

Institut für Prozessrechentchnik, Automation und Robotik  
Fakultät für Informatik  
Universität Karlsruhe (TH)

---

# *Diplomarbeit*

*Gestenerkennung zur Anweisung eines  
mobilen Roboters*

---

Tobias Lütticke

Juli 2000

Referent: Prof. Dr.-Ing. Rüdiger Dillmann

Korreferent: Prof. Dr.-Ing. H. Wörn

Betreuer: Dipl.-Inform. Markus Ehrenmann



Hiermit erkläre ich, die vorliegende Diplomarbeit selbständig angefertigt zu haben. Die verwendeten Quellen sind im Text gekennzeichnet und im Literaturverzeichnis aufgeführt.

Karlsruhe, 3. Juli 2000

---

Tobias Lütticke



# ***Inhalt***

<b><i>Abbildungsverzeichnis</i></b>	<b>11</b>
<b><i>Tabellenverzeichnis</i></b>	<b>13</b>
<b>1 <i>Einleitung</i></b>	<b>13</b>
1.1 Hintergründe und Motivation . . . . .	13
1.2 Aufgabenstellung . . . . .	14
1.3 Überblick . . . . .	14
<b>2 <i>Mensch-Roboterinteraktion</i></b>	<b>17</b>
2.1 Das Projekt Morpha . . . . .	17
2.2 Beitrag zum Projekt Morpha . . . . .	18
2.2.1 Architektur des Verarbeitungssystems . . . . .	18
2.2.2 Komponenten des Verarbeitungssystems . . . . .	19
2.3 Einordnung dieser Arbeit in das Projekt . . . . .	20
2.4 Stand der Technik . . . . .	21
2.4.1 Beobachtung und Szenenanalyse . . . . .	21
2.4.2 Gestenerkennung . . . . .	21
2.4.2.1 Erkennung koreanischer Zeichensprache . . . . .	22
2.4.2.2 Erkennung amerikanischer Zeichensprache mit Hidden-Markov-Modellen . . . . .	23
2.4.2.3 Visuelle Gestenerkennung durch Roboter . . . . .	24
2.4.2.4 Gestenerkennung und Handverfolgung mittels aktiver Konturen . . . . .	26

<b>3</b>	<b><i>Technische Grundlagen</i></b>	<b>29</b>
3.1	Bildverarbeitung . . . . .	29
3.1.1	Farbräume und Farbraumkonvertierung . . . . .	30
3.1.2	Filter . . . . .	31
3.2	Hidden-Markov-Modelle . . . . .	32
3.2.1	Diskrete Markov-Prozesse . . . . .	33
3.2.2	Struktur und Elemente eines Hidden-Markov-Modells . . . . .	34
3.2.2.1	Struktur eines Hidden-Markov-Modells . . . . .	34
3.2.2.2	Elemente eines Hidden-Markov-Modells . . . . .	36
3.2.3	Arten von Hidden-Markov-Modellen . . . . .	37
3.2.4	Grundlegende Probleme bei der Anwendung von Hidden-Markov-Modellen . . . . .	40
3.2.5	Der Vorwärts-Algorithmus . . . . .	40
3.2.6	Der Viterbi-Algorithmus . . . . .	44
3.2.6.1	Beispielablauf des Viterbi-Algorithmus . . . . .	46
3.2.7	Der Baum-Welch-Algorithmus . . . . .	47
<b>4</b>	<b><i>Entwurf und Implementierung</i></b>	<b>53</b>
4.1	Gestenauswahl . . . . .	54
4.2	Objekterkennung und Objektverfolgung . . . . .	55
4.2.1	Farbraumkonvertierung und Binarisierung . . . . .	55
4.2.2	Handverfolgung . . . . .	57
4.3	Vektorquantisierung . . . . .	59
4.3.1	Codebucherzeugung . . . . .	59
4.4	Hidden-Markov-Modelle . . . . .	63
4.4.1	Hidden-Markov-Modelle versus neuronale Netze . . . . .	63
4.4.2	Referenzmodelle . . . . .	64
4.4.3	Bestimmung initialer Modellparameter . . . . .	69
4.4.3.1	Wahl der Sprungbegrenzung . . . . .	70
4.4.3.2	Bestimmung der Symbolausgabewahrscheinlichkeiten . . . . .	70

---

4.4.4	Training der Referenzmodelle . . . . .	74
4.4.4.1	Training mit mehrfachen Beobachtungssequenzen	75
4.4.4.2	Variable versus feste Längen von Beobachtungssequenzen . . . . .	78
4.4.4.3	Modelltraining . . . . .	80
4.4.5	Schwellwertmodell . . . . .	81
4.4.5.1	Beispiel zum Aufbau eines Schwellwertmodells .	84
4.5	Mehrstufiges Filterkonzept . . . . .	85
4.6	Hierarchische Gestenklassifikation . . . . .	88
4.7	Gesamtablauf . . . . .	91
4.7.1	Erkennungstests . . . . .	93
4.8	Experimentelle Ergebnisse . . . . .	95
4.8.1	Einfluss der Benutzeranzahl . . . . .	100
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>103</b>
5.1	Zusammenfassung . . . . .	103
5.2	Ausblick . . . . .	104
5.2.1	Handverfolgung . . . . .	104
5.2.2	Erkennung dreidimensionaler Gesten . . . . .	104
5.2.3	Berücksichtigung der Fingerstellung . . . . .	105
5.2.4	Überdeckungen . . . . .	105
5.2.5	Erkennung mittels Ebenenbildung . . . . .	106
5.2.6	Verbundene Parameter für Hidden-Markov-Modelle . . .	107
5.2.7	Optimierungskriterien für das Hidden-Markov-Modell-Training . . . . .	107
	<b>Literatur</b>	<b>109</b>



# Abbildungsverzeichnis

2.1	Gesamtarchitektur des Verarbeitungssystems . . . . .	22
2.2	Kamerakopf des Verarbeitungssystems . . . . .	24
3.1	Beziehung zwischen den Farbräumen RGB und HSI . . . . .	35
3.2	Bei der Bildverarbeitung eingesetzte Filter . . . . .	36
3.3	Markov-Modell mit den vier Zuständen und ausgewählten Übergängen . . . . .	37
3.4	Endlicher Automat eines Hidden-Markov-Modells . . . . .	39
3.5	Gitterdiagramm eines Hidden-Markov-Modells . . . . .	39
3.6	Links-Rechts-Hidden-Markov-Modell mit Sprungbegrenzung .	42
3.7	Gitterdiagramm eines Links-Rechts-Hidden-Markov-Modells .	43
3.8	Schematische Darstellung der Berechnung einer Vorwärtsvariablen	45
3.9	Schematische Darstellung der Berechnung einer Vorwärtsvariablen mit zugehörigem Hidden-Markov-Modell . . . . .	46
3.10	Schematische Darstellung der Berechnung einer Rückwärtsvariablen . . . . .	47
3.11	Gitterdiagramm zur Viterbi-Pfadermittlung . . . . .	51
3.12	Schrittweise Entwicklung des Viterbi-Ergebnispfades . . . . .	52
3.13	Berechnung des gemeinsamen Ereignisses, dass sich das System bei $t$ in Zustand $s_i$ und bei $t + 1$ in Zustand $s_j$ befindet . . . . .	53
3.14	Training eines Hidden-Markov-Modells mit dem Baum-Welch-Algorithmus . . . . .	56
4.1	Die fünf vom System zu erkennenden Gesten . . . . .	58

4.2	Beispiel für Häufung und Verteilung von Hautfarbwerten . . . . .	60
4.3	Beispiel für eine Binarisierung: Entfernung der Sättigungs- und Intensitätsinformationen eines HSI-Bildes . . . . .	61
4.4	Ergebnisbild nach Anwendung der Opening- und Closing-Filter	61
4.5	Ablauf bei der Handverfolgung, basierend auf einer Kamerabildsequenz . . . . .	62
4.6	Beispiel für eine zweidimensionale Regionenaufteilung bei der Vektorquantisierung . . . . .	64
4.7	Das zur Vektorquantisierung verwendete Codebuch mit 16 Codewörtern (Richtungsindizes) . . . . .	66
4.8	Entstehung der Bewegungsvektoren aus aufeinander folgenden Pixelkoordinaten . . . . .	66
4.9	Reduzierung der Segmentanzahl einer Gestentrajektorie mittels Filterung . . . . .	72
4.10	Wahrscheinlichkeiten bei der Erkennung für Modelle mit unterschiedlichem $\Delta$ . . . . .	74
4.11	Filterung und Gruppierung einer Beobachtungssequenz als Vorbereitung zur Bestimmung der Werte für $B$ . . . . .	78
4.12	Entwicklung der Erkennungswahrscheinlichkeiten beim Training mit mehrfachen Beobachtungssequenzen unterschiedlicher und gleicher Länge . . . . .	83
4.13	Struktur des Schwellwertmodells als ergodisches Hidden-Markov-Modell . . . . .	87
4.14	Schematischer Ablauf des Erkennungsmechanismus mit Hilfe des Schwellwertmodells . . . . .	87
4.15	Ergebnisse beim Filtern einer Beobachtungssequenz . . . . .	91
4.16	Abnahme der Segmentanzahl der Gestentrajektorien bei Anwendung aller Filter . . . . .	92
4.17	Klassifizierung der Referenzgesten gemäß ihrer Komplexität . . .	93
4.18	Der Gesten-Erkennungsprozess mit seinen Einzelschritten im Überblick . . . . .	96
4.19	Testhierarchie mit möglichen Einstellungen zum Erkennungstest	97
4.20	Trajektorien von erkannten Gesten des Typs drei mit ihren jeweiligen Erkennungswahrscheinlichkeiten . . . . .	101

# Tabellenverzeichnis

2.1	Technische Daten der verwendeten Kameras . . . . .	24
4.1	Zuordnung zwischen Bewegungsvektoren und Richtungsindizes	67
4.2	Erkennungswahrscheinlichkeiten bei wachsender Anzahl von Zuständen . . . . .	69
4.3	Experimentell bestimmte Zustandsanzahl der Referenzmodelle .	72
4.4	Wahrscheinlichkeiten bei gleichmäßig verteiltem und manuell bestimmtem $B$ . . . . .	75
4.5	Wahrscheinlichkeitsentwicklung einzelner Beobachtungssequenzen bei gleichmäßig verteiltem und manuell bestimmtem $B$ . . .	75
4.6	Wahrscheinlichkeitsentwicklung beim Training von drei Beobachtungssequenzen . . . . .	81
4.7	Art und Anzahl der zum Training der Referenzmodelle verwendeten Gesten . . . . .	85
4.8	Verschiedene Sequenzlängen für die Gesten, mit deren Hilfe die Grundlage für die hierarchische Gestenklassifikation bestimmt wird	95
4.9	Parameter für das Erkennungssystem . . . . .	99
4.10	Erkennungserfolg der einzelnen Gesten beim Verfahren der maximalen Wahrscheinlichkeit . . . . .	100
4.11	Erkennungserfolg der einzelnen Gesten bei Nutzung des Schwellwertmodells . . . . .	102
4.12	Erkennungserfolg der einzelnen Gesten bei Anwendung der hierarchischen Gestenklassifikation . . . . .	103
4.13	Erkennungserfolg der einzelnen Gesten bei Anwendung der hierarchischen Gestenklassifikation kombiniert mit dem Schwellwertmodell . . . . .	104

4.14 Unterschiedlicher Erkennungserfolg in Abhängigkeit davon, ob auch mit Gesten der Testperson trainiert wurde . . . . .	104
---	-----





# **1 Einleitung**

## **1.1 Hintergründe und Motivation**

Der Bedarf an Robotersystemen und deren Einsatz weitet sich zunehmend dahingehend aus, dass nicht mehr primär stationäre, sondern vielmehr auch mobile Roboter eingesetzt werden. Typische Einsatzgebiete finden sich im Service-Bereich. In Werkstatt und Haushalt sind zum Beispiel vielfältige Einsatzmöglichkeiten denkbar. Etablierte industriell genutzte Robotersysteme zeichnen sich durch Abhängigkeit von der Vorstrukturierung ihrer Umgebung und der Unfähigkeit, sich selbständig auf veränderte Bedingungen einzustellen, aus. Sie sind für eine bestimmte Aufgabe entworfen und programmiert. Veränderte räumliche Gegebenheiten oder eine neue Aufgabe erfordern eine Umprogrammierung.

Im Unterschied zu diesen klassischen Systemen bringen Anwendungen für mobile Roboter zum Beispiel im Service-Bereich die Notwendigkeit mit sich, ständig auf neue Rahmenbedingungen reagieren zu müssen. Hierzu zählen nicht nur die mit wechselnden Einsatzorten einhergehenden räumlichen Veränderungen, sondern auch eine Vielzahl von menschlichen Bedienern, die den mobilen Robotern ihre jeweiligen Aufgaben zuweisen. Da es sich bei diesen Personen in der Regel nicht um Fachkräfte, sondern um Anwender handelt, muss die Bedienung möglichst unkompliziert und intuitiv erfolgen können.

Weiterhin soll die Kommandierung der Robotersysteme ohne zusätzliche Schnittstellengeräte möglich sein, da der Anwender diese als hinderlich empfindet und sie deswegen im Zusammenhang mit der angestrebten Akzeptanz durch den Anwender kontraproduktiv wirken. Die Kommandierung soll also sowohl ohne Spezialkleidung wie zum Beispiel den Datenhandschuh, als auch kontextabhängig und für den Einzelfall erfolgen können. Zu diesem Zweck ist beispielsweise ein aktives Sichtsystem in Form eines Kamerakopfes auf Seiten des Roboters zur Umwelterfassung nach menschlichem Vorbild interessant.

In diesem Sinne werden Roboter immer mehr als Assistenzsysteme betrachtet, die dem Menschen bei der Bewältigung von vielerlei Aufgaben zur Hand gehen.

## 1.2 Aufgabenstellung

Um den oben ausgeführten Anforderungen gerecht werden zu können, soll ein System entwickelt werden, das die Aspekte der einfachen Bedienbarkeit, Unabhängigkeit von zusätzlicher Hardware und möglicher Bedienung durch wechselnde Benutzer in sich vereint. Zu diesem Zweck ist ein kamerabasiertes und echtzeitfähiges System zu entwerfen, das die Kommandierung eines mobilen Assistenten durch Gesten erlaubt. Eine *Geste* ist eine menschliche Handbewegung, die sich aus einem dynamischen Teil, der eigentlichen Bewegung, und einem statischen Teil, der abschließenden Handposition und Fingerstellung, zusammensetzt. Während der statische Teil der Geste in dieser Arbeit unberücksichtigt bleibt, beschränkt sich die Auswertung auf die Bewegung und die Ermittlung ihrer Bedeutung. Die Hinzunahme des statischen Teils ist Gegenstand weiterführender Untersuchungen (Abschnitt 5.2.3). Funktionalität ist in den beiden Bereichen Bildverarbeitung und Gestenerkennung zu realisieren.

Die Bildverarbeitung soll die Extraktion der zur Handverfolgung nötigen Informationen ermöglichen. Sie soll dies in einer Geschwindigkeit leisten, die auch noch mit dem nachfolgenden zur Gestenerkennung nötigen Rechenaufwand eine Erkennung in Echtzeit erlaubt. Unterschiedlich schnelle Handbewegungen sollen sowohl für die Handverfolgung als auch für die Erkennung keinen Unterschied bedeuten.

Für die Modellierung und die Erkennung der Gesten sind *Hidden-Markov-Modelle* (HMM) und ihre Algorithmen anzuwenden. Die Modelle sollen dabei so realisiert werden, dass sich die Erkennung möglichst robust gestaltet. Dies soll verhindern, dass Gesten eine falsche Bedeutung zugeordnet wird oder bedeutungslose Handlungen als sinnvolle interpretiert werden.

## 1.3 Überblick

Die beiden vorangegangenen Abschnitte haben das Interesse an der Gestenerkennung begründet und die Aufgabenstellung für diese Diplomarbeit dargelegt. Der restliche Teil der Arbeit gliedert sich folgendermaßen: Der Rahmen dieser Arbeit wird durch das Projekt *Intelligente anthropomorphe Assistenzsysteme (Morpha)*, Leitprojekt des Bundesministeriums für Bildung und Forschung (BMBF) im Rahmenprogramm *Mensch-Technik-Interaktion* vorgegeben [BMBF00] und in Kapitel 2 näher dargestellt. Dabei wird die Architektur des Verarbeitungssystems skizziert und in den Stand der Technik eingeführt. Ferner werden unterschiedliche Projekte und Systeme beleuchtet, die sich ebenfalls

mit dem Problem der Gestenerkennung auseinander setzen, aber andersartige Lösungsansätze verfolgen.

Kapitel 3 gibt eine Einführung in die technischen Grundlagen, auf denen diese Arbeit basiert und die der Umsetzung zugrunde liegen. Das dort vermittelte Verständnis der Hidden–Markov–Modelle und der Verfahren der Bildverarbeitung ist für die nachfolgende Realisierung unabdingbar.

Die Umsetzung der Anforderungen wird im Kapitel 4 beschrieben. Hier wird näher auf den Lösungsweg eingegangen und das Ergebnis begründet. Außerdem werden die Auswirkungen von Parameteränderungen und Maßnahmen zur Effizienzsteigerung untersucht.

Als Abschluss dieser Arbeit dient Kapitel 5. Hier werden die erreichten Ergebnisse zusammenfassend skizziert und es wird ein Ausblick auf mögliche weitere Entwicklungen gegeben.



## 2 *Mensch–Roboterinteraktion*

### 2.1 *Das Projekt Morpha*

Über den eingangs (Abschnitt 1.1) erwähnten Aspekt der Mobilität hinaus erhöhen sich außerdem die Anforderungen an Robotersysteme zusehends. Sie sollen möglichst selbständig und in komplexen, sich ändernden Umgebungen agieren können. Genauso wie die sehr verbreiteten stationären Systeme gehen die bisherigen Ansätze zur Implementierung von Robotersteuerungen und mobilen Systemen von einem hohen Grad an Spezialisierung aus. Die Funktionsfähigkeit ist auf eine bestimmte Aufgabe in einer bestimmten Arbeitsumgebung begrenzt, die Benutzerinteraktion auf maschinennahe Schnittstellen reduziert.

Um nun mobile Assistenten entwickeln zu können, die auch in komplexen Umwelten sicher agieren, müssen bestehende Ansätze im Hinblick auf zwei Aspekte erweitert werden:

- Die Bereiche *Belehrbarkeit* und *Adaptivität* müssen dahingehend ausgebaut werden, dass wechselnden Umwelteigenschaften und Benutzern Rechnung getragen werden kann. Die Einsatzgebiete der intelligenten Assistenten umfassen sowohl die Industrie als auch den Heim- und Service-Bereich, so dass unterschiedlichste Anforderungen berücksichtigt werden müssen.
- Eine hohe Anzahl von wechselnden Benutzern mit unterschiedlichsten Voraussetzungen macht eine Schulung oder ein Training aller dieser Benutzer sehr kostenintensiv und deswegen unerwünscht. Statt dessen muss eine *anthropomorphe Interaktion* mit den Benutzern ermöglicht werden, so dass das Assistenzsystem auch mit ungeschulten Personen interagieren kann.

Im Zuge der Umsetzung dieser Anforderungen soll im Projekt *Morpha* ein erweiterter Zugang zu Assistenzsystemen geschaffen werden, über den diese kommandiert und belehrt werden können. Die Einsatzmöglichkeiten intelligenter

anthropomorpher Assistenten können auf diese Weise adaptiv und inkrementell erweitert werden. Ein zusätzlicher Vorteil besteht in der hiermit einhergehenden zunehmenden Akzeptanz durch die Benutzer.

Im ersten Schritt werden im Institut für Prozessrechenstechnik, Automation und Robotik (IPR) neue Kommunikationskanäle in bereits bestehende Assistenzsysteme eingefügt. Dabei ist die Einbindung von Modulen zur visuellen und auditiven Kommandierung geplant.

## 2.2 Beitrag zum Projekt Morpha

### 2.2.1 Architektur des Verarbeitungssystems

Der in dieser Arbeit vorgestellte Ansatz zur Gestenerkennung ist bildbasiert und folglich auf Kameras als Sensoren zur Beschaffung der Eingabedaten angewiesen. Die Architektur des Verarbeitungssystems gliedert sich in die drei Bereiche Arbeitsraum, Sensoren und Datenverarbeitung (Abbildung 2.1).

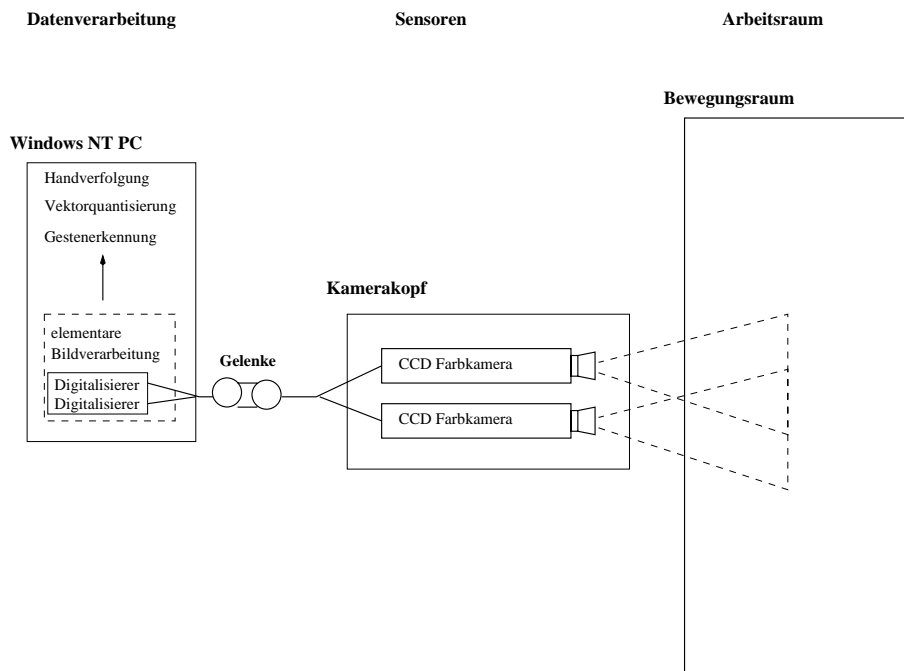


Abbildung 2.1: Gesamtarchitektur des Verarbeitungssystems

Der Arbeitsraum umfasst den Bereich, in dem die Geste vollführt wird. Er ist in der Tiefe dadurch beschränkt, dass weiter entfernte und damit kleiner erscheinende Objekte bei der Bildverarbeitung andere Parameter erfordern. Auf

Entfernungsveränderungen wird derzeit keine Rücksicht genommen. Ein bestimmter Abstand zur Kamera ist einzuhalten. Die Breite des Arbeitsraums unterliegt letztendlich auch nur der Entfernungsbeschränkung, da der drehbare Kamerakopf in der Lage ist, die Hand auch in großräumigeren Bewegungen zu verfolgen.

Der Bereich der Sensoren beinhaltet den aus zwei Kameras bestehenden Kamerakopf, von dem ein kontinuierlicher Strom von Farbbildern geliefert wird. Der Kamerakopf bietet mit seinen beiden Gelenken die Möglichkeit, die Handbewegung zuverlässig in alle Richtungen zu verfolgen (2.2).

Die Funktionalität der eigentlichen Gestenerkennung ist im Bereich Datenverarbeitung angesiedelt. Die vom Kamerakopf akquirierten Bilder fließen in die Digitalisierer, in denen sie gepuffert werden. Ausgehend von diesen Bildinformationen werden die zur Handverfolgung nötigen Steueranweisungen für den Kamerakopf generiert. Die Informationen über die Trajektorie, die die Handbewegung beschreibt, müssen sukzessive aus den Kamerabildern extrahiert werden. Die entstehende Koordinatenfolge dient als Eingabe in die Vektorquantisierung, deren Ergebnisse wiederum den Hidden-Markov-Modellen zugänglich gemacht werden. Diese treffen dann eine Entscheidung über Erkennen oder Nichterkennen der Geste. Einen detaillierten Überblick über diesen Ablauf liefert Abschnitt 4.7.

### **2.2.2 Komponenten des Verarbeitungssystems**

Das im vorigen Abschnitt erläuterte Verarbeitungssystem setzt sich im Wesentlichen aus den nachfolgend aufgeführten Komponenten zusammen:

- **Verarbeitungsrechner:**

Ein Windows NT-Rechner mit der folgenden Ausstattung:

- Doppelprozessorsystem mit zwei Intel Pentium III Prozessoren und einer Taktfrequenz von 500 MHz
- 256 Megabyte Arbeitsspeicher
- Matrox G400 Grafikkarte

- **Kameras:**

Beide Kameras sind vom Typ Sony 777AP und haben die in Tabelle 2.1 aufgelisteten Eigenschaften.

Merkmal	Eigenschaft
Chip-Größe	1/3" CCD Sensor
Verschlussgeschwindigkeit	1/60 s – 1/4000 s
Auflösung	752 × 582 Pixel
Signal-Rauschverhältnis	46 dB
Bildaufnahmeverfahren	Zeilensprungverfahren

Tabelle 2.1: Technische Daten der verwendeten Kameras



Abbildung 2.2: Kamerakopf des Verarbeitungssystems

- **Bilddigitalisierer:**  
Es sind zwei Matrox Meteor II Digitalisierer installiert, von denen jeder an eine der Kameras angeschlossen ist. Hier werden die aufgenommenen Bilder bis zur Auswertung zwischengelagert.
- **Bildverarbeitungsbibliothek:**  
Zur Bildverarbeitung wird die *Matrox Imaging Library* eingesetzt [Matr99a, Matr99b], die als hardwareunabhängige Bibliothek eine Vielzahl von Routinen bereitstellt.

## 2.3 Einordnung dieser Arbeit in das Projekt

Die vorliegende Arbeit widmet sich dem visuellen Teilbereich des Projekts *Morpha*, der die pragmatische Interpretation menschlicher Interaktionsmuster und Handlungen zum Ziel hat. Es soll über ein Stereo-Sichtsystem menschliche Handbewegungen interpretieren, so dass im Folgenden diesen Anweisungen

entsprochen werden kann. Zur Interpretation dieser Handbewegungen ist zum Einen die Verfolgung der Hand über Maßnahmen der Bildverarbeitung zu realisieren. Weiterhin muss die Bedeutung der Gesten erkannt werden. Zu diesem Zweck wird ein geeignetes Modell eingeführt, das Training und Erkennung der Gesten ermöglicht. Die hierzu verwendeten Hidden-Markov-Modelle und ihre Anwendung bilden den Schwerpunkt dieser Arbeit und werden im Folgenden eingehend betrachtet.

## **2.4 *Stand der Technik***

### **2.4.1 *Beobachtung und Szenenanalyse***

Die Aufgabenstellung dieser Arbeit erfordert Lösungen aus zwei Bereichen des *Morpha* Projektes. Es handelt sich dabei um die dynamische Situationserkennung und die visuelle Kommandierung. Die Szenenanalyse lässt sich weiter unterteilen in Objektlokalisierung und Objektklassifikation. Konkret bedeutet dies das Finden der Hand in den Eingabebildern sowie die Einordnung der dadurch gegebenen Informationen.

Die Verfolgung von Objekten und Erfassung realer Koordinaten ist in der Robotik immer dann nötig, wenn Applikationen in ihre Umwelt eingreifen. Dieser Bereich wird bereits seit den 80er Jahren bearbeitet [AlWB88] und beschäftigt sich schwerpunktmäßig mit Maßnahmen zur Kamerakalibrierung und der Rekonstruktion von Raumpunkten aus stereoskopischen Aufnahmen [Ude96, Faug93]. Grundlegende Erkenntnisse in der Bildverarbeitung und Merkmalsextraktion wurden durch [Hara80, MaHi80, Cann86] gewonnen.

Die visuelle Kommandierung beinhaltet die eigentliche Gestenerkennung, die auf der Bildverarbeitung aufsetzt. Häufig verfolgte Ansätze finden sich in der Berechnung des optischen Flusses in Bildern [TrMa97, NoUh96], bei Stereosichtsystemen in der Korrelation der beiden Kamerabilder [ArSa97, ReMu96] und bei der Anwendung von Farbsegmentierungen mit der Hautfarbe [SiKC99]. Verarbeitungsstufen der nächsthöheren Ebene setzen hierauf auf. Dabei werden die gefundenen Trajektorien meistens in einem Phasenraum untersucht [Crow95a, CrBC95]. Alternativ können auch dreidimensionale Modelle der Hand zum Einsatz kommen [ReKa94].

### **2.4.2 *Gestenerkennung***

In diesem Abschnitt wird ein Überblick über vergleichbare Arbeiten auf dem Gebiet der Gestenerkennung gegeben. Die Vergleichbarkeit bezieht sich dabei

auf das Ziel, nämlich die Erkennung menschlicher Gesten. Dies kann die Interpretation von Gesten der Gebärdensprache sein wie in den Abschnitten 2.4.2.1 und 2.4.2.2 oder aber zur Steuerung von Robotern wie in Abschnitt 2.4.2.3 und Abschnitt 2.4.2.4 dienen. Die Arbeiten in den beiden letztgenannten Abschnitten beinhalten das Verfolgen von Handbewegungen und die Erkennung der Geste. Beide behandeln bei einer Geste nicht die Handbewegung, sondern nur die abschließende Fingerstellung und Position der Hand.

Als einzige der hier betrachteten Arbeiten verwendet [StPe95] Hidden–Markov–Modelle zur Gestenerkennung. Im Unterschied zu den anderen wird hier die Bewegung der Hände als Bedeutungsträger angesehen und Fingerstellungen sowie Handpositionen außer Acht gelassen.

Die unterschiedlichen Ansätze werden im Folgenden jeweils kurz skizziert, ihre zugrunde liegende Idee und die verwendete Technik erläutert.

#### **2.4.2.1 Erkennung koreanischer Zeichensprache**

In [KiJB96] wird ein System zur Erkennung der Koreanischen Gebärdensprache vorgestellt. Diese Gebärdensprache ermöglicht die Kommunikation zwischen Taubstummten und besteht aus über 6000 darstellbaren Wörtern, die sich aus einer kleinen Menge von Basisgesten zusammensetzen.

Die Gesten der Gebärdensprache werden dabei aus einem statischen und einem dynamischen Teil gebildet. Der statische Teil beschreibt Handposition und Fingerstellung und unterscheidet 31 verschiedene Möglichkeiten. Die Bewegungen mit beiden Händen stellen den dynamischen Teil und damit den Kern der Gebärdensprache dar. Sie erlauben es, Bedeutungsunterschiede auszudrücken und geben den statischen Gesten unterschiedliche Bedeutungen.

Das System soll die Gebärden erkennen und in koreanischen Text überführen. Es besteht aus zwei Datenhandschuhen, die die Informationen über Handbewegung und Fingerstellung liefern und einem Fuzzy–Min–Max neuronalen Netz, das eine Gestenklassifizierung vornimmt. Dieses Fuzzy–Min–Max neuronale Netz soll kein vorhergehendes Lernen benötigen.

Probleme beim Datenaufkommen bereiten unterschiedliche anatomische Merkmale verschiedener Personen. Unterschiedliche Handgrößen liefern auch unterschiedliche numerische Daten. Darüber hinaus sind die durch den Handschuh erfassten Daten oft stark verunreinigt, bedingt durch Zittern oder bedeutungslose Bewegungen zwischen den Gesten. Weiterhin überlappen sich die Musterklassen, in die die Menge aller Gebärden aufgeteilt ist. Eine Geste ist also nicht eindeutig zuzuordnen. Gesten müssen außerdem unabhängig vom ihrem Startpunkt aus erkannt werden können.

Das in [KiJB96] beschriebene System teilt die durch den Datenhandschuh gelieferten Rohdaten in Regionen ein, so dass aus der resultierenden Menge von Regionendaten eine Richtungsklasse ermittelt werden kann. Der dynamische Teil der Geste, also die Bewegung beider Hände, wird anhand der Richtungsklasse erkannt. Das neuronale Netz wiederum soll den statischen Teil, die Hand- und Fingerposition erkennen.

Gegenwärtig erkennt das System nicht mehr als 85 Prozent der Gesten richtig und ist anfällig gegenüber Sensorfehlern. Probleme bereiten außerdem bedeutungslose Handbewegungen, die beispielsweise dann auftreten, wenn die Hände nach einer Bewegung zurück in die Ruheposition geführt werden.

Für beide Hände steht jeweils ein Datenhandschuh zur Verfügung, der die Messdaten über eine RS232C Schnittstelle mit einer Übertragungsrate von 9600 Baud an eine Sun Sparc Station II sendet.

#### **2.4.2.2 *Erkennung amerikanischer Zeichensprache mit Hidden-Markov-Modellen***

Die Erkennung von amerikanischer Zeichensprache (*American Sign Language*) hat das in [StPe95] beschriebene Projekt zum Ziel. Ausgehend von visuellen Informationen, die durch eine einzelne Kamera gewonnen werden, sollen Sätze der Zeichensprache erkannt werden. Dabei wird die Einschränkung vorgenommen, dass Zeichen, die bestimmte Fingerstellungen und Mimik beinhalten, unberücksichtigt bleiben. Die einzelnen Zeichen der Zeichensprache bestehen aus mit beiden Händen ausgeführten Gesten, so dass auch beide Hände verfolgt werden müssen.

Das vorgestellte System ist echtzeitfähig und versucht nicht, eine genaue Beschreibung der Handform zu erlangen, da für die Verfolgung allgemeinere Informationen ausreichen und Fingerstellungen insgesamt nicht ausgewertet werden. Die Verfolgung liefert demzufolge nur ungefähre Informationen über die Form der Hand, die Orientierung und die Trajektorie. Diese Informationen bilden auch die Eingabeparameter für das Hidden-Markov-Modell. Die Handverfolgung geschieht auf Grundlage von Farbinformationen. Das wird im ersten Versuch über farbige Handschuhe und im zweiten über die Hautfarbe realisiert.

Um die Hände im Bild zu finden, überprüft ein Suchalgorithmus alle Pixel. Sobald ein hautfarbener Pixel beziehungsweise einer in der Farbe des Handschuhs entdeckt wird, dehnt sich die Suche auf seine acht Nachbarn aus. Gleichfarbige benachbarte Pixel werden als einer Hand zugehörig betrachtet. Sofern Handschuhe zum Einsatz kommen, haben der linke und der rechte Handschuh unterschiedliche Farben, im anderen Fall wird der linke Trefferbereich im Bild der rechten Hand zugeordnet und der rechte Bereich der linken Hand.

Zweideutigkeiten entstehen, wenn sich beide Hände oder Hände und Gesicht überlappen. Da die zeichengebende Person in dieser Arbeit bis auf die Handbewegungen unbeweglich bleibt, verharrt das Gesicht im selben Bereich und kann so erkannt werden. Bei der Überlagerung zweier Hände erscheinen diese als ein einziger großer Bereich im Bild. Jeder Hand wird dann die Position dieses Bereichs zugewiesen. Trotz zahlreicher Überlagerungen können die in dieser Arbeit trainierten Gesten erkannt werden.

Obwohl aufgrund von schwankenden Lichtverhältnissen die Handverfolgung stellenweise stockt, wird bei der Benutzung der Handschuhe eine Erkennungsrate von 91,3 Prozent erreicht. Wird zusätzlich noch eine Grammatik benutzt, so steigert sich die Rate auf 99,2 Prozent. Die Grammatik schränkt die Möglichkeiten zur Satzbildung ein und erlaubt nur Sätze der Form *Personalpronomen, Verb, Nomen, Adjektiv, Personalpronomen*.

Werden die Handschuhe als Hilfsmittel weggelassen, so ergibt sich eine Erkennungsquote von 84,7 Prozent mit Grammatik und 74,5 Prozent ohne Grammatik. Problematisch sind hier schon leichte Drehungen des Körpers und veränderte Positionen der Hände im Verhältnis zum Körper. Letzteres verändert die Größe der Hand im aufgenommenen Bild und beeinflusst die Folge absoluter Koordinaten. Sobald allerdings die Handfläche und die relative Verschiebung in  $x$ - und  $y$ -Richtung zwischen aufeinanderfolgenden Bildern als zusätzliche Parameter in die Auswertung mit aufgenommen werden, verbessert sich das Resultat deutlich. Es wird dann eine Erfolgsquote von 91,9 Prozent erreicht.

Die allgemeine Einsatzfähigkeit ist dadurch beschränkt, dass die Testperson auf einem Stuhl vor der Kamera sitzen muss und sich möglichst nicht bewegen darf, sowie durch die Tatsache, dass, sobald der Erkennungsvorgang begonnen hat, nur gültige Gesten der amerikanischen Zeichensprache ausgeführt werden dürfen, da das System bedeutungslose Gesten nicht herausfiltern kann.

In Bezug auf die Echtzeitfähigkeit ist anzumerken, dass eine Rate von fünf Bildern pro Sekunde bei Verwendung einer Silicon Graphics Indigo 2 erreicht wird. Die Auflösung beträgt  $320 \times 243$  Pixel. Bei Verlust der zu verfolgenden Hand wird diese zwar in der Regel nach dem nächsten Bild wiedergefunden, das vorhergehende Bild aber gelöscht, so dass sich keine kontinuierliche Datenrate ergibt.

### **2.4.2.3 Visuelle Gestenerkennung durch Roboter**

Das in [TrMa97] vorgestellte System zur Gestenerkennung basiert auf der Auswertung visueller Informationen, die durch ein Stereo-Kamerasystem gewonnen werden. Dadurch soll dem Benutzer das Tragen von Spezialkleidung, etwa

Datenhandschuhen, erspart bleiben. Es macht sich dabei Unterschiede in Bewegung, Farbe und den Stereobildern zunutze.

In Bezug auf die Bewegung von Objekten werden Differenzbilder aufeinander folgender Bilder hinsichtlich der Intensitätskomponente ermittelt. Die Bewegungsermittlung reagiert allerdings auch auf alle verschobenen Regionen, also nicht nur auf die zu verfolgende Hand, sondern auch auf den Schatten der Hand, der auf stationäre Objekte fällt. Darüber hinaus führt eine Bewegung des Kamerakopfes zu Verwirrung, da eine solche Verschiebungen im gesamten Bild verursacht.

Zur Ermittlung der Hautfarbe werden die Farb- und die Sättigungskomponenten eines Bildes ausgewertet. Bei veränderten Lichtverhältnissen muss das Modell für die Hautfarbe angepasst werden, um den neuen Gegebenheiten Rechnung zu tragen. Bevor das System in Aktion tritt, wird es gemäß der aktuellen Lichtverhältnisse ausgerichtet und auf die spezielle Hautfarbe der zu verfolgenden Hand eingestellt. Nachteilig wirkt sich dabei aus, dass eine Rekalibrierung im laufenden Betrieb nicht möglich ist.

Für jede der beiden Kameras werden sogenannte *attention maps* berechnet, die Informationen über Bewegungs- und Farbveränderungen zusammenführen. Hierbei wird der Farbfaktor übergewichtet, da sich die Hand nicht immer bewegt und deswegen Bewegungsänderungen nicht immer vorliegen. Die gemeinsame Auswertung der beiden Faktoren soll sicherstellen, dass das System auch bei Ausfall einer der beiden Informationsquellen trotzdem weiterhin funktionstüchtig bleibt.

Im weiteren Verlauf werden die *attention maps* beider Kameras addiert, was dazu führt, dass ausschließlich von beiden Kameras fixierte Objekte deutliche Rückgabewerte liefern.

Die aktive Verfolgung des Zielobjekts wird auf Basis von drei Differenzbildern realisiert, die durch Addition der *attention maps* der beiden Kameras entstehen. Diese werden einmal mit negativer, einmal mit positiver und einmal ohne Verschiebung addiert. Die Suche nach globalen Maxima in diesen Bildern liefert dann Informationen darüber, wie die Kameras im Verhältnis zueinander bewegt werden müssen und ob das Objekt das aktuelle Sichtfeld verlässt. Im letzten Fall werden die Kameras bewegt, um das Objekt wieder ins Zentrum zu schieben. Während dieser Bewegungen stoppt die Bilderfassung.

Das System erlaubt die gleichzeitige Verfolgung mehrerer Objekte, zum Beispiel der beiden Hände und des Kopfes. Zu diesem Zweck wird im aktuellen Bild an Positionen gesucht, an denen sich im vorherigen die betreffenden Objekte befanden. Diese Technik scheitert allerdings, sobald sich zwei Objekte überlappen, da sie anschließend nicht mehr korrekt zugeordnet werden können.

Die abschließende Erkennung von Hand- und Fingerstellungen wird mit dem *Elastischen-Graphen-Vergleich* gelöst. Die zu verfolgenden Objekte im Bild werden dabei jeweils durch einen Graphen repräsentiert, dessen Knoten lokale Bildinformationen tragen und dessen Kanten mit geometrischen Informationen versehen sind. Ein solcher Referenzgraph wird zur Erkennung solange über das Bild bewegt, bis jeder Knoten in eine Region zu liegen kommt, in die er am besten passt. Bei der Suche werden der Reihe nach alle Graphen auf das Bild angewendet und der am besten passende bildet den Treffer.

Das System erreicht bei zehn möglichen dreidimensionalen Gesten eine Erkennungsquote von 86 Prozent. Ein solcher Erkennungsvorgang benötigt allerdings 16 Sekunden auf einer Sun Ultra Sparc Workstation. Bei nur zehn zur Auswahl stehenden Gesten kann die Erkennung um den Faktor drei bis fünf beschleunigt werden. Diese Auswertungen sind auf Grauwertbilder beschränkt. Die Anwendung auf farbige Bilder macht die Zuhilfenahme von Farbinformationen bei der Auswertung möglich und steigert den Erkennungserfolg von 54 Prozent auf 70 Prozent bei einem Versuch, dessen Graphen nur auf Basis eines einzigen Bildes gebildet wurden.

#### **2.4.2.4 Gestenerkennung und Handverfolgung mittels aktiver Konturen**

Der Ansatz in [HeSa95] setzt die Technik der *aktiven Konturen* ein, um die Verfolgung der Hand zu realisieren. Wie bei dem System in [TrMa97], das in Abschnitt 2.4.2.3 beschrieben wird, handelt es sich hierbei um ein visuelles System, das jedoch nur eine Kamera einsetzt.

Die Handverfolgung beginnt mit einer globalen Suche auf den visuellen Eingangsdaten, um die gewünschte Hand zu finden. Hierzu wird ein genetischer Algorithmus eingesetzt, da der erfolgreiche Einsatz aktiver Konturen wesentlich von einer guten Startabschätzung abhängig ist. Ist die Hand im Bild gefunden, wird eine Kontur, die grob der Form entspricht, um die Hand gelegt. Die Kontur wird von in der Nähe liegenden Kanten angezogen, und nähert sich so der Hand an, bis sie diese exakt umschließt. Der iterative Prozess läuft in kleinen Schritten ab.

Diese Technik wird hier dadurch erweitert, dass die Hand über eine Serie von Bildern verfolgt wird. Hierbei wird die Position der Hand in einem Bild als Startabschätzung für die Handposition im nächsten Bild gewählt.

Das eingeführte *Point Distribution Model* (PDM) soll zwei wesentliche Probleme beim Einsatz aktiver Konturen lösen. Zum Einen sollen unerwünschte Verformungen der Kontur verboten und zum anderen die Vielzahl möglicher Änderungen in der Handposition trotzdem erfasst werden können.

Das Point Distribution Model ist Ergebnis eines Trainingsvorgangs basierend auf statistischer Analyse und ermittelt allgemeine Bewegungsarten und eine durchschnittliche Form der Hand.

Die Gestenerkennung soll auf Basis einiger weniger charakteristischer Parameter erfolgen, die Teil des PDM sind. Da aufgrund des Trainings etwa bekannt ist, wie diese Parameter mit bestimmten Gesten korrespondieren, kann eine Zuordnung erfolgen. Alternativ wird ein neuronales Netz zur Klassifizierung und Erkennung empfohlen, falls die Zusammenhänge nicht eindeutig sind.

Die wesentliche Einschränkung dieses Erkennungssystems besteht darin, dass nur eine geöffnete Hand verfolgt werden kann und dass, um ausreichende Lichtverhältnisse zu erreichen, zwei zusätzliche Lichtquellen aufgestellt werden müssen, die mittels Reflektoren ein diffuses Licht auf die zu verfolgende Hand werfen. Der Einsatz nur einer Kamera beschränkt die Nutzungsmöglichkeiten auf zweidimensionale Anwendungen.

Der Echtzeitcharakter dieser Anwendung liegt in der Fähigkeit, mindestens 25 Bilder pro Sekunde verarbeiten zu können. Gearbeitet wird dabei mit einer Digital Equipment Alpha Unix Workstation. Die 15 Bit-Farbbilder werden über eine von mehreren Kameras geliefert, die über ein ATM-Netzwerk angeschlossen sind. Ein zu schnelles Bewegen der Hand führt dazu, dass der Algorithmus die Hand verliert. Nachteilig wirkt sich weiterhin aus, dass die Kamera der Handbewegung nicht folgt. Die Geste muss sich also über den gesamten Erkennungszeitraum im Erfassungsbereich der Kamera befinden.



## **3 Technische Grundlagen**

Dieses Kapitel soll einen Überblick über die technischen Grundlagen geben, die zum Einsatz kommen, und ein grundlegendes Verständnis für die Lösungsidee und ihre Umsetzung schaffen. Es werden außerdem Fachbegriffe erklärt, die im weiteren Verlauf auftreten.

Der Beschreibung der Hidden-Markov-Modelle, denen in dieser Arbeit eine zentrale Bedeutung zukommt, geht eine ausschnittsweise Einführung in die Bildverarbeitung im Rahmen der erforderlichen Bereiche voraus.

Anschließend werden Grundlagen über Hidden-Markov-Modelle vermittelt. Aufbauend auf dem Wissen über Arten und Struktur dieser Modelle werden zwei wesentliche Algorithmen vorgestellt und ihre Anwendung erklärt.

Für eine detailliertere und umfassendere Beschreibung dieser Themenkomplexe muss auf die Fachliteratur verwiesen werden, da im Rahmen dieser Arbeit nur ein Einblick in die für ein Gesamtverständnis wesentlichen Bereiche gegeben werden kann.

Die Grundlagen der Bildverarbeitung werden in [JaKS95] behandelt. Der spezielle Aspekt der Hautfarbsegmentierung wird in [YaLW97] eingeführt. Eine umfassende Einführung in Hidden-Markov-Modelle zusammen mit ihren Algorithmen liefert [Rabi89] am Beispiel der Spracherkennung. Implementierungsbeispiele für Hidden-Markov-Modelle unterschiedlichen Umfangs enthalten [Aibi99] und [Kanu98].

### **3.1 Bildverarbeitung**

Maßnahmen aus der Bildverarbeitung kommen in dieser Arbeit zum Einsatz, um die gewünschten Informationen aus dem eingehenden Bilderstrom zu extrahieren und die Hidden-Markov-Modelle mit Eingabedaten zu versorgen. Demzufolge sind diese Methoden der Nutzung der Hidden-Markov-Modelle vorgeschaltet. Die von dem Kamerakopf akquirierten Bildsequenzen müssen in eine

Form gebracht werden, die eine weitere Verarbeitung ermöglicht und außerdem die Extraktion der menschlichen Gesten aus diesen Bildern in einer für die Hidden-Markov-Modelle verständlichen Form erlaubt.

Die charakteristischen Farbeigenschaften der Haut erlauben ein Finden und Verfolgen der zu beobachtenden Hand, wie es in Abschnitt 4.2 dargestellt wird. Um jedoch von diesen speziellen Eigenschaften der Hautfarbe profitieren zu können, sind vorausgehende Verarbeitungsschritte nötig, die zum Beispiel in Form von Farbraumkonvertierung und Filterung im Folgenden kurz skizziert werden.

### 3.1.1 Farbräume und Farbraumkonvertierung

Da sich die Verarbeitung der vom Kamerakopf aufgenommenen Bilder die charakteristischen Eigenschaften der Hautfarbe zunutze macht, muss die Tatsache beachtet werden, dass es sich bei Farbe nicht um eine physikalische Eigenschaft handelt. Es ist vielmehr ein wahrnehmungsabhängiges Phänomen, das von den Spektraleigenschaften elektromagnetischer Strahlung abhängt, die im sichtbaren Wellenlängenbereich auf das Auge auftreffen [WySt82].

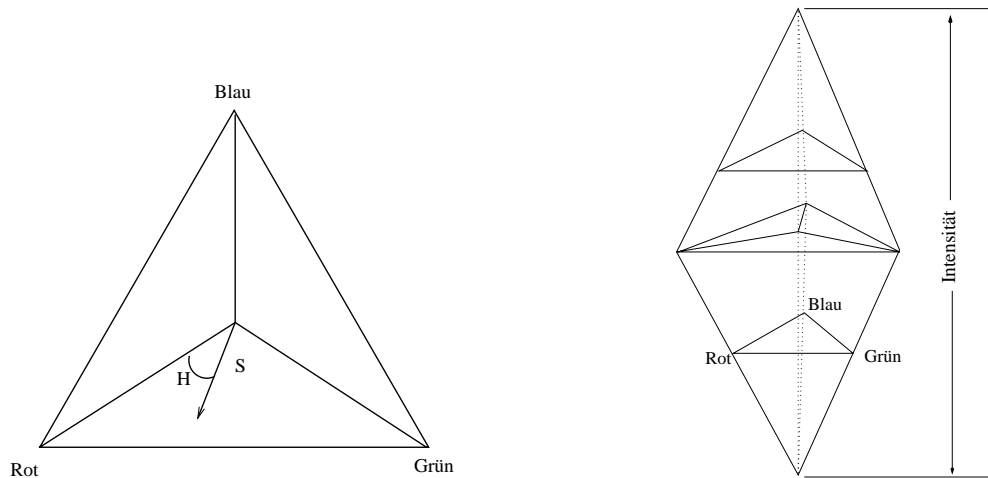
Die Darstellung von Bildern kann in verschiedenen sogenannten Farbräumen erfolgen. Gängige Farbräume sind zum Beispiel RGB und HSI. RGB bezeichnet die Primärfarben rot, grün und blau, so dass die Farbe eines jeden Pixels durch seinen anteiligen Rotwert, Grünwert und Blauwert beschrieben werden kann. HSI wiederum spezifiziert einen Pixel durch Farbton (*Hue*), Sättigung (*Saturation*) und Helligkeit (*Intensity*). Den Zusammenhang zwischen RGB und HSI verdeutlicht Abbildung 3.1.

Ausgehend von diesen beiden Darstellungen lassen sich in der einen Form dargestellte Bilder in die jeweils andere umsetzen. Die in dieser Arbeit eingesetzte  $\text{RGB} \Rightarrow \text{HSI}$  Transformation wird durch die folgende komponentenweise Umrechnung durchgeführt [JaKS95]:

$$I = \frac{1}{3}(R + G + B) \quad (3.1)$$

$$\cos H = \frac{2R - G - B}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \quad (3.2)$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \quad (3.3)$$



(a) Das HSI-Farbdreieck zeigt, wie aus den Grundfarben neue Farbwerte kombiniert werden können.

(b) Die Anzahl der Farben, die als Kombination der Primärfarben gebildet werden können, ist beschränkt.

Abbildung 3.1: Beziehung zwischen den Farbräumen RGB und HSI

Die Farbraumkonvertierung von RGB nach HSI wird verwendet, um gezielt auf die Eigenschaften des Farbtons (Parameter H) zugreifen zu können. Die restlichen Informationen (Parameter S und I) können dann herausgefiltert werden, so dass nur die zur Handverfolgung benötigten Farbinformationen bleiben.

### 3.1.2 Filter

Filter sind in der Bildverarbeitung von Bedeutung, um Bilder im Hinblick auf gewisse Zielvorstellungen hin zu manipulieren. Dies kann zum Beispiel bedeuten, dass Kanten hervorgehoben oder geglättet, Störungen beseitigt und morphologische Operationen durchgeführt werden.

Bei den zum Finden und Verfolgen der Hand in den Bildsequenzen genutzten Filtern handelt es sich um sogenannte *Opening*- und *Closing*-Operationen. Beide Filter wirken dergestalt, dass sie jeden Bildpunkt mit seinen Nachbarn vergleichen. Die Aufgabe des *Opening*-Filters ist es, einzelne Pixel zu eliminieren. Werden also einzelne, isoliert vorkommende Pixel gefunden, so werden sie gelöscht.

Der *Closing*-Filter dient dazu, Löcher und Risse in zusammenhängenden Objekten im Bild zu stopfen. Ist ein nicht gesetzter Bildpunkt ausschließlich von gesetzten Nachbarn umgeben, so erhält er den Farbwert eben dieser Nachbarn.

Abbildung 3.2 veranschaulicht diesen Vorgang.

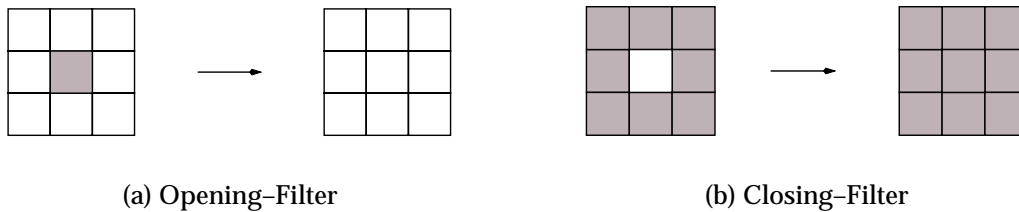


Abbildung 3.2: Bei der Bildverarbeitung eingesetzte Filter

Die beiden vorgestellten Filter werden benötigt, da spätere Operationen zusammenhängende Bildobjekte untersuchen. Hierbei sind sowohl Löcher als auch isolierte Pixel hinderlich.

## 3.2 Hidden-Markov-Modelle

Obwohl *Hidden-Markov-Modelle* (HMM) schon seit Ende der 60er Jahre bekannt sind und auch im Bereich der Spracherkennung zahlreich eingesetzt werden, haben sie erst in den letzten Jahren auch in anderen Bereichen Anwendung gefunden. Ein typisches Beispiel ist die Handschriftenerkennung, die sich die Hidden-Markov-Modelle in zunehmendem Maße zu Nutze macht. Seit der Einsatz von Hidden-Markov-Modellen zur Gestenerkennung entdeckt wurde, sind auch auf diesem Gebiet einige grundlegende Arbeiten vorgenommen worden [RyNu93b, StPe95, LeKi99].

Nach einer allgemeinen Einführung werden in diesem Abschnitt kurz diskrete Markov-Prozesse skizziert, dann ihre Erweiterung auf Hidden-Markov-Modelle ausgeführt und diese im Detail vorgestellt.

Der Reichtum an mathematischen Strukturen, der den Hidden-Markov-Modellen zu eigen ist, lässt sie zu einer möglichen Grundlage für zahlreiche Anwendungen werden, wie die Bereiche Sprach-, Handschrift- und Gestenerkennung zeigen. Sie ermöglichen die Modellierung räumlicher und zeitlicher Informationen auf natürliche Art und Weise und bilden so ein Modell für Zeitfolgen mit variierenden räumlich-zeitlichen Eigenschaften, ohne dass eine explizite zeitliche Ausrichtung durchgeführt werden muss [RyNu93b]. Die Existenz von effizienten Algorithmen (Abschnitte 3.2.6 und 3.2.7) zum Lernen und zur Analyse stellt ein weiteres Argument für ihren Einsatz dar.

### 3.2.1 Diskrete Markov-Prozesse

Ein beobachtbares *Markov-Modell* beschreibt ein System bestehend aus einer Menge von  $N$  unterschiedlichen Zuständen  $s_1, s_2, \dots, s_N$  und ihren Übergängen (Abbildung 3.3).

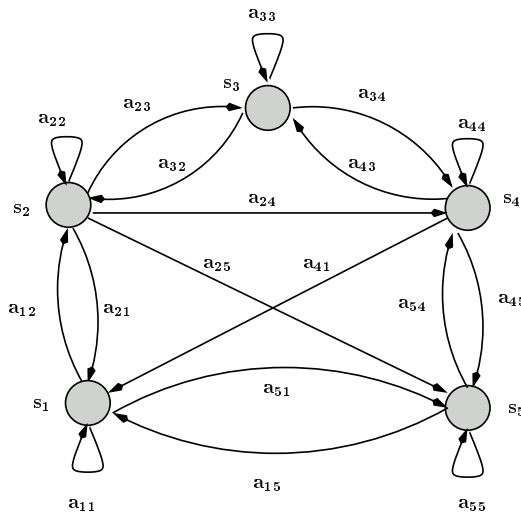


Abbildung 3.3: Markov-Modell mit den vier Zuständen  $s_1$  bis  $s_4$  und ausgewählten Übergängen

Zu diskreten aufeinander folgenden Zeitpunkten findet ein Zustandswechsel gemäß den Wahrscheinlichkeiten statt, die den Zuständen zugeordnet sind. Der Zustand des Modells zum Zeitpunkt  $t$ ,  $t = 1, 2, \dots$  wird als  $q_t$  bezeichnet. Eine vollständige Beschreibung des Modells zum Zeitpunkt  $t$  macht die Bezeichnung des aktuellen Zustands sowie aller Vorgängerzustände nötig. Im diskreten Fall einer Markov-Kette erster Ordnung kann diese Beschreibung auf den aktuellen sowie einen Vorgängerzustand reduziert werden [Rabi89]:

$$P(q_t = s_j \mid q_{t-1} = s_i, q_{t-2} = s_k, \dots) \quad (3.4)$$

$$= P(q_t = s_j \mid q_{t-1} = s_i). \quad (3.5)$$

Dies wird durch die Markov-Eigenschaft ermöglicht, die nur dem Vorgängerzustand Bedeutung zubilligt. Alle weiter zurückliegenden Zustände sind für den aktuellen irrelevant.

Eine Betrachtung, die nur Prozesse aus (3.5) zulässt, die wiederum zeitunabhängig sind, führt zu einer Menge von Zustandsübergangswahrscheinlichkeiten  $a_{ij}$ :

$$a_{ij} = P(q_t = s_i \mid q_{t-1} = s_j) \quad 1 \leq i, j \leq N. \quad (3.6)$$

Für die einzelnen Übergangswahrscheinlichkeiten gilt dabei:

$$a_{ij} \geq 0 \quad \text{und} \quad (3.7)$$

$$\sum_{j=1}^N a_{ij} = 1. \quad (3.8)$$

### 3.2.2 Struktur und Elemente eines Hidden-Markov-Modells

Wie ein diskreter Markov-Prozess besteht auch ein Hidden-Markov-Modell aus einer Menge von Zuständen, die durch Transitionen verbunden sind. Es unterscheidet sich jedoch von einfachen Markov-Modellen dadurch, dass jedem Zustand ein physikalisches Ereignis oder eine Beobachtung zugeordnet werden kann. Im Fall eines Hidden-Markov-Modells stellt die Beobachtung eine stochastische Funktion in Abhängigkeit des Zustandes dar. Bei dem resultierenden Modell handelt es sich also um einen doppelten stochastischen Prozess, von dem der zugrunde liegende nicht beobachtbar ist. Er kann nur durch eine zweite Menge stochastischer Prozesse wahrgenommen werden, die eine Folge von Beobachtungssymbolen erzeugen. Im Unterschied hierzu ist beim Markov-Modell der zugrunde liegende Prozess bekannt, also alle denkbaren Zustände und ihre Übergänge des nachzubildenden Signals. Betrachtet man den Bereich Kommunikation, in dem Markov-Modelle ein breites Anwendungsspektrum gefunden haben [Forn72], so bedeutet dies, dass nur der Beobachtungsprozess stochastischer Natur ist, weil er durch Störungen während der Beobachtung beeinflusst wird. Welches Signal in welchem Zustand erzeugt wird, ist jedoch bekannt.

#### 3.2.2.1 Struktur eines Hidden-Markov-Modells

Ein Hidden-Markov-Modell kann durch einen endlichen Automaten repräsentiert werden, der sich wiederum in Form eines Graphen oder eines Gitterdiagramms darstellen lässt (Abbildungen 3.4 und 3.5). Ein Zustand im Gitterdiagramm entspricht einem Zustand im Hidden-Markov-Modell zu einem bestimmten Zeitpunkt. Die Äste repräsentieren die Zustandsübergänge von einem diskreten Zeitpunkt zum nächsten.

Die tatsächliche Struktur des endlichen Automaten, der durch das Hidden-Markov-Modell modelliert wird, bleibt versteckt, so dass die Struktur des Modells allgemein genug gewählt werden muss, um alle möglichen Zustände zu berücksichtigen. Die Zustände und Zustandsübergänge des Hidden-Markov-Modells bilden dabei nur Schätzungen des eigentlichen endlichen Automaten. Jedem Zustandsübergang und jedem Ausgabesymbol sind dabei eine Menge von Wahrscheinlichkeitswerten zugeordnet. Mit ihrer Hilfe kann bestimmt werden, mit welcher Wahrscheinlichkeit die Beobachtungen durch das Modell erzeugt wurden, welches das Hidden-Markov-Modell repräsentiert [RyNu93b].

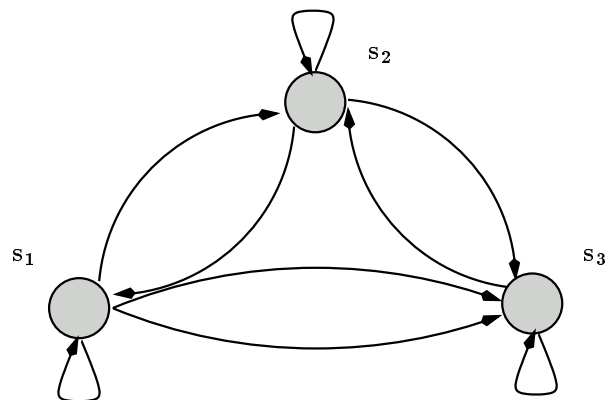


Abbildung 3.4: Endlicher Automat eines Hidden-Markov-Modells mit drei Zuständen

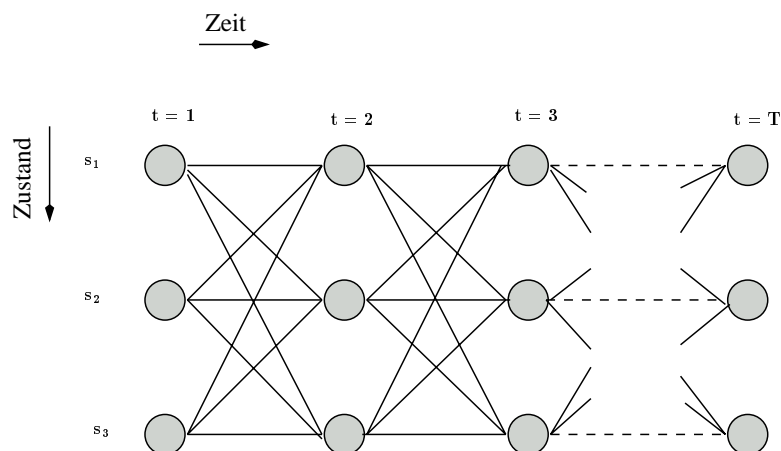


Abbildung 3.5: Gitterdiagramm eines Hidden-Markov-Modells mit drei Zuständen

### 3.2.2.2 Elemente eines Hidden-Markov-Modells

Formal wird ein Hidden-Markov-Modell durch die folgenden Elemente charakterisiert:

1. Eine Menge  $S$  von  $N$  Zuständen.

$$S = \{s_1, s_2, s_3, \dots, s_N\}$$

Der Zustand zum Zeitpunkt  $t$  wird dabei als  $q_t$  bezeichnet.

2. Eine Menge  $V$  von  $M$  paarweise verschiedenen Beobachtungssymbolen, die das diskrete Alphabet bilden.

$$V = \{v_1, v_2, v_3, \dots, v_M\}$$

Die Beobachtung zum Zeitpunkt  $t$  wird durch  $O_t$  ausgedrückt. Die Beobachtungssymbole entsprechen dabei denen des physikalischen Systems, das modelliert wird.

3. Eine Wahrscheinlichkeitsverteilung  $A = \{a_{ij}\}$  der Zustandsübergänge, wobei  $A$  eine reelle  $N \times N$  Matrix ist und  $a_{ij}$  die Wahrscheinlichkeit des Übergangs vom Zustand  $s_i$  nach  $s_j$  bezeichnet.

$$a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i), \quad 1 \leq i, j \leq N$$

Ist wie im Fall des ergodischen Modells (Abschnitt 3.2.3) jeder Zustand von jedem anderen aus in einem einzigen Schritt erreichbar, so gilt  $a_{ij} > 0$  für alle  $i, j$ .

4. Eine Wahrscheinlichkeitsverteilung  $B = \{b_j(k)\}$  der Beobachtungssymbole, wobei  $b_j(k)$  die Wahrscheinlichkeit bezeichnet, dass im Zustand  $s_j$  zum Zeitpunkt  $t$  das Symbol  $v_k$  auftritt.

$$b_j(k) = P(O_t = v_k \mid q_t = s_j), \quad 1 \leq j \leq N \text{ und } 1 \leq k \leq M$$

5. Die anfängliche Zustandsverteilung  $\pi = \{\pi_i\}$ , wobei  $\pi_i$  die Wahrscheinlichkeit angibt, mit der Zustand  $s_i$  der Initialzustand ist.

$$\pi_i = P(q_1 = s_i), \quad 1 \leq i \leq N$$

Außerdem gelten für die Wahrscheinlichkeitsvariablen  $A$ ,  $B$  und  $\pi$  die folgenden Bedingungen:

$$\sum_{j=1}^N a_{ij} = 1 \quad \forall i \quad \text{und} \quad a_{ij} \geq 0, \quad (3.9)$$

$$\sum_{k=1}^M b_j(k) = 1 \quad \forall j \quad \text{und} \quad b_j(k) \geq 0, \quad (3.10)$$

$$\sum_{i=1}^N \pi_i = 1 \quad \text{und} \quad \pi_i \geq 0. \quad (3.11)$$

Zur vollständigen Charakterisierung eines Hidden-Markov-Modells werden also die beiden Modellparameter  $M$  und  $N$  benötigt, die Beobachtungssymbole sowie die Wahrscheinlichkeitsvariablen  $A$ ,  $B$  und  $\pi$ . Zusammenfassend lässt sich ein vollständiges Hidden-Markov-Modell schreiben als

$$\lambda = (A, B, \pi).$$

### 3.2.3 Arten von Hidden-Markov-Modellen

Bei der Klassifizierung von Hidden-Markov-Modellen unterscheidet man mehrere Arten, die im Folgenden kurz dargestellt werden. Dabei wird auf die ersten beiden näher eingegangen, da sie für diese Arbeit von besonderer Bedeutung sind.

1. *Ergodische (ergodic)* Hidden-Markov-Modelle
2. *Links-Rechts (left-right)* Hidden-Markov-Modelle
3. *Autoregressive (autoregressive)* Hidden-Markov-Modelle
4. Hidden-Markov-Modelle mit kontinuierlichen Ausgabesymbolen

Als Abwandlung ist ferner denkbar, die Beobachtungssymbole nicht den Zuständen, sondern vielmehr den Zustandsübergängen zuzuordnen [BJM83]. Darüberhinaus können Zustandsübergänge eingeführt werden, die kein Beobachtungssymbol erzeugen, sogenannte *Nulltransitionen*.

Das ergodische Modell zeichnet sich dadurch aus, dass jeder Zustand von jedem anderen aus in einem Schritt erreichbar ist. Formal bedeutet dies, dass alle

Koeffizienten  $a_{ij} > 0$  sind. Für das in Abbildung 3.4 dargestellte ergodische Hidden-Markov-Modell ergibt sich eine Matrix  $A$  mit der folgenden Struktur:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (3.12)$$

Das in Abbildung 3.6 dargestellte Links-Rechts-Hidden-Markov-Modell hat die Eigenschaft, dass sich der Zustandsindex mit der Zeit erhöht beziehungsweise gleich bleibt. Die Zustände werden dem gemäß von links nach rechts durchschritten. Dies ist auch die definierende Eigenschaft, die den Links-Rechts-Modellen ihren Namen gibt.

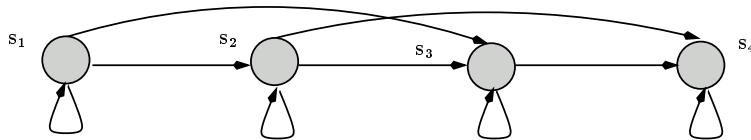


Abbildung 3.6: Links-Rechts-Hidden-Markov-Modell mit vier Zuständen und einer Sprungbegrenzung von  $\Delta = 2$

Ein solches Links-Rechts-Modell ist demzufolge besonders für die Modellierung von Signalen geeignet, deren Eigenschaften sich über die Zeit verändern. Formal hat es die folgende Eigenschaft:

$$a_{ij} = 0, \quad i < j. \quad (3.13)$$

Sie besagt, dass Rückschritte verboten sind, also kein Übergang in einen Zustand stattfinden darf, dessen Index kleiner als der aktuelle ist. Abbildung 3.7 verdeutlicht diesen Zusammenhang anhand des zugehörigen Gitterdiagramms.

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

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (3.14)$$

Möchte man zu große Sprünge bei Übergängen verhindern, so kann eine weitere Beschränkung eingeführt werden:

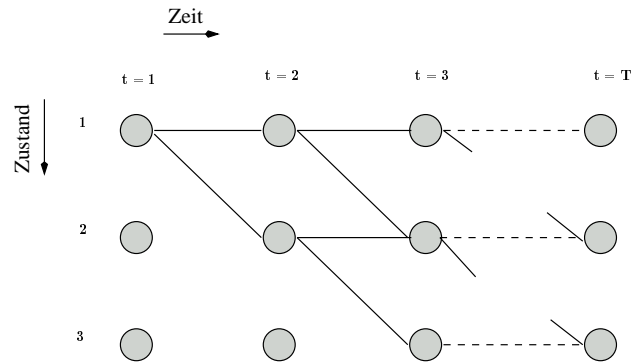


Abbildung 3.7: Gitterdiagramm eines Links-Rechts-Hidden-Markov-Modells mit drei Zuständen

$$a_{ij} = 0, \quad j > i + \Delta. \quad (3.15)$$

Bei  $\Delta$  handelt es sich um eine Sprungbegrenzung, mittels derer sich die Anzahl der übersprungenen Zustände regulieren lässt. In Abbildung 3.6 hat  $\Delta$  den Wert zwei, es können also keine Übergänge über mehr als zwei Zustände stattfinden. Die Übergangsmatrix hätte dann die folgende Gestalt:

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

Bei Links-Rechts-Modellen gilt für den letzten Zustand  $s_N$  immer, dass die Wahrscheinlichkeit eines Übergangs in einen vorherigen Zustand gleich Null ist und gleichzeitig die Wahrscheinlichkeit für eine Selbsttransition gleich 1:

$$a_{NN} = 1, \quad (3.16)$$

$$a_{Ni} = 0, \quad i < N. \quad (3.17)$$

### 3.2.4 Grundlegende Probleme bei der Anwendung von Hidden-Markov-Modellen

Um Hidden-Markov-Modelle, wie sie in den vorigen Abschnitten eingeführt wurden, in Anwendungen nutzbringend einsetzen zu können, sind drei grund-

legende Probleme zu lösen:

1. Zu einem gegebenen Modell  $\lambda = (A, B, \pi)$  und einer Folge von Beobachtungen  $O = O_1, O_2, O_3, \dots, O_T$  muss die Wahrscheinlichkeit  $P(O|\lambda)$  der Zustandsfolge berechnet werden können.
2. Zu einem gegebenen Modell  $\lambda = (A, B, \pi)$  und einer Folge von Beobachtungen  $O = O_1, O_2, O_3, \dots, O_T$  muss eine optimale Zustandsfolge  $Q = q_1, q_2, q_3, \dots, q_T$  gefunden werden. Dabei bedeutet optimal, die Zustandsfolge mit der höchsten Wahrscheinlichkeit zu finden, die  $O$  emittiert hat.
3. Die Parameter eines gegebenen Modells  $\lambda$  müssen so modifiziert werden können, dass  $P(O|\lambda)$  maximal wird.

Möglichkeiten zur Lösung dieser drei Probleme, die auch in dieser Arbeit zur Gestenerkennung bearbeitet werden müssen, sind in den folgenden Abschnitten im Einzelnen dargestellt.

### 3.2.5 Der Vorwärts-Algorithmus

Zur Lösung des ersten der aufgeführten Probleme wird der *Vorwärts-Algorithmus* genutzt.

Der Vorwärts-Algorithmus kommt zum Einsatz, da eine direkte Berechnung, die aus der Betrachtung der Wahrscheinlichkeiten aller möglichen Zustandsfolgen besteht, von exponentiellem Aufwand ist ( $O(2T \cdot N^T)$ ,  $N$  = Anzahl der Zustände und  $T$  = Anzahl der diskreten Zeitpunkte), da  $N^T$  verschiedene Zustandsfolgen existieren und für jede dieser Zustandsfolgen  $2T$  Berechnungen nötig sind [Rabi89]. Der Vorwärts-Algorithmus hingegen weist eine Komplexität von  $O(N^2 \cdot T)$  auf, so dass sich seine Benutzung, insbesondere im Rahmen von Anwendungen unter Echtzeitbedingungen, anbietet.

Der Algorithmus bedient sich einer Hilfsvariablen  $\alpha_t(i)$ , der sogenannten *Vorwärtsvariablen*, die wie folgt definiert ist:

$$\alpha_t(i) := P(O_1, O_2, \dots, O_t, q_t = i \mid \lambda). \quad (3.18)$$

Demzufolge bezeichnet  $\alpha_t(i)$  die Wahrscheinlichkeit der Beobachtungsteilfolge  $O_1, O_2, \dots, O_t$  zum Zeitpunkt  $t$  und im Zustand  $s_i$  bei gegebenem Modell  $\lambda$ .

Beginnend mit

$$\alpha_1(i) := \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (3.19)$$

gilt für die Vorwärtsvariablen

$$\alpha_{t+1}(j) := \left( \sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \text{ und } 1 \leq j \leq N. \quad (3.20)$$

Mit Hilfe dieser schrittweisen Berechnung kann nun  $\alpha_T(j)$ ,  $1 \leq j \leq N$ , bestimmt werden und damit auch die Wahrscheinlichkeit, dass die betrachtete Beobachtungsfolge vom gegebenen Modell erzeugt wurde:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i). \quad (3.21)$$

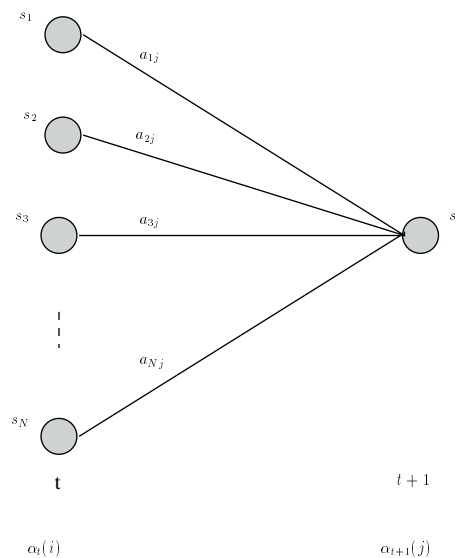


Abbildung 3.8: Schematische Darstellung der Berechnung einer Vorwärtsvariablen

Abbildung 3.8 zeigt, wie der Zustand  $s_j$  zum Zeitpunkt  $t+1$  von  $N$  möglichen Zuständen  $s_i$  zum Zeitpunkt  $t$  aus erreicht werden kann. Die Bedeutungen der Terme in der Vorwärtsvariable können dabei folgendermaßen aufgeschlüsselt werden:

- $\alpha_t(i)$  bezeichnet die gemeinsame Wahrscheinlichkeit, dass  $O_1, O_2, \dots, O_t$  beobachtet wird und der Zustand zum Zeitpunkt  $t$   $s_i$  ist.
- $\alpha_t(i)a_{ij}$  gibt die gemeinsame Wahrscheinlichkeit an, dass  $O_1, O_2, \dots, O_t$  beobachtet wird und der Zustand  $s_j$  zum Zeitpunkt  $t + 1$  über den Zustand  $s_i$  zum Zeitpunkt  $t$  erreicht wird.
- Eine Summierung über alle möglichen  $N$  Zustände  $s_i$  liefert dann die Wahrscheinlichkeit für Zustand  $s_j$  bei  $t + 1$ .

Abbildung 3.9 veranschaulicht die Vorwärtsvariablen auf eine weitere Art und Weise. In der hier gezeigten Gitterstruktur sind nur die erlaubten Übergänge vom Ausgangszustand  $i$  zum Zielzustand  $j$  gezeigt. Jede Spalte aus 3 Zuständen wird zum Zeitpunkt  $t - 1$  vollständig berechnet, bevor zum Zeitpunkt  $t$  die nächste Spalte bearbeitet wird. Ist die Berechnung für die Zustände der letzten Spalte abgeschlossen, so enthält der letzte Zustand in der letzten Spalte die Wahrscheinlichkeit für die Erzeugung der gegebenen Beobachtungssequenz  $O$ .

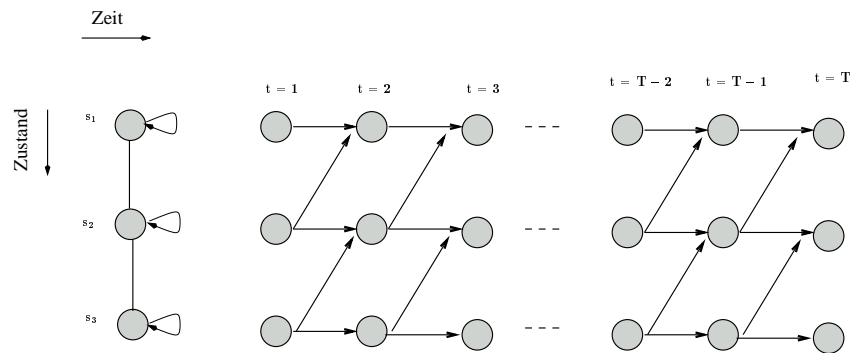


Abbildung 3.9: Schematische Darstellung der Berechnung einer Vorwärtsvariablen mit zugehörigem Hidden-Markov-Modell

Außer in dem hier gezeigten Fall nutzt auch der Baum-Welch-Algorithmus (Abschnitt 3.2.7) die Vorwärtsvariablen. Dort werden auch die in ähnlicher Art und Weise definierten *Rückwärtsvariablen* zum Training von Hidden-Markov-Modellen eingesetzt.

Diese Rückwärtsvariablen sind in Analogie zu den soeben eingeführten Vorwärtsvariablen definiert:

$$\beta_t(i) := P(O_{t+1}, O_{t+2}, \dots, O_T \mid q_t = s_i, \lambda). \quad (3.22)$$

Sie beschreiben die Wahrscheinlichkeit der Zustandsteilfolge vom Zeitpunkt  $t + 1$  bis zum Endzeitpunkt  $T$  bei gegebenem Zustand  $s_i$  und Modell  $\lambda$  zum Zeitpunkt  $t$ .

Ausgehend von

$$\beta_T(i) := 1, \quad 1 \leq i \leq N \quad (3.23)$$

ergeben sich die Rückwärtsvariablen durch

$$\beta_t(i) := \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T - 1. \quad (3.24)$$

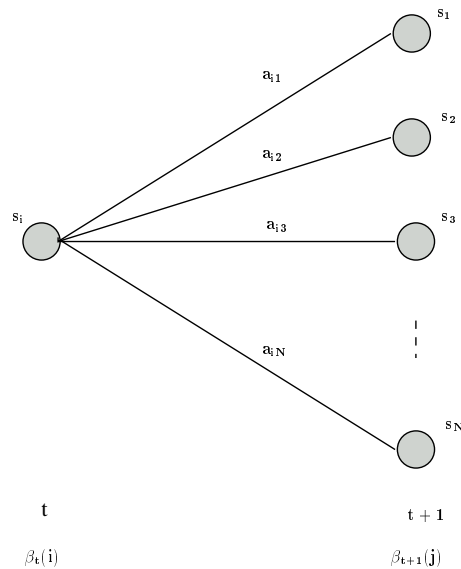


Abbildung 3.10: Schematische Darstellung der Berechnung einer Rückwärtsvariablen

Abbildung 3.10 zeigt, dass alle möglichen Zustände  $s_j$  bei  $t + 1$  betrachtet werden müssen, um zum Zeitpunkt  $t$  in Zustand  $s_i$  gewesen zu sein und die Beobachtungsfolge vom Zeitpunkt  $t + 1$  an erkannt zu haben. Die Bestandteile der Rückwärtsvariablen haben dabei die folgenden Bedeutungen:

- $a_{ij}$  gibt die Wahrscheinlichkeit für den Zustandsübergang  $s_i$  nach  $s_j$  an.

- $b_j(O_{t+1})$  ist die Wahrscheinlichkeit für die Beobachtung des Symbols  $O_{t+1}$  in Zustand  $s_j$ .
- Die Wahrscheinlichkeit für die verbleibende Beobachtungsteilfolge von Zustand  $s_j$  aus wird durch  $\beta_{t+1}(j)$  ausgedrückt.

### 3.2.6 Der Viterbi-Algorithmus

Der *Viterbi-Algorithmus* gibt Antwort auf das zweite grundlegende Problem im Umgang mit Hidden-Markov-Modellen (Abschnitt 3.2.4). Er findet zu einer gegebenen Folge von beobachteten Symbolen diejenige Zustandsfolge, die diese Beobachtungsfolge am wahrscheinlichsten erzeugt hat. Dies ist gleichbedeutend damit, den wahrscheinlichsten Weg durch ein Gitterdiagramm zu finden beziehungsweise den kürzesten Weg durch den Graphen, der das Modell repräsentiert [Forn72, RyNu93a].

Als Kriterium zur Definition des Optimums kann hierbei eine dieser Möglichkeiten angesehen werden:

- Wähle Zustände  $q_t$ , die jeweils für sich am wahrscheinlichsten sind. Hierbei wird die Anzahl der erwarteten korrekten Zustände maximiert. Diese Lösung weist den Nachteil auf, dass das gefundene, vermeintlich optimale Ergebnis keine gültige Zustandsfolge ist. Dies ist darauf zurückzuführen, dass nur die Wahrscheinlichkeit einzelner Zustände untersucht wird, nicht aber die von Zustandsfolgen. Ein für sich genommener wahrscheinlicher Zustand mit Übergangswahrscheinlichkeit  $a_{ij} = 0$  wäre solch ein ungültiges Ergebnis.
- Maximiere die Anzahl korrekter Zustandspaare oder Zustandstripel.
- Finde die wahrscheinlichste Einzelzustandsfolge. Maximiere also dazu  $P(Q|O, \lambda) \Leftrightarrow P(Q, O|\lambda)$ . Der Viterbi-Algorithmus verfolgt genau diesen Ansatz.

Der Viterbi-Algorithmus geht dabei so vor, dass er schrittweise mehrere Ergebnispfade ermittelt, von denen dann abschließend der wahrscheinlichste ausgewählt wird. Formal stellt er sich folgendermaßen dar:

Definiert man

$$\delta_t(i) := \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_t | \lambda), \quad (3.25)$$

so bezeichnet  $\delta_t(i)$  die höchste Wahrscheinlichkeit für einen Pfad, der im Zustand  $s_i$  endet. Daraus folgt:

$$\delta_{t+1}(j) = \max_i (\delta_t(i) a_{ij}) b_j(O_{t+1}), \quad 1 \leq j \leq N. \quad (3.26)$$

Der Viterbi-Algorithmus findet dann gemäß den folgenden Schritten den gewünschten Pfad.

### 1. Initialisierung

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1), \quad 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned}$$

### 2. Ablauf

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}) b_j(o_t) \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \end{aligned}$$

### 3. Schluss

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} (\delta_T(i)) \\ q_T^* &= \arg \max_{1 \leq i \leq N} (\delta_T(i)) \end{aligned}$$

### 4. Pfadsuche (via Backtracking)

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1$$

Anschaulich ergibt sich dieser Ablauf:

1. Untersuche jeden Zustand zum Zeitpunkt  $t$ . Entscheide, welcher der Zustandsübergänge, die in diesen Zustand führen, am wahrscheinlichsten ist. Sind mehrere Übergänge gleich wahrscheinlich, wähle zufällig einen als den wahrscheinlichsten aus.
2. Bilde mit dem soeben bestimmten Wahrscheinlichkeitswert die aktuelle Ergebnisfadewahrscheinlichkeit  $\delta_t(i)$ .

3. Verwerfe alle anderen in diesen Zustand führenden Übergänge und hänge diesen Zustand an den Ergebnispfad des Zustands, von dem der Übergang ausging, bei  $t - 1$  an. Dies wird dann der Ergebnispfad des zum Zeitpunkt  $t$  betrachteten Zustands.  
Führe diese Operation für alle Zustände zum Zeitpunkt  $t$  aus.
4. Gehe weiter zum Zeitpunkt  $t + 1$  und fahre bei Schritt 1 fort.  
Gilt  $t = T$ , bestimme die Ergebnispfade gemäß Schritt eins bis drei und suche von allen Ergebnispfaden denjenigen mit der größten Wahrscheinlichkeit aus. Dieser bildet den endgültigen Ergebnispfad  $q_t^*$  mit der Wahrscheinlichkeit  $P^*$ .

Der Viterbi-Algorithmus wird im Hinblick auf seine eigentliche Aufgabe, der Pfadermittlung, im Rahmen des entwickelten Erkennungssystems nicht eingesetzt. Er steht aber für eine mögliche Endpunkterkennung einer Geste zur Verfügung. Diese Problematik ergibt sich, sobald Gesten erkannt werden sollen, die in einander enthalten sind. Eine sofortige Erkennungsmeldung der Teilgeste würde eine Erkennung der umfassenderen Geste dauerhaft verhindern. Mit dem Viterbi-Algorithmus als Grundlage kann dieses Problem gelöst werden.

Obwohl er in dieser Funktion zur Zeit nicht zur Anwendung kommt, kann der Algorithmus aber zum Beispiel zur Plausibilitätsprüfung verwendet werden. Durch seinen Einsatz lassen sich Informationen darüber erhalten, ob die Referenzmodelle korrekt aufgebaut wurden oder ob hier Nachbesserungsbedarf besteht. Das Design der Referenzmodelle mit dem in Abschnitt 4.4.3 dargestellten Verfahren läßt sich so verifizieren.

### 3.2.6.1 Beispielablauf des Viterbi-Algorithmus

In Anlehnung an [Forn72] werden jetzt die einzelnen Schritte beim Ablauf des Viterbi-Algorithmus anhand eines Beispiels aufgezeigt.

Als Basis dient das in Abbildung 3.11 gezeigte Gitterdiagramm mit vier Zuständen und fünf diskreten Zeitpunkten. Die einzelnen Äste sind mit ihren jeweiligen Längen beschriftet. Danach werden in Abbildung 3.12 die Schritte gezeigt, in denen der Algorithmus den kürzesten Pfad vom Start- bis zum Endzustand bestimmt. Auf jeder Stufe werden die Ergebnispfade mit ihrer jeweiligen Länge gezeigt.

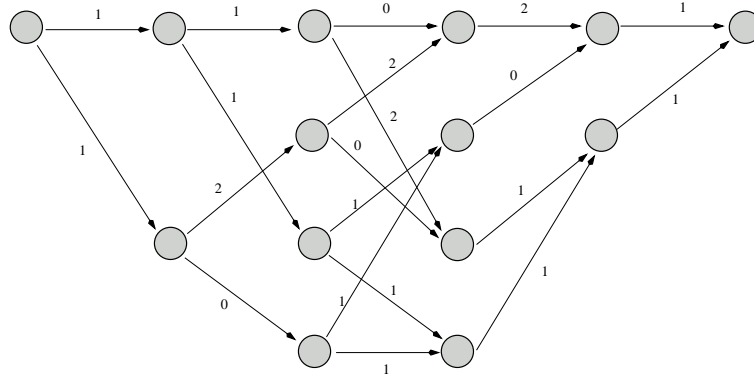


Abbildung 3.11: Ausgangssituation zur Viterbi-Pfadermittlung: Gitterdiagramm mit 4 Zuständen und 5 diskreten Zeitpunkten

### 3.2.7 Der Baum-Welch-Algorithmus

Der *Baum-Welch-Algorithmus* zielt darauf ab, das dritte grundlegende Problem im Umgang mit Hidden-Markov-Modellen zu lösen (Abschnitt 3.2.4). Die Modellparameter  $(A, B, \pi)$  sollen so angepasst werden, dass die Wahrscheinlichkeit für eine Beobachtungsfolge bei gegebenem Modell maximal wird. Dies ist gleichbedeutend mit einem Training des Modells, so dass nach erfolgtem Training mit Hilfe des Viterbi-Algorithmus die wahrscheinlichste Zustandsfolge ermittelt werden kann.

Der Baum-Welch-Algorithmus ist ein iteratives Verfahren, das die Modellparameter schrittweise so verfeinert, dass ein lokales Maximum für  $P(O|\lambda)$  gefunden wird. Alternativ lassen sich auch gradientenbasierte Verfahren einsetzen, wie zum Beispiel in [LeRS83] beschrieben.

Da kein analytisches Verfahren zur Bestimmung optimaler Modellparameter bekannt ist [Rabi89], wird in dieser Arbeit der Baum-Welch-Algorithmus [DeLR77] eingesetzt, dessen Optimierungen auf dem Konzept des Zählens von besuchten Zuständen fußt. Dabei sollen die Häufigkeiten für das Hidden-Markov-Modell als Spezialfall der Markov-Kette aus der Sequenz der Ausgabesymbole abgeleitet werden (vergleiche Abschnitt 3.2.6). Diese Häufigkeiten bilden dann eine ausreichende Wahrscheinlichkeit für die zugrunde liegende Wahrscheinlichkeitsverteilung.

Weil sich im allgemeinen Fall die Zustände eines Hidden-Markov-Modells nicht direkt beobachten lassen, bleibt auch ihre Ausgabe verborgen. Die benötigten Häufigkeiten können also nicht direkt berechnet werden. Stattdessen werden iterativ Schätzwerte für die Modellparameter bestimmt. Ausgehend von initialen Zufalls- oder Schätzwerten (Abschnitt 4.4.3) berechnen sich die Häufig-

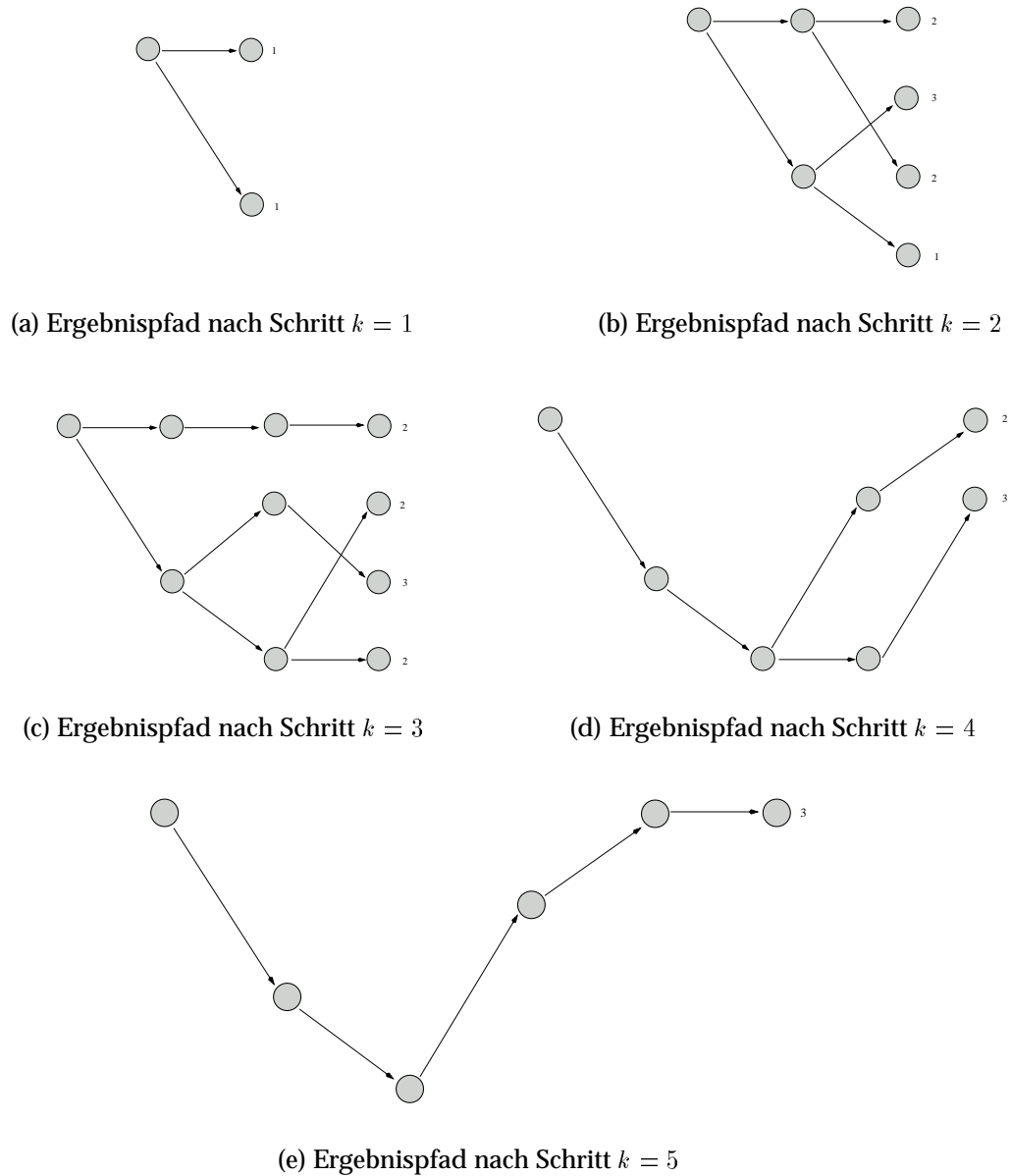


Abbildung 3.12: Schrittweise Entwicklung des Viterbi-Ergebnispfades

keiten bei gegebenem Modell und gegebener Beobachtungssequenz. Man erhält sie durch eine Gewichtung der beobachteten Transitionen mit der im Modell angegebenen Wahrscheinlichkeit. Die so erhaltenen Häufigkeitswerte ersetzen die bisherigen im Modell so lange, bis sich bei keiner weiteren Iteration eine Verbesserung einstellt. Jede Iteration erhöht somit die Wahrscheinlichkeit, dass die gegebene Beobachtungssequenz von diesem Modell emittiert wurde.

Bevor der eigentliche Algorithmus dargestellt werden kann, müssen zusätzlich zu den Vorwärts- und Rückwärtsvariablen noch zwei weitere Hilfsvariablen eingeführt werden. Hierbei handelt es sich zum Einen um  $\xi_t(i, j)$ , die Wahrscheinlichkeit, sich bei gegebenem Modell und gegebener Beobachtungsfolge zum Zeitpunkt  $t$  im Zustand  $s_i$  und zum Zeitpunkt  $t + 1$  im Zustand  $s_j$  zu befinden (Abbildung 3.13):

$$\xi_t(i, j) := P(q_t = s_i, q_{t+1} = s_j \mid O, \lambda). \quad (3.27)$$

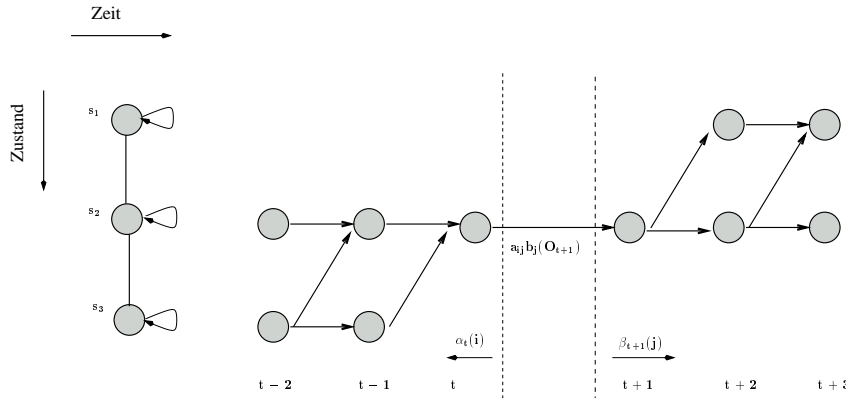


Abbildung 3.13: Berechnung des gemeinsamen Ereignisses, dass sich das System bei  $t$  in Zustand  $s_i$  und bei  $t + 1$  in Zustand  $s_j$  befindet

Mit Hilfe der Vorwärts- und Rückwärtsvariablen lässt sich dies so ausdrücken:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \quad (3.28)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}. \quad (3.29)$$

Setzt man nun  $\gamma_t(i)$  als diejenige Variable, die die Wahrscheinlichkeit, sich zum Zeitpunkt  $t$  im Zustand  $s_i$  zu befinden, in Beziehung zu  $\xi_t(i, j)$ , so erhält man:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j). \quad (3.30)$$

Betrachtet man  $\gamma_t(i)$  über die Zeit von  $t = 1$  bis  $t = T - 1$ , so liefert eine Summierung über  $t$  zwei Maßzahlen für die Anzahl zu erwartender Transitionen:

- Anzahl erwarteter Transitionen von  $s_i$ :

$$\sum_{t=1}^{T-1} \gamma_t(i) \quad (3.31)$$

- Anzahl zu erwartender Zustandsübergänge von  $s_i$  nach  $s_j$ :

$$\sum_{t=1}^{T-1} \xi_t(i, j) \quad (3.32)$$

Die bis hierher definierten Hilfsvariablen legen die Grundlage für den Baum-Welch-Algorithmus. Geht man vom aktuellen Modell, gegeben durch  $\lambda = (A, B, \pi)$  aus, so ergibt sich das verfeinerte Modell nach Anpassung der Parameter als  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ . Werden die neuen Parameter, wie im Folgenden beschrieben, ermittelt, so tritt einer dieser beiden Fälle ein [Rabi89]:

1. Das Modell  $\lambda$  definiert einen kritischen Punkt der Wahrscheinlichkeitsfunktion. In diesem Fall gilt  $\lambda = \bar{\lambda}$ .
2. Das Modell  $\bar{\lambda}$  ist wahrscheinlicher als  $\lambda$ , so dass  $P(O|\bar{\lambda}) \geq P(O|\lambda)$  gilt. Es wurde ein Modell gefunden, das mit einer größeren Wahrscheinlichkeit die Beobachtungsfolge hervorgebracht hat.

Die neuen Modellparameter sind folgendermaßen definiert:

- $\bar{\pi}_i$  bezeichnet die erwartete Häufigkeit eines Aufenthalts in Zustand  $s_i$  zum Zeitpunkt  $t = 1$ . Dies ist gleichbedeutend mit  $\gamma_1(i)$ .

$$\bar{\pi}_i = \gamma_1(i) \quad (3.33)$$

- $\bar{a}_{ij}$  ist der Quotient aus der erwarteten Anzahl von Transitionen von  $s_i$  nach  $s_j$  und der erwarteten Anzahl von Transitionen von  $s_i$ :

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \quad (3.34)$$

- $\bar{b}_j(k)$  ist der Quotient aus der erwarteten Häufigkeit eines Aufenthalts in Zustand  $s_j$  und der Beobachtung des Symbols  $v_k$  sowie der erwarteten Anzahl der Aufenthalte in Zustand  $s_j$ :

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (3.35)$$

Wird bei einer iterativen Anwendung der gezeigten Vorschriften jeweils  $\bar{\lambda}$  anstelle von  $\lambda$  benutzt, so ergibt sich eine schrittweise Verbesserung der angestrebten Wahrscheinlichkeit.

Der Algorithmusablauf sieht demnach so aus:

1. Das initiale Modell sei  $\lambda_0$ .
2. Berechne das neue Modell  $\lambda$  ausgehend vom Modell  $\lambda_0$  und der Beobachtung  $O$ .
3. Falls  $\log P(O|\lambda) - \log P(O|\lambda_0) < \Delta$  gilt, ist ein lokales Maximum gefunden. STOP.
4. Andernfalls setze  $\lambda_0 = \lambda$  und gehe zu Schritt zwei.

In Abbildung 3.14 wird der Trainingsprozess noch einmal veranschaulicht.

Problematisch ist, dass nur lokale Maxima gefunden werden, so dass als Alternative auch die Baum'sche Hilfsfunktion genutzt werden kann:

$$Q(\lambda, \bar{\lambda}) = \sum_Q P(Q|O, \lambda) \log[P(O, q|\bar{\lambda})]. \quad (3.36)$$

Eine Maximierung von  $Q(\lambda, \bar{\lambda})$  hat den gewünschten Effekt, so dass gilt:

$$\max_{\bar{\lambda}} [Q(\lambda, \bar{\lambda})] \Rightarrow P(O|\bar{\lambda}) \geq P(O|\lambda). \quad (3.37)$$

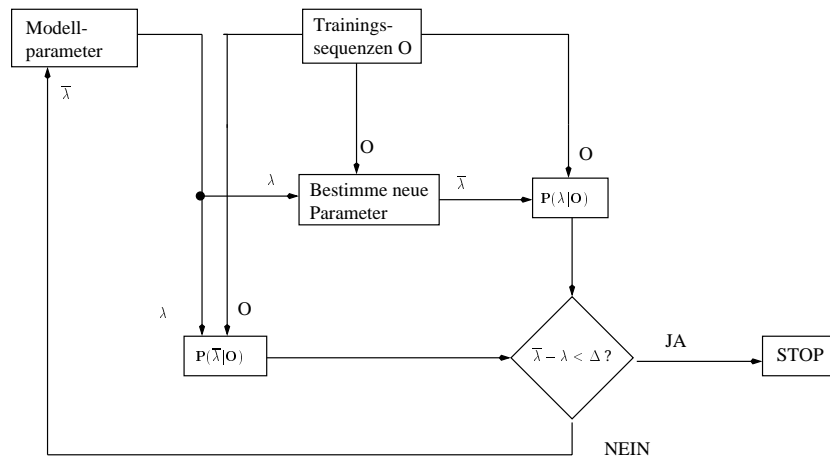


Abbildung 3.14: Training eines Hidden-Markov-Modells mit dem Baum-Welch-Algorithmus

Zu beachten beim Baum-Welch-Algorithmus ist, dass Beschränkungen, wie sie zum Beispiel bei Links-Rechts-Hidden-Markov-Modellen auftreten, den Algorithmus unberührt lassen. Parameter, die zu Beginn den Wert Null haben, behalten diesen auch während jedes Iterationsschrittes.

Außerdem werden die stochastischen Beschränkungen für die Wahrscheinlichkeitsvariablen in jedem Schritt erfüllt. Es gilt also immer:

$$\sum_{i=1}^N \bar{\pi}_i = 1, \quad (3.38)$$

$$\sum_{j=1}^N \bar{a}_{ij} = 1, \quad 1 \leq i \leq N, \quad (3.39)$$

$$\sum_{k=1}^M \bar{b}_j(k) = 1. \quad (3.40)$$

# 4 *Entwurf und Implementierung*

Die Erkennung ausgeführter Gesten basiert in dieser Arbeit im Wesentlichen auf Hidden-Markov-Modellen, deren Training und den in den Abschnitten 3.2.5, 3.2.6 und 3.2.7 eingeführten Algorithmen. Die hierzu notwendigen Modelle sowie die Entwurfsentscheidungen, die zu der Art und Weise führen, wie diese Entscheidungen umgesetzt wurden, werden in Abschnitt 4.4 erläutert.

Dem eigentlichen Erkennungsvorgang durch die Hidden-Markov-Modelle vorgeschaltet ist die Aufnahme der Gesten über ein Stereo-Kamerasystem und deren Umsetzung in Eingabedaten für die Erkennungsalgorithmen. Dazu behandelt Abschnitt 4.2 die Gestenaufnahme und -verfolgung mittels der Kameras und die dazu erforderlichen Vorverarbeitungsschritte. Abschnitt 4.3 widmet sich der Umwandlung von kontinuierlichen in diskrete Daten.

Der Strom diskreter Daten fließt in den Erkennungsmechanismus, der eine erkannte Geste zurückmeldet (Abschnitt 4.7). Grundlage dafür ist Aufbau und Training der Referenzmodelle (Abschnitt 4.4) sowie eine zur Erkennung gehörende Klassifikation (Abschnitt 4.6). Auf den Datenstrom werden dabei an mehreren Stellen Filter angewendet, die die Menge der zu verarbeitenden Daten reduzieren (Abschnitt 4.5).

Abschnitt 4.7 beschreibt zusammenfassend den Aufbau des Systems und den Ablauf der Gestenerkennung, und Abschnitt 4.8 schließt mit der Darstellung der experimentellen Ergebnisse und einer zusammenfassenden Gütebewertung der getesteten Verfahren.

Der Ablauf des gesamten Erkennungsprozesses gestaltet sich wie im Folgenden skizziert, woran sich dann auch die Reihenfolge der nachfolgend ausgeführten Einzelschritte orientiert:

1. Finden und Verfolgen der Hand über das Kamerasystem
2. Quantisierung und Filterung der Daten
3. Klassifikation der Daten und Erkennung der Geste

## 4.1 Gestenauswahl

Abbildung 4.1 zeigt eine Auswahl von fünf Gesten, die das System erkennen soll. Diese Gesten wurden so gewählt, dass sie nicht übermäßig komplex und nicht ineinander enthalten sind, um Irrtümer zu vermeiden. Weiterhin sind sie möglichst verschieden voneinander, um im ersten Schritt eine Erkennung nicht durch Mehrdeutigkeiten und Unsicherheiten im Zusammenhang mit ähnlichen Gesten zu erschweren.

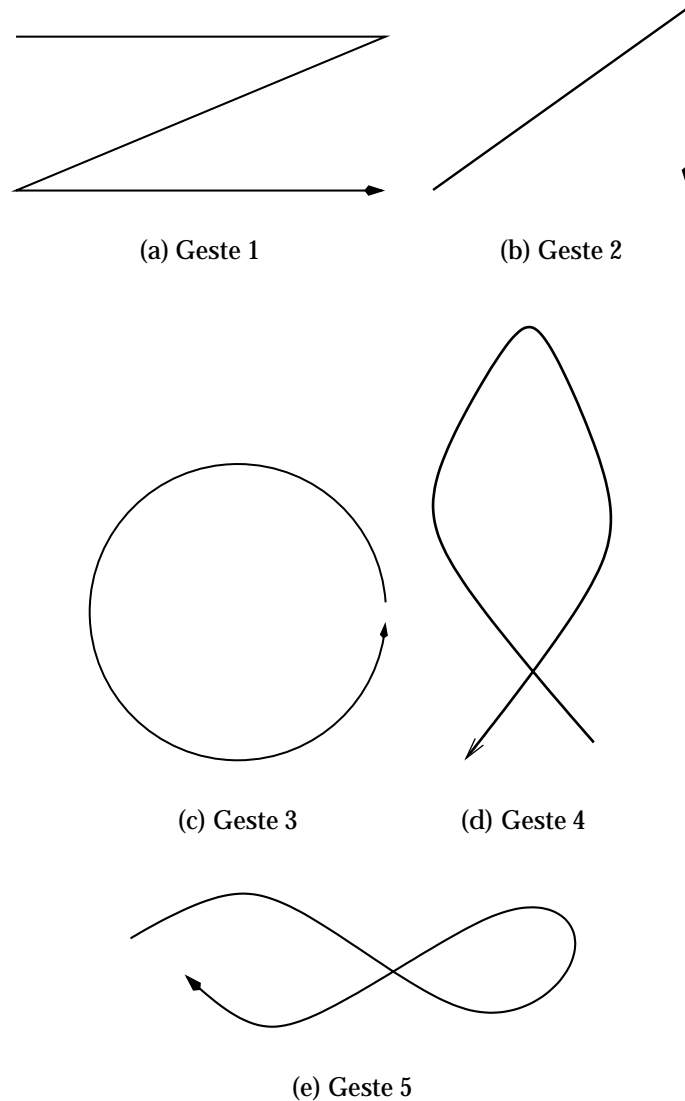


Abbildung 4.1: Die fünf vom System zu erkennenden Gesten

## 4.2 Objekterkennung und Objektverfolgung

Um eine menschliche Geste erfassen und anschließend auch interpretieren zu können, muss die Hand, die diese Geste vollführt, verfolgt werden. Voraussetzung dafür ist wiederum, diese Hand in den vom Kamerakopf gelieferten Bildern zu finden.

Das Finden der Hand im Bild und ihre Verfolgung basieren auf charakteristischen Eigenschaften der Hautfarbe, so dass eine Detektion möglich wird.

Die Analyse von Bildern im Hinblick auf das Vorhandensein oder das Finden von hautfarbenen Flächen birgt jedoch mehrere Probleme in sich. Zum Einen ist mit wechselnden Lichtverhältnissen zu rechnen, die die Bildqualität beeinflussen. Unterschiedliche Hautfarben verschiedener Personen sowie die Tatsache, dass unterschiedliche Kameras differierende Farbwerte sogar bei gleichen Rahmenbedingungen liefern, verursachen weitere Schwierigkeiten [YaLW97].

In [YaLW97] wird gezeigt, dass die Verschiedenheit menschlicher Hautfarben im Wesentlichen auf unterschiedlicher Intensität beruht und weniger auf den Farbwert selbst zurückzuführen sind. Demzufolge häufen sich die Farbwerte für Haut in bestimmten, relativ kleinen Bereichen eines Farbraums (Abbildung 4.2). Ausgehend von diesen Informationen kann in einem Bild, das ausschließlich auf Hautfarbwerte reduziert wurde, die erwünschte Hand zuverlässig gefunden werden.

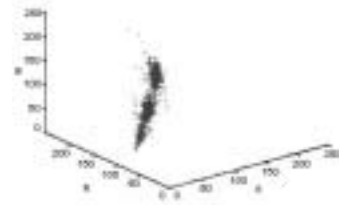
### 4.2.1 Farbraumkonvertierung und Binarisierung

Akquirierte Bilder müssen nun als erstes in eine Form gebracht werden, die eine Reduzierung auf ausschließlich Hautfarbwerte zulässt. Dies wird durch eine Farbraumkonvertierung von RGB nach HSI erreicht (Abschnitt 3.1.1). Aus dem so erhaltenen Bild können die S- und I-Werte für Sättigungs- und Intensitätsinformationen verworfen werden, so dass, wie gewünscht, lediglich die Information über den Farbwert zurückbleibt. Ein wesentlicher Vorteil besteht dabei darin, dass der die folgende Verarbeitung erschwerende Helligkeitsanteil herausgefiltert wird.

Im nächsten Schritt werden alle Farbwerte außerhalb des Farbintervalls  $[3, 31]$  verworfen, so dass nur solche zurückbleiben, die die Hautfarbe repräsentieren. Die nun folgende *Binarisierung* (Abbildung 4.3) reduziert den Speicherbedarf von ursprünglich  $3 \cdot 8$  Bit pro Pixel auf ein Bit pro Pixel. Es bleibt ein zweifarbiges Bild, in dem die Hintergrundfarbe dominiert, aber auch alle hautfarbenen Pixel vorhanden sind.



(a) Farbbild mit hautfarbenen Flächen



(b) Häufung von Hautfarbwerten



(c) Verteilung der Hautfarbwerte

Abbildung 4.2: Beispiel für Häufung und Verteilung von Hautfarbwerten

Auf das binarisierte Bild werden nun Erosions- und Dilationsfilter angewendet, um eine Bereinigung durchzuführen. Die Filter lassen eine Vielzahl von hautfarbenen *Blobs* zurück, aus denen drei aufgrund ihrer Größe hervorstechen (Abbildung 4.2.1). Als *Blobs* bezeichnet man zusammenhängende gleichfarbige Flächen in einem Bild. Bei diesen dominierenden drei *Blobs* handelt es sich um die beiden Hände und das Gesicht.

Um Fehler bei der Zuordnung der Hände und des Gesichts zu den *Blobs* zu vermeiden, wird eine Mindest- und eine Maximalgröße für Hand- und Gesicht-*Blobs* festgelegt. Dies soll verhindern, dass eventuell auftretende kleinere Flächen irrtümlich für Hand oder Gesicht gehalten werden. Zusätzlich können auch die drei größten der noch in Frage kommenden *Blobs* ausgewählt und alle anderen verworfen werden.

Heuristisch hat sich erwiesen, dass der linke der unteren *Blobs* die rechte Hand darstellt. Genau diese wird nun bei einer Handbewegung verfolgt. Voraussetzung ist natürlich, dass beide Arme an der Seite herunterhängen. Da es sich hierbei um eine sehr natürliche Position handelt, kann sie als Ausgangsposition angenommen werden.



(a) Bild vor der Binarisierung



(b) Bild nach der Binarisierung

Abbildung 4.3: Beispiel für eine Binarisierung: Entfernung der Sättigungs- und Intensitätsinformationen eines HSI-Bildes



Abbildung 4.4: Ergebnisbild nach Anwendung der Opening- und Closing-Filter. Löcher in zusammenhängenden Flächen sind gestopft, isolierte Pixel entfernt.

### 4.2.2 Handverfolgung

Sobald die Hand im Bild lokalisiert wurde, kann die Bewegung und ihre Verfolgung beginnen. Um eine aufwändige globale Suche nach der neuen Handpo-

sition in jedem neuen Bild zu vermeiden, beschränkt sich die Betrachtung auf einen Ausschnitt, nämlich ein lokales Fenster um die initiale Handposition. Die Abmessungen dieses lokalen Fensters werden so großzügig gewählt, dass sich die Hand auch nach der Bewegung im nächsten Bild immer noch innerhalb des lokalen Fensters befindet.

Die weitere Handverfolgung beschränkt sich nun nur noch auf das lokale Fenster. Die Fensterposition im neuen Bild wird so angepasst, dass der Hand-Blob im lokalen Fenster zentriert ist. Des Weiteren wird nun der gleiche Ablauf lokal wiederholt, wie er bereits global stattgefunden hat. Das heißt, die Schritte Konvertierung, Binarisierung, Filterung und Blob-Suche werden von jetzt an beschränkt auf das lokale Fenster durchgeführt. Ein Verlust der Hand durch eine zu schnelle Bewegung führt zu einer erneuten globalen Suche. Eine solche globale Suche benötigt etwa 300 ms, wohingegen die Bearbeitung des lokalen Fensters bei einer Größe von  $150 \times 150$  Pixel nur etwa 15 ms in Anspruch nimmt. Diese Geschwindigkeit ermöglicht eine Handverfolgung in Echtzeit.

Abbildung 4.5 zeigt den Ablauf als Ganzes.

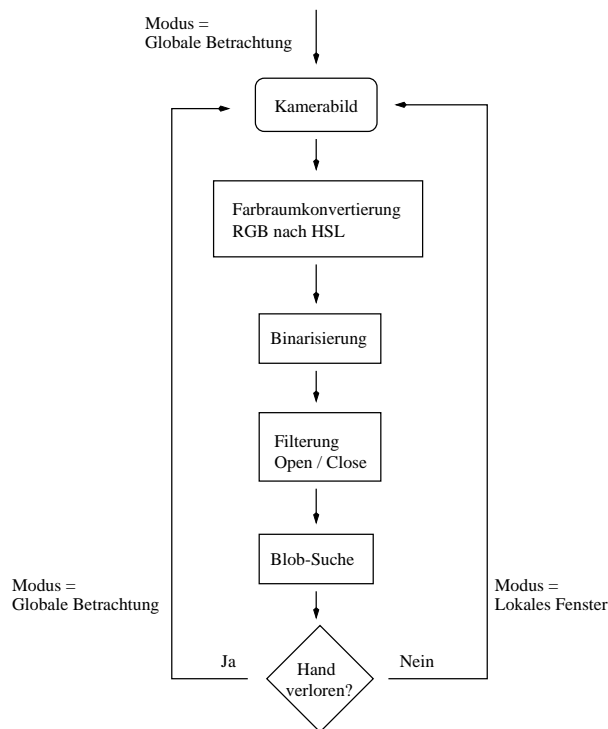


Abbildung 4.5: Ablauf bei der Handverfolgung, basierend auf einer Kamerabildsequenz

Für den Blob der rechten Hand wird in jedem Bild der Schwerpunkt ermittelt

und dessen Koordinaten an das Erkennungssystem zur Verarbeitung weitergeleitet.

Bei der Handbewegung ist darauf zu achten, dass es zu keinen Überlappungen zwischen den Händen oder einer Hand und dem Gesicht kommt. Bei solchen Überlappungen verschmelzen nämlich die beiden ursprünglichen Blobs zu einem einzigen. Sobald die Handbewegung weitergeht und sich die Blobs wieder trennen, kann aber nur schwer entschieden werden, welcher Blob welche Hand beziehungsweise das Gesicht repräsentiert.

## 4.3 Vektorquantisierung

Für den in dieser Arbeit gewählten Lösungsentwurf werden Hidden-Markov-Modelle in ihrer diskreten Form angewendet. Es handelt sich hierbei um ein zeitdiskretes Modell, bei dem auch die Ausgabesymbole einem diskreten Alphabet angehören.

Bei der *Vektorquantisierung* handelt es sich um eine Technik, die Prototypen von Beobachtungen hervorbringt. Aus der Menge aller möglichen Beobachtungen wird jede Beobachtung einem Prototyp zugeordnet. So kann die unendlich große Menge von Beobachtungen durch eine endliche Menge von Prototypen dargestellt werden. Der kontinuierliche Eingabestrom wird also auf diskrete Werte abgebildet. Prototypen werden auch als *Codewörter* bezeichnet und die Menge der Codewörter als *Codebuch* [HuAJ90].

Im Gegensatz zur skalaren Quantisierung, bei der den Signalparametern unabhängig voneinander ein Wert zugeordnet wird, bildet die Vektorquantisierung eine Menge von Parametern auf einmal ab.

Angenommen, es handele sich bei  $x = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}^d$  um einen  $d$ -dimensionalen Vektor mit kontinuierlichen Komponenten  $x_k$ , so bildet die Quantisierung den Vektor  $x$  auf einen Vektor  $z$  folgendermaßen ab:

$$z = q(x). \quad (4.1)$$

$q$  bezeichnet die Quantisierungsfunktion,  $z$  einen Wert aus dem Codebuch  $Z = \{z_i \mid 1 \leq i \leq L\}$  mit  $z_i = (z_{i1}, z_{i2}, \dots, z_{id})$  und  $L$  die Größe des Codebuchs.

### 4.3.1 Codebucherzeugung

Da das Codebuch wesentlicher Bestandteil des Quantisierungsvorgangs ist, kommt seinem Design im Hinblick auf brauchbare Ergebnisse eine besondere

Bedeutung zu.

Dazu wird der  $d$ -dimensionale Raum des Vektors  $x$  in  $L$  Regionen  $C_i$  aufgeteilt. Jeder dieser Regionen ist ein Vektor  $z_i$  zugeordnet. Die Quantisierungsfunktion bildet dann den Vektor, der in der Region  $C_i$  liegt, auf den Vektor  $z_i$  ab:

$$q(x) = z_i, \text{ wenn } x \in C_i. \quad (4.2)$$

Die Erzeugung eines Codebuchs wird auch als *Codebuch-Training* bezeichnet. Bei der Abbildung von Werten ( $x$ ) aus einer unendlichen Menge auf solche aus einer endlichen ( $z$ ) tritt unvermeidlich ein Quantisierungsfehler auf.  $d(x, z)$  gibt dann die Verzerrung an und kann als Maß für die Quantisierungsqualität angesehen werden.

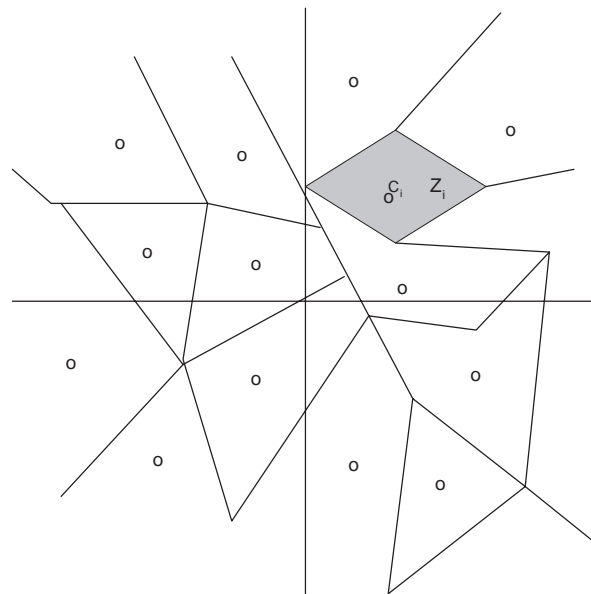


Abbildung 4.6: Beispiel für eine zweidimensionale Regionenaufteilung bei der Vektorquantisierung

Ein Beispiel für eine Aufteilung in Regionen im zweidimensionalen Fall ist in Abbildung 4.6 skizziert.

Die Formen der einzelnen Regionen müssen nicht identisch sein. Die Position der Codewörter in den jeweiligen Regionen wird durch eine Minimierung der Verzerrung der jeweiligen Region bestimmt. Die Codewörter in den Regionen von Abbildung 4.6 sind durch Kreise veranschaulicht.

Setzt man bei der Codebucherzeugung die Minimierung der gesamten durchschnittlichen Verzerrung als Kriterium an, so bietet sich der *k-means Algorithmus* zur Realisierung an [HuAJ90].

Die Bedingungen für ein optimales Ergebnis sind:

1. Nachbarschaftsbedingung.  $q(x) = z_i$ , falls  $d(x, z_i) \leq d(x, z_j)$ ,  $j \neq i$
2. Verzerrungsbedingung. Jedes Codewort  $z_i$  muss so gewählt werden, dass die durchschnittliche Verzerrung in der Region  $C_i$  minimal ist.

Der Ablauf des Algorithmus lässt sich in etwa so skizzieren:

1. **Initialisierung:** Bestimme ein initiales Codebuch.
2. **Klassifizierung** Klassifiziere jedes Element der Trainingsvektoren nach Zugehörigkeit zu einer Region  $C_i$ . Dies geschieht durch Auswahl des nächsten Codeworts  $z_i$ .
3. **Codebuch-Aktualisierung:** Aktualisiere das Codewort einer jeden Region durch Berechnung des Zentrums des jeweiligen Trainingsvektors.
4. **Schluss:** Ist die Gesamtverzerrung im aktuellen Schritt im Vergleich zur Gesamtverzerrung im vorherigen Schritt unter einen Grenzwert gesunken, halte an. Andernfalls fahre fort bei Schritt zwei.

Beim Design des Codebuchs muss im Hinblick auf seinen Umfang zwischen Genauigkeit und Aufwand abgewogen werden. Ein Codebuch mit vielen Codewörtern reduziert das Ausmaß der unvermeidlichen Quantisierungsfehler, wohingegen die Anzahl der notwendigen Parameter für die Hidden-Markov-Modelle steigt und einen deutlich höheren Rechenaufwand nach sich zieht. In diesem Fall wurde eine Aufteilung auf 16 Codewörter gewählt, da auf der einen Seite der Rechenaufwand in einem vertretbaren Rahmen gehalten wird und auf der anderen Seite aber auch alle Bewegungsrichtungen einer Geste im Wesentlichen abgedeckt sind. Die Wahl von 16 Codewörtern führt demzufolge bei den Referenzmodellen zu einem diskreten Alphabet aus 16 Zeichen. Das in dieser Arbeit zur Quantisierung verwendete Codebuch mit seinen 16 Richtungsindizes ist in Abbildung 4.7 gezeigt.

Wie in Abbildung 4.8 veranschaulicht, wird aus den Pixelkoordinaten zweier Punkte mit (4.3) der Winkel  $\alpha$  bestimmt.

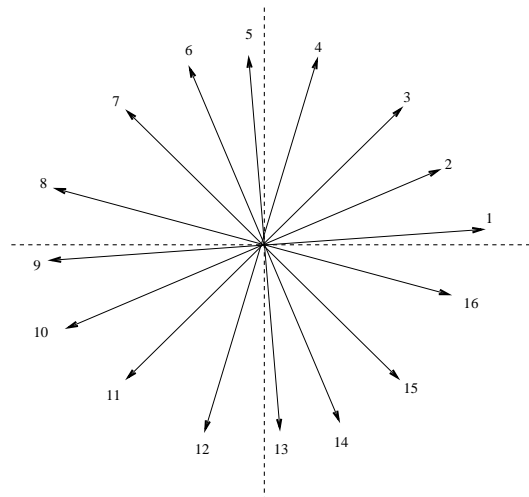


Abbildung 4.7: Das zur Vektorquantisierung verwendete Codebuch mit 16 Codewörtern (Richtungsindizes)

$$\alpha = \arctan \frac{y_2 - y_1}{x_2 - x_1}, \quad x_1 < x_2 \quad (4.3)$$

Dieser wird dann gemäß Tabelle 4.1 auf einen Richtungsindex des Codebuchs (Abbildung 4.7) abgebildet. Als Kriterium bei der Zuordnung dient die Nachbarschaftsbedingung.

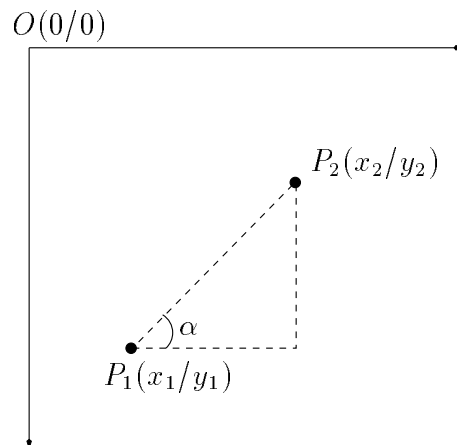


Abbildung 4.8: Entstehung der Bewegungsvektoren aus aufeinander folgenden Pixelkoordinaten

Winkel	Richtungsindex
0 °	1
22,5 °	2
45 °	3
77,5 °	4
90 °	5
112,5°	6
135 °	7
157,5°	8
180 °	9
202,5°	10
225 °	11
247,5°	12
270 °	13
292,5°	14
315 °	15
337,5°	16

Tabelle 4.1: Zuordnung zwischen Bewegungsvektoren und Richtungsindizes

## 4.4 Hidden-Markov-Modelle

Nachdem beginnend mit Abschnitt 3.2 das Konzept der Hidden-Markov-Modelle eingeführt wurde und ihre Anwendung in den Abschnitten 1.1 und 4.4.1 motiviert wird, ist im Folgenden ihre Anwendung und ihr Einsatz im Hinblick auf das zu erreichende Ziel der Gestenerkennung Zentrum der Betrachtung.

An die Beschreibung des Aufbaus der Referenzmodelle unter Beachtung von Randbedingungen und Besonderheiten (Abschnitt 4.4.2) schließt sich die Bestimmung der initialen Modellparameter (Abschnitt 4.4.3) sowie die Einführung in das Konzept des Schwellwertmodells (Abschnitt 4.4.5) an.

### 4.4.1 Hidden-Markov-Modelle versus neuronale Netze

Sobald Aufgabenstellungen zu lösen sind, die eine große Menge an mehrdimensionalen und schwach strukturierten Daten mit sich bringen, gewinnt der Einsatz von neuronalen Netzen an Gewicht. So ist die Verwendung von unterschiedlichen neuronalen Netzen in der Spracherkennung bereits weit verbreitet. Sind zeitliche Faktoren zu berücksichtigen, kommt der Einsatz von *Time-Delay*

*Neural Networks* (TDNN) in Betracht. TDNN haben die Fähigkeit, zeitliche Zusammenhänge zu erlernen.

Die Fragestellung, die im Zusammenhang mit der Verwendung neuronaler Netze bei der Gestenerkennung zu beantworten ist, liegt in der Modellierung der nicht bedeutungstragenden Gesten. Nach [LeKi99] sind neuronale Netze zwar sehr wohl zur Erkennung statischer Muster geeignet, aber zur Bearbeitung dynamischer gänzlich ungeeignet. Das bedeutet, dass Handposition und Fingerstellung zwar trainiert und effizient erkannt werden können, aus komplexen Bewegungsabläufen bestehende Gesten den Rahmen des Möglichen aber sprengen. Aus diesen Gründen kommt eine Verwendung von neuronalen Netzen für das dieser Arbeit zugrunde liegende Projekt nicht in Frage.

#### 4.4.2 Referenzmodelle

Für jede der gebildeten und vom System zu erkennenden Geste wird ein Referenzmodell erzeugt, das mit Hilfe des Baum–Welch–Algorithmus (Abschnitt 3.2.7) und Gesten von unterschiedlichen Personen trainiert wird (Abschnitt 4.4.4.3). Bei diesen Referenzmodellen handelt es sich um die in Abschnitt 3.2.3 eingeführten Links–Rechts–Modelle, so dass für sie gilt:

$$\pi_1 = 1 \text{ und } \pi_i = 0, i \neq 1, \quad (4.4)$$

$$a_{ij} = 0, i \leq j. \quad (4.5)$$

Links–Rechts–Modelle sind zur Modellierung der Gesten besonders geeignet, da sich mit ihnen die räumlich–zeitlichen Gegebenheiten einer menschlichen Geste gut modellieren lassen. Zustandsübergänge in Links–Rechts–Modellen sind dahingehend vorgegeben, dass nur Übergänge von links nach rechts erlaubt sind. Zu jedem diskreten Zeitpunkt  $t$  findet ein Übergang statt, so dass sich der Bewegungsablauf einer Geste auf das Modell abbilden lässt. Die räumliche Positionsänderung der Hand in Form einer Folge von Richtungsindizes korrespondiert dabei zu den vom Modell ausgegebenen Symbolen. Der zeitliche Ablauf wird durch die Zustandsübergänge dargestellt.

Es wird eine Sprungbegrenzung von  $\Delta = 2$  eingeführt, die das Überspringen eines einzelnen Zustands erlaubt, aber weitere Sprünge verhindert. Die Möglichkeit zum Überspringen von Zuständen führt dazu, dass die Erkennung toleranter wird. Eine ausgeführte Geste muss also nicht völlig identisch mit der Referenzgeste sein, um erkannt zu werden (Abschnitt 4.4.3). Auf der anderen Seite ist eine Begrenzung der Sprungweite notwendig, um die fehlerhafte Zuordnung von einer nur ähnlichen Geste zu verhindern.

Nach [LeKi99] und [Rabi89] hat die Anzahl der Zustände nur bedingten Einfluss auf die Erfolgsrate des Erkennungssystems. Obwohl bei einer bestimmten Anzahl von Zuständen ein lokales Maximum erreicht wird, führt die weitere Hinzunahme von Zuständen zu keiner signifikanten Verbesserung. Nachteilig wirkt sie sich aber auf den Berechnungsaufwand bei Training und Erkennung aus, da eine höhere Anzahl von Parametern mit einbezogen werden muss. Eine hohe Anzahl von Zuständen hat insbesondere bei der Konstruktion des Schwellwertmodells negative Folgen (Abschnitt 4.4.5).

Tabelle 4.2 zeigt die Entwicklung der Erkennungswahrscheinlichkeiten für die Gesten eins und fünf bei wachsender Anzahl von Zuständen und bei der Anwendung auf eine Beobachtungssequenz der Länge 16. Mit angegeben ist die Anzahl der zur Erkennung nötigen Berechnungen, die sich durch  $N^2T$  ergibt.

Zustände	$P(O   \text{Geste 1})$	$P(O   \text{Geste 5})$	Berechnungen
3	1,374206E-04	6,364537E-13	144
4	1,405220E-03	8,566302E-12	256
5	1,537310E-03	2,597614E-11	400
6	1,689013E-03	2,872422E-10	576
8	4,477910E-03	3,251160E-09	1024
10	2,431920E-02	2,010504E-08	1600
12	./.	8,175456E-07	2304
15	./.	8,065692E-05	3600

Tabelle 4.2: Erkennungswahrscheinlichkeiten bei wachsender Anzahl von Zuständen für Geste Nummer eins und fünf

Es ist erkennbar, dass mit steigender Anzahl von Zuständen auch die Erkennungswahrscheinlichkeit ansteigt. Aus der Wertentwicklung für Geste eins lässt sich erkennen, dass ab einer gewissen Anzahl von Zuständen ein signifikanter Zuwachs bei der Wahrscheinlichkeit nur durch eine deutlich höhere Zustandsanzahl erkaufte werden kann. Die verbesserten Wahrscheinlichkeiten bei Geste fünf für die aufgeführten Zustandsanzahlen begründen sich in der größeren Komplexität dieser Geste. Für ihre Modellierung sind mehr Zustände nötig als für die vom Bewegungsablauf her einfachere erste Geste.

Da mit steigender Anzahl von Zuständen eines Modells aber auch der Berechnungsaufwand quadratisch wächst, muss bei der Wahl der Zustandsanzahl abgewogen werden. Modelle mit mehr Zuständen liefern tendenziell auch bessere Ergebnisse, allerdings beschränken diese aufgrund der zusätzlichen Rechenzeit möglicherweise die Echtzeitfähigkeit.

Genauso wie die initialen Modellparameter auf Schätzungen basieren, muss die beste Anzahl der Zustände für jedes Referenzmodell experimentell bestimmt

werden, da keine allgemeinen Regeln existieren, um die zu modellierenden Signale abzubilden. Gewisse Rahmenbedingungen sind aber dadurch gegeben, dass die von den Modellen emittierten Ausgabesymbole den zu modellierenden physikalischen Signalen entsprechen. Dadurch sind die Zustände und damit auch ihre Anzahl an eben diese Rahmenbedingungen geknüpft und somit der Spielraum beschränkt.

Die Experimente zur Bestimmung der Zustandsanzahl sind zweigleisiger Natur. Allgemein richtet sich die Anzahl der Zustände der Referenzmodelle nach der Komplexität der korrespondierenden Geste. Mit wachsender Anzahl von Richtungsindizes, die eine Geste beschreiben und die den Ausgabesymbolen der Hidden-Markov-Modelle entsprechen, wächst auch die Anzahl der Zustände, da diese die Symbole emittieren müssen.

Richtwerte für die Zustandsanzahl der einzelnen Referenzmodelle werden mittels *reverse engineering* bestimmt. Mit dieser sowie mit einer leicht erhöhten und leicht verringerten Anzahl werden dann Erkennungsexperimente durchgeführt, um die Qualität weiter zu verbessern.

Diese Experimente basieren auf dem Testen von mehrfachen Beobachtungssequenzen. Es wird das Produkt  $P(O|\lambda)$  der einzelnen Wahrscheinlichkeiten als Gütemaß herangezogen (4.9), um eine möglichst allgemeine Aussage über die Erkennungsqualität treffen zu können.

Der Entscheidungsprozess, der zur Wahl der Anzahl der Zustände eines jeden Referenzmodells führt, gliedert sich in die folgenden Schritte:

1. Mehrmaliges Aufnehmen der zu modellierenden Referenzgeste
2. Vektorquantisierung und Filterung gleicher aufeinanderfolgender Richtungsindizes
3. Aufbau des Referenzmodells ausgehend vom Ergebnis aus Schritt zwei
4. Erkennungstests des Referenzmodells mit variierender Zustandsanzahl und den trainierten Gesten als Beobachtungssequenzen
5. Entscheidung für eine Zustandsanzahl auf Basis der Ergebnisse aus Schritt vier

Grundlage des Modellaufbaus ist die zu modellierende Geste. Um eine repräsentative Basis zu haben, wird die Geste in ihrer Idealform mehrmals aufgenommen. Dies soll sicherstellen, dass keine Spezialfälle oder Besonderheiten modelliert werden und ist nicht mit dem späteren Training des Modells zu verwechseln. Deswegen sind hier einige wenige Aufnahmen ausreichend, wohingegen

beim Training die Auswahl möglichst groß und breit gefächert sein sollte (Abschnitt 4.4.4).

Auf jede dieser Gesten wird die Vektorquantisierung angewendet. Als Ergebnis erhält man pro Aufnahme eine Folge von Richtungsindizes. Diese Folge dient als Eingabe in den Gleichheitsfilter, der gleiche aufeinander folgende Indizes entfernt, so dass die Folge anschließend aus paarweise verschiedenen Richtungsindizes besteht (Abschnitt 4.5). Die der Vektorquantisierung vorhergehende Start/Stop-Filterung entfernt eine Häufung nahe beieinander liegender Punkte. Diese entstehen dadurch, dass Beginn und Ende einer Geste über eine Pause in der Handbewegung mit einer bestimmten Mindestlänge verdeutlicht werden (Abschnitt 4.7).

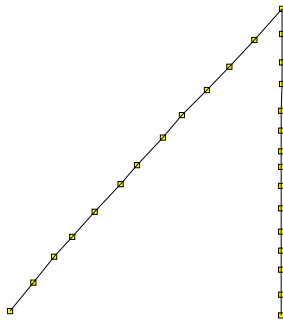
Abbildung 4.9 illustriert den Ablauf für eine künstlich modellierte ideale Geste und für eine real aufgenommene.

Die hierdurch entstehende Folge von Richtungsindizes bildet die Entscheidungsgrundlage für die Wahl der Zustandsanzahl. Im Idealfall entspräche ein Ausgabesymbol einem Zustand im Hidden-Markov-Modell. Eine solche Realisierung führt aber zu Modellen mit sehr hoher Anzahl von Zuständen, was wiederum einen hohen Berechnungsaufwand nach sich zieht. Gesten, die aus Geraden zusammengesetzt sind, können durch einen Richtungsindex pro Gerade beschrieben werden und führen zu Modellen, die einen Zustand pro Gerade aufweisen. Die Praxis hat aber gezeigt, dass Handbewegungen entlang einer Geraden nahezu nie dem Ideal entsprechen und sich oft eine Folge zweier benachbarter und alternierender Richtungsindizes ergibt. Eine schematische eins zu eins Zuordnung würde also zum oben beschriebenen unerwünschten Ergebnis führen.

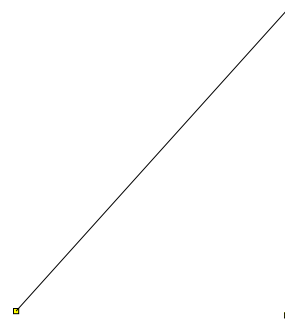
Da die Zustände eines Links-Rechts-Modells nun aber in der Lage sind, mehrere Symbole auszugeben, bevor ein Übergang zum nächsten Zustand stattfindet, können die alternierenden Symbole dennoch einem Zustand im Referenzmodell zugeschlagen werden. Das Wissen über die Richtungsindizes, die in diesen Fällen behandelt werden, kann dann bei der Schätzung der initialen Parameter für die Matrix  $B$  der Symbolausgabewahrscheinlichkeiten (Abschnitt 4.4.3) nutzbringend eingesetzt werden.

Abschließende Experimente mit leicht variierenden Zustandszahlen und verschiedenen Beobachtungssequenzen bringen Klarheit über die endgültige Zustandsanzahl.

Die aus den Versuchen resultierende Zustandsverteilung für die Referenzmodelle zeigt Tabelle 4.3.



(a) Künstlich modellierte Geste mit 26 Segmenten



(b) Künstliche Geste nach Filterung mit 2 Segmenten



(c) Reale Geste mit 121 Segmenten



(d) Reale gefilterte Geste mit 18 Segmenten

Abbildung 4.9: Reduzierung der Segmentanzahl einer Gestentrajektorie durch Anwendung des Gleichheits- und des Start/Stop-Filters

Geste Nr.	Geste	Zustände
1		5
2		5
3		6
4		6
5		8

Tabelle 4.3: Experimentell bestimmte Zustandsanzahl der Referenzmodelle

### 4.4.3 Bestimmung initialer Modellparameter

Bevor der Baum–Welch–Algorithmus zum Training der Referenzmodelle auf diese angewendet werden kann, müssen die Modellparameter  $A$ ,  $B$  und  $\pi$  mit Werten belegt werden. Diesen initialen Werten kommt eine besondere Bedeutung zu, da sie Einfluss auf das Trainingsergebnis haben.

Der Baum–Welch–Algorithmus garantiert ein lokales Maximum der Wahrscheinlichkeit zu finden, mit der die zum Training verwendete Beobachtungssequenz von diesem Modell erzeugt wurde. Ziel ist es nun, die initialen Modellparameter so zu wählen, dass das gefundene lokale Maximum dem globalen Maximum entspricht.

Bevor allerdings die Modellparameter  $A$ ,  $B$  und  $\pi$  festgelegt werden können, muss die grundsätzliche Entscheidung über die Struktur der Modelle getroffen werden. Dies beinhaltet die Festlegung der Zustandsanzahl sowie der Verbindungsstruktur, die entweder ergodisch oder Links–Rechts sein kann. Diese Punkte wurden bereits im Abschnitt 4.4.2 erläutert, so dass sich der folgende Teil der Wertfindung der Modellparameter widmet.

Die Anfangswahrscheinlichkeitsverteilung  $\pi$  ist festgelegt durch die Bedingung, dass für Links–Rechts Modelle gilt:  $\pi_1 = 1$  und  $\pi_i = 0, i \neq 1$ .

Für die Übergangswahrscheinlichkeiten in  $A$  und die Symbolausgabewahrscheinlichkeiten in  $B$  gibt es keine analytische, sondern nur heuristische Bestimmungsmethoden [Rabi89]. Für  $A$  bietet sich entweder eine zufällige Auswahl der Parameter an, die lediglich den stochastischen Randbedingungen genügt, oder die Auswahl gleichartiger Werte, wieder unter Beachtung der Randbedingungen. Gleichartig bedeutet hierbei, dass gilt:

$$a_{ij} = \frac{1}{N_i}. \quad (4.6)$$

$N_i$  bezeichnet die Anzahl der Werte in Zeile  $i$  der Matrix  $A$ , die ungleich Null sind.

In dieser Arbeit werden für  $A$  gleichartige Parameter eingesetzt, da die so entstehenden Modelle dem Ablauf der realen Gesten am besten entsprechen. Ferner wird eine Sprungbegrenzung von  $\Delta = 2$  gewählt, da sie sich in Experimenten als dem Gesamterfolg am förderlichsten erwiesen hat.

#### 4.4.3.1 Wahl der Sprungbegrenzung

Abbildung 4.10 zeigt, dass die Wahl von  $\Delta = 2$ , also bei der Erlaubnis nicht nur zum nächsten Zustand, sondern auch zum übernächsten zu wechseln, zu schlechteren Werten bei der Erkennung führen kann. In diesem Fall wurde jeweils  $P(O|\lambda)$  für ein Modell mit 3, 4, 5, 6, 8, 10, und 15 Zuständen bestimmt (Abschnitt 4.4.4.1). Dabei werden die Fälle  $\Delta = 1$  und  $\Delta = 2$  unterschieden. Bei den zum Testen verwendeten Sequenzen  $O$  handelt es sich um diejenigen, die auch zum Training genutzt wurden. Dadurch erklären sich auch die verschlechterten Wahrscheinlichkeiten, da sich eine weniger restriktive Sprungbegrenzung erst bei Sequenzen positiv bemerkbar macht, die nicht den Idealdaten entsprechen. Ein größerer Wert für  $\Delta$  erlaubt also eine flexiblere und damit bessere Erkennung, jedoch geht diese Flexibilität zu Lasten der Erkennung von idealen Sequenzen. Da aber ausgeführte Gesten in der Praxis nicht genau den Referenzgesten entsprechen, ist dieser Punkt vernachlässigbar.

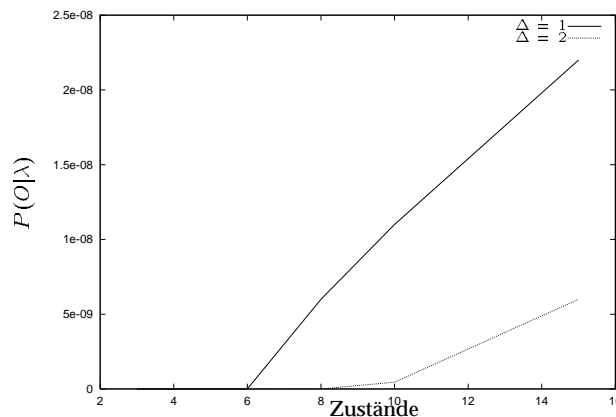


Abbildung 4.10: Wahrscheinlichkeiten bei der Erkennung für Modelle mit  $\Delta = 1$  und  $\Delta = 2$  bei idealen Daten

#### 4.4.3.2 Bestimmung der Symbolausgabewahrscheinlichkeiten

Besonderes Augenmerk sollte auf die Auswahl der Werte für  $B$  gerichtet werden, da hier die Auswahl der Werte den Trainingsprozess sehr wohl unterstützen kann [Rabi89]. Können keine Schätzungen für die Symbolausgabewahrscheinlichkeiten abgegeben werden, so ist die Belegung von  $B$  mit zufälligen Werten kontraproduktiv und sollte vermieden werden, da sie den Baum-Welch-Algorithmus unter Umständen in eine falsche Richtung lenkt. Statt dessen sollte in diesem Fall  $b_{ij} = \frac{1}{N}$  gelten, so dass bei  $N$  Zuständen für alle Symbole die gleiche Ausgabewahrscheinlichkeit vorliegt.

In Tabelle 4.4 ist der Erfolg  $P(O|\lambda)$  für verschiedene Modelle bei gleichmäßig verteiltem und bei geschätztem  $B$  gezeigt. Die Werte für beide Fäl-

Geste	$P(O \lambda)$ (G)	Iterationen	$P(O \lambda)$ (M)	Iterationen
1	1,231185E-11	112	1,502043E-11	100
2	6,767004E-09	19	3,772834E-08	41
3	1,081666E-15	66	1,329379E-15	52
4	4,422434E-19	141	1,223805E-18	115
5	3,975434E-17	93	4,761217E-17	48

Tabelle 4.4: Wahrscheinlichkeiten bei gleichmäßig verteiltem (G) und manuell (M) bestimmtem  $B$  für die Referenzmodelle

le von  $P(O|\lambda)$  berechnen sich aus den Erkennungswahrscheinlichkeiten für drei Beobachtungssequenzen der Länge 16. Man erkennt, dass die Ergebnisse für die manuelle Vergabe der Parameter von  $B$  in jedem einzelnen Fall besser sind als für den Fall der Gleichverteilung. Bei den Gesten, deren Wahrscheinlichkeiten sich nur marginal unterscheiden (Geste eins und Geste drei) wurde das bessere Ergebnis zumindest mit einer geringeren Anzahl von Iterationen des Baum-Welch-Algorithmus erreicht, so dass auch hier zu sehen ist, dass die manuelle Bestimmung der  $b_{ij}$  den Algorithmus in die richtige Richtung lenkt und das Ergebnis schneller erreicht wird.

Die Ergebnisse für einzelne Beobachtungssequenzen können sehr unterschiedlich ausfallen, unter Umständen ist sogar die gleichartige Verteilung im Einzelfall besser (siehe Tabelle 4.5). Solche Fälle sind hervorgehoben dargestellt. Dies kann insbesondere dann auftreten, wenn Beobachtungssequenzen betrachtet werden, die im Training nicht vorgekommen sind. Die durch  $P(O|\lambda)$  gegebene Gesamtwahrscheinlichkeit liegt aber trotz allem in allen betrachteten Fällen höher.

Geste	$P(O_1 \lambda)$ (G)	$P(O_1 \lambda)$ (M)	$P(O_2 \lambda)$ (G)	$P(O_2 \lambda)$ (M)
1	<b>7,978579E-03</b>	6,991249E-03	1,019114E-05	7,320465E-06
2	7,618537E-02	7,721584E-02	<b>1,350219E-04</b>	8,608718E-05
3	2,952937E-06	9,326242E-06	<b>8,171027E-05</b>	2,687727E-05
4	1,550210E-07	1,649954E-06	<b>1,924102E-05</b>	4,919075E-06
5	5,869429E-07	1,360196E-05	<b>9,787033E-06</b>	1,019945E-07

Tabelle 4.5: Wahrscheinlichkeitsentwicklung einzelner Beobachtungssequenzen bei gleichmäßig verteiltem (G) und manuell (M) bestimmtem  $B$

Liegen Anhaltspunkte vor, so bestehen denkbare Möglichkeiten zur Ermittlung der initialen Werte in einer manuellen Segmentierung der Beobachtungssequenz

und Zuordnung zu Zuständen oder in einer Maximum-Likelihood Segmentierung mit Mittelwertbildung oder in der Anwendung des *k-means* Algorithmus [RaWJ86].

Für das im Rahmen dieser Arbeit zu entwerfenden Erkennungssystem mit einer geringen Anzahl von zu modellierenden Gesten findet die Segmentierung manuell statt.

### Manuelle Segmentierung

Aus einer modellierten oder per Kamerakopf erfassten Geste ergibt sich schrittweise die Segmentierung, die sich dem zugehörigen Referenzmodell und seinen Zuständen zuordnen lässt. Aus dieser werden dann die Parameter für  $B$  abgeleitet. Sinnvollerweise wird hier nicht mit künstlichen Gesten gearbeitet, sondern mit beispielhaft ausgeführten realen Gesten, da bei einer künstlichen Modellierung die Gefahr besteht, den Anforderungen im praktischen Einsatz nicht gerecht zu werden. Der nachfolgend beschriebene Zuordnungsprozess sollte mit mehreren Varianten jeder Geste durchgeführt werden, um eine möglichst allgemeine Modellierung zu erhalten.

Das Verfahren zur Bestimmung der Parameter für  $B$  ähnelt dem der Ermittlung der Zustandsanzahl bei den Referenzmodellen. Vor der Vektorquantisierung wird der Start/Stop-Filter auf die Gesten angewendet. Nach der Quantisierung werden wiederum gleiche aufeinanderfolgende Symbole gefiltert. Da die Entscheidung über die Zustandsanzahl der einzelnen Modelle bereits gefallen ist, muss nun die verbleibende Folge von Richtungsindizes den Zuständen des jeweiligen Modells zugeordnet werden, denn die Richtungsindizes entsprechen den Ausgabesymbolen der Hidden-Markov-Modelle.

Die Richtungsindizes werden nun gemäß der Zustandsanzahl des Referenzmodells der untersuchten Geste in Gruppen eingeteilt. Die  $k$ -te Gruppe besteht aus den Symbolen, die der  $k$ -te Zustand des Referenzmodells ausgeben kann. Alle  $b_{ij}$  für Symbole, die in dieser Gruppe nicht vorkommen, also von diesem Zustand nicht ausgegeben werden können, werden auf Null beziehungsweise auf einen Wert nahe bei Null gesetzt. So wird festgelegt, dass diese Symbole in diesem Zustand nicht emittiert werden oder dass ihre Ausgabe äußerst unwahrscheinlich ist. Für die einzelnen Symbole aus der  $k$ -ten Gruppe muss individuell abgeschätzt werden, wie wahrscheinlich ihre Ausgabe ist. Die stochastischen Rahmenbedingungen für Hidden-Markov-Modelle legen fest, dass die Zeilensummen von  $B$  eins ergeben müssen, so dass die für diesen Zustand zur Verfügung stehende Gesamtwahrscheinlichkeit von eins nun auf die Symbole der Gruppe  $k$  gemäß der Häufigkeit ihres Vorkommens aufgeteilt werden muss.

Verfährt man derart für jeden Zustand des Referenzmodells, so erhält man die geschätzte Matrix  $B$ . Zu beachten bei diesem Vorgehen ist, dass es sich nur um

eine ungefähre Zuordnung handeln kann, die zwangsläufig ungenau ist. Eine äquidistante Aufteilung der Folge von Richtungsindizes ist schon deswegen oft unmöglich, weil die Anzahl der Zustände des Referenzmodells dies verhindert. Bei der Vergabe der Wahrscheinlichkeiten für die  $b_{ij}$  ist deswegen zu berücksichtigen, dass die beschriebene Gruppierung in der Regel nicht völlig korrekt sein kann und eine zu restriktive Wertevergabe, also das völlige Ausschließen von Symbolen durch Nullsetzung, zu vermeiden ist.

### Einschränkungen

Erschwerend kommt hinzu, dass eine Überprüfung und Bewertung der Parameter von  $B$  ausgesprochen problematisch ist. Aufgrund der Tatsache, dass der stochastische Prozess der Zustandsübergänge in einem Hidden-Markov-Modell verdeckt abläuft und nur indirekt über die ausgegebenen Symbole zu beobachten ist, kann nur über den Viterbi-Algorithmus auf die wahrscheinlichste Zustandsfolge rückgeschlossen werden. Die manuelle Vergabe von Werten für  $B$  ist also notwendigerweise ungenau, kann das reale Geschehen nur teilweise modellieren und ist deswegen auch nur Basis des Trainingsprozesses und kein Ersatz für diesen. Abbildung 4.11 veranschaulicht den geschilderten Vorgang in seinen Einzelschritten.

Bei der Vergabe initialer Werte für  $B$  muss wie beschrieben darauf geachtet werden, dass nicht zu viele der  $b_{ij}$  auf Null gesetzt werden. Aufgrund der Eigenschaft des Baum-Welch Algorithmus, Nullwerte unverändert beizubehalten, besteht bei einer zu großen Anzahl von Nullwerten die Gefahr, die Erkennungsfähigkeit des Modells unnötig einzuschränken. Die Tatsache, dass ein bestimmtes Ausgabesymbol in einem gewissen Zustand zum Trainingszeitpunkt nicht ausgegeben wurde, bedeutet nicht, dass es auch zur Erkennungszeit keine solchen Symbole geben wird. Ein Auftauchen von Symbolen in einer Beobachtungssequenz  $O$ , deren Ausgabewahrscheinlichkeiten in der Matrix  $B$  des Modells  $\lambda$  Null sind, führt zu einem sehr niedrigen Wert von  $P(O|\lambda)$ . Die beobachtete Sequenz wird also als sehr unwahrscheinlich für das betrachtete Modell eingeschätzt.

Aus diesem Grund werden Werte von  $B$  für Ausgabewahrscheinlichkeiten, die zwar extrem unwahrscheinlich, prinzipiell aber möglich sind, auf einen Wert gesetzt, der auch beim Training nicht unter eine festzulegende Schwelle von  $\varepsilon$  fällt.

Ein Aufstellen initialer Parameter für das Schwellwertmodell entfällt, da es aus den Parametern der Referenzmodelle und einigen Anpassungen entsteht.

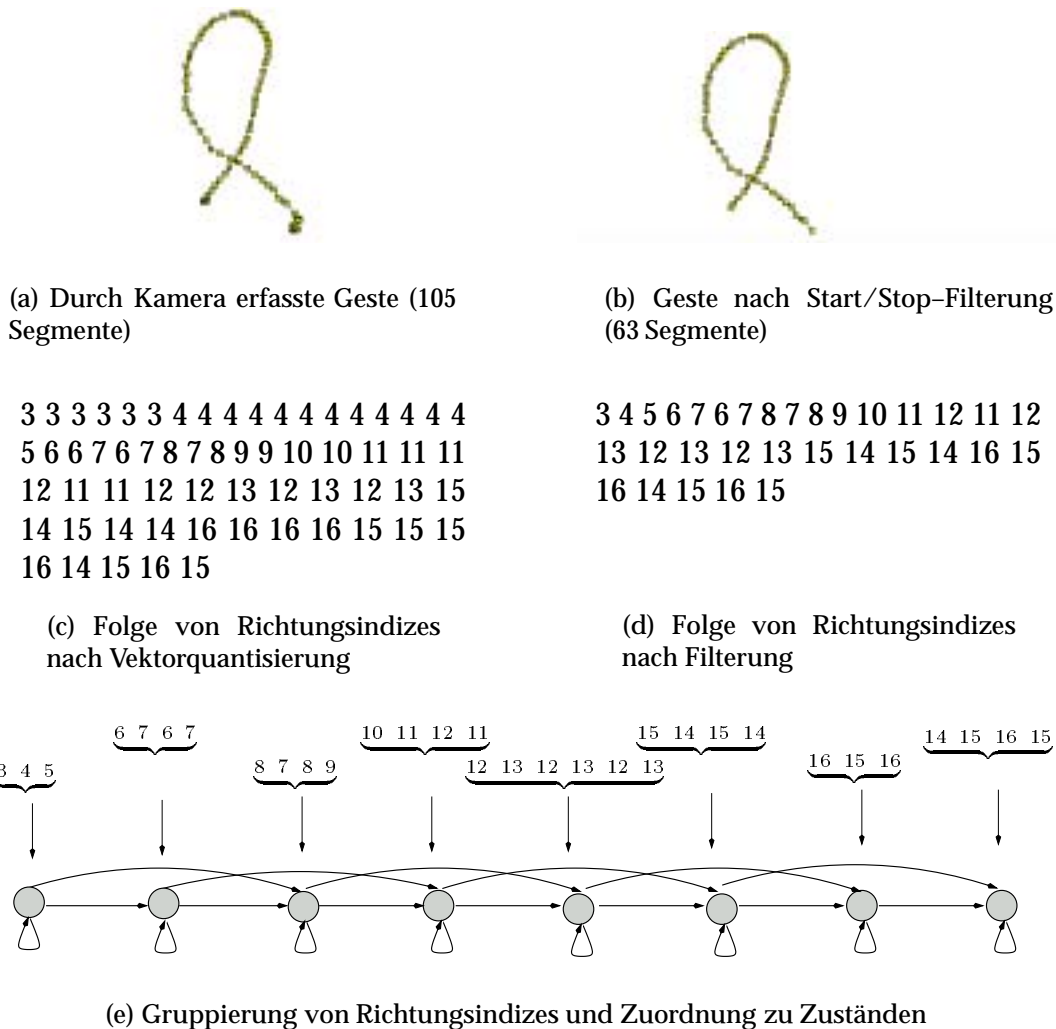


Abbildung 4.11: Filterung und Gruppierung einer Beobachtungssequenz als Vorbereitung zur Bestimmung der Werte für  $B$

#### 4.4.4 Training der Referenzmodelle

Sobald die Entwurfsentscheidungen bezüglich der Hidden-Markov-Modelle getroffen und wie in den vorherigen Abschnitten beschrieben umgesetzt worden sind, kann mit dem Training der Referenzmodelle begonnen werden.

Bei der Auswahl der Trainingsdaten ist zu beachten, dass ihr Umfang ausreichend groß ist. Da die zum Training verwendeten Beobachtungssequenzen endlich sind, kann ein zu geringes Auftreten von bestimmten Ereignissen möglich sein, als dass eine gute Bestimmung der Modellparameter abgegeben werden kann [Rabi89].

Um dem entgegenzuwirken, gibt es zwei Möglichkeiten. Einerseits kann die Menge der zum Training verwendeten Beobachtungssequenzen vergrößert werden. Falls dies nicht machbar oder nicht ausreichend ist, läßt sich andererseits auch das Modell vereinfachen, etwa indem man die Anzahl seiner Zustände verringert. Der letzten Möglichkeit sind allerdings Grenzen gesetzt, da in der Regel das Modell im Hinblick auf bestimmte physikalische Gegebenheiten entworfen wurde und nicht beliebig geändert werden kann.

Für den Erkennungsprozess und das zugrunde liegende Training muss eine Entscheidung über das Zulassen von Beobachtungssequenzen unterschiedlicher Länge oder deren Begrenzung auf ein bestimmtes Maß getroffen werden, da diese Fälle unterschiedlich behandelt werden müssen und Auswirkungen auf das Erkennungsverhalten haben.

Abschließend ist die Möglichkeit des Trainings mehrfacher Beobachtungssequenzen wesentliche Voraussetzung für einen späteren Erkennungserfolg (Abschnitt 4.4.4.1).

#### 4.4.4.1 Training mit mehrfachen Beobachtungssequenzen

Beschäftigt man sich mit dem Training der Referenzmodelle, so stößt man auf das Problem, mehrfache Beobachtungssequenzen trainieren zu müssen.

Die Schwierigkeit besteht darin, dass die Referenzmodelle in ihrer Struktur als Links-Rechts-Modelle mit mehr als einer Beobachtungssequenz trainiert werden müssen. Aufgrund des Übergangscharakters dieser Modelle kann aber jeder Zustand nur eine begrenzte Zahl von Symbolen emittieren, bevor ein Übergang zum nächsten Zustand vollzogen wird [Rabi89].

Ein sukzessives Training desselben Modells mit mehreren Beobachtungssequenzen hätte zur Folge, dass die Modellparameter auf die jeweils aktuelle Sequenz angepasst werden, das heißt es wird jeweils  $P(O|\lambda)$  für ein spezifisches  $O$  maximiert. Dabei werden die erlernten Werte des jeweils vorhergehenden Trainingsvorgangs aber überschrieben und der Trainingserfolg geht verloren.

Der in Abschnitt 3.2.7 eingeführte Baum-Welch-Trainingsalgorithmus basiert auf den Häufigkeiten des Eintretens bestimmter Ereignisse. Er lässt sich dergestalt erweitern, dass die individuellen Häufigkeiten jeder Beobachtungssequenz summiert werden und so alle Sequenzen gemeinsam trainiert werden.

Liegt also die mehrfache Beobachtungssequenz  $O$  zum Training vor, mit

$$O = (O^{(1)}, O^{(2)}, \dots, O^{(k)}) \quad (4.7)$$

und

$$\mathbf{O}^{(k)} = \left( \mathbf{O}_1^{(k)}, \mathbf{O}_2^{(k)}, \dots, \mathbf{O}_{T_k}^{(k)} \right) \quad (4.8)$$

als  $k$ -te Sequenz der Länge  $T_k$ , so kann unter der Annahme, dass jede Sequenz von jeder anderen unabhängig ist, der Trainingsvorgang durchgeführt werden. Er hat zum Ziel, die Parameter des Referenzmodells  $\lambda$  so zu bestimmen, dass die Gesamtwahrscheinlichkeit in (4.9) maximal wird.

$$P(\mathbf{O}|\lambda) = \prod_{k=1}^K P(\mathbf{O}^k|\lambda) \quad (4.9)$$

$$= \prod_{k=1}^K P_k \quad (4.10)$$

Für die Werte der Matrix  $A$  der Übergangswahrscheinlichkeiten und der Matrix  $B$  der Symbolausgabewahrscheinlichkeiten ergibt sich dann unter Zuhilfenahme der Vorwärts- und Rückwärtvariablen (Abschnitt 3.2.5):

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \quad (4.11)$$

$$\bar{b}_j(k) = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{\substack{t=1 \\ \text{mit } o_t=v_t}}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}. \quad (4.12)$$

Die Anfangswahrscheinlichkeitsverteilung  $\pi$  bleibt unangetastet, da sie durch die für Links-Rechts-Modelle geltenden Beschränkungen festgelegt ist.

Für das Schwellwertmodell besteht keine Notwendigkeit des Trainings, da es aus den schon trainierten Referenzmodellen entsteht und deren Parameter zum großen Teil übernimmt. Spezifische Anpassungen sind zwar nötig (Abschnitt 4.4.5), ein explizites Training aber entfällt.

In Tabelle 4.6 ist die Entwicklung der Wahrscheinlichkeiten beim Training von drei Beobachtungssequenzen unterschiedlicher Länge protokolliert. Sowohl die Erkennungswahrscheinlichkeiten für die einzelnen Sequenzen  $P(O_i|\lambda)$  als auch die zu maximierende Gesamtwahrscheinlichkeit  $P(O|\lambda)$  sind aufgelistet. Dabei sind die Maxima der einzelnen Entwicklungen hervorgehoben.

Iteration	$P(O_1 \lambda)$	$P(O_2 \lambda)$	$P(O_3 \lambda)$	$P(O \lambda)$
1	0,000000	0,000000	0,000000	0,000000E+00
2	0,000000	0,000000	0,000244	0,000000E+00
3	0,000694	0,030582	0,170231	3,612967E-06
4	<b>0,000714</b>	0,034632	0,188469	4,660320E-06
5	0,000670	<b>0,036613</b>	0,212085	5,202596E-06
6	0,000606	0,036609	0,237448	<b>5,267797E-06</b>
7	0,000546	0,035188	0,261297	5,020207E-06
8	0,000495	0,033012	0,284557	4,649929E-06
9	0,000449	0,030349	0,310289	4,228215E-06
10	0,000399	0,027218	0,341618	3,709965E-06
11	0,000343	0,023661	0,381338	3,094834E-06
12	0,000277	0,019873	0,431461	2,375116E-06
13	0,000205	0,016205	0,492063	1,634646E-06
14	0,000132	0,013001	0,560154	9,612982E-07
15	0,000070	0,010368	0,630589	4,576563E-07
16	0,000030	0,008100	0,699149	1,698932E-07
17	0,000002	0,005926	0,763751	4,525988E-08
18	0,000000	0,003842	0,823097	6,324677E-09
19	0,000000	0,002109	0,875698	0,000000E+00
20	0,000000	0,000942	0,919741	0,000000E+00
21	0,000000	0,000329	0,953532	0,000000E+00
22	0,000000	0,000087	0,976456	0,000000E+00
23	0,000000	0,000017	0,989765	0,000000E+00
24	0,000000	0,000002	0,996228	0,000000E+00
...	...	...	...	...
31	0,000000	0,000000	<b>1,000000</b>	0,000000E+00

Tabelle 4.6: Wahrscheinlichkeitsentwicklung beim Training von drei Beobachtungssequenzen

Man erkennt deutlich, dass ab Iterationsschritt sechs, bei dem das Gesamtmaximum erreicht wird, nur noch die Wahrscheinlichkeit für die Beobachtungssequenz  $O_3$  maximiert wird, während die Wahrscheinlichkeit für  $O_1$  und  $O_2$  sowie die Gesamtwahrscheinlichkeit gegen Null gehen. Die guten Werte für  $O_3$ , bei der es sich um die längste der drei Sequenzen handelt, gehen auf Kosten des Gesamterfolges. Aus diesem Grunde bricht der Algorithmus nach Schritt sechs ab, weil hier das Gesamtmaximum erreicht wurde. Eine individuelle Optimierung pro Beobachtungssequenz führt nicht zu einem optimalen Gesamtergebnis, da sich das Gesamtmaximum nicht aus dem Produkt der einzelnen Maxima ergibt.

#### 4.4.4.2 Variable versus feste Längen von Beobachtungssequenzen

Abhängig von technischen Gegebenheiten kann die Menge der Eingabedaten unter Umständen deutlich variieren. Ist das Kamerasystem in der Lage, eine große Anzahl von Bildern pro Sekunde aufzunehmen, so bedeutet dies, dass auch viele Richtungsvektoren beim Erkennungsvorgang berücksichtigt werden müssen. Die Tatsache, dass dieser Faktor unbeständig ist, hat zur Folge, dass ein und dieselbe Geste in mehreren Fällen durch eine völlig unterschiedliche Anzahl von Richtungsindizes dargestellt werden kann. Eine Maßnahme, die diesen Effekt abschwächen, nicht aber beseitigen kann, ist die bereits beschriebene Filterung.

Je mehr eine menschliche Geste dem modellierten Ideal ähnelt, desto geringer ist die Anzahl der Richtungsindizes, die diese Geste beschreiben. Schon eine leichte Wellenbewegung der Hand anstelle einer geradlinigen produziert eine Vielzahl von wechselnden Indizes, wo nach der Filterung im Idealfall nur ein einziger vorhanden sein sollte. Dies ist insofern problematisch, als dass diese Bandbreite unterschiedlicher Segmentierungen einer Geste beim Training und beim Testen berücksichtigt werden muss.

Trainiert man die Referenzmodelle nun mit mehrfachen Beobachtungssequenzen unterschiedlicher Länge, so lässt sich folgendes Phänomen beobachten: Je länger die Trainingssequenzen sind, desto geringer wird die Fähigkeit zur Erkennung der kurzen Sequenzen, ohne dass allerdings die der längeren besser wird. Insgesamt sinkende Erkennungswahrscheinlichkeiten sind zu erwarten, da eine breit gefächerte allgemeine Fähigkeit zur Erkennung zu Lasten der Erkennung einzelner Gesten geht. Im geschilderten Fall sollte also ein Absinken der Wahrscheinlichkeiten bei kurzen Sequenzen, sobald die längeren hinzugenommen werden, mit einem ähnlichen starken Anstieg bei den längeren einhergehen. Abbildung 4.12 zeigt, dass dies nicht der Fall ist. Hier wird ein Hidden-Markov-Modell mit drei Zuständen und gleichmäßig verteilten Werten in  $A$  und  $B$  jeweils mit einer steigenden Anzahl von Beobachtungssequenzen trai-

niert. Tabelle 4.12 (a) zeigt die Erkennungswahrscheinlichkeiten für unterschiedlich lange Sequenzen, Tabelle 4.12 (b) für gleich lange Sequenzen.  $N$  bezeichnet dabei die Anzahl der trainierten Sequenzen,  $L$  ihre Länge.

N	L	$P(O \lambda)$
1	3	1,000000E-00
2	3, 6	7,059579E-04
3	3, 6, 12	2,708259E-11
4	3, 6, 12, 32	1,195539E-31

N	L	$P(O \lambda)$
1	12	1,673044E-03
2	12, 12	1,819442E-09
3	12, 12, 12	2,162391E-16
4	12, 12, 12, 12	6,200790E-26

(a) Mehrfache Sequenzen mit unterschiedlicher Länge

(b) Mehrfache Sequenzen mit gleicher Länge

Abbildung 4.12: Entwicklung der Erkennungswahrscheinlichkeiten beim Training mit mehrfachen Beobachtungssequenzen unterschiedlicher und gleicher Länge

Tabelle 4.12 (a) verdeutlicht, dass die Hinzunahme längerer Sequenzen die Erkennungswahrscheinlichkeit deutlich verringert. Das Training von Sequenzen identischer Länge führt dagegen zu insgesamt weniger guten Ergebnissen, liefert aber bei steigender Anzahl von Sequenzen bessere Werte (Tabelle 4.12 (b)).

Der Einsatz von Sequenzen gleicher Länge wird auch von [LeKi99] favorisiert, ist aber unter Umständen problematisch und abhängig vom Gesamtkonzept. Sequenzen gleicher Länge haben den Vorteil, dass sie ein einheitliches Training und Testen ermöglichen und die eingangs geschilderte Problematik umgangen wird.

Dennoch weist das Konzept einheitlicher Längen einige Schwächen auf:

1. Die Repräsentation einer Geste entspricht nur in den seltensten Fällen der Beobachtungssequenz mit einer fixen Länge. Diese Länge muss also zwangsläufig ein Mittelwert aller zu erkennender Gesten sein. Damit wird man aber den Charakteristika der einzelnen Gesten keineswegs gerecht. Während einzelne besser durch eine geringe Anzahl von Symbolen dargestellt werden, ist für andere eine hohe Anzahl zwingend erforderlich. Die Verwendung eines einheitlichen Wertes senkt also die Erkennungsqualität für alle Gesten.
2. Das Erreichen eines bestimmten Richtwertes für die Anzahl von Beobachtungssymbolen ist dahingehend problematisch, dass der Einsatz eines Filterungsmechanismus bei Gesten, die sich aus Geraden zusammensetzen, weniger Symbole liefert, als erwartet. Der Einsatz eines Filters ist aber

wünschenswert, um redundante Informationen zu entfernen. Auf der anderen Seite müssen Beobachtungssequenzen, deren Länge die des festgelegten Wertes überschreitet, gekürzt werden. Da ein einfaches Abschneiden am Anfang oder Ende die Geste verstümmelt, ist ein intelligenteres Verfahren notwendig, um die Länge auf den gewünschten Wert anzupassen. Ein solches dem Erkennungsvorgang vorgeschaltetes Verfahren ist aber aufwändig und verzögert die Erkennung.

Aus diesen Gründen werden in dieser Arbeit Beobachtungssequenzen mit variabler Länge eingesetzt. Sie ermöglichen die Modellierung individueller Charakteristika einzelner Gesten und ein Filtern zur Datenreduktion.

#### 4.4.4.3 Modelltraining

Beginnend mit Abschnitt 4.4.4 wurden die Anforderungen dargestellt und die Parameter für das Training der Referenzmodelle beschrieben. Darauf aufbauend schließt sich das eigentliche Training an. Hierzu werden von mehreren Testpersonen Gestenbeispiele zu den fünf Referenzgesten gesammelt. Diese Trainingsdaten werden nach Gesten gruppiert und die entsprechenden Referenzmodelle mit ihnen trainiert.

Tabelle 4.7 schlüsselt die verwendeten Trainingsdaten auf. Der Trainingsablauf selber gliedert sich in die folgenden Schritte:

1. Erfassung der Trainingsgesten aller Personen
2. Quantisierung aller aufgezeichneten Gesten
3. Gruppierung der Trainingsdaten nach Geste
4. Training aller Referenzmodelle mit ihren jeweiligen Trainingsdaten

Ein direkt auf die Aufnahme folgendes Training der Referenzmodelle (*Online-Training*) ist nicht möglich, da das Training mit mehrfachen Beobachtungssequenzen erfolgen muss (Abschnitt 4.4.4.1), um das Überschreiben des jeweils vorhergehenden Trainingsvorgangs zu verhindern. Es müssen also erst alle Daten gesammelt, quantisiert und gruppiert werden, bevor jedes Modell mit allen seinen Daten trainiert werden kann.

Geste Nummer eins wurde nur mit Trainingsdaten einer einzigen Person trainiert, um das Erkennungsverhalten für Probanden zu testen, von denen keine Beispielgesten vorliegen (Abschnitt 4.8).

Geste	Anzahl Personen	Gesten gesamt
1	1	14
2	2	8
3	2	9
4	2	10
5	2	7

Tabelle 4.7: Art und Anzahl der zum Training der Referenzmodelle verwendeten Gesten

#### 4.4.5 Schwellwertmodell

Im Zusammenhang mit der Modellierung der Referenzmodelle (Abschnitt 4.4.2) ergibt sich die Frage nach der Darstellung von nicht bedeutungstragenden Bewegungen, also bedeutungslosen Gesten, denen nicht fälschlicherweise ein Sinn zuerkannt werden darf. Dabei können unterschiedliche Ansätze verfolgt werden. Während im Bereich der Sprach- und Handschriftenerkennung ein Füllmodell definiert wird [WiBu92], dessen Aufgabe es ist, Laute ohne Bedeutung zu erkennen, gehen [LeKi99] einen anderen Weg. Dieser wird besprochen, da das Füllmodell mit einer endlichen Anzahl von Beispiellauten trainiert wird, im Bereich der Gestenerkennung aber ein nahezu unerschöpflicher Vorrat von bedeutungslosen Bewegungen existiert, deren Erfassung und Modellierung angesichts ihrer Masse unmöglich ist. Ein Lösungsansatz, der zwei Modelle definiert, von denen eines bedeutungstragende und das andere bedeutungslose Gesten beschreibt, ist also nicht praktikabel.

Der Einsatz einer Technik zur Erkennung bedeutungsloser Gesten ist notwendig, weil ein einfacher Test der Handbewegung auf ein Referenzmodell nicht ausreicht. Es liegt in der Natur der Hidden-Markov-Modelle, dass sie in der Regel selbst für eine Geste, die dem Modell nicht entspricht, keine Wahrscheinlichkeit von Null liefern. Stattdessen erhält man einen Wert sehr nahe bei, aber ungleich Null. Von diesem muss nun entschieden werden, ob er Erfolg oder Fehlschlag signalisiert. Hierzu wird eine Hilfestellung etwa in Form eines Schwellwertes benötigt. In [LeKi99] signalisiert zum Beispiel in einem Fall die sehr geringe Wahrscheinlichkeit von  $10^{-18}$  eine erfolgreich erkannte Geste.

Um nun Wahrscheinlichkeiten als Erfolg oder Fehlschlag charakterisieren zu können, soll ein Schwellwert eingeführt werden. Ergebnisse unter diesem Wert werden verworfen. Um dem individuellen Charakter der einzelnen Referenzmodelle gerecht zu werden, wird allerdings für jedes einzelne dieser Modelle ein Schwellwert benötigt. Wünschenswert wäre weiterhin, die einzelnen Schwellwerte flexibel wählen zu können, um auf jeden einzelnen Erkennungsvorgang gezielt zu reagieren. Legt man die Schwellwerte a priori nach dem Training der

Modelle und vor den Erkennungsvorgängen fest, so orientieren sie sich zwangsläufig an der Qualität der Trainingsdaten. Ist aber eine Erkennung in einem weniger idealen Umfeld nötig, das so nicht vorhergesehen und damit auch nicht trainiert werden konnte, so erhält man unter Umständen für Gesten, die als korrekt erkannt werden sollten, eine Wahrscheinlichkeit, die unter dem gewählten Schwellwert liegt. Um Fehlentscheidungen dieser Art vermeiden zu können, sollte der Schwellwert adaptiven Charakter haben.

Der in [LeKi99] verfolgte Ansatz erfüllt diese Anforderungen und beschreibt ein Schwellwertmodell, das sich aus Zustandskopien aller Referenzmodelle der trainierten Gesten des Systems zusammensetzt. Bei diesem Schwellwertmodell handelt es sich ebenfalls um ein Hidden-Markov-Modell, das alle Gesten erkennt, die sich aus Teilgesten der in den Referenzmustern modellierten zusammensetzen. Solche Gesten können dabei aus Teilgesten in beliebiger Reihenfolge gebildet werden.

Die vom Schwellwertmodell gelieferte Wahrscheinlichkeit erfüllt eine doppelte Funktion. Dieser Schwellwert hat den gewünschten adaptiven Charakter, denn er ist höher und damit umso besser, je mehr bekannte Teilgesten in der untersuchten Geste enthalten sind. Dabei profitiert das Schwellwertmodell von der sogenannten *internen Segmentierungseigenschaft* der Hidden-Markov-Modelle, die besagt, dass die Zustände und die Zustandsübergänge eines trainierten Modells Teilgesten eben dieses Modells repräsentieren.

Ein Überschreiten des Schwellwertes stellt also eine Art vorläufigen Treffer dar, der noch durch einen Test mit dem Referenzmodell bestätigt werden muss. Liegt die vom Referenzmodell gelieferte Wahrscheinlichkeit über dem Schwellwert, so handelt es sich um eine gültige Geste, andernfalls um eine ähnliche aber bedeutungslose Handbewegung. Die irrtümliche Klassifizierung von bedeutungslosen Handbewegungen als Gesten soll so verhindert werden.

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

Eine mögliche Architektur des Schwellwertmodells lässt sich in Gestalt eines ergodischen Modells finden, das dadurch entsteht, dass alle Zustände der Referenzmodelle übernommen und miteinander verbunden werden. So kann jeder Zustand von jedem anderen in einem einzigen Übergang erreicht und die Erkennung der Teilgesten realisiert werden. Abbildung 4.13 skizziert eine vereinfachte Struktur des Schwellwertmodells und Abbildung 4.14 den Ablauf des Erkennungsmechanismus.

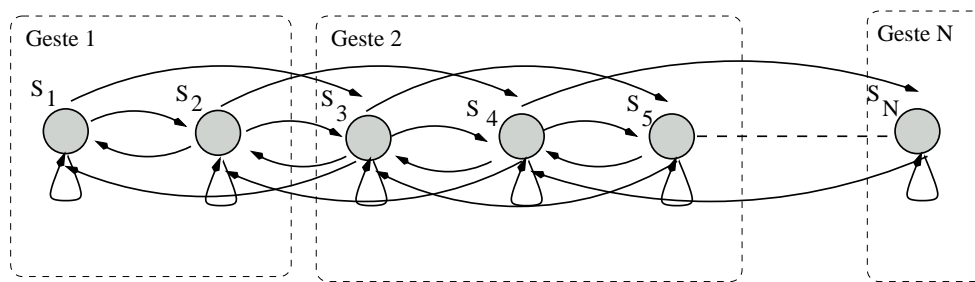


Abbildung 4.13: Struktur des Schwellwertmodells als ergodisches Hidden-Markov-Modell

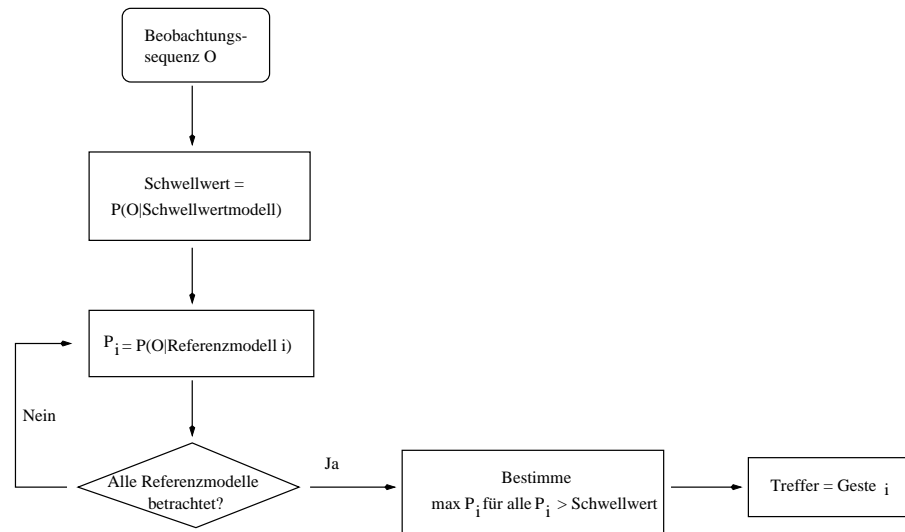


Abbildung 4.14: Schematischer Ablauf des Erkennungsmechanismus mit Hilfe des Schwellwertmodells

In das so entstehende neue Modell werden die Beobachtungswahrscheinlichkeiten für die Ausgabesymbole übernommen. Das Gleiche gilt für die Wahrscheinlichkeitswerte der Zustandsübergänge. Für alle anderen Zustandsübergangswahrscheinlichkeiten gilt:

$$a_{ij} = \frac{1 - a_{ii}}{N - 1}, \quad \forall i, j, i \neq j. \quad (4.13)$$

$N$  bezeichnet hierbei die Anzahl der Zustände,  $a_{ij}$  die Wahrscheinlichkeit eines Übergangs von Zustand  $s_i$  nach  $s_j$ .

Das Beibehalten der Wahrscheinlichkeiten für Selbstübergänge und Beobachtungen korrespondiert zur internen Segmentierungseigenschaft der Hidden-

Markov-Modelle. Es stellt sicher, dass das Schwellwertmodell jede Geste, die aus beliebigen Teilen der Referenzgesten in beliebiger Reihenfolge zusammengesetzt ist, erkennt. Durch die verringerte Zustandsübergangswahrscheinlichkeit (4.13) ist jedoch garantiert, dass die Wahrscheinlichkeit des Referenzmodells größer ist als die des Schwellwertmodells.

Die Anfangswahrscheinlichkeit  $\pi$  wird auf alle Zustände gleichermaßen verteilt, so daß gilt:

$$\pi_i = \frac{1}{N}, \quad i = 1 \dots N. \quad (4.14)$$

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

#### 4.4.5.1 Beispiel zum Aufbau eines Schwellwertmodells

Ausgehend von den Matrizen  $A_1, B_1, A_2, B_2$  in (4.15) und (4.16), die die Links-Rechts-Modelle  $\lambda_1 = (A_1, B_1, \pi_1)$ ,  $\lambda_2 = (A_2, B_2, \pi_2)$  mit je drei Zuständen beschreiben, und dem diskreten Alphabet von Beobachtungssymbolen  $V = \{0, 1, 2, 3\}$  lassen sich daraus  $A_S$  und  $B_S$  für das Schwellwertmodell ableiten:

$$A_1 = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{2}{3} \\ 0 & 0 & 1 \end{pmatrix} \quad B_1 = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{pmatrix} \quad \pi_1 = (1, 0, 0) \quad (4.15)$$

$$A_2 = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \quad B_2 = \begin{pmatrix} \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{pmatrix} \quad \pi_2 = (1, 0, 0) \quad (4.16)$$

$$A_S = \begin{pmatrix} \frac{1}{3} & \frac{2}{15} & \frac{2}{15} & \frac{2}{15} & \frac{2}{15} & \frac{2}{15} \\ \frac{2}{15} & \frac{1}{3} & \frac{2}{15} & \frac{2}{15} & \frac{2}{15} & \frac{2}{15} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{2} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{2} & \frac{1}{10} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad B_S = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{pmatrix} \quad (4.17)$$

Die Matrix  $B_S$  entsteht aus der Zusammensetzung von  $B_1$  und  $B_2$ . Bei  $A_S$  handelt es sich um eine  $6 \times 6$  Matrix, die die Zustandsübergangswahrscheinlichkeiten des ergodischen Modells bestehend aus sechs Zuständen enthält, das aus den beiden einzelnen Links-Rechts-Modellen mit je drei Zuständen hervorgegangen ist. Die Diagonalelemente von  $A_S$  sind die Diagonalelemente von  $A_1$  und  $A_2$ , die restlichen werden gemäß (4.13) berechnet.

## 4.5 Mehrstufiges Filterkonzept

In den Abschnitten 4.4.4 und 4.6 wurden bereits die angedeuteten Auswirkungen großer Modelle oder langer Beobachtungssequenzen auf den Berechnungsaufwand besprochen. Lange Beobachtungssequenzen führen erstens zu größeren Referenzmodellen, die die Geste, die durch diese Sequenz beschrieben wird, modellieren. Weiterhin machen lange Sequenzen ein häufigeres Testen mit den Referenzmodellen nötig. Dieses Testen findet außerdem mit den größeren Modellen statt, so dass der Aufwand um ein Vielfaches steigt.

Um eben diesen Aufwand zu verringern und den Echtzeitanforderungen gerecht zu werden, wird ein mehrstufiges Filterkonzept eingeführt, welches das Datenaufkommen frühzeitig beschränken soll. Es besteht aus einer Menge von hintereinander geschalteten Filtern, die zu verschiedenen Zeitpunkten in den Erkennungsablauf eingefügt sind und den Rechenaufwand im jeweils folgenden Schritt reduzieren.

Die zum Einsatz kommenden Filter sind in der Reihenfolge ihrer Anwendung:

1. Start/Stop Filter
2. Nachbarschaftsfilter
3. Gleichheitsfilter

Der Start/Stop Filter entfernt Häufungen von Punkten zu Beginn und am Ende der Geste. Diese Häufungen bezeichnen den Anfangs- und den Endpunkt und können auf jeweils einen einzigen Punkt reduziert werden.

Aufgabe des Nachbarschaftsfilters ist es, Punkte zu verwerfen, deren Entfernung zu ihrem Vorgänger unter einem bestimmten Grenzwert liegt. Liefert der Kamerakopf eine große Anzahl von Bildern pro Sekunde, so ist die Wahrscheinlichkeit, dass sich von einem Bild zum nächsten wesentliche Fortschritte in der Bewegung ergeben haben, gering. Eine solche Filterung entfernt einerseits nicht übermäßig viele der eingehenden Informationen, andererseits ist die gewonnene Leistungssteigerung aber doch erheblich. Die Anwendung des Nachbarschaftsfilters bedeutet also einen weiteren Schritt im Hinblick auf eine Begrenzung auf den Anteil der relevanten Informationen. Zur flexiblen Anpassung an wechselnde Rahmenbedingungen kann die Granularität des Nachbarschaftsfilters beliebig verändert und so das Maß der gefilterten Informationen reguliert werden.

Identische aufeinander folgende Richtungsindizes werden vom Gleichheitsfilter dergestalt reduziert, dass die Folge der Indizes nach der Filterung paarweise verschieden ist. Da jeder Richtungsindex aus einem Richtungsvektor entsteht und dessen Orientierung repräsentiert, reicht ein einziger Index aus, um eine Bewegungsrichtung zu verdeutlichen. Mehrere aufeinander folgende gleiche Indizes sind also redundant und können auf einen einzigen abgebildet werden. Dies entspricht dem Zusammenfassen linear abhängiger Richtungsvektoren. Da für den Verlauf der Geste nur die Richtung der Handbewegung wesentlich ist, kommt nur der Folge von Richtungsänderungen Bedeutung zu. Dies entspricht einer Sequenz paarweise verschiedener Richtungsindizes.

Über die bereits erwähnten positiven Eigenschaften wie eine verringerte Anzahl von Testvorgängen und aus weniger Zuständen bestehende Referenzmodelle hinaus ergibt sich ein weiterer Vorteil. Durch die der Erkennung vorgeschaltete Verarbeitung der Eingabedaten ist gewährleistet, dass die Erkennung relativ unabhängig von der verwendeten Hardware, insbesondere des Kamerasystems, ist. Eine hohe Bilderfassungsrate ist wesentlich für die Handverfolgung, um die Hand bei der Bewegung nicht zu verlieren. Andererseits enthält diese Vielzahl an Informationen einen hohen redundanten Anteil, der vor dem Erkennungsprozess herausgefiltert werden kann. Abbildung 4.15 zeigt den Wirkungsgrad der verschiedenen Filter auf.

In jeder der drei Tabellen ist die initiale Anzahl der Segmente der Trajektorie angegeben, wie sie sich als Folge der ermittelten Handschwerpunkte ergibt. Weiterhin ist die Anzahl der Segmente beziehungsweise Symbole nach Anwendung der jeweiligen Filter aufgeführt. Die letzte Spalte zeigt die Reduzierung (R) der Informationen vor der Filterung auf die Menge nachher in Prozent. Bei allen

Geste	Segmente vorher	Segmente nachher	R in %
1	64	40	38
2	55	41	25
3	75	39	48
4	65	33	49
5	77	57	26

(a) Wirkung des Nachbarschaftsfilters ohne Start- und Endzustände

Geste	Segmente vorher	Segmente nachher	R in %
1	136	68	50
2	110	95	14
3	126	78	84
4	107	71	34
5	105	83	21

(b) Wirkung des Start/Stop-Filters und des Nachbarschaftsfilters mit Codierung der Start- und Endzustände

Geste	Segmente initial	Segmente nach S/T- und N-Filter	Symbole nach G-Filter	R in %
1	138	19	6	96
2	134	17	7	95
3	85	28	14	50
4	118	34	15	44
5	113	34	19	56

(c) Einsatz aller Filter, wobei S/T den Start/Stop-Filter, N den Nachbarschaftsfilter bezeichnet, G den für gleiche aufeinander folgende Symbole

Abbildung 4.15: Ergebnisse beim Filtern einer Beobachtungssequenz

Werten handelt es sich um Mittelwerte. Der Grenzwert für den Nachbarschaftsfilter beträgt fünf Pixel.

Tabelle 4.15 (a) zeigt die Wirkung des Nachbarschaftsfilters, wobei die Ausgangsdaten bezüglich der Start- und Endpunkthäufung bereinigt sind, also nur die reine Geste repräsentiert wird.

In Tabelle 4.15 (b) kommt der Start/Stop-Filter hinzu. Die prozentuale Verbesserung sollte hier erkennbar deutlicher sein als im vorhergehenden Fall. Abweichungen sind durch eine unruhige Hand möglich, da hierdurch die Punkte, die den Start- oder Endzustand signalisieren sollen, nicht nahe genug beieinander liegen und als Teil der Geste missverstanden werden können. Der Filterungserfolg ist dann entsprechend geringer.

Abschließend beschreibt Tabelle 4.15 (c) die Wirkung beim sukzessiven Einsatz aller drei Filter. Das Datenaufkommen wird dabei um 44 bis 96 Prozent reduziert. Deutlich zu erkennen ist die unterschiedliche prozentuale Veränderung zwischen den Gesten eins und zwei einerseits und drei bis fünf andererseits.

Dies ist darauf zurückzuführen, dass es sich bei den ersten beiden um Gesten handelt, die nur aus Geraden zusammengesetzt sind. Solche Gesten lassen sich durch wesentlich weniger Richtungsindizes beschreiben als solche, die viele Bögen enthalten. Im Idealfall reicht für jede Gerade sogar ein einziger Richtungsindex aus. Abbildung 4.16 stellt den Wirkungsgrad der Filterung bei Einsatz aller Filter grafisch dar.

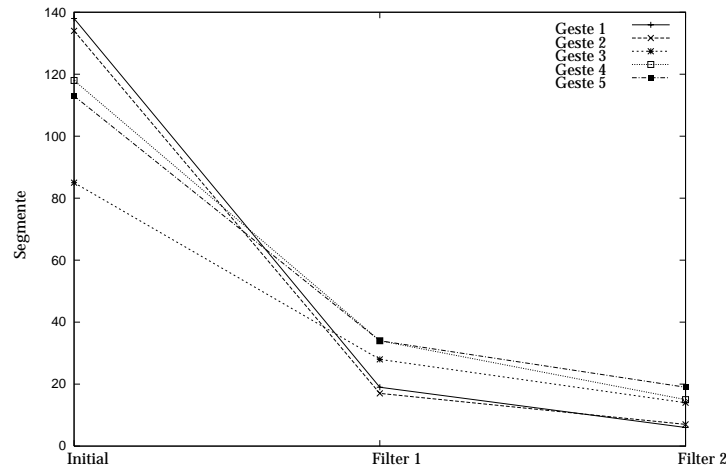


Abbildung 4.16: Abnahme der Segmentanzahl der Gestentrajektorien bei Anwendung aller Filter

Von diesem deutlichen Unterschied profitiert die hierarchische Gestenklassifikation (Abschnitt 4.6), weil hierdurch eine Einordnung erleichtert wird. Der Unterschied zwischen den Gesten drei und vier im Vergleich zu Geste fünf, die jeweils in eine andere Klasse fallen (Abbildung 4.17) ist zwar bei Betrachtung der absoluten Segmentanzahl weniger deutlich, bei Auswertung der Prozentwerte aber hinreichend gut.

## 4.6 Hierarchische Gestenklassifikation

Im Gegensatz zur Handschriftenerkennung muß sich die Gestenerkennung mit dem Problem auseinandersetzen, dass ausgeführte Gesten nicht klar voneinander getrennt sind, sondern ineinander übergehen können. Während geschriebene Wörter durch Leerzeichen getrennt sind, kann auf eine Geste eine bedeutungslose Bewegung oder einen weitere Geste folgen, ohne dass die Handbewegung unterbrochen wird und die Hand eine Art Ruheposition einnimmt.

Diese Tatsache bringt die Notwendigkeit mit sich, den eingehenden Strom von Richtungsindizes zu jedem diskreten Zeitpunkt  $t$  dahingehend zu prüfen, ob

er eine bekannte Geste darstellt. Dies bedeutet, dass, abhängig von der Anzahl aufgenommener Bilder pro Sekunde, eine Vielzahl von Testvorgängen ausgeführt werden muss. Jeder dieser Testvorgänge setzt sich wiederum aus einem Erkennungstest der Richtungsindexfolge mit jedem Referenzmodell zusammen. Je nach Komplexität der Referenzmodelle erhöht sich der Berechnungsaufwand. Er steigt quadratisch mit der Anzahl der Zustände des jeweiligen Referenzmodells. Bindet man zusätzlich das Schwellwertmodell (Abschnitt 4.4.5) in den Erkennungsmechanismus ein, so erhöht sich der Aufwand ein weiteres mal, insbesondere weil das Schwellwertmodell aus der Verschmelzung aller Referenzmodelle entsteht und somit eine hohe Anzahl von Zuständen aufweist.

Um den mit den einzelnen Tests verbundenen Rechenaufwand zu begrenzen und damit die Entscheidung, ob es sich bis hierher überhaupt um eine Geste handelt, zu beschleunigen, wird eine Klassifikation der Referenzgesten eingeführt, die die Anzahl der Tests deutlich reduziert. Hierbei werden die Gesten und mit ihnen ihre Referenzmodelle gemäß ihrer Komplexität hierarchisch angeordnet. Es sind drei Komplexitätsklassen vorgesehen, die sich aber jederzeit ergänzen lassen. Abbildung 4.17 veranschaulicht die Klassifizierung.

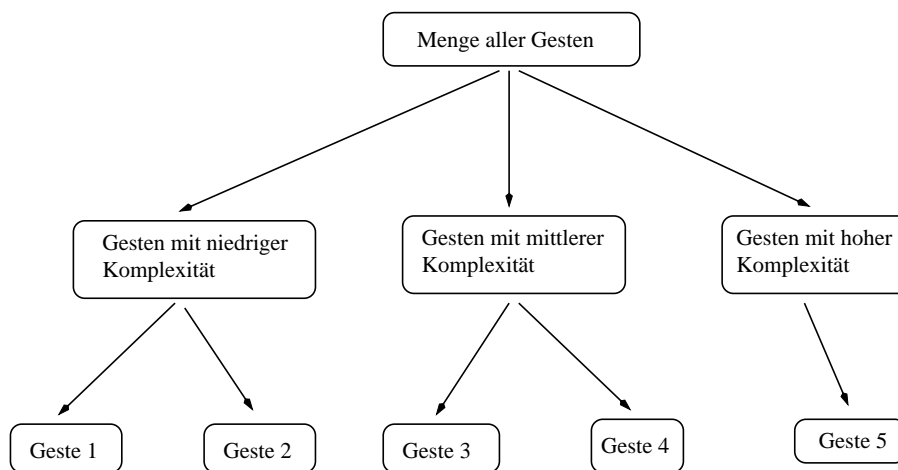


Abbildung 4.17: Klassifizierung der Referenzgesten gemäß ihrer Komplexität

Anhand der eingehenden Folge von Richtungsindizes wird nun entschieden, mit welchen Gesten der Test auf Erkennung durchgeführt werden soll. Dabei wird eine Klasse gewählt und alle darin enthaltenen Referenzmodelle getestet. Die Länge der Richtungsindexfolge ist Indiz für eine mögliche Geste. Kurze Folgen brauchen auf komplexe Modelle nicht getestet werden, da eine komplexe Geste sich auch aus einer hohen Anzahl von Richtungsindizes zusammensetzt. Beim Test kann also in diesem Fall die Klasse mit Gesten hoher Komplexität außer Acht gelassen werden. Mit dem kontinuierlichen Anwachsen der Rich-

tungsindexfolge bei noch nicht erkannter Geste kann wiederum die Klasse einfacher Gesten außen vor gelassen werden, da eine einfache Geste auch nur durch eine begrenzte Anzahl von Richtungsindizes beschrieben wird. In nicht eindeutig entscheidbaren Grenzfällen können auch die Gesten zweier Klassen getestet werden.

Die Länge der erfassten Richtungsindexfolge kann als Kriterium für die Komplexität herangezogen werden, da der Erkennung ein Filter vorgeschaltet ist, der vor Verfälschungen schützt. Ein Innehalten bei der Handbewegung beziehungsweise eine extrem langsame Bewegung produziert auch bei einer einfachen Geste eine hohe Anzahl von Richtungsindizes. Um hier irreführende Schlussfolgerungen zu vermeiden, entfernt der Filter gleiche aufeinander folgende Richtungsindizes. Auf diese Art und Weise gelangt man wieder zu einer Folgenlänge, die charakteristisch für die zugrunde liegende Geste ist. Des Weiteren bringt die Anwendung des Filters einen deutlichen Effizienzgewinn mit sich, da jeder herausgefilterte Richtungsindex einen gesparten Testvorgang bedeutet, der sich wiederum aus mehreren Einzeltests zusammensetzt (Abschnitt 4.5).

Die Vorteile dieses Klassifikationsschemas liegen zum Einen im Bereich der Rechenökonomie, zum anderen in seiner dynamischen Erweiterbarkeit. Weiterhin können die Modelle im Hinblick auf ihre Zustandsanzahl gezielt an die zu repräsentierenden Gesten angepasst werden, womit sich die in Abschnitt 4.4.2 geschilderte Problematik fester Beobachtungssequenzen vermeiden läßt. Darüber hinaus erlaubt dieses Schema die Erweiterung durch Einbindung des Schwellwertmodells. Anstelle eines globalen Schwellwertmodells wird dann für jede Klasse ein eigenes angelegt, das zusätzliche Zuverlässigkeit bei der Erkennung liefern kann. Die Größe und der damit verbundene Rechenaufwand bleiben vertretbar, da das Modell nur aus den Referenzmodellen der jeweiligen Klasse entsteht.

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

Ausgehend von einem Fundus aus Beispielgesten muss eine Klasseneinteilung auf Grundlage der Länge der Richtungsindexfolgen festgelegt werden. Tabelle 4.8 zeigt eine Übersicht einer solchen Menge von Beispielgesten. Für jede Geste ist die Länge der kürzesten und der längsten Richtungsindexsequenz aufgeführt, die zum Training der Referenzmodelle verwendet wurden. Abschließend ist wiederum für jede Geste das Intervall beschrieben, das auf Grundlage der Sequenzlängen gewählt wurde.

Geste	Klasse	Min. Länge	Max. Länge	Intervall
1	1	3	13	[1,11]
2		4	12	
3	2	9	19	[12,17]
4		7	18	
5	3	16	23	[18,∞)

*Tabelle 4.8: Verschiedene Sequenzlängen für die Gesten, mit deren Hilfe die Grundlage für die hierarchische Gestenklassifikation bestimmt wird*

Als Intervallgrenzen kommen nicht immer die Länge der kürzesten Sequenz der Klasse und die Länge der längsten Sequenz in Frage, da sich die Intervalle dann überlappen können. Wählt man diesen Ansatz, so erhält man als Intervall für Klasse eins [3,13] und für Klasse zwei [7, 19]. Da aber Voraussetzung für die Klassifizierung einer Geste die eindeutige Zuordnung zu einer Klasse ist, muss ein solches Überlappen verhindert werden. Hierbei hilft eine genauere Analyse der Sequenzen, deren Länge in Tabelle 4.8 dargestellt ist. Sind die Sequenzen in Klasse eins mehrheitlich kürzer als 12 und die in Klasse zwei meistens länger als 11, so können die Intervallgrenzen wie dargestellt festgelegt werden. Sequenzen, die in diese Klassen gehören, aber länger beziehungsweise kürzer sind und deswegen falsch zugeordnet würden, werden zur Erreichung einer eindeutigen Entscheidung in Kauf genommen. Die hier dargestellten Intervalle sind auch diejenigen, die den Versuchen in Abschnitt 4.8 zugrunde liegen.

## 4.7 Gesamtablauf

Mit den in den vorangegangenen Abschnitten eingeführten Techniken und Konzepten lässt sich nun das Erkennungssystem als Ganzes beschreiben. Dabei sieht der Gesamtablauf aus, wie in Abbildung 4.18 dargestellt.

Wie bereits im Einzelnen in den jeweiligen Abschnitten beschrieben, setzt sich der Gesamtablauf aus den Schritten Handverfolgung, Quantisierung und Klassifikation mit mehreren dazwischengeschalteten Filtern zusammen. Während die ersten beiden Schritte für die Akquirierung und Vorverarbeitung der Daten zuständig sind, findet die eigentliche Gestenerkennung unter dem Punkt Klassifikation statt. Hier werden die eingehenden Beobachtungssequenzen mit den Referenzmodellen getestet und die abschließende Entscheidung über Erfolg oder Misserfolg getroffen.

Die erfolgreiche Gestenerkennung steht und fällt mit einer zuverlässigen Endpunkterkennung. Im Gegensatz zur Handschrifterkennung, bei der die Tren-

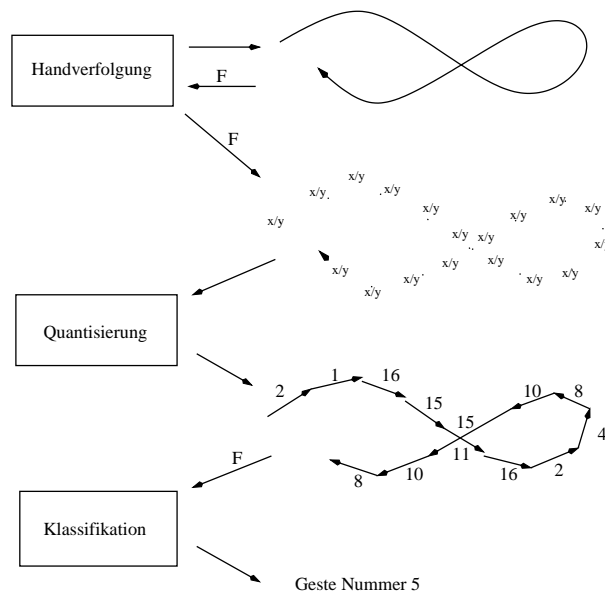


Abbildung 4.18: Der Gesten-Erkennungsprozess mit seinen Einzelschritten im Überblick. Der Einsatz von Filtern ist durch ein „F“ gekennzeichnet.

nung der Wörter durch Leerzeichen vorgegeben ist, gibt es solche expliziten Endesignale bei Handbewegungen nicht. Trotzdem muss eine Geste von der eventuell nachfolgenden bedeutungslosen Handbewegung getrennt werden können. Zur Endpunkterkennung einer Geste bieten sich zwei Möglichkeiten an:

- Testen der bis hierher vorliegenden Daten mit den Referenzmodellen zu jedem diskreten Zeitpunkt  $t$ .
- Einführung von speziellen Signalen, um den Beginn und das Ende einer Geste zu verdeutlichen. Der Test erfolgt nach Erhalt des Ende-Signals.

Bei dem realisierten Erkennungssystem ist die Wahl auf die zweite Möglichkeit gefallen, um eine zukünftige Erweiterung um die Erkennung dreidimensionaler Gesten zu ermöglichen (Abschnitt 5.2.2). Die hierzu nötigen Verfahren setzen eine Kenntnis der gesamten Trajektorie, die die Geste repräsentiert, voraus. Nur so können durch Analyse die beiden wichtigsten Dimensionen ermittelt und die Geste in die durch diese aufgespannte Ebene abgebildet werden. Ein Testen zu jedem diskreten Zeitpunkt schließt also den Einsatz solcher Verfahren in vertretbarer Rechenzeit aus.

Zu beachten ist, daß die vorgestellten Verfahren der Filterung und der hierarchischen Klassifikation ihre Fähigkeiten zur Datenreduktion erst bei Testvorgängen zu jedem Zeitpunkt  $t$  voll ausspielen. Trotzdem ist aber auch im anderen

Fall die mit diesen Verfahren einhergehende Rechenzeiterparnis erstrebenswert und anwendbar.

Das Erkennungssystem läßt beide der geschilderten Möglichkeiten zur Endpunkterkennung zu, so daß zwischen ihnen gewählt werden kann.

### 4.7.1 Erkennungstests

Der durch das erkannte Endesignal ausgelöste Testvorgang kann auf mehrere Weisen umgesetzt werden. Abbildung 4.19 verdeutlicht die verschiedenen Möglichkeiten, Schwellwertmodell und Filterung mit einzubeziehen oder sogar zu kombinieren.

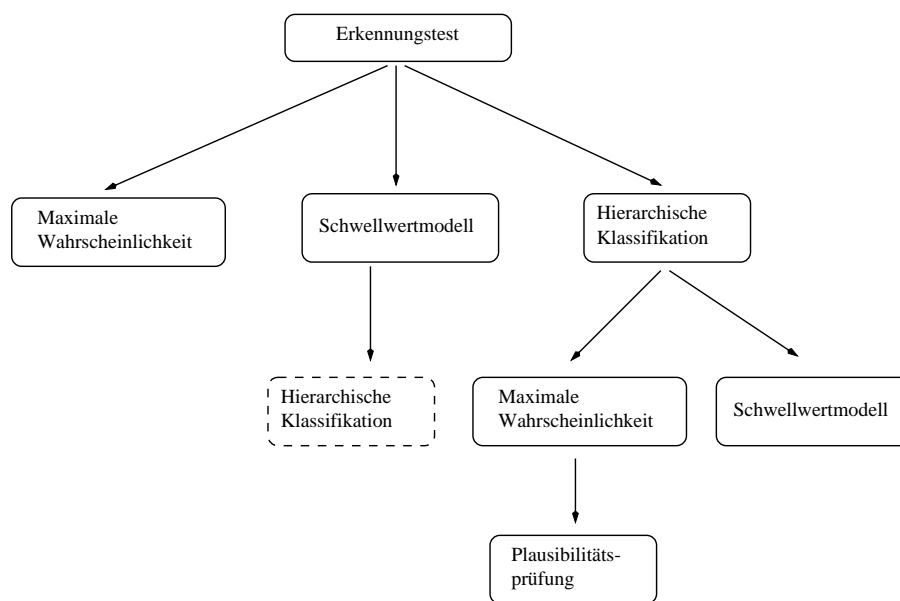


Abbildung 4.19: Testhierarchie mit möglichen Einstellungen zum Erkennungstest

Die einfachste Erkennungsvariante besteht darin, die bisherige Folge von Richtungsindizes auf alle Referenzmodelle zu testen und diejenige Geste als Treffer zu bewerten, deren Referenzmodell die höchste Wahrscheinlichkeit geliefert hat. Diese Methode hat den Vorteil, daß sie sehr tolerant gegenüber den ausgeführten Gesten ist. So können auch Gesten erkannt werden, die verglichen mit der Idealform relativ stark verzerrt oder andersartig verfälscht sind. Die durch das Referenzmodell gelieferte Erkennungswahrscheinlichkeit ist zwar unter Umständen außerordentlich gering, aber dennoch groß genug, um aus den modellierten Gesten die korrekte herauszufinden. Dieses Verfahren geht allerdings von der Prämisse aus, dass nur sinnvolle Gesten ausgeführt werden. Ist das

nicht der Fall, so wird zwangsläufig auch eine bedeutungslose Geste als korrekt erkannt. Dieser Ansatz ist nicht in der Lage, sinnvolle von sinnlosen Gesten zu trennen.

Die zweite Möglichkeit, dargestellt in der Mitte der Testhierarchie in Abbildung 4.19, erlaubt das Einbeziehen des Schwellwertmodells. Auch hier wird die Richtungsindexfolge mit jedem der Referenzmodelle getestet. Darüberhinaus wird diese Folge aber auch mit dem Schwellwertmodell getestet, das durch Verschmelzung aller Referenzmodelle entsteht (Abschnitt 4.4.5). Ein Treffer wird signalisiert, falls die größte erhaltene Wahrscheinlichkeit der Referenzmodelle auch über der des Schwellwertmodells, dem Schwellwert, liegt. Durch Einsatz dieses Schwellwertes wird die irrtümliche Erkennung bedeutungsloser Handbewegungen verhindert.

Die Nutzung des Schwellwertmodells läßt sich ferner mit der hierarchischen Gestenklassifikation verbinden. Allerdings ist eine solche Verbindung nur sinnvoll, wenn die hierarchische Klassifikation vor der Prüfung des Schwellwertes stattfindet. Insofern ist dieser Fall identisch mit dem in Abbildung 4.19 rechts außen veranschaulichten Fall.

Die hierarchische Gestenklassifikation ist mit beiden bereits vorgestellten Techniken kombinierbar. Die Anzahl der Richtungsindizes liefert den Anhalt für die Wahl der Klasse, mit deren Referenzmodellen die Indexfolge im Weiteren getestet wird. In der einfachen Variante wird hier die Geste des Referenzmodells ausgewählt, das die höchste Erkennungswahrscheinlichkeit innerhalb dieser Klasse aufweist. Dieser Fall entspricht dem ersten, bei dem ebenfalls die maximale Wahrscheinlichkeit einzige Grundlage für die Entscheidung ist. Diese Methode weist dementsprechend auch den Nachteil auf, dass sie bedeutungslose nicht von bedeutungstragenden Gesten trennen kann. Der Unterschied zur eingangs geschilderten Technik besteht in dem weniger aufwändigen Testen, da nicht alle Referenzmodelle mit einbezogen werden müssen. Es handelt sich bei der hierarchischen Gestenklassifikation primär um eine Maßnahme zur Datenreduktion, die nur im Zusammenwirken mit weiteren Techniken bei der Erkennung sinnvoll einsetzbar ist.

Eine solche ergänzende Maßnahme liegt in der bereits angedeuteten Kombination mit dem Schwellwertmodell. Nach abgeschlossener Klassifikation erfolgt innerhalb der gewählten Klasse ein Testen mit allen Referenzmodellen und einem Schwellwertmodell, das durch Verschmelzung der Referenzmodelle nur dieser Klasse entstanden ist. Im Gegensatz zum globalen Schwellwertmodell, dessen Grundlage alle Referenzmodelle sind, ist ein solches lokales Schwellwertmodell auf die jeweilige Klasse beschränkt. Um eine Geste erfolgreich zu erkennen, muss auch hier die Erkennungswahrscheinlichkeit des Referenzmodells über dem Schwellwert liegen.

Alternativ kann aber auch die Entscheidung nach maximaler Wahrscheinlichkeit um eine Plausibilitätsprüfung erweitert werden. Die Prüfung der Plausibilität bezieht sich allerdings auf die Klassifikation und unterstützt damit nur indirekt die Erkennung. Diese Plausibilitätsprüfung erfordert, dass das Referenzmodell mit der maximalen Erkennungswahrscheinlichkeit in dieser Klasse auch das mit der maximalen Wahrscheinlichkeit aller Modelle sein muss, um einen Treffer darzustellen. Wesentlicher Nachteil hierbei ist die Tatsache, dass zur Plausibilitätsprüfung der Test mit allen Referenzmodellen durchgeführt werden muß, also der Vorteil des ersparten Rechenaufwandes bei der hierarchischen Klassifikation zunichte gemacht wird.

Abschließend läßt sich sagen, dass, um eine zuverlässige Erkennung zu gewährleisten, das Schwellwertmodell eingesetzt werden sollte. Es kann um eine vorgeschaltete hierarchische Klassifikation erweitert werden, um den Rechenaufwand zu begrenzen. Dies wird allerdings erst bei einer steigenden Anzahl von zu erkennenden Gesten oder bei Testvorgängen zu allen diskreten Zeitpunkten interessant.

## 4.8 Experimentelle Ergebnisse

Nachfolgend werden die experimentellen Ergebnisse der in den vorhergehenden Abschnitten beschriebenden Verfahren zur Gestenerkennung wiedergegeben und erläutert. Stärken und Schwächen werden betrachtet und die Ergebnisse interpretiert. Tabelle 4.9 faßt noch einmal die Parameter für die Test- und die vorausgehenden Trainingsvorgänge zusammen.

Parameter	Eigenschaft
Anzahl der Gesten	5
∅ Trainingsdaten pro Geste	10
Anzahl der Erkennungsverfahren	4
∅ Anzahl Tests pro Geste und Verfahren	23

Tabelle 4.9: Parameter für das Erkennungssystem

Gemäß der in Abbildung 4.19 dargestellten Verfahren wurden Versuche mit den folgenden Erkennungsmethoden durchgeführt:

- Maximale Wahrscheinlichkeit
- Schwellwertmodell

- Hierarchische Klassifikation
- Hierarchische Klassifikation mit Schwellwertmodell

Die Erkennungsrate für die nachfolgend analysierten Testvorgänge ergibt sich folgendermaßen:

$$\text{Erkennungsrate} = \frac{\text{Anzahl erkannter Gesten}}{\text{Anzahl aller Testgesten}}. \quad (4.18)$$

Das erste der getesteten Verfahren implementiert die Erkennung gemäß der maximalen Erkennungswahrscheinlichkeit. Als Treffer wird also diejenige Geste ausgegeben, für deren Referenzmodell  $i$  gilt:

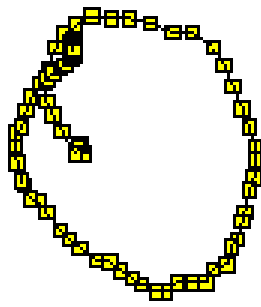
$$P(O|\lambda_i) = \max \{P(O|\lambda_j) \mid j = 1, 2, \dots, 5\}. \quad (4.19)$$

Für dies Verfahren ergibt sich für alle Gesten fast durchgängig eine Erkennungswahrscheinlichkeit von 100 Prozent (Tabelle 4.10).

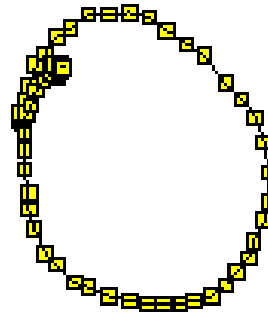
Geste	Tests	Fehler	Erkennung
1	15	0	100 %
2	15	1	93,33 %
3	15	0	100 %
4	15	0	100 %
5	15	0	100 %
Gesamt	75	1	98,66 %

*Tabelle 4.10: Erkennungserfolg der einzelnen Gesten beim Verfahren der maximalen Wahrscheinlichkeit*

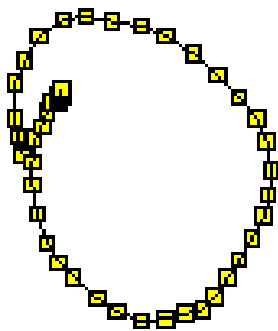
Diese außerordentlich guten Ergebnisse sind darauf zurückzuführen, dass die modellierten Referenzgesten einander kaum ähnlich sind. Dies führt wiederum dazu, dass sich selbst bei sehr geringen Erkennungswahrscheinlichkeiten die Testergebnisse für die einzelnen Modelle noch so deutlich unterscheiden, dass die richtige Geste zweifelsfrei identifiziert werden kann. Abbildung 4.20 zeigt Beispieltrajektorien für die dritte Geste, die alle korrekt erkannt wurden, obwohl die Erkennungswahrscheinlichkeit sogar bis zur Größenordnung von  $10^{-35}$  abnimmt. Im praktischen Einsatz ist eine Gestenerkennung mit der Methode der maximalen Wahrscheinlichkeit als einzigem Kriterium trotz der guten



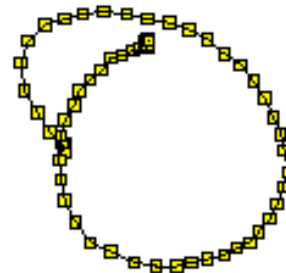
$$(a) P(\lambda|O) = 2,6845E - 21$$



$$(b) P(\lambda|O) = 6,4585E - 08$$



$$(c) P(\lambda|O) = 1,7190E - 35$$



$$(d) P(\lambda|O) = 1,0110E - 24$$

Abbildung 4.20: Trajektorien von erkannten Gesten des Typs drei mit ihren jeweiligen Erkennungswahrscheinlichkeiten

Ergebnisse untauglich, da bedeutungslose Gesten nicht erkannt werden können (Abschnitt 4.7.1).

Beim zweiten untersuchten Verfahren handelt es sich um Erkennungstests mit dem in Abschnitt 4.4.5 eingeführten Schwellwertmodell. Das aus der Verschmelzung aller Referenzmodelle entstehende Schwellwertmodell liefert einen Wert, der von der Erkennungswahrscheinlichkeit eines der Referenzmodelle übertroffen werden muss, um eine erkannte Geste zu signalisieren. Dies verhindert, daß bedeutungslose Gesten irrtümlich als erkannt gemeldet werden. Tabelle 4.11 zeigt die Ergebnisse der Tests mit dem Schwellertmodell.

In allen Testfällen hätte die Methode der maximalen Wahrscheinlichkeit die kor-

Geste	Tests	Fehler	Erkennung
1	30	7	76,66 %
2	24	5	79,16 %
3	25	6	76 %
4	26	4	84,61 %
5	24	0	100 %
Gesamt	129	22	82,94 %

*Tabelle 4.11: Erkennungserfolg der einzelnen Gesten bei Nutzung des Schwellwertmodells*

rekte Geste ermittelt. Dass das Verfahren mit Schwellwertmodell dazu in diesem Maß nicht in der Lage ist, hat mehrere Gründe. Die im Vergleich schlechteren Ergebnisse für die Gesten zwei und drei sind zum Teil auf Probleme bei der Bildverarbeitung zurückzuführen. In diesen beiden Fällen hat der Start/Stop-Filter das Ende der Gesten nicht rechtzeitig erkannt, so dass sie durch eine unnatürliche Häufung von Punkten am Ende der Trajektorie verfälscht wurden. Die Erkennungswahrscheinlichkeit der betreffenden Modelle lag zwar immer höher als die der anderen, reichte aber nicht aus, um den Schwellwert zu überschreiten. Die Tatsache, dass das Referenzmodell der Geste nur mit Daten einer einzigen Person trainiert wurde, hat weiterhin Einfluss auf das weniger gute Abschneiden (Abschnitt 4.8.1). Zu bedenken ist außerdem die verhältnismäßig geringe Anzahl von Gesten, mit denen die Referenzmodelle trainiert wurden (Tabelle 4.9).

Das Verfahren der hierarchischen Gestenklassifikation eignet sich zwar auch zur Gestenerkennung, ist aber prädestiniert für die Datenreduktion (Abschnitt 4.7.1). Bei der Auswertung der hierzu ermittelten Testergebnisse müssen die Werte für Geste eins und zwei, Geste drei und vier und Geste fünf zusammen betrachtet werden, da sie jeweils in einer gemeinsamen Klasse liegen (Abschnitt 4.6). Die hierarchische Gestenklassifikation hat genauso wie die Methode der maximalen Wahrscheinlichkeit die Eigenschaft, immer eine Geste als erkannt zu melden. Dabei handelt es sich um die Geste mit der maximalen Wahrscheinlichkeit in der ausgewählten Klasse. Ist die Erkennung fehlerhaft, so bedeutet dies, dass der Algorithmus die falsche Klasse und folglich auch eine falsche Geste als erkannt ausgewählt hat (Tabelle 4.12).

Als problematisch bei der hierarchischen Klassifikation haben sich schnelle Handbewegungen erwiesen. Da die Auswahl der Klasse auf Grundlage der Anzahl der Richtungsindizes, die aus der Gestentrajektorie gewonnen werden, entsteht, kann eine aus wenig Punkten bestehende Trajektorie irreführend wirken. Eine schnelle Handbewegung erzeugt nun eine eben solche Trajektorie aus we-

Geste	Tests	Fehler	Erkennung
1	21	0	100 %
2	25	0	100 %
3	25	5	80 %
4	25	3	88 %
5	45	8	82,22 %
Gesamt	141	16	88,65 %

*Tabelle 4.12: Erkennungserfolg der einzelnen Gesten bei Anwendung der hierarchischen Gestenklassifikation*

nig Punkten. Die daraus resultierende ebenfalls sehr kurze Folge von Richtungsindizes deutet dann auf eine Geste in der ersten Klasse hin. Die Experimente haben gezeigt, dass alle Fehlentscheidungen für die Gesten drei und vier in einem Sprung in Klasse eins, nicht aber in Klasse drei bestanden. Gleiches gilt auch für Geste fünf.

Unproblematisch hingegen sind langsam ausgeführte Gesten, die immer in eine lange Richtungsindexfolge resultieren. Dies und eine damit einhergehende Entscheidung für eine falsche Klasse mit Gesten aus langen Indexfolgen wird durch den Einsatz der Filter verhindert (Abschnitt 4.5). Sowohl der Pixelfilter als auch der Gleichheitsfilter verhindern eine solche Fehlentscheidung.

Die Intervallgrößen, die festlegen, welche Längen der Richtungsindexfolgen zu welcher Klasse führen, sind experimentell bestimmt und, da sie Mittelwerte darstellen, zwangsläufig ungenau. Weil sich die Intervalle nicht überlappen dürfen, um eine eindeutige Klassifikation zu ermöglichen, muß man einige wenige Fehleinschätzungen in Kauf nehmen. Dem läßt sich allerdings entgegenwirken, indem nur Gesten gewählt werden, die so unterschiedlich sind, dass sie auch eine signifikant andere Anzahl von Richtungsindizes bei der Quantisierung hervorbringen (Abschnitt 4.6).

Bei dem letzten eingesetzten Verfahren handelt es sich um eine Kombination aus hierarchischer Klassifikation und Schwellwertmodell. Während die Anzahl der Richtungsindizes, die die Geste ausmachen, die Wahl der Klasse vorgibt, fällt die Entscheidung über ein Erkennen innerhalb der Klasse mit Hilfe des Schwellwertmodells. Die Testergebnisse sind in Tabelle 4.13 zusammengefaßt.

Die im Vergleich zu den anderen Verfahren spürbar schlechteren Ergebnisse sind durch die doppelten Anforderungen und dadurch auch zweifache Fehleranfälligkeit bedingt. Um eine Geste korrekt erkennen zu können, muss nicht nur die Entscheidung für die richtige Klasse fallen, auch muss die Erkennungswahrscheinlichkeit über dem Schwellwert liegen. Ist eine dieser beiden Voraussetzungen nicht erfüllt, so scheitert die Erkennung.

Geste	Tests	Fehler	Erkennung
1	40	13	67,5 %
2	29	5	82,75 %
3	25	4	84 %
4	15	4	73,33 %
5	15	4	73,33 %
Gesamt	124	30	75,81%

*Tabelle 4.13: Erkennungserfolg der einzelnen Gesten bei Anwendung der hierarchischen Gestenklassifikation kombiniert mit dem Schwellwertmodell*

Die Fehler lassen sich also zum Einen auf die Wahl der falschen Klasse zurückführen und zum anderen auf eine Ablehnung durch das Schwellwertmodell. Insbesondere das schlechte Resultat für Geste eins hat mit dem Schwellwertmodell zu tun, ist aber wesentlich durch die Tatsache bedingt, dass das zugehörige Referenzmodell nur mit Gesten eines Nutzers trainiert wurde (Abschnitt 4.8.1). Eine Verbesserung der Erkennungsrate kann erzielt werden, indem die Intervalle, die der Wahl der Klasse zugrunde liegen, angepasst werden.

#### **4.8.1 Einfluss der Benutzeranzahl**

Die Erkennungsversuche haben die Aussagen von Abschnitt 4.4.4 insofern bestätigt, als dass die Erkennungsraten für Geste eins bei den beiden Verfahren, bei denen das Schwellwertmodell beteiligt ist, unterdurchschnittlich sind. Das Training des Referenzmodells von Geste eins hebt sich von dem der anderen dadurch ab, dass die Trainingsgesten ausschließlich von einer einzigen Person stammen. Getestet wurde jedoch mit zwei Personen. Dabei hat sich gezeigt, dass die Gesten der Person, die auch das Referenzmodell trainiert hat, besser erkannt werden, als die der unbekannteren. Tabelle 4.14 schlüsselt die Testergebnisse nach Testperson und Verfahren auf. Person eins hat die Trainingsdaten für das Referenzmodell geliefert, wohingegen Person zwei nur zum Testen zur Verfügung stand.

	Schwellwertmodell		Hierarchische Klassifikation & Schwellwertmodell	
	Person 1	Person 2	Person 1	Person 2
Geste 1	86,66 %	66,66 %	75 %	60 %

*Tabelle 4.14: Unterschiedlicher Erkennungserfolg in Abhängigkeit davon, ob auch mit Gesten der Testperson trainiert wurde*

Der Erkennungserfolg beim Verfahren der maximalen Wahrscheinlichkeit wurde durch Testen mit einer am Training unbeteiligten Person nicht beeinträchtigt. Obwohl die Gesten zu fast 100 Prozent erkannt wurden, haben diese Tests aber auch gezeigt, dass die Erkennungswahrscheinlichkeiten für die Person, die auch Trainingsdaten zur Verfügung gestellt hat, im Mittel höher waren. Für die hierarchische Gestenklassifikation gilt das Gleiche.

Zusammenfassend läßt sich sagen, dass die Referenzmodelle auf einzelne Personen zugeschnitten werden können, indem sie nur mit Gesten dieser einen Person trainiert werden. Sie können dann auch noch Gesten anderer Personen erkennen, allerdings zu einem weniger großen Anteil, mindestens aber mit einer weniger großen Wahrscheinlichkeit. Auf der anderen Seite lassen sich die Modelle mittels Training von Gesten, die von möglichst vielen verschiedenen Probanden stammen, gut und allgemein auf die spätere Erkennungsaufgabe vorbereiten. Die Erkennungswahrscheinlichkeiten werden dann im Mittel geringer ausfallen, als wenn nur mit einer Person trainiert wird. Dies hat allerdings keinerlei negative Konsequenzen, da in diesem Fall auch der Schwellwert, den es zu überschreiten gilt, kleiner wird. Sollen also Gesten von mehreren Personen erkannt werden, so ist ein Training mit Daten aller dieser Personen empfehlenswert.



# **5 Zusammenfassung und Ausblick**

## **5.1 Zusammenfassung**

Im Rahmen des Projekts Morpha wurde mit dieser Arbeit ein System zur Erkennung dynamischer Gesten vorgestellt. Diese schriftliche Ausarbeitung beschreibt dabei die Spezifikation, stellt die wesentlichen Anforderungen vor und begründet die getroffenen Entwurfsentscheidungen. Schließlich werden die einzelnen Teilbereiche der Implementation und ihr Zusammenwirken beschrieben.

Das Projekt gliedert sich in die Bereiche Bildverarbeitung und Gestenerkennung, wobei die Bildverarbeitung die Handverfolgung mittels eines Stereokamerakopfes mit allen dazu nötigen Verarbeitungsschritten umfasst. Bei der Gestenerkennung liegt der Schwerpunkt auf dem Entwurf und der Anwendung der Hidden-Markov-Modelle. Der Erkennungserfolg ist dabei wesentlich von der gewählten Struktur der Modelle und ihrem Training abhängig.

In diesem Zusammenhang wurde ein Verfahren zum Aufbau der Referenzmodelle entwickelt und besonderes Augenmerk auf Maßnahmen gelegt, die dem Erkennungserfolg förderlich sind. Dazu zählen eine sorgfältige Bestimmung der initialen Modellparameter und die Möglichkeit zum Training mehrfacher Beobachtungssequenzen. Der Einsatz eines Schwellwertmodells soll einer irrtümlichen Erkennung vorbeugen, ist aber optional. Die Anwendung mehrerer Filter verringert das Datenaufkommen und dadurch auch den Berechnungsaufwand deutlich und soll so den Echtzeitcharakter der Anwendung fördern. Die hierarchische Klassifikation der Gesten und ihrer Referenzmodelle ermöglicht eine klare Gliederung und vermeidet die Bildung großer Hidden-Markov-Modelle, die wiederum einen gesteigerten Aufwand mit sich brächten.

Die Kapselung der Bereiche Bildverarbeitung und Gestenerkennung ermöglicht jeweils eine individuelle Weiterentwicklung. Die modulare Struktur des Systems zur Gestenerkennung erlaubt das einfache Ändern, Einfügen oder Austauschen von Komponenten wie zum Beispiel der Filter oder des Schwellwertmodells, um sich veränderten Rahmenbedingungen und Anforderungen anpassen zu können. Die hierarchische Gestenklassifikation ihrerseits erlaubt das ein-

fache Hinzunehmen neuer Gesten und ihrer Auswertung.

## **5.2 Ausblick**

In beiden wesentlichen Bereichen des entwickelten Systems, sowohl der Bildverarbeitung und der damit verbundenen Technik als auch der Gestenerkennung und den Hidden-Markov-Modellen als integralem Bestandteil, sind Erweiterungen und Verbesserungen denkbar und angebracht.

Auf Seiten der Bildverarbeitung können diese Erweiterungen eine verbesserte Handverfolgung, die Erkennung dreidimensionaler Gesten und die Erkennung bedeutungstragender Fingerstellungen beinhalten.

Im Bereich der Hidden-Markov-Modelle existieren eine Vielzahl von Möglichkeiten und Ansätzen, gegebenenfalls bessere Ergebnisse zu erzielen. Der Nutzen dieser Maßnahmen muss allerdings experimentell für den Einzelfall bestimmt werden, da auch hier keine analytischen Bestimmungsmethoden existieren. Interessante Erweiterungen wären der Einsatz verbundener Parameter, unterschiedliche Optimalitätskriterien oder das Verfahren mit Ebenenbildung.

### **5.2.1 Handverfolgung**

Im Komplex Bildverarbeitung ist die Handverfolgung dahin gehend zu erweitern, dass der Kamerakopf der Bewegung folgt. Dies soll verhindern, dass die Hand aus dem Erfassungsbereich der Kameras verschwindet. Auf diese Weise werden ausladendere und größere Bewegungen möglich, ohne dass das System die Hand bei der Verfolgung verliert. Auch muss der Mensch sich dann nicht so positionieren, dass seine Geste den Erfassungsbereich nicht verlässt. Es würde ausreichen, zur Ermittlung der Position der rechten Hand beide Hände und das Gesicht im Fokus zu haben. Alle weiteren Positionen und Bewegungen wären beliebig.

### **5.2.2 Erkennung dreidimensionaler Gesten**

Der Aufbau des Kamerakopfes als Stereosichtsystem ermöglicht eine Tiefenschätzung und damit auch die Aufnahme dreidimensionaler Gesten. Eine Erweiterung des Erkennungssystems im Hinblick auf die Verarbeitung dreidimensionaler Trajektorien ist also denkbar. Als problematisch könnte sich die Modellierung räumlicher Gesten mit Hidden-Markov-Modellen erweisen. Wie bisher

schon im zweidimensionalen Fall muss eine Bewegungsrichtung auf einen diskreten Wert abgebildet werden, um für die Anwendung der Hidden-Markov-Modelle nutzbar gemacht zu werden. Das hiermit einhergehende deutlich erhöhte Datenaufkommen führt damit auch zu größeren Referenzmodellen und längeren Verarbeitungszeiten.

Alternativ kann aber auch eine dreidimensionale Geste mit Hilfe der *Hauptkomponentenanalyse* oder einer Variante des *Iterative Endpoint Fit* auf eine Bewegung in der Ebene abgebildet werden, so dass eine Weiterverwendung des schon bestehenden Erkennungssystems möglich ist. Menschliche Gesten nutzen oft nur zwei Dimensionen beziehungsweise vollführen den größten Teil ihrer Bewegung in nur zwei Dimensionen. Kennt man diese beiden Hauptbewegungsrichtungen, so kann die dritte ausgeschlossen und eine Abbildung in die Ebene erreicht werden.

### 5.2.3 Berücksichtigung der Fingerstellung

Interessant wäre eine Erweiterung um die Auswertung der Fingerstellung. Eine Erkennung von Zeigegesten würde das Erkennungssystem universeller einsetzbar machen und den Anwendungsbereich abrunden.

Eine Auswertung von beispielsweise Zeigegesten als Vertreter der Klasse statischer Gesten bietet sich als Fortsetzung des bis hierher implementierten Ansatzes an. Mit der dynamischen Geste ließe sich dann die gewünschte Aktion spezifizieren, der abschließende statische und möglicherweise auch optionale Teil verdeutlicht dann, worauf sich die Aktion beziehen soll.

Zur Auswertung der Fingerstellung existieren bereits kamerabasierte Ansätze, aber auch die Nutzung eines Datenhandschuhs ist denkbar.

### 5.2.4 Überdeckungen

Als problematisch haben sich Überlappungen der beiden Hände oder von Hand und Gesicht erwiesen. Da die Verfolgung der Hand auf einer Hautfarbsegmentierung basiert (Abschnitt 4.2.2), besteht bei solchen Überdeckungen die Gefahr, die Hand zu verlieren, welche die Geste ausführt. Bei der Bildverarbeitung verschmelzen in einem solchen Fall die beiden Blobs, die etwa eine Hand und das Gesicht repräsentieren, zu einem großen Blob. Trennt sich dieser wieder in zwei Blobs auf, hat also die Hand bei ihrer Bewegung das Gesicht verlassen, so kann allein auf Grundlage der Hautfarbsegmentierung nicht mehr entschieden werden, bei welchem Blob es sich um die Hand beziehungsweise das Gesicht gehandelt hat.

Während sich manche Fälle noch durch den Einsatz von Heuristiken lösen lassen, bleiben andere unentscheidbar. Eine Überlappung von Hand und Gesicht ließe sich zum Beispiel dadurch klären, dass man davon ausgeht, dass die Blob-Position des Gesichts sich nicht verändert, der sich weiterbewegende Blob also die Hand darstellen muss. Dies ist durchaus plausibel, da eine Handbewegung im Allgemeinen nicht vor dem Gesicht endet.

Überlappen sich allerdings zwei sich bewegende Hände, so kann nur aufgrund ihrer Bewegungsrichtung darauf geschlossen werden, welcher Blob welcher Hand zuzuordnen ist. Findet während der Überdeckung ein Richtungswechsel statt, so ist eine Lösung des Konflikts bei Einsatz der Hautfarbsegmentierung nicht möglich.

Abhilfe können hier kinematische Modelle schaffen, die allerdings sehr aufwändig und im Zusammenhang mit den Echtzeitanforderungen problematisch sind. Für teilweise Verdeckungen lassen sich auch *aktive Konturen* einsetzen [Blls98], so dass diese Konflikte eventuell lösbar sind.

### 5.2.5 Erkennung mittels Ebenenbildung

Zusammengesetzte Wörter im Bereich der Spracherkennung zu erkennen ist ein ähnlich gelagertes Problem wie das der Gestenerkennung. So wie Gesten ineinander übergehen oder eine Geste Bestandteil einer anderen sein kann, müssen zusammengesetzte Wörter mit Hilfe von Referenzmodellen erkannt werden, die jeweils einzelne Wörter repräsentieren. Es muss die optimale Folge von Modellen ermittelt werden, die jeweils ein einzelnes Wort darstellen. Das hierbei eingesetzte Konzept stellt eine mögliche Alternative zur Erkennung mit Schwellwertmodell dar, die in dieser Arbeit verwendet wird.

Bei diesem Verfahren entspricht eine Ebene einer Geste in einer Folge mehrerer Gesten. Die Ebenenbildung läuft so ab, dass für jedes Referenzmodell auf jeder Ebene eine Viterbi-Erkennung für die Beobachtungssequenz durchgeführt wird. Jeder dieser Erkennungsvorgänge startet bei Zeitintervall eins und auf Ebene eins. Für jeden Zeitpunkt wird die kumulierte Wahrscheinlichkeit des besten Pfades bis zu diesem Zeitpunkt auf dieser Ebene in diesem Modell vermerkt. Ferner wird eine Referenz auf den Punkt gespeichert, an dem der Pfad auf dieser Ebene begonnen hat. Am Ende einer jeden Ebene, wobei eine Ebene der Position der Geste in der Gestenfolge entspricht, wird das wahrscheinlichste Modell ermittelt. Jede neue Ebene beginnt mit der initial besten Wahrscheinlichkeit zum vorhergehenden Zeitpunkt auf der vorigen Ebene. Hier wird das Viterbi-Ergebnis durch den Vergleich mit den Referenzmodellen zum neuen Zeitpunkt erhöht. Dieser Prozess wird über alle Ebenen wiederholt, wobei die Anzahl der Ebenen der maximal erwarteten Anzahl von Gesten entspricht.

Am Ende jeder Ebene wird die beste Folge von Modellen durch Backtracking ermittelt. Das Maximum über alle Ebenen entspricht der insgesamt wahrscheinlichsten Folge von Gesten.

Der Gesamttablauf setzt sich aus drei Schritten zusammen:

1. Die Vektorquantisierung der eingehenden Daten bildet die Beobachtungssequenz.
2. Die Beobachtungssequenz wird unter Einsatz des Verfahrens zur Ebenenbildung mit den Referenzmodellen getestet. Es werden mehrere mögliche Kandidatenfolgen erzeugt.
3. Im abschließenden Verarbeitungsschritt werden aus den Kandidatenfolgen die unwahrscheinlicheren unter Zuhilfenahme weiterer Kriterien eliminiert. Aus den verbleibenden wird die wahrscheinlichste als Endergebnis ausgewählt.

### **5.2.6 Verbundene Parameter für Hidden–Markov–Modelle**

So genannte *verbundene Parameter* werden in Fällen genutzt, in denen eine nur ungenügende Auswahl an Trainingsdaten zur Verfügung steht, um eine größere Anzahl von Modellparametern zuverlässig abzuschätzen.

Die zugrunde liegende Idee besteht im Wesentlichen darin, dass zwischen Modellparametern unterschiedlicher Zustände eine Äquivalenzrelation formuliert wird. Dies kann zum Beispiel in Fällen geschehen, in denen die Symbolausgabewahrscheinlichkeiten identisch sind.

Durch das Verbinden von Parametern wird die Anzahl unabhängiger Parameter reduziert und die Abschätzung der Werte vereinfacht.

### **5.2.7 Optimierungskriterien für das Hidden–Markov–Modell–Training**

Der Erfolg beim Einsatz von Hidden–Markov–Modellen bei Aufgaben wie der Gesten- oder Handschriftenerkennung steht und fällt mit der Qualität der Modellparameter. Besondere Aufmerksamkeit gebührt deswegen der Abschätzung initialer Modellparameter (Abschnitt 4.4.3) und dem Trainingsverfahren (Abschnitt 3.2.7), die beide wesentlichen Einfluss auf die Ausprägung der Modellparameter und damit auf das Ergebnis der Erkennungsalgorithmen haben.

Probleme, denen beim Aufbau der Modelle und der Aufstellung der Parameter begegnet werden muss, stellen sich dergestalt, dass das zu modellierende Signal sich nicht den Beschränkungen der Hidden-Markov-Modelle anpassen lässt oder dass es äußerst schwierig ist, zuverlässige Schätzwerte für die Modellparameter abzuleiten.

Als Alternative zum Baum-Welch-Algorithmus, der den *Maximum Likelihood* Ansatz (ML) verfolgt, haben sich im Wesentlichen zwei weitere Ansätze herausgebildet, die auf eine zuverlässige Abschätzung der Parameter abzielen. Bei diesen beiden Alternativen handelt es sich um den *Maximum Mutual Information* Ansatz (MMI) und den *Minimum Discrimination Information* Ansatz (MDI).

Ersterer arbeitet mit der gemeinsamen Information einer Beobachtungssequenz und einer Menge von Modellen, die maximiert werden soll. Er kann eingesetzt werden, wenn mehrere Modelle erstellt und ihre Unterschiede betont werden sollen. Auf der Fähigkeit, Beobachtungssequenzen dem korrekten Modell (Referenzmodell) zuzuordnen und andere möglichst zielsicher auszusondern, liegt hier der Schwerpunkt.

Das zweite Verfahren basiert auf der Annahme, dass das zu modellierende Signal nicht notwendigerweise von einer Markovquelle erzeugt wurde, trotzdem aber gewissen Rahmenbedingungen genügt. Dabei sollen die Modellparameter so gewählt werden, dass die Unterschiede zwischen einem gültigen Satz von Signalwahrscheinlichkeitsverteilungen und dem Satz von Hidden-Markov-Modell-Wahrscheinlichkeitsverteilungen minimiert wird. Diese Methode hat den Nachteil, besonders rechenaufwändig zu sein.

# Literatur

- [Aibi99] Aibing Rao: *Report of Project II: Hidden Markov Model (HMM)*. CSE655 Pattern Recognition, State University of New York at Buffalo, Mai 1999.  
<http://www.cse.buffalo.edu/~arao/CSE655/proj2/proj2.html>
- [AIWB88] John Y. Aloimonos, I. Weiss, A. Bandyopadhyay: *Active Vision*. *International Journal of Computer Vision*, Januar 1988, 1(4): 333–356.
- [ArSa97] A. Arsenio, J. Santos-Victor: *Robust Visual Tracking by an Active Observer*. *Proceedings IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1997, 3: 1342–1347.
- [BJM83] L. R. Bahl, F. Jelinek, R. L. Mercer: *A Maximum Likelihood Approach to Continuous Speech Recognition*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983, PAMI-5(2): 179–190.
- [BIs98] A. Blake, Michael Isard: *Active Contours*. Springer Verlag, 1998.
- [BMBF00] Bundesministerium für Bildung und Forschung: *Mensch–Technik–Interaktion in der Wissenschaftsgesellschaft*. 2000.  
<http://www.bmbf.de>
- [Cann86] John Canny: *A Computational Approach to Edge Detection*. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 1986, 8(6): 679–698.
- [CrBC95] J. L. Crowley, F. Berard, J. Coutaz: *Finger Tracking as an Input Device for Augmented Reality*. *IWAGFR*, Juni 1995.
- [Crow95a] J. L. Crowley: *Comparison of Correlation Techniques*. *IAS*, März 1995.
- [DeLR77] A. P. Dempster, N. M. Laird, D. B. Rubin: *Maximum Likelihood from Incomplete Data via the EM Algorithm*. *Journal of the Royal Statistical Society*, 1977, 39(1): 1–38.

- [DiHu91] Rüdiger Dillmann, Martin Huck: *Informationsverarbeitung in der Robotik*. Springer Verlag, 1991.
- [ElAM95] Robert J. Elliott, Lakhdar Aggoun, John B. Moore: *Hidden Markov Models: Estimation and Control*. Applications of Mathematics 29. Springer Verlag, First edition, 1995.
- [Fara79] Raouf F. H. Farag: *Word-Level Recognition of Cursive Script*. *IEEE Transactions on Computers*, Februar 1979, C-28(2): 172–175.
- [Faug93] O. Faugeras: *Three-dimensional Computer Vision. A Geometric Viewpoint*. The MIT Press Cambridge, Massachusetts, 1993.
- [Forn72] G. David Forney Jr.: *The Viterbi Algorithm*. *Proceedings of the IEEE*, März 1973, 61(3): 268–278.
- [Hara80] R. Haralick: *Edge and Region Analysis for Digital Image Data*. *Computer Graphics and Image Processing*, 1980, 12(1): 60–73.
- [HeSa95] Tony Heap, Ferdinando Samaria: *Real Time Hand Tracking and Gesture Recognition Using Smart Snakes*. Technical Report, Olivetti Research Limited, Juni 1995.
- [HuAJ90] Xuedong D. Huang, Yasuo Ariki, Mervyn A. Jack: *Hidden Markov Models for Speech Recognition*. Edinburgh Information Technology Series 7. Edinburgh University Press, 1990.
- [JaKS95] Ramesh Jain, Rangachar Kasturi, Brian G. Schunk: *Machine Vision*. Computer Science Series. McGRAW-HILL, International edition, 1995.
- [Kanu98] Tapas Kanungo: *UMDHMM*, February 1998. Version 1.02.  
<http://www.cfar.umd.edu/~kanungo/software/umdhmm-v1.02.tar>
- [KiJB96] Jong-Sung Kim, Won Jang, Zeungnam Bien: *A Dynamic Gesture Recognition System for the Korean Sign Language (KSL)*. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, Februar 1996, 26(2): 354–359.
- [LeKi99] Hyeon-Kyu Lee, Jin Hyung Kim: *An HMM-Based Threshold Model Approach for Gesture Recognition*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Oktober 1999, 21(10): 961–973.

- 
- [LeRS83] S. E. Levinson, Lawrence R. Rabiner, M. M. Sondhi: *An Introduction to the Application of the Theory of Propabalistic Functions of a Markov Process to Automatic Speech Recognition*. *Bell Systems Technical Journal*, April 1983, 62(4): 1035–1074.
- [MaHi80] D. Marr, E Hildreth: *Theory of Edge Detection*. *Proceedings of the Royal Society London*, 1980, 207: 187–217.
- [Matr99a] *Matrox Imaging Library User Guide*, Version 6.0 Manual No. 10513-MN-0600, 19989.  
<http://www.matrox.com/>
- [Matr99b] *Matrox Imaging Library Command Reference*, Version 6.0 Manual No. 10512-MS-0600, 1999.  
<http://www.matrox.com/>
- [NoUh96] P. Nordlund, T. Uhlin: *Closing the Loop: Detection and Pursuit of a Moving Object with Sensors Onboard a Moving Robot*. *Image Vision and Computing*, 1996, 14(4): 265–275.
- [Rabi89] Lawrence R. Rabiner: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. *Proceedings of the IEEE*, Februar 1989, 77(2): 257–285.
- [RaWJ86] Lawrence R. Rabiner, J. G. Wilpon, B. H. Juang: *A Segmental k-means Training Procedure for Connected Word Recognition*. *AT&T Technical Journal*, Mai–Juni 1986, 65(3): 21–31.
- [ReKa94] J. M. Rehg, Takeo Kanade: *DigitEyes: Vision-Based Human Hand Tracking*. *Proceedings of the European Conference on Computer Vision*, Mai 1994, S. 265–275.
- [ReMu96] I. D. Reid, D. W. Murray: *Active Tracking of Foveated Feature Clusters using Affine Structure*. *International Journal of Computer Vision*, 1996, 18: 41–60.
- [RyNu93a] M. S. Ryan, G. R. Nudd: *The Viterbi Algorithm*. Warwick Research Report RR 238, Department of Computer Science, University of Warwick, Coventry, Februar 1993.  
<http://www.dcs.warwick.ac.uk/pub/reports/rr/238.html>
- [RyNu93b] M. S. Ryan, G. R. Nudd: *Dynamic Character Recognition Using Hidden Markov Models*. Warwick Research Report RR 244, Department of Computer Science, University of Warwick, Coventry, Mai 1993.  
<http://www.dcs.warwick.ac.uk/pub/reports/rr/244.html>

- [SiKC99] H. Sidenbladh, D. Kragic, H. I. Christensen: *A Person following Behaviour for a Mobile Robot. To appear in IEEE International Conference on Robotics and Automation*, Mai 1999.
- [SiSc99] Leonid Sigal, Stan Sclaroff: *Estimation and Prediction of Evolving Color Distributions for Skin Segmentation Under Varying Illumination*. Boston University Computer Science Technical Report 1999-015, Boston University Computer Science Department, Dezember 1999.  
<http://www.cs.bu.edu/techreports/99-015-ColorSkinTracker.ps.Z>
- [StPe95] Thad Starner, Alex Pentland: *Real-Time American Sign Language Recognition from Video Using Hidden Markov Models*. Perceptual Computing Section Technical Report No. 375, M.I.T. Media Laboratory, 1995.  
<ftp://whitechapel.media.mit.edu/pub/tech-reports/TR-375.ps.Z>
- [TrMa97] Jochen Triesch, Christoph von der Malsburg: *Robotic Gesture Recognition*. In *Proceedings of the Bielefeld Gesture Workshop, September 17-19, Germany*, Lecture Notes in Artificial Intelligence 1371, S. 233-244. Springer, September 1997.  
<ftp://ftp.neuroinformatik.ruhr-uni-bochum.de/pub/manuscripts/articles/gw97reprint.ps.gz>
- [Ude96] A. Ude: *Rekonstruktion von Trajektorien aus Stereobildfolgen für die Programmierung von Roboterbahnen*. Dissertation, Universität Karlsruhe, 1996.
- [WiBu92] L. D. Wilcox, M. A. Bush: *Training And Search Algorithms for an Interactive Wordspotting System*. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1992, II: 97-100.
- [WySt82] G. Wyszecki, W. S. Styles: *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, New York, Second edition, 1982.
- [YaLW97] Jie Yang, Weier Lu, Alex Waibel: *Skin-Color Modeling and Adaption*. *Lecture Notes in Computer Science*, 1997, 1352: 687.