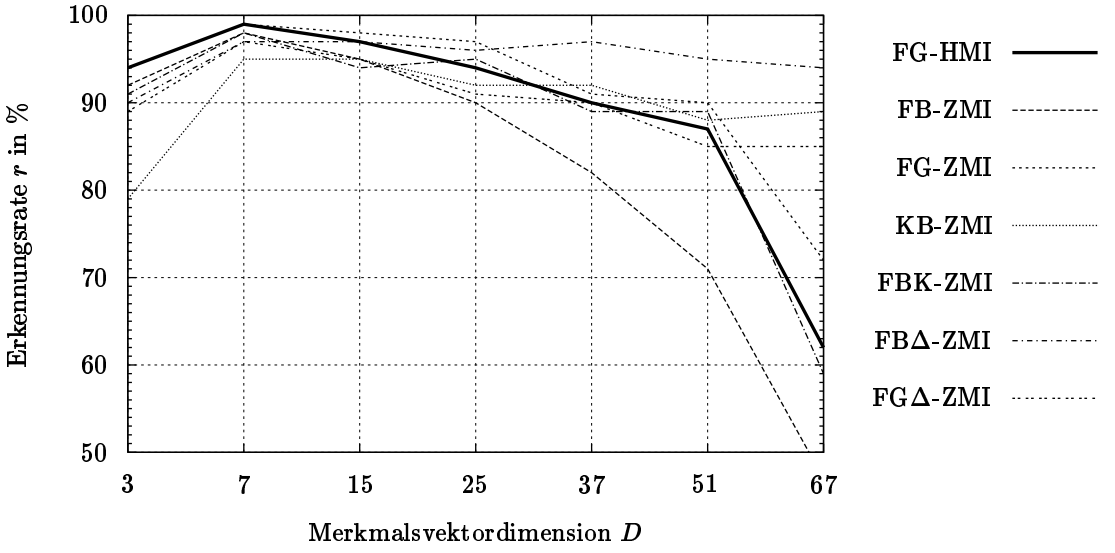


Bild 8.4: Darstellung der Erkennungsraten r zu Tabelle 8.5

8.5.2 Vergleich von Bildvektoren mit Bildstreifen

Das BV-Verfahren soll zunächst im Vergleich mit dem BS-Verfahren seine Leistungsfähigkeit unter Beweis stellen. Dazu wird der Analysedatensatz (s. Anh. C.1.3) verwendet, der so aufgenommen wurde, daß er nicht unbedingt auf invariante Merkmale angewiesen ist. Darüberhinaus wird untersucht, welches Potential in einer zusätzlichen Normalisierung der BVs steckt. Eine HMM-Vorverarbeitung nach Gl. (6.35) ist bei diesen Merkmalen aufgrund des begrenzten Wertebereichs nicht nötig. Alle Erkennungsergebnisse wurden mit Modellen gewonnen, deren Codebuch *nicht* nachgeschätzt wurde: es wird also für die Erkennung das Startcodebuch verwendet (s. Kap. 6.3.2.1). Mit einem nachgeschätzten Codebuch ergeben sich für den Vergleich von BV- und BS-Merkmalen qualitativ dieselben Relationen und Aussagen. Die dargestellten Ergebnisse sind Mittelwerte über $N = 5$–15 Zustände. In diesem Bereich bleiben die Erkennungsraten in Sättigung und durch die Mittelung werden die Raten geglättet. Außerdem wurde nur mit $1/16$ der ursprünglichen Bildgröße gearbeitet (vgl. Auswirkungen der Skalierung auf die Erkennungsrate in Kap. 8.6.1). Diese Modellierungsparameter reichen für einen Katalog mit 12 Gesten völlig aus (vgl. [Mor97a]). Es wird immer nur die Erkennungsrate r angegeben, da die Erkennungssicherheit s für die Bewertung nicht benötigt wird. Die Abkürzungen zu den Evaluierungen sind in Tabelle 8.6 zusammengefaßt.

8.5.2.1 Anwendung auf Grauwertbilder

In Tabelle 8.7 werden die Erkennungsraten bei Verwendung von BV- und BS-Merkmalen auf der Grundlage von Grauwertbildern verglichen. Es wird die Codebuchgröße variiert. Dabei fällt auf, daß die Lage der Bildstreifen (vertikal oder horizontal) extrem unterschiedliche Erkennungsraten bedingt. Eine mögliche Erklärung liegt in der Natur der untersuchten Gesten des Analysedatensatzes: es überwiegen Gesten mit horizontalen Bewegungskomponenten bezogen auf die Bildebene (s. Anh. B.3.2). Somit ergibt sich aus asymmetrischen Merkmalen ein asymmetrisches Verhalten bei den Erkennungsraten (vgl. Kap. 7.4.2).

Bildstreifensequenzialisierung	
V	vertikale Streifen horizontal emittiert
H	horizontale Streifen vertikal emittiert
Bildvektorsequenzialisierung	
K	Kontrollsequenz
O	Optimalsequenz
W	Wiederholsequenz
OW	Optimal-Wiederholsequenz
Normierungsarten bei Bildvektoren	
T1	Translationsinvarianz auf 1. Bild
TR1	Translations- und Rotationsinvarianz auf 1. Bild
TA	Translationsinvarianz bei allen Bildern
TRA	Translations- und Rotationsinvarianz bei allen Bildern

Tabelle 8.6: Abkürzungen im Zusammenhang mit Bildstreifen- und Bildvektorverfahren (vgl. Kap. 7.3.5 und 7.3.6)

		Bildvektoren								Bildstreifen	
Norm.		—	—	—	—	T1	TR1	TA	TRA	—	—
Sequ.		K	O	W	OW	O	OW	O	OW	V	H
L	2	46,21	37,65	26,82	30,53	46,44	36,52	42,58	24,02	18,33	18,18
	4	68,86	74,39	66,29	55,91	89,39	81,14	66,59	55,08	41,36	18,33
	8	58,86	90,98	85,45	81,29	96,74	91,36	85,45	82,42	68,41	29,62
	16	83,11	94,77	88,03	91,36	99,02	98,03	94,32	92,20	85,83	30,45
	32	80,76	93,41	83,18	93,11	100,00	99,17	99,39	95,61	91,82	64,39

Tabelle 8.7: Vergleich der Erkennungsraten r (in %) für Bildvektor- und Bildstreifen-Verfahren bei Grauwertbildern (Variation der Codebuchgröße L der Normalisierung und Sequenzialisierung; Mittelwert für Zustandszahl $N = 5$ bis 15; Initialisierungsraster 6×4; jeweils beste Sequenzialisierung bei vorgegebener Normalisierung; Abkürzungen s. Tabelle 8.6)

Die BVs haben keine Vorzugsrichtung. Für den unnormierten Fall liefert die Sequenz aus optimalen BVs (O) die besten Erkennungsraten. Die Erkennungsraten liegen für alle Codebuchgrößen *deutlich* über den Raten des BS-Verfahrens. Ein besonderes Kennzeichen der BVs besteht darin, daß sich schon mit geringen Codebuchgrößen sehr gute Erkennungsraten erzielen lassen: offensichtlich stellen BVs eine sehr gute Repräsentation der Bildsequenzen dar. Bei $L = 4$ Prototypen beträgt der Abstand zum besten BS-Ergebnis absolut über 33 %. Die anderen Sequenzialisierungsarten sind bei allen Codebuchgrößen (OW) oder bei den meisten Codebuchgrößen (W) immer noch besser als das beste Ergebnis unter Einsatz des BS-Verfahrens.

Zu Vergleichszwecken wurden die initialen BVs einfach emittiert (K). Hier zeigt sich, welche Verbesserungen sich mit der Optimierung und den verschiedenen Sequenzialisierungsarten erreichen lassen.

Normalisiert man die BVs, was mit den BS-Merkmalen nicht möglich ist, so werden je nach Normierungsart leicht Erkennungsraten von 99 % und mehr erreicht: für die beste Normierungsmethode (T1) sogar schon mit $L = 16$ Prototypen. Es ist immer die

		Bildvektoren					Bildstreifen	
Norm.		—	T1	TR1	TA	TRA	—	—
Sequ.		OW					V	H
L	2	22,80	27,58	36,06	8,33	8,33	25,23	16,67
	4	86,74	80,53	79,39	60,76	49,24	60,56	50,38
	8	96,97	99,17	91,59	89,77	71,36	83,56	8,33
	16	97,95	100,00	97,80	95,68	93,86	53,71	18,64
	32	98,41	100,00	99,92	99,85	97,12	93,86	25,45
	64	98,86	99,92	100,00	99,77	98,64	95,45	67,80
	128	97,42	100,00	100,00	99,55	98,79	89,02	81,97

Tabelle 8.8: Vergleich der Erkennungsraten r (in %) für Bildvektor- und Bildstreifen-Verfahren bei Kantenbildern (Variation der Codebuchgröße L der Normalisierung und Sequenzialisierung; Mittelwert für Zustandszahl $N = 5$ bis 15; Initialisierungsraster 6×4; jeweils beste Sequenzialisierung bei vorgegebener Normalisierung; Abkürzungen s. Tabelle 8.6)

beste Sequenzialisierung bei gegebener Normierung angegeben. Interessant ist die Normierung TRA: hier werden die Vektoren bezogen auf das jeweils zugrundeliegende Bild translations- und rotationsnormiert. Als Information bleibt somit nur noch die Flächen- und Gestaltänderung der Hand. Obwohl damit der hohe Informationsgehalt der Trajektorie verlorengeht, werden immer noch bis über 95 % Erkennungsrate erzielt und das Verfahren ist in jedem Fall besser als die Bildstreifen-Methode.

8.5.2.2 Anwendung auf Kantenbilder

Wendet man die Verfahren auf Kantenbilder an (s. Tabelle 8.8), so steigen die Erkennungsraten für die Bildstreifen deutlich an. Allerdings steigen die Erkennungsraten für die unnormierten BVs noch mehr: im Sättigungsfall ist der Abstand in den Erkennungsraten mindestens 3,5 % (die OW-Sequenz ist immer die beste). Für die Aussagen zur Symmetrie der BSs gilt dasselbe wie bei den Grauwertbildern: die horizontalen Streifen sind deutlich schlechter als die vertikalen.

Werden die BVs normalisiert, so läßt sich leicht die 100 %-Marke in der Erkennungsrate erreichen. Die Sättigung beim besten Normalisierungsverfahren (T1) wird praktisch schon mit $L = 8$ Prototypen erreicht.

In Kap. 7.3.6 wird dargestellt, wie sich mit Hilfe der Orientierung der Kanten die Iteration zur Bestimmung der optimalen BVs beschleunigen läßt. In [Mor97b] ist dargelegt, wie sich dadurch die Anzahl der Iterationsschritte um mehr als das 3,5-fache verringern läßt, wobei die Erkennungsrate im normierten Fall unverändert bleibt. Mit der Berücksichtigung der Orientierung läßt sich somit der Berechnungsaufwand der BVs erheblich verringern.

Als Charakteristik der Bildvektoren läßt sich festhalten, daß sie auch im Zusammenspiel mit niedrig dimensionierten HMMs für hohe Erkennungsraten sorgen. Daß sie darüberhinaus in der Lage sind, Bildsequenzen bei Bedarf sehr genau zu modellieren, konnte in [Mor97b] festgestellt werden, wo sie auch zur Klassifikation dynamischer Mimik auf der Grundlage von Kantenbildern herangezogen wurden. Dadurch wird die universelle Einsatzfähigkeit der Bildvektoren bestätigt.

		Ver-fahren	N	D_{max}/ Raster	Anz. MVs	L 2	4	8	16	32	64	128	256
r	Grauwert	FG	9	7	1	12,38	25,70	38,84	76,55	92,31	97,94	97,94	99,06
		O	15	6 × 4	5	15,01	51,59	82,74	92,31	94,00	96,81	95,50	95,50
			15	8 × 6	10	19,89	51,41	77,30	92,50	96,44	97,75	97,94	96,62
			15	11 × 8	18	17,64	49,91	85,93	92,50	96,44	97,37	97,19	96,62
			15	15 × 11	33	14,26	52,16	84,99	94,37	97,19	97,56	98,50	98,31
	Gradient	FBΔ	9	15	1	16,14	36,77	71,11	84,62	91,37	95,31	97,19	97,94
		OW	21	6 × 4	15	14,82	55,91	92,68	96,06	97,56	98,69	98,12	98,31
			21	8 × 6	30	17,45	51,97	90,24	96,44	98,31	98,50	98,31	98,31
			21	11 × 8	45	15,57	52,16	88,93	97,75	97,56	97,56	97,94	98,12
			21	15 × 11	60	15,20	47,09	91,37	97,94	97,75	97,94	97,94	98,50
s	Grauwert	FG	9	7	1	42,93	29,58	31,80	37,93	60,26	77,95	89,59	96,02
		O	15	6 × 4	5	0,95	4,41	7,88	17,68	30,06	44,96	59,32	69,04
			15	8 × 6	10	0,42	3,50	6,87	13,90	22,06	36,83	50,68	62,54
			15	11 × 8	18	0,43	3,32	5,57	11,75	20,26	31,11	42,59	53,98
			15	15 × 11	33	0,55	2,76	5,20	11,62	19,23	27,34	36,65	45,80
	Gradient	FBΔ	9	15	1	7,14	16,67	38,03	55,99	71,96	85,57	91,54	94,30
		OW	21	6 × 4	15	0,60	3,48	8,24	17,95	35,12	52,79	67,66	76,22
			21	8 × 6	30	0,32	3,41	6,72	14,60	27,57	43,75	58,37	69,34
			21	11 × 8	45	0,25	2,47	6,18	13,05	24,05	38,89	53,36	64,17
			21	15 × 11	60	0,20	2,50	5,97	12,28	22,66	35,41	48,24	58,75

Tabelle 8.9: Vergleich der Erkennungsraten r (in %) für Bildvektor- und HMI-Verfahren bei Grauwert- und Kantenbildern (Variation der Codebuchgröße L und der Größe des Initialisierungsrasters; Normalisierung T1; jeweils beste Sequenzialisierung; Abkürzungen s. Tabellen 8.1 und 8.6; Darstellung ausgewählter r in Bild 8.5)

8.5.3 Vergleich von Moment-Invarianten mit Bildvektoren

Da das Bildvektor-Verfahren normierbar ist, kann ein Vergleich der Erkennungsleistung mit den Moment-Invarianten-Verfahren auf Basis des kompletten Übungsdatensatzes erfolgen. Für das BV-Verfahren wird dabei lediglich eine translatorische Normierung auf das erste Bild einer Sequenz vorgenommen. Dies hat sich entsprechend den Ergebnissen in Kap. 8.5.2 bei den Analysedaten als ausreichende und beste Normalisierung erwiesen und konnte für die Übungsdaten bestätigt werden.

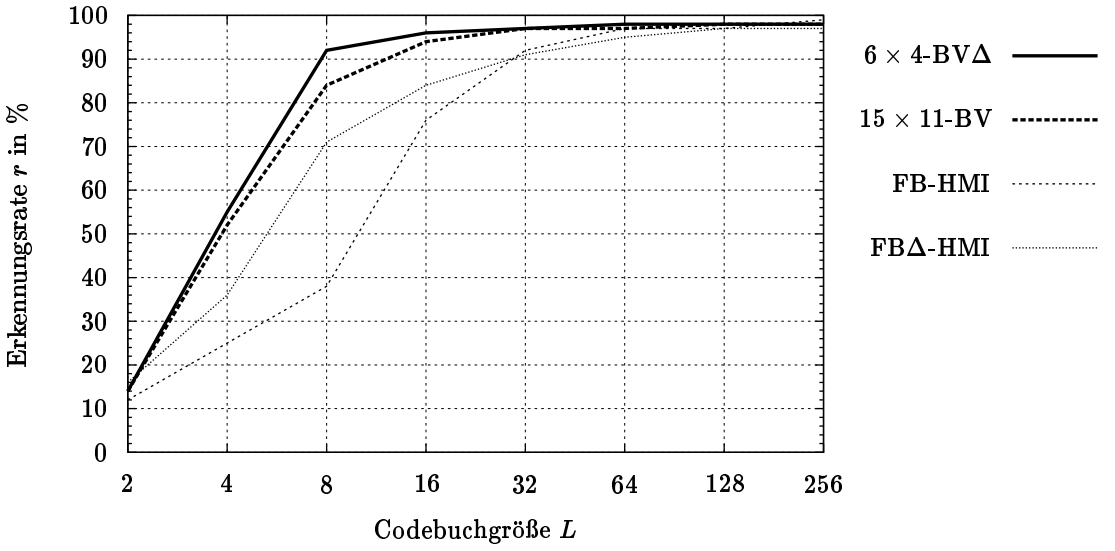


Bild 8.5: Darstellung ausgewählter Erkennungsraten r aus Tabelle 8.9 in % (6 × 4-BVΔ: Bildvektoren auf Kantenbild, Initialisierungsraster 6 × 4; 15 × 11-BV: Bildvektoren auf Grauwertbild mit Initialisierungsraster 15 × 11; sonstige Abkürzungen s. Tabelle 8.1)

In Tabelle 8.9 werden Erkennungsraten r und -sicherheiten s gezeigt, die sich bei Anwendung des BV-Verfahrens bei der jeweils optimalen Sequenzialisierung erreichen lassen. Es sind Ergebnisse für Grauwert- und Kantenbilder bei verschieden großen Initialisierungsrastern aufgeführt. Zum Vergleich sind in der Tabelle auch die Ergebnisse des entsprechenden HMI-Verfahrens angegeben. Im Unterschied zu dem HMI- benötigt das BV-Verfahren auch bei den Übungsdaten keine HMM-Vorverarbeitung, da der Wertebereich streng begrenzt ist und die Werte der Merkmalsvektorkomponenten in vergleichbaren Größenordnungen liegen. Während die HMI-Vektoren mit nur $N = 9$ Zuständen bereits optimal arbeiten, lassen sich die Erkennungsraten des BV-Verfahrens mit etwas mehr Zuständen nochmals leicht steigern (um ca. 1 % absolut). Dies liegt daran, daß beim BV-Verfahren mehrere Merkmalsvektoren pro Bild emittiert werden, die sich besser mit einer größeren Zustandszahl modellieren lassen.

Die Charakteristika der BV-Verfahren aus der Evaluierung in Kap. 8.5.2 zeigen sich auch im Übungsdatensatz: schon bei geringen Codebuchgrößen läßt sich eine sehr hohe Erkennungsrate erzielen (s. Tabelle 8.9). Die BVs sind zwar im Zusammenspiel mit Gradientenbildern immer noch besser als mit Grauwertbildern, allerdings sind die Unterschiede nicht mehr so groß wie beim Analysedatensatz. Interessant ist, daß das HMI-Verfahren genau die entgegengesetzte Tendenz aufweist: hier sind die Ergebnisse mit Gradientenbildern bei hohen Codebuchgrößen schlechter als mit Grauwertbildern (s. ausgewählte Erkennungsraten in Bild 8.5). In dieser Darstellung erkennt man auch deutlich den Unterschied der Erkennungsraten bei niedrigen Codebuchgrößen: das BV-Verfahren weist teilweise eine absolut um fast 50 % bessere Erkennungsrate als die MI-Verfahren auf. Für größere Codebuchgrößen gleichen sich die Erkennungsraten zwar allmählich an, jedoch weist hier das BV-Verfahren mit dem Raster 15 × 11 schon bei $L = 128$ eine Erkennungsrate von 98,5 % auf, die von dem HMI-Verfahren erst bei $L = 256$ erzielt wird.

Bei den Erkennungssicherheiten fällt das BV-Verfahren ab: es erreicht nie mehr als 76 %, während das HMI-Verfahren auf bis zu 96 % kommt.

Als Fazit ergibt sich: will man mit geringen Codebuchgrößen von beispielsweise $L = 16$ schon sehr gute Erkennungsraten von rund 98 % erzielen, so kommt nur das BV-Verfahren in Frage. Bei größeren Codebüchern ab $L = 128$ kommen die Ergebnisse von HMI- und BV-Verfahren in vergleichbare Regionen; dort ist dann das HMI-Verfahren aufgrund der höheren Erkennungssicherheit die bessere Wahl.

8.6 Untersuchungen zur praktischen Einsatzfähigkeit

Das wichtigste Kriterium für den praktische Einsatz ist die Echtzeitfähigkeit (Kriterium E4), die von der gegebenen Datenrate abhängt (s. Kap. 8.6.1). In Kap. 8.6.2 werden Laufzeitmessungen präsentiert, die über das Echtzeitkriterium eine endgültige Auswahl des optimalen Merkmalsextraktionsverfahrens ermöglichen. Mit dem optimalen Verfahren werden dann Untersuchungen zum Verhalten bei Verminderung der Datenrate (s. Kap. 8.6.3) durchgeführt.

Evaluierungen zum Einfluß von Bewegungsunschärfe (Kriterium E5, s. Kap. 8.6.4) und zur Personenunabhängigkeit (Kriterium E6, s. Kap. 8.6.5) runden die Untersuchungen zur praktischen Einsatzfähigkeit ab.

8.6.1 Möglichkeiten zur Verminderung der Datenrate

Für die Untersuchungen wurden nach der D1-Norm digitalisierte PAL-Bildsequenzen verwendet [Sch94b], bei denen nach dem Zeilensprungverfahren versetzte Halbbilder zu ineinander verschränkten Vollbildern zusammengefügt sind. Da diese Halbbilder zeitlich nacheinander aufgenommen wurden, können bei Bewegungen in der Bildsequenz starke Artefakte in Form von „Kammstrukturen“ auftreten [Sch94b]. Um solche Bewegungsartefakte zu vermeiden, wurde stets mit digitalisierten *Halbbildern* gearbeitet, und die Halbbilder wurden in ihrer ursprünglichen Reihenfolge hintereinander angeordnet. Damit ergibt sich bei einer Halbbildgröße von 720×288 Pixel eine Halbbildwiederholrate von 50 Hz.

In der D1-Norm wird die *Farb*information einer Zeile nur mit der halben räumlichen Auflösung abgetastet [Sch94b]. Die Farbsegmentierung (s. Kap. 5.1), die allen Merkmalsextraktionsverfahren vorgeschaltet ist, kann daher nur mit einer Genauigkeit von 360×288 Pixel pro Halbbild arbeiten. Die implementierte räumliche Segmentierung liefert daher auch nur ein Halbbild dieser Größe.

Die verwendete Hardware erlaubt auch die Verringerung der Bildgröße in Echtzeit, wobei das Verhältnis von Bildhöhe zu -breite immer gleich bleibt [Sch94b]. Der Skalierungsfaktor f_{skal} gibt das Verhältnis der reduzierten zur vollen Bildseitenlänge an. Dabei sind nur Faktoren der Art $f_{\text{skal}} = 1/n$ mit ganzzahligem n spezifizierbar. Die nächst kleinere Bildgröße ist daher 360×144 Pixel, was nach der Segmentierung 180×144 Pixel ergibt (Zusammenfassung s. Tabelle 8.10). Kleinere Bildgrößen erscheinen für eine Untersuchung nicht mehr sinnvoll.

Bei der Wahl der Bildwiederholrate f_{R} ist zu beachten, daß die zeitlich diskreten Bilder hardwareseitig nicht interpoliert werden können. Will man die Wiederholrate senken und eine *gleichmäßig* ablaufende Bildsequenz erhalten, so kann man immer nur in einem festen Rhythmus ein Halbbild der Sequenz berücksichtigen und dann ein Bild oder mehrere Bilder weglassen [Sch94b]. Die Anzahl der zu unterdrückenden Halbbilder pro berücksichtigtem Bild wird mit dem Parameter f_{sup} angegeben, so daß gilt: $f_{\text{R}} = 50\,\text{Hz}/(1 + f_{\text{sup}})$. Während es im Offline-Betrieb kein Problem ist, eine Bildsequenz

f_{skal}	relative Halbbildgröße	absolute Halbbildgröße	
		vor Segmentierung	nach Segmentierung
1,0	100 %	720 × 288 Pixel	360 × 288 Pixel
0,5	25 %	360 × 144 Pixel	180 × 144 Pixel

Tabelle 8.10: Mögliche Einstellungen des Skalierungsfaktors f_{skal} und Zusammenhang von f_{skal} und der Bildgröße

f_{sup}	absolute Halbbild-wiederholrate f_{R}	relative Halbbildwieder-holrate	verfügbare Zeit pro Halbbild
0	50,00 Hz	100,00 %	20 ms
1	**25,00 Hz**	**50,00 %**	**40 ms**
2	16,67 Hz	33,33 %	60 ms
3	**12,50 Hz**	**25,00 %**	**80 ms**

Tabelle 8.11: Einstellungen des Unterdrückungsparameters f_{sup} mit resultierenden Halbbildwiederholraten (mögliche Hardwareeinstellungen fett hervorgehoben)

aus Halbbildern mit verdoppelter Vollbildwiederholrate zu simulieren, kann die verwendete Hardware sowohl Voll- als auch Halbbilder nur mit der maximalen Wiederholrate von $f_{\text{R}} = 25\,\text{Hz}$ liefern[1]. Da nicht zwischen „geraden“ und „ungeraden“ Halbbildern[2] gewechselt werden kann, ist die nächste gültige Einstellung $f_{\text{sup}} = 3$, entsprechend einer Wiederholfrequenz von $f_{\text{R}} = 12,5\,\text{Hz}$ (Zusammenfassung s. Tabelle 8.11).

Die resultierenden Datenraten bei Kombination von Bildskalierung und Bildwiederholfrequenz sind in Tabelle 8.12 aufgeführt. Man erkennt, daß eine Bildwiederholrate von $f_{\text{R}} = 12,5\,\text{Hz}$ einem Skalierungsfaktor von $f_{\text{skal}} = 0,5$ entspricht. Kombiniert man beides, so beträgt die relative Datenrate nur noch 6,25 % der maximalen Datenrate.

8.6.2 Berücksichtigung der Echtzeitbedingung und endgültige Auswahl des optimalen Merkmalsextraktionsverfahrens

Bei den folgenden Betrachtungen wird davon ausgegangen, daß für die Verarbeitung der Bildsequenzen ein Rechner mit zwei Prozessoren oder zwei über Netzwerk gekoppelte Rechner mit einem Prozessor zur Verfügung stehen. Dadurch kann der Zeitbedarf der Bilddatenaufbereitung von dem der Erkennung entkoppelt werden. Welche Bedingungen das System erfüllen muß, um echtzeitfähig zu sein, hängt im wesentlichen vom gewählten Ansatz für die kontinuierliche Erkennung ab: Beim *zweistufigen Ansatz* nach Bild 1.2 in Kap. 1.3 (ausführlich behandelt in Kap. 9.2) muß auf jeden Fall die gesamte

[1] Das gilt sowohl für die verwendeten Rechnertypen *Indy* als auch *Indigo2* der Firma *Silicon Graphics*.
[2] Die Bezeichnung erfolgt nach den beteiligten geradzahligen bzw. ungeradzahligen Bildzeilen [Sch94b].

f_{skal}	f_{R}			
	50,00 Hz	25,00 Hz	16,67 Hz	12,50 Hz
1,0	100,00 %	50,00 %	33,33 %	25,00 %
0,5	25,00 %	12,50 %	8,33 %	6,25 %

Tabelle 8.12: Relative Datenrate in Bezug zu voller Halbbildgröße und Bildwiederholrate

	f_{skal}	
	1,0	0,5
Segmentierung	24,89	6,22
Kantenbildberechnung	138,06	31,28

Tabelle 8.13: Vorverarbeitungs-Berechnungszeiten (in ms) über Skalierungsfaktor f_{skal} (gemessen auf *Silicon Graphics Indigo2 High Impact* mit *250 MHz Mips R4400 CPU*)

Bilddatenaufbereitung (bestehend aus räumlicher Segmentierung, eventueller Gradientenbildberechnung, Merkmalsextraktion und Bewegungsdetektion)[3] schritthaltend mit dem Videostrom arbeiten können. Da die Erkennung danach gewissermaßen offline erfolgt, bestimmt die Berechnungsdauer für den Klassifikationsschritt lediglich die Antwortzeit des Systems. Deren maximale Dauer hängt von der subjektiven Toleranz des Benutzers ab. Beim *Spotting-Ansatz* nach Bild 1.3 in Kap. 1.3 (genauer s. Kap. 9.3) muß neben den oben genannten Schritten (ohne Bewegungsdetektion) auch die Erkennung schritthaltend arbeiten.

Aus der Bildwiederholrate ergibt sich direkt die Zeit, die pro Bild des Videostroms für Berechnungen zur Verfügung steht. In Tabelle 8.11 ist ersichtlich, daß bei $f_{\text{R}} = 25\,\text{Hz}$ eine Verarbeitungszeit von 40 ms und bei $f_{\text{R}} = 12,5\,\text{Hz}$ eine Verarbeitungszeit von 80 ms ausreichen muß. Da die kürzesten Gesten aus Übungs- und Dialogdatensatz bei $f_{\text{R}} = 50\,\text{Hz}$ nur 10 Halbbilder lang sind (vgl. Tabellen C.1 und C.2 im Anh. C.1), erscheint eine Bildwiederholrate von $f_{\text{R}} = 12,5\,\text{Hz}$ für die Unterscheidbarkeit der Gesten viel zu kurz (es bleiben dann für die kürzesten Gesten 3 Bilder übrig; die Erkennungsraten in Tabelle 8.16 in Kap. 8.6.3 bestätigen, daß dies nicht ausreicht). Daher gilt im folgenden eine Verarbeitungszeit von 40 ms als oberster Grenzwert.

In Kap. 10.6.1 wird sich zeigen, daß die Ausführungszeit für die optimale Bewegungsdetektion vernachlässigt werden kann. Die Berechnungszeiten für die Farbsegmentierung und die Kantenbildberechnung sind in Tabelle 8.13 für 100 %- und 25 %-Halbbilder angegeben. Diese Zeiten sind unabhängig von der Größe der Fläche, die den von den Händen eingenommen wird. Man erkennt, daß die Segmentierung auch für die 100 %-Halbbilder noch in Echtzeit funktioniert, während dies für die Kantenberechnung nur für eine Größe von 25 % gilt. Berücksichtigt man, daß *vor* der Kantenbildberechnung ebenfalls noch eine Segmentierung durchgeführt werden muß, so ist für Merkmalsextraktionsverfahren, die auf Kantenbildern beruhen, das zur Verfügung stehende Zeitkontingent mit 37,5 ms bei 25 %-Halbbildern fast schon ausgeschöpft.

Tabelle 8.14 zeigt die Berechnungszeiten, die die verschiedenen untersuchten Merkmalsextraktionszeiten benötigen. Die Zeiten sind jeweils für unterschiedliche Skalierungsfaktoren und für die Gesten 1 und 36 angegeben. Damit soll erkennbar werden, wie stark die Verfahren von der von der Hand oder den Händen bedeckten Bildfläche abhängen: Geste 1 ist einhändig und zeigt die Hand von der Seite, Geste 36 ist beidhändig und zeigt die Hände von oben mit ausgestreckten Fingern (vgl. Tabelle B.5 in Kap. B.3.1). Für die Echtzeitabschätzung ist der Wert bei Geste 36 als ungefährer oberer Grenzwert wich-

[3] Die Zeit für die Bilddigitalisierung und Übertragung über den Datenbus kann im Vergleich dazu bei den verwendeten Rechnern *Indy* und *Indigo2* mit *Impact Video* und *Compression Options* der Firma *Silicon Graphics* vernachlässigt werden [Sch94b].

Verfahren	Variation	D bzw. Raster (opt.)	f_{skal}				+
			1,0		0,5		
			Geste Nr.				
			1	36	1	36	
HMI	FB	7	13,12	17,48	3,43	4,61	S
HMI	FG	7	14,76	21,21	3,72	5,30	S
HMI	KB	15	3,22	5,89	1,84	3,13	S
HMI	FBK	7	3,24	5,99	1,95	3,26	S
HMI	FBΔ	15	13,16	14,74	3,95	4,65	S + K
HMI	FGΔ	7	13,11	14,82	3,84	4,52	S + K
BV	Grauwertbild	15×11	525,62	2776,70	90,67	486,07	S
BV	Kantenbild	6×4	289,93	423,80	58,42	111,74	S + K

Tabelle 8.14: Berechnungszeiten (in ms) für die verschiedenen Merkmalsextraktionsverfahren über Skalierungsfaktor f_{skal} bei zwei verschiedenen Gesten (in der Spalte „+“ ist angegeben, welche der Berechnungszeiten für die Segmentierung (S) oder die Kantenbildberechnung (K) aus Tabelle 8.13 noch zu addieren sind; gemessen auf *Silicon Graphics Indigo2 High Impact* mit *250 MHz Mips R4400 CPU*; weitere Abkürzungen s. Tabelle 8.1)

tig[4]. Es ist jeweils die optimale Einstellung für die Merkmalsvektordimension bzw. für die Größe des Initialisierungsrasters angegeben.

Auf den ersten Blick erkennt man, daß die BV-Verfahren um Größenordnungen langsamer sind als alle anderen Verfahren. Das ist nicht erstaunlich, da die BV-Verfahren iterativ arbeiten. Das BV-Verfahren auf Gradientenbildern kann allerdings durch Ausnutzung der Kantenorientierung noch um Faktor 3,5 beschleunigt werden (s. Kap. 8.5.2). Addiert man noch die Zeiten für die Vorverarbeitung hinzu, so kann die Berechnung von BV-Merkmalen auf 25 %-Halbbildern etwa in doppelter Echtzeit erfolgen.

Pauschal läßt sich ebenfalls erkennen, daß *keines* der Verfahren, das *Kantenbilder* erfordert, in Echtzeit funktionieren kann — nicht einmal für die verkleinerten Bilder (für die Merkmalsextraktion stehen für diesen Fall noch 2,5 ms zur Verfügung).

Stellvertretend für die Verfahren, die auf Moment-Invarianten beruhen, sind die HMI-Verfahren angegeben: die von den ZMI-Verfahren benötigte Rechenzeiten liegen für alle Variationen dicht über oder unter den HMI-Zeiten. Somit kann gesagt werden, daß alle MI-Verfahren, die auf der Kontur beruhen (KB und FBK) bei voller Halbbildgröße die Echtzeitanforderung leicht erfüllen. Die anderen, flächenbasierten Verfahren (FB und FG) liegen bei unskalierten Bildern aber nur *wenig über* der Echtzeitanforderung.

Geht man auf 25 %-Halbbilder über, so können *alle* MI-Verfahren, die mit Grauwertbildern arbeiten, problemlos für die Merkmalsextraktion in Echtzeit herangezogen werden. Dies gilt insbesondere für das aufwendigste und optimale Verfahren (FG).

Damit liegen nun endgültig die optimalen Merkmale fest: es handelt sich um die Kombination aus differentiellen Trajektorienmerkmalen, Hu-Moment-Invarianten und differentiellen Hu-Moment-Invarianten, die über der Fläche mit Grauwertgewichtung berechnet werden. Auch die Trajektorienmerkmale werden aus den HMIs bestimmt. Es dürfen für den optimalen Fall nur HMIs bis zur zweiten Ordnung verwendet werden.

[4]Es kann natürlich nie ausgeschlossen werden, daß *noch* größere Flächen auftreten, insbesondere, wenn die Hände näher zur Kamera hin gehalten werden. Deshalb ist eine gewisse Sicherheitsreserve notwendig.

N	L							
	2	4	8	16	32	64	128	256
2	6,33	8,35	58,18	32,43	72,81	193,75	205,13	440,55
3	10,93	12,87	93,40	106,89	107,83	365,81	300,28	648,91
4	14,60	17,84	83,83	123,30	176,39	408,51	395,72	855,93
5	18,25	22,15	89,90	142,48	159,30	398,41	459,67	1126,08
10	40,21	45,58	95,01	341,33	269,97	912,01	1031,15	2683,50
15	57,29	68,87	464,99	485,53	994,64	959,08	1555,71	4431,44

Tabelle 8.15: Durchschnittliche Rechenzeiten für die Erkennung (in ms) (Demonstratorkatalog, $D = 7$, $f_{\mathrm{R}} = 25\,\mathrm{Hz}$; gemessen auf *Sun Ultra2* mit *UltraSparc 168 MHz CPUs*)

8.6.3 Einfluß von Bildratenverminderung und Bildverkleinerung auf die Erkennungsleistung

Anhand der Tabelle 8.15 mit den Ausführungszeiten für die Erkennung läßt sich feststellen, daß maximal Codebuchgrößen bis $L = 128$ für ein echtzeitfähiges System noch tolerierbar sind.[5] Je weniger Zustände bei einer akzeptablen Erkennungsrate verwendet werden können, desto besser.

Mit den optimalen grauwertgewichteten HMI-Merkmalen soll nun die Auswirkung von Bildverkleinerung und Bildratenverminderung auf die Erkennungsleistung untersucht werden (s. Tabelle 8.16). Es sind Ergebnisse für 100 % und 25 % Halbbildgröße sowie für vier verschiedene Bildwiederholraten aufgeführt, von denen nur die zwei hervorgehobenen technisch realisiert werden können (s. Kap. 8.6.1). Bei den zwei Codebuchgrößen von $L = 64$ und $L = 128$ wird die Zahl der Zustände jeweils von $N = 2$ bis $N = 5$ variiert. Bei $f_{\mathrm{R}} = 12,5\,\mathrm{Hz}$ können aufgrund der kleinsten vorkommenden Gestenlänge beim Training nur bis maximal $N = 3$ Zustände verwendet werden.

Man erkennt, daß bei den gezeigten Bild- und Codebuchgrößen sowohl die Erkennungsraten r als auch die Erkennungssicherheiten s ein deutliches *Maximum* bei $f_{\mathrm{R}} = 25,00\,\mathrm{Hz}$ zeigen. Dieses der Erwartung entgegengesetze Verhalten rührt daher, daß sich HMMs mit nur relativ *wenigen* Zuständen und *geringen* Codebuchgrößen mit *kurzen* Bildsequenzen besser trainieren lassen als mit längeren Bildsequenzen. Diese können zwar die Bewegungen genauer darstellen, verlangen dann aber wiederum wesentlich genauer auflösende HMMs.

Bei der nächst geringeren realisierbaren Bildwiederholrate von $f_{\mathrm{R}} = 12,50\,\mathrm{Hz}$ ist die Erkennungsleistung insbesondere bei 25 %-Halbbildgröße schon so stark abgefallen, daß sie selbst durch den Gewinn an Verarbeitungszeit nicht mehr gerechtfertigt werden kann. Es bleibt also bei der Vorgabe von 40 ms Verarbeitungszeit pro Bild, die in Kap. 8.6.2 aus der Überlegung heraus gewählt wurde.

8.6.4 Einfluß von Bewegungsunschärfe auf die Erkennungsleistung

Alle Untersuchungen sind bisher mit Daten erfolgt, die möglichst wenig Bewegungsunschärfe zeigten (Übungs- und Analysedatensatz, s. Anh. C.1.1 und C.1.3). Dies kann nur mit einer starken künstlichen Beleuchtung des Arbeitsplatzes erreicht werden, die es gestattet, den elektronischen Verschluß der Kamera auf kurze Zeiten zu stellen.

[5] Das unsystematische Verhalten der Ausführungszeiten ist auf Zufälligkeiten bei der Cachezuteilung zurückzuführen.

f_{skal}	L	N	f_{R}							
			50,00 Hz		**25,00 Hz**		16,67 Hz		**12,50 Hz**	
			r	s	r	s	r	s	r	s
1,0	64	2	95,87	69,05	96,25	76,72	94,00	69,06	90,62	69,79
1,0	64	3	92,50	74,35	95,68	78,42	94,18	74,59	89,31	78,11
1,0	64	4	94,93	78,20	97,00	83,13	93,25	76,02	—	—
1,0	64	5	96,44	76,37	95,87	86,43	94,37	76,44	—	—
0,5	64	2	94,18	64,85	95,12	70,46	91,37	65,36	88,74	67,37
0,5	64	3	92,12	66,05	94,93	75,59	93,06	72,57	90,81	71,79
0,5	64	4	93,06	70,81	95,87	79,28	93,62	75,46	—	—
0,5	64	5	93,06	71,03	94,56	81,45	94,56	74,85	—	—
1,0	128	2	98,12	88,77	98,12	92,80	93,62	77,91	96,25	83,92
1,0	128	3	98,31	90,09	98,31	93,11	94,56	83,27	96,44	89,83
1,0	128	4	98,31	91,17	98,31	95,34	95,31	85,51	—	—
1,0	128	5	98,31	90,83	98,69	95,60	94,93	84,91	—	—
0,5	128	2	96,81	81,50	98,12	84,04	96,62	79,79	94,37	82,03
0,5	128	3	97,37	86,95	98,12	86,92	95,68	85,32	94,37	88,13
0,5	128	4	98,31	84,14	99,06	89,85	95,31	86,37	—	—
0,5	128	5	98,50	85,11	98,31	89,16	95,50	88,18	—	—

Tabelle 8.16: Einfluß von Bildskalierung f_{skal} und Bildwiederholrate f_{R} auf Erkennungsrate r und Erkennungssicherheit s (grauwertgewichtete HMI-Merkmale, $D = 7$; Übungsdatensatz; für $f_{\text{R}} = 12,5\,\text{Hz}$ sind die Gestensequenzen teilweise so kurz, daß nur noch HMMs mit max. $N = 3$ Zuständen trainiert werden können)

Solche Aufnahmen mit scharfen Konturen waren notwendig, um alle möglichen Merkmalsextraktionsverfahren untersuchen zu können. Von Bewegungsunschärfe verwischte Objekte weisen nach der Farbsegmentierung unter Umständen sehr „löchrige“ Bereiche auf, in denen sich Vorder- und Hintergrund oft abwechseln. Verfahren, die auf einer Konturverfolgung beruhen, sind in diesen Bereichen fehleranfällig und ineffektiv: je nach gewählter Einstellung werden nicht alle Konturanfänge gefunden, der Näherungsfehler steigt mit kleineren Teilgebietsgrößen und der Geschwindigkeitsvorteil sinkt mit zunehmend ineinander verschachtelten inneren und äußeren Konturen (vgl. Kap. 7.3.1.2 und Gl. (7.11) Seite 64).

An einem üblichen Arbeitsplatz ist eine starke und heiße Beleuchtung nicht tolerabel. Hier ist es erstrebenswert, mit der vorhandenen Umgebungsbeleuchtung auszukommen. Für das gefundene optimale flächenbasierte HMI-Verfahren sind „Löcher“ im Vordergrundbereich kein Problem. Es soll nun untersucht werden, ob sich diese Fehlsegmentierungen auf die Erkennungsraten auswirken.

Die *Demonstratordaten* wurden unter solchen normalen Lichtverhältnissen aufgenommen (s. Anh. C.1.4). Zur Vergleichbarkeit mit den Übungsdaten wurden für diese Untersuchung nur die Kerngesten gelabelt, so daß die Gesten völlig übereinstimmten. In Tabelle 8.17 sind für den direkten Vergleich die Erkennungsergebnisse für die Übungsdaten denen der Demonstratordaten gegenübergestellt. Man erkennt, daß durch die Bewegungsunschärfe kein Verlust an Erkennungsleistung entsteht. Es ist im Gegenteil sogar so, daß sich mit den als Kerngesten gelabelten Demonstratordaten in der Regel sogar leicht bessere Erkennungsraten und -sicherheiten erzielen lassen als mit den Übungsdaten.

	N	Aufn.	L								
			16	32	64	128	256	512	1024	2048	4096
r	6	Ü	73,73	92,68	97,94	98,50	99,06	99,62	99,06	100,00	99,06
		DS-K	83,68	92,12	95,87	98,50	98,69	99,62	99,81	99,44	97,00
	9	Ü	76,55	92,31	97,94	97,94	99,06	99,44	99,25	99,44	96,44
		DS-K	86,12	93,62	96,81	98,50	99,44	99,44	99,81	99,62	97,19
s	6	Ü	36,72	54,48	77,52	86,47	95,07	97,67	98,68	98,72	98,98
		DS-K	42,32	61,62	79,66	91,61	97,73	99,05	99,34	99,71	97,66
	9	Ü	37,93	60,26	77,95	89,59	96,02	97,55	98,75	99,07	97,27
		DS-K	45,16	67,17	81,25	92,11	97,78	98,95	99,31	99,57	99,90

Tabelle 8.17: Erkennungsraten r (in %) für verschiedene Aufnahmen bei Variation der Codebuchgröße L und der Anzahl der HMM-Zustände N: Übungsdaten Ü, Demosystemdaten mit gelabelten Kerngesten DS-K (grauwertgewichtete HMI-Merkmale, $D = 7$)

8.6.5 Untersuchung zur Personenunabhängigkeit und kategoriale Erkennungsrate

Während bisher alle Trainings- und Testdaten von einer Person stammten, ist es für den praktischen Einsatz wichtig, daß ein System *personenunabhängig* arbeitet, so daß ein unbekannter Benutzer sofort mit dem System umgehen kann. Für die Beurteilung der Personenunabhängigkeit wurden Übungsdaten von fünf Versuchspersonen digitalisiert (vgl. Anh. C.1.1). Wünschenswert wären eigentlich Daten von mehreren hundert Versuchspersonen, doch dies ist beim aktuellen Stand der Technik nicht praktikabel: selbst mit extremer Motion-JPEG-Komprimierung (s. Anmerkung in Kap. C.1) im Verhältnis 1:30 wurden für eine digitalisierte Aufzeichnung von 45 min Länge ca. 2 Gigabyte Festplattenplatz benötigt (s. Anh. C.1). Es wurden drei Szenarien untersucht:

- **Szenarium I:** Die Modelle wurden mit allen vorhandenen Daten von jeweils vier Versuchspersonen trainiert; die Erkennung wurde mit allen Gestendaten der verbleibenden, nicht am Training beteiligten Versuchsperson durchgeführt. Dabei wurden zyklisch alle Versuchspersonen durchlaufen und die Ergebnisse am Ende gemittelt. Obwohl die vorhandenen Datenmengen der einzelnen Versuchspersonen stark schwanken (s. Anh. C.1.1), wurde diese Mittelung zu gleichen Teilen vorgenommen. Die Bestimmung der Mittelwerts- und Kovarianzvektoren für die HMM-Vorverarbeitung (s. Kap. 6.4) erfolgte jeweils mit den Trainingsdaten. Sie wurden dann zur Merkmalstransformation bei Training *und* Erkennung eingesetzt. Bei diesem Szenarium handelt es sich um eine *echte* Benutzerunabhängigkeit.

- **Szenarium II:** Die Modelle wurden mit zwei Drittel der Daten *aller* Versuchspersonen trainiert. Die Erkennung erfolgte mit dem verbleibenden Drittel der Daten *aller* Versuchspersonen. Für die Merkmalstransformation bei Training und Erkennung wurden jeweils derselbe Mittelwerts- und Kovarianzvektor verwendet; beide wurden mit den Trainingsdaten *aller* Versuchspersonen geschätzt.

- **Szenarium III:** Im Unterschied zu Szenarium II wurden sowohl bei Training als auch Erkennung die Mittelwerts- und Kovarianzvektoren für die Merkmalstransformation mit den *personenspezifischen* Trainingsdaten ermittelt. Damit wird der Endzustand einer Adaption an den Benutzer simuliert: im Laufe des Erkennungsbetriebs

können nämlich die HMM-Vorverarbeitungs-Vektoren ebenfalls schritthaltend nachgeschätzt werden, so daß allmählich eine Anpassung an den Benutzer erfolgt.

Für die korrekte gestische Bedienung muß nicht jede einzelne Geste richtig erkannt werden, es reicht, wenn innerhalb einer Gesten*kategorie* richtig erkannt wird. Wie in Tabelle B.6 auf Seite 167 aufgelistet, werden nämlich bis zu vier Einzelgesten auf ein und dieselbe Kategorie (oder auf einen *Befehl*) abgebildet.

Parallel zur Einzelerkennungsrate r wird daher die *Kategorieerkennungsrate* r_{K} eingeführt, die als die relative Anzahl der korrekt klassifizierten Kategorien bezogen auf die Gesamtzahl der Gesten im Erkennungsdatensatz $\mathbf{X}^{\mathrm{Er}}$ definiert ist (vgl. Gl. (8.3) und Angaben in Kap. 8.4 für die Nomenklatur). Dazu wird der Operator kat[.] eingeführt, der gemäß Tabelle B.6 auf Seite 167 die Gestennummern den Kategorienummern zuordnet:

$$r_{\mathrm{K}} = \frac{1}{\left|\mathbf{X}^{\mathrm{Er}}\right|} \sum_{l=1}^{M} \sum_{v=1}^{V^{g_l}} \delta\left(\mathrm{kat}\left[\lambda_v^{\mathrm{Er},g_l}\right] - \mathrm{kat}[g_l]\right). \tag{8.6}$$

Es ist zu erwarten, daß die Kategorieerkennungsrate *besser* ausfällt als die Einzelerkennungsrate, da die Gesten innerhalb der Kategorien gewisse Ähnlichkeiten haben: so beinhalten beispielsweise alle Gesten der Kategorie „Verschieben nach rechts“ (vgl. Tabelle B.4) eine Bewegungskomponente nach rechts, lediglich der Verlauf der Handform ist unterschiedlich. Daher werden Fehlklassifikationen oft nicht zufällig ausfallen, sondern sie bleiben innerhalb der richtigen Kategorie.

Bisher wurde immer die Einzelerkennungsrate als das schärfere Kriterium bestimmt, weil es darum ging, die Leistungsfähigkeit der verschiedenen Verfahren zu vergleichen. Die Kategorieerkennungsrate spezifiziert die Leistungsfähigkeit des Systems im praktischen Einsatz.

Tabelle 8.18 zeigt die Erkennungsleistungen für das Szenarium I getrennt nach einzelnen Personen und den Mittelwert über alle fünf Personen. Im Mittel läßt sich nur eine maximale Einzelerkennungsrate von ca. 59 % erzielen. Zwar liegt die für die Anwendung entscheidende Kategorieerkennungsrate r_{K} um ca. 15 % höher, doch absolut gesehen ist sie immer noch zu niedrig.

Die Maximalwerte der Einzelerkennungsraten (fett hervorgehoben) schwanken zwischen den einzelnen Versuchspersonen um fast 17 %, die der Kategorieerkennungsraten aber nur um knapp 7 %. Die Maxima der Einzelerkennungsraten und der Kategorieerkennungsraten werden nicht immer bei denselbem HMM-Parametereinstellungen erreicht. Wie gut die Gesten einer Versuchsperson erkannt werden, hängt offenbar in etwa davon ab, wieviele Trainingsdaten der jeweiligen Person in die Modelle eingingen: je mehr Trainingsdaten vorliegen, desto besser werden die Gesten einer Person erkannt, da die Modelle dann auch besser an die entsprechende Person angepaßt sind (vgl. Tabelle C.1 Seite 171).

Zwei der Versuchspersonen (VP 3 und VP 6) waren Linkshänder, die anderen Rechtshänder. Es läßt sich kein signifikanter Zusammenhang zwischen Erkennungsrate und Links- oder Rechtshändigkeit feststellen. Während die Unabhängigkeit von Links- oder Rechtshändigkeit für die HMI-Komponenten der Merkmalsvektoren wegen ihrer Spiegelungsinvarianz vorhersehbar war (s. Kap. 7.3.2), unterscheidet sich offenbar auch der Trajektorienanteil zwischen Links- und Rechtshändern nicht so stark, daß sich dies auf die Erkennungsraten niederschlägt.

Daß die Erkennungsraten relativ schlecht ausfallen, liegt daran, daß für eine echte Benutzerabhängigkeit, wie sie durch dieses Szenarium getestet wird, viel zu wenige Personen am Training beteiligt sind. Nur wenn die Merkmale sehr vieler Personen in die Modelle

L	Versuchsperson								
	VP 16			VP 2			VP 3		
	r	r_K	s	r	r_K	s	r	r_K	s
2	13,76	17,03	12,00	8,96	13,10	11,09	13,11	18,41	10,74
4	17,34	24,77	24,71	10,43	17,65	30,43	17,75	27,15	24,93
8	27,78	43,41	27,09	26,47	46,39	25,23	34,83	56,42	19,66
16	49,53	63,97	32,01	37,83	53,61	25,11	38,41	55,89	32,07
32	54,21	73,00	38,08	56,82	70,99	35,32	41,46	60,93	40,36
64	56,70	70,56	45,81	62,03	74,60	44,09	56,69	74,30	53,02
128	**66,93**	**79,85**	58,81	65,37	73,66	54,29	59,60	**75,76**	59,80
256	58,52	72,01	62,01	**67,38**	**75,27**	66,98	56,56	69,67	71,20
512	56,28	69,00	67,79	64,17	73,53	74,07	59,74	73,38	74,01
1024	55,56	69,21	75,20	65,78	74,73	79,24	**60,40**	75,36	78,96

L	Versuchsperson						Mittelwerte		
	VP 6			VP 7					
	r	r_K	s	r	r_K	s	r	r_K	s
2	8,06	11,36	3,69	8,93	11,65	14,92	10,57	14,32	10,49
4	14,29	20,15	40,42	15,73	21,63	26,84	15,11	22,27	29,47
8	30,77	53,11	33,62	28,29	43,27	19,16	29,63	48,52	24,96
16	38,46	60,44	32,37	41,60	62,33	28,44	41,17	59,25	30,00
32	39,19	68,13	49,82	44,78	65,81	39,38	47,30	67,78	40,60
64	43,22	67,03	59,31	50,08	65,36	53,03	53,75	70,38	51,06
128	**50,55**	**73,26**	63,18	53,71	70,05	64,00	**59,24**	**74,52**	60,02
256	50,55	71,06	71,92	56,88	**72,62**	67,62	57,98	72,13	67,95
512	41,39	63,74	75,41	**57,64**	70,80	72,29	55,85	70,09	72,72
1024	40,66	62,27	74,37	57,49	71,10	72,54	55,98	70,54	76,07

Tabelle 8.18: Personenunabhängige Erkennungsraten r für Übungsdaten nach Szenarium I (grauwertgewichtete HMI-Merkmale, $N = 5$, $D = 7$, beste Werte fett hervorgehoben)

einfließen, kann man davon ausgehen, daß alle typischen Variationen der Bewegungsmuster „abgedeckt“ sind, so daß auch die Gesten einer unbekannten Person zuverlässig erkannt werden können [Rab89, Hua90].

Daß Merkmalsvektoren und HMMs in der Lage sind, dieselben fünf Personen *gleichzeitig* zu modellieren, zeigen die Ergebnisse in Tabelle 8.19. Die Einzelerkennungsraten fallen auch ohne besondere Anpassungsmaßnahmen (Szenarium II) schon sehr gut aus. Bestimmt man zusätzlich noch personenspezifische HMM-Vorverarbeitungsvektoren (Szenarium III), so lassen sich die Erkennungsraten noch etwas steigern, so daß maximal eine Kategorieerkennungsrate von über 99 % erreicht werden kann. Dabei ist das Szenarium III nur als Endzustand einer Adaption zu betrachten, in deren Verlauf die personenspezifischen HMM-Vorverarbeitungsvektoren nachgeschätzt werden. Da die Raten insgesamt schon sehr hoch ausfallen, liegen Einzel- und Kategorieerkennungsraten eng beisammen. Außerdem ist gegenüber Szenarium I auffällig, daß eine Steigerung der Modellierungsgenauigkeit durch Erhöhung der Codebuchgröße auch stets mit einer Steigerung der Erkennungsleistung einhergeht, während im Szenarium I der maximal mögliche Wert schon mit kleineren Codebuchgrößen erreicht wird. Dies ist ein weiterer Beleg für die obige Aussage, daß für die echte Benutzerunabhängigkeit Trainingsdaten von zu wenigen Personen vor-

	Szenarium II			Szenarium III		
L	r	r_K	s	r	r_K	s
2	13,89	17,18	12,88	13,39	16,54	13,88
4	17,97	25,05	19,78	19,83	30,14	8,00
8	46,74	64,21	21,98	47,89	65,86	23,93
16	61,70	74,30	30,27	63,28	81,53	29,80
32	73,66	85,40	40,51	79,53	86,97	45,81
64	83,25	89,98	52,62	82,82	90,91	53,85
128	89,33	94,06	67,66	90,19	94,92	67,84
256	92,63	96,06	76,25	94,42	97,21	76,92
512	96,06	98,00	83,17	96,78	98,43	84,27
1024	**97,49**	**98,85**	89,50	**98,28**	**99,28**	89,94

Tabelle 8.19: Personenunabhängige Erkennungsraten r für Übungsdaten nach Szenarium II und III (grauwertgewichtete HMI-Merkmale, $N = 5$, $D = 7$, beste Werte fett hervorgehoben)

liegen: werden die HMMs zu gut an die Trainingsdaten angepaßt, so verschlechtern sich die Erkennungsergebnisse, weil dann die unbekannte Erkennungsperson immer schlechter auf die Modelle paßt.

8.7 Abschließende Beurteilung

Die Evaluierung der verschiedenen pixelbasierten Merkmalsextraktionsverfahren hat gezeigt, daß Merkmalsvektoren auf der Basis von HMIs, die über grauwertgewichtete allgemeine Momente bestimmt werden, in Verbindung mit Information über die Bewegungstrajektorie nach dem Kriterienkatalog aus Kap. 8.4 das optimale Verfahren darstellen. Dies ist umso erstaunlicher, als Moment-Invarianten in erster Linie für die Klassifikation *starrer* Objekte eingesetzt werden (beispielsweise in [Tea80]).

Die „mißbräuchliche“ Verwendung der Moment-Invarianten für die Bewegungsklassifikation wirkt sich allerdings in der mangelnden Skalierbarkeit aus: eine Erhöhung der Dimension des Merkmalsvektors über die optimale Dimension hinaus resultiert sofort in eine Verschlechterung der Erkennungsrate (vgl. Ergebnisse in Kap. 8.5.1). Bei der Klassifikation starrer Objekte steigt die Erkennungsleistung dagegen mit steigender Ordnung der Momente weiter an [Bel91]. Offenbar sind die Bildvektoren für die Modellierung der stark variablen gestischen Bewegungen besser geeignet, denn bei ihnen zeichnet sich ein Optimum nicht so stark ab (vgl. Kap. 8.5.3).

Für die Erkennung kontinuierlicher Gesten in Kap. 10 wird nur noch das optimale HMI-Verfahren verwendet. Durch die differentiellen Trajektorien- und die invarianten HMI-Komponenten ist das Verfahren prinzipiell für den „Endlosbetrieb“ geeignet: Bezugspunkt für die Normierung ist immer nur der ein Zeittakt zurückliegende Merkmalsvektor. Dagegen muß die Klassifikation an die kontinuierliche Erkennung angepaßt werden. Die hierzu untersuchten Möglichkeiten werden in Kap. 9 vorgestellt.

Kapitel 9

Kontinuierliche Erkennung

9.1 Verdeutlichung der Problemstellung

Die in Kap. 6 vorgestellen Modellierungsverfahren eigenen sich zunächst für die *isolierte* Erkennung. Das bedeutet, daß definierte Start- und Endzeitpunkte der zu erkennenden Sequenzen bekannt sein müssen. Die Ergebnisse aus Kap. 8 zeigen, daß sich dann — mit geeigneten Merkmalsextraktionsverfahren — trotz des umfangreichen Gestenkataloges äußerst gute Erkennungsleistungen erzielen lassen. Auf diese Weise konnte die prinzipielle Eignung der stochastischen Modellierung für die Klassifikation von Bildsequenzen gezeigt werden.

In einem *realen* System für den visuellen Dialog sind die Grenzen der Bewegungssequenzen jedoch zunächst unbekannt. Ein solches System erhält am Eingang lediglich einen unstrukturierten, kontinuierlichen Strom von Videobildern. In diesem Strom finden sich nun Abschnitte, die Bewegungspausen oder *bedeutungslose*, beiläufige Bewegungen enthalten und Segmente mit *bedeutungstragenden* Bewegungen. Eine *kontinuierliche Erkennung* läßt sich daher in zwei Aufgaben zerlegen: Mit einer *zeitlichen Segmentierung* müssen im kontinuierlichen Datenstrom zunächst die Segmente identifiziert werden, die die gesuchten bedeutungstragenden Bewegungen enthalten. Eine anschließende *Klassifikation* ordnet diesen Bewegungen dann ihre eigentliche Bedeutung zu.

Im Rahmen dieser Arbeit wurden zwei Ansätze zur kontinuierlichen Erkennung entwickelt und untersucht. Der erste Ansatz ist *zweistufig* und wird in Kap. 9.2 näher beschrieben, Der zweite Ansatz ist *einstufig* und wird in Kap. 9.3 behandelt (vgl. Bilder 1.2 und 1.3 in Kap. 1.3). Wie die Evaluierung in Kap. 10 zeigen wird, verhalten sich die beiden Vorgehensweisen unterschiedlich und sind unterschiedlich leistungsfähig.

9.2 Zweistufiger Ansatz aus Bewegungsdetektion und isolierter Erkennung

Die beiden in Kap. 9.1 verdeutlichten Verarbeitungsschritte werden beim zweistufigen Ansatz getrennt umgesetzt. Die zeitliche Segmentierung wird näherungsweise durch eine Stufe zur Bewegungsdetektion erreicht (s. Kap. 9.2.1). Für die anschließende Klassifikation haben die gefundenen Bewegungssegmente daher einen festen Anfangs- und Endpunkt, so daß hierzu als zweite Stufe eine einfache isolierte Erkennung eingesetzt werden kann

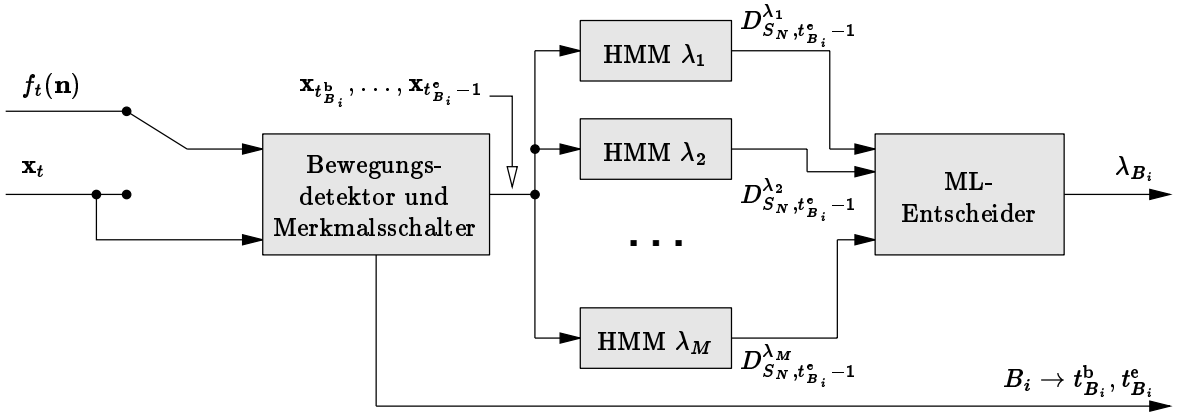


Bild 9.1: Systemkomponenten für die zweistufige kontinuierliche Erkennung

(s. Kap. 9.2.2). Die Anordnung für das zweistufige System ist nochmals in Bild 9.1 dargestellt. Es wurde im Rahmen dieser Arbeit in [Ben98] entwickelt und zum ersten Mal in [Mor99] vorgestellt.

9.2.1 1. Stufe: Bewegungsdetektion

9.2.1.1 Berechnung des Bewegungswertes

Die Bewegungsdetektion basiert auf einem sog. *Bewegungswert* m_t, der zu jedem Zeitpunkt t berechnet wird. Dieser Bewegungswert wird aus der zeitlichen Änderung des Bildinhaltes abgeleitet. Prinzipiell kann man zu seiner Berechnung zwei Möglichkeiten unterscheiden:

1. Der *pixelbasierte* Bewegungswert verwendet direkt die Pixelinformation der Segmentierungsmaske $f_{\mathrm{b},t}(\mathbf{n})$, des segmentierten Bildes $\mathbf{f}_{YUV,\mathrm{s},t}(\mathbf{n})$ oder Komponenten hiervon. Um Berechnungszeit zu sparen, wird dazu lediglich ein Pixel*raster* $K \times L$ betrachtet, das nach dem Formalismus von Gl. (7.26) im Zusammenhang mit der Berechnung von Bildstreifen in Kap. 7.3.5 aus den Rasterpunkten $\mathbf{v}_{ij}$ besteht. An jeder Rasterkoordinate wird die Betragsdifferenz eines Bildpunktes zum Zeitpunkt t und $t-1$ gebildet. Sind mehrere Komponenten vorhanden, so werden diese betragsmäßig aufsummiert:

$$\Delta f_{\mathrm{b},t}(\mathbf{v}_{ij}) = |f_{\mathrm{b},t}(\mathbf{v}_{ij}) - f_{\mathrm{b},t-1}(\mathbf{v}_{ij})| \quad \text{bzw.} \tag{9.1}$$

$$\Delta f_{YUV,t}(\mathbf{v}_{ij}) = \sum_{x \in \{Y,U,V\}} |f_{x,\mathrm{s},t}(\mathbf{v}_{ij}) - f_{x,\mathrm{s},t-1}(\mathbf{v}_{ij})| \tag{9.2}$$

Zur Bestimmung des pixelbasierten Bewegungswertes werden dann die Einzelbetragsdifferenzen wiederum über alle Rasterpunkte $\mathbf{v}_{ij}$ aufsummiert und auf ihre Anzahl normiert:

$$m_{\mathrm{b},t} = \frac{1}{K \cdot L} \sum_{i=1}^{K} \sum_{j=1}^{L} \Delta f_{\mathrm{b},t}(\mathbf{v}_{ij}) \quad \text{bzw.} \quad m_{YUV,t} = \frac{1}{K \cdot L} \sum_{i=1}^{K} \sum_{j=1}^{L} \Delta f_{YUV,t}(\mathbf{v}_{ij}). \tag{9.3}$$

Der Bewegungswert kann auch nur über die Y-Komponente ($m_{Y,t}$) oder die UV-Komponente ($m_{UV,t}$) berechnet werden, wobei die Summe in Gl. (9.2) entsprechend über weniger Elemente läuft.

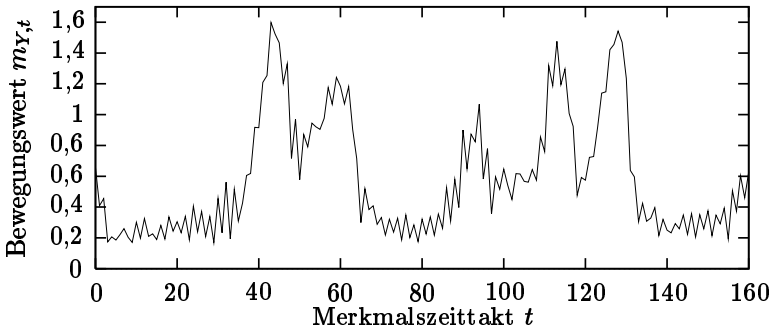


Bild 9.2: Beispielverlauf des pixelbasierten Bewegungswertes $m_{Y,t}$

2. Der *merkmalsbasierte* Bewegungswert verwendet die Merkmale, die für die nachgeschaltete Erkennung sowieso berechnet werden müssen. Der zusätzliche Aufwand für eine solche Bewegungswertberechnung ist im Vergleich zur Merkmalsberechnung vernachlässigbar.

   Der aus den HMIs gebildete Merkmalsvektor lieferte in der durch Gl. (7.25) in Kap. 7.3.4 dargestellten Form die besten Ergebnisse bei der isolierten Erkennung (s. Kap. 8). In diesem Merkmalsvektor sind bereits Komponenten enthalten, die aus zeitlichen Differenzen von Bildmerkmalen bestimmt werden. Für den Bewegungswert können diese Komponenten direkt verwendet werden. Zu einem Vektor $\mathbf{x}_t^{\mathrm{m}}$ zusammengefaßt läßt sich mit der Nomenklatur aus Kap. 7.3.2 und Kap. 7.3.4 schreiben:

$$\mathbf{x}_t^{\mathrm{m}} = [\Delta A_t, \Delta\bar{n}_{1,t}, \Delta\bar{n}_{2,t}, \Delta H_{1,t}, \Delta H_{2,t}, \ldots, \Delta H_{N_O,t}]^{\mathrm{T}}. \tag{9.4}$$

   Der merkmalsbasierte Bewegungswert $m_{H_O,t}$ wird nun definiert als

$$m_{H_O,t} = |\mathbf{x}_t^{\mathrm{m}}| = \sqrt{(\Delta A_t)^2 + (\Delta\bar{n}_{1,t})^2 + (\Delta\bar{n}_{2,t})^2 + (\Delta H_{1,t})^2 + \cdots + (\Delta H_{N_O,t})^2}. \tag{9.5}$$

Typische Verläufe eines pixelbasierten bzw. eines merkmalsbasierten Bewegungswertes sind in Bild 9.2 bzw. 9.3 für jeweils dieselbe Videosequenz gezeigt. Im nächsten Abschnitt wird nun beschrieben, wie aus dem Bewegungswert zusammenhängende Bewegungsintervalle bestimmt werden können. Welcher Bewegungswert die besten Detektionsresulate liefert, läßt sich nur anhand einer Evaluierung feststellen, deren Ergebnisse in Kap. 10.6.1 zu finden sind.

9.2.1.2 Detektionsregeln

Der Bewegungswert m_t ist — unabhängig von der Art, wie er bestimmt wird — typischerweise sehr klein oder verschwindet, wenn keine Bewegungen in einer Bildsequenz vorkommen. Bildrauschen und die Zitterbewegung, die durch die Hintereinanderreihung von geraden und ungeraden Halbbildern entsteht (vgl. Kap. 8.6.1), verhindern, daß der Bewegungswert vollständig auf Null absinkt. Der Bewegungswert wird mit steigender Geschwindigkeit aber auch mit steigender Fläche der Bewegungsanteile in einer Bildsequenz größer. Bewegt sich daher eine Hand parallel zur Bildebene mit einer konstanten Geschwindigkeit, so hängt die Größe des Bewegungswertes von der Abbildungsgröße der Hand und damit von der Entfernung der Hand zur Kamera ab. Dieser Effekt darf nicht

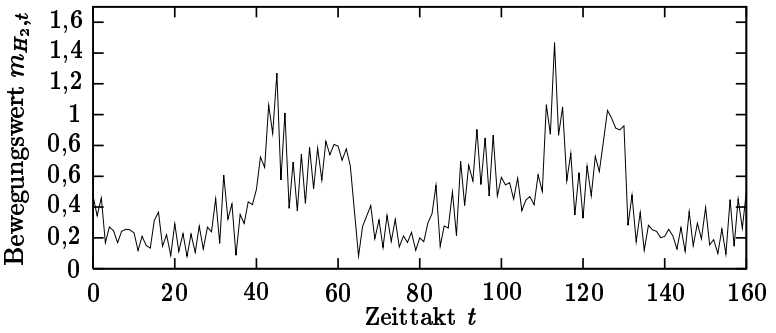


Bild 9.3: Beispielverlauf des merkmalsbasierten Bewegungswert $m_{H_2,t}$

durch Normierung des Bewegungswertes auf die bewegte Fläche beseitigt werden, weil sonst die Bewegungskomponente senkrecht zur Bildebene, die bei *einer* Kamera nur über eine Flächenänderung erfaßt werden kann, wegnormiert würde.

Diese Effekte spielen jedoch eine untergeordnete Rolle, da für die geforderte Detektion von Bewegungsintervallen nicht die *Größe* sondern nur das *Vorhandensein* einer Bewegung erfaßt werden muß. Dies kann prinzipiell durch Anwendung einer *Bewegungsschwelle* m_{S} erreicht werden, die innerhalb eines Bewegungsintervalles vom Bewegungswert überschritten werden muß. Der stark zerklüftete Charakter des Bewegungswertes (s. Bilder 9.2 und 9.3) erfordert jedoch die Einführung von Regeln, die über entsprechende Zeitkonstanten sicherstellen, daß *zusammenhängende* Bewegungsintervalle $B_i = [t^{\mathrm{b}}_{B_i}, t^{\mathrm{e}}_{B_i}[$ gefunden werden:

Startkriterium (B1): Ein Bewegungssegment B_i *beginnt* zum Zeitpunkt $t^{\mathrm{b}}_{B_i}$, wenn der Bewegungswert für mindestens $\tau^{\mathrm{b}}_{\mathrm{min}}$ Zeittakte *über* der Bewegungsschwelle liegt:

$$m_t > m_{\mathrm{S}} \quad \text{für} \quad t = t^{\mathrm{b}}_{B_i}, \ldots, t^{\mathrm{b}}_{B_i} + \tau^{\mathrm{b}}_{\mathrm{min}} - 1. \tag{9.6}$$

Endekriterium (B2): Ein Bewegungssegment B_i gilt zum Zeitpunkt $t^{\mathrm{e}}_{B_i}$ als *beendet*, wenn der Bewegungswert für mindestens $\tau^{\mathrm{e}}_{\mathrm{min}}$ Zeittakte *kleiner oder gleich* der Bewegungsschwelle ist:

$$m_t \leq m_{\mathrm{S}} \quad \text{für} \quad t = t^{\mathrm{e}}_{B_i}, \ldots, t^{\mathrm{e}}_{B_i} + \tau^{\mathrm{e}}_{\mathrm{min}} - 1. \tag{9.7}$$

Mindestlänge (B3): Dabei muß das Bewegungssegment B_i eine *Mindestlänge* $\tau^{\mathrm{l}}_{\mathrm{min}} \leq t^{\mathrm{e}}_{B_i} - t^{\mathrm{b}}_{B_i}$ und

Mindestabstand (B4): einen *Mindestabstand* $\tau^{\mathrm{d}}_{\mathrm{min}} \leq t^{\mathrm{b}}_{B_i} - t^{\mathrm{e}}_{B_{i-1}}$ zum vorherigen Bewegungssegment B_{i-1} aufweisen.

Regel B1 sorgt dafür, daß kurze Peaks in einem ansonsten niedrigen Bewegungswertverlauf nicht fälschlicherweise als Beginn eines Bewegungsintervalles angezeigt werden. Durch B2 wird verhindert, daß ein kurzfristiges Absinken des Bewegungswertes innerhalb einer Bewegung nicht zur Unterbrechung eines Bewegungsintervalles führt. Ein solches Absinken des Bewegungswertes tritt beispielsweise an Umkehrpunkten einer gestischen Bewegung auf. Die Mindestlänge der Regel B3 hilft, zu kurze Gesamtbewegungen herauszufiltern, auch wenn diese die Anfangsbedingung B1 erfüllen. Der Mindestabstand zur Vorgängerbewegung in Regel B4 verhindert die Mehrfachindikation einer Geste, falls sie durch B2 nicht als zusammenhängende Bewegung detektiert werden kann.

Die Regelparameter sind stark voneinander abhängig und damit schwierig zu justieren. Wie sie optimal eingestellt werden können und welche Detektionsergebnisse sich damit erzielen lassen, wird in Kap. 10.6.1 dargestellt.

9.2.2 2. Stufe: Erkennung

Nachdem die Bewegungsdetektion Beginn $t^{\mathrm{b}}_{B_i}$ und Ende $t^{\mathrm{e}}_{B_i}$ eines Bewegungssegmentes festgestellt hat, werden die im Intervall enthaltenen Merkmale zu einer Erkennungssequenz $\mathbf{X}_{B_i}$ der Länge $T_{B_i} = t^{\mathrm{e}}_{B_i} - t^{\mathrm{b}}_{B_i}$ zusammengefaßt:

$$\mathbf{X}_{B_i} = \mathbf{x}_{t^{\mathrm{b}}_{B_i}}, \mathbf{x}_{t^{\mathrm{b}}_{B_i}+1}, \ldots, \mathbf{x}_{t^{\mathrm{e}}_{B_i}-1}. \tag{9.8}$$

Analog zur Verfahrensweise in Kap. 6.3.5 kann nun für jedes der Modelle λ_l der Ausgangsscore bestimmt werden. Eine ML-Entscheidung liefert für das Bewegungsintervall B_i das Modell mit der größten approximierten Erzeugungswahrscheinlichkeit:

$$\lambda_{B_i} = \operatorname*{argmax}_l \tilde{F}(\mathbf{X}_{B_i}|\lambda_l) \quad \text{für} \quad l = 1, \ldots, M. \tag{9.9}$$

Da die Bewegungsdetektionsstufe *jede* Bewegung an die Erkennung weitergibt, kann es vorkommen, daß die Erkennungsstufe unter Umständen auch Bewegungssegmente klassifizieren muß, die keiner gültigen Geste entsprechen können. Bei nicht bedeutungstragenden Bewegungen wird sich demnach ein zufälliges Klassifikationsergebnis einstellen. Diese zufälligen Erkennungen werden in aller Regel zu einer Erhöhung des Erkennungsfehlers beitragen (s. Kap. 10.6).

Die Möglichkeit, eine Zurückweisungsschwelle zu definieren, ist nicht sinnvoll, da es sich im Umgang mit dem Datenmaterial gezeigt hat, daß die Erzeugungswahrscheinlichkeitsdichten, die sich bei gültigen Bewegungen ergeben, sehr starken Schwankungen unterworfen sind. Dies liegt daran, daß die Bewegungsdetektionsstufe nur ungenau arbeitet und daß somit sehr oft Teile von Bewegungen zu früh abgeschnitten oder verlängert werden. Die *absolute* Größe einer gültigen Erzeugungswahrscheinlichkeitsdichte schwankt also sehr stark in Abhängigkeit von der Genauigkeit der Bestimmung des Bewegungsintervalles. Eine Zurückweisungsschwelle setzt aber voraus, daß die Werte der Erzeugungswahrscheinlichkeitsdichten wenigstens bezogen auf das jeweilige Modell einigermaßen konstant sind. Für die ML-Entscheidung sind diese Schwankungen allerdings ohne Belang, da sie nur einen *relativen* Vergleich durchführt.

9.3 Einstufiger HMM-Spotting-Ansatz

Der einstufige oder sog. *Spotting*-Ansatz behandelt die beiden Aufgaben der zeitlichen Segmentierung und der Klassifikation in einem integralen Verfahren [Mor98b, Mor98c, Mor99]. Die grundlegende Idee besteht darin, die Modelle zu jedem Merkmalszeittakt permanent mit den Merkmalen zu beaufschlagen. Während beim zweistufigen Ansatz die Ausgangsscores nur am Ende jeder gefundenen Bewegung betrachtet werden und dann eine ML-Entscheidung getroffen wird, interessiert beim Spotting der Verlauf der Ausgangsscores über die Zeit. Durch ständiges Beobachten der Ausgangsscores erlauben Charakteristika in diesen Scoreverläufen Rückschlüsse auf die gesuchten Bewegungen am Eingang.

Um die Modelle in einem „Endlosbetrieb“ ständig mit neuen Merkmalen beaufschlagen zu können, muß am Viterbi-Algorithmus eine zeitliche Normierung vorgenommen

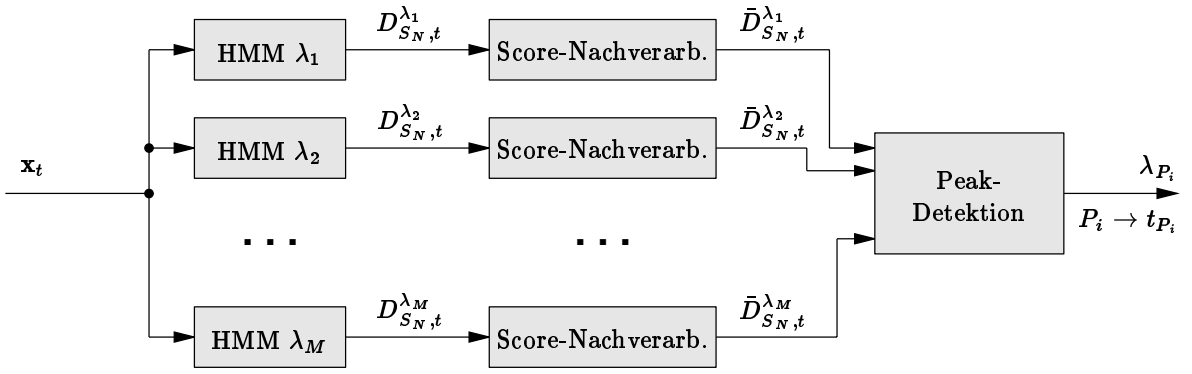


Bild 9.4: Systemkomponenten für die einstufige kontinuierliche Erkennung

(s. Kap. 9.3.1) und ein ständiger Beginn neuer Viterbi-Pfade ermöglicht werden (s. Kap. 9.3.2). Erst eine Nachverarbeitung der Ausgangsscores (s. Kap. 9.3.4) läßt eine zuverlässige Suche von Charakteristika in den Ausgangsscores (s. Kap. 9.3.5) möglich werden.

Die Anordnung der Systemkomponenten für das Spotting ist in Bild 9.4 gezeigt. Die einzelnen Schritte werden nun im folgenden näher erläutert.

9.3.1 Normierung des Viterbi-Algorithmus

Die Modelle beim Spotting werden *trainiert* wie bei der isolierten Erkennung oder dem zweistufigen kontinuierlichen Erkennungsansatz: für jede zu erkennende Bewegung muß ein Modell vorliegen.

Für die Erkennung kommt wiederum der Viterbi-Algorithmus zum Einsatz. Entscheidend ist der Rekursionsschritt, wie er in Gl. (6.19) von Kap. 6.3.3.2 formuliert wurde. Setzt man vereinfachend $F^{\lambda_l}_{S_j,t} = F^{\lambda_l}_{S_j}(\mathbf{x}_t)$, so ergibt sich:

$$D^{\lambda_l}_{S_j,t} = \max_i \left[D^{\lambda_l}_{S_i,t-1} + A^{\lambda_l}_{S_i S_j} \right] + F^{\lambda_l}_{S_j,t}. \tag{9.10}$$

Zu jedem Zeitpunkt wird also in jedem Zustand S_j eines Modells λ_l ein *lokaler Score* $D^{\lambda_l}_{S_j,t}$ als Summe aus der aktuellen Zustandswahrscheinlichkeitsdichte $F^{\lambda_l}_{S_j,t}$ und der maximal möglichen Summe aus einem zeitlich zurückliegenden lokalen Score und der dazughörigen Übergangswahrscheinlichkeit berechnet.

Im Spotting-Betrieb interessiert, wie bei der isolierten Erkennung, der Ausgangsscore $D^{\lambda_l}_{S_N,t}$ der Modelle λ_l. Allerdings gibt es im Unterschied zur isolierten Erkennung keinen definierten Endzeitpunkt, sondern es wird kontinuierlich über die Zeit t beobachtet. Verwendet man die Iteration nach Gl. (9.10), so erkennt man allerdings, daß der Ausgangsscore eines Modells im Mittel ständig anwächst oder abfällt, je nach der mittleren Größe der Zustands-WDF $F^{\lambda_l}_{S_j,t}$. Dies kann auch empirisch bestätigt werden.

Um den mittleren Ausgangsscore zu stabilisieren, muß er daher *normalisiert* werden. Dazu wurden zwei Ansätze untersucht. Die erste Methode der *konstanten Normalisierungslänge* wurde zum ersten Mal in [Mor98b] präsentiert. Der zweite Ansatz einer mitgeführten *lokalen Normalisierungslänge* stammt ursprünglich aus dem Bereich der Spracherkennung [Jun96] und konnte für die kontinuierliche Gestikerkennung übernommen und entscheidend verbessert werden [Mor98c]:

Konstante Normalisierungslänge (N1): Setzt man eine konstante Normalisierungslänge L_n an, so kann der Rekursionsschritt nach Gl. (9.10) als

$$D^{\lambda_l}_{S_j,t} = \left[\max_i \left[D^{\lambda_l}_{S_i,t-1} \cdot L_\mathrm{n} + A^{\lambda_l}_{S_i S_j}\right] + F^{\lambda_l}_{S_j,t}\right] \frac{1}{L_\mathrm{n}+1} \tag{9.11}$$

formuliert werden. Das bedeutet, daß der lokale Score aus dem letzten Iterationsschritt *entnormiert*, mit den lokalen Größen verrechnet und dann wieder mit einer um eins inkrementierten Länge *normiert* wird.

Durch ein solches mitgeführtes „Normalisierungsfenster“ haben die zeitlich weiter zurückliegenden Scores in einem Viterbi-Pfad einen immer weiter abnehmenden Einfluß auf den aktuellen lokalen Score, womit der Ausgangsscore im Mittel stabil bleibt (Beispiel für einen Scoreverlauf s. Bild 9.5a Seite 114) [Mor98b]. Ist die konstante Länge L_n im Extremfall auf 0 gesetzt, dann wird der lokale Score ohne die zurückliegenden Scores aus der Vergangenheit gebildet.

Lokale Normalisierungslänge (N2): Um über die *tatsächliche* Länge des Viterbi-Pfades Buch zu führen, der zu einem bestimmten lokalen Score $D^{\lambda_l}_{S_j,t}$ gehört, wird analog zum lokalen Score eine *lokale Pfadlänge* $L^{\lambda_l}_{S_j,t}$ für jeden Zustand S_j eines Modells λ_l eingeführt. Parallel zur Rekombination lokaler Scores durch den Maximierungsschritt in der Viterbi-Iteration wird dann die lokale Pfadlänge des besten Vorgängerzustandes übernommen und um eins verlängert [Jun96]:

$$\begin{aligned} D^{\lambda_l}_{S_j,t} &= \max_i \left[\frac{D^{\lambda_l}_{S_i,t-1} \cdot L^{\lambda_l}_{S_i,t-1} + A^{\lambda_l}_{S_i S_j} + F^{\lambda_l}_{S_j,t}}{L^{\lambda_l}_{S_i,t-1}+1}\right] \quad \text{und} \\ L^{\lambda_l}_{S_j,t} &= L^{\lambda_l}_{S_k,t-1} + 1 \quad \text{mit } S_k \text{ als bestem Vorgänger von } S_j. \end{aligned} \tag{9.12}$$

Auch hier wird zuerst eine *Entnormierung* des alten Scores vorgenommen, dann werden die lokalen Berechnungen durchgeführt, um anschließend wieder auf die neue Länge zu *normieren*. Es ist sofort ersichtlich, daß der lokale Score stabilisiert wird, da die lokale Pfadlänge mit jedem Iterationsschritt mitwächst. Diese Art der Normierung hat nur dann die erhoffte Wirkung, wenn nach bestimmten Kriterien *neue* Pfade im ersten Zustand beginnen können. Dieser Vorgang wird *Pfad-Triggerung* genannt und in Kap. 9.3.2 ausführlicher behandelt [Mor98c]. Werden solche neuen Pfade zugelassen, so bewirkt die Normierung mit der lokalen Länge, daß für die Maximierungsentscheidung auch lokale Scores vergleichbar bleiben, die *verschiedene Pfadlängen* haben.

Unabhängig von der Art der Normierung ergibt sich das folgende qualitative Verhalten im Ausgangsscore eines Modells, wenn eine zum Modell passende Geste beginnt: im 1. Zustand eines Modells werden sich größere Zustandsdichtewerte ergeben als bei einer nicht passenden Bewegung. Als Folge beginnt der lokale Score im 1. Zustand zu steigen. Dieser Anstieg wird sich je nach Ausprägung über die anderen Zustände bis in den letzten Zustand fortsetzen; dort ist der erhöhte Score allerdings durch die Verknüpfung mit den Übergangswahrscheinlichkeiten abgeschwächt. Aufgrund der Modellstruktur (s. Kap. 6.3.1) dauert das mindestens halb so viele Merkmalszeittakte wie Zustände vorhanden sind. Frühestens nach dieser Zeit wird sich auch im letzten Zustand eine Erhöhung des Scores abzeichnen.

Parallel dazu „wandert“ die gestische Bewegung durch die Zustände: bestimmte Bewegungssegmente werden in aufsteigender Reihenfolge (wiederum bedingt durch die Modellstruktur) die gerade am besten passende Zustands-WDF erhöhen. Auch diese Erhöhungen pflanzen sich unter Umständen sukzessiv durch die Zustände bis zum letzten Zustand fort. Die Abschwächung der erhöhten lokalen Scores bis zum letzten Zustand wird immer geringer ausfallen, da die Kette der dazwischenliegenden Übergangswahrscheinlichkeiten immer kleiner wird.

Am Ende der Bewegung ist der Ausgangsscore am größten, da die erhöhte Zustands-WDF nicht mehr durch die Übergangswahrscheinlichkeiten abgeschwächt wird. Ist die Bewegung vorbei, so nehmen alle Zustandsdichten relativ niedrige Werte an, und der Ausgangsscore klingt wieder schnell ab.

Als resultierender Verlauf ergibt sich ein ständiger Anstieg des Ausgangsscores, der nach einer gewissen Durchlaufzeit durch die Zustände mit Beginn der Bewegung anfängt. Der Score steigt im Verlauf der Bewegung weiter an, der Anstieg flacht aber mit der Zeit immer mehr ab. Mit Ende der Bewegung hat der Score seinen Höchstpunkt erreicht um danach wieder steil abzufallen.

Es ergibt sich somit ein Gipfel oder *Peak* im Ausgangsscore, der charakteristisch für das *Ende* einer passenden Geste ist. Gesten-Spotting wird damit auf die Aufgabe reduziert, Peaks in den Ausgangsscores der Modelle zu finden (s. Kap. 9.3.5). Zunächst werden verschiedene Möglichkeiten besprochen, wie im Zusammenhang mit der Normalisierung N2 neue Viterbi-Pfade getriggert werden können (s. Kap. 9.3.2) und wie eine implizite Verweildauer-Modellierung erreicht wird (s. Kap. 9.3.3). Unabhängig vom Normierungsverfahren müssen die Ausgangsscores vor der Peak-Detektion noch nachverarbeitet werden (s. Kap. 9.3.4).

9.3.2 Triggern neuer Viterbi-Pfade

Die Triggerung steuert den möglichen Neubeginn eines Viterbi-Pfades bei der Normalisierung N2 nach Gl. (9.12) im 1. Zustand. Eine solcher Neubeginn steht im Wettbewerb zur regulären Weiterführung eines Viterbi-Pfades. Eine Triggerung erfolgt zum Trigger-Zeitpunkt t_{tr} immer dann, wenn der lokale Score im 1. Zustand kleiner ist als eine *Score-Schwelle* $D_{\mathrm{tr}}^{\lambda_l}$:

$$D_{S_1,t_{\mathrm{tr}}}^{\lambda_l} < D_{\mathrm{tr}}^{\lambda_l}. \tag{9.13}$$

Die erwünschte Wirkung einer Triggerung kann man sich wie folgt vorstellen: Je kürzer die Pfadlänge, desto schneller und intensiver setzen sich Änderungen der Zustandswahrscheinlichkeitsdichte im Scoreverlauf durch; je länger der Pfad, desto stabiler wird der Scoreverlauf gegenüber schnellen Änderungen der Zustands-WDFs. Damit der lokale Score im Verlauf einer Gestenbewegung schnell ansteigen kann, sollte ein neuer Pfad daher möglichst zu Beginn einer neuen Geste triggern. Es wurden drei Triggermethoden untersucht. Die erste Methode (T1) ist in [Jun96] zu finden, die letzten beiden (T2 und T3) wurden in [Mor98c] vorgestellt:

Passives Triggern (T1): Setzt man die Triggerschwelle auf

$$D_{\mathrm{tr}}^{\lambda_l} = 0 \tag{9.14}$$

und den Anfangsscore des neuen Pfades auf die Triggerschwelle selbst, so erhält man die *passive* Triggerung mit:

$$D_{S_1,t_{\mathrm{tr}}}^{\lambda_l} = D_{\mathrm{tr}}^{\lambda_l} = 0 \tag{9.15}$$

und der initialen Länge

$$L^{\lambda_l}_{S_1,t_{\mathrm{tr}}} = 1. \tag{9.16}$$

Die Grundidee dieses Verfahren ist die, daß die Maximierungs-Entscheidung des Viterbi-Algorithmus nach Gl. (9.12) im 1. Zustand konsistent um den möglichen Seiteneinstieg eines neuen Pfades mit dem logarithmierten Wert der Eintrittswahrscheinlichkeit 1 erweitert wird. Dies ist ein Sonderfall des mit der Triggerbedingung in Gl. (9.13) formulierten allgemeineren Konzeptes.

Aus Sichtweise der Triggerung kann man diese Methode als passiv bezeichnen, weil die Schwelle unveränderlich ist und abgewartet wird, bis der Score $D^{\lambda_l}_{S_1,t}$ im 1. Zustand *unter* die Schwelle fällt. Dies entspricht genau *nicht* der Forderung, daß ein neuer Pfad mit Beginn einer passenden Bewegung anfangen sollte. Außerdem ist der Schwellwert von 0, im Vergleich mit den Scorewerten, wie sie sich im 1. Zustand einstellen, willkürlich gewählt.

Aktives Triggern (T2): Will man erreichen, daß die Triggerung mit Beginn einer passenden Bewegung erfolgt, so muß man die Triggerschwelle derart adaptiv gestalten, daß sie bei Bedarf schneller wachsen kann als der lokale Score des 1. Zustandes und daß so nach Gl. (9.13) die Triggerung ausgelöst wird.

Als eine solche *aktive*, zeitabhängige Triggerschwelle $D^{\lambda_l}_{\mathrm{tr},t}$ eignet sich die *Nachbildung* des Scoreverlaufes im 1. Zustand, wie sie sich bei Verwendung der konstanten Normierung N1 nach Gl. (9.11) ergeben würde:

$$D^{\lambda_l}_{\mathrm{tr},t} = \left[D^{\lambda_l}_{\mathrm{tr},t-1} \cdot L_{\mathrm{s}} + A^{\lambda_l}_{S_1 S_1} + F^{\lambda_l}_{S_1,t}\right] \frac{1}{L_{\mathrm{s}}+1}. \tag{9.17}$$

Nach erfolgter Triggerung wird rückwirkend zum Zeitpunkt $t-1$ der lokale Score im 1. Zustand auf die Länge 1 gesetzt:

$$L^{\lambda_l}_{S_1,t_{\mathrm{tr}}-1} = 1. \tag{9.18}$$

Dadurch wird der zurückliegende lokale Score $D^{\lambda_l}_{S_1,t-1}$ nach Gl. (9.12) automatisch zum Einsprung-Score des neuen Pfades.

Die konstante Normierungslänge L_{s} glättet den Scoreverlauf und beeinflußt das dynamische Verhalten der Triggerschwelle $D^{\lambda_l}_{\mathrm{tr},t}$: wird die Zustands-WDF des 1. Zustandes $F^{\lambda_l}_{S_1,t}$ größer, weil eine für das Modell λ_l passende Geste beginnt, und ist die Glättungslänge L_{s} klein zur lokalen Pfadlänge $L^{\lambda_l}_{S_1,t}$, die sich im 1. Zustand bis zu diesem Zeitpunkt gebildet hat, so kann die Schwelle $D^{\lambda_l}_{\mathrm{tr},t}$ schneller steigen als der lokale Score im 1. Zustand. Die Triggerbedingung Gl. (9.13) ist somit erfüllt und eine neuer Pfad beginnt. Da dann — bei entsprechender Justierung von L_{s} — die Länge dieses Pfades kurzfristig kleiner ist als die konstante Normierungslänge L_{s} der Triggerschwelle, kann der Score die Triggerschwelle übersteigen und der neue Pfad fortgesetzt werden.

Zusätzlich wird zum Triggerzeitpunkt t_{tr} ein *Score-Eingangsgewicht* W zum lokalen S_1-Score $D^{\lambda_l}_{S_1,t_{\mathrm{tr}}}$ dazu addiert, so daß sich der Triggerscore zu

$$\hat{D}^{\lambda_l}_{S_1,t_{\mathrm{tr}}} = D^{\lambda_l}_{S_1,t_{\mathrm{tr}}} + W \tag{9.19}$$

ergibt (mit dem Score $\hat{D}^{\lambda_l}_{S_1,t_{\text{tr}}}$ wird dann weiter gearbeitet). Mit diesem Eingangsgewicht kann der Triggerscore gegenüber den „umliegenden“ Scores anderer Zustände vergrößert werden, so daß sich der neue Pfad besser durch die folgenden Zustände durchsetzen kann. Die Wirkung dieses Eingangsgewichtes läßt durch die Normierung in Gl. (9.12) mit zunehmendem zeitlichem Abstand zum Triggerzeitpunkt nach, da W nur einmal addiert wird.

Permanentes Triggern (T3): Es ist auch möglich, im 1. Zustand zu *jedem* Zeitpunkt t einen neuen Pfad zu triggern, so daß im Vergleich zu T2 auf keinen Fall ein Gestenanfang verpaßt werden kann. Verwendet man wiederum das Score-Eingangsgewicht W wie bei der aktiven Triggerschwelle unter T2, so ergibt sich für den lokalen Score im 1. Zustand:

$$\hat{D}^{\lambda_l}_{S_1,t} = D^{\lambda_l}_{S_1,t} + W \tag{9.20}$$

mit einer konstanten Pfad-Länge von $L^{\lambda_l}_{S_1,t-1} = 1$.

9.3.3 Verweildauer-Modellierung über Beeinflussung der lokalen Normierungslänge

HMMs modellieren die mittlere Verweildauer in einem Zustand implizit durch eine exponentiell abklingende Verteilung [Rab89]. Sehr rechenaufwendige Verweildauer-Modellierungen bilden ein realistischeres Verhalten nach und können die Modellierungsfähigkeit der Modelle stark verbessern.

Die Normierungslänge $L^{\lambda_l}_{S_i,t}$ kann nun als *Funktion* aufgefaßt und so *verzerrt* werden, daß sich eine einfache implizite Modellierung für die Verweildauer ergibt [Mor99]. Ausgangspunkt ist die mittlere Verweildauer $\bar{\tau}^{\lambda_l}_{S_i}$ in einem Zustand S_i, $i = 1, \ldots, N-1$ nach Gl. (6.14) auf Seite 51. Im letzten Zustand S_N ist die Verweildauer nicht definiert, da in einem Links-Rechts-Modell stets $a^{\lambda_l}_{S_N S_N} = 1$ ist. Die mittlere Verweildauer im *ganzen Modell* läßt sich dann annähern durch:

$$\bar{T}^{\lambda_l} = \sum_{i=1}^{N} \bar{\tau}^{\lambda_l}_{S_i} \approx \frac{N}{N-1} \sum_{i=1}^{N-1} \frac{1}{1 - a^{\lambda_l}_{S_i S_i}}. \tag{9.21}$$

Durch den Faktor $N/(N-1)$ soll die mittlere Verweildauer der ersten $N-1$ Zustände um einen Zustand erweitert werden, weil die Verweildauer im letzten Zustand nicht angebbar ist. Mit dieser Verweildauer wird nun die Normalisierunglänge $L^{\lambda_l}_{S_i,t}$ modifiziert:

$$\tilde{L}^{\lambda_l}_{S_i,t} = \begin{cases} L^{\lambda_l}_{S_i,t} & \text{für} \quad L^{\lambda_l}_{S_i,t} < \bar{T}^{\lambda_l} \\ v \cdot (L^{\lambda_l}_{S_i,t} - \bar{T}^{\lambda_l}) + \bar{T}^{\lambda_l} & \text{für} \quad L^{\lambda_l}_{S_i,t} \geq \bar{T}^{\lambda_l} \end{cases}. \tag{9.22}$$

Wendet man diese Funktion mit einem Parameter v größer als 1 an, so bleiben die Pfadlängen unverändert, bis die Pfadlänge die mittlere HMM-Verweildauer $\bar{T}^{\lambda_l}$ überschreitet: dann werden die Viterbi-Pfade künstlich verlängert. Dadurch werden lange Pfade benachteiligt und kürzere Pfade haben es leichter, die langen Pfade zu verdrängen.

Die Auswirkungen dieser optionalen Verweildauer-Modellierung wird in Anh. D.2 untersucht.

9.3.4 Nachverarbeitung der Ausgangsscores

9.3.4.1 Glättung

Wie in Kap. 9.3.1 dargelegt, äußert sich das *Ende* einer Geste durch einen *Peak* im Ausgangsscore. Unabhängig von der Wahl des Normalisierungsverfahrens, ergibt sich aber ein sehr „zerklüfteter" Ausgangsscore, der neben den Haupt- noch viele Nebenpeaks enthält (s. Bild 9.5a und c).

Damit der Hauptpeak durch die nachfolgende Peak-Detektionsstufe (s. Kap. 9.3.5) einfacher gefunden werden kann, müssen die Nebenpeaks durch *Glättung* des Ausgangsscores beseitigt werden. Dies geschieht einfach durch eine Mittelung des Scores im Bereich des *Glättungsintervalles* $[t-\tau_{\mathrm{s}}^{\mathrm{b}},t+\tau_{\mathrm{s}}^{\mathrm{e}}]$. Der gemittelte Ausgangs-Score $\bar{D}_{S_N,t}^{\lambda_l}$ ergibt sich dann zu:

$$\bar{D}_{S_N,t}^{\lambda_l} = \frac{1}{\tau_{\mathrm{s}}^{\mathrm{e}}+\tau_{\mathrm{s}}^{\mathrm{b}}+1} \sum_{\tau=-\tau_{\mathrm{s}}^{\mathrm{b}}}^{\tau_{\mathrm{s}}^{\mathrm{e}}} D_{S_N,t+\tau}^{\lambda_l} \tag{9.23}$$

Die Glättung ist unabdingbar, hat aber zwei Nachteile:

1. Läßt man den Ende-Glättungswert $\tau_{\mathrm{s}}^{\mathrm{e}}$ größer Null werden, so ergibt sich eine systematische Verzögerung des Ausgangsscores um die Größe von $\tau_{\mathrm{s}}^{\mathrm{e}}$. Deshalb wird man diese Grenze möglichst klein wählen oder zu Null setzen.

2. Will man trotzdem eine ausreichende Glättung erreichen, so muß der Anfangs-Glättungswert $\tau_{\mathrm{s}}^{\mathrm{b}}$ relativ groß gewählt werden. Durch diese asymmetrische Mittelung verschieben sich aber die Peaks auf der Zeitachse nach rechts (diese Verschiebung ist im Vergleich der Bilder 9.5a und c mit den Bildern 9.5b und d gut sichtbar).

9.3.4.2 Peak-Verstärkung

Die Peaks haben, wie in den Score-Verläufen in Bild 9.5 sichtbar, oft einen sehr flachen Verlauf. Dieser Verlauf kann sich ergeben, wenn die Geste an ihrem Ende angelangt ist und die Hand in dieser Position beinahe unbeweglich verharrt: der Score bleibt dann in Sättigung, bis die Anschlußbewegung zur nächsten Gesten ausgeführt wird. Falls die Verweildauer-Modellierung (s. Gl.(9.22)) angewendet wird, sinkt der Score zwar bei einer längeren Bewegungspause allmählich wieder ab, es entstehen aber trotzdem immer noch sehr langgezogene Peaks.

Die Form der Peaks kann etwas günstiger gestaltet werden, indem man zum geglätteten Score noch die durch einen Mischungsfaktor C_{mix} gewichtete zeitliche Differenz des geglätteten Scores addiert:

$$\tilde{D}_{S_N,t}^{\lambda_l} = \bar{D}_{S_N,t}^{\lambda_l} + C_{\mathrm{mix}} \cdot \left[\bar{D}_{S_N,t}^{\lambda_l} - \bar{D}_{S_N,t-1}^{\lambda_l}\right]. \tag{9.24}$$

Dadurch entstehen an der vorderen Flanke der Peaks kontrollierte Überschwinger: das Peak-Maximum ist somit zeitlich vorverlagert und die Peaks selbst sind etwas deutlicher ausgeprägt (s. Bild 9.6). Der Mischungsfaktor C_{mix} muß aber vorsichtig dosiert werden, da sonst der Glättungseffekt wieder rückgängig gemacht wird und die Nebenpeaks wieder verstärkt werden.

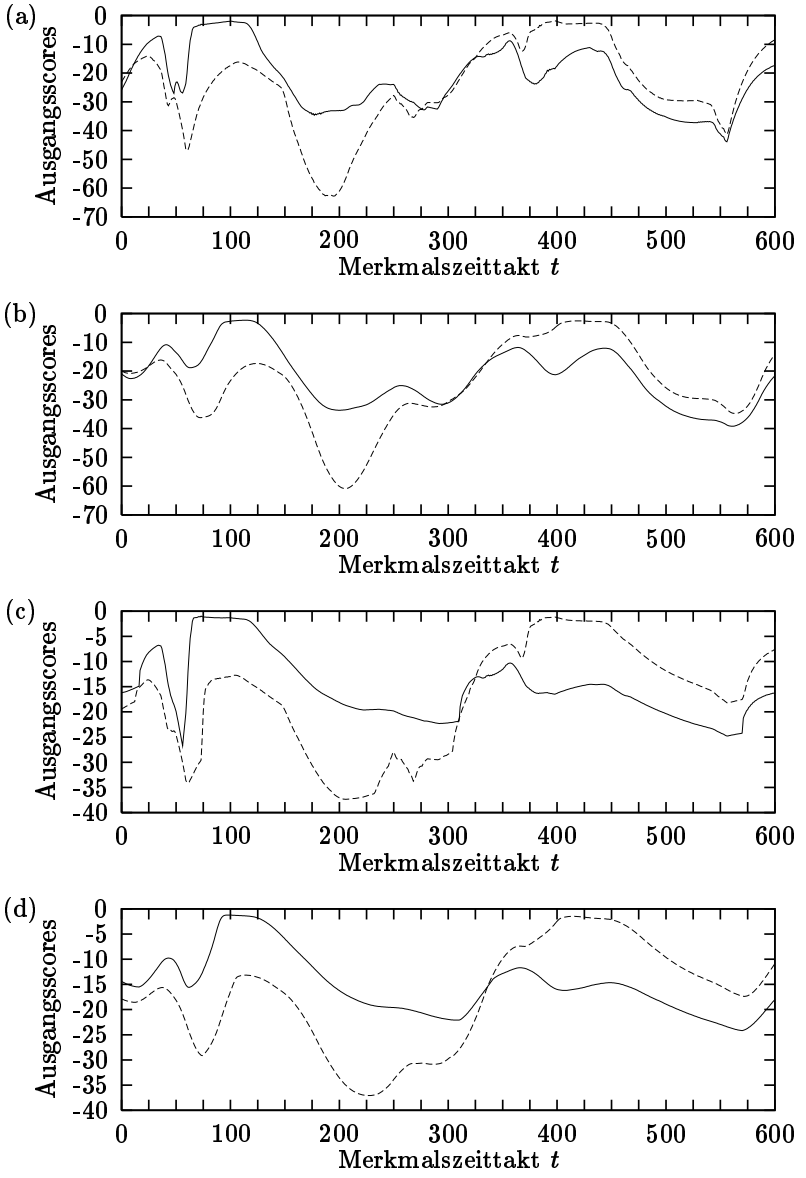


Bild 9.5: Beispiel für den Ausgangsscore zweier Modelle λ_1 (durchgezogene Linie) und λ_2 (gestrichelte Linie); die jeweils passenden Gesten enden zum Zeitpunkt $t_{P_1} = 100$ und $t_{P_2} = 400$: (a) N1 ($L_\mathrm{n} = 20$), (b) N1 geglättet ($L_\mathrm{n} = 20$, $\tau_\mathrm{s}^\mathrm{b} = 30$, $\tau_\mathrm{s}^\mathrm{e} = 0$), (c) N2 ($W = 45$), (d) N2 geglättet ($W = 45$, $\tau_\mathrm{s}^\mathrm{b} = 30$, $\tau_\mathrm{s}^\mathrm{e} = 0$)

9.3.5 Regeln für die Peak-Suche

Nach der Glättung (s. Gl. (9.23)) und einer eventuellen Peak-Verstärkung (s. Gl. (9.24)) erfolgt die eigentliche *Peak-Detektion*. Sie liefert Peaks P_i, die durch den Modellindex λ_{P_i} — also den Index der vermuteten Geste — und den Peak-Zeitpunkt t_{P_i} gekennzeichnet sind. Ein Peak P_i ist das Ergebnis der Anwendung der vier Regeln zur Peaksuche P1–P4 [Mor98b]. Der geglättete Ausgangsscore $\bar{D}^{\lambda_{P_i}}_{S_N,t_{P_i}}$ des Modells λ_{P_i} zum Zeitpunkt t_{P_i}

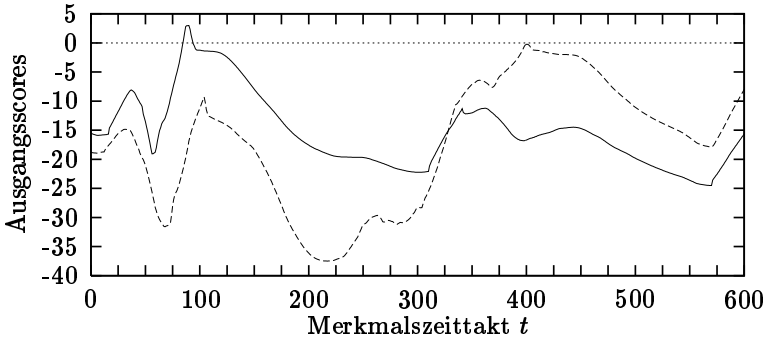


Bild 9.6: Beispiel für Ausgangsscore bei Anwendung der Peak-Verstärkung (N2 geglättet ($W = 45$, $\tau_{\mathrm{s}}^{\mathrm{b}} = 30$, $\tau_{\mathrm{s}}^{\mathrm{e}} = 0, C_{\mathrm{mix}} = 3$)

Relatives Maximum (P1): muß im Peak-Suchintervall $[t - \tau_{\mathrm{p}}^{\mathrm{b}}, t + \tau_{\mathrm{p}}^{\mathrm{e}}]$ maximal sein:

$$\bar{D}^{\lambda_{P_i}}_{S_N, t_{P_i} - \tau} \geq \bar{D}^{\lambda_{P_i}}_{S_N, t_{P_i} - \tau - 1} \quad \text{für} \quad \tau = 0, \ldots, \tau_{\mathrm{p}}^{\mathrm{b}} - 1 \quad \text{und} \tag{9.25}$$

$$\bar{D}^{\lambda_{P_i}}_{S_N, t_{P_i} + \tau} \geq \bar{D}^{\lambda_{P_i}}_{S_N, t_{P_i} + \tau + 1} \quad \text{für} \quad \tau = 0, \ldots, \tau_{\mathrm{p}}^{\mathrm{e}} - 1; \tag{9.26}$$

Absolutes Maximum (P2): muß größer als die Scores aller anderen Modelle sein:

$$\bar{D}^{\lambda_{P_i}}_{S_N, t_{P_i}} > \bar{D}^{\lambda_k}_{S_N, t_{P_i}} \quad \text{für} \quad k = 1, \ldots, M \quad \text{und} \quad \lambda_k \neq \lambda_{P_i}; \tag{9.27}$$

Rückweisungsschwelle (P3): muß größer sein als eine modellspezifische Rückweisungsschwelle $D_{\mathrm{rw}}^{\lambda_{P_i}}$:

$$\bar{D}^{\lambda_{P_i}}_{S_N, t_{P_i}} > D_{\mathrm{rw}}^{\lambda_{P_i}}; \tag{9.28}$$

Mindestabstand (P4): muß einen minimalen zeitlichen Abstand τ_{dist} vom letzten gefundenen Peak haben:

$$t_{P_i} - t_{P_{i-1}} \geq \tau_{\mathrm{dist}}. \tag{9.29}$$

Die modellspezifische Rückweisungsschwelle der Regel P3 wird auf eine einzige *relative* Rückweisungsschwelle S_{rel} zurückgeführt, die über die modellspezifischen minimalen und maximalen Scores definiert ist:

$$D_{\mathrm{rw}}^{\lambda_{P_i}} = \bar{D}_{\mathrm{max}}^{\lambda_{P_i}} - S_{\mathrm{rel}} \cdot \left[\bar{D}_{\mathrm{max}}^{\lambda_{P_i}} - \bar{D}_{\mathrm{min}}^{\lambda_{P_i}} \right]. \tag{9.30}$$

Die Extremscores $\bar{D}_{\mathrm{min}}^{\lambda_l}$ und $\bar{D}_{\mathrm{max}}^{\lambda_l}$ müssen für jedes Modell λ_l in einem eigenen Trainingslauf aus dem Verlauf des geglätteten Scores $\bar{D}^{\lambda_l}_{S_N, t}$ bestimmt werden.

Der Glättungsprozeß und die Peak-Detektionsregeln haben sich als sehr wichtig für die Robustheit und Leistungsfähigkeit des gesamten Spotting-Prozesses erwiesen [Mor98b].

9.4 Prinzipieller Vergleich der Verfahren

Es läßt sich schon im Vorfeld angeben, daß der zweistufige und der einstufige Ansatz nicht dasselbe leisten können: während mit dem Spotting-Verfahren aktiv modellierte Bewegungen herausgearbeitet und nicht-modellierte Bewegungen unterdrückt werden

können, wird die Bewegungsdetektionsstufe des zweistufigen Verfahrens ohne eine Bewertungsmöglichkeit jede einigermaßen ausgeprägte Bewegung durchlassen. Dagegen handelt es sich bei der Klassifikationsstufe des zweistufigen Ansatzes um eine robuste, bewährte und sehr leistungsfähige isolierte Erkennung (s. Kap. 8), die eventuelle Unzulänglichkeiten bei der Bewegungsdetektion wieder ausgleicht.

Wie entscheidend diese prinzipielle Eigenschaften für die Gesamtleistung der System sind, läßt sich erst durch die Evaluierung dieser System durch reale Daten angeben. Dies geschieht im nächsten Kapitel.

Kapitel 10

Evaluierung der Erkennung verbundener Gesten

10.1 Vorbemerkungen

Bei der Evaluierung der Erkennung kontinuierlicher Gesten steigt die Anzahl der variierbaren Parameter im Vergleich zur isolierten Erkennung noch einmal deutlich an. Um den Untersuchungsaufwand und den benötigten Platz für die Darstellung in Grenzen zu halten, werden in diesem Kapitel nur die spezifischen Parameter der kontinuierlichen Erkennung untersucht und optimiert.

Alle Untersuchungen werden mit den optimalen Merkmalsvektoren der isolierten Erkennung durchgeführt: deren Komponenten enthalten grauwertgewichtete HMIs, deren zeitlichen Differenzen und zeitliche Differenzen von Trajektorienwerten.

Die Betrachtungen für die praktische Einsatzfähigkeit werden für die kontinuierliche Erkennung nicht mehr wiederholt: an dieser Stelle wird davon ausgegangen, daß sich die Erkenntnisse aus den Untersuchungen für die isolierte Erkennung sinngemäß übertragen lassen. Im einzelnen heißt dies: es wird ausschließlich mit der vollen Bildwiederholrate und der vollen Halbbildgröße gearbeitet. Nur Datensätze, die keine Bewegungsunschärfe enthalten, werden verwendet. Weiterhin kommen nur die Daten einer Person zum Einsatz und auf die Bestimmung der kategorialen Erkennungsrate wird verzichtet. Erneute Messungen von Rechenzeiten sind nicht erforderlich, da die Echtzeitfähigkeit der Algorithmen mit den Messungen aus Kap. 8 abgeschätzt werden können (s. Kap. 11).

10.2 Trainings- und Testmaterial

Für die meisten Auswertungen dieses Kapitels werden sowohl die Übungsdaten (s. Kap. C.1.1) als auch die Dialogdaten (s. Kap. C.1.2) verwendet. Diese Daten stehen für zwei Extreme im Gestenkontext (s. Kap. 10.3), so daß die Ergebnisse jeweils unterschiedliche Interpretationen erlauben.

Lediglich für die grundlegende Evaluierung und Auswahl der verschiedenen Normalisierungs- und Triggerverfahren der einstufigen Erkennung (s. Kap. 10.7.1) werden die Analysedaten herangezogen, die zu kontinuierlichen Datensätzen synthetisiert wurden (s. Kap. C.1.3). Damit können die Abfolge und die zeitlichen Abstände der Gesten gezielt gesteuert werden.

10.3 Untersuchungen zum Gestenkontext

Prinzipielle Überlegungen zum Gestenkontext werden im Anh. C.2 angestellt. Während der Kontext bei der isolierten Erkennung lediglich für das Festlegen der Grenzen beim Labeln wichtig war, spielt er bei der kontinuierlichen Erkennung eine wesentlich weitreichendere Rolle: im Gestenkontext unterscheiden sich nämlich die Datensätze ganz erheblich:

- Die Übungsdaten stehen für Gesten mit geringer Kontextabhängigkeit, die gleichzeitig mit einer gewissen Sorgfalt und relativ gleichmäßig ausgeführt wurden (s. Kap. C.1.1).
- Die Dialogdaten stehen für Gesten mit hoher Kontextabhängigkeit. Darüberhinaus lag die Konzentration mehr auf der Anwendung und der Zielsetzung der Aufgabenstellung als auf der gewissenhaften Ausführung der Gesten (s. Kap. C.1.2).

Um die Bedeutung des Kontextes für verschiedene Datensätze untersuchen zu können, wird der sogenannte *Überhang* eingeführt. Hat eine gelabelte Geste die untere Grenze $t^{\mathrm{b}}_{L_i}$ und die obere Grenze $t^{\mathrm{e}}_{L_i}$, so führt dies zum Labelintervall $L_i = [t^{\mathrm{b}}_{L_i}, t^{\mathrm{e}}_{L_i}[$. Dieses Intervall kann durch Subtraktion des Beginn-Überhangs o^{b} und die Addition des Ende-Überhangs o^{e} entsprechend zu $[t^{\mathrm{b}}_{L_i} - o^{\mathrm{b}}, t^{\mathrm{e}}_{L_i} + o^{\mathrm{e}}[$ erweitert werden. Sind beide Überhänge gleich groß, so werden sie nur mit $o = o^{\mathrm{b}} = o^{\mathrm{e}}$ bezeichnet. Dann spricht man auch von einem symmetrischen Überhang.

Bei den Untersuchungen zur isolierten Erkennung in Kap. 8 wurde immer der Überhang $o = 0$ verwendet, so daß nur die Kerngesten für die Evaluierung herangezogen wurden. Erhöht man nun diesen Überhang, so verläßt man den Bereich der Kerngesten und bezieht auch die Vor- und Nachbereitungsbewegung in Training und Erkennung ein (s. Kap. 2.2 und Kap. C.2). Alle Bewegungen außerhalb der Kernbewegung werden aber durch den Kontext bestimmt, in dem sich die Geste befindet. Deshalb wirkt sich das Einbeziehen eines Überhanges auch bei den unterschiedlichen Datensätzen unterschiedlich aus. Diese Auswirkungen werden zunächst für die *isolierte Erkennung* untersucht.

Tabelle 10.1 zeigt Erkennungsraten r für die Übungsdaten bei symmetrischer Variation des Überhanges o. Die Erkennungsraten steigen mit zunehmendem Überhang bis zu $o = 10$ deutlich an. Dann bleiben sie auf diesem Niveau oder fallen ganz leicht wieder ab. Der Effekt geht mit steigender Codebuchgröße etwas zurück. Da bei den Übungsdaten längere Serien gleicher Gesten hintereinander gemacht wurden, ähneln sich die Vor- und Nachbereitungsbewegungen oft so stark, daß sich eine Vergrößerung des Überhangs positiv auf die Erkennungsrate auswirkt. Mit steigender Größe des Überhangs wird die Modellierung durch die HMMs bei gleicher Zustandszahl wieder schwieriger, was einem weiteren Anstieg der Erkennungsraten entgegen wirkt.

Das Überhang-Verhalten ist bei den Dialogdaten erwartungsgemäß verschieden (s. Tabelle 10.2). Hier steigen die Erkennungsraten mit steigendem Überhang nur sehr leicht an, zeigen etwa bei $o = 5$ ein Maximum, um dann mit weiter steigendem Überhang wieder relativ stark abzufallen. Die Erklärung liegt im ständig wechselnden Kontext, in den die Gesten in einem echten gestischen Dialog eingebettet ist. Hierbei unterscheiden sich Vor- und Nachbereitungsbewegungen in Abhängigkeit von Vorgänger- und Folgegeste sehr stark, so daß sich ein größerer Überhang ab einem gewissen Grad eher negativ auswirkt. Daß die Erkennungsraten auch beim Überhang 0 schlechter sind als bei den Übungsdaten, läßt sogar den Schluß zu, daß selbst die Kerngesten durch den wechselnden Kontext deutlich verändert werden. Allerdings werden die schlechteren Erkennungsraten auch durch

L	Überhang o				
	0	5	10	15	20
2	12,38	12,38	14,82	14,45	11,82
4	25,70	29,64	35,46	37,15	30,02
8	38,84	49,16	59,85	67,17	63,79
16	76,55	80,68	91,56	90,24	90,24
32	92,31	95,31	97,00	96,62	96,25
64	97,94	98,12	98,31	98,12	98,50
128	97,94	99,25	99,25	99,06	99,25
256	99,06	98,87	99,44	99,25	99,25

Tabelle 10.1: Erkennungsraten r (in %) für Variation des Überhangs o und der Codebuchgröße L für die Übungsdaten (grauwertgewichtete HMI-Merkmale, Zustandszahl $N = 9$)

L	Überhang o				
	0	5	10	15	20
2	16,34	13,14	18,83	16,87	21,31
4	45,83	35,17	35,17	35,17	27,71
8	65,90	63,23	63,59	54,88	55,77
16	80,11	82,59	75,84	69,80	65,72
32	83,13	85,61	84,01	75,31	76,38
64	89,17	91,65	90,23	84,90	80,28
128	92,54	93,61	93,07	89,70	86,68
256	92,01	93,96	94,32	91,30	92,54

Tabelle 10.2: Erkennungsraten r (in %) für Variation des Überhangs o und der Codebuchgröße L für die Dialogdaten (grauwertgewichtete HMI-Merkmale, Zustandszahl $N = 9$)

die stärker schwankende Ausführungsgeschwindigkeit und die größere Nachlässigkeit beim Gestikulieren bei gleichzeitiger Konzentration auf die Anwendung verursacht.

Wie sehr sich die Gesten der Übungs- und der Dialogdaten unterscheiden, läßt sich erkennen, wenn mit den Übungdaten trainiert und mit den Dialogdaten die Erkennung durchgeführt wird (s. Tabelle 10.3). Hier halbieren sich im Vergleich zum konsistenten Training die Erkennungsraten fast. Ein steigender Überhang bewirkt weitere drastische Verringerungen der Erkennungsraten. Damit wird einmal mehr verdeutlicht, wie stark sich der Kontext der Gesten auf die Ausführung der Gesten auswirkt.

Aus den Ergebnissen mit der isolierten Erkennung kann geschlossen werden, daß sich mit den Übungsdaten auch bei der kontinuierlichen Erkennung bessere Erkennungsergebnisse erzielen lassen, als mit den Dialogdaten. Würden Gesten im Dialogzusammenhang eingesetzt, die mit Rücksicht auf das System sorgfältiger ausgeführt werden, so ist zu erwarten, daß die Erkennungsraten zwischen denen für die Übungs- und Dialogdaten liegen. Somit bilden die beiden zur den folgenden Evaluierungen herangezogenen Datensätze obere und untere Abschätzungen für die mit einem realen System erzielbare Erkennungsleistung.

L	Überhang o				
	0	5	10	15	20
2	19,18	17,23	17,23	15,81	16,87
4	35,17	27,00	22,74	20,78	21,49
8	37,12	36,77	23,09	18,83	17,76
16	38,90	40,14	23,62	18,47	15,81
32	47,78	43,69	38,19	22,38	18,12
64	55,24	45,29	33,21	28,60	30,02
128	52,40	48,31	37,83	30,02	23,45
256	52,75	46,36	38,90	30,91	22,02

Tabelle 10.3: Erkennungsraten r (in %) für Variation des Überhangs o und der Codebuchgröße L für das Training mit den Übungsdaten und die Erkennung mit den Dialogdaten (grauwertgewichtete HMI-Merkmale, Zustandszahl $N = 9$)

10.4 Ablauf von Training und Erkennung

Das Training läuft exakt so ab wie bei der isolierten Erkennung in Kap. 8.3. Für das Training müssen daher selbstverständlich genauso manuell gelabelte Daten zur Verfügung stehen (Einzelheiten s. Kap. C.1.1 und Kap. C.1.2). Die Labelgrenzen werden nach der in Kap. 10.3 beschriebenen Vorgehensweise um den Anfangs- und Endeüberhang o^{b} und o^{e} verschoben. Mit diesen Daten werden Mittelwerts- und Varianzvektor für die HMM-Vorverarbeitung, das Codebuch und schließlich die eigentlichen Modelle λ_l gebildet.

Beim Spotting-Verfahren müssen zusätzlich die maximalen und minimalen Score-Werte $\bar{D}^{\lambda_l}_{\mathrm{max}}$ und $\bar{D}^{\lambda_l}_{\mathrm{min}}$ trainiert werden (s. Kap. 9.3.5). Dazu werden die isolierten Trainingsdaten zu einer kontinuierlichen Sequenz direkt aneinandergehängt. Mit diesen kontinuierlichen Trainingsdaten werden dann die trainierten Modelle beaufschlagt, so daß Maxima und Minima bestimmt werden können. Die sich ergebenden maximalen und minimalen Score-Werte bei der Erkennung können natürlich etwas abweichen. Dies bedingt, daß die relative Schwelle nach Gl. (9.30) auf Seite 115 unter Umständen auch negative Werte annehmen muß, um die Falschakzeptanzrate (s. Kap. 10.5) auf 0 zu drücken.

Zur Erkennung mit den Übungs- und Dialogdaten wird jeweils der *gesamte* kontinuierliche Datensatz benutzt. Dadurch sind Trainings- und Erkennungsdaten nicht mehr vollständig getrennt. Eine solche Trennung wäre wünschenswert, sie ist aber beim vorhandenen Datenmaterial nicht möglich bzw. sinnvoll (s. Angaben zum Übungsdatensatz in Kap. C.1.1 und zum Dialogdatensatz in Kap. C.1.2; bei den in Kap. C.1.3 beschriebenen Analysedaten ist dagegen eine vollständige Trennung von Training und Erkennung möglich). Je nach Datensatz sind jedoch nur etwa die Hälfte bzw. ein Drittel der Gesten vom Training her bekannt. Damit sind die Ergebnisse nicht direkt mit der isolierten Erkennung vergleichbar. Es wird sich allerdings bei der Definition der Evaluierungskriterien im nächsten Unterkapitel zeigen, daß die Ergebnisse aufgrund der neu hinzukommenden Zeitkomponente vom Prinzip her schon nicht mehr vergleichbar sind. Die Vergleichbarkeit gilt aber auf jeden Fall für die verschiedenen kontinuierlichen Verfahren untereinander.

Die von der Bewegungsdetektion gefundenen Bewegungsintervalle des zweistufigen Ansatzes werden wiederum um den Anfangs- und Ende-Überhang o^{b} und o^{e} aufgeweitet, wie dies beim Training der Modelle geschehen ist. Bei der einstufigen Erkennung kann der Überhang nicht explizit berücksichtigt werden. Er wirkt sich allerdings indirekt auf die Lage der Peaks aus.

Zwar ist für den Erkennungsvorgang kein manuelles Labeln mehr erforderlich. Allerdings müssen die Gestenlabel auch hier vorliegen, damit die Erkennungsleistung evaluiert werden kann.

10.5 Evaluierungskriterien

Die kontinuierliche Erkennung liefert nun abhängig vom verwendeten Ansatz zusätzlich zum Gestenindex λ_{B_i} bzw. λ_{P_i} einen Zeitpunkt $t^{\mathrm{e}}_{B_i}$ bzw. t_{P_i}, der das Ende der Geste angibt (vgl. zweistufiger Ansatz in Kap. 9.2.2 mit einstufigem Ansatz in Kap. 9.3.5). Die folgende Darstellung orientiert sich am Spotting-Verfahren, so daß das Ergebnis der Erkennung mit Wertepaaren der Form $\{\lambda_{P_i}, t_{P_i}\}, i = 1, \ldots, E$ beschrieben werden kann. In alle folgenden Evaluierungsformeln können aber ebenso die Größen λ_{B_i} und $t^{\mathrm{e}}_{B_i}$ für die zweistufige Erkennung eingesetzt werden. Die tatsächlich vorhandenen Gesten g_{L_j} sind mit Labeln L_j bezeichnet und enden zum Zeitpunkt $t^{\mathrm{e}}_{L_j}$ (s. Kap. 10.3). Sie lassen sich als Label-Wertepaare $\{g_{L_j}, t^{\mathrm{e}}_{L_j}\}, j = 1, \ldots, G$ schreiben.

Für die Evaluierung der Erkennung wird einem Erkennungs-Wertepaar $\{\lambda_{P_i}, t_{P_i}\}$ das zeitlich am nächsten liegende Label-Wertepaar $\{g_{L'_i}, t^{\mathrm{e}}_{L'_i}\}$ zugeordnet:

$$\{\lambda_{P_i}, t_{P_i}\} \rightarrow \{g_{L'_i}, t^{\mathrm{e}}_{L'_i}\} \quad \text{mit} \quad L'_i = L'(P_i) = \underset{L_j}{\operatorname{argmin}} |t_{P_i} - t^{\mathrm{e}}_{L_j}|, \; j = 1, \ldots, G. \tag{10.1}$$

Damit läßt sich für eine erkannte Geste $\{\lambda_{P_i}, t_{P_i}\}$ eine *Erkennungsverzögerung*

$$\tau^{\mathrm{v}}_{P_i L'_i} = t_{P_i} - t^{\mathrm{e}}_{L'_i} \tag{10.2}$$

angeben. Die in Gl. (10.1) ausgedrückte Zuordnung macht nur Sinn, wenn eine gewisse Erkennungsverzögerung nicht überschritten wird. Ab einem gewissen Abstand *kann* kein Zusammenhang mehr zwischen einem Erkennungs- und einem Label-Wertepaar bestehen, selbst wenn sich durch Gl. (10.1) immer eine Zuordnung herstellen läßt. Diese Schwelle wird betragsmäßig mit der *maximalen Erkennungsverzögerung* $\tau^{\mathrm{v}}_{\max}$ festgelegt, wodurch ein maximal zulässiges Erkennungsintervall $-\tau^{\mathrm{v}}_{\max} \leq \tau^{\mathrm{v}}_{P_i L'_i} \leq \tau^{\mathrm{v}}_{\max}$ im Einzugsbereich eines Labels L'_i gebildet wird. Somit gilt ein Erkennungs-Wertepaar $\{\lambda_{P_i}, t_{P_i}\}$ als korrekt erkannt, wenn gilt:

$$\lambda_{P_i} = g_{L'_i} \quad \text{und} \quad |t_{P_i} - t^{\mathrm{e}}_{L'_i}| \leq \tau^{\mathrm{v}}_{\max}. \tag{10.3}$$

Der Wert von $\tau^{\mathrm{v}}_{\max}$ sollte nicht zu klein eingestellt werden; seine Justierung ist aber ansonsten unkritisch. Ein *falscher* Ergebniswert wird innerhalb oder außerhalb des Erkennungsintervalles als Fehler gewertet. Allerdings kann ein korrekter Ergebniswert als falsch interpretiert werden, wenn das Intervall zu eng wird (s. u.).

Label-Wertepaare, für die mindestens ein korrekt erkanntes Erkennungs-Wertepaar existiert, werden mit $\{g_{L^*_k}, t^{\mathrm{e}}_{L^*_k}\}$, $k = 1, \ldots, K$ bezeichnet. Es kann nicht ausgeschlossen werden, daß mehrere korrekt erkannte Gesten demselben Label zugeordnet werden, oder daß — bei umgekehrter Sichtweise — in der Umgebung eines Label-Wertepaares $\{g_{L^*_k}, t^{\mathrm{e}}_{L^*_k}\}$ *mehrere* korrekt erkannte Gesten gefunden werden. Daher existiert zu jedem Label L^*_k, $k = 1, \ldots, K$ eine Menge

$$R_{L^*_k} = \left\{ \{\lambda_{P_i}, t_{P_i}\} \,\middle|\, \lambda_{P_i} = g_{L^*_k} \quad \text{und} \quad |t_{P_i} - t^{\mathrm{e}}_{L^*_k}| \leq \tau^{\mathrm{v}}_{\max}, \; i = 1, \ldots, E \right\} \tag{10.4}$$

von *korrekt* erkannten Gesten, die mindestens ein Element enthält. Darin sei immer ein Erkennungs-Wertepaar $\{\lambda_{P_k^*}, t_{P_k^*}\}$ mit dem geringsten Abstand zu $t^{\mathrm{e}}_{L_k^*}$ ausgezeichnet:

$$\{g_{L_k^*}, t^{\mathrm{e}}_{L_k^*}\} \rightarrow \{\lambda_{P_k^*}, t_{P_k^*}\} \quad \text{mit} \quad P_k^* = P^*(L_k) = \underset{P_i}{\operatorname{argmin}} |t_{P_i} - t^{\mathrm{e}}_{L_k}| \text{ und}$$
$$\{\lambda_{P_i}, t_{P_i}\} \in R_{L_k^*},\ k = 1, \ldots, K. \tag{10.5}$$

Mit den eingeführten Größen lassen sich nun vier quantitative Evaluierungsmaße definieren, die für die Beurteilung der kontinuierlichen Erkennung herangezogen werden können. Beim zweistufigen Ansatz wird dabei die *Gesamt*erkennungsleistung über beide Stufen verstanden. Die Schreibweise wird durch die Einführung der Fensterfunktion

$$w(\tau, \tau^{\mathrm{v}}_{\max}) = \begin{cases} 1 & \text{für} \quad -\tau^{\mathrm{v}}_{\max} \leq \tau \leq \tau^{\mathrm{v}}_{\max} \\ 0 & \text{sonst} \end{cases} \tag{10.6}$$

vereinfacht:

Erkennungsrate (E1): Analog zur Gl. (8.3) auf Seite 79 ergibt sich die Erkennungsrate r zusammen mit der zeitlichen Randbedingung aus Gl. (10.3) als das Verhältnis der korrekt klassifizierten Bildsequenzen, die einem Label am nächsten liegen, bezogen auf die Anzahl G der in einer Testsequenz gelabelten Gesten:

$$r = \frac{1}{G} \sum_{k=1}^{K} \delta\left(\lambda_{P_k^*} - g_{L_k^*}\right) w\left(\tau^{\mathrm{v}}_{P_k^* L_k^*}, \tau^{\mathrm{v}}_{\max}\right) = \frac{K}{G}. \tag{10.7}$$

Falschakzeptanzrate (E2): Zur *absoluten Zahl* der fälschlicherweise akzeptierten Gesten f_{A} tragen alle Gesten bei, die zum einen innerhalb des Erkennungsintervalles falsch klassifiziert wurden und die zum anderen unabhängig von ihrer Klassifikation außerhalb jedes Erkennungsintervalles liegen:

$$f_{\mathrm{A}} = \sum_{i=1}^{E} \left(\left[1 - \delta\left(\lambda_{P_i} - g_{L_i'}\right)\right] w\left(\tau^{\mathrm{v}}_{P_i L_i'}, \tau^{\mathrm{v}}_{\max}\right) + \left[1 - w\left(\tau^{\mathrm{v}}_{P_i L_i'}, \tau^{\mathrm{v}}_{\max}\right)\right] \right). \tag{10.8}$$

Während nicht mehr Gesten korrekt erkannt werden können als Label G vorhanden sind, kann die Anzahl der falsch erkannten Gesten die Anzahl der Label G im kontinuierlichen Fall prinzipiell übersteigen. Außerdem steigen die Fehlermöglichkeiten mit der Anzahl der möglichen Modelle M und der Länge der Erkennungssequenz. Daher wird die Falschakzeptanz*rate* in Analogie zum *word spotting* in der Spracherkennung (beispielsweise in [Jun96]) definiert als die Anzahl der falsch erkannten Gesten f_{A} bezogen auf die Größe des Gestenkataloges M und die Gesamtlänge der kontinuierlichen Sequenz $T_{\mathbf{X}}$:

$$f = \frac{f_{\mathrm{A}}}{M \cdot T_{\mathbf{X}}}. \tag{10.9}$$

Es ist üblich, $T_{\mathbf{X}}$ in Stunden anzugeben, so daß f die Einheit 1/h hat.

Mehrfacherkennungsrate (E3): Die Mehrfacherkennungsrate r_{M} wird durch die Gesten gebildet, die innerhalb des Erkennungsintervalles richtig klassifiziert aber nicht bei der Erkennungsrate angerechnet wurden:

$$r_{\mathrm{M}} = \left[\frac{1}{G} \sum_{i=1}^{E} \delta\left(\lambda_{P_i} - g_{L_i'}\right) w\left(\tau^{\mathrm{v}}_{P_i L_i'}, \tau^{\mathrm{v}}_{\max}\right) \right] - r. \tag{10.10}$$

Bei der Mehrfacherkennung handelt es sich prinzipiell um einen Erkennungs*fehler*, dennoch ist dieser Fehler von weitaus geringerer Tragweite als die Falscherkennung. Dies liegt am Dialogkonzept des dieser Arbeit zugrundeliegenden Systems. Eine Mehrfacherkennung wirkt sich dadurch aus, daß dem System mehrmals die korrekte Geste gemeldet wird. Je nach dem inneren Zustandes der Dialogverwaltung kann dabei folgendes auftreten (vgl. Kap. 3.5):

- Da Mehrfacherkennungen innerhalb der kurzen Zeit des Erkennungsintervalles auftreten (sonst werden sie nicht mehr als solche gewertet), wird in den meisten Fällen eine gerade laufende Bewegung im Szenen-Editor abgebrochen, um gleich wieder fortgesetzt zu werden. Der Benutzer wird höchstens wahrnehmen können, daß sich die Zeitdauer, bis die Bewegung von selbst zum Stillstand kommt, etwas verlängert hat.
- Wird der Dialogverwaltung innerhalb einer *sehr kurzen* Zeitdauer dieselbe Geste noch einmal gemeldet, so werden bestimmte Bewegungen (beispielsweise Translationsbewegungen der Hand) um eine Stufe beschleunigt. Allerdings bleibt die korrekte Richtung der Bewegung erhalten. Der Benutzer wird also in einem solchen seltenen Fall feststellen, daß eine Bewegung ohne seine Absicht beschleunigt wird, was außer einer eventuell erforderlichen schnelleren Reaktionszeit keine weiteren Konsequenzen hat.
- In einigen Fällen (beispielsweise bei einer Greifbewegung) wird das System nur auf die erste Geste reagieren und bei einer nachfolgenden identischen Geste keine Reaktion mehr zeigen.

Für die Beurteilung der Leistungsfähigkeit eines Systems spielt die Mehrfacherkennungsrate also keine Rolle. Deswegen werden zum einen keine Maßnahmen ergriffen, die Mehrfacherkennungsraten zu verringern, und zum anderen wird sie auch nicht für den Vergleich der Systeme untereinander herangezogen. Die Mehrfacherkennungsraten werden in den folgenden Evaluierungen allerdings angegeben, weil sie Rückschlüsse darauf zulassen, wie exakt ein Erkennungssystem arbeitet.

Mittlere Erkennungsverzögerung (E4): Die mittlere Erkennungsverzögerung $\bar{\tau}^{\mathrm{v}}$ ist der Mittelwert der Erkennungssverzögerungen aller zur Erkennungsrate angerechneten Gesten:

$$\bar{\tau}^{\mathrm{v}} = \frac{1}{K} \sum_{i=1}^{K} \tau^{\mathrm{v}}_{P_k^* L_k^*}. \tag{10.11}$$

Die mittlere Erkennungsverzögerung ist ein Maß für die durchschnittliche Lage der Bewegungsintervallenden oder der gültigen Gesten-Peaks. Im Online-Betrieb addieren sich dazu jedoch noch weitere Verzögerungen und bilden die mittlere Online-Erkennungsverzögerung $\bar{\tau}^{\mathrm{vo}}$:

- Zweistufiges System: Die Bewegungsdetektion kann das Ende eines Bewegungsintervalles erst erkennen, wenn die Mindestruhezeit $\tau^{\mathrm{e}}_{\mathrm{min}}$ überschritten wurde. Nach erfolgter Detektion muß beim zweistufigen System noch der gewünschte Ende-Überhang o^{e} abgewartet werden, bis das Bewegungssegment an den Erkenner geschickt werden kann. Zur Detektionsverzögerung addiert sich das Maximum beider Werte:

$$\bar{\tau}^{\mathrm{vo}} = \bar{\tau}^{\mathrm{v}} + \max(\tau^{\mathrm{e}}_{\mathrm{min}}, o^{\mathrm{e}}). \tag{10.12}$$

- Spotting-System: Um einen Peak detektieren zu können, ist eine Vorlaufzeit sowohl für die Glättung von $\tau_{\mathrm{s}}^{\mathrm{e}}$ als auch für die nachfolgende Peak-Detektion von $\tau_{\mathrm{p}}^{\mathrm{e}}$ erforderlich. Beide Zeiten müssen zur Erkennungsverzögerung addiert werden:

$$\bar{\tau}^{\mathrm{vo}} = \bar{\tau}^{\mathrm{v}} + \tau_{\mathrm{s}}^{\mathrm{e}} + \tau_{\mathrm{p}}^{\mathrm{e}}. \tag{10.13}$$

  Der Überhang wird lediglich beim Training zu den isolierten Gesten addiert. Da dadurch das Gestenende um den Ende-Überhang verschoben wird, wird sich auch ein entsprechender Peak nach hinten verschieben. Der Überhang muß aber nicht mehr extra zur Erkennungsverzögerung dazuaddiert werden.

Für die getrennte Evaluierung der *Bewegungsdetektion* beim zweistufigen Ansatz werden in Analogie zu den Kriterien E1–E4 weitere Evaluierungswerte benötigt. Dazu wird zu jedem Label ein Detektionsintervall vorgegeben, das über die *maximale Detektionsverzögerung* $\tau_{\mathrm{d,max}}^{\mathrm{v}}$ definiert ist:

Detektionsrate (B1): Zur Bestimmung der Detektionsrate r_{d} werden analog zur Erkennungsrate unter E1 die besten Bewegungsendpunkte innerhalb der Detektionsintervalle gezählt und auf die Anzahl der Label bezogen.

Falschdetektionsrate (B2): Die Falschdetektionsrate f_{d} setzt sich aus den Bewegungsendpunkten zusammen, die in *keinem* Detektionsintervall liegen. Dieser Wert wird im Unterschied zur Falschakzeptanzrate unter E2 relativ zur Labelzahl angegeben.

Mehrfachdetektionsrate (B3): Zur Bestimmung der Mehrfachdetektionsrate $r_{\mathrm{d,M}}$ werden alle noch nicht für die Detektionsrate verrechneten Bewegungsendpunkte innerhalb der Detektionsintervalle berücksichtigt. Da nicht feststeht, was die Klassifikationsstufe aus einer mehrfach detektierten Bewegung macht, ist dieses Maß im Gegensatz zur Mehrfach*erkennungsrate* unter E3 durchaus relevant.

Mittlere Detektionsabweichungen (B4): Es lassen sich Mittelwerte für die Abweichungen des Beginns $\bar{\tau}_{\mathrm{d}}^{\mathrm{va}}$ und des Endes $\bar{\tau}_{\mathrm{d}}^{\mathrm{ve}}$ des Bewegungsintervalles vom Labelintervall angeben. Diese Werte werden über alle Bewegungsintervalle gebildet, die innerhalb der Detektionsintervalle liegen, und sind dementsprechend nur begrenzt aussagekräftig, da die Bewertung der gefundenen Bewegungsintervalle durch die anschließende Klassifikation noch aussteht.

10.6 Evaluierung des zweistufiges Systems

Betrachtet man ein System, das auf einer Bewegungsdetektion mit einer anschließenden isolierten Erkennung beruht, so sind von vornherein zwei Einschränkungen zu machen:

- Die Bewegungsdetektion wird auf jede Bewegung gleich ansprechen, egal ob es sich um eine bedeutungstragende oder nicht-bedeutungstragende Bewegung handelt. Es wird also in Kauf genommen, daß das System schon vom Ansatz her gewisse Situationen nicht bewältigen kann.

- Die Bewegungsdetektion soll Kerngesten detektieren. Diese Kerngesten sind allerdings von Vor- und Nachbereitungsbewegungen begleitet (vgl. Kap. 10.3), die

von einer Bewegungsdetektion ebenfalls als Teil der Bewegung angezeigt werden. Dies wird in der Hoffnung toleriert, daß durch eine optimale Parametereinstellung ein für die Klassifikation sinnvolles Bewegungsintervall gefunden wird. Außerdem wird darauf vertraut, daß die HMMs nicht zu empfindlich auf Segmentierungsfehler reagieren, solange entscheidende Teile der Kernbewegung im Bewegungsintervall enthalten sind.

Bei der Evaluierung des zweistufigen Systems soll zunächst die Bewegungsdetektion isoliert betrachtet und optimiert werden (s. Kap. 10.6.1). In Kap. 10.6.2 wird dann das gesamte System bewertet.

10.6.1 Getrennte Evaluierung der Bewegungsdetektion

10.6.1.1 Optimale Wahl der Detektionsparameter

Wie in Kap. 9.2.1.2 dargelegt, gibt es unabhängig von der Art, wie der Bewegungswert bestimmt wird (s. Kap. 9.2.1.1) fünf Parameter, mit denen sich die Bewegungsdetektion steuern läßt: die Bewegungsschwelle m_S, die minimale Anfangsbewegungsdauer $\tau^\mathrm{b}_\mathrm{min}$, die minimale Endruhedauer $\tau^\mathrm{e}_\mathrm{min}$, die Mindestlänge $\tau^\mathrm{l}_\mathrm{min}$ und der Mindestabstand $\tau^\mathrm{d}_\mathrm{min}$.

Die Evaluierungsergebnisse werden ebenfalls durch die maximale Detektionsverzögerung $\tau^\mathrm{v}_\mathrm{d,max}$ als dem zentralen Evaluierungsparameter beeinflußt. Dieser unerwünschte Effekt ist weitgehend vermeidbar, wenn $\tau^\mathrm{v}_\mathrm{d,max}$ groß genug gewählt wird (s. Kap. 10.5). Mit einem empirisch ermittelten Wert von $\tau^\mathrm{v}_\mathrm{d,max} = 50$ wird sichergestellt, daß potentielle Gesten nicht schon in der ersten Stufe verloren gehen: die Detektionsrate wird dadurch relativ groß ausfallen. Erst die anschließende Klassifikationsstufe wird die „richtigen“ von den „falschen“ Bewegungen trennen.

Um die Detektionsparameter optimal bestimmen zu können, wird eine Optimierungsfunktion $f_\mathrm{opt}(m_\mathrm{S}, \tau^\mathrm{b}_\mathrm{min}, \tau^\mathrm{e}_\mathrm{min}, \tau^\mathrm{l}_\mathrm{min}, \tau^\mathrm{d}_\mathrm{min})$ als Kombination aus Detektionsrate und den Fehlerwerten definiert:

$$f_\mathrm{opt} = g \cdot r_\mathrm{d} - r_\mathrm{d,M} - f_\mathrm{d}. \tag{10.14}$$

Diese Funktion muß *maximiert* werden. Über das Optimierungsgewicht g läßt sich das Verhalten der Optimierungsfunktion steuern: je größer g, desto stärker wird die Detektionsrate gewichtet und desto weniger beeinflussen die Falsch- und Mehrfachdetektionsrate den Optimierungsprozeß.

Als numerisches Optimierungsverfahren wurde das *Gauß-Seidel*-Verfahren verwendet: dabei werden die Optimierungsparameter zyklisch durchlaufen und in jeder Dimension getrennt optimiert [Ste87, Pre90]. Der Durchlauf wird so lange wiederholt, bis sich die Optima aller Komponenten nicht mehr ändern. Dieses Verfahren führt nur zu einem lokalen Optimum und setzt voraus, daß sinnvolle Anfangswerte vorgegeben werden.

Als Anfangswerte der *ganzzahligen* Parameter wurden Einstellungen gewählt, die sich empirisch als sinnvoll herausstellten: $\tau^\mathrm{b}_\mathrm{min} = 3$, $\tau^\mathrm{e}_\mathrm{min} = 3$, $\tau^\mathrm{d}_\mathrm{min} = 3$ und $\tau^\mathrm{l}_\mathrm{min} = 9$. Die Optimierung konnte dann mit einer äquidistanten Suche [Ste87, Pre90] mit der Schrittweite eins erfolgen, da dies durch den ganzzahligen Wertebereich keinen großen Rechenaufwand erforderte. Gleichzeitig durften die Mindestwerte $\tau^\mathrm{b}_\mathrm{min} = 2$, $\tau^\mathrm{e}_\mathrm{min} = 2$, $\tau^\mathrm{d}_\mathrm{min} = 0$ und $\tau^\mathrm{l}_\mathrm{min} = 9$ nicht unterschritten werden.

Die *kontinuierliche* Bewegungsschwelle wurde auf den Mittelwert des Bewegungswertes m_t an den Labelendpositionen gesetzt (s. Kap. 10.3). Ab dann erfolgte eine

BW-Typ	optimale Detektionsparameter					Detektionsergebnisse					
	m_{S}	$\tau^{\mathrm{l}}_{\mathrm{min}}$	$\tau^{\mathrm{d}}_{\mathrm{min}}$	$\tau^{\mathrm{b}}_{\mathrm{min}}$	$\tau^{\mathrm{e}}_{\mathrm{min}}$	r_{d}	f_{d}	$r_{\mathrm{d,M}}$	$\bar{\tau}^{\mathrm{va}}_{\mathrm{d}}$	$\bar{\tau}^{\mathrm{ve}}_{\mathrm{d}}$	$r^{\mathrm{eff}}_{\mathrm{d}}$
$m_{\mathrm{b},t}$	0,0038	9	4	3	4	83,23	3,63	27,78	-2,44	0,64	51,82
$m_{UV,t}$	1,10	9	3	5	3	83,85	3,84	28,35	-2,59	1,83	51,66
$m_{Y,t}$	0,75	9	3	3	2	87,18	4,10	30,11	-2,30	-1,11	52,96
$m_{YUV,t}$	1,72	9	3	5	2	84,89	4,00	28,76	-2,72	0,37	52,13
$m_{H_0,t}$	0,51	10	3	3	3	88,79	4,10	28,30	-0,98	6,03	56,39
$m_{H_2,t}$	0,53	9	3	3	3	88,63	4,31	30,63	-0,86	6,03	53,69
$m_{H_3,t}$	0,52	9	3	3	3	87,69	4,41	29,23	-1,05	6,84	54,05
$m_{H_4,t}$	0,52	9	3	3	3	87,18	4,41	28,92	-1,14	7,07	53,84
$m_{H_5,t}$	0,53	9	3	4	3	86,81	4,10	27,83	-1,16	7,49	54,88
$m_{H_6,t}$	0,53	9	3	4	3	86,55	4,15	27,26	-1,17	7,93	55,14
$m_{H_7,t}$	0,55	9	3	3	3	87,07	4,52	29,18	-1,11	7,12	53,37

Tabelle 10.4: Ergebnisse der Bewegungsdetektion für die Übungsdaten bei Verwendung unterschiedlicher Bewegungswerte (BW-Typ) bei der jeweils optimalen Parametereinstellung (r_{d}, $r_{\mathrm{d,M}}$, f_{d} und $r^{\mathrm{eff}}_{\mathrm{d}}$ in %, τ-Werte in Merkmalszeittakten, Detektionsschwelle $\tau^{\mathrm{v}}_{\mathrm{d,max}} = 50$, Optimierungsparameter $g = 3$, Raster 144×114)

logarithmisch-äquidistante Optimierung, bei der ein Optimierungsschritt in einer multiplikativen Veränderung des vorherigen Wertes um 5 % bestand. Für den Bewegungswert wurde kein Mindestwert vorgeschrieben.

10.6.1.2 Bestimmung des optimalen Bewegungswert-Verfahrens

Tabelle 10.4 zeigt die optimalen Parameter und die Ergebnisse der Bewegungsdetektion für die Übungsdaten unter Verwendung der verschiedenen Möglichkeiten, den Bewegungswert zu bestimmen (vgl. Kap. 9.2.1.1). Der gewählte Optimierungsparameter $g = 3$ (s. Gl. (10.14)), das Raster 144×114 und der Übungsdatensatz zeigen Ergebnistendenzen, die für andere Einstellungen und auch den Dialogdatensatz repräsentativ sind.

Da es für einen Vergleich der Ergebnisse schwierig wird, die drei Evaluierungsparameter Detektionsrate r_{d}, Mehrfachdetektionsrate $r_{\mathrm{d,M}}$ und Falschdetektionsrate f_{d} gleichzeitig im Auge zu behalten, wird zusätzlich eine um die Fehlerwerte reduzierte *effektive Detektionsrate*

$$r^{\mathrm{eff}}_{\mathrm{d}} = r_{\mathrm{d}} - r_{\mathrm{d,M}} - f_{\mathrm{d}} \tag{10.15}$$

angegeben. Es ist erkennbar, daß sich die pixelbasierten Bewegungswerte *alle* schlechter für die Bewegungsdetektion eignen als die merkmalsbasierten Bewegungswerte. Von den pixelbasierten Werten verhält sich $m_{Y,t}$ noch am günstigsten: offenbar ist die Luminanzinformation die wichtigste Pixelkomponente zur Detektion einer Bewegung. Die pixelbasierten Bewegungswerte wären durchaus für eine echtzeitfähige Bewegungsdetektion geeignet: die Berechnungszeiten betragen bei einem Raster von 144×114 Punkten bei voller Halbbildgröße je nach Verfahren nur etwa 2–6 ms.

Bei den merkmalsbasierten Bewegungswerten zeigt sich überraschenderweise, daß die ausschließliche Verwendung der Trajektorienkomponenten ($m_{H_0,t}$) das beste Resultat liefert. Offensichtlich stören die hinzukommenden Momentenkomponenten bei der Bestimmung der Bewegungsintervalle mehr als sie nützen. Dabei läßt sich allerdings keine klare Tendenz der Ergebnisse mit zunehmender Dimension der Merkmalsvektoren feststellen.

g	optimale Detektionsparameter					Detektionsergebnisse					
	m_S	τ^l_{min}	τ^d_{min}	τ^b_{min}	τ^e_{min}	r_d	f_d	$r_{d,M}$	$\bar{\tau}^{va}_d$	$\bar{\tau}^{ve}_d$	r^{eff}_d
1	0,43	14	5	3	3	76,64	3,53	9,29	-2,94	12,70	63,81
2	0,51	10	3	3	3	88,79	4,10	28,30	-0,98	6,03	56,39
4	0,51	9	5	3	2	87,69	3,79	19,47	-1,74	1,37	64,43
5	0,59	9	4	3	3	89,10	4,05	31,46	-1,01	1,57	53,58

Tabelle 10.5: Ergebnisse der Bewegungsdetektion für die Übungsdaten bei Variation des Gewichtungsfaktors der Optimierungsfunktion bei der jeweils optimalen Parametereinstellung (r_d, $r_{d,M}$, f_d und r^{eff}_d in %, τ-Werte in Merkmalszeittakten, $\tau^v_{d,max} = 50$, optimaler Bewegungswert $m_{H_0,t}$, Ergebnisse sind identisch für $g = 2$ und 3 sowie $g = 5$, 6 und 7)

Die Berechnung des Merkmalsvektors muß für die Erkennung in jedem Fall durchgeführt werden. Der Zusatzaufwand für die Berechnung des merkmalsbasierten Bewegungswertes ist daher vernachlässigbar klein.

Für die weiteren Betrachtungen wird nur noch der merkmalsbasierte Bewegungswert $m_{H_0,t}$, der ausschließlich Trajektorienkomponenten enthält, verwendet. Für die nachfolgende Erkennungsstufe (s. Kap. 10.6.2) werden dagegen nach wie vor Merkmalsvektoren verwendet, die grauwertgewichtete HMIs bis zur 2. Ordnung enthalten, was sich bei der isolierten Erkennung als optimal herausgestellt hat (s. Kap. 8).

10.6.1.3 Evaluierung der Bewegungsdetektion bei Variation der Optimierungsfunktion

Durch Variation des Optimierungsparameters g soll nun die beste Einstellung der Detektionsparameter ermittelt werden. Bei der getrennten Beurteilung der Bewegungsdetektion gibt es die Schwierigkeit, daß man nicht vorhersagen kann, wie die Klassifikationsentscheidung für die gefundenen Bewegungsintervalle ausfällt. Es ist lediglich sicher, daß die Gesamterkennungsrate r die Detektionsrate r_d nicht übersteigen kann. Daher sollte die Detektionsrate über ein großes Detektionsintervall möglichst hoch eingestellt werden. Falsche Klassifikationen erniedrigen die Erkennungsraten und erhöhen gleichzeitig die Falschakzeptanzrate f. Wie sich die Falschdetektionsrate auf die Erkennungwerte auswirkt, hängt von der Größe des Erkennungsintervalles im Vergleich zum Detektionsintervall ab. Bei einer sinnvollen Einstellung sollte eine Falschdetektion zur Falschakzeptanz beitragen — allerdings wird f nicht in Prozent sondern in der Einheit 1/h angegeben. Die Mehrfachdetektionsrate $r_{d,M}$ schließlich kann zur Erkennungs-, Mehrfacherkennungs- oder zur Falschakzeptanzrate beitragen.

Die Tabellen 10.5 und 10.6 zeigen die optimalen Parametereinstellungen und die Detektionsergebnisse für die Übungs- bzw. die Dialogdaten, wobei g von 1–7 variiert wurde. Es ist zu erkennen, daß sich die optimalen Detektionsparameter mit steigendem g erst viel und dann immer weniger ändern. Teilweise konvergieren sie auch für aufeinanderfolgende g auf gleiche Werte. Der Wert der effektiven Detektionsraten r^{eff}_d legt nahe, daß bei den Übungsdaten die optimale Parametereinstellung für $g = 4$ und bei den Dialogdaten für $g = 1$ erreicht wird. Allerdings fallen die Detektionsraten bei den Dialogdaten wesentlich besser aus als bei den Übungsdaten, was den Erwartungen widerspricht (s. Kap. 10.3). Dies legt nahe, daß die Detektionsraten nur begrenzt interpretationsfähig sind. Um endgültige Aussagen über die Leistungsfähigkeit des Systems zu erhalten, wer-

g	optimale Detektionsparameter					Detektionsergebnisse					
	m_S	τ^l_{min}	τ^d_{min}	τ^b_{min}	τ^e_{min}	r_d	f_d	$r_{d,M}$	$\bar{\tau}^{va}_d$	$\bar{\tau}^{ve}_d$	r^{eff}_d
1	0,48	14	5	3	15	92,64	7,48	1,29	-12,62	7,40	83,88
2	0,29	16	1	4	4	94,51	9,00	4,44	-13,13	12,96	81,07
4	0,35	11	5	4	4	96,14	8,64	7,94	-11,40	6,71	79,56
5	0,34	12	4	3	3	96,61	9,35	9,58	-11,23	6,16	77,69
6	0,36	10	5	4	4	96,61	9,00	8,64	-11,17	6,35	78,97

Tabelle 10.6: Ergebnisse der Bewegungsdetektion für die Dialogdaten bei Variation des Gewichtungsfaktors der Optimierungsfunktion bei der jeweils optimalen Parametereinstellung (r_d, $r_{d,M}$, f_d und r^{eff}_d in %, τ-Werte in Merkmalszeittakten, $\tau^v_{d,max} = 50$, optimaler Bewegungswert $m_{H_0,t}$, Ergebnisse sind identisch für $g = 2$ und 3 sowie $g = 6$ und 7)

den daher für alle auftretenden Parameterkombinationen die Gesamterkennungsraten bestimmt.

10.6.2 Gemeinsame Evaluierung von Bewegungsdetektion und anschließender isolierter Erkennung

Tabelle 10.7 zeigt, daß die Erkennungsraten des zweistufigen Systems für die Übungsdaten für die Optimierungsparameter $g = 2$ und $g = 5$ am besten sind (es wurde immer mit der optimalen Zustandszahl $N_{opt} = 5$ gearbeitet). Das beste Detektionsergebnis in Tabelle 10.5 für $g = 4$ ist allerdings auch in der Erkennungsleistung am ausgeglichensten, wenn die Mehrfachdetektionsrate in Betracht gezogen wird. Die Erkennungsraten steigen fast alle mit steigender Codebuchgröße bis zu $L = 1024$ an, um dann wieder leicht abzufallen. Mit der Erkennungrate steigt allerdings auch die Mehrfacherkennungsrate stark an, während die Falschakzeptanzrate deutlich abfällt. Für die optimale Codebuchgröße von $L = 1024$ liegen die Erkennungsraten nur wenig unter den Detektionsraten. Die Online-Erkennungsverzögerung $\bar{\tau}^{vo}$ ist unkritisch und bleibt immer unter 30.

Die Ergebnisse der Tabelle 10.7 wurden mit einem relativ großen Überhang von $o = 20$ erzielt. Dies wurde getan, damit auch bei größeren Ungenauigkeiten der Bewegungsdetektion möglichst die Kerngeste im Bewegungsintervall enthalten ist. Tabelle 10.8 zeigt exemplarisch für die optimale Parametereinstellung, daß eine Reduzierung des Überhanges zu einer Verringerung der Erkennungsleistung führt: die Erkennungsraten nehmen drastisch ab und die Falschakzeptanzraten zu; lediglich die Mehrfacherkennung reduziert sich. Damit wurde bestätigt, daß beim zweistufigen Ansatz ein großer Überhang zum Ausgleich von Detektionstoleranzen sinnvoll ist: anscheinend „verkraften“ die HMMs zu große Bewegungsintervalle wesentlich besser als unter Umständen fehlende Teile der Kerngeste. Dies erklärt auch die Diskrepanz zu den Ergebnissen bei der isolierten Erkennung mit Überhang (s. Tabelle 10.1 Seite 119), bei der sich ein mittelgroßer Überhang als optimal herausstellte: dort war garantiert, daß die Kerngeste auf jeden Fall nicht beschnitten wurde.

Die Erkennungsraten für die Dialogdaten sind im Vergleich zu denen für die Übungsdaten nur wenig geringer (s. Tabelle 10.9). Allerdings liegen die Falschakzeptanzraten deutlich höher. Das beste Erkennungsergebnis wird für eine Codebuchgröße von $L = 512$ und $g = 6$ bei einer optimalen Zustandszahl von $N_{opt} = 15$ erreicht. Die Erkennungsverzögerung ist jetzt etwas größer geworden; sie liegt aber immer noch im akzeptablen Bereich. Es zeigt sich, daß die guten Detektionsleistungen aus Tabelle 10.6 durch die

		$g=1$					$g=2$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	58,57	16,58	3,74	7,48	27,48	67,29	28,92	6,54	2,82	22,82
	64	63,81	12,49	5,19	6,71	26,71	78,04	19,15	11,79	2,30	22,30
	128	69,68	8,40	6,02	8,31	28,31	83,28	13,63	15,58	2,28	22,28
	256	72,12	6,09	7,37	8,86	28,86	85,31	9,13	20,92	1,79	21,79
	512	73,62	5,10	7,48	9,25	29,25	86,19	8,02	21,86	2,27	22,27
	1024	74,14	4,76	7,53	9,94	29,94	86,76	6,60	23,62	1,67	21,67
	2048	73,83	4,95	7,53	9,58	29,58	87,07	6,31	23,78	2,23	22,23
		$g=4$					$g=5$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	66,25	24,48	4,62	-0,88	19,12	67,45	31,07	6,28	-0,58	19,42
	64	75,13	16,84	8,26	-1,99	18,01	78,87	20,39	12,36	-1,72	18,28
	128	80,63	11,86	10,90	-2,49	17,51	84,16	13,86	17,76	-2,23	17,77
	256	83,33	8,15	14,28	-2,87	17,13	85,62	9,16	23,99	-3,33	16,67
	512	84,68	6,91	14,95	-2,43	17,57	86,71	8,09	24,66	-2,91	17,09
	1024	85,05	6,28	15,63	-2,68	17,22	87,12	6,88	26,22	-3,35	16,65
	2048	84,63	6,34	15,94	-2,71	17,29	86,86	6,72	26,74	-3,26	16,74

Tabelle 10.7: Erkennungsergebnisse des zweistufigen Systems für die Übungsdaten bei Variation des sich aus g ergebenden optimalen Parametersatzes und der Codebuchgröße L (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, $N_{\mathrm{opt}}=5$, $\tau^{\mathrm{v}}_{\mathrm{max}}=50$, $o=20$, Bewegungsdetektionsparameter s. Tabelle 10.5)

		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
o	20	85,05	6,28	15,63	-2,68	17,32
	15	82,50	9,86	12,31	-2,63	12,37
	10	74,97	17,60	7,17	-0,90	9,10
	5	64,85	26,13	3,32	0,00	5,00
	0	50,16	36,37	1,25	-0,93	1,07

Tabelle 10.8: Erkennungsergebnisse für die Variation des Überhangs o bei $L=1024$ und $g=4$ (sonst wie Tabelle 10.7)

anschließende Klassifikation nicht bestätigt werden konnten: offenbar waren viele der gefundenen Bewegungsintervalle fehlerhaft.

Es wurde auch bei den Dialogdaten wieder mit einem Überhang von $o=20$ gearbeitet. Reduziert man den Überhang (s. Tabelle 10.10 für die optimalen Parameter $L=512$ und $g=6$) so fällt die Erkennungsrate bis zum Überhang $o=10$ zunächst nur sehr leicht ab. Für kleinere Überhänge ergeben sich dann allerdings sehr starke Einbußen. Der zunächst schwache Abfall steht im Unterschied zum Verhalten bei den Übungsdaten. Obwohl die Dialogdaten kontextbehafteter sind als die Übungsdaten (s. Kap. 10.3), liefert auch hier ein großer Überhang von $o=20$ die besten Ergebnisse (vgl. Aussagen zu Übungsdaten oben).

		$g=1$					$g=2$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	70,33	36,79	0,00	4,07	24,07	71,03	43,43	0,23	11,04	31,04
	64	174,42	31,68	0,23	4,28	24,28	75,58	37,76	0,47	11,10	31,10
	128	78,97	26,42	0,12	4,96	24,96	79,91	32,78	0,35	11,78	31,78
	256	83,41	21,30	0,00	5,33	25,33	83,88	27,80	0,58	11,51	31,51
	512	82,59	22,27	0,00	4,97	24,97	83,88	27,53	0,82	11,38	31,38
	1024	82,24	22,69	0,00	5,20	25,20	83,41	28,08	0,82	11,39	31,39
	2048	81,54	23,10	0,35	4,97	24,97	83,06	28,50	0,82	11,41	31,41

		$g=4$					$g=5$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	72,78	46,62	0,58	5,48	25,48	73,72	48,83	0,58	4,76	24,76
	64	78,15	40,25	0,58	5,89	25,89	78,86	42,47	0,82	5,43	25,43
	128	81,43	35,96	0,93	5,87	25,87	81,89	38,59	1,05	5,31	25,31
	256	86,45	30,16	0,82	6,12	26,12	87,27	32,51	0,82	5,83	25,83
	512	86,45	29,74	1,17	5,87	25,87	87,27	31,95	1,29	5,42	25,42
	1024	85,86	30,29	1,29	6,23	26,23	86,22	33,20	1,29	5,49	25,49
	2048	84,35	31,95	1,40	5,91	25,91	85,40	33,89	1,52	5,23	25,23

		$g=6$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	73,13	48,00	0,58	5,16	25,16
	64	78,50	41,50	0,70	5,34	25,34
	128	82,13	36,93	0,93	5,43	25,43
	256	87,15	31,12	0,82	5,79	25,79
	512	86,92	30,85	1,29	5,57	25,57
	1024	86,33	31,40	1,40	5,96	25,96
	2048	84,81	33,34	1,29	5,60	25,60

Tabelle 10.9: Erkennungsergebnisse des zweistufigen Systems für die Dialogdaten bei Variation des sich aus g ergebenden optimalen Parametersatzes und der Codebuchgröße L (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, $N_{\mathrm{opt}} = 15$, $\tau^{\mathrm{v}}_{\mathrm{max}} = 50$, $o = 20$, Bewegungsdetektionsparameter s. Tabelle 10.6)

		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
o	20	86,92	30,85	1,29	5,57	25,57
	15	84,58	33,75	1,17	5,62	20,62
	10	85,86	32,64	0,82	5,64	15,64
	5	76,05	44,40	0,70	6,14	11,14
	0	70,21	51,60	0,47	6,11	10,11

Tabelle 10.10: Erkennungsergebnisse für die Variation des Überhangs o bei $L = 512$ und $g = 6$ (sonst wie Tabelle 10.9)

10.7 Evaluierung des einstufigen Spotting-Systems

Um aus der Vielfalt der möglichen Normierungs- (s. Kap. 9.3.1) und Triggerverfahren (s. Kap. 9.3.2) das Optimum zu finden, werden zunächst kontinuierliche Testdaten verwendet, die aus den Analysedaten synthetisiert werden (s. Anh. C.1.3). Die Tests mit

diesen *Synthesedaten* werden in Kap. 10.7.1 durchgeführt. Daraus geht dann die optimale Kombination aus Normierung und Triggerung hervor. Darüberhinaus kann die Fähigkeit zur Unterdrückung bedeutungsloser Bewegungen nachgewiesen werden. Das optimale Verfahren wird schließlich zur Evaluierung mit den real-kontinuierlichen Übungs- und Dialogdaten verwendet (s. Kap. 10.7.2). Dabei werden noch die weiteren Modifikationen an den Grundalgorithmen getestet.

10.7.1 Optimierung der Erkennung mit synthetisiert-kontinuierlichen Testdaten

Die Evaluierung mit den Synthesedaten hat mehrere Vorteile: es ist sichergestellt, daß alle Gesten in gleicher und kontrollierbarer Anzahl vorkommen und nicht durch Zufallsbewegungen gestört werden. Außerdem werden die Kerngesten garantiert nicht durch den Gestenkontext verändert. Zusätzlich läßt sich der Abstand der Kerngesten genau einstellen, was aussagekräftigere Untersuchungen ermöglicht. Für alle folgenden Tests wurden vier verschiedenen Füllsequenzlängen $L_\mathrm{f} = 35$, 70, 105 und 140 verwendet (s. Anh. C.1.3). Die Teilergebnisse bei verschiedenen Füllsequenzlängen geben Aufschluß über das Verhalten der verschiedenen Normierungen und Triggerverfahren. Das eigentliche Ergebnis ist der Mittelwert über alle Füllsequenzlängen.

Alle HMMs hatten eine Codebuchgröße von $L = 256$ und $N = 25$ Zustände; diese Werte stellten sich für den zugrundliegenden Datensatz als optimal heraus. Die Modelle wurden ohne Nachschätzung des Codebuchs trainiert, um die Unterschiede in den Erkennungsergebnissen bei den verschiedenen Verfahren deutlicher hervortreten zu lassen [Mor98b, Mor98c].

Die optimalen Spotting-Parameter wurden empirisch durch gezielte Variationen gefunden, wobei für alle Berechnungen ein Erkennungsintervall von $\tau^\mathrm{v}_\mathrm{max} = 35$ voreingestellt war. Eine automatische Optimierung wie bei den Detektions-Parametern, die sich getrennt von der Erkennung evaluieren ließen, war aufgrund des großen Rechenzeitbedarfs nicht durchführbar. Für das Normierungsverfahren N1 erwiesen sich die Parameter $\tau^\mathrm{b}_\mathrm{s} = 20$, $\tau^\mathrm{e}_\mathrm{s} = 10$, $\tau^\mathrm{b}_\mathrm{p} = 30$, $\tau^\mathrm{e}_\mathrm{p} = 3$ und $\tau_\mathrm{dist} = 70$ als optimal [Mor98b]. Die Normierungsverfahren N2 lieferten für alle Triggermöglichkeiten T1–T3 bei $\tau^\mathrm{b}_\mathrm{s} = 30$, $\tau^\mathrm{e}_\mathrm{s} = 1$, $\tau^\mathrm{b}_\mathrm{p} = 30$, $\tau^\mathrm{e}_\mathrm{p} = 1$ und $\tau_\mathrm{dist} = 10$ die besten Ergebnisse [Mor98c].

Die Erkennungsergebnisse werden im folgenden jeweils ohne Schwelle S_rel und mit optimaler Schwelle angegeben. Die optimale Schwelle wurde so gelegt, daß die Falschakzeptanzrate stark vermindert und gleichzeitig die Erkennung gar nicht oder nur unwesentlich angetastet wurde. Prinzipiell ist es möglich, daß die Erkennungsrate bei Verminderung der Schwelle leicht ansteigt: dieses zunächst unerwartete Verhalten wird durch die Bedingung des Mindestabstands τ_dist der Peak-Detektion hervorgerufen.

10.7.1.1 Optimierung des Normierungs- und Triggerverfahrens

Tabelle 10.11 zeigt die Ergebnisse für das Normierungsverfahren N1 unter Variation der Normierungslänge L_n (s. Gl. (9.11) in Kap. 9.3.1). Bei der besten Normierungslänge $L_\mathrm{n} = 15$ ergibt sich nur eine mäßige Erkennungsrate von unter 80 %. Die vorsichtige Anwendung einer Rückweisungsschwelle von $S_\mathrm{rel} = 0,03$ kann die Falschdetektionsrate nicht unter $35\,\mathrm{h}^{-1}$ senken. Allerdings fällt im Vergleich zum zweistufigen Ansatz auf, daß keine Mehrfachdetektionen vorliegen. Wie sich in Kap. 10.7.2 zeigen wird, ist dies aber ein Effekt des Datensatzes und nicht des Verfahrens. Aufgeschlüsselt nach Füllsequenzlängen

L_n	$S_{rel} = \infty$					$S_{rel} = 0,03$				
	r	f	r_M	$\bar{\tau}^v$	$\bar{\tau}^{vo}$	r	f	r_M	$\bar{\tau}^v$	$\bar{\tau}^{vo}$
10	71,25	55,40	0,00	3,85	16,85	71,04	40,58	0,00	3,86	16,86
15	**78,75**	**48,91**	**0,00**	**4,46**	**14,46**	**77,29**	**35,85**	**0,00**	**4,59**	**14,59**
20	76,67	49,58	0,00	5,59	18,59	75,42	36,90	0,00	5,73	18,73
25	70,42	53,81	0,00	6,84	19,84	70,63	37,53	0,00	7,17	20,17
30	67,50	56,85	0,00	7,21	20,21	67,50	39,54	0,00	7,44	20,44

Tabelle 10.11: Erkennungsergebnisse des Spotting-Verfahrens mit Normierung N1 bei Variation der Normierungslänge L_n (r und r_M in %, f in 1/h, τ-Werte in Merkmalszeittakten, Mittelwerte über alle Füllsequenzlängen L_f, optimale Einstellung fett hervorgehoben)

L_f	$S_{rel} = \infty$					$S_{rel} = 0,03$				
	r	f	r_M	$\bar{\tau}^v$	$\bar{\tau}^{vo}$	r	f	r_M	$\bar{\tau}^v$	$\bar{\tau}^{vo}$
35	95,83	5,95	0,00	3,78	16,78	81,67	5,95	0,00	3,97	16,97
70	74,17	49,11	0,00	4,39	17,39	81,67	18,75	0,00	4,56	17,56
105	74,17	68,57	0,00	4,49	17,49	73,33	58,57	0,00	4,78	17,78
140	70,83	72,02	0,00	5,15	18,15	72,50	60,12	0,00	5,05	18,05
∅	78,75	48,91	0,00	4,46	17,46	77,29	35,85	0,00	4,59	17,59

Tabelle 10.12: Aufschlüsselung der Ergebnisse aus Tabelle 10.11 nach Füllsequenzlänge L_f für Normierungslänge $L_n = 15$ (Mittelwert ∅)

L_f in Tabelle 10.12 zeigt sich, daß eine Vergrößerung von L_f zu einem dramatischen Anstieg der Falschakzeptanzrate f führt.

Die Ergebnisse in den folgenden Tabellen 10.13-10.19 lassen erkennen, daß eine mitgeführte lokale Normierungslänge (Normierungsverfahren N2, s. Gl. (9.12) in Kap. 9.3.1) in jedem Fall bessere Ergebnisse liefert. Die Güte des Verfahrens wird jedoch ganz entscheidend durch die Wahl des Triggerverfahrens beeinflußt (s. Kap. 9.3.2).

Mit der passiven Triggerung T1 läßt sich die Erkennungsleistung im Vergleich zum Normierungsverfahren N1 um einiges verbessern (s. Tabelle 10.13). Insbesondere wirken sich längere Füllsequenzlängen im Vergleich zur Normierung N1 deutlich weniger negativ auf die Falschakzeptanzraten aus. Offenbar bewirkt die Möglichkeit eines Quereinstieges neuer Pfade, daß sich Score-Verläufe sowohl beim Anstieg als auch beim Abstieg besser an die Bewegungen anpassen können.

L_f	$S_{rel} = \infty$					$S_{rel} = 0,05$				
	r	f	r_M	$\bar{\tau}^v$	$\bar{\tau}^{vo}$	r	f	r_M	$\bar{\tau}^v$	$\bar{\tau}^{vo}$
35	90,83	29,76	0,00	10,73	12,73	87,50	13,10	0,00	10,71	12,71
70	86,67	39,29	0,00	12,51	14,51	85,00	18,75	0,00	12,64	14,64
105	86,67	40,71	0,00	14,08	16,08	85,00	20,00	0,00	14,24	16,24
140	85,83	42,26	0,00	15,41	17,41	84,17	19,05	0,00	15,58	17,58
∅	87,50	38,01	0,00	13,18	15,18	85,42	17,72	0,00	13,29	15,29

Tabelle 10.13: Erkennungsergebnisse des Spotting-Verfahrens mit Normierung N2 und Triggerung T1 (r und r_M in %, f in 1/h, τ-Werte in Merkmalszeittakten, Aufschlüsselung nach Füllsequenzlängen L_f und Mittelwert ∅)

W	$S_{\text{rel}} = \infty$					$S_{\text{rel}} = 0,08$				
	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
0	83,75	40,63	0,21	13,07	15,07	83,75	22,78	0,00	13,07	15,07
45	90,42	35,15	0,00	10,50	12,50	90,21	14,09	0,00	10,52	12,52
90	**93,13**	**32,47**	**0,21**	**9,83**	**11,83**	**91,25**	**8,41**	**0,00**	**9,86**	**11,86**
135	95,00	38,63	0,21	9,56	11,56	85,42	6,90	0,00	9,31	11,31

Tabelle 10.14: Erkennungsergebnisse des Spotting-Verfahrens mit Normierung N2 und Triggerung T2 bei Variation des Score-Eingangsgewichtes W (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Trigger-Normierungslänge $L_{\text{s}} = 3$, Mittelwerte über alle Füllsequenzlängen L_{f}, optimale Einstellung fett hervorgehoben)

L_{f}	$S_{\text{rel}} = \infty$					$S_{\text{rel}} = 0,08$				
	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
35	97,50	30,95	0,00	8,64	10,64	92,50	9,52	0,00	8,61	10,61
70	94,17	32,14	0,00	10,33	12,33	92,50	2,68	0,00	10,40	12,40
105	93,33	29,29	0,00	10,28	12,28	93,33	7,14	0,00	10,28	12,28
140	87,50	37,50	0,83	10,07	12,07	86,67	14,29	0,00	10,14	12,14
∅	93,13	32,47	0,21	9,83	11,83	91,25	8,41	0,00	9,86	11,86

Tabelle 10.15: Aufschlüsselung der Ergebnisse aus Tabelle 10.14 nach Füllsequenzlänge L_{f} für Score-Eingangsgewicht $W = 90$ (Mittelwert ∅)

Mit der aktiven T2-Triggerung läßt sich die Erkennungsleistung nochmals deutlich verbessern (s. Tabelle 10.14). Jetzt erzeugen die mittleren Füllsequenzlängen sogar die geringsten Falschakzeptanzraten (s. Tabelle 10.15). Es zeigt sich, daß dem Score-Eingangsgewicht W eine entscheidende Bedeutung zukommt: offenbar können damit mehr korrekten als falschen Gesten zum „Durchbruch“ verholfen werden. Aus der Tatsache, daß das Eingangsgewicht am Anfang eines Viterbi-Pfades addiert wird, läßt sich schließen, daß der korrekte Viterbi-Pfad insbesondere zu Beginn eine Bewegung durch falsche Pfade „gefährdet“ ist und einer Unterstützung bedarf.

Die Ergebnisse in den Tabellen 10.16 und 10.17 belegen, daß die permanente Triggerung T3 von allen Verfahren am besten abschneidet. Damit zeigt sich, daß es besser ist, zu oft zu triggern und darauf zu vertrauen, daß falsche Pfade sich wieder im Verlauf der Viterbi-Rekursion verlieren, als eventuell einen wichtigen Triggerzeitpunkt zu versäumen, wie es mit der aktiven Triggerung T2 passieren kann. Mit der optimalen Einstellung von $W = 150$ kann mit dem synthetischen Datensatz eine Erkennungsrate von 100 % bei vernachlässigbarer Falschakzeptanzrate erreicht werden.

Damit konnte nachgewiesen werden, daß die Normierungsmethode N2 mit mitgeführter lokaler Länge in Verbindung mit einer permanenten Triggerung T3 bei gleichzeitiger Verwendung eines Score-Eingangsgewichtes W die besten Ergebnisse liefert. Im weiteren Verlauf wird nur noch mit diesen Einstellungen gearbeitet.

10.7.1.2 Nachweis für die Unterdrückung bedeutungsloser Bewegungen

Die kontrollierbare Zusammensetzung der Synthesedaten bietet sich an, die Fähigkeit des Spotting-Verfahrens zu untersuchen, nicht-bedeutungstragende Bewegungen zu unterdrücken. Hierzu werden bei der Erkennung nur noch die ersten sechs Modelle für

W	$S_{\text{rel}} = \infty$					$S_{\text{rel}} = 0,08$				
	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
0	84,79	40,67	0,21	13,28	15,28	84,58	24,03	0,21	13,26	15,26
30	94,58	30,07	0,00	10,29	12,29	94,58	11,76	0,00	10,29	12,29
60	99,17	26,62	0,21	9,27	11,27	99,17	5,30	0,00	9,27	11,27
90	99,17	28,32	0,00	8,38	10,38	99,17	2,50	0,00	8,38	10,38
120	100,00	33,45	0,00	7,76	9,76	100,00	1,40	0,00	7,76	9,76
150	**100,00**	**42,71**	**0,00**	**7,20**	**9,20**	**100,00**	**0,70**	**0,00**	**7,20**	**9,20**
180	100,00	66,22	0,00	6,50	8,50	99,17	9,64	0,00	6,45	8,45
210	97,92	90,19	0,00	6,08	8,08	96,88	31,80	0,00	6,03	8,03

Tabelle 10.16: Erkennungsergebnisse des Spotting-Verfahrens mit Normierung N2 und Triggerung T3 bei Variation des Score-Eingangsgewichtes W (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Mittelwerte über alle Füllsequenzlängen L_{f}, optimale Einstellung fett hervorgehoben)

L_{f}	$S_{\text{rel}} = \infty$					$S_{\text{rel}} = 0,08$				
	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
35	100,00	46,43	0,00	6,85	8,85	100,00	0,00	0,00	6,85	8,85
70	100,00	50,00	0,00	7,27	9,27	100,00	0,89	0,00	7,27	9,27
105	100,00	39,29	0,00	7,33	9,33	100,00	0,71	0,00	7,33	9,33
140	100,00	35,12	0,00	7,37	9,37	100,00	1,19	0,00	7,37	9,37
∅	100,00	42,71	0,00	7,20	9,20	100,00	0,70	0,00	7,20	9,20

Tabelle 10.17: Aufschlüsselung der Ergebnisse aus Tabelle 10.16 nach Füllsequenzlänge L_{f} für Score-Eingangsgewicht $W = 150$ (Mittelwert ∅)

die Erkennung herangezogen, während die Zusammensetzung der Testdaten unverändert bleibt. Damit kann das System nur noch die Hälfte der vorkommenden Bewegungen aktiv erkennen. Die andere Hälfte stellen für das System unbekannte Bewegungen dar.

Die Tabellen 10.18 und 10.19 zeigen im Vergleich mit den Tabellen 10.16 und 10.17, daß die unbekannten Bewegungen ohne Rückweisungsschwelle naturgemäß zu sehr großen Falschakzeptanzraten f führen. Allerdings liegen die Score-Maxima der unbekannten Bewegungen offenbar deutlich unter den gültigen Score-Maxima, so daß sie sich durch den Einsatz einer Rückweisungsschwelle effektiv unterdrücken lassen. Die Erkennungsrate von 100 % und die nicht vorhandene Falschakzeptanz belegen, daß bedeutungslose Bewegungen wirkungsvoll unterdrückt werden können. Allerdings ist hierbei die richtige Justierung des Score-Eingangsgewichtes W noch entscheidender, da eine Verschiebung von W den Fehler stark ansteigen läßt. Das Optimum von $W = 150$ verschiebt sich im Vergleich zum vollen Katalog allerdings nicht, so daß eine Justierung unproblematisch ist.

10.7.2 Erkennung mit real-kontinuierlichen Daten

Nachdem sich durch die Evaluierung des Spottingsystems mit synthetisierten Testdaten das Normierungsverfahren N2 mit der Triggerung T3 als optimal herausgestellt hat, sollen die weiteren Tests mit den Übungs- und Dialogdaten durchgeführt werden, um Aussagen darüber zu erhalten, wie das System mit realen Daten funktioniert. Es hat sich gezeigt, daß das Score-Eingangsgewicht abhängig von der Ausprägung der Daten genau justiert

W	$S_{\text{rel}} = \infty$					$S_{\text{rel}} = 0,08$				
	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
0	82,92	320,42	0,42	13,92	15,92	82,92	54,94	0,42	13,92	15,92
30	91,25	281,43	0,00	12,95	14,95	91,25	35,24	0,00	12,95	14,95
60	98,33	280,42	0,00	11,75	13,75	98,33	16,79	0,00	11,75	13,75
90	98,33	288,39	0,00	10,83	12,83	98,33	8,81	0,00	10,83	12,83
120	100,00	288,27	0,00	10,27	12,27	100,00	5,00	0,00	10,27	12,27
150	**100,00**	**331,07**	**0,00**	**9,74**	**11,74**	**100,00**	**0,00**	**0,00**	**9,74**	**11,74**
180	100,00	402,14	0,00	9,23	11,23	98,33	20,12	0,00	9,18	11,18
210	100,00	446,67	0,00	8,81	10,81	97,92	80,00	0,00	8,78	10,78

Tabelle 10.18: Erkennungsergebnisse des Spotting-Verfahrens mit Normierung N2 und Triggerung T3 bei Variation des Score-Eingangsgewichtes W mit erster Hälfte des Erkennungs-Kataloges (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Mittelwerte über alle Füllsequenzlängen L_{f}, optimale Einstellung fett hervorgehoben)

L_{f}	$S_{\text{rel}} = \infty$					$S_{\text{rel}} = 0,08$				
	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
35	100,00	390,48	0,00	9,53	11,53	100,00	0,00	0,00	9,53	11,53
70	100,00	371,43	0,00	9,75	11,75	100,00	0,00	0,00	9,75	11,75
105	100,00	302,86	0,00	9,82	11,82	100,00	0,00	0,00	9,82	11,82
140	100,00	259,52	0,00	9,87	11,87	100,00	0,00	0,00	9,87	11,87
∅	100,00	331,07	0,00	9,74	11,74	100,00	0,00	0,00	9,74	11,74

Tabelle 10.19: Aufschlüsselung der Ergebnisse aus Tabelle 10.18 nach Füllsequenzlänge L_{f} für Score-Eingangsgewicht $W = 150$ (Mittelwert ∅)

werden muß (s. Kap. 10.7.1.2): die Einstellung von W muß daher mit den realen Daten nochmals vorgenommen werden.

Die Ergebnisse werden zunächst jeweils in Tabellen ohne aktivierte Rückweisungsschwelle S_{rel} dargestellt. Dadurch läßt sich das Verhalten der Erkennungsleistung bei Ändern eines Parameters schnell überblicken. Mit der Rückweisungsschwelle S_{rel} läßt sich dann die Falschakzeptanzrate f prinzipiell beliebig auf Kosten der Erkennungsrate r reduzieren. Die sich einstellenden Kombinationen von r und f werden dann in einem sog. ROC-Darstellung (*receiver operating characteristic*) mit der Rückweisungsschwelle S_{rel} als Parameter aufgetragen (s. beispielsweise [Kro86]). Anhand einer ROC-Darstellung läßt sich demnach das gewünschtes r-f-Verhältnis einstellen. Da in dieser Arbeit nur verschiedene Systemcharakteristiken miteinander verglichen werden sollen, werden in die ROC-Darstellungen die Werte von S_{rel} *nicht* eingetragen, um die Diagramme nicht zu unübersichtlich werden zu lassen.

Empirisch wurde ermittelt, daß sowohl bei den Übungs- als auch bei den Dialogdaten mit $\tau_{\text{s}}^{\text{b}} = 30$, $\tau_{\text{s}}^{\text{e}} = 0$, $\tau_{\text{p}}^{\text{b}} = 10$, $\tau_{\text{p}}^{\text{e}} = 1$ die besten Ergebnisse erzielt werden können (vgl. Kap. 10.7.1). Alle folgenden Ergebnisse beruhen auf diesen Einstellungen. Der minimale Peak-Abstand wurde, wenn nicht anders angegeben, auf $\tau_{\text{dist}} = 0$ gesetzt. Die Variation dieses Parameters wird in Anh. D.3 untersucht. Dort wird sich zeigen, daß durch Vergrößerung von τ_{dist} ein ähnlicher Effekt wie mit der Verringerung der Rückweisungsschwelle erzielt werden kann.

Wie schon bei den synthetischen Daten ergibt sich beim Spotting stets eine *positive* mittlere Erkennungsverzögerung $\bar{\tau}^{\mathrm{v}}$, was auch für jede einzelne Erkennungsverzögerung bestätigt werden konnte. Damit nicht die Gefahr besteht, daß eine Geste dem vorherigen Label zugerechnet wird, wurde das Erkennungsintervall für die folgenden Auswertungen $[0, \tau^{\mathrm{v}}_{\mathrm{max}}]$ daher *einseitig* ausgelegt und die maximale Erkennugsverzögerung auf den Wert $\tau^{\mathrm{v}}_{\mathrm{max}} = 200$ gesetzt.

10.7.2.1 Optimale HMM-Parameter und Wahl des Überhangs

Es fällt auf, daß für eine optimale Leistungsfähigkeit beim Spotting wesentlich mehr HMM-Zustände N erforderlich sind als bei der isolierten Erkennung oder dem zweistufigen Ansatz (s. Tabelle 10.20). Sowohl für die Übungs- als auch die Dialogdaten ergibt sich die beste Erkennungsleistung für $N_{\mathrm{opt}} = 30$ (die Erkennungsleistung nimmt mit $N > 30$ Zuständen wieder allmählich ab, was in der Tabelle allerdings nicht dargestellt ist). Es wurde zunächst ein Score-Eingangsgewicht von $W = 0$ vorgegeben. Ausgehend von den Erfahrungen mit dem zweistufigen Ansatz wurde der Überhang zunächst auf einen großen Wert von $o = 20$ gesetzt. Die Erkennungs- und die Falschakzeptanzraten zeigen unabhängig von der Zustandszahl für Codebuchgrößen von $L = 1024$ in der Regel ein deutliches Optimum. Wie erwartet, sind die Übungsdaten auch vom Spotting-System einfacher zu handhaben als die Dialogdaten. Allerdings lassen sich im Vergleich zum zweistufigen Ansatz deutlich bessere Erkennungsraten erzielen: sie sind in etwa vergleichbar mit der isolierten Erkennung (s. Kap. 10.3).

Im Vergleich zum zweistufigen Ansatz fällt die größere mittlere Online-Erkennungsverzögerung $\bar{\tau}^{\mathrm{vo}}$ ins Auge. Verdeutlicht man sich allerdings, daß die Angaben in der Tabelle bei den Übungsdaten einer Verzögerung von ca. 0,5 Sekunden und bei den Dialogdaten von etwa 1 Sekunde entsprechen, so liegen diese Werte in der Größenordnung der Reaktionszeiten des Versuchsleiters bei den Wizard-of-Oz-Versuchen. Diese Reaktionszeit wurde durch einen stichprobenartigen Vergleich der Protokollfiles mit den Videoaufnahmen der Versuche festgestellt (vgl. Aufbau in Kap. 4.1). In Kap. 4.2.2 wurde die Reaktionszeit bei den Usability-Versuchen als angemessen beurteilt. Für das automatische System stellt eine Erkennungsverzögerung sogar indirekt eine Erleichterung dar, weil während der Experimente beobachtet wurde, daß mit steigender Reaktionszeit die Gestenfolgefrequenz tendenziell verringert wird.

Die ROC-Darstellungen für die optimalen Parameter $N = 30$ und $L = 1024$ in Bild 10.1 zeigen nocheinmal deutlich die unterschiedliche Erkennungsleistung für die beiden Datensätze: selbst bei Verringerung der Rückweisungsschwelle verlaufen beide Kennlinien immer deutlich getrennt.

Tabelle 10.21 zeigt, wie sich die Verringerung der Überhänge o^{b} und o^{e} auswirkt. Dabei wurden auch asymmetrische Überhänge mit $o^{\mathrm{e}} = 0$ verwendet, um eine möglichst geringe mittlere Erkennungsverzögerung zu erhalten. Mit Veränderung des Überhanges ändert sich auch in der Regel die Zustandszahl N_{opt}, mit der die maximale Erkennungsleistung erzielt werden kann. Es ist erkennbar, daß die Falschakzeptanzrate mit kleiner werdendem Überhang in jedem Fall stark ansteigt, während die Erkennungsrate meistens fällt. Nur bei den Dialogdaten ist die Erkennungsrate für mittelgroße Überhänge etwas größer als für $o = 20$. Bis auf einen konstanten Anteil ist die Erkennungsverzögerung $\bar{\tau}^{\mathrm{v}}$ näherungsweise proportional zu o^{e}. Um die Erkennungsverzögerung nicht mehr weiter ansteigen zu lassen, ist es nicht sinnvoll, einen Overhead über $o = 20$ zu verwenden.

Die ROC-Darstellungen in den Bildern 10.2 und 10.3 zeigen, daß sich über den gesamten möglichen r-f-Bereich die Rangordnung der Ergebnisse nicht ändert: die beste

N	L	Übungsdaten				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
10	128	96,05	24,83	0,62	31,31	32,31
	256	96,73	20,01	0,36	30,91	31,91
	512	96,78	20,04	0,78	30,58	31,58
	1024	98,39	14,84	0,47	30,24	31,24
	2048	93,09	16,74	0,31	29,81	30,81
20	128	96,42	24,19	0,57	33,78	34,78
	256	97,72	20,51	0,52	33,57	34,57
	512	97,14	17,31	0,57	33,56	34,56
	1024	98,65	15,54	0,52	33,09	34,09
	2048	94,03	16,65	0,16	32,74	33,74
30	128	94,86	23,11	0,42	36,19	37,19
	256	97,14	21,91	0,47	36,11	37,11
	512	97,51	17,41	0,62	36,02	37,02
	1024	**98,81**	**16,90**	**0,68**	**35,65**	**36,65**
	2048	93,35	17,91	0,21	35,18	36,18

N	L	Dialogdaten				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
10	128	84,88	64,32	5,51	48,71	49,71
	256	85,35	58,65	6,21	51,44	52,44
	512	88,28	57,96	4,69	49,10	50,10
	1024	88,28	60,03	5,74	49,38	50,38
	2048	88,28	53,95	8,32	47,95	48,95
20	128	87,34	50,21	5,86	51,34	52,34
	256	88,04	48,69	6,33	51,91	52,91
	512	89,68	42,19	5,51	52,09	53,09
	1024	90,62	39,98	6,33	51,83	52,83
	2048	89,21	42,47	6,45	50,52	51,52
30	128	86,75	45,23	7,15	52,36	53,36
	256	89,33	39,84	4,69	55,05	56,05
	512	89,33	39,01	4,45	53,72	54,72
	1024	**91,56**	**35,13**	**7,03**	**53,58**	**54,58**
	2048	89,33	37,35	4,57	52,28	53,28

Tabelle 10.20: Erkennungsergebnisse des Spotting-Verfahrens mit Übungs- und Dialogdaten bei Variation von Zustandszahl N und Codebuchgröße L (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Score-Eingangsgewicht $W = 0$, Überhang $o = 20$, optimale Einstellungen fett hervorgehoben)

Gesamtleistung wird mit $o = 20$ erzielt. Lediglich in r-Bereichen unter 65 %, die für den praktischen Einsatz nicht mehr interessant sind, gewinnt bei den Dialogdaten die Kennlinie für $o = 10$ etwas die Überhand. Insbesondere zeigen die ROCs, daß die unter Umständen für sinkenden Überhang leicht steigende Erkennungsrate r bei den Dialogdaten (s. Tabelle 10.21) an der Dominanz der Erkennungsleistung für $o = 20$ nichts ändert.

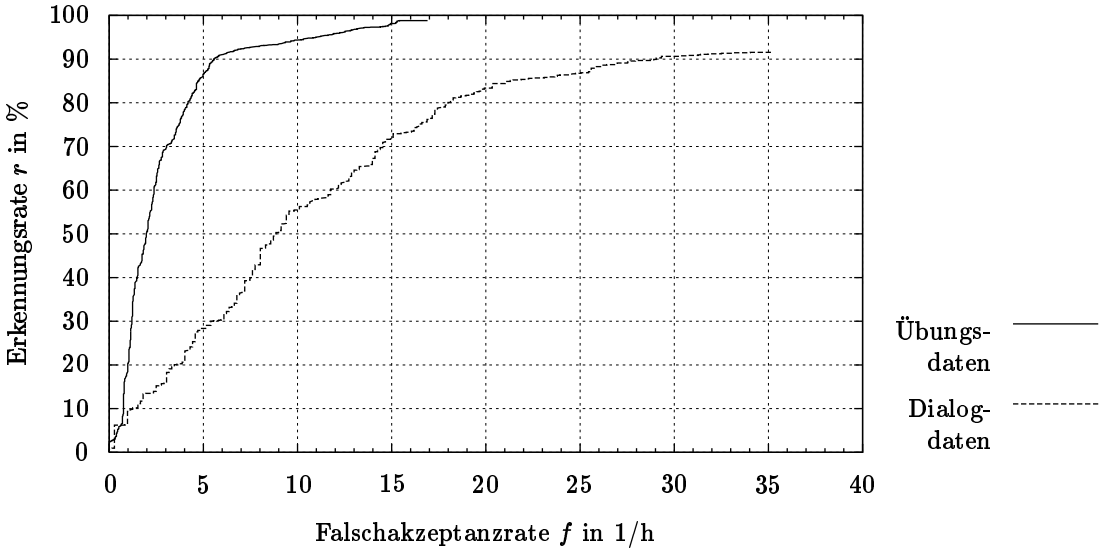


Bild 10.1: ROC-Darstellungen der optimalen Erkennungsergebnisse aus Tabelle 10.20 erzeugt durch Variation der Rückweisungsschwelle S_{rel} ($N = 30$, $L = 1024$, $W = 0$, $o = 20$)

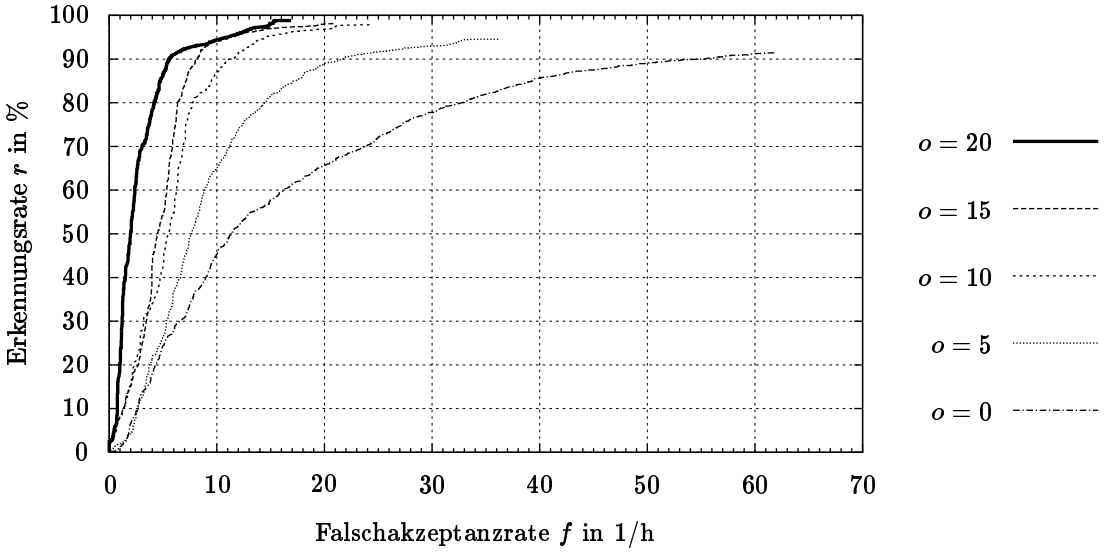


Bild 10.2: ROC-Darstellungen ausgewählter Erkennungsergebnisse aus Tabelle 10.21 für den Übungsdatensatz erzeugt durch Variation der Rückweisungsschwelle S_{rel} für verschiedene symmetrische Überhänge o ($L = 1024$, $W = 0$)

Dieselben Tendenzen gelten für die aus Gründen der Übersichtlichkeit nicht dargestellen asymmetrischen Überhänge.

Im folgenden werden jetzt auf Grundlage der gefundenen optimalen HMM-Parameter $N = 30$ und $L = 1024$ sowie des für die Gesamtleistung optimalen Überhanges $o = 20$ weitere Evaluierungen und Optimierungen besprochen.

o^{b}	o^{e}	Übungsdaten					
		N_{opt}	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
20	20	30	**98,81**	**16,90**	**0,68**	**35,65**	**36,65**
15	15	20	98,13	20,83	1,45	30,82	31,82
10	10	20	97,82	24,19	1,30	28,58	29,58
5	5	10	94,55	36,18	1,82	22,74	23,74
0	0	5	91,43	61,80	0,93	16,79	17,79
20	0	20	94,18	36,46	0,99	18,67	19,67
15	0	15	95,59	44,52	1,14	18,14	19,14
10	0	10	95,07	50,57	0,78	17,41	18,41
5	0	5	93,30	60,91	0,73	15,61	16,61
o^{b}	o^{e}	Dialogdaten					
		N_{opt}	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
20	20	30	**91,56**	**35,13**	**7,03**	**53,58**	**54,58**
15	15	25	92,38	45.23	6,45	45.63	46.63
10	10	20	92,50	57,13	5,16	40,51	41,51
5	5	10	87,10	95,72	7,03	35,18	36,18
0	0	10	71,40	123,39	11,37	30,81	31,81
20	0	20	90,27	107,34	4,57	26,50	27,50
15	0	20	92,97	103,47	5,63	27,84	28,84
10	0	20	92,61	84,79	5,39	28,88	29,88
5	0	10	86,87	116.33	7,39	31,79	32,79

Tabelle 10.21: Erkennungsergebnisse des Spotting-Verfahrens mit Übungs- und Dialogdaten bei Variation der Überhänge o^{b} und o^{e} und jeweils optimaler Zustandszahl N_{opt} (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Codebuchgröße $L = 1024$, Score-Eingangsgewicht $W = 0$, optimale Einstellungen fett hervorgehoben)

10.7.2.2 Wahl des optimalen Score-Eingangsgewichtes

Bei den Synthesedaten beeinflußte die Einstellung des Score-Eingangsgewichtes W die Leistung des Systems erheblich (s. Kap. 10.7.1). Variiert man diesen Parameter bei den realen Daten (s. Tabelle 10.22), so stellt man unterschiedliche Wirkungen fest. Bei den Übungsdaten ist außer einem leichten Anstieg der Falschakzeptanzrate mit steigendem W so gut wie keine Veränderung sichtbar. Bei den Dialogdaten ergibt sich jedoch eine deutliche Steigerung der Erkennungsleistung: ausgehend von $W = 0$ steigt die Erkennungrate r bis zum Optimum bei $W = 30$ um rund 3 % absolut an, während die Falschakzeptanzrate gleichzeitig sogar um ca. 6,5 1/h abnimmt. Außerdem sinkt die Erkennungsverzögerung merklich. Für Gewichte $W > 30$ bleibt die Erkennungsrate auf ähnlich hohem Niveau, während der Fehler wieder leicht zunimmt. Da sich für die Übungsdaten praktisch keine Veränderung ergibt, kann $W = 30$ als optimale Einstellung für alle Datenarten gelten.

In Bild 10.4 sind die entsprechenden ROCs für $W = 0$ und $W = 30$ dargestellt. Hier ist die deutliche Verbesserung der Erkennungsleistung für die Dialogdaten gut sichtbar. Außerdem erkennt man, daß auch die Erkennungscharakteristik für die Übungsdaten verbessert wird: dieser Effekt kommt jedoch erst für $r < 90$ % zum Tragen.

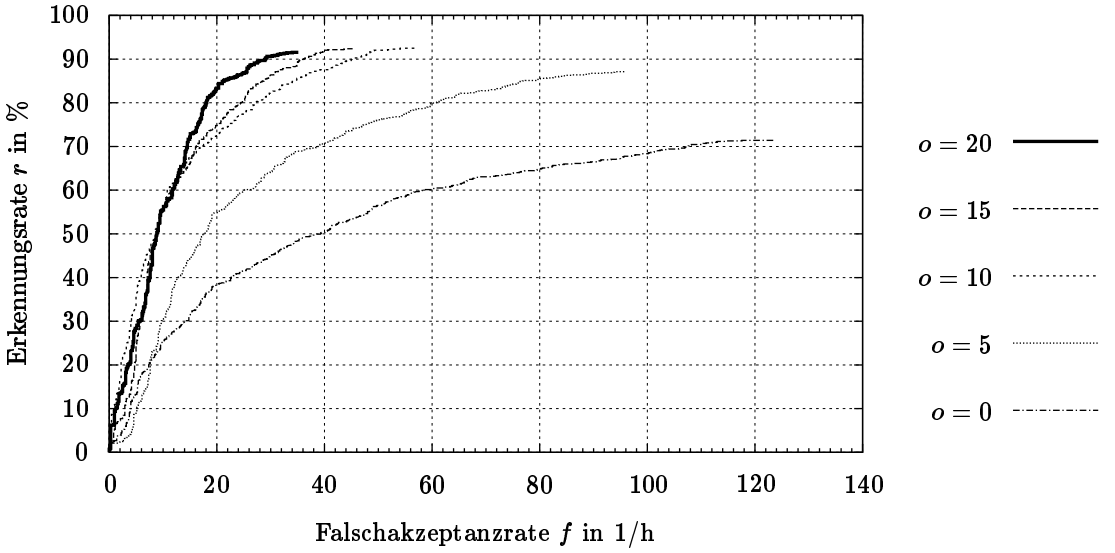


Bild 10.3: ROC-Darstellungen ausgewählter Erkennungsergebnisse aus Tabelle 10.21 für den Dialogdatensatz erzeugt durch Variation der Rückweisungsschwelle S_{rel} für verschiedene symmetrische Überhänge o ($L = 1024$, $W = 0$)

	Übungsdaten					Dialogdaten				
W	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
-10	97,40	16,27	0,57	35,77	36,77	87,22	37,62	5,16	54,85	55,85
-5	98,49	16,65	0,62	35,70	36,70	89,68	35,13	6,57	53,82	54,82
0	98,81	16,90	0,68	35,65	36,65	91,56	35,13	7,03	53,58	54,58
5	98,81	17,60	0,62	35,62	36,62	92,38	32,92	7,39	52,86	53,86
10	98,75	17,44	0,47	35,66	36,66	92,85	31,12	6,45	52,54	53,54
15	98,75	17,50	0,42	35,62	36,62	93,08	30,43	7,39	51,94	52,94
20	98,75	17,63	0,47	35,57	36,57	93,55	30,02	7,50	51,32	52,32
25	98,81	17,50	0,52	35,46	36,46	93,90	29,46	7,62	50,81	51,81
30	**98,75**	**17,69**	**0,42**	**35,39**	**36,39**	**94,37**	**28,77**	**8,32**	**50,30**	**51,30**
35	98,75	18,23	0,42	35,35	36,35	94,37	29,05	7,85	49,95	50,95
40	98,81	18,39	0,36	35,31	36,31	94,26	30,29	7,50	49,47	50,47
45	98,81	18,74	0,36	35,22	36,22	94,37	30,16	7,39	48,85	49,85
50	98,81	18,96	0,47	35,15	36,15	94,37	30,02	7,62	48,18	49,18
55	98,86	19,47	0,36	35,11	36,11	94,26	30,71	7,03	47,74	48,74
60	98,86	19,50	0,42	35,08	36,08	94,37	31,40	6,80	47,34	48,34

Tabelle 10.22: Erkennungsergebnisse des Spotting-Verfahrens mit Übungs- und Dialogdaten bei Variation des Score-Eingangsgewichtes W (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Überhang $o = 20$, Zustandszahl $N = 30$, Codebuchgröße $L = 1024$, optimale Einstellung fett hervorgehoben)

10.7.2.3 Weitere Optimierungen

Durch die Anwendung der Peak-Verstärkung (s. Kap. 9.3.4.2) kann die mittlere Erkennungsverzögerung auf Kosten der Erkennungsleistung reduziert werden (s. Anh. D.1). Mit der Modellierung der Verweildauer (s. Kap. 9.3.3) kann die Erkennungsleistung für Gesten-

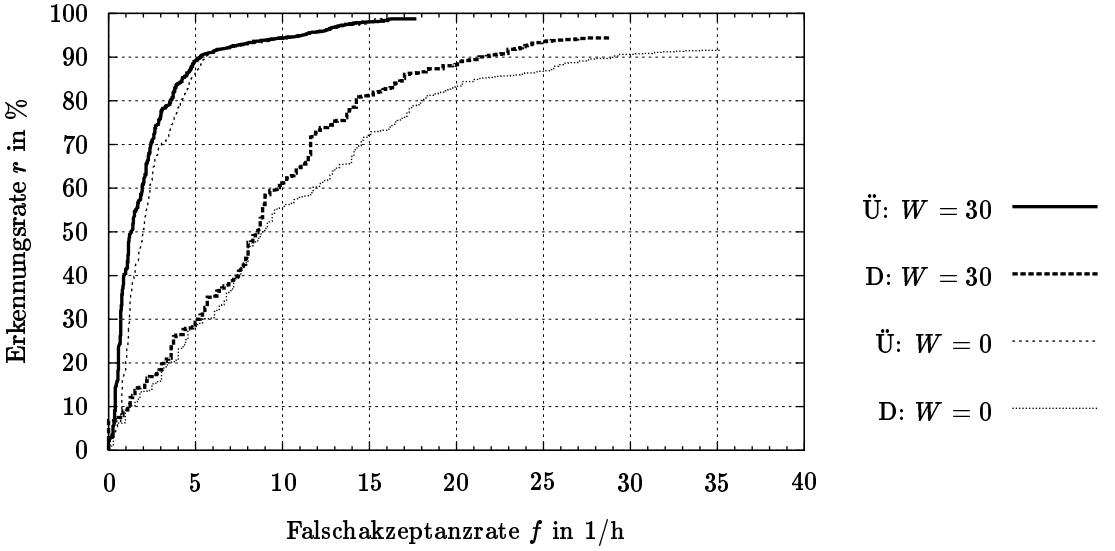


Bild 10.4: ROC-Darstellungen ausgewählter Erkennungsergebnisse aus Tabelle 10.22 erzeugt durch Variation der Rückweisungsschwelle S_{rel} für verschiedene Score-Eingangsgewichte W (Ü: Übungsdaten, D: Dialogdaten, $o = 20$, $N = 30$, $L = 1024$)

material, das den Übungsdaten ähnelt, deutlich verbessert werden (s. Anh. D.2). Durch geschickte Wahl des minimalen Peak-Abstandes (nach Regel P4 in Kap. 9.3.5) kann man den Verlauf der ROC etwas unterhalb der maximal erreichbaren Erkennungsrate günstig beeinflussen (s. Anh. D.3).

Diese Optimierungen gelten daher immer für spezielle Anwendungsfälle oder Dialogszenarien, die wiederum die Charakteristik der Gestendaten beeinflussen. Wenn diese Bedingungen gelten, so lassen sich die Erkennungseigenschaften des Spotting-Verfahrens jeweils deutlich verbessern.

10.8 Vergleich der Systeme

Für jede der Datenarten werden sehr unterschiedliche Ergebnisse erzielt. Es macht daher nur Sinn, die Ergebnisse getrennt nach Datenart zu vergleichen. Zum Vergleich werden jeweils die diskret vorliegenden Erkennungs- und Falschakzeptanzraten r und f des zweistufigen Systems (s. Tabellen 10.7 und 10.8) zusammen mit der jeweiligen ROC des Spotting-Systems (für ein Score-Eingangsgewicht von $W = 30$, s. Bild 10.4) dargestellt. Es werden alle vorhandenen Werte des zweistufigen Systems dargestellt, da die optimale r-f-Kombination im Vergleich zum Spotting-Systems nicht von vornherein feststeht. Außerdem vermittelt die so entstehende Punktewolke in Ermangelung eines Pendants zum ROC-Diagramm einen ungefähren Eindruck von der r-f-Charakteristik des zweistufigen Systems. Auf die Visualisierung und den Vergleich der mittleren Online-Erkennungsverzögerung $\bar{\tau}^{\text{vo}}$ und der Mehrfachdetektionsrate r_{M} kann in diesem Zusammenhang verzichtet werden, da beide Parameter für die Funktionsfähigkeit eines visuellen Dialogsystems unkritisch sind.

Die so entstandenen Bilder 10.5 und 10.6 zeigen, daß bei der jeweils optimalen Parametereinstellung der Verfahren sowohl für die Übungsdaten als auch für die Dialogdaten das Spotting-Verfahren dem zweistufigen Ansatz über den gesamten r-f-Bereich klar überlegen ist.

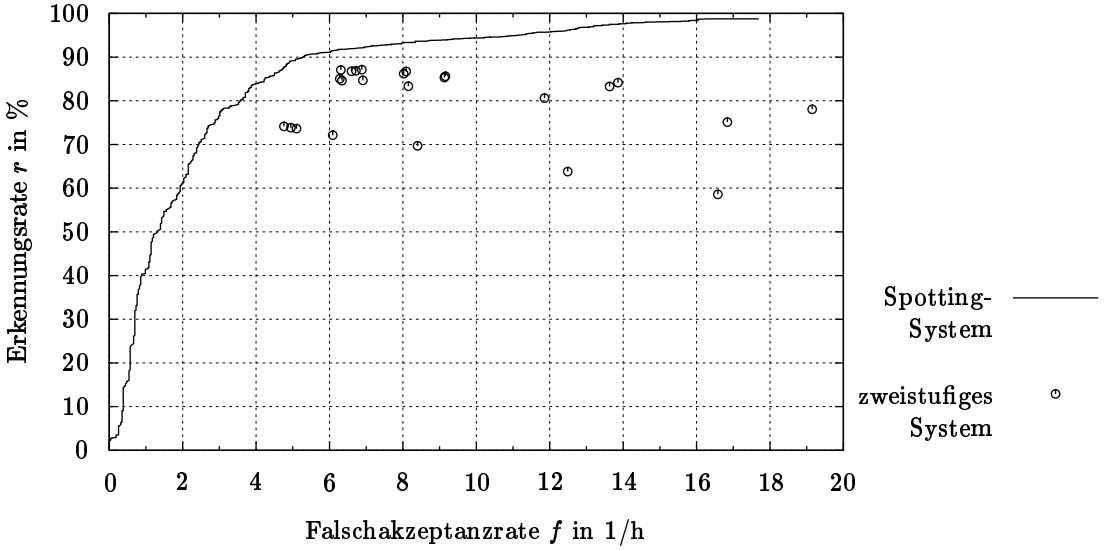


Bild 10.5: Vergleich der ROC-Darstellung des Spotting-Systems (s. Bild 10.4 für Score-Eingangsgewicht $W = 30$) mit den Ergebnissen des zweistufigen Systems aus Tabelle 10.7 für den Übungsdatensatz

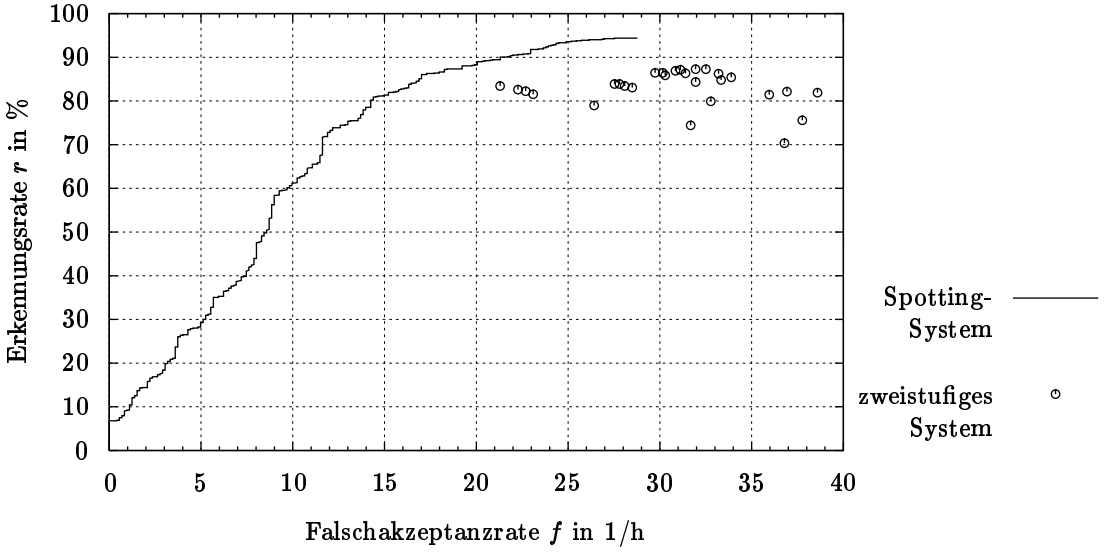


Bild 10.6: Vergleich der ROC-Darstellung des Spotting-Systems mit den Ergebnissen des zweistufigen Systems (s. Bild 10.4 für Score-Eingangsgewicht $W = 30$) mit den Ergebnissen des zweistufigen Systems aus Tabelle 10.9 für den Dialogdatensatz

Die Diagramme belegen auch, daß sich mit dem Spotting-Verfahren für die Erkennung *verbundener* Gesten ähnliche Erkennungsraten wie für die wesentlich einfachere Klassifikation *isolierter* Gesten erzielen lassen (vgl. Kap. 10.3). Dabei ist die kontinuierliche Erkennung nicht nur durch die zusätzliche zeitliche Segmentierungsaufgabe erschwert: da für die Evaluierung „unbehandelte" Datensätze aus den Usability-Experimenten verwendet wurden, muß im kontinuierlichen Fall auch noch für die Unterdrückung bedeutungsloser Bewegungen gesorgt werden (s. Kap. C.2). Daß dies dem Spotting-Verfahren besser als dem zweistufigen Ansatz gelingt, ist einer der Gründe für die überlegene Erkennungsleistung des Spotting-Ansatzes.

Kapitel 11

Der Demonstrator für den visuellen Dialog

11.1 Kontextunabhängige Demonstratorgesten

Die Evaluierungen des letzten Kapitels zeigen, daß eine optimale kontinuierliche Erkennung für Dialogdaten auf eine Codebuchgröße von $L = 1024$ und eine Zustandszahl von $N = 15$ bzw. 30 (zweistufiger bzw. Spotting-Ansatz) angewiesen ist. Betrachtet man die HMM-Rechenzeittabelle 8.15 auf Seite 96, so erkennt man, daß beide Systeme von der Möglichkeit eines Echtzeitbetriebes weit entfernt sind (vgl. Betrachtungen in Kap. 8.6.2). Selbst wenn bei der kontinuierlichen Erkennung eine Halbbildwiederholrate von $f_{\mathrm{R}} = 25\,\mathrm{Hz}$ ausreichen sollte — was für die kontinuierliche Erkennung nicht getestet wurde — so ergeben Messungen sowohl für die zweistufige als auch für die einstufige Erkennung einen Rechenzeitbedarf in der Größenordnung der zehnfachen Echtzeit.

Um dennoch einen echtzeitfähigen Demonstrator zu erhalten, wurden spezielle *Demonstratorgesten* konstruiert, die eine besonders einfache und robuste Erkennung ermöglichen. Außerdem sind diese Gesten speziell für den Einsatz des zweistufigen Systems geeignet, so daß die Echtzeitbedingung noch mehr entschärft wird: der Klassifikationsschritt des zweistufigen Systems muß nicht schritthaltend arbeiten (s. Kap. 8.6.2).

Die Demonstratorgesten unterscheiden sich von den Gesten des Übungs- oder des Dialogdatensatzes durch ihre Vor- und Nachbereitungsbewegung (vgl. Anh. C.1.4): Alle Gesten starten von einer definierten Ausgangslage (oder Ruhelage), münden in die Bewegung der Kerngeste und werden dann so abgeschlossen, daß sie wieder in der Ausgangslage enden. Auf diese Weise wurde zu jeder Kerngeste des Hauptkatalogs eine *symmetrische* Demonstratorgeste konstruiert (zum Begriff der Symmetrie s. Anh. C.2). Die Demonstratorgesten sind *kontextunabhängig*, da sie aufgrund ihrer Symmetrie beliebig hintereinander ausgeführt werden können, ohne daß sich Vor- und Nachbereitungsbewegungen ändern müssen.

Aufgrund ihrer Kontextfreiheit variieren die Demonstratorgesten auch beim Einsatz im Dialog nur sehr wenig, so daß einfache HMMs für Ihre Klassifikation ausreichen sollten. Die Ausgangs- und Endlage für alle Gesten ist eine Ruhelage mit entspannt ausgestreckten Fingern. Dadurch ergibt sich mit ein wenig Kooperation von Seiten des Benutzers eine kleine Pause zwischen den Gesten, die für einen sicheren Betrieb der Bewegungsdetektion ausreichen sollte.

g	optimale Detektionsparameter					Detektionsergebnisse					
	m_S	τ^l_{min}	τ^d_{min}	τ^b_{min}	τ^e_{min}	r_d	f_d	$r_{d,M}$	$\bar{\tau}^{va}_d$	$\bar{\tau}^{ve}_d$	r^{eff}_d
1	0,15	13	6	6	2	99,57	0,16	0,32	2,03	-2,27	99,09
2	0,15	13	3	2	3	99,79	0,16	0,48	1,92	-2,11	99,14
3	0,15	13	3	6	2	99,79	0,16	0,32	2,03	-2,27	99,30
4	0,15	13	5	6	2	99,73	0,16	0,32	2,03	-2,27	99,25

Tabelle 11.1: Ergebnisse der Bewegungsdetektion für die Demonstratordaten bei Variation des Gewichtungsfaktors der Optimierungsfunktion bei der jeweils optimalen Parametereinstellung (r_d, $r_{d,M}$, f_d und r^{eff}_d in %, τ-Werte in Merkmalszeittakten, $\tau^v_{d,max} = 50$, optimaler Bewegungswert $m_{H_0,t}$, Ergebnisse sind identisch für $g = 4$, 5, 6 und 7, für weitere Angaben vgl. Kap. 10.6.1)

11.2 Evaluierung des Demonstrators

Analog zum Übungsdatensatz wurde von *einer Person* ein *Demonstratordatensatz* aufgenommen, in der jede der Gesten des Hauptkataloges gleich oft vorkommt. Wegen der Kontextunabhängigkeit spielt die Reihenfolge, in der die Gesten aufgenommen wurden, keine Rolle. Die Gesten wurden bei normaler Raumbeleuchtung aufgenommen: die Aufnahmen enthalten also *Bewegungsunschärfe* (weitere Angaben s. Anh. C.1.4).

Alle Evaluierungen wurden bei einem Viertel der Halbbildgröße und bei einer Bildwiederholfrequenz von $f_R = 25\,\text{Hz}$ durchgeführt. Dies ist notwendig, damit räumliche Segmentierung und Merkmalsextraktion in Echtzeit durchgeführt werden können (s. Kap. 8.6.2).

Bei der Optimierung der Bewegungsdetektionsparameter wurde analog zum Verfahren in Kap. 10.6.1.1 vorgegangen. Tabelle 11.1 zeigt, daß zwar die optimalen Parameter unterschiedlich ausfallen, daß allerdings die Detektionsergebnisse sehr dicht beieinander liegen. Ab einem Optimierungsparameter von $g = 4$ sind die Ergebnisse völlig identisch (Untersuchung bis $g = 7$).

Entsprechend den Echtzeit-Anforderungen wurde das Gesamtsystem nur bis $L = 256$ evaluiert. Da die Gesten aufgrund ihrer symmetrischen Konstruktion schon einen immanenten Überhang aufweisen und relativ lange dauern (s. Tabelle C.2 in Kap. C.1.4 im Vergleich mit den anderen Datensätzen in Kap. C.1), wird kein weiterer Überhang benötigt ($o = 0$).

Es zeigte sich, daß bei nur $N = 2$ Zuständen ein Optimum der Erkennungsleistung erreicht wird, weshalb in Tabelle 11.2 nur Ergebnisse für diese Zustandszahl gezeigt werden. Die besten Ergebnisse ergeben sich für die Detektionsparameter, die zu einem Optimierungsparameter von $g = 3$ gehören. Bereits für $L = 128$ ergibt sich die sehr gute Erkennungsrate r von etwa 99 % bei einem vernachlässigbaren Fehler von 0,55 1/h. Dabei zeigt das System eine ebenfalls vernachlässigbare Online-Erkennungsverzögerung $\bar{\tau}^{do}$.

Dies ist eine Bestätigung dafür, daß die Konstruktion der Demonstratorgesten erfolgreich war: die Gesten arbeiten mit dem zweistufigen System ideal zusammen und benötigen HMMs mit nur $L = 128$ Prototypen und $N = 2$ Zuständen. Aus der Rechenzeittabelle 8.15 auf Seite 96 ergibt sich, daß dazu lediglich eine Ausführungszeit von etwas mehr als 0,2 s notwendig ist.

Im praktischen Betrieb hat sich gezeigt, daß viele Gesten des Systems personenunabhängig benutzbar sind, was allerdings nicht experimentell untermauert wurde. Offen-

		$g=1$					$g=2$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	92,91	3,17	0,05	-2,14	-0,14	93,82	2,93	0,05	-2,10	1,10
	64	96,24	1,68	0,05	-2,18	-0,18	96,89	1,56	0,05	-2,09	1,09
	128	98,82	0,55	0,00	-2,17	-0,17	98,98	0,65	0,00	-2,05	1,05
	256	99,09	0,41	0,05	-2,20	-0,20	99,36	0,46	0,05	-2,05	1,05
		$g=3$					$g=4$				
		r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$	r	f	r_{M}	$\bar{\tau}^{\mathrm{v}}$	$\bar{\tau}^{\mathrm{vo}}$
L	32	93,07	3,19	0,05	-2,14	-0,14	93,02	3,19	0,05	-2,14	-0,14
	64	96,46	1,68	0,05	-2,17	-0,17	96,40	1,68	0,05	-2,17	-0,17
	128	99,03	0,55	0,00	-2,17	-0,17	98,98	0,55	0,00	-2,17	-0,17
	256	99,30	0,41	0,05	-2,20	-0,20	99,25	0,41	0,05	-2,20	-0,20

Tabelle 11.2: Erkennungsergebnisse des Demonstrators für die Demonstratordaten bei Variation des sich aus g ergebenden optimalen Parametersatzes und der Codebuchgröße L (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, $N_{\mathrm{opt}} = 2$, $\tau^{\mathrm{v}}_{\mathrm{max}} = 50$, $o = 0$, Bewegungsdetektionsparameter s. Tabelle 11.1)

bar sorgt die erzwungene Ruhelage für ein Angleichen der Bewegungen bei verschiedenen Benutzern.

11.3 Aufbau des Demonstrators

Beim Demonstrator handelt es sich um den gestengesteuerten 3D-Szenen-Editor, wie er im Rahmen der Einführung des visuellen Dialogkonzeptes in Kap. 3.5 beschrieben wurde. Der Name des Demonstrators ist IVIS, was für *Intuitive Visual Interaction System*[1] steht. Die Komponenten des Demonstrators sind auf *drei Verarbeitungsblöcke* verteilt, die durch drei Prozesse repräsentiert werden. Die Prozesse kommunizieren über sog. *sockets* miteinander, was bedeutet, daß sie beliebig auf einen oder mehrere über ein Netzwerk gekoppelte Rechner verteilt werden können. Die Prozesse werden zentral durch ein graphisches Benutzungsinterface — dem sog. IVIS Dispatcher — gesteuert (s. Bild 11.1).

Die Funktionsblöcke werden durch Anklicken mit der Maus gestartet. Dabei wird durch die graphische Oberfläche eine bestimmte Reihenfolge erzwungen, so daß sich die Prozesse automatisch miteinander verbinden können. Ebenso können die Prozesse zentral wieder beendet werden. Für jeden Funktionsblock kann dabei interaktiv ein Rechner benannt werden, auf dem der jeweilige Prozeß laufen soll, und ein Rechner, auf dem die graphische Ausgabe des jeweiligen Prozesses angezeigt werden soll.

Den drei Verarbeitungsblöcken sind im einzelnen folgende Aufgaben zugeteilt:

Block 1: Hier erfolgt die Kameraansteuerung, die Bilddigitalisierung, die Vorverarbeitung, die räumliche Segmentierung, die Merkmalsvektorberechnung und die Bewegungsdetektion. Zur Kontrolle zeigt dieser Block das segmentierte Bild der Hand und die errechneten Mermalsvektoren auf dem Bildschirm an.

Da mit diesem Block die Kamera verbunden wird, muß hierfür ein Rechner mit entsprechender Hardwareausstattung verwendet werden. Für den Demonstrator wurde dafür eine *Silicon Graphics Indy* mit einer *140 MHz Mips R4400 CPU* verwendet.

[1] Intuitives visuelles Dialogsystem

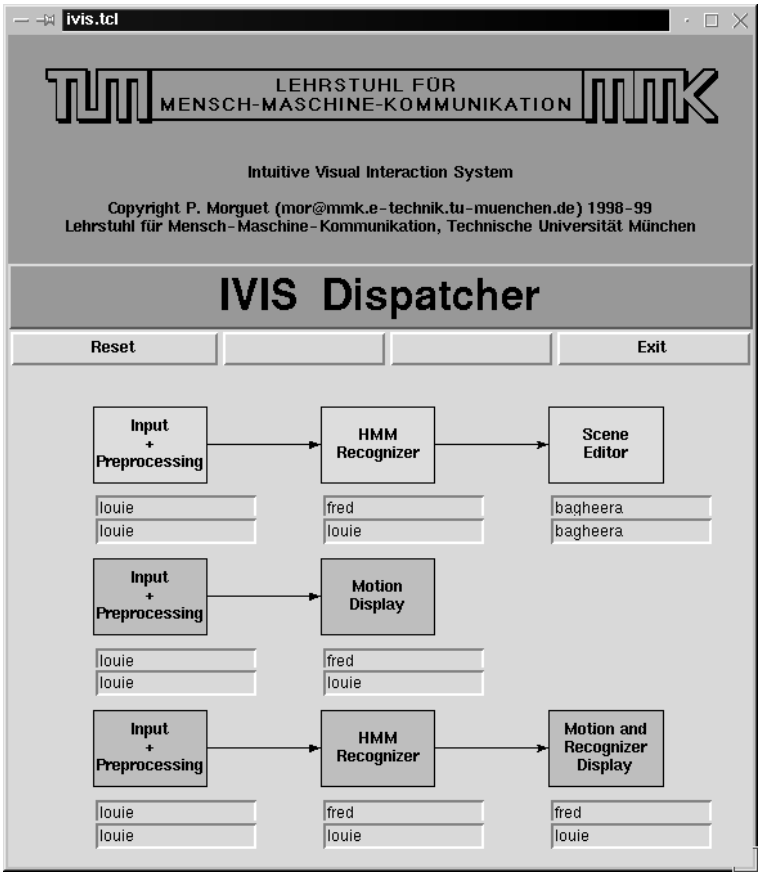


Bild 11.1: Graphische Benutzungsoberfläche des Demonstrators IVIS

Nur wenn die Bewegungsdetektionsstufe eine Bewegung erkannt hat, werden an den nächsten Block die zu einer Bewegung gehörigen Merkmalsvektoren geschickt. Dazu kommt die Information über Anfang und Ende eines Bewegungsintervalls.

Block 2: Der zweite Block nimmt die Merkmalsvektoren entgegen und führt zwischen der Anfangs- und Endemarkierung die Klassifikation durch. Dafür muß natürlich Zugriff auf die trainierten HMMs bestehen. Das Klassifikationsergebnis wird in einem Textfenster angezeigt.

Da die Klassifikation keine besonderen Anforderungen an die Hardware stellt, kann für diesen Verarbeitungsblock jeder vernetzte Rechner verwendet werden. Beim Demonstrator wurde für diesen Block in der Regel eine *Sun Ultra2* mit zwei *168 Mhz UltraSparc CPUs* eingesetzt (nur eine CPU wurde verwendet).

Über die Kommunikationsschnittstelle zum nächsten Verarbeitungsblock wird lediglich das Klassifikationsergebnis in Form eines Gestenindexes geschickt.

Block 3: Der dritte Block nimmt die Gestenindizes entgegen und sorgt mit der Dialogsteuerung, der Zustandsverwaltung sowie den Komponenten mit der Editor-Funktionalität und der dreidimensionalen Darstellung für den korrekten Ablauf und die Visualisierung der eigentlichen Anwendung, des 3D-Szenen-Editors.

Wegen dieses aufwendigen Rendervorgangs ist wiederum eine spezielle Hardware erforderlich. Für diesen Block wurde eine *Silcon Graphics Indigo2 High Impact* mit einer *250 MHz Mips R4400 CPU* mit 240 MHz eingesetzt.

Die Verarbeitungsschritte im ersten Block bis zur Berechnung der Merkmalsvektoren greifen alle direkt auf die Bildfunktion zu. Es ist zwingend notwendig, diese Verarbeitungsschritte im selben Prozeßblock zusammenzufassen, da sonst — trotz reduzierter Bildgröße und Bildwiederholrate — ein sehr großer Kommunikationsaufwand und bei verteilten Rechnern eine große Netzwerkbandbreite erforderlich wäre. Bei der gewählten Aufteilung der Funktionsblöcke ist jedoch nur eine geringe Bandbreite notwendig, so daß eine Vernetzung über ein normales 10 MBit Ethernet völlig ausreicht.

Zum genaueren Studium des Verhaltens der einzelnen Funktionsblöcke können über die Oberfläche in Bild 11.1 außer dem Gesamtsystem noch zwei Teilkonfigurationen verschaltet werden. Die erste Teilkonfiguration besteht aus dem ersten Funktionsblock und einer Anzeige des Ergebnisses der Bewegungsdetektion über eine graphische „Bewegungsampel". Bei der zweiten Konfiguration werden die ersten zwei Blöcke verschaltet, und das Ergebnis von Bewegungsdetektion und Klassifikation werden ebenfalls wieder graphisch angezeigt.

Obwohl es prinzipiell möglich ist, alle Blöcke auf einem Rechner laufen zu lassen, ist die Verteilung auf drei Rechner für einen reibungslosen Betrieb ideal. Dadurch wird garantiert, daß auf der Eingangsseite keine Bilder der Bewegungssequenz verloren gehen und auf der Ausgangsseite die graphische Animation ruckfrei abläuft (gehen zu viele Bilder verloren, so sinkt die Erkennungsleistung). Außerdem ist dadurch sichergestellt, daß *gleichzeitig* mit einer laufenden Klassifikation und Animation wieder neue Bewegungen detektiert werden können. Dies ist für einen intuitiven gestischen Dialog sehr empfehlenswert.

Prinzipiell ist allerdings auch mit nur einem Prozessor ein echtzeitfähiges System für den gestischen Dialog denkbar. Um einen sicheren Betrieb ohne Bildverluste zu gewährleisten, sollten dann allerdings die beiden ersten Blöcke nur im Wechsel betrieben werden: damit ist eine erneute Gesteneingabe immer nur nach Abschluß eines Klassifikationsschrittes möglich. Da bei der Bemessung des echtzeitkritischen ersten Blocks Spielraum gelassen wurde, wird die Leistung eines Prozessors auch noch für eine Visualisierung ausreichen, wenn sie mit etwas weniger Rechenleistung als der implementierte 3D-Szenen-Editor auskommt.

Mit dem hier vorgestellten Demonstrator konnten die Prinzipien eines gestischen Dialoges erfolgreich umgesetzt werden. Die notwendigen kleinen Einschränkungen, die sich aus der Echtzeitforderung eines Demonstrators ergeben, können beim zu erwartenden Fortschritt der Rechnertechnologie für zukünftige Systeme bald überwunden werden.

Kapitel 12

Schlußbetrachtungen und Ausblick

Mit dieser Arbeit konnte gezeigt werden, daß ein visueller Dialog mit Gesten zur vollständigen Steuerung einer graphischen Anwendung einsetzbar ist. Prinzipiell ist diese Art der Steuerung universell auf beliebige Anwendungen übertragbar, falls es gelingt, die erforderlichen Aktionen räumlich und damit gestengerecht umzusetzen. Das entwickelte Konzept für die visuelle Interaktion ist angelehnt an die zwischenmenschliche gestische Kommunikation und basiert auf dynamischen Gesten und dem Paradigma der indirekten Manipulation. In einem solchen Dialograhmen erweist sich die Bedienung mit Gesten als intuitiv und selbsterklärend.

Natürliche Gesten sind komplexe und stark variierende menschliche Bewegungen. Für die Modellierung der resultierenden Bildsequenzen wurden Hidden-Markov-Modelle als ein stochastischer Ansatz gewählt. Damit solche Modelle verwendet werden können, mußten Merkmalsextraktionsverfahren konzipiert werden, die die räumlich-zeitliche Bildsequenzinformation in eine rein zeitliche Abfolge von Merkmalsvektoren umsetzen. Es stellte sich heraus, daß verbesserte bzw. neu konzipierte pixelbasierte Verfahren in Verbindung mit der stochastischen Modellierung am besten geeignet sind. Zusammen mit einer leicht adaptierbaren Farbsegmentierung ist das resultierende Gesamtsystem auch mit einem großen Gestenvokabular universell einsetzbar und sehr robust.

Die Dialoganwendung erfordert eine kontinuierliche Erkennung: zusätzlich zur eigentlichen Klassifikation muß dazu eine zeitliche Segmentierung des Bildstromes durchgeführt werden. Es wurden zwei Ansätze für die kontinuierliche Erkennung von Gesten untersucht. Dabei ließen sich mit einem neuartigen einstufigen, integralen Spotting-Ansatz die besten Erkennungsleistungen erzielen. Außerdem konnte gezeigt werden, daß dieser Ansatz in der Lage ist, bedeutungstragende von bedeutungslosen Bewegungen zu trennen.

Das Gestenspotting beruht auf allgemeingültigen Prinzipien und ist daher universell zum Suchen von vorgegebenen Bewegungen in Videosequenzen geeignet. Neben der Dialoganwendung kann durch das Bewegungs-Spotting beispielsweise das immer mehr an Bedeutung gewinnende Problem der inhaltsbasierten Suche in Video-Datenbanken systematisiert und verbessert werden: bewegungsindizierende Verfahren sind bisher meist heuristischer Natur (s. beispielsweise [Den97, Sah97, Moh98]).

Mit einem einfacheren zweistufigen Ansatz — eine Verbindung einer Bewegungsdetektion mit einer isolierten Erkennung — waren die Erkennungsleistungen im Vergleich zum Spotting-Verfahren etwas schlechter. Allerdings stellt dieser Ansatz geringere Anforderungen an die Rechnerleistung, so daß in Verbindung mit speziell für diesen Ansatz

konzipierten Gesten erfolgreich ein echtzeitfähiger Demonstrator für den visuellen Dialog aufgebaut werden konnte.

Das größte Potential für Weiterentwicklungen steckt in der Integration weiterer Wissensquellen in die kontinuierliche Erkennung. Die einfachste Form besteht darin, dem Erkenner den inneren Zustand der Anwendung zurückzumelden, so daß ein zustandsabhängig eingeschränktes Gestenvokabular zur weiteren Steigerung der Erkennungsleistung führt.

Auch ist es denkbar, als zusätzliche Wissensquelle den Bewegungswert des zweistufigen Erkenners mit der Triggerschwelle des Gestenspotters zu verrechnen. Ebenso könnte der Bewegungswert zur genaueren Steuerung der Peak-Detektion verwendet werden.

Eine weitere Wissensquelle liegt in der typischen Abfolge von Gesten, die sich durch eine Gestefolgestatistik ausdrücken läßt. Diese statistische Abhängigkeit der Gesten untereinander ergibt sich dadurch, daß in der Anwendung bestimmte Ziele verfolgt werden, die durch eine typische Abfolge von Gesten erreicht werden können. Die Abhängigkeit erstreckt sich damit auch über einen größeren Zeitrahmen. Diese Gestenfolgestatistik entspricht dem Sprachmodell bei der kontinuierlichen Spracherkennung. Im Unterschied zur Spracherkennung ist diese Statistik zusätzlich zeitabhängig (dies ist beispielsweise auch im Histogramm der Gestenfolgezeiten in Bild 4.2 Seite 30 erkennbar).

In einem ersten Ansatz mit vorsegmentierten Gesten konnte nachgewiesen werden, daß sich die Erkennungsleistung durch die Integration der Gestenfolgestatistik in die Viterbi-Suche verbessern läßt [Ste99]. Die Verbesserung steigert sich nochmals durch Ausnutzung der Zeitabhängigkeit. Auch im zweistufigen kontinuierlichen Erkennungsansatz — also mit einer selbsttätigen Segmentierung der Gesten — konnte die Wirksamkeit der Gestenfolgemodellierung nachgewiesen werden. Diese Ergebnisse lassen es aussichtsreich erscheinen, die zeitabhängige Gestenfolgestatistik auch in das Gesten-Spotting zu integrieren.

Der nächste Schritt in der Verbesserung der Mensch-Maschine-Kommunikation besteht im Zusammenführen mehrerer Eingabekanäle zum *multimodalen Dialog*. Solche Systeme wurden ansatzweise immer wieder exemplarisch umgesetzt, indem Modalitäten paarweise zusammengeführt wurden (vgl. Kap. 3.1). Dabei wurde aber oft nur mit heuristischen Mitteln für einzelne Beispielfälle gearbeitet, so daß von einer echten Zusammenführung der Modalitäten nicht gesprochen werden kann.

Da durch die verwendete stochastische Modellierung der kontinuierlichen Gestikerkennung in dieser Arbeit methodische Parallelen zu existierenden Systemen beispielsweise zum Verstehen natürlicher Sprache [Mül97, Sta97] oder zur Interpretation von handgeschriebenen Formeln [Win96] bestehen, scheint eine konsistente Zusammenführung der einzelnen Ansätze zu einem synergetischen multimodalen Dialog möglich geworden zu sein (vgl. [Pot98]).

Dadurch wird man dem eigentlichen Ziel einer benutzeradäquaten Mensch-Maschine-Kommunikation um einiges näher kommen: der Mensch sollte sich weniger um die Bedienung des Systems kümmern müssen. Vielmehr sollte er sich frei auf die Aufgabe konzentrieren können, die er mit dem System lösen will [Lan94a, Lan94b].

Anhang A

Wichtige Formeln und Herleitungen

A.1 Formeln zu den Hu-Moment-Invarianten

Zwischen den algebraischen Invarianten I_{uv} der Ordnung $p = u + v$ und den normierten zentralen Momenten μ^{N} aus Gl. (7.6) Seite 63 besteht ein Zusammenhang, der in einem System aus $p+1$ linearen, komplexen Gleichungen besteht (geschlossener Ausdruck abgeleitet aus [Hu62]):

$$I_{p-r,r} = \sum_{k=0}^{p-2r} (-j)^k \begin{pmatrix} p-2r \\ k \end{pmatrix} \sum_{l=0}^{r} \begin{pmatrix} r \\ l \end{pmatrix} \mu^{\mathrm{N}}_{p-2l-k,2l+k} \qquad \text{mit } p-2r \geq 0,$$

$$I_{p/2,p/2} = \sum_{k=0}^{p/2} \begin{pmatrix} p/2 \\ k \end{pmatrix} \mu^{\mathrm{N}}_{p-2k,2k} \qquad \text{mit } p \text{ gerade.} \tag{A.1}$$

Die Invarianten mit zwei gleichen Indizes sind rein reell. Invarianten mit vertauschten Indizes sind zueinander konjugiert komplex:

$$I_{r,p-r} = I^*_{p-r,r}. \tag{A.2}$$

Die HMIs der Ordnung p werden aus algebraischen Invarianten bis zur Ordnung p gebildet. Bei der Ordnung 2 lassen sich 2, bei Ordnungen $p > 2$ allgemein genau $p+1$ unabhängige HMIs bilden. Für die verschiedenen Ordnungen gilt [Hu62, Bel91]:

- HMIs 2. Ordnung:

$$H_1 = I_{11} \quad \text{und} \tag{A.3}$$
$$H_2 = I_{20} I_{02}. \tag{A.4}$$

- HMIs 3. Ordnung:

$$H_3 = I_{30} I_{03}, \tag{A.5}$$
$$H_4 = I_{21} I_{12}, \tag{A.6}$$
$$H_5 = I_{30} I_{12}^3 + I_{03} I_{21}^3 \quad \text{und} \tag{A.7}$$
$$H_6 = I_{20} I_{12}^2 + I_{02} I_{21}^2. \tag{A.8}$$

- HMIs 4. Ordnung:

$$H_7 = I_{40}I_{04} = |I_{40}|^2, \tag{A.9}$$
$$H_8 = I_{31}I_{13} = |I_{31}|^2, \tag{A.10}$$
$$H_9 = I_{22}, \tag{A.11}$$
$$H_{10} = I_{31}I_{02} + I_{13}I_{20} \quad \text{und} \tag{A.12}$$
$$H_{11} = I_{40}I_{02}^2 + I_{04}I_{20}^2. \tag{A.13}$$

- Für $p > 4$ lassen sich allgemeine Bildungsgesetze für die jeweils $p+1$ HMIs angeben ($[p/2]$ ist dabei der ganzzahlige Anteil von $p/2$) (abgeleitet aus [Hu62]):

  1. $I_{p-r,r}I_{r,p-r}$, für $0 \leq r < [p/2]$,
  2. $I_{p/2,p/2}$, falls p gerade,
  3. $I_{p-r,r}I_{r-1,p-r-1} + I_{r,p-r}I_{p-r-1,r-1}$, für $1 \leq r < [p/2]$,
  4. $I^2_{[p/2],[p/2]+1}I_{20} + I^2_{[p/2]+1,[p/2]}I_{0,2}$, falls p ungerade,
  5. $I_{p/2-1,p/2+1}I_{2,0} + I_{p/2+1,p/2-1}I_{02}$, falls p gerade.

A.2 Formeln zu den Zernike-Moment-Invarianten

Zwischen den Zernike-Polynomen A_{nl} und den normalisierten Zentralmomenten μ^{N} besteht der Zusammenhang [Tea80]:

$$A_{nl} = \frac{n+1}{\pi} \sum_{f=l}^{n} \sum_{g=0}^{(f-l)/2} \sum_{h=0}^{l} (-j)^h \binom{\frac{f-l}{2}}{g} \binom{l}{h} B_{nlf} \cdot \mu^{\mathrm{N}}_{f-2g-l+h,2g+l-h}, \tag{A.14}$$

wobei $f-l$ gerade sein muß. Die Koeffizienten B_{nlf} lassen sich aus

$$B_{nlf} = \frac{(-1)^{(n-f)/2}[(n+f)/2]!}{[(n-f)/2]![(l+f)/2]![(f-l)/2]!} \tag{A.15}$$

berechnen [Bel91].

Bei der Bildung von ZMIs wird ausgenutzt, daß sich bei der Rotation der Bildfunktion die Phase der Zernike-Polynome linear ändert, während der Betrag gleich bleibt. Durch geeignete Kombination konjugiert komplexer Zernike-Momente lassen sich daher ZMIs bilden, die rotationsinvariant sind. Die Ordnung der ZMIs richtet sich nach der Ordnung der zugrundeliegenden normierten Zentralmomente. Die Zernike-Momente erster Ordnung sind trivial. Es lassen sich pro Ordnung exakt so viele unabhängige ZMIs wie HMIs bilden.

- ZMIs 2. Ordnung:

$$S_1 = A_{20} \quad \text{und} \tag{A.16}$$
$$S_2 = A_{22}A^*_{22}. \tag{A.17}$$

- ZMIs 3. Ordnung:

$$S_3 = A_{33}A_{33}^*, \tag{A.18}$$
$$S_4 = A_{31}A_{31}^*, \tag{A.19}$$
$$S_5 = A_{33}^*A_{31}^3 + A_{33}A_{31}^{*3} \quad \text{und} \tag{A.20}$$
$$S_6 = A_{22}^*A_{31}^2 + A_{22}A_{31}^{*2}. \tag{A.21}$$

- ZMIs 4. Ordnung:

$$S_7 = A_{44}A_{44}^*, \tag{A.22}$$
$$S_8 = A_{42}A_{42}^*, \tag{A.23}$$
$$S_9 = A_{40}, \tag{A.24}$$
$$S_{10} = A_{44}^*A_{42}^2 + A_{44}A_{42}^{*2} \quad \text{und} \tag{A.25}$$
$$S_{11} = A_{42}A_{22}^* + A_{42}^*A_{22}. \tag{A.26}$$

- ZMIs 5. Ordnung:

$$S_{12} = A_{55}A_{55}^*, \tag{A.27}$$
$$S_{13} = A_{53}A_{53}^*, \tag{A.28}$$
$$S_{14} = A_{51}A_{51}^*, \tag{A.29}$$
$$S_{15} = A_{51}^*A_{31} + A_{51}A_{31}^*, \tag{A.30}$$
$$S_{16} = A_{53}^*A_{33} + A_{53}A_{33}^* \quad \text{und} \tag{A.31}$$
$$S_{17} = A_{55}^*A_{31}^5 + A_{55}A_{31}^{*5}. \tag{A.32}$$

- ZMIs 6. Ordnung:

$$S_{18} = A_{66}A_{66}^*, \tag{A.33}$$
$$S_{19} = A_{64}A_{64}^*, \tag{A.34}$$
$$S_{20} = A_{62}A_{62}^*, \tag{A.35}$$
$$S_{21} = A_{60}, \tag{A.36}$$
$$S_{22} = A_{66}^*A_{33}^2 + A_{66}A_{33}^{*2}, \tag{A.37}$$
$$S_{23} = A_{64}^*A_{44} + A_{64}A_{44}^* \quad \text{und} \tag{A.38}$$
$$S_{24} = A_{62}^*A_{22} + A_{62}A_{22}^*. \tag{A.39}$$

- ZMIs 7. Ordnung:

$$S_{25} = A_{77}A_{77}^*, \tag{A.40}$$
$$S_{26} = A_{75}A_{75}^*, \tag{A.41}$$
$$S_{27} = A_{73}A_{73}^*, \tag{A.42}$$
$$S_{28} = A_{71}A_{71}*, \tag{A.43}$$
$$S_{29} = A_{77}^*A_{31}^7 + A_{77}A_{31}^{*7}, \tag{A.44}$$
$$S_{30} = A_{75}^*A_{55} + A_{75}A_{55}^*, \tag{A.45}$$
$$S_{31} = A_{73}^*A_{33} + A_{73}A_{33}^* \quad \text{und} \tag{A.46}$$
$$S_{32} = A_{71}^*A_{31} + A_{71}A_{31}^*. \tag{A.47}$$

Anhang B

Daten und zusätzliche Ergebnisse zu den Usability-Untersuchungen

B.1 Versuchsreihe 1 (VR 1)

B.1.1 Fragebogen VR 1

Der Fragebogen ist in fünf Themengruppen gegliedert, die als Zwischenüberschriften angegeben sind. In den obersten, hier nicht aufgeführten Feldern werden Angaben zu Name, Alter, Geschlecht und der Tätigkeit bzw. Ausbildung erbeten. Es folgen Fragen, die freie Antworten erfordern, und Fragen, die mit einer Bewertung von (1) bis (5) zu beantworten sind. Auch Ja/Nein-Fragen sollen auf einer Skala von (1) bis (5) beantwortet werden, wobei (1) einer starken Zustimmung und (5) einer starken Ablehnung entspricht.

Allgemeine Versuchsbedingungen

1. Wie war es für Sie, über Monitor und Kamera mit einem anderen Menschen zu kommunizieren?

   .....

2. War Ihnen dauernd (1) oder gar nicht (5) bewußt, daß die Programme von einem Menschen fernbedient wurden?

   ja (1) (2) (3) (4) (5) nein

3. a Wie empfanden Sie die Versuchsatmosphäre?

   angenehm (1) (2) (3) (4) (5) unangenehm

3. b Was fanden Sie dabei am angenehmsten oder unangenehmsten?

   .....

Bedienung mit Gesten

4. a Haben Sie während des Versuchs Alltagsgesten benutzt, oder mußten Sie sich die Gesten ausdenken?

alles Alltagsgesten (1) (2) (3) (4) (5) jede Geste ausgedacht

4. b Falls Gesten für diesen Versuch während des Ablaufs erfunden wurden: war dieser Erfindungsprozeß anstrengend oder ging das intuitiv?

anstrengend (1) (2) (3) (4) (5) intuitiv

5. Wären vorgegebene Gesten hilfreich gewesen?

ja (1) (2) (3) (4) (5) nein

6. Haben Sie eher eine oder beide Hände benutzt?

eher eine (1) (2) (3) (4) (5) beide

Konzept und Realisierung der Bedienung (Anwendung Auto-Einparksimulator)

7. a Hätten Sie beim Einparken an einen Menschen andere Erwartungen gehabt?

ja (1) (2) (3) (4) (5) nein

7. b Wenn ja, welche?

.....

Konzept und Realisierung der Bedienung (Anwendung 3D-Szenen-Editor)

8. War die Reaktionszeit zu langsam oder zu schnell ((3) = genau richtig)?

zu langsam (1) (2) (3) (4) (5) zu schnell

9. Fanden Sie den Dialogpartner ansprechend?

ja (1) (2) (3) (4) (5) nein

10. Entsprachen die Reaktionen des Dialogpartners Ihren Absichten?

ja (1) (2) (3) (4) (5) nein

11. Hat der Dialogpartner so reagiert, wie Sie es von einem Menschen in einer vergleichbaren Situation erwarten würden?

ja (1) (2) (3) (4) (5) nein

Frage Nr.	Versuchsperson Nr.														∅
	1	2	3	4	5	8	9	10	11	12	13	14	15	16	
2.						3	3	4	4	4	3	3	2	2	3,1
3. a						2	2	2	2	2	3	3	2	1	2,1
4. a	1	3	2	2	1	2	3	1	2,5	4	3	2	2	2	2,2
4. b	5	4	4	5	5	4	4	–	4	4	2	4	5	1,5	4,0
5.	5	4	5	2	5	5	3	5	2	2	1	1	4	1	3,2
6.	4	3	1	1	1	2	1	1	1	3	2	1	1	1	1,6
7. a						5	5	1	1	1	1	1	5	5	2,8
8.	2,5	2	2	2	2	2	3	3	2	3	3	1	2	2,5	2,3
9.	1	3	1	–	1	1	–	2	1	4	3	4	1	–	2,0
10.	1	3	2	2	3	2	2	2	2	4	3	3	3	3	2,5
11.	2	3	3	–	2	2	2	3	2	3	3	3	3	2	2,5
12.	3	3	4	2	4	1	1	3	2	4	4	5	1	2,5	2,8

Tabelle B.1: Ergebnisse der quantitativen Auswertung der Fragebogen für VR 1 (Mittelwert ∅)

Sonstiges

12. Haben Sie Erfahrung im Umgang mit Computern?

    erfahrene(r) Programmierer(in) (1) (2) (3) (4) (5) keine Erfahrung

13. Kommentare:

    .....

B.1.2 Quantitative Auswertung VR 1

In Tabelle B.1 sind die Ergebnisse der quantitativen Auswertung der Fragebogen für Versuchsreihe 1 über alle Versuchspersonen aufgeführt. An beiden Versuchsreihen zusammen haben 17 VPs teilgenommen. Die Nummern der Versuchspersonen wurden für VR 1 und VR 2 zusammen durchgezählt. Nummern von Personen, die in der Tabelle nicht auftauchen, haben also nur an VR 2 teilgenommen. Für die ersten fünf Versuchspersonen wurde eine reduzierte Form des Fragebogens verwendet, was die Lücken in der Tabelle erklärt. Nicht beantwortete Fragen werden mit einem Gedankenstrich gekennzeichnet. Teilweise gaben die Versuchspersonen auch Zwischenwerte oder Wertebereiche an. Dies erklärt die Nachkommastellen bei einigen Tabellenwerten. In der letzten Spalte ist der Mittelwert (∅) über die jeweilige Zeile angegeben. Die Auswertungen werden in Kap. 4.2.1 besprochen.

B.2 Versuchsreihe 2 (VR 2)

B.2.1 Fragebogen VR 2

Der Fragebogen ist im Unterschied zum Fragebogen zur VR 1 nur noch in vier Themengruppen gegliedert, weil nur noch mit dem 3D-Szenen-Editor gearbeitet wurde. Die

Fragen mußten an die geänderten Versuchsbedingungen angepaßt werden, weshalb sich lediglich sieben der Fragen mit dem ersten Bogen überschneiden. Ansonsten gilt das unter Anh. B.1.1 einleitend Gesagte:

Allgemeine Versuchsbedingungen

1. Hat es Sie gestört, mit einem Computer wie mit einem Menschen zu kommunizieren?

   ja (1) (2) (3) (4) (5) nein

2. Woran konnte man merken, daß ein Computer die Gesten interpretiert?

   .....

3. Haben Sie die Gesten mit Rücksicht auf den Computer bewußt deutlicher gemacht?

   viel Rücksicht genommen (1) (2) (3) (4) (5) keine Rücksicht genommen

4. Wie empfanden Sie die Versuchsatmosphäre?

   angenehm (1) (2) (3) (4) (5) unangenehm

Bedienung mit Gesten

5. Waren die vorgegebenen Gesten fremd oder Alltagsgesten?

   fremd (1) (2) (3) (4) (5) Alltagsgesten

6. War es schwierig, mit den vorgegebenen Gesten zu arbeiten?

   ja (1) (2) (3) (4) (5) nein

7. a Fanden Sie „Ihre“ Handgesten aus der ersten Versuchsreihe wieder?

   ja, alle (1) (2) (3) (4) (5) nein, nur neue Gesten

7. b Falls Sie bestimmte Handgesten vermißt haben: welche waren das?

   .....

8. War die Beschränkung auf eine Hand hinderlich?

   ja (1) (2) (3) (4) (5) nein

9. Haben Sie Kopfgesten vermißt?

   ja (1) (2) (3) (4) (5) nein

10. Empfanden Sie die Gewöhnung an die ruhige Gestenfolge als schwierig?

    ja (1) (2) (3) (4) (5) nein

Konzept und Realisierung der Bedienung

11. Fühlten Sie sich durch die Kontrollampe unter Druck gesetzt?

ja (1) (2) (3) (4) (5) nein

12. War die Reaktionszeit zu langsam oder zu schnell ((3) = genau richtig)?

zu langsam (1) (2) (3) (4) (5) zu schnell

13. Fanden Sie den Dialogpartner ansprechend?

ja (1) (2) (3) (4) (5) nein

14. Entsprachen die Reaktionen des Dialogpartners Ihren Absichten?

ja (1) (2) (3) (4) (5) nein

15. Hat der Dialogpartner so reagiert, wie Sie es von einem Menschen in einer vergleichbaren Situation erwarten würden?

ja (1) (2) (3) (4) (5) nein

Sonstiges

16. Haben Sie Erfahrung im Umgang mit Computern?

erfahrene(r) Programmierer(in) (1) (2) (3) (4) (5) keine Erfahrung

17. Kommentare:

.....

B.2.2 Quantitative Auswertung VR 2 und Vergleich der VRs

In Tabelle B.2 stehen die Ergebnisse der Versuchsreihe 2 über alle Versuchspersonen. Es gelten dieselben Erläuterungen wie zur Tabelle von Versuchsreihe 1 in Anh. B.1.2. 11 VPs sind mit denen aus VR 1 identisch, 3 sind neu.

Die Tabelle B.3 enthält einen Vergleich der Fragebogenauswertungen zwischen Versuchsreihe 1 und 2 über alle Versuchspersonen. Es sind direkt vergleichbare Fragen enthalten, die in beiden Versuchsreihen vorhanden sind (fett hervorgehoben), und Fragen, deren Aussagen sich sinngemäß in Beziehung setzen lassen. Mit dieser Tabelle soll sichtbar gemacht werden, wie sich die Bewertungen verschoben haben. Auswertung und Gegenüberstellung werden in Kap. 4.2.2 besprochen.

Frage Nr.	Versuchsperson Nr.														∅
	2	3	6	7	8	9	10	11	12	13	14	15	16	17	
1.					5	5	5	5	5	4	3	5	4	4	4,5
3.					2	2	3	2	4	2	1	4	1	2	2,3
4.	3	3	4	1	3	1	2	3	2	2	2	3	2	1	2,3
5.	4	4	3,5	2,5	4	4	5	4	2	4	4	3	4	5	3,8
6.	2	5	5	3	5	5	5	3	4	2	5	4	4	5	4,1
7. a					1	3	1	3	2	3	3	2	2	2	2,2
8.	5	3	1	4	5	5	5	4,5	2	1	5	5	4	4	3,8
9.					5	5	5	5	5	1	5	5	5	5	4,6
10.	3	3	1	3	5	1	5	3	4	4	5	4	4	4	3,5
11.					5	5	5	4	4	3	5	5	4	5	4,5
12.	2	3	2	2	3	1,5	3	2,5	3	3	3	2	3	3	2,6
13.	3	1	5	2	1	–	2	2	2	2	3	1	2	3	2,2
14.	2	1	2	2	2	–	2	3	2	2	2	2	2	1	1,9
15.	2	2	1	2	2	4	2	2,5	2	2	2	3	2	1	2,1
16.	3	4	3	2	1	1	3	2	4	4	4,5	1	2	3	2,7

Tabelle B.2: Ergebnisse der quantitativen Auswertung der Fragebogen für VR 2 (Mittelwert ∅)

Frage Nr. in VR 1	3. a	4. a	5.	6.	8.	9.	10.	11.	12.
Frage Nr. in VR 2	4.	5.	6.	8.	12.	13.	14.	15.	16.
Bewertung VR 1	**2,1**	2,2	3,2	1,6	**2,3**	**2,0**	**2,5**	**2,5**	**2,8**
Bewertung VR 2	**2,3**	3,8	4.1	3.8	**2,6**	**2,2**	**1,9**	**2,1**	**2,7**

Tabelle B.3: Vergleich der quantitativen Fragebogenauswertungen von VR 1 und 2 (direkt vergleichbare Zahlen sind fett gedruckt)

B.3 Die Gestenkataloge

B.3.1 Hauptkatalog

Der Hauptkatalog hat sich als ein Ergebnis der Auswertungen der Videobänder aus der Versuchsreihe 1 ergeben. In den Aufzeichnungen fanden sich ca. 70 verschiedene Gesten. Diese große Variabilität ist die Konsequenz der Tatsache, daß die Gesten frei gewählt werden konnten. Durch Häufigkeitsauswertungen konnte die Anzahl der Gesten auf die in Tabelle B.4 aufgeführten 41 Gesten reduziert werden — ein Kompromiß zwischen Flexibilität der Bedienung und dem zu investierenden Erkennungsaufwand.

Die Tabelle zeigt neben der laufenden Nummer und der verbalen Beschreibung der Gesten des Hauptkataloges eine dreistufige Gliederung in *Funktion*, *Erscheinungsform* und *Richtung*. Diese Gliederung ist auf die Anwendung der Gesten auf den 3D-Szenen-Editor bezogen (s. Kap. 3.5).

Dabei beschreibt die *Funktion* einen von acht abstrakten Vorgängen, den eine Geste in der Anwendung auslöst. Die Funktion ergibt zusammen mit der *Richtung* eine Gesten*kategorie* oder ein spezifisches Gestenkommando, wie es beispielsweise auch der Versuchsleiter zur Steuerung des 3D-Szenen-Editors in den Wizard-of-Oz-Versuchen eingeben mußte (s. Kap. 4.1). Je nach zugrundeliegender Funktion gibt es keine oder bis zu

sechs verschiedene Richtungen. Wie aus Tabelle B.6 hervorgeht, existieren 16 verschiedene Kategorien. Bei gleichbleibender Kategorie gibt es bis zu vier verschiedene zugelassene *Erscheinungsformen* der Gesten, womit sichergestellt wird, daß die Anwendung flexibel und intuitiv bedienbar ist. Die in der Tabelle zwischen Nr. 23 und 24 aufgeführte Geste ergibt sich aus der Gliederung. Sie ist aber nicht von Geste 22 der gleichen Kategorie zu unterscheiden, weshalb sie keine Nummer erhält.

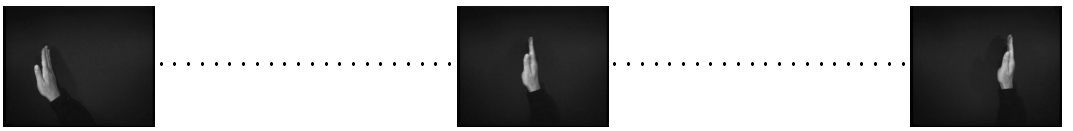
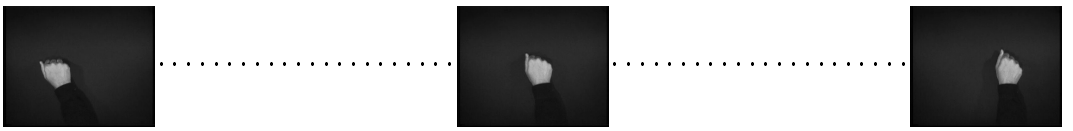
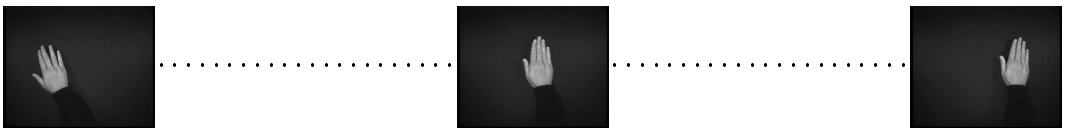
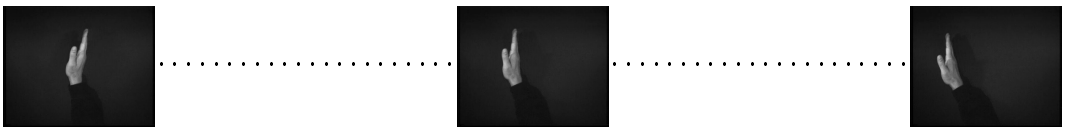
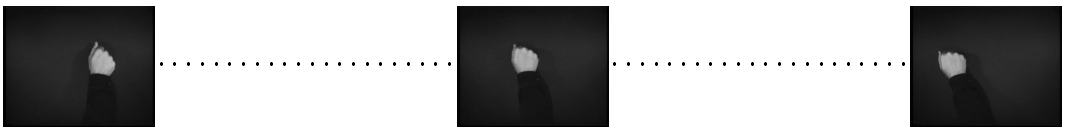
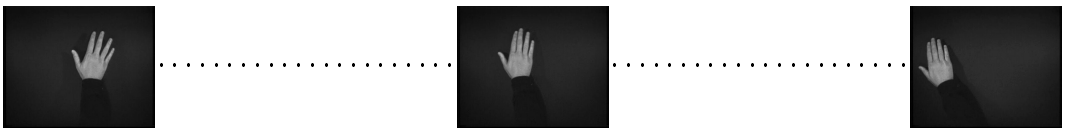
Nr.	Funktion	Erscheinungsform	Richtung	Beschreibung
1	Verschieben	winkend	rechts	aus dem Handgelenk winken (Handfläche in Winkrichtung)
2	Verschieben	schiebend	rechts	mit der flachen Hand schieben (Handfläche in Bewegungsrichtung)
3	Verschieben	Faust	rechts	mit der Faust schieben (Handrücken immer oben)
4	Verschieben	flach	rechts	mit der flachen Hand Bewegungsrichtung vormachen (Handrücken immer oben)
5	Verschieben	winkend	links	aus dem Handgelenk winken (Handfläche in Winkrichtung)
6	Verschieben	schieben	links	mit der flachen Hand schieben (Handfläche in Bewegungsrichtung)
7	Verschieben	Faust	links	mit der Faust schieben (Handrücken immer oben)
8	Verschieben	flach	links	mit der flachen Hand Bewegungsrichtung vormachen (Handrücken immer oben)
9	Verschieben	winken	zurück	aus dem Handgelenk winken (Handfläche in Winkrichtung)
10	Verschieben	schieben	zurück	mit der flachen Hand schieben (Handfläche in Bewegungsrichtung)
11	Verschieben	Faust	zurück	mit der Faust schieben (Handrücken immer oben)
12	Verschieben	flach	zurück	mit der flachen Hand Bewegungsrichtung vormachen (Handrücken immer oben)
13	Verschieben	winken	vor	aus dem Handgelenk winken (Handfläche in Winkrichtung)
14	Verschieben	schieben	vor	mit der flachen Hand schieben (Handfläche in Bewegungsrichtung)
15	Verschieben	Faust	vor	mit der Faust schieben (Handrücken immer oben)
16	Verschieben	flach	vor	mit der flachen Hand Bewegungsrichtung vormachen (Handrücken immer oben)
17	Verschieben	winken	hoch	aus dem Handgelenk winken (Handfläche in Winkrichtung)

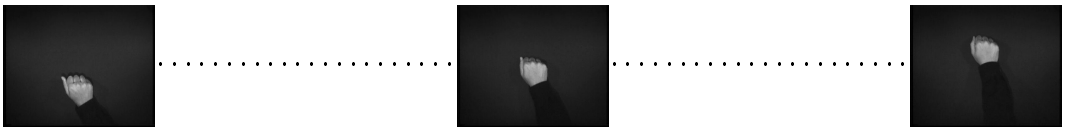
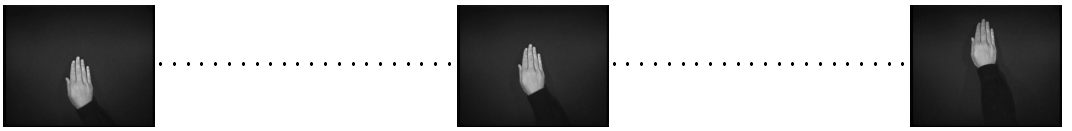
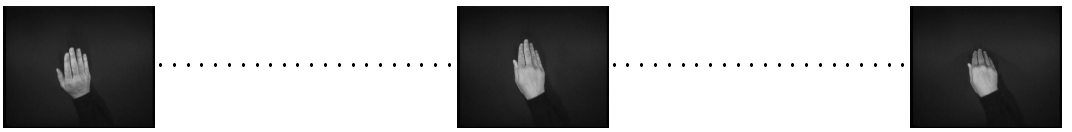
Nr.	Funktion	Erscheinungsform	Richtung	Beschreibung
18	Verschieben	schieben	hoch	mit der flachen Hand schieben (Handfläche in Bewegungsrichtung)
19	Verschieben	Faust	hoch	mit der Faust schieben (Handrücken immer oben)
20	Verschieben	flach	hoch	mit der flachen Hand Bewegungsrichtung vormachen (Handrücken immer oben)
21	Verschieben	winken	hinunter	aus dem Handgelenk winken (Handfläche in Winkrichtung)
22	Verschieben	schieben	hinunter	mit der flachen Hand schieben (Handfläche in Bewegungsrichtung)
23	Verschieben	Faust	hinunter	mit der Faust schieben (Handrücken immer oben)
—	Verschieben	flach	hinunter	mit der flachen Hand Bewegungsrichtung vormachen (Handrücken immer oben); identisch mit Geste Nr. 22
24	Drehen	fünf Finger	rechts	fünf Finger ausgestreckt nach unten (wie wenn Kugel umfaßt wird)
25	Drehen	Zeigefinger	rechts	wie beim Wählen mit Wählscheibe („Telefongeste“)
26	Drehen	fünf Finger	links	fünf Finger ausgestreckt nach unten (wie wenn Kugel umfaßt wird)
27	Drehen	Zeigefinger	links	wie beim Wählen mit Wählscheibe („Telefongeste“)
28	Kippen	Zeigefinger	hoch	ausgestreckter Zeigefinger wird nach oben geführt
29	Kippen	Zeigefinger	hinunter	ausgestreckter Zeigefinger wird nach unten geführt
30	Skalieren	zwei Finger	größer	Daumen und Zeigefinger bewegen sich auseinander
31	Skalieren	beidhändig	größer	zwei flache Hände bewegen sich auseinander (Handflächen nach innen)
32	Skalieren	zwei Finger	kleiner	Daumen und Zeigefinger bewegen sich aufeinander zu
33	Skalieren	beidhändig	kleiner	zwei flache Hände bewegen sich aufeinander zu (Handflächen nach innen)
34	Stoppen	einhändig	—	flache Hand wird hochgerissen
35	Stoppen	beidhändig	—	zwei flache Hände überkreuzen sich (Start außen)
36	Freigeben	Faust	—	Faust öffnet sich
37	Freigeben	wischen	nach rechts	wischen von links nach rechts und wieder zurück
38	Freigeben	wischen	nach links	wischen von rechts nach links und wieder zurück

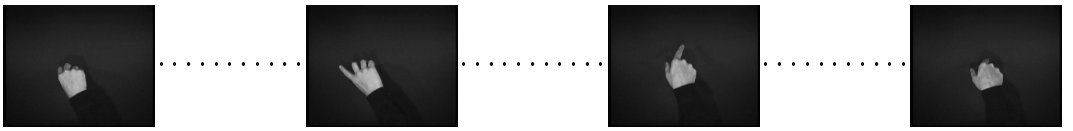
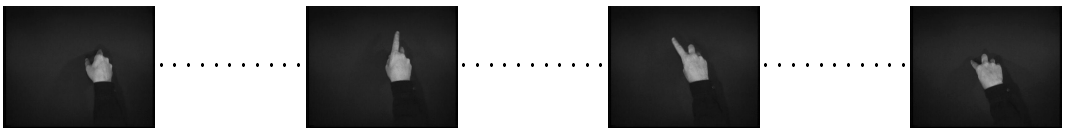
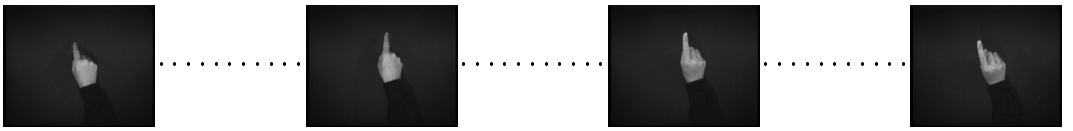
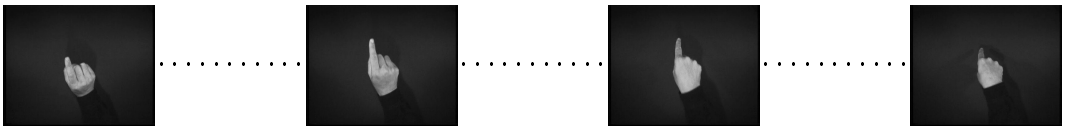
Nr.	Funktion	Erscheinungsform	Richtung	Beschreibung
39	Zeigen	Zeigefinger	—	Zeigefinger aus Faust ausfahren
40	Auslösen	greifen	—	geöffnete Hand schließt sich zur Faust
41	Auslösen	drücken	—	Zeigefinger aus Faust ausfahren, Druck nach vorne, Zeigefinger einfahren

Tabelle B.4: Die Gesten des Hauptkataloges mit laufender Nummer, den drei Gliederungsebenen Funktion, Erscheinungsform und Richtung sowie einer verbalen Beschreibung der Geste

Um die Gesten zusätzlich zur verbalen Beschreibung in Tabelle B.4 zu illustrieren, werden in Tabelle B.5 typische Momentaufnahmen aus den Gesten-Bildsequenzen gezeigt. Dazu gehört außer einem Bild zu Beginn und Ende der Geste noch mindestens ein Bild, mit dem sich der Verlauf der Bewegung am besten charakterisieren läßt:

Nr.	Momentaufnahmen
1	
2	
3	
4	
5	
6	
7	
8	

Nr.	Momentaufnahmen
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	

Nr.	Momentaufnahmen
22	
23	
—	
24	
25	
26	
27	
28	
29	
30	
31	
32	

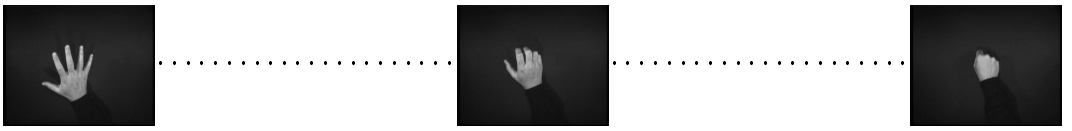
Nr.	Momentaufnahmen
33	
34	
35	
36	
37	
38	
39	
40	
41	

Tabelle B.5: Charakteristische Momentaufnahmen zu den Gesten des Hauptkataloges

In der Tabelle B.6 werden zu den einzelnen Gesten mit den angezeigten Nummern noch weitere Eigenschaften zusammenfassend aufgelistet, auf die jeweils im Haupttext Bezug genommen wird. Der Begriff *Komplementärgeste* wird in Anh. C.2 erklärt. Die *Klasse* einer Geste wird in Kap. 2.5 eingeführt.

B.3.2 Der Nebenkatalog

Der Nebenkatalog enthält 12 Gesten, die im wesentlichen einen Ausschnitt aus dem Hauptkatalog darstellen. Dabei handelt es sich um die neun Gesten-Nummern 1, 5, 9, 13, 34, 36, 38, 39, 40 (s. Tabelle B.4), eine Dialog-Verneinungsgeste (Zeigefinger hin und her schwenken, Start nach links) sowie zwei Variationen der Greifgeste Nr. 40 (ein Greifvor-

Gesten-Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Kategorie-Nr.	1	1	1	1	2	2	2	2	3	3	3	3	4	4
Anzahl der Hände	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Symmetrie	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Komplementärg.-Nr.	5	6	7	8	1	2	3	4	13	14	15	16	9	10
Klasse	k	m	m	k	k	m	m	k	k	m	m	k	k	m
Gesten-Nr.	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Kategorie-Nr.	4	4	5	5	5	5	6	6	6	7	7	8	8	9
Anzahl der Hände	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Symmetrie	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Komplementärg.-Nr.	11	12	21	22	23	22	17	18	19	26	27	24	25	29
Klasse	m	k	k	m	m	k	k	m	m	m	m	m	m	m
Gesten-Nr.	29	30	31	32	33	34	35	36	37	38	39	40	41	
Kategorie-Nr.	10	11	11	12	12	13	13	14	14	14	15	16	16	
Anzahl der Hände	1	1	1	2	1	2	1	2	1	1	1	1	1	
Symmetrie	n	n	n	n	n	n	j	n	j	j	n	n	j	
Komplementärg.-Nr.	28	32	3	30	31	21	-	40	38	37	-	36	-	
Klasse	m	s	s	s	s	s	s	m	s	s	m	m	m	

Tabelle B.6: Eigenschaften der Gesten des Hauptkatalogs (Symmetrie j: ja, n: nein; Klasse m: mimisch, k: kinemimisch, s: symbolisch, Gesten-Nr. nach Tabelle B.4)

gang verbunden mit einer Handdrehung und einem Schwenk nach links bzw. nach rechts). Bei der Aufnahme der Gesten wurde versucht, immer wieder denselben Anfangspunkt zu treffen, so daß auch eine *unnormierte* Erkennung möglich ist.

Die Gesten wurden nicht aus dialogtechnischen Gründen zusammengestellt. Vielmehr wurde ein reduzierter Katalog erzeugt, in dem Gesten enthalten sind, die schwierig auseinanderzuhalten sind, so daß die Anforderungen an die Erkennung erhöht wurden. Aus diesem Grund sind Gesten enthalten, die sich bei einer identischen Trajektorie nur durch Gestaltänderungen und einen spezifischen Geschwindigkeitsverlauf unterscheiden (wie beispielsweise Winken und Greifen nach Links).

Anhang C

Einzelheiten zu Trainings- und Testdaten

C.1 Angaben zu den verschiedenen Datensätzen

Für die räumliche Segmentierung wurde als optimales Verfahren eine UV-Farbsegmentierung auf Basis der Vordergrund-LUT bestimmt (s. Kap. 5.1.6). Da dieses Verfahren bei einer konstanten Beleuchtung, einem schwarzen Hintergrund und entsprechender Parameterjustierung praktisch fehlerfrei arbeitet, wurde das gesamte Trainings- und Testmaterial unter diesen Bedingungen aufgenommen, damit mögliche Segmentierungsfehler von vornherein ausgeschlossen werden können. Wie in Kap. 4 über die Usability-Experimente beschrieben, wurde die Kamera senkrecht mit einem Abstand von rund 85 cm über der Arbeitsfläche montiert. Es wurde die CCD-Kamera *XC-003 P* der Firma *Sony* mit drei 1/3-Zoll Sensoren und einem Weitwinkelobjektiv mit 6 mm Brennweite verwendet. Die Beispielbilder im illustrierten Gestenkatalog (s. Tabelle B.5 in Kap. B.3.1) zeigen den Bildausschnitt, der sich aus diesen Einstellungen ergibt.

Die Gesten des Analysedatensatzes (s. Anh. C.1.3) wurden einzeln direkt auf Festplatte aufgenommen. Die Aufnahme erfolgte unkomprimiert mit ein Viertel der Halbbildgröße bei voller Bildwiederholrate. Dafür wurden ca. 4 Gigabyte Plattenplatz benötigt. Die anderen Datensätze wurden kontinuierlich bei voller Halbbildgröße und bei voller Bildwiederholrate aufgenommen. Dabei wurde ein hardwareunterstütztes Motion-JPEG-Verfahren[1] zur Kompression verwendet, mit dem die Daten etwa im Verhältnis 1:30 komprimiert werden konnten. Diese hohe Komprimierung war bei annehmbarer Qualität nur aufgrund des homogenen Hintergrundes möglich. Eine Stunde komprimierten Videomaterials benötigt ca. 2,7 Gigabyte Plattenplatz.

Alle Erkennungsergebnisse beruhen wenn möglich auf einer strikten Trennung von Trainings- und Erkennungsdaten. Als Grundprinzip wurden idealerweise zwei Drittel der Daten für das Training verwendet und ein Drittel für die Erkennung. Um dabei in Training und Erkennung möglichst die Variation über die Zeit zu berücksichtigen, wurden Trainings- und Testdaten ineinander „verschränkt“ über den jeweiligen gesamten Datensatz verteilt.

[1] JPEG (*Joint Photographic Experts Group*) ist ein Kompressionsverfahren für Einzelbilder. *Motion*-JPEG ist für Videosequenzen geeignet, da die Halbbilder getrennt komprimiert werden, so daß die durch das Zeilensprungverfahren hervorgerufenen Bewegungsartefakte die Kompression nicht erschweren können [Sch94b].

C.1.1 Übungsdaten

Die Übungsdaten stammen aus dem 1. Teil der 2. Versuchsreihe: hier wurde mit den Probanden der Hauptkatalog, der sich aus der 1. Versuchsreihe ergibt, systematisch geübt. Alle Versionen einer Geste werden in diesen Aufnahmen direkt hintereinander ausgeführt. Die Übungsdaten stehen daher für Gesten mit geringer Kontextabhängigkeit, die gleichzeitig mit einer gewissen Sorgfalt und relativ gleichmäßig ausgeführt wurden. Es sind alle 41 Gesten des Hauptkataloges in ungefähr gleicher Anzahl vorhanden.

Der Aufnahmeplatz wurde mit hellen Halogenscheinwerfern bzw. Hochfrequenz-Leuchtstoffröhren so ausgeleuchtet, daß der elektronische Verschluß der Kamera auf 1/125 s eingestellt werden konnte. Dadurch wurde garantiert, daß in den einzelnen Bildern praktisch keine Bewegungsunschärfe vorhanden ist.

Für alle personen*abhängigen* Auswertungen der Übungsdaten wurden die Daten von Versuchsperson 8 (VP 8) verwendet, weil von ihr die längste Aufzeichnung vorliegt. Für die personen*unabhängigen* Evaluierungen wurden noch die Daten von VP 2, 3, 6 und 7 dazu genommen. VP 3 und VP 6 sind Linkshänder, alle weiteren Versuchspersonen sind Rechtshänder. Tabelle C.1 zeigt für jede Versuchsperson und jede Geste die Anzahl (A.), mit der die Geste in der jeweiligen Videoaufnahme auftritt, sowie die minimale ($\downarrow$) und die maximale Länge ($\uparrow$) der Geste.

Für die verschiedenartigen Evaluierungen mit den Dialogdaten müssen folgende Fälle unterschieden werden:

- **Personenabhängigkeit, isolierte Erkennung:** Die minimale Gestenanzahl für die VP 8 beträgt 41. Um alle Gesten genau gleich zu behandeln und eine exakte Aufteilung zu erhalten, wurden 26 Aufnahmen jeder Geste für das Training und 13 Aufnahmen für die Erkennung genommen. Damit wurde jedes Modell beim Training gleich gut trainiert und bei der Erkennung gehen gleichberechtigt alle Gesten ein. Insgesamt wurden also 1066 Gesten für das Training und 533 Gesten für die Erkennung verwendet; beide Datenmengen sind vollständig getrennt.

- **Personenunabhängigkeit, isolierte Erkennung:** Die Anzahl der Gesten ist für verschiedene VPs sehr unterschiedlich verteilt: während bei VP 8 mindestens 41 Aufnahmen pro Geste vorliegen, sind es bei VP 6 nur 6. Um genug Daten für die Evaluierung zu erhalten, mußte auf eine Gleichbehandlung der VPs und der Gesten verzichtet werden. Bei allen VPs wurden daher alle Gesten verwendet und die Gestenversionen wurden wiederum im Verhältnis 2/3 für das Training und 1/3 für die Erkennung aufgeteilt. Reste dieser Aufteilung wurden den Trainingsdaten zugeschlagen. Dadurch ergibt sich eine Anzahl von 2965 Gesten für das Training sowie 1397 Gesten für die Erkennung, wobei sich Trainings- und Testdatenmengen nicht überschneiden.

- **Personenabhängigkeit, kontinuierliche Erkennung:** Beim Training für die kontinuierliche Erkennung wurde das Schema für den personenabhängigen, isolierten Fall beibehalten. Für die kontinuierliche Erkennung war eine Aufteilung der Daten nicht möglich, da alle Versionen einer Geste immer hintereinander aufgenommen wurden. Daher mußte eine Überschneidung von Trainings- und Testdatenmaterial in Kauf genommen werden. Bei insgesamt 1926 Gesten in der vollständigen Videosequenz sind aber immerhin noch etwa 50 % der Gesten unbekannt, was immer noch aussagekräftig genug erscheint.

	VP 2 (29 Min)			VP 3 (32 Min)			VP 6 (15 Min)			VP 7 (25 Min)			VP 8 (48 Min)		
Nr.	A.	↓	↑	A.	↓	↑	A.	↓	↑	A.	↓	↑	A.	↓	↑
1	18	11	24	20	21	35	6	17	37	15	19	42	42	14	26
2	15	39	67	25	24	46	6	36	43	15	25	40	47	16	39
3	17	24	36	28	37	56	6	40	54	15	24	37	43	20	30
4	21	23	36	21	31	46	6	33	47	15	29	38	47	13	28
5	18	11	21	17	21	39	6	16	25	15	14	28	40	10	26
6	15	25	47	18	29	51	7	12	18	15	27	42	52	12	30
7	17	24	37	19	33	77	6	22	39	15	28	45	52	18	31
8	16	26	42	17	25	45	5	16	26	15	30	49	42	18	30
9	27	10	20	24	10	30	7	11	15	16	13	24	50	10	20
10	21	26	55	16	17	41	7	15	25	19	35	60	50	10	42
11	20	28	43	20	22	40	14	10	24	15	36	60	54	16	29
12	21	33	46	18	20	35	7	15	31	17	18	50	46	16	36
13	19	10	21	26	12	26	6	10	16	16	12	25	48	10	31
14	18	27	49	31	15	34	9	13	27	15	23	71	46	13	37
15	20	30	45	16	25	36	10	11	33	15	26	44	46	19	41
16	26	24	40	17	19	42	8	12	26	15	24	39	59	16	37
17	22	12	28	23	16	26	0	—	—	18	10	37	41	12	30
18	16	24	48	18	29	47	6	13	38	17	27	41	44	10	28
19	15	28	40	16	24	32	6	30	41	15	23	39	49	17	33
20	15	27	58	17	21	51	6	26	44	18	21	42	44	17	32
21	18	17	26	20	15	34	6	10	13	15	15	30	44	10	16
22	20	28	56	29	19	35	6	24	38	15	29	53	41	15	32
23	19	21	31	16	28	35	6	25	38	16	35	55	41	18	38
24	17	17	33	17	13	30	6	20	39	22	24	46	44	22	44
25	19	18	49	16	21	39	6	21	33	21	23	42	46	24	41
26	15	19	33	20	17	32	6	19	30	15	23	37	50	27	50
27	17	23	35	16	21	31	6	18	29	17	33	49	50	15	41
28	17	19	33	18	12	18	7	11	25	19	20	33	47	15	36
29	17	18	29	15	13	25	6	11	16	14	18	32	43	12	24
30	16	13	23	17	10	23	9	10	14	18	17	32	49	12	37
31	17	19	31	16	18	36	6	12	26	15	40	68	47	14	24
32	17	10	23	15	12	19	6	12	17	14	18	33	45	15	46
33	15	20	32	15	18	34	6	11	17	13	45	71	47	10	31
34	18	18	34	0	—	—	8	13	25	16	14	40	46	11	29
35	17	44	57	20	52	76	7	18	38	16	32	58	46	20	56
36	15	14	23	16	10	23	6	10	15	15	14	26	43	10	32
37	18	27	41	19	24	75	10	23	46	17	31	46	51	29	57
38	21	22	66	16	43	65	8	12	35	22	37	58	63	23	53
39	17	14	30	16	20	34	6	11	19	15	15	22	48	13	31
40	17	17	35	16	16	26	6	13	20	15	20	35	44	12	25
41	25	34	95	17	36	51	6	38	78	15	41	67	49	32	65

Tabelle C.1: Gestenmaterial in den Übungsdaten (Versuchsperson VP, Anzahl A., minimale Länge ↓, maximale Länge ↑, Längeneinheit Halbbilder)

C.1.2 Dialogdaten

Die Dialogdaten (Darstellung in Tabelle C.2, linke Hälfte) stammen aus dem 2. Teil der 2. Versuchreihe: hier konnten die Probanden frei mit dem 3D-Szenen-Editor interagieren; es wurde ebenfalls der Hauptkatalog verbindlich vorgegeben. Die Abfolge der Gesten wird somit durch die vom Probanden verfolgten Ziele bestimmt, die wiederum indirekt durch die Aufgabenstellung gegeben waren. Die Dialogdaten stehen daher für Gesten mit hoher Kontextabhängigkeit. Darüberhinaus lag die Konzentration mehr auf der Anwendung und der Zielsetzung der Aufgabenstellung als auf der gewissenhaften Ausführung der Gesten. Für die Beleuchtung und den Kameraverschluß gilt dasselbe wie für die Übungsdaten unter Punkt 1.

Obwohl für den Dialog die Gesten des Hauptkataloges vorgegeben waren, verwenden alle Benutzer jeweils nur einen bevorzugten Ausschnitt aus den 41 verschiedenen Gesten. Über alle Versuchspersonen werden jedoch fast alle Gesten verwendet [Vol97]. Die Häufigkeit der vorkommenden Gesten ist bei jeder Versuchsperson darüberhinaus stark ungleichmäßig verteilt.

Die Dialogdaten wurden nur personenabhängig für VP 8 evaluiert, weil für diese VP wiederum die längste Videoaufzeichnung vorlag. Die 18 freien Tabellenzeilen entsprechen den im Videomaterial nicht vorkommenden Gesten. 9 der verbleibenden 23 Gesten kommen in einer deutlich geringeren Anzahl (< 10) als die übrigen Gesten vor. Es erschien nicht sinnvoll, diese Gesten in die Evaluierung mit aufzunehmen, da die Anzahl für ein Training der Modelle nicht ausreicht. Es bleiben also noch 14 Gesten für das Training übrig. Zwei Variationen müssen unterschieden werden; dabei sollen ebenfalls wieder die unter Anh. C.1 genannten Grundsätze eingehalten werden (es handelt sich dabei immer um den personenabhängigen Fall):

- **Isolierte Erkennung:** Analog zum entsprechenden Fall bei den Dialogdaten wurden wenn möglich 26 Versionen einer Geste für das Training verwendet. War die Anzahl kleiner als 39, so wurden zwei Drittel der Versionen zum Trainieren benutzt (Divisionsreste wurden den Trainingsdaten zugeschlagen). Da teilweise nur sehr wenige Daten für die Erkennung übrig waren und schon beim Training ein Ungleichgewicht zwischen den verschiedenen Gesten nicht vermieden werden konnte, wurden *alle restlichen* Daten für die Erkennung verwendet. Damit sind Trainings- und Testdaten auf jeden Fall vollständig getrennt. Es wurden somit 293 Gesten für das Training und 563 für die Erkennung verwendet.

- **Kontinuierliche Erkennung:** Die Konstellation für das Training wurde vom isolierten Fall übernommen. Da die einzelnen Gesten sehr unterschiedlich in der kontinuierlichen Sequenz verteilt sind, war es nicht möglich, einen zusammenhängenden Abschnitt zu finden, in dem alle Gesten einigermaßen gleichmäßig verteilt auftraten. Es wurde daher wie bei den Dialogdaten die *gesamte* Sequenz für die kontinuierliche Erkennung benutzt. Damit sind Teile der Erkennungsdaten schon vom Training her bekannt. Bei insgesamt 898 Gesten (nicht alle gelabelten Gesten wurden auch modelliert) sind allerdings nur etwa ein Drittel der Gesten bekannt.

Da einige Gesten aufgrund ihrer geringen Anzahl nicht modelliert wurden, ergeben sich von vornherein schon 42 vom Labeln her bekannte Gesten, die für das System unbekannte Bewegungen darstellen. Dazu kommt, daß sich gerade in den Dialogdaten sehr viele nicht gezählte, zufällige Bewegungen finden, die den kontinuierlichen Erkennungsvorgang stören. Damit sind die Dialogdaten sehr gut dafür geeignet festzustellen, wie gut eine

Nr.	Dialogdaten					Demonstratordaten				
	A.	$\downarrow$	$\uparrow$	Tr.	Erk.	A.	$\downarrow$	$\uparrow$	$\Downarrow$	$\Uparrow$
1	82	14	29	26	56	45	16	24	67	87
2						45	18	27	70	90
3						48	16	27	72	104
4						45	16	25	65	87
5	100	14	26	26	74	45	16	22	73	91
6						45	16	22	73	94
7						45	16	25	66	90
8						45	16	25	61	77
9	117	14	24	26	91	45	16	21	64	86
10						45	16	23	66	89
11	6	18	20	—	—	45	16	26	58	77
12						45	16	24	58	72
13	54	14	32	26	28	45	16	23	58	86
14						45	16	27	74	93
15						50	16	25	65	83
16						45	16	23	62	74
17	145	14	24	26	119	45	16	24	69	86
18	18	14	27	12	6	45	16	25	71	89
19						45	18	36	72	103
20						45	21	36	68	98
21	106	14	23	26	80	46	16	23	80	97
22						45	17	24	77	103
23	3	21	24	—	—	45	16	25	73	96
24	8	15	23	—	—	45	18	27	75	97
25	45	14	33	26	19	47	17	28	68	94
26	4	19	21	—	—	45	20	30	71	89
27	23	15	26	16	7	45	16	24	65	80
28	3	17	39	—	—	45	16	24	65	94
29	1	17	17	—	—	45	17	26	74	105
30	3	19	21	—	—	45	16	29	61	82
31	5	19	26	—	—	45	16	23	61	90
32						46	16	26	52	77
33						45	16	23	65	91
34	13	14	20	9	4	45	19	30	65	96
35						44	41	51	57	77
36	26	14	28	18	8	46	16	20	64	94
37	17	14	37	12	5	46	42	56	59	81
38	9	22	35	—	—	46	41	55	57	70
39	26	14	24	18	8	46	16	30	54	75
40	84	14	25	26	58	46	16	30	65	93
41						46	27	43	56	81

Tabelle C.2: Gestenmaterial in den Dialog- und Demonstratordaten (Anzahl A., minimale Länge Kerngeste $\downarrow$, maximale Länge Kerngeste $\uparrow$, entsprechend $\Downarrow$ und $\Uparrow$ für Komplettgeste; Tr.: Anzahl Gesten für Training, Erk.: Anzahl Gesten für Erkennung, alle Daten stammen von Versuchsperson 8, Längeneinheit Halbbilder)

kontinuierliche Erkennung unbekannte (und damit ungültige) Bewegungen unterdrücken kann. Auf der anderen Seite ist die Erkennung im Vergleich zu den Dialogdaten etwas einfacher, da nur etwa ein Drittel des Gestenkataloges überhaupt erkannt werden muß. Damit sind die Ergebnisse bei Dialog- und Übungsdaten nicht direkt vergleichbar.

C.1.3 Analysedaten

Die Analysedaten basieren auf dem Nebenkatalog (s. Anh. B.3.2). Die Gesten wurden isoliert aufgenommen. So gut es ging wurde darauf geachtet, daß jede Geste vom selben Punkt aus gestartet wurde. Damit sind die Analysedaten auch für Merkmalsextraktionsverfahren geeignet, die keine invarianten Merkmalsvektoren liefern.

Für alle Gesten wurde eine *konstante* Aufnahmelänge von exakt 70 Halbbildern bei voller Bildwiederholrate eingestellt. Die Daten liegen nur von einer Person vor, so daß auch nur eine personenabhängige Erkennung möglich ist. Bei der Aufnahme wurde eine sehr helle Beleuchtung mit Halogenscheinwerfern verwendet, die es gestattete, die Kameraverschlußzeit auf 1/250 s einzustellen, so daß die Aufnahmen frei von Bewegungsunschärfe sind. Für die Evaluierung müssen zwei Fälle unterschieden werden:

- **Isolierte Erkennung:** Es wurden jeweils pro Geste 20 Aufnahmen für das Training und 10 für die Erkennung verwendet. Das sind insgesamt also 240 bzw. 120 Aufnahmen, die für Training und Erkennung getrennt sind.

- **Kontinuierliche Erkennung:** Hierfür wurden die isoliert aufgenommenen Daten zu kontinuierlichen Sequenzen, den sog. **Synthesedaten**, zusammengesetzt. Dafür wurden die 120 Erkennungsaufnahmen aus dem isolierten Fall verwendet und in zyklischer Abfolge mit sog. Füllsequenzen zu langen, kontinuierlichen Datensätzen verbunden. Die Füllsequenzen eines Testdatensatzes sind alle gleich lang; sie verbinden End- und Anfangspunkte aufeinanderfolgender Gesten durch lineare Interpolation der Merkmalskomponenten.

  Das Training erfolgt, wie im isolierten Fall, mit den übrigen 240 Aufnahmen. Auf diese Weise war es auch für die kontinuierliche Erkennung möglich, für Training und Erkennung völlig getrennte Datenmengen zu verwenden.

C.1.4 Demonstratordaten

Die Demonstratordaten wurden von *einer* Person aufgenommen. Ihnen liegt der komplette Hauptkatalog mit 41 Gesten zugrunde. Sie sind symmetrisch und kontextunabhängig, so daß sie sich insbesondere für die Erkennung mit dem zweistufigen System eignen (genauere Beschreibung s. Kap. 11.1). Aufgrund der kontrollierten Aufnahmebedingungen sind die Anzahlen der Versionen pro Geste sehr gleichmäßig verteilt (s. Tabelle C.2; rechte Hälfte). Es wurde die normale (vorhandene) Raumbeleuchtung verwendet, so daß es erforderlich war, den elektronischen Kameraverschluß auszuschalten. Somit ist auf vielen Einzelbildern mitunter eine starke Bewegungsunschärfe sichtbar.

Die Vorgehensweise für Training und Erkennung ähnelt derjenigen für die Dialogdaten, mit der Ausnahme daß bei den Demosystemdaten nur personenabhängige Erkennung möglich ist:

- **Isolierte Erkennung:** Es wurden exakt 26 Gesten für das Training und 13 für die Erkennung verwendet. Die Datenmengen von 1066 bzw. 533 Gesten sind völlig getrennt.

- **Kontinuierliche Erkennung:** Wiederum war aufgrund des Aufnahmemodus keine Auftrennung der Daten für die Erkennung möglich (alle Versionen einer Geste wurden hintereinander aufgenommen). Es wurden wiederum 26 Gesten für das Training und der komplette Datensatz für die Erkennung verwendet. Bei insgesamt 1862 Gesten ergibt sich wiederum eine Überschneidung von Trainings- und Erkennungsdatensatz von rund 50 %.

C.2 Gestenkontext und Labeln der Daten

Während die Gesten aus dem Analysedatensatz (s. Anh. C.1.3) isoliert aufgenommen wurden, liegt bei den anderen Datensätzen (s. Kap. C.1.1, C.1.2 und C.1.4) für jede Person eine kontinuierliche Aufnahme vor, in der alle Gesten auf einmal enthalten sind und zwar in genau der Reihenfolge, mit der sie auch bei der Aufnahme gezeigt wurden.

Außerdem enthalten die kontinuierlichen Aufnahmen auch viele nicht-gestische Bewegungen und einige Störungen (beispielsweise kann kurzfristig der Kopf der Versuchsperson im Bild sichtbar werden), wie sie sich bei solch langen Aufnahmen (typischerweise 30–45 min) nicht vermeiden lassen. Diese Störungen und bedeutungslosen Bewegungen wurden *nicht* aus den Files beseitigt, um den Realismus der Evaluierungen nicht zu beeinträchtigen.

Damit das Training der Modelle und die isolierte Erkennung durchgeführt werden können, müssen Beginn und Ende der Gesten gekennzeichnet werden. Dieser Vorgang wird *Labeln* genannt. Insbesondere bei den Dialogdaten — aber auch bei allen anderen kontinuierlichen Aufnahmen — ergibt sich nun das Problem, zu erkennen, welcher Teil einer Bewegung als Geste gekennzeichnet werden soll. Es ist offensichtlich, daß Teile einer gestischen Bewegung von der Vorgänger- und der Nachfolgergeste (also dem Gesten*kontext*) beeinflußt werden. Dies ist auch schon in den empirischen Untersuchungen von [Ken86] und anderen Studien (Übersicht in [Wex94]) festgestellt worden, weshalb eine Gestenbewegung allgemein in drei Phasen eingeteilt wird. Der *Gestenkern* wird von einer Vor- und einer Nachbereitungsbewegung begleitet, die für die Verbindung der Gesten untereinander verwendet werden (s. Kap. 2.2).

Als Gestenkern läßt sich nun in den Aufnahmen der Dialogdaten jener Teil einer gestischen Bewegung definieren, der unabhängig vom Kontext der Geste bleibt. Diese Unabhängigkeit ist natürlich nicht vollständig gegeben, so daß sich oft nur grobe Grenzen des Gestenkerns angeben lassen. Weiterhin läßt diese Definition auch immer noch einen gewissen Spielraum für das Labeln. Man darf jedoch nicht übersehen, daß der Begriff der Kerngeste nur ein Hilfsbegriff ist, um eine bestimmte Art einer komplexen menschlichen Bewegung zu beschreiben. Wie Experimente von [Ken86] gezeigt haben (s. Kap. 2.2), ist der Mensch sehr wohl in der Lage, eine Geste — auch eine ihm unbekannte Geste — als solche zu identifizieren und gewisse Kriterien für die Gliederung einer gestischen Bewegung anzugeben. Es hat sich gezeigt, daß dort, wo das Kriterium der Kontextunabhängigkeit die Kerngeste nur grob eingrenzen konnte, immer Kriterien gefunden werden konnten, mit der sich definierte und *reproduzierbare* Anfangs- und Endpunkte einer Kernbewegung angeben ließen. Solche Kriterien sind beispielsweise der „Öffnungsgrad“ einer Faust, die „Ausgestrecktheit“ der Finger, das Sichtbarwerden des Handrückens oder das vorübergehende Pausieren der globalen Handbewegung am Gestenende. Die durch diese reproduzierbar angebbaren Grenzen gefundenen Bewegungen werden in dieser Arbeit ebenfalls als Kerngesten definiert.

Die auf diese Weise gefundenen Kerngesten zerfallen in sogenannte *symmetrische* und *asymmetrische* Gesten. Dieses Symmetriekriterium bezieht sich dabei nur auf den groben Ablauf der Kernbewegung und nicht auf eine völlige Spiegelsymmetrie der Bildsequenz. Am häufigsten treten die *asymmetrischen* Gesten auf, die dadurch gekennzeichnet sind, daß Beginn und Ende der Kernbewegung räumlich voneinander getrennt sind. Die asymmetrischen Gesten sind zeitlich nur kurz und weisen keine Bewegungspausen auf. Bei den relativ seltenen *symmetrischen* Gesten endet die Kernbewegung dort, wo sie angefangen hat. Dadurch weisen alle symmetrischen Kerngesten eine kurze Pause im Umkehrpunkt der Bewegung auf und der Bewegungskern der Geste dauert im Mittel etwas länger. In Tabelle B.6 im Anh. B.3.1 ist für jede Geste des Hauptkataloges angegeben, ob sie symmetrisch ist oder nicht. Auch die Bilder in Tabelle B.5 lassen diese Symmetrieeigenschaft erkennen.

Auf der einen Seite gibt es nun Gesten, die sich nur aufgrund ihrer Symmetrieeigenschaft unterscheiden (beispielsweise Geste 39 und 41: „zeigen“ und „Taste drücken“). Auf der anderen Seite gibt es sehr viele Paare asymmetrischer Gesten, die sich zu potentiellen symmetrischen Gesten komplettieren (sog. *Komplementärgesten* wie beispielsweise Geste 1 und 5: „winken nach rechts“ und „winken nach links“). Dadurch, daß nur die asymmetrischen Kerngesten einer solche Geste gelabelt wurden, besteht die Gefahr, daß — je nach Kontext — die Nachbereitungsbewegung einer solchen Geste als Komplementärgeste erkannt wird. Es hat sich jedoch gezeigt, daß diese Verwechslungsgefahr gering ist, da sich eine Nachbereitungsbewegung und eine eventuell passende Komplementärgeste durch die Bewegungsgeschwindigkeit und die Anspannung der Hand unterscheiden: die Kernbewegung wird immer mit Nachdruck und damit einer relativ hohen Geschwindigkeit ausgeführt und die Hand ist sehr angespannt, was eine sehr geordnete Fingerstellung bewirkt.

Die Auswirkung des Kontextes auf die *Erkennung* wird erst im Zusammenhang mit der kontinuierlichen Erkennung in Kap. 10 untersucht. Bei der isolierten Erkennung dagegen hat der Kontext keine Bedeutung, da nur die Kerngesten untereinander verglichen werden.

Anhang D

Weitere Ergebnisse zur Erkennung verbundener Gesten

D.1 Anwendung der Peak-Verstärkung

Obwohl die in Tabelle 10.20 (s. Kap. 10.7.2.1) gezeigte mittlere Erkennungsverzögerung für Dialoganwendungen unkritisch ist, kann sie bei Bedarf über die Anwendung der Peak-Verstärkung nach Gl. (9.24) (s. Kap. 9.3.4.2) verkürzt werden (s. Tabelle D.1). Während sich bei den Übungsdaten durch vorsichtiges Erhöhen des Mischungsverhältnisses C_{mix} außer einer leichten Vergrößerung der Falschakzeptanzrate f kaum etwas verändert, läßt sich die mittlere Erkennungsverzögerung bei den Dialogdaten auf das Niveau der Übungsdaten auf etwas über 0,5 Sekunden reduzieren. Obwohl dabei sogar die Erkennungsrate noch um 2 % zunimmt, sollte der Einsatz der Peak-Verstärkung wohl überlegt werden, da sich damit bei den Dialogdaten die Falschakzeptanzrate f leicht verdoppeln kann. Diese Tendenz ergibt sich auch beim optimalen Score-Eingangsgewicht $W = 30$; sie setzt sich außerdem in den ROCs über den gesamten r-f-Bereich fort (nicht dargestellt).

D.2 Anwendung der Verweildauer-Modellierung

Bei der Anwendung der Verweildauer-Modellierung nach Gl. (9.22) in Kap. 9.3.3 sind die Übungsdaten bevorzugt (s. Tabelle D.2): hier ergibt sich mit steigendem Parameter v eine starke Verringerung der Falschakzeptanzrate bei nur geringen Verlusten in der Erkennungsrate. Leider zeigt sich bei den Dialogdaten der entgegengesetzte Effekt: hier sinkt die Erkennungsleistung. Das Verhalten bestätigt den unterschiedlichen Charakter der Datensätze: die Erkennung der relativ gleichmäßig ausgeführten Gesten in den Übungsdaten wird durch die „Bestrafung“ von Längenausreißern verbessert, während die ungleichmäßigeren Gesten in den Dialogdaten dadurch nicht mehr flexibel genug modelliert werden können.

Für den guten Kompromißwert von $v = 3$ kann man an den ROCs in Bild D.1 erkennen, daß der Verlust an Erkennungsrate bei den Dialogdaten noch durch die Anhebung des Score-Eingangsgewichtes auf $W = 30$ mehr als ausgeglichen werden kann. Wie vorher schon gezeigt wurde, entsteht durch die W-Anhebung in den oberen r-Bereichen der Übungsdaten kein nennenswerter Effekt: hier kann dann der v-Parameter seine Wirkung entfalten. Setzt man also die Verweildauer-Modellierung ein, so werden Daten mit der

	Übungsdaten					Dialogdaten				
C_{mix}	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
0	98,81	16,90	0,68	35,65	36,65	91,56	35,13	7,03	53,58	54,58
0,5	98,81	17,09	0,68	34,86	35,86	91,79	43,99	17,00	48,24	49,24
1,0	98,81	17,91	1,04	34,06	35,06	92,15	48,97	29,78	41,40	42,40
1,5	98,86	18,87	1,30	33,38	34,38	92,73	54,64	35,52	37,34	38,34
2,0	98,81	20,26	1,35	33,03	34,03	92,61	59,62	37,87	35,07	36,07
2,5	98,86	20,71	1,14	32,81	33,81	92,97	63,22	38,45	34,10	35,10
3,0	98,81	21,28	1,04	32,57	33,57	93,32	66,81	38,80	33,57	34,57
3,5	98,81	22,07	0,88	32,44	33,44	93,08	70,41	37,05	33,15	34,15
4,0	98,75	22,61	1,09	32,25	33,25	93,32	76,36	37,05	32,75	33,75
4,5	98,70	22,96	1,09	32,13	33,13	93,20	78,29	37,16	32,45	33,45
5,0	98,70	23,46	1,09	31,98	32,98	93,43	79,68	37,40	32,09	33,09
5,5	98,75	24,54	1,19	31,86	32,86	93,20	83,41	36,58	31,91	32,91
6,0	98,70	25,56	1,30	31,72	32,72	92,97	87,98	37,51	31,55	32,55
9,0	98,34	31,36	2,34	30,92	31,92	92,61	113,43	38,57	29,79	30,79
12,0	98,03	39,70	3,06	30,26	31,26	91,09	137,91	38,10	28,19	29,19

Tabelle D.1: Erkennungsergebnisse des Spotting-Verfahrens mit Übungs- und Dialogdaten bei Variation des Mischungsfaktors C_{mix} (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Überhang $o = 20$, Zustandszahl $N = 30$, Codebuchgröße $L = 1024$, Score-Eingangsgewicht $W = 0$)

	Übungsdaten					Dialogdaten				
v	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$	r	f	r_{M}	$\bar{\tau}^{\text{v}}$	$\bar{\tau}^{\text{vo}}$
1,0	98,81	16,90	0,68	35,65	36,65	91,56	35,13	7,03	53,58	54,58
1,5	98,81	15,22	0,52	35,83	36,83	90,86	33,61	6,57	53,54	54,54
2,0	98,75	14,33	0,52	36,01	37,01	90,04	32,09	6,33	53,24	54,24
2,5	98,34	12,52	0,52	36,03	37,03	89,57	33,06	6,21	53,41	54,41
3,0	97,92	11,00	0,57	36,04	37,04	88,51	33,75	5,98	53,44	54,44
3,5	97,51	10,40	0,57	36,11	37,11	86,99	35,69	5,74	53,21	54,21
4,0	97,09	9,67	0,62	36,21	37,21	86,05	36,24	5,86	53,22	54,22
5,0	96,21	8,81	0,52	36,36	37,36	84,06	38,45	5,74	52,97	53,97
6,0	95,48	7,58	0,52	36,51	37,51	81,95	39,28	5,28	52,85	53,85
7,0	94,70	7,07	0,47	36,67	37,67	79,84	40,53	5,51	52,83	53,83
8,0	93,77	6,56	0,36	36,79	37,79	77,96	41,50	5,39	52,84	53,84
9,0	92,73	6,15	0,47	36,93	37,93	76,44	43,99	5,16	53,07	54,07
10,0	91,64	5,90	0,47	37,06	38,06	74,91	45,23	4,92	53,32	54,32
11,0	90,97	5,90	0,42	37,14	38,14	72,92	45,65	4,57	52,83	53,83
12,0	90,29	5,74	0,52	37,20	38,20	71,28	46,34	4,57	53,09	54,09

Tabelle D.2: Erkennungsergebnisse des Spotting-Verfahrens mit Übungs- und Dialogdaten bei Variation des Normierungslängen-Parameters v (r und r_{M} in %, f in 1/h, τ-Werte in Merkmalszeittakten, Überhang $o = 20$, Zustandszahl $N = 30$, Codebuchgröße $L = 1024$, Score-Eingangsgewicht $W = 0$)

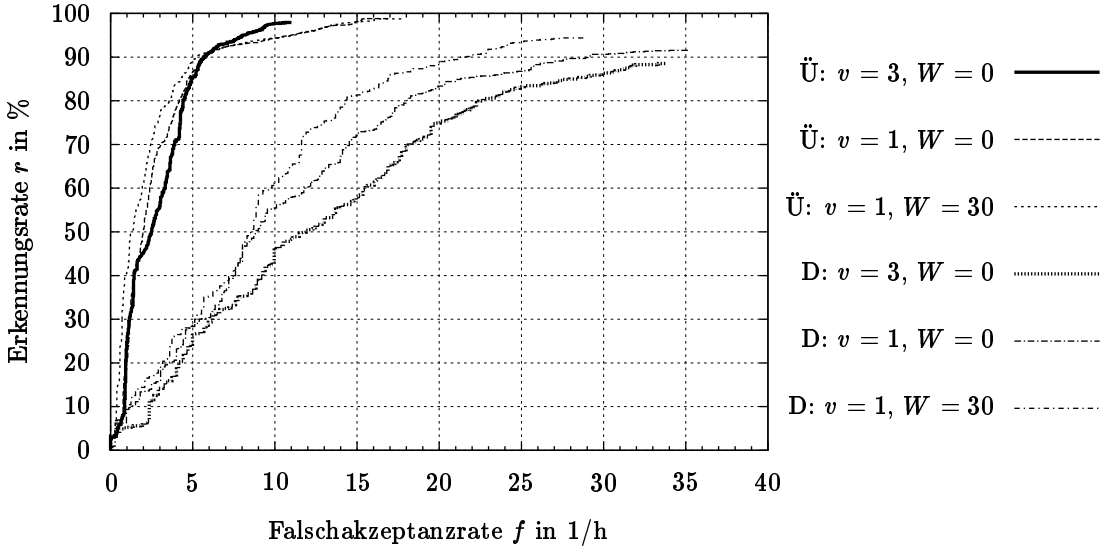


Bild D.1: ROC-Darstellungen ausgewählter Erkennungsergebnisse aus Tabelle D.2 und Tabelle 10.22 Seite 140 erzeugt durch Variation der Rückweisungsschwelle S_{rel} zur Verdeutlichung der Wirkung des Normierungslängen-Parameters v im Vergleich zur Wirkung des Score-eingangsgewichtes W (Ü: Übungsdaten, D: Dialogdaten, $o = 20$, $N = 30$, $L = 1024$)

Charakteristik der Übungsdaten besser erkannt, während Daten, die den Dialogdaten ähnlich sind, etwa so gut wie ohne Einsatz eines Score-Eingangsgewichtes verarbeitet werden.

D.3 Optimierung des minimalen Peak-Abstandes

Neben der Rückweisungsschwelle S_{rel} kann auch der minimale Peak-Abstand τ_{dist} (Regel P4 in Kap. 9.3.5) eingesetzt werden, um die Falschakzeptanzrate f auf Kosten der Erkennungsrate r zu verringern. Die damit erzeugbaren r-f-Kennlinien sollen Pseudo-ROCs genannt werden, weil der Begriff ROC für den „klassischen“ Einsatz einer (absoluten) Score-Rückweisungsschwelle reserviert ist. In Bild D.2 sind die Pseudo-ROCs zusammen mit den echten ROCs dargestellt. Man erkennt, daß die Pseudo-ROC- wesentlich unruhiger verlaufen als die ROC-Diagramme. Insbesondere gibt es in den Kennlinien nun rückläufige Bereiche. Die Ursache liegt darin, daß der minimale Peak-Abstand eine verkettete Wirkung haben kann: fällt durch Vergrößerung des Peak-Abstandes τ_{dist} ein Peak weg, so können darauf folgende, bisher „verdeckte“ Peaks freigelegt werden — und umgekehrt. Von diesem Vorgang sind sowohl gültige als auch ungültige Peaks betroffen, so daß r und f beeinflußt werden.

In Bild D.2 ist jedoch auch erkennbar, daß die Pseudo-ROC- die ROC-Kennlinie sowohl für die Übungs- als auch die Dialogdaten jeweils in einem kurzen Abschnitt übersteigt. Für beide Datensätze liegen diese Abschnitte im Bereich von $\tau_{\text{dist}} = 42$.

Berechnet man für einen festen minimalen Peak-Abstand von $\tau_{\text{dist}} = 42$ die ROC, so stellt man sowohl für die Übungs- als auch für die Dialogdaten in einem weiten r-f-Bereich einen Anstieg der Erkennungsleistung fest (s. Bild D.3). Es ist erkennbar, daß die Regel des minimalen Peakabstandes dann gewinnbringend eingesetzt werden kann, wenn nicht die maximal mögliche Erkennungsrate benötigt wird.

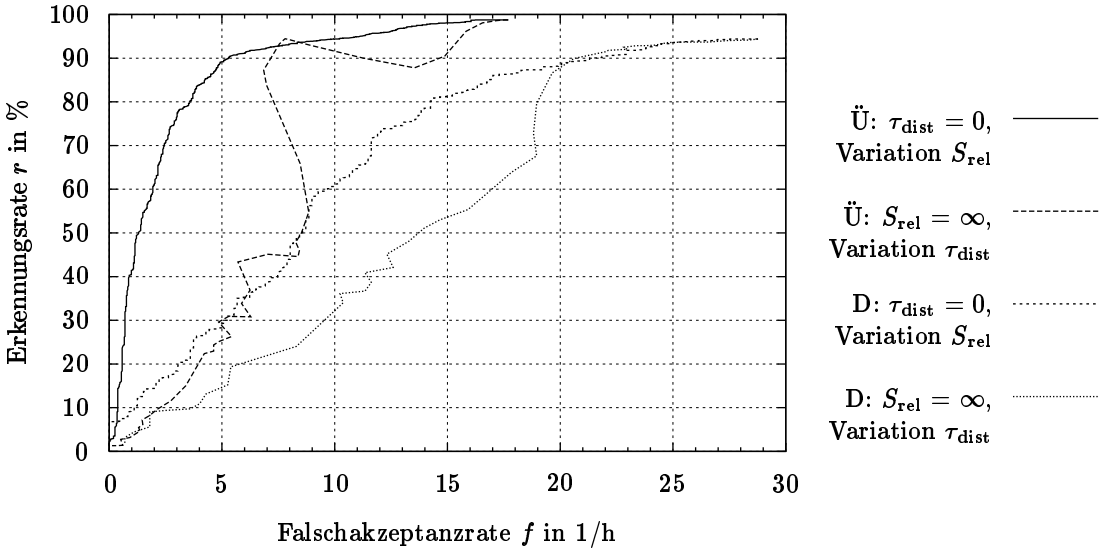


Bild D.2: Pseudo-ROC-Diagramme für Übungs- und Dialogdatensatz erzeugt durch Variation des minimalen Peak-Abstandes τ_{dist} mit Score-Eingangsgewicht $W = 30$ und Vergleich mit durch Variation der Rückweisungsschwelle S_{rel} erzeugten ROC-Kennlinien mit Score-Eingangsgewicht $W = 30$ (Ü: Übungsdaten, D: Dialogdaten, Überhang $o = 20$, Zustandszahl $N = 30$, Codebuchgröße $L = 1024$)

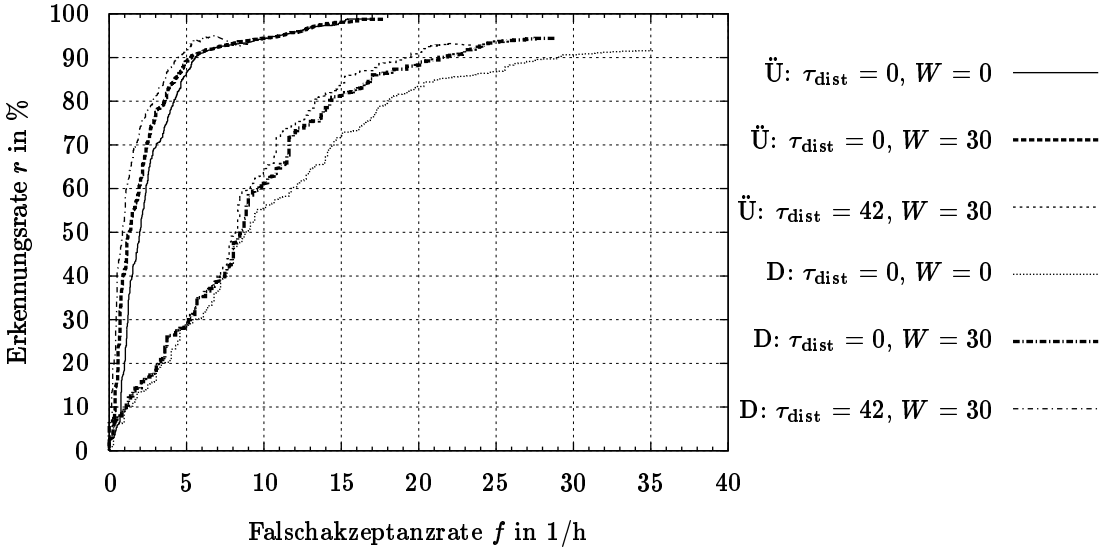


Bild D.3: ROC-Diagramme für Übungs- und Dialogdatensatz erzeugt durch Variation der Rückweisungsschwelle S_{rel} für minimale Peak-Abstände $\tau_{\text{dist}} = 0$ und 42 (Ü: Übungsdaten, D: Dialogdaten, Überhang $o = 20$, Zustandszahl $N = 30$, Codebuchgröße $L = 1024$, Score-Eingangsgewicht $W = 0$ und $W = 30$)

Abkürzungsverzeichnis

Weitere Abkürzungen gelten im Zusammenhang mit den Ergebnistabellen der isolierten Erkennung und sind in den Tabellen 8.1 auf Seite 82 und 8.6 auf Seite 88 aufgeführt.

3D	dreidimensional
B1–B4	Evaluierungskriterien für die kontinuierliche Detektion (s. Kap. 10.5)
BDF	Bilddichtefunktion
BS	Bildstreifen
BV	Bildvektor
BWA	Baum-Welch-Algorithmus
CAD	*computer aided design*
CPU	*central processing unit*
D1–D4	Regeln für Bewegungsdetektion (s. Kap. 9.2.1.2)
DGE	direkte graphische Extraktion
E1–E4	Evaluierungskriterien für die kontinuierliche Erkennung (s. Kap. 10.5)
G1–G3	Grundlagen des gestischen Dialogs (s. Kap. 3.3)
HG	Hintergrund
HMI	Hu-Moment-Invariante
HMM	Hidden-Markov-Modell
K1–K3	Kennzeichen der indirekten Manipulation (s. Kap. 3.4)
LUT	*look up table*
MEV	Merkmalsextraktionsverfahren
MI	Moment-Invariante
ML	*maximum likelihood*
N1–N2	Normierungsverfahren (s. Kap. 9.3.1)
P1–P4	Regeln für die Peaksuche (s. Kap. 9.3.5)
ROC	*receiver operating characteristic*
S1–S5	Evaluierungskriterien für die räumliche Segmentierung (s. Kap. 5.1.2)
T1–T3	Triggerverfahren (s. Kap. 9.3.2)
VA	Viterbi-Algorithmus
VG	Vordergrund
VP	Versuchsperson
VR	Versuchsreihe
WDF	Wahrscheinlichkeitsdichtefunktion
ZMI	Zernike-Moment-Invariante

Verzeichnis der Formelzeichen

Durch tief- und hochgestellte Indizes werden Größen näher spezifiziert. Um eine übersichtlichere Schreibweise zu erhalten, werden solche Indizes unter Umständen weggelassen, wenn sie zur Darstellung eines bestimmten Sachverhaltes nicht benötigt werden und Mißverständnisse ausgeschlossen sind: ein Merkmalsvektor $\mathbf{x}$ ist beispielsweise immer zeitabhängig und wird daher vollständig als $\mathbf{x}_t$ geschrieben. Manchmal wird ein Index auch weggelassen, um eine einheitliche Schreibweise zu ermöglichen, die mehrere Alternativen beinhaltet: so kann die Bildfunktion $\mathbf{f}_t$ sowohl im RGB-Farbraum ($\mathbf{f}_{RGB,t}$) als auch im YUV-Farbraum ($\mathbf{f}_{YUV,t}$) definiert sein. Die gültigen Alternativen gehen aus dem Zusammenhang hervor.

Der Zeitindex t bezeichnet den Bildzeittakt, t' den Merkmalszeittakt. Zur Vereinfachung der Schreibweise und da Verwechslungen durch den jeweiligen Kontext ausgeschlossen sind, wird in der Regel auch für den Merkmalszeittakt t verwendet (s. Kap. 7.1).

Bildfunktionen und -sequenzen

$\mathbf{f}_{RGB,t}(\mathbf{n})$, $\mathbf{f}_{YUV,t}(\mathbf{n})$	diskrete Bildfunktion im RGB-, YUV-Farbraum
$\mathbf{f}_{\mathbf{x},t}(\mathbf{n})$, $f_{x,t}(\mathbf{n})$	diskrete Bildfunktion im Farbraum oder Unterfarbraum $\mathbf{x}$, Komponente x einer diskreten Bildfunktion
$f_{\mathrm{b},t}(\mathbf{n})$, $\mathbf{f}_{\mathrm{s},t}(\mathbf{n})$	Segmentierungsmaske, segmentierte Bildfunktion
$\mathbf{f}_{\mathrm{g},t}(\mathbf{n}) = [f_{\mathrm{g},t}(\mathbf{n}), \delta_{\mathrm{g},t}(\mathbf{n})]^{\mathrm{T}}$	Gradientenbild bestehend aus Kanten- und Orientierungsbild
$f(x, y)$	Komponente einer kontinuierlichen Bildfunktion
$\mathbf{n} = (n_1, n_2)$	diskrete Ortskoordinaten einer Bildfunktion
R, G, B	Farbkomponenten Rot, Grün und Blau des RGB-Farbraumes
t	Bildzeittakt
(x, y)	kontinuierliche Ortskoordinaten
Y, U, V	Luminanzkomponente Y und Chrominanzkomponenten U und V des YUV-Farbraums

Vorverarbeitung inkl. räumlicher Segmentierung

e_{S}, $e_{\mathrm{S}}^{\mathrm{Hg}}$	Gesamt-, Hintergrund-Segmentierungsfehler
$g_{\mathrm{S}}^{\mathrm{Vg}}$	Vordergrund-Segmentierungsgrad
$L_{\mathbf{x}}^{\mathrm{Hg}}(\mathbf{x})$, $L_{\mathbf{x}}^{\mathrm{Vg}}(\mathbf{x})$	mehrdimensionale LUT des Hinter- bzw. des Vordergrundes der Bildkomponente $\mathbf{x}$
$N^{\mathrm{Hg}}(\mathbf{x})$	mehrdimensionales Verbund-Hintergrund-Histogramm mit absoluter Häufigkeit der Bildkomponente $\mathbf{x}$
r^{Hg}, r^{Hg}	Aufweitungsradius für Hintergrund-, Vordergrund-LUT
$\mathcal{V}$	Vordergrund-Bildbereich

Modellierung mit Hidden-Markov-Modellen

$a_{S_i S_j}^{\lambda_l}$, $A_{S_i S_j}^{\lambda_l}$, $\mathbf{A}^{\lambda_l}$	Übergangswahrscheinlichkeit von Zustand S_i nach Zustand S_j, logarithmierte Form, zusammenfassende Matrix
$c_{S_i v_k}^{\lambda_l}$, $\mathbf{C}^{\lambda_l}$	Mixturkoeffizient für den v_k-ten Prototyp des Zustandes S_i im Modell λ_l, zusammenfassende Matrix
D	Dimension der Merkmalsvektoren
$D_{S_j,t}^{\lambda_l}$	lokaler Score im Zustand S_j zum Zeitpunkt t im Modell λ_l
$f_{v_k}(\mathbf{x})$	v_k-ter Prototyp des Codebuchs
$f(v_k\|\mathbf{x})$	Rückschlußwahrscheinlichkeit zum v_k-ten Prototypen
$f_{S_i}^{\lambda_l}(\mathbf{x})$, $F_{S_j}^{\lambda_l}(\mathbf{x})$	Wahrscheinlichkeitsdichte im Zustand S_i des Modells λ_l, logarithmierte Form
$F(\mathbf{X}\|\lambda_l)$	Erzeugungswahrscheinlichkeitsdichte für eine Merkmalssequenz $\mathbf{X}$ bezogen auf ein Modell λ_l
λ_l	Modell mit Index l
$\lambda_{\mathbf{X}}^{\mathrm{Er}}$	Modellindex als Ergebnis der Klassifikation der Erkennungs-Merkmalssequenz $\mathbf{X}$
L	Anzahl der Prototypen im Codebuch
M	Anzahl der Modelle (Anzahl der Gesten im Gestenkatalog)
$\boldsymbol{\mu}_{v_k}$, μ_{i,v_k}, $\mathbf{M}$	Mittelwertsvektor des v_k-ten Prototypen, Komponente, Matrix der Mittelwertsvektoren
$\boldsymbol{\mu}^{\mathrm{Tr}}$	Mittelwertsvektor für die HMM-Vorverarbeitung
N	Anzahl der Zustände
$\pi_{S_i}^{\lambda_l}$, $\Pi_{S_j}^{\lambda_l}$, $\boldsymbol{\pi}^{\lambda_l}$	Einsprungwahrscheinlichkeit in Zustand S_i des Modells λ_l, logarithmierter Wert, Vektor der Einsprungwahrscheinlichkeiten
S_i	i-ter Zustand eines Modells
$S = s_1, s_2, \ldots, s_T$	zeitliche Zustandsfolge
$\boldsymbol{\Sigma}_{v_k}$, σ_{ii,v_k}, $\boldsymbol{\sigma}_{v_k}$, $\boldsymbol{\Sigma}$	Kovarianzmatrix, Komponente, Kovarianzvektor des v_k-ten Prototypen, Matrix der Kovarianzvektoren
$\boldsymbol{\sigma}^{\mathrm{Tr}}$	Varianzvektor für die HMM-Vorverarbeitung
T	Länge einer Merkmalssequenz
$\bar{\tau}_{S_i}^{\lambda_l}$, $\bar{T}^{\lambda_l}$	mittlere Verweildauer im Zustand S_i eines Modells λ_l, im gesamten Modell
$\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_T$	Sequenz von Merkmalsvektoren der Länge T,
x_i	i-te Komponente des Merkmalsvektors $\mathbf{x}$
$\mathbf{x}_t$, $\mathbf{x}_t'$	Merkmalsvektor nach, vor der HMM-Vorverarbeitung

Merkmalsextraktion

A_t	Flächenäquivalent zum Zeitpunkt t
d_{ij}	mittlere Orientierung in N_{ij}
$H_{i,t}$	i-te Hu-Moment-Invariante zum Zeitpunkt t
$\mathcal{K}$	Kontur des Vordergrundbereichs

K, L	Anzahl der Rasterpunkte für Bildstreifen- und Bildvektorverfahren
m_{pq}, μ_{pq}, μ_{pq}^{N}	allgemeines Moment, Zentralmoment, normiertes Zentralmoment der Ordnung $p+q$
$M_{i,t}$	i-te Moment-Invariante zum Zeitpunkt t
$(\bar{n}_{1,t}, \bar{n}_{2,t})$	Bildschwerpunkt zum Zeitpunkt t
N_O	Anzahl von Moment-Invarianten bis zur Ordnung O
N_1, N_2	maximale Ortskoordinaten einer diskreten Bildfunktion
N_{ij}	Rastergebiete bzw. Nachbarschaften für Rastermittelpunkte bzw. Bildvektoren $\mathbf{v}_{ij}$
n_{ij}	mittlerer Grauwert in N_{ij}
O_t	Anzahl der Merkmalsvektoren pro Bild
p, q	Potenzen bei der Momentenberechnung
Φ	Bild-Hauptorientierung
$S_{i,t}$	i-te Zernike-Moment-Invariante zum Zeitpunkt t
t', t	Merkmalszeittakt
$(\bar{x}, \bar{y})$	Bildschwerpunkt in kontinuierlichen Koordinaten
$\mathbf{x}_i^{\mathrm{V}}$, $\mathbf{x}_i^{\mathrm{H}}$	i-ter vertikaler, horizontaler Bildstreifen

Evaluierung der isolierten Erkennung

f_{skal}	Skalierungsfaktor zur Bestimmung der relativen Bildgröße
f_{R}	Bildwiederholrate
g_l	Gestenindex
$\lambda_v^{\mathrm{Er},g_l}$	Klassifikationsergebnis für Datensatz $\mathbf{X}_v^{\mathrm{Er},g_l}$
r, r_{K}	Erkennungsrate, Kategorieerkennungsrate
s	Erkennungssicherheit
$\mathbf{X}^{\mathrm{Tr},\lambda_l}$	Trainingsdatensatz zum Trainieren eines Modells λ_l
$\mathbf{X}_v^{\mathrm{Er},g_l}$	v-te Version des Erkennungsdatensatzes von Geste g_l

Kontinuierliche Erkennung

B_i	i-tes Bewegungsintervall nach Bewegungsdetektion
C_{mix}	Mischungsfaktor für Peak-Verstärkung
$D_{\mathrm{tr}}^{\lambda_l}$	Trigger-Scoreschwelle
$\bar{D}_{\mathrm{max}}^{\lambda_{P_i}}$, $\bar{D}_{\mathrm{min}}^{\lambda_{P_i}}$	minimaler, maximaler geglätteter Score des Modells des Peaks P_i
$F_{S_j,t}^{\lambda_l}$	vereinfachte Schreibweise zu $F_{S_j}^{\lambda_l}(\mathbf{x}_t)$
L_{n}, L_{s}	konstante Normalisierungslänge, Glättungslänge für Triggerschwelle
$L_{S_j,t}^{\lambda_l}$	lokale Pfadlänge im Zustand S_j zum Zeitpunkt t im Modell λ_l
λ_{B_i}	Gestenindex zum i-ten Bewegungsintervall
λ_{P_i}	Gestenindex zum i-ten Peak

$m_{\mathrm{b},t}$, $m_{\mathbf{x},t}$, $m_{H_O,t}$	pixelbasierter Bewegungswert auf Segmentierungsmaske, im Farbraum $\mathbf{x}$, merkmalsbasierter Bewegungswert mit HMIs bis zur Ordnung O zum Zeitpunkt t
m_{S}	Bewegungswertschwelle
P_i	i-ter Peak nach Peak-Detektion
S_{rel}	relative Rückweisungsschwelle
t_{P_i}	Zeitpunkt des i-ten Peaks
$\tau_{\mathrm{s}}^{\mathrm{b}}$, $\tau_{\mathrm{s}}^{\mathrm{e}}$	Kennwert für linke, rechte Grenze des Glättungsintervalles
$\tau_{\mathrm{p}}^{\mathrm{b}}$, $\tau_{\mathrm{p}}^{\mathrm{e}}$	Kennwert für linke, rechte Grenze des Peak-Suchintervalles
τ_{dist}	minimaler zeitlicher Peak-Abstand
$t_{B_i}^{\mathrm{b}}$, $t_{B_i}^{\mathrm{e}}$	Anfangs-, Ende-Zeitpunkt des i-ten Bewegungsintervalles
T_{B_i}	Länge des Bewegungsintervalles B_i
$\tau_{\mathrm{min}}^{\mathrm{b}}$, $\tau_{\mathrm{min}}^{\mathrm{e}}$, $\tau_{\mathrm{min}}^{\mathrm{l}}$, $\tau_{\mathrm{min}}^{\mathrm{d}}$	Zeitkonstanten für die Bewegungsdetektion
t_{tr}	Triggerzeitpunkt
W	Score-Eingangsgewicht beim Triggern
$\mathbf{X}_{B_i}$	Sequenz von Merkmalsvektoren im Bewegungsintervall B_i

Evaluierung der kontinuierlichen Erkennung

E	Anzahl der detektierten Peaks oder Bewegungsintervalle
f_{A}, f_{d}	Falschakzeptanzrate, Falschdetektionsrate
g_{L_j}	Gestenindex des Labels L_j
G	Anzahl der Label eines Datensatzes
g	Optimierungsparameter für die Bewegungsdetektion
K	Anzahl der korrekt erkannten Label
L_i, L_k^*	i-tes Labelintervall, k-tes Label, für das ein korrekter erkanntes Bewegungsintervall oder ein korrekter Peak existiert
L_{f}	Füllsequenzlänge
o, o^{b}, o^{e}	Überhang, Beginn-, Ende-Überhang
P_k^*	korrekter Peak mit dem geringsten Abstand zu L_k^*
r_{M}, $r_{\mathrm{d,m}}$, $r_{\mathrm{d}}^{\mathrm{eff}}$	Mehrfacherkennungsrate, Mehrfachdetektionsrate, effektive Detektionsrate
$t_{L_i}^{\mathrm{b}}$, $t_{L_i}^{\mathrm{e}}$	untere, obere Grenze des Labels L_i
$\tau_{\mathrm{max}}^{\mathrm{v}}$, $\tau_{\mathrm{d,max}}^{\mathrm{v}}$	maximale Erkennungs-, maximale Detektionsverzögerung
$\bar{\tau}^{\mathrm{v}}$, $\bar{\tau}^{\mathrm{vo}}$, $\bar{\tau}^{\mathrm{va}}$, $\bar{\tau}^{\mathrm{ve}}$	mittlere Erkennungs-, Online-Erkennungs-, Anfangs-Detektions-, Ende-Detektionsverzögerung

Literaturverzeichnis

[Ass98] M. Assan, K. Grobel: *Video-Based Sign Language Recognition Using Hidden Markov Models.* Gesture and Sign Language in Human-Computer Interaction: Proc. of the Int. Gesture Workshop 1997, Bielefeld, Germany; Springer, Berlin, pp. 97–109, 1998.

[Bal82] D. H. Ballard, C. M. Brown: *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1982.

[Bau67] L. E. Baum, J. A. Egon: *An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology.* Bulletin of the American Meteorological Society, vol. 73, pp. 360–363, 1967.

[Bau70] L. E. Baum, T. Petrie, G. Soules, N. Weiss: *A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains.* Annals of Mathematical Statistics, vol. 41, no. 1, pp. 164–171, 1970.

[Bel91] S. O. Belkasim, M. Shridhar, M. Ahmadi: *Pattern Recognition with Moment Invariants: A Comparative Study and New Results.* Pattern Recognition, vol. 24, no. 12, pp. 1117–1138, 1991.

[Ben98] S. Ben Yedder: *Labelung, zeitliche Segmentierung und Klassifikation zusammenhängender Gesten.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1998.

[Bol80] R. A. Bolt: *"Put-That-There": Voice and Gesture at the Graphics Interface.* Computer Graphics, vol. 14, no. 3 (SIGGRAPH-80 proceedings), pp. 262–270, 1980.

[Bol92] R. A. Bolt, E. Herranz: *Two-Handed Gesture in Multi-Modal Natural Dialog.* ACM Proc. of the 11th Annual Symposium on User Interface Software and Technology (UIST-92), Monterey, USA, pp. 7–14, 1992.

[Bow97] R. Bowden, T. Heap, D. Hogg: *Real Time Hand Tracking and Gesture Recognition as a 3D Input Device for Graphical Applications.* Progress in Gestural Interaction: Proc. of the Int. Gesture Workshop 1996, York, Great Britain; Springer, London, Great Britain, pp. 117–129, 1997.

[Brö95] U. Bröckl-Fox: *Real-Time 3-D Interaction with up to 16 Degrees of Freedom from Monocular Video Image Flows.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 172–178, 1995.

[Bro81] I. N. Bronstein, K. A. Semendjajew: *Taschenbuch der Mathematik.* 21. Auflage, Teubner, Leipzig, 1981.

[Bru95] B. Brus: *Bildverarbeitungsgestützte Analyse statischer Handgesten.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1995.

[Bun85] H. Bunke: *Modellgesteuerte Bildanalyse: Dargestellt anhand eines Systems zur automatischen Auswertung von Sequenzszintigrammen des menschlichen Herzens.* Teubner, Stuttgart, 1985.

[Cam95] L. Campbell, A. Bobick: *Using Phase Space Constraints to Represent Human Body Motion.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 338–343, 1995.

[Cam96] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, A. Pentland: *Invariant Features for 3-D Gesture Recognition.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 157–162, 1996.

[Chi96] G. I. Chiou and J.-N. Hwang: *Lipreading from Color Motion Video.* IEEE Proc. of the 1996 Int. Conference on Acoustics, Speech, and Signal Processing, Atlanta, USA, pp. 2156–2159, 1996.

[Cho93] G. Chow and X. Li: *Toward a System for Automatic Facial Feature Detection.* Pattern Recognition, vol. 26, no. 12, pp. 1739–1755, 1993.

[Cro95] J. L. Crowley, F. Berard, J. Coutaz: *Finger Tracking as an Input Device for Augmented Reality.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 195–200, 1995.

[Dar95] T. Darrell and A. P. Pentland: *Attention-Driven Expression and Gesture Analysis in an Interactive Environment.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 135–140, 1995.

[Dav80] S. B. Davis, P. Mermelstein: *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences.* IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-28, no. 4, pp. 357–366, 1980.

[Del98] Q. Delamarre, O. Faugeras: *Finding pose of hand in video images: a stereo-based approach.* IEEE Proc. of the 3rd Int. Conference on Automatic Face- and Gesture-Recognition, Nara, Japan, pp. 585–590, 1998.

[Den97] Y. Deng, B. S. Manjunath: *Content-Based Search of Video Using Color, Texture, and Motion.* IEEE Proc. of the 1997 Int. Conference on Image Processing, Santa Barbara, USA, vol. II, pp. 534–537, 1997.

[Efr72] D. Efron: *Gesture, Race and Culture.* Mouton & Co., The Hague, The Netherlands, 1972 (Reprint of D. Efron: *Gesture and Environment.* King's Crown Press, New York, USA, 1941).

[Ekm95] P. Ekman: *Essential Behavioral Science of the Face and Gesture that Computer Scientists Need to Know.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 7–11, 1995.

[Fel84] K. Fellbaum: *Sprachverarbeitung und Sprachübertragung.* Springer, Berlin, 1984.

[Fie95] K. H. Fielding, D. W. Ruck: *Recognition of Moving Light Displays Using Hidden Markov Models.* Pattern Recognition, vol. 28, no. 9, pp. 1415–1421, 1995.

[Fol90] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes: *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, Massachusetts, USA, 1990.

[Fre95] W. T. Freeman and C. D. Weissman: *Television Control by Hand Gestures.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 179–183, 1995.

[Fre96] W. T. Freeman, K. Tanaka, J. Ohta, K. Kyuma: *Computer Vision for Computer Games.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 100–105, 1996.

[Gon87] R. C. Gonzalez, P. Wintz: *Digital Image Processing.* 2nd edition, Addison-Wesley, Reading, Massachusetts, USA, 1987.

[Har91] R. M. Haralick, L. G. Shapiro: *Glossary of Computer Vision Terms.* Pattern Recognition vol. 24, no. 1, pp. 69–93, 1991.

[Hau89] A. G. Hauptmann: *Speech and Gestures for Graphic Image Manipulation.* ACM Proc. of the Human Factors in Computing Systems (CHI-89), Austin, USA, pp. 241–245, 1989.

[Hea96] T. Heap and D. Hogg: *Towards 3D Hand Tracking Using a Deformable Model.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 140–145, 1996.

[Hie96] H. Hienz, K. Grobel, G. Offner: *Real-Time Hand-Arm Motion Analysis Using a Single Video Camera.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 323–327, 1996.

[Hof98] F. G. Hofmann, P. Heyer, G. Hommel: *Velocity Profile Based Recognition of Dynamic Gestures with Discrete Hidden Markov Models.* Gesture and Sign Language in Human-Computer Interaction: Proc. of the Int. Gesture Workshop 1997, Bielefeld, Germany; Springer, Berlin, pp. 81–95, 1998.

[Hou92] S. Houde: *Iterative Design of an Interface for Easy 3-D Direct Manipulation.* ACM Proc. of the Human Factors in Computing Systems (CHI-92), Monterey, USA, pp. 135–142, 1992.

[Hu62] M.-K. Hu: *Visual Pattern Recognition by Moment Invariants.* IRE Transactions on Information Theory, vol. IT-8, pp. 179–187, 1962.

[Hua90] X. D. Huang, Y. Ariki, M. A. Jack: *Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh, Great Britain, 1990.

[Hua95] T. S. Huang, V. I. Pavlović: *Hand Gesture Modeling, Analysis, and Synthesis.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 73–79, 1995.

[Jai89] A. K. Jain: *Fundamentals of Digital Image Processing.* Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1989.

[Jo98] K.-H. Jo, Y. Kuno, Y. Shirai: *Manipulative Hand Gesture Recognition Using Task Knowledge for Human Computer Interaction.* IEEE Proc. of the 3rd Int. Conference on Automatic Face- and Gesture-Recognition, Nara, Japan, pp. 468–473, 1998.

[Jua85] B.-H. Juang: *Maximum Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains.* AT&T Technical Journal, vol. 64, no. 6, pp. 1235–1270, 1985.

[Jun96] J. Junkawitsch, L. Neubauer, H. Höge, G. Ruske: *A New Keyword Spotting Algorithm with Pre-Calculated Optimal Thresholds*: IEEE Proc. of the 1996 Int. Conference on Spoken Language Processing, Philadelphia, USA, pp. 2067-2070, 1996.

[Ken86] A. Kendon: *Current Issues in the Study of Gesture.* In J.-L. Nespoulous, P. Perron, A. R. Lecours (eds.): The Biological Foundation of Gestures: Motor and Semiotic Aspects, Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA, pp. 23–47, 1986.

[Kje96] R. Kjeldsen, J. Kender: *Finding Skin in Color Images.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 312–317, 1996.

[Kri90] D. J. Kriegman, J. Ponce: *On Recognizing and Positioning Curved 3-D Objects from Image Contours.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 12, pp. 1127–1137, 1990.

[Kro86] K. Kroschel: *Statistische Nachrichtentheorie. Erster Teil: Signalerkennung und Parameterschätzung.* 2. Auflage, Springer, Berlin, 1986.

[Kru91] M. W. Krueger: *Artificial Reality II.* 2nd ed., Addison-Wesley, Reading, Massachusetts, USA, 1991.

[Lan94a] M. Lang: *Aspects of Human-Machine-Communication.* Proceedings of the CRIM/FORWISS Workshop on Progress and Prospects of Speech Research and Technology, München, Germany, pp. 1–8, 1994.

[Lan94b] M. Lang: *Towards User Adequate Human-Computer-Interaction.* In B. Horvath, Z. Kačič (eds.): Modern Modes of Man-Machine-Communication, BiG Design, Maribor, Slovenia, pp. 1/1–1/9, 1994.

[Lan94c] M. Lang, H. Stahl: *Spracherkennung für einen ergonomischen Mensch-Maschine-Dialog.* mikroelektronik, Bd. 8, Nr. 2, S. 78–82, 1994.

[Lan99a] M. Lang: *Mensch-Maschine-Kommunikation 1*, Vorlesungsmanuskript, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 3. Auflage, 1999.

[Lan99b] M. Lang: *Mensch-Maschine-Kommunikation 2.* Vorlesungsmanuskript, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1999.

[Lan95] A. Lanitis, C. J. Taylor, T. F. Cootes, T. Ahmed: *Automatic Interpretation of Human Faces and Hand Gestures Using Flexible Models.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 98–103, 1995.

[Li91] B.-C. Li, J. Shen: *Fast Computation of Moment Invariants.* Pattern Recognition, vol. 24, no. 8, pp. 807–813, 1991.

[Lia98] R.-H. Liang, M. Ouhyoung: *A Real-Time Continuous Gesture Recognition System for Sign Language*, IEEE Proc. of the 3rd Int. Conference on Automatic Face- and Gesture-Recognition, Nara, Japan, pp. 558–565, 1998.

[Lin80] Y. Linde, A. Buzo, R. M. Gray: *An Algorithm for Vector Quantizer Design.* IEEE Transactions on Communications, vol. COM-28, no. 1, 84–95, 1980.

[Lück96] Michael Lücke: *Dreidimensionale graphische Modellierung der Hand und Anpassung der Modellparameter an zweidimensionale Aufnahmen.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1996.

[Mag93] C. Maggioni: *A Novel Gestural Input Device for Virtual Reality.* Proc. of the IEEE Annual Virtual Reality International Symposium (VRAIS-93), Seattle, USA, pp. 118–124, 1993.

[Mag94] C. Maggioni: *Non Immersive Control of Virtual Environments.* Proc. of the Virtual Reality '94 — Anwendungen und Trends, Stuttgart; Springer, Berlin, pp. 129–142, 1994.

[Moh98] R. Mohan: *Video Sequence Matching*, IEEE Proc. of the 1998 Int. Conference on Acoustics, Speech, and Signal Processing, Seattle, USA, pp. 3697–3700, 1998.

[Mor81] D. Morris: *Der Mensch, mit dem wir leben: Ein Handbuch unseres Verhaltens.* Droemersche Verlagsanstalt, München, 1981.

[Mor97a] P. Morguet, M. Lang: *Feature Extraction Methods for Consistent Spatio-Temporal Image Sequence Classification Using Hidden Markov Models.* IEEE Proc. of the 1997 Int. Conference on Acoustics, Speech, and Signal Processing 1997, München, Germany, pp. 2893–2896, 1997.

[Mor97b] P. Morguet, M. Lang: *A Universal HMM-Based Approach to Image Sequence Classification.* IEEE Proc. of the 1997 Int. Conference on Image Processing, Santa Barbara, USA, vol. III, pp. 146–149, 1997.

[Mor98a] P. Morguet, M. Lang: *Dynamische Gesten und indirekte Manipulation als Grundlage für eine intuitive Mensch-Maschine-Kommunikation.* Tagungsband ITG-Fachtagung: Technik für den Menschen — Gestaltung und Einsatz benutzerfreundlicher Produkte, Eichstätt; VDE-Verlag, Berlin, S. 131–139, 1998.

[Mor98b] P. Morguet, M. Lang: *An Integral Stochastic Approach to Image Sequence Segmentation and Classification.* IEEE Proc. of the 1998 Int. Conference on Acoustics, Speech, and Signal Processing, Seattle, USA, pp. 2705–2708, 1998.

[Mor98c] P. Morguet, M. Lang: *Spotting Dynamic Hand Gestures in Video Image Sequences Using Hidden Markov Models.* IEEE Proc. of the 1998 Int. Conference on Image Processing, Chicago, USA, vol. III, pp. 193–197, 1998.

[Mor99] P. Morguet, M. Lang: *Comparison of Approaches to Continuous Hand Gesture Recognition for a Visual Dialog System.* IEEE Proc. of the 1999 Int. Conference on Acoustics, Speech, and Signal Processing, Phoenix, USA, pp. 3549-3552, 1999.

[Mül97] J. Müller: *Die semantische Gliederung zur Repräsentation des Bedeutungsinhalts innerhalb sprachverstehender Systeme.* Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, 1997.

[Nag96] S. Nagaya, S. Seki, R. Oka: *A Theoretical Consideration of Pattern Space Trajectory for Gesture Spotting Recognition.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 72–77, 1996.

[Nat95] K. S. Nathan, H. S. M. Beigi, J. Subrahmonia, G. J. Clary, H. Maruyama: *Real-Time On-Line Unconstrained Handwriting Recognition Using Statistical Methods.* IEEE Proc. of the 1995 Int. Conference on Acoustics, Speech, and Signal Processing, Detroit, USA, pp. 2619–2622, 1995.

[Nef98] A. V. Nefian, M. H. Hayes III: *Hidden Markov Models for Face Recognition.* IEEE Proc. of the 1998 Int. Conference on Acoustics, Speech, and Signal Processing, Seattle, USA, pp. 2721–2724, 1998.

[Nes86] J.-L. Nespoulous, P. Perron, A. R. Lecours: *Introduction.* In J.-L. Nespoulous, P. Perron, A. R. Lecours (eds.): The Biological Foundation of Gestures: Motor and Semiotic Aspects, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.

[Nes95] P. Nesi, A. Del Bimbo: *Hand Pose Tracking for 3-D Mouse Emulation.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 302–307, 1995.

[New92] W. Newman, P. Wellner: *A Desk Supporting Computer-Based Interaction with Paper Documents*, ACM Proc. of the Human Factors in Computing Systems (CHI-92), Monterey, USA, pp. 587–592, 1992.

[Nis96] T. Nishimura, R. Oka: *Spotting Recognition of Human Gestures from Time-Varying Images.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 318–322, 1996.

[Pal93] N. R. Pal, S. K. Pal: *A Review on Image Segmentation Techniques.* Pattern Recognition, vol. 26, no. 9, pp. 1277–1294, 1993.

[Pav94] T. Pavlidis: *Algorithmen zur Grafik- und Bildverarbeitung.* 5. Auflage, Heise, Hannover, 1994.

[Pav96] V. I. Pavlović, R. Sharma, T. S. Huang: *Gestural Interface to a Visual Computing Environment for Molecular Biologists.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 30–35, 1996.

[Pav97] V. I. Pavolović, G. A. Berry, T. S. Huang: *Integration of Audio/Visual Information for Use in Human-Computer Intelligent Interaction.* IEEE Proc. of the 1997 Int. Conference on Image Processing, Santa Barbara, USA, vol. I, pp. 121–124, 1997.

[Phi93] W. Philips: *A New Fast Algorithm for Moment Computation.* Pattern Recognition, vol. 26, no. 11, pp. 1619–1621, 1993.

[Pla95] B. Plannerer: *Erkennung fließender Sprache mit integrierten Suchmethoden.* Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, 1995.

[Pot98] G. Potamianos, H. P. Graf: *Discriminative Training of HMM Stream Exponents for Audio-Visual Speech Recognition.* IEEE Proc. of the 1998 Int. Conference on Acoustics, Speech, and Signal Processing, Seattle, USA, pp. 3733–3736, 1998.

[Pra91] W. K. Pratt: *Digital Image Processing.* 2nd edition, John Wiley and Sons, New York, USA, 1991.

[Pre90] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery: *Numerical Recipes in C: the Art of Scientific Computation.* 2nd edition, Cambridge University Press, New York, USA, 1994.

[Que95] F. K. H. Queck, T. Mysliwiec, M. Zhao: *FingerMouse: A Freehand Pointing Interface.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 372–377, 1995.

[Que96a] F. K. H. Quek: *Unencumbered Gestural Interaction.* IEEE MultiMedia, vol. 3, no. 4, pp. 36–47, 1996.

[Que96b] F. K. H. Quek, M. Zhao: *Inductive Learning in Hand Pose Recognition.* IEEE Proc. of the 2nd Int. Conference on Automatic Face- and Gesture-Recognition, Killington, Vermont, USA, pp. 78–83, 1996.

[Rab89] L. R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[Rei96] W. Reichl: *Diskriminative Lernverfahren für die automatische Spracherkennung.* Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, 1996.

[Rig96] G. Rigoll, A. Kosmala, M. Schuster: *A New Approach to Video Sequence Recognition Based on Statistical Methods.* IEEE Proc. of the 1996 Int. Conference on Image Processing, Lausanne, Switzerland, vol. III, p. 839–842, 1996.

[Rig97] G. Rigoll, A. Kosmala: *New Improved Feature Extraction Methods for Real-Time High Performance Image Sequence Recognition.* IEEE Proc. of the 1997 Int. Conference on Acoustics, Speech, and Signal Processing, München, Germany, pp. 2901–2904, 1997.

[Rig98] G. Rigoll, A. Kosmala, S. Eickeler: *High Performance Real-Time Gesture Recognition Using Hidden Markov Models.* Gesture and Sign Language in Human-Computer Interaction: Proc. of the Int. Gesture Workshop 1997, Bielefeld, Germany; Springer, Berlin, pp. 69–80, 1998.

[Rus93] G. Ruske: *Datenanalyse und Informationsreduktion.* Vorlesungsmanuskript, Lehrstuhl für Mensch-Maschine-Kommunikation, 1993.

[Sah97] E. Sahouria, A. Zakhor: *Motion Indexing of Video.* IEEE Proc. of the 1997 Int. Conference on Image Processing, Santa Barbara, USA, vol. II, pp. 526–529, 1997.

[Sch93] R. Schuster, S. Ahmad: *Modellbasierte Beschreibung von Farbhistogrammen und Segmentation von Farbbildern.* Tagungsband DAGM-Symposium Mustererkennung 1993, Lübeck; Springer, Berlin, S. 303–311, 1993.

[Sch94a] J. Schlenzig, E. Hunter, R. Jain: *Recursive Identification of Gesture Inputs Using Hidden Markov Models.* IEEE Proc. of the 2nd Annual Conference on Applications of Computer Vision, pp. 187–194, 1994.

[Sch96a] A. Schreyer: *Bildverarbeitungsgestützte Verfahren zur Modellierung der Augen für den visuellen Mensch-Maschine-Dialog.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1996.

[Sch96b] M. Schuster, G. Rigoll: *Fast Online Video Image Sequence Recognition with Statistical Methods.* IEEE Proc. of the 1996 Int. Conference on Acoustics, Speech, and Signal Processing, Atlanta, USA, pp. 3450–3453, 1996.

[Sch94b] N. Schweiger (ed.): *IRIS Media Libraries Programming Guide.* Silicon Graphics, Inc., 1994.

[Spe95] T. Sperlich: *Digitale Kreaturen: Charakteranimation auf dem Weg zum „virtuellen Menschen“.* c't, Magazin für Computertechnik, Nr. 3, S. 92–105, 1995.

[Sta97] H. Stahl: *Konsistente Integration stochastischer Wissensquellen zur semantischen Decodierung gesprochener Äußerungen.* Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, 1997.

[Sta95] T. Starner and A. Pentland: *Visual Recognition of American Sign Language Using Hidden Markov Models.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 189–194, 1995.

[Ste87] W. Stehle: *Allgemeine Optimierungsverfahren.* Vorlesungsmanuskript, Institut für Nachrichtensysteme, Universität Karlsruhe (TH), 1987.

[Ste99] G. Steinmaßl: *Evaluierung und Optimierung einer stochastischen Gestikerkennung.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1999.

[Stö98] G. Stöckl: *Merkmalsextraktionsverfahren für die Bildsequenzerkennung.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1998.

[Tea80] M. R. Teague: *Image Analysis via the General Theory of Moments.* Journal of the Optical Society of America, vol. 70, no. 8, pp. 920–930, 1980.

[Ven93] D. Venolia: *Facile 3D Direct Manipulation.* Proc. of the Human Factors in Computing Systems (INTERCHI-93), Amsterdam, The Netherlands, pp. 31–36, 1993.

[Vol97] D. Vollmerhaus: *Implementierung und Evaluierung von 3-D-Applikationen für den gestikbasierten Mensch-Maschine-Dialog.* Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1997.

[Wat98] T. Watanabe, M. Yachida: *Real Time Recognition Using Eigenspace from Multi Input Image Sequences.* IEEE Proc. of the 3rd Int. Conference on Automatic Face- and Gesture-Recognition, Nara, Japan, pp. 428–433, 1998.

[Wei89] D. Weimer, S. K. Ganapathy: *A Synthetic Visual Environment with Hand Gesturing and Voice Input.* ACM Proc. of the Human Factors in Computing Systems (CHI-89), Austin, USA, pp. 235–240, 1989.

[Wel91] P. Wellner. *The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display*, ACM Proc. of the 10th Annual Symposium on User Interface Software and Technology (UIST-91), Hilton Head, USA, pp. 27–33, 1991.

[Wel93a] P. Wellner: *Interacting with Paper on the DigitalDesk*, Communications of the ACM, vol. 36, no. 7, pp. 87–96, 1993.

[Wel93b] P. Wellner, W. Mackay, R. Gold: *Computer-Augmented Environments: Back to the Real World.* Special Issue of Communications of the ACM, vol. 36, no. 7, 1993.

[Wex94] A. D. Wexelblat: *A Feature-Based Approach to Continuous-Gesture Analysis.* Master's thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, 1994.

[Wil95] A. D. Wilson, A. F. Bobick: *Configuration States for the Representation and Recognition of Gesture.* Proc. of the 1st Int. Workshop on Automatic Face- and Gesture-Recognition, Zürich, Switzerland, pp. 129–134, 1995.

[Win96] H.-J. Winkler: *Entwurf und Realisierung eines auf statistischen Ansätzen basierenden Systems zur Erkennung handgeschriebener mathematischer Formeln.* Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, 1997.

[Yam92] J. Yamato, J. Ohya, K. Ishii: *Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model.* IEEE Proc. of the Int. Conference on Computer Vision and Pattern Recognition 1992, Champaign, Illinois, USA, pp. 379–385, 1992.

[Yea99] M. Yeasin and S. Chaudhuri: *Dynamic Hand Gesture Understanding — A New Approach.* IEEE Proc. of the 1999 Int. Conference on Acoustics, Speech, and Signal Processing, Pheonix, USA, p. 3073–3076, 1999.

[Yui89] A. L. Yuille, D. S. Cohen, P. W. Hallinan: *Feature Extraction from Faces Using Deformable Templates.* IEEE Proc. of the 1989 Int. Conference on Computer Vision and Pattern Recognition, San Diego, USA, pp. 104–109, 1989.