# Segmentierung und Klassifizierung von Bildern und Bildsequenzen mit Hidden-Markov-Modellen

Vom Fachbereich Elektrotechnik

der Gerhard-Mercator-Universität - Gesamthochschule Duisburg

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigte Dissertation

von

Stefan Müller

aus Rheinhausen

Referent: Prof. Dr.-Ing. habil. G. Rigoll

Korreferent: Prof. Dr. H. Müller

Tag der mündlichen Prüfung: 17. Dezember 2001

# Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Technische Informatik des Fachbereichs Elektrotechnik der Gerhard-Mercator-Universität Duisburg.
Dem Leiter des Fachgebietes Herrn Prof. Dr.-Ing. habil. G. Rigoll gilt mein besonderer Dank, sowohl für die Themenstellung, als auch für die wissenschaftliche Betreuung. Die wertvollen Anregungen und Diskussionen haben diese Arbeit wesentlich gefördert.
Für die Übernahme des Korreferates und dem damit verbundenen Aufwand möchte ich mich bei Herrn Prof. Dr. H. Müller vom Lehrstuhl Informatik VII (Graphische Systeme) der Universität Dortmund bedanken.
Die Unterstützung meiner Kollegen am Fachgebiet hat wesentlich zum Gelingen dieser Arbeit beigetragen. Insbesondere möchte ich mich bei Frau Dipl.-Ing. Anja Brakensiek, Frau Martha Larson, M.A., sowie Herrn Dipl.-Ing. Frank Wallhoff für die vielfältige Mithilfe bedanken. Für zahlreiche Diskussionen zum Thema statistische Mustererkennung, sowie für die Durchsicht der Arbeit möchte ich Herrn Dr.-Ing. Christoph Neukirchen und Herrn Dr.-Ing. Daniel Willett danken.
Frau Stella Gummersbach danke ich für Ihre Unterstützung bei administrativen Aufgaben.
Schließlich danke ich Herrn Dipl.-Ing. Bernard Große-Rhode sowie Frau Dipl.-Ing. Simone Schneiders für die gewährte Unterstützung.

Duisburg, im Juli 2001

# Inhaltsverzeichnis

# Verzeichnis der verwendeten Formelzeichen

| | |
|---|---|
| $I(X;\Omega)$ | Transinformation |
| $P$ | Wahrscheinlichkeit / Wahrscheinlichkeitsdichte |
| $q_t$ | Beliebiger HMM-Zustand an Position $t$ einer Zustandsfolge |
| $Q$ | Zustandssequenz |
| $S_j$ | Zustand $j$ eines HMMs |
| $t$ | Zeitpunkt |
| $N$ | Anzahl der Zustände eines HMMs |
| $a_{ij}$ | Übergangswahrscheinlichkeit von $S_i$ nach $S_j$ |
| $A$ | Übergangsmatrix |
| $\pi_j$ | Wahrscheinlichkeit des Zustands $S_j$ beim Modelleintritt |
| $\vec{\pi}$ | Vektor der $\pi_j$ |
| $O$ | Symbolsequenz (Observationssequenz) $O = \{o_1, \ldots, o_T\}$ |
| $o_i$ | $i$-te Observation |
| $T$ | Länge der Beobachtung $O$ (Anzahl Merkmalvektoren) |
| $b_j(k)$ | Ausgabeverteilungsfunktion des Zustands $S_j$ |
| $V$ | Ausgabealphabet |
| $M$ | Größe des Ausgabealphabets |
| $v_i$ | $i$-tes Symbol des Alphabets |
| $\vec{b}$ | Vektor der Ausgabewahrscheinlichkeiten |
| $\lambda$ | Parametersatz eines Hidden-Markov-Modells |
| $\alpha_t(j)$ | Vorwärtswahrscheinlichkeit |
| $\beta_t(j)$ | Rückwärtswahrscheinlichkeit |
| $Q^\star$ | Wahrscheinlichste Zustandssequenz |
| $q_t^\star$ | Wahrscheinlichster Zustand zum Zeitpunkt $t$ |
| $\vartheta_t(j)$ | höchste Wahrscheinlichkeit der Zustandssequenz, die in $S_j$ endet |
| $\psi_t(i)$ | Rückverfolgungsmatrix |
| $P^\star$ | Näherungswert für die Produktionswahrscheinlichkeit |
| $\xi_t(i,j)$ | Wahrscheinlichkeit, daß Modell bei $t$ im Zustand $S_i$ ist und $t+1$ im Zustand $S_j$ |

| | |
|---|---|
| $\gamma_t(i)$ | Wahrscheinlichkeit, daß Modell bei $t$ im Zustand $S_i$ ist |
| $\hat{\pi}_i$ | Schätzwert für $\pi_i$ |
| $\hat{a}_{ij}$ | Schätzwert für $a_{ij}$ |
| $\hat{b}_j(y)$ | Schätzwert für $b_j$ |
| $\overline{\pi}_i$ | Häufigkeitswert für $\pi_i$ |
| $\overline{a}_{ij}$ | Häufigkeitswert für $a_{ij}$ |
| $\overline{b}_{jk}$ | Häufigkeitswert für $b_{jk}$ |
| $\chi_{[A]}$ | Kronecker-Operator (1, wenn die Aussage $A$ wahr ist, sonst 0) |
| $\vec{O}$ | vektorwertige Observationssequenz $\vec{O} = \{\vec{o}_1, \ldots, \vec{o}_T\}$ |
| $\vec{o}$ | Beobachtungsvektor (Merkmalvektor) |
| $\vec{\mu}_{jm}$ | Mittelwertvektor der $m$-ten Gaußverteilungskomponte von Zustand $S_j$ |
| $\Sigma_{jm}$ | Kovarianzmatrix der $m$-ten Gaußverteilungskomponente von Zustand $S_j$ |
| $c_{jm}$ | Gewichtung der $m$-ten Gaußverteilungskomponente von Zustand $S_j$ |
| $\mathcal{N}(\vec{o}, \vec{\mu}, \Sigma)$ | Multivariate Gaußverteilung mit Mittelwertvektor $\mu$ und Kovarianzmatrix $\Sigma$ |
| $M$ | Anzahl der Mischungskomponenten |
| $\Sigma^{-1}$ | Inverse der Kovarianzmatrix |
| $D$ | Dimension des Beobachtungsvektors |
| $\zeta$ | Wahrscheinlichkeit |
| $m_t$ | Mischungskomponente zum Zeitpunkt $t$ |
| $\hat{c}_{jm}$ | Schätzwert für $c_{jm}$ |
| $\hat{\mu}_{jm}$ | Schätzwert für $\mu_{jm}$ |
| $\hat{\Sigma}_{jm}$ | Schätzwert für $\Sigma_{jm}$ |
| $S$ | Anzahl des Merkmalströme |
| $\gamma_s$ | Gewicht des $s$-ten Merkmalstroms |
| $\vec{S}$ | Flächenschwerpunkt |
| $I(x,y)$ | Bild mit diskreten Abtastwerten $x, y$ |
| $r_{\max}$ | maximaler Radius eines Objekts |
| $\Delta r$ | Abtastintervall in radialer Richtung |
| $\Delta\varphi$ | Abtastwinkel |
| $I_s(x,y)$ | Form-Matrix |
| $R$ | Redundanz |
| $\varphi^\star$ | Schätzwert für Rotationswinkel |
| $f_i$ | Anzahl $i$ an Merkmalvektoren |
| $m_{p,q}$ | geometrisches Moment der Ordnung $(p+q)$ |
| $\nu_{p,q}$ | Zentralmoment der Ordnung $(p+q)$ |
| $\mu_{p,q}$ | Normalisiertes Moment der Ordnung $(p+q)$ |
| $A_{p,q}$ | Zernike Moment der Ordnung $(p+q)$ |
| $w$ | komplexe Zahl |
| $\eta_T$ | Retrieval-Effizienz |

| | |
|---|---|
| $S$ | Menge an Sites |
| $s_i$ | $i$-te Site |
| $\mathscr{N}$ | Nachbarschaftssystem |
| $\mathscr{N}_i$ | Nachbarn der Site $s_i$ |
| $\mathscr{L}$ | Menge der Label |
| $l_i$ | $i$-tes Label |
| $F$ | Markovsches Wahrscheinlichkeitsfeld, Markov-Random-Field |
| $f$ | eine Realisierung von $F$ |
| $Z$ | Partition Funktion der Gibbs'schen Verteilung |
| $U(f)$ | Energiefunktion |
| $T$ | Temperatur |
| $V_c(f)$ | Clique Potential |
| $C$ | Menge an Cliques |
| $A_{ij,kl,mn}$ | Übergangswahrscheinlichkeit von $S_{ij}$ und $S_{kl}$ nach $S_{mn}$ |
| $a^V_{ij,kl}$ | vertikale Übergangswahrscheinlichkeit |
| $a^H_{ij,kl}$ | horizontale Übergangswahrscheinlichkeit |
| $S_j$ | Metazustand $j$ eines P2DHMM |
| $S^n_j$ | Zustand $j$ des dem $n$-ten Metazustand zugeordneten HMMs |
| $a_{ij}$ | Übergangswahrscheinlichkeit von Metazuständen des P2DHMM |
| $N^j$ | Anzahl der Zustände des dem $j$-ten Metazustand zugeordneten HMMs |
| $a^j_{kl}$ | Übergangswahrscheinlichkeit von $S^j_k$ nach $S^j_l$ im $j$-ten Metazustand |
| $q_{x,y}$ | Beliebiger Zustand am Ort $(x,y)$ |
| $\pi^j_i$ | Wahrscheinlichkeit für den Anfangszustand $S^j_i$ |
| $b^j_i$ | Ausgabeverteilungsfunktion des Zustands $S^j_i$ |
| $O_{xy}$ | Observation am Ort $(x,y)$ |
| $\vartheta^j_{xy}(i)$ | höchste Wahrscheinlichkeit der Zustandssequenz, die in $S^j_i$ endet, für den Metazustand $S_j$ |
| $\psi^j_{xy}(i)$ | Rückverfolgungsmatrix, für den Metazustand $S_j$ |
| $P_j(x)$ | Wahrscheinlichkeit, daß der Metazustand $S_j$ die Bildspalte $x$ produziert hat |
| $D_x(j)$ | höchste Wahrscheinlichkeit der Zustandssequenz, die in $S_j$ endet |
| $\gamma_x(j)$ | Rückverfolgungsmatrix |
| $C(u,v)$ | Koeffizienten der Diskreten-Cosinus-Transformation |
| $\alpha(u)$ | Hilfsgröße |
| $\vec{x}_k$ | Zustandsvektor zum Zeitpunkt $t_k$ |
| $A$ | Übergangsmatrix |
| $\vec{w}$ | Prozeßrauschen (Vektor mit Zufallsvariablen) |
| $Q$ | Kovarianzmatrix von $\vec{w}$ |
| $\vec{z}_k$ | Meßvektor zum Zeitpunkt $t_k$ |
| $H_k$ | Meßmatrix |

| | |
|---|---|
| $\vec{v}$ | Meßrauschen (Vektor mit Zufallsvariablen) |
| $R$ | Kovarianzmatrix von $\vec{v}$ |
| $\hat{\vec{x}}_k$ | a posteriori Schätzwert für $\vec{x}_k$ |
| $\hat{\vec{x}}_k^-$ | a priori Schätzwert für $\vec{x}_k$ |
| $P_k^-$ | Kovarianzmatrix des Fehlers bei der Bestimmung von $\vec{x}_k$ (Messung $\vec{z}_k$ nicht berücksichtigt) |
| $e_k^-$ | Fehler bei der Bestimmung von $\vec{x}_k$ (Messung $\vec{z}_k$ nicht berücksichtigt) |
| $P_k$ | Kovarianzmatrix des Fehlers bei der Bestimmung von $\vec{x}_k$ (Messung $\vec{z}_k$ berücksichtigt) |
| $e_k$ | Fehler bei der Bestimmung von $\vec{x}_k$ (Messung $\vec{z}_k$ berücksichtigt) |
| $K_k$ | Kalman-Verstärkung |
| $I$ | Einheitsmatrix |
| $(x_s, y_s)$ | Koordinaten des Schwerpunktes einer Person |
| $(v_x, v_y)$ | horizontale, bzw. vertikale Geschwindigkeit des Schwerpunktes |
| $(w, h)$ | Breite, bzw. Höhe |
| $A_{ijk,lmn,opq,rst}$ | Übergangswahrscheinlichkeit von $S_{ijk}$ und $S_{lmn}$ und $S_{opq}$ nach $S_{rst}$ |
| $q_{x,y,t}$ | Beliebiger Zustand am Ort $(x,y)$ zur Zeit $t$ |
| $S_{i,j,k}$ | Zustand $(i,j,k)$ eines dreidimensionalen HMMs |
| $o_{xyt}$ | Observation am Ort $(x,y)$ zur Zeit $t$ |
| $S_{\ i\ k}^{\ j}$ | Zustand $k$ eines HMMs, $j$-ter Hyperzustand, $i$-ter Metazustand |
| $L^j$ | Anzahl der Metazustände eines P2DHMMs, $j$-ter Hyperzustand |
| $a_{kl}^j$ | Übergangswahrscheinlichkeit von $S_k^j$ nach $S_l^j$ |
| $\pi_i^j$ | Wahrscheinlichkeit für den Anfangszustand $S_i^j$ |
| $M_i^j$ | Anzahl der Zustände eines HMMs, $j$-ter Hyperzustand, $i$-ter Metazustand |
| $a_{\ i\ kl}^{\ j}$ | Übergangswahrscheinlichkeit von $S_{\ i\ k}^{\ j}$ nach $S_{\ i\ l}^{\ j}$ |
| $\pi_{\ i\ k}^{\ j}$ | Wahrscheinlichkeit für den Anfangszustand $S_{\ i\ k}^{\ j}$ |
| $b_{\ i\ k}^{\ j}(l)$ | Ausgabeverteilungsfunktion des Zustands $S_{\ i\ k}^{\ j}$ |

# Verzeichnis der verwendeten Abkürzungen

| | |
|---|---|
| HMM | Hidden-Markov-Modell |
| KNN | Künstliches Neuronales Netz |
| HTK | Hidden Markov Model Toolkit |
| DBN | Dynamic Bayesian Network |
| ML | Maximum Likelihood |
| WWW | World Wide Web |
| MRF | Markov-Random-Field |
| 2DHMM | Zweidimensionales Hidden-Markov-Modell |
| P2DHMM | Pseudo zweidimensionales Hidden-Markov-Modell |
| ORL | Olivetti Research Laboratory |
| OCR | Optical Character Recognition |
| DCT | Diskrete Cosinus Transformation |
| P3DHMM | Pseudo dreidimensionales Hidden-Markov-Modell |

# Kapitel 1

# Einleitung

Gegenstand dieser Arbeit ist die Anwendung und Evaluierung von statistischen Verfahren für die rechnergestützte Erkennung von Mustern in Bildern und Bildsequenzen. Die verwendeten statistischen Verfahren basieren auf sogenannten Hidden-Markov-Modellen (HMM), die eine Erweiterung der Markov-Quellen darstellen. Die Hidden-Markov-Modelle sind, seit ihrer Einführung in dieses Gebiet Mitte der 70er Jahre, die dominierende Modellierungstechnik im Bereich der automatischen Spracherkennung geworden. Diese Dominanz ist vor allem dadurch zu erklären, daß die Markov-Modellierung auf die unsegmentierte Merkmalsequenz eines fließend gesprochenen Satzes angewendet werden kann. Anders formuliert, ist keine Vorsegmentierung in Einzelworte erforderlich. Eine solche Vorsegmentierung würde ihrerseits segmentierungstypische Fehler in das Gesamtsystem einbringen. Hidden-Markov-Modelle bieten also die Möglichkeit, einen fließend gesprochenen Satz in einem Schritt zu segmentieren und die Einzelkomponenten des Satzes (Worte, bzw. Phoneme) zu erkennen. Ein wichtiges Ziel dieser Arbeit ist es, dieses in der Spracherkennung entwickelte Prinzip auf Bilder und Bildsequenzen zu übertragen. Auf Bilder übertragen bedeutet dies etwa, ein Objekt aus einer Menge von definierten Klassen in einem gegebenen Bild aufzufinden und *simultan* das Objekt zu klassifizieren. In der vorliegenden Arbeit wird der integrierte Segmentierungs- und Erkennungsansatz beispielsweise auf handschriftliche Skizzen angewendet, in denen vordefinierte Symbole in einem ebenfalls von Hand schraffierten Hintergrund eingebettet sind (siehe auch Abb. 1.1). Die Erkenntnisse, die durch die Experimente auf diesen handschriftlich erstellten Mustern gewonnen wurden, konnten ferner genutzt werden, um Personen in Bildern von realen Szenen aufzufinden und zu identifizieren.

Ein offensichtliches Problem, welches bei der Anwendung von Hidden-Markov-Modellen auf Bildern auftritt, ist die inhärente eindimensionale Struktur dieser Modelle. In dieser Arbeit werden jedoch eine Vielzahl von effizienten Algorithmen für mehrdimensionale Mustererkennungsaufgaben, die auf den Hidden-Markov-Modellen basieren, vorgestellt. Dabei kommen vor allem Modellierungsmethoden zum Einsatz, die unter teilweiser Vernachlässigung von Nachbarschaftsbeziehungen höherdimensionale Daten modellieren. Dies

sind die sogenannten *pseudo* zwei- und dreidimensionalen HMMs, die erfolgreich auf Mustererkennungsaufgaben in Bildern und Bildsequenzen angewendet werden können.

## 1.1 Mustererkennung

Wie Eingangs erwähnt wurde, befaßt sich diese Arbeit mit statistischen Verfahren für die automatische Mustererkennung. Es ist somit sinnvoll zunächst festzulegen, was unter dem Begriff *Muster* verstanden werden soll und welche Realisierungen dieses Begriffs in dieser Arbeit verwendet werden. In [Jai00] findet sich die Definition eines Musters als *das Gegenteil von Chaos; es ist eine Entität, vage definiert, der ein Namen geben werden kann*. Die hier verwendeten Muster entstammen den Anwendungsgebieten Mensch-Maschine-Kommunikation sowie den Multimedia-Anwendungen. Im Speziellen wurden die folgenden Muster verwendet: handschriftlich eingegebene Piktogramme und Skizzen, Abbildungen von Werkzeugen, technische Zeichnungen, Personen in natürlichen Szenen, sowie Gesten. Insbesondere werden diese Muster in der Kombination mit Störeffekten, wie beispielsweise einer Teilüberdeckung oder einer Einbettung in eine komplexe Szene betrachtet und untersucht. Ein repräsentatives Beispiel für die hier untersuchten Muster ist in Abb. 1.1 dargestellt.

Abbildung 1.1: Beispiel für ein Muster, das in eine komplexe Szene integriert ist

Die Abbildung zeigt ein von Hand skizziertes Piktogramm, das Mensch oder Person bedeutet und welches sich in einer Umgebung befindet, die aus ähnlichen Konstruktionselementen, nämlich Strichen, zusammengesetzt ist, wie das Piktogramm selbst. Wenn nun die Aufgabe darin besteht, solche Bilder dahingehend zu analysieren, welches Piktogramm sie enthalten, dann muß der verwendete Algorithmus nicht nur das Piktogramm klassifizieren, sondern es auch innerhalb des Bildes auffinden. Dies kann mit Bezug auf die angegebene Definition auch beschrieben werden als die Aufgabe, das *Muster* im *Chaos* aufzufinden.

## 1.2 Integrierte Ansätze zur Segmentierung und Klassifizierung

Anhand von Abb. 1.1 wurde im letzten Abschnitt beschrieben, daß die Aufgabe, ein solches Bild dahingehend zu analysieren, welches Piktogramm es enthält, im wesentlichen aus zwei Unteraufgaben besteht: Zunächst ist das gegebene Bild in die Bestandteile *Piktogramm* und *Hintergrund* zu segmentieren und anschließend ist das gefundene Piktogramm zu klassifizieren.

Die Segmentierung ist ein erster, wichtiger Schritt in der Bildverarbeitung und kann definiert werden als ein Prozeß, der ein Bild in nicht überlappende Regionen unterteilt und dies derart, daß die einzelnen Regionen homogen sind und die Vereinigung zweier benachbarter Regionen keine homogene Region ergibt (aus [Pal93]). Ein Segmentierungsschritt wird zum Beispiel für die folgenden Anwendungen benötigt: beim Auffinden von Schlüsselworten in verrauschtem, eingescanntem Text, bei der Identifizierung von Personen in natürlichen Bildern, et cetera. Zusätzlich zu dem Segmentierungsschritt wird bei den genannten Anwendungen ein Klassifikationsschritt erforderlich sein. In vielen konventionellen Bildbearbeitungsansätzen werden der Segmentierungs- und der Klassifizierungsschritt weitgehend unabhängig voneinander entwickelt.

Diese Unterteilung in voneinander unabhängig entwickelte Verarbeitungsstufen, entsprechend des *Top-Down*-Ansatzes, führt in vielen Fällen zu suboptimalen Ergebnissen. Der Grund hierfür ist die Existenz eines informationstheoretischen Theorems, welches besagt, daß während der Informationsübertragung über verbundene Einzelstufen wichtige Informationen verloren gehen können. Formal ausgedrückt, gehorcht der Transinformationsverlust $\Delta I_\Omega(X,Y) = I(X;\Omega) - I(Y;\Omega)$ bezogen auf die Größe $\Omega$ über eine Kette von $N-1$ Transformationsstufen der Gleichung

$$\Delta I_\Omega(X_1,X_N) = \sum_{i=1}^{N-1} \left[ I(X_i;\Omega) - I(X_{i+1};\Omega) \right] \tag{1.1}$$

Da die Größe $\Delta I_\Omega(X,Y)$ für jede Transformation $Y = f(X)$ stets nicht-negativ ist, kann der Transinformationsverlust einer Transformation in dieser Reihe nicht wiedergewonnen werden. Ein üblicher Ansatz um diesem Effekt zu begegnen, ist, die Größe $\Delta I_\Omega$ für die einzelnen Transformationen individuell zu minimieren, was das Risiko birgt, daß in einer frühen Stufe der Kette die Information in eine Form transformiert wird, die es den folgenden Stufen erschwert, den eigenen Informationsverlust zu minimierten (siehe auch [Plu91]). Um diese Nachteile der voneinander unabhängig entwickelten Verarbeitungsstufen zu vermeiden, wurden alternative, integrierte Verfahren entwickelt (siehe z.B. [Fri97]). Im Bereich der automatischen Spracherkennung ist ein Ansatz überaus populär geworden, der im nächsten Abschnitt vorgestellt wird und der die integrierte Segmentierung und Klassifizierung von Worten in fließend gesprochenen Sätzen ermöglicht.

## 1.3 Statistische Mustererkennung mit Hidden-Markov-Modellen

Wie bereits erwähnt wurde, ist die populärste Anwendung der Hidden-Markov-Modelle der Bereich der Zeitreihen-Klassifikation und hier insbesondere die rechnergestützte Spracherkennung. In den letzten Jahren wurde diese Methode ebenfalls in der Online-Handschrifterkennung erfolgreich eingesetzt (siehe z.B. [Nat93]), welches ein Anwendungsgebiet ist, das große Ähnlichkeit mit der Spracherkennung aufweist, da ebenfalls Zeitreihen verarbeitet werden. Erkennungsaufgaben, bei denen das Muster sich nicht über die *Zeit*, sondern mit dem *Ort* verändert, sind ebenfalls mit Hidden-Markov-Modellen gelöst worden. Solche Aufgaben sind beispielsweise die Erkennung von Gesichtern ([Sam94b, Eic99b, Eic00]), die Abfrage von Bilddatenbanken ([Lin97, Mul98a]) oder handgezeichneten Piktogrammen ([Mul98b, Mul99f]). Die hierbei zu verarbeitenden Daten sind zweidimensional und somit reichen die aus der Spracherkennung bekannten Verfahren nicht mehr aus. Erst die Einführung sog. *pseudo* zweidimensionaler Hidden-Markov-Modelle ermöglichte die Bearbeitung solcher Mustererkennungsaufgaben.

Zeitlich sich verändernde Sequenzen von Bildern stellen ein dreidimensionales Mustererkennungsproblem dar, und daher werden in der vorliegenden Arbeit die pseudo zweidimensionalen Hidden-Markov-Modelle erweitert auf pseudo dreidimensionale HMMs. Mustererkennungsaufgaben, die mit Sequenzen von Bildern arbeiten, sind z.B. die Erkennung von Gesten ([Rig96]), sowie die Indexierung von Videoaufnahmen ([Eic99a]). Die Verwendung von HMMs in den oben genannten Anwendungen hat die folgenden Vorteile: Die Modelle werden mit Beispieldaten trainiert und können somit leicht auf eine gegebene Mustererkennungsaufgabe adaptiert werden. Es ist weiterhin möglich, heterogene Merkmale mit einem einzelnen Modell zu verarbeiten. So besteht der Merkmalvektor in der Spracherkennung oftmals aus cepstralen Merkmalen und zusätzlich z.B. aus der Energie eines Sprachsignals. Die Eigenschaft der HMMs, die eine zentrale Rolle in dieser Arbeit einnimmt, ist, auf integrierte Weise Segmentierungen zu finden und ein Klassifizierungsergebnis zu erzeugen. Dieser Ansatz wurde in der vorliegenden Arbeit erfolgreich auf Bildern und Bildsequenzen eingesetzt. Dabei werden neben den pseudo zwei- und dreidimensionalen Hidden-Markov-Modellen auch die aus der Spracherkennung bekannten eindimensionalen Modelle verwendet, die beispielsweise auf elegante Weise die Umrißkurve einer geometrischen Form beschreiben können. Eingesetzt werden die HMMs dabei bei Problemen der handschriftlichen Piktogrammerkennung, beim Auffinden von Bildern in Bilddatenbasen, der Personenerkennung und -detektion und in der Gestikerkennung.

## 1.4 Gliederung der Arbeit

Die vorliegende Arbeit ist wie folgt gegliedert: Im Anschluß an diese Einführung führt Kapitel 2 in die Theorie der eindimensionalen Hidden-Markov-Modelle ein. Hidden-Markov-Modelle stellen einen doppelt stochastischen Prozeß dar und werden aufbauend auf den zugrundeliegenden sog. Markov-Quellen beschrieben. Anschließend wird in Kapitel 3 eine neuartige rotationsinvariante Modellierung von Piktogrammen und Objektformen mit eindimensionalen Hidden-Markov-Modellen vorgestellt. Die integrierten Segmentierungs- und Klassifizierungseigenschaften der HMMs werden dabei verwendet, um die Orientierung eines gedrehten Objektes herauszufinden und das Objekt zu erkennen.

Kapitel 4 führt in die Theorie der statistischen, zweidimensionalen Modellierung ein und beschreibt zunächst die Markovschen Wahrscheinlichkeitsfelder (engl. Markov-Random-Fields). Aus diesen gehen durch Einführung einer kausalen Abhängigkeit und eines zweiten stochastischen Prozesses die zweidimensionalen Hidden-Markov-Modelle hervor. Die Vereinfachungen, die gemacht werden müssen, um von diesen Modellen zu den pseudo zweidimensionalen HMMs zu gelangen, sind ebenfalls Gegenstand des Kapitels 4. Diese P2DHMMs werden im darauf folgenden Kapitel 5 verwendet, um zweidimensionale Muster in komplexen Umgebungen aufzufinden und zu klassifizieren. Dies erfolgt durch ein neuartiges Verfahren, das P2DHMMs in Kombination mit an den Bildkontext adaptierten Umgebungszuständen verwendet. Auch bei diesem Verfahren erfolgt die Klassifikation und die Segmentierung auf integrierte Weise in einem algorithmischen Schritt.

Kapitel 6 beschreibt die neuartigen pseudo dreidimensionalen Hidden-Markov-Modelle, die für die Erkennung von Bildsequenzen eingesetzt werden können. Das Kapitel gibt eine theoretische Einführung in diese Modelle und beschreibt Experimente, die auf einer Gestendatenbasis durchgeführt wurden. Dabei werden die P3DHMMs eingesetzt, um 12 verschiedene Gesten zu klassifizieren. Es wird ferner ein Ausblick auf einen integrierten Ansatz gegeben, der die gemeinsame Segmentierung und Klassifikation von dreidimensionalen Mustern ermöglicht. Den Abschluß der Arbeit bildet eine Zusammenfassung der erzielten Ergebnisse.

# Kapitel 2

# Theorie eindimensionaler Hidden-Markov-Modelle

## 2.1 Markov-Quellen

Hidden-Markov-Modelle können als eine Erweiterung der Markov-Quellen angesehen werden. Markov-Quellen emittieren Markov-Ketten, die eine Realisierung von Markov-Prozessen darstellen ([Rab89]). Die Markov-Kette besteht aus diskreten Parameterwerten (z.B. diskreten Zeitwerten) und besitzt die sie kennzeichnende Eigenschaft eines (zeitlich-) eingeschränkten *Gedächtnisses*. Dies bedeutet, daß bei einer Markov-Kette $m$-ter Ordnung eine statistische Abhängigkeit zwischen einem emittierten Symbol und $m$ diesem unmittelbar vorausgehenden Symbolen besteht. Aus obiger Annahme folgt unmittelbar die Eigenschaft der *Kausalität*. Der im Zusammenhang mit der vorliegenden Arbeit wichtigste Fall $m = 1$ kann formal folgendermaßen formuliert werden:

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots) = P(q_t = S_j | q_{t-1} = S_i) \tag{2.1}$$

In der obigen Gleichung ist mit $S_j$ der $j$-te sog. *Zustand* der Markov-Quelle bezeichnet, während $q_t$ die Zufallsvariable für die Einnahme eines Zustands aus der Menge der möglichen Zustände $(S_1, \ldots, S_N)$ zum Zeitpunkt $t$ darstellt. Es sei an dieser Stelle angemerkt, daß in diesem Kapitel, wie in der Literatur allgemein üblich, die Markov-Quellen als über die *Zeit* emittierend angesehen wird. Diese Annahme verdeutlicht zunächst die kausalen Eigenschaften der Markov-Quellen, jedoch werden in späteren Kapiteln überwiegend *örtliche* diskrete Werte (z.B. $x_k$) betrachtet. Die im folgenden gemachten Annahmen bzw. vorgestellten Algorithmen gelten auch für diesen Fall. Jeder der $N$ Zustände der Markov-Quelle emittiert genau ein Symbol $Z = Z(q_t)$ aus dem Symbolalphabet. Der Ausdruck auf der rechten Seite in Gleichung 2.1 definiert eine weitere Größe der Markov-Quelle, nämlich die Übergangswahrscheinlichkeit zwischen zwei Zuständen

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \tag{2.2}$$

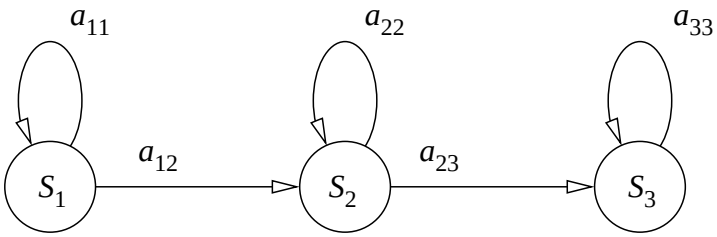


Abbildung 2.1: Graphische Darstellung einer Markov-Quelle mit drei Zuständen

Diese Übergangswahrscheinlichkeiten können zur Übergangsmatrix $A$ zusammengefaßt werden

$$A = \begin{pmatrix} a_{11}, & \dots & ,a_{1N} \\ a_{21}, & \dots & ,a_{2N} \\ \dots, & \dots & ,\dots \\ a_{N1}, & \dots & ,a_{NN} \end{pmatrix} \tag{2.3}$$

Die Addition der Elemente der Zeilen dieser Matrix ergibt stets 1. Matrizen mit den in Gleichung 2.4 angegebenen Eigenschaften werden Markov-Matrizen oder stochastische Matrizen genannt (siehe auch [Dic98]). Es gilt:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{und} \quad a_{ij} \geq 0 \tag{2.4}$$

Die Abbildung 2.1 zeigt eine graphische Darstellung einer Markov-Quelle mit $N = 3$ Zuständen und zugeordneten Übergangswahrscheinlichkeiten. Die Elemente der $3 \times 3$ Matrix $A$ in dem hier illustrierten Fall sind *Null* mit Ausnahme der in der Abbildung explizit angegebenen Übergangswahrscheinlichkeiten $a_{ij}$. Eine solche graphische Darstellung stellt eine Interpretation der Markov-Quelle als einen endlichen statistischen Automaten dar. In der Abbildung ist impliziert, daß der Zustand $S_1$ der Startzustand des Automaten ist. Bei der vollständigen Definition einer Markov-Quelle muß dies jedoch durch das Festlegen der Wahrscheinlichkeiten für die Startzustände explizit angegeben werden. Diese ist üblicherweise für den Zustand $S_j$ folgendermaßen definiert:

$$\pi_j = P(q_1 = S_j) \quad \text{für} \quad 1 \leq j \leq N \tag{2.5}$$

Ähnlich wie die Übergangswahrscheinlichkeiten $a_{ij}$ bei $N$ Zuständen zu einer $N \times N$ Matrix $A$ zusammengefaßt werden können, können die Wahrscheinlichkeiten für den Startzustand zu einem $N$-dimensionalen Vektor zusammengefaßt werden.

$$\vec{\pi} = (\pi_1, \dots, \pi_N)^T \tag{2.6}$$

Für den Fall, daß eine Markov-Quelle vollständig definiert ist, d.h. die Anzahl $N$ der Zustände, die jeweilige Symbolausgabe $Z_j(S_j)$, der Vektor $\vec{\pi}$, sowie die Übergangsmatrix $A$

festgelegt sind, kann die Wahrscheinlichkeit dafür angegeben werden, daß eine vorgegebene Symbolsequenz von dieser Markov-Quelle erzeugt wurde. Sei die Symbolsequenz $O = \{o_1, \ldots, o_T\}$, und sei ferner dieser eine eindeutige Zustandssequenz $Q = \{q_1, \ldots, q_T\}$ zugeordnet, so ist

$$P(O|Modell) = P(Q|Modell) = P(q_1) \cdot \prod_{t=1}^{T} P(q_t|q_{t-1}) \tag{2.7}$$

Dabei ist vorausgesetzt worden, das die Markov-Quelle *stationär* ist, also die Parameter der Quelle nicht von der Zeit abhängig sind. Es ist bei der Markov-Quelle möglich, durch die Beobachtung einer ausgegebenen Sequenz auf deren Zustandsabfolge zu schließen. Anders formuliert, sind die *inneren* Zustände der Quelle von außen sichtbar.

## 2.2 Hidden-Markov-Modelle

Der Übergang von der Markov-Quelle zum Hidden-Markov-Modell, kurz HMM, geschieht durch die Einführung eines zweiten statistischen Prozesses. In dieser Terminologie ist mit dem *ersten statistischen Prozeß* der Wechsel der Zustände mittels vorgegebener Übergangswahrscheinlichkeiten bezeichnet. Der zusätzlich eingeführte zweite statistische Prozeß ist die Ausgabe von Symbolen über Ausgabeverteilungsfunktionen bzw. Ausgabeverteilungsdichten. Diese Ausgabeverteilungsfunktionen bzw. Ausgabeverteilungsdichten sind den einzelnen Zuständen des Markov-Modells[1] fest zugeordnet. Die Einführung dieses zweiten statistischen Prozesses hat zur Folge, daß im allgemeinen Fall von einer beobachteten Symbolfolge nicht mehr auf die dieser zugeordneten Zustandssequenz zurückgeschlossen werden kann. Der Wegfall der Möglichkeit, die Zustandssequenz eindeutig *aufdecken* zu können, führte zu der Bezeichnung *Hidden*-Markov-Modell.

### 2.2.1 Modelldefinition

Es ist in der Literatur üblich (siehe z.B. [Rab89, ST95]), je nach Art der Ausgabeverteilung die Markov-Modelle in diskrete und kontinuierliche Modelle einzuteilen. Die Abb. 2.2 zeigt ein Modell mit drei Zuständen und dem jeweiligen Modelltyp zugeordneten Verteilungs- bzw. Dichtefunktionen. In der Abbildung ist ebenfalls der Fall einer Verteilungsfunktion mit der Wahrscheinlichkeit 1.0 für jeweils genau ein Zeichen des Zeichenvorrates angedeutet, der das Markov-Modell in eine Markov-Quelle überführt. Zunächst wird im folgenden das Hidden-Markov-Modell und dessen Algorithmen für diskrete Ausgabeverteilungen vorgestellt. Den kontinuierlichen Markov-Modellen ist ein eigenes Unterkapitel gewidmet (Kap. 2.2.4).

[1] Aus Gründen der besseren Lesbarkeit wird im folgenden oft der Ausdruck *Markov-Modell* anstelle von *Hidden-Markov-Modell* verwendet. Dies ist stets eindeutig, da der den HMMs zugrundeliegende, einfache statistische Prozeß durchgehend mit *Markov-Quelle* bezeichnet wird.

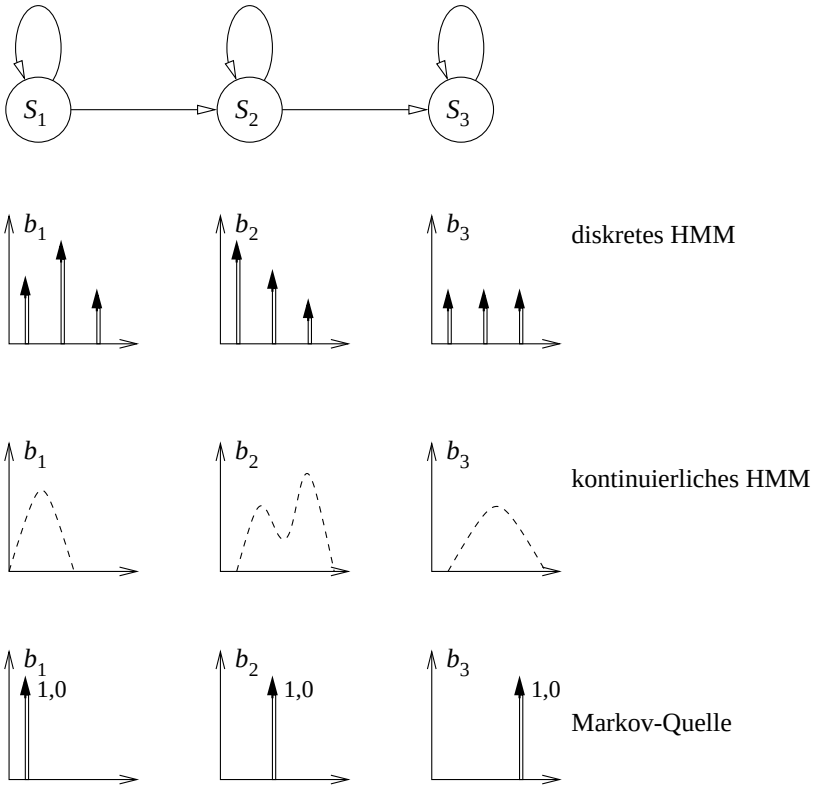


Abbildung 2.2: Ausgabeverteilungsfunktionen und -verteilungsdichten

Bei einem diskreten Markov-Modell gehört zu jedem Modellzustand $S_j$ eine diskrete Ausgabeverteilungsfunktion $b_j(k)$ über einem festgelegten Alphabet der Größe $M$. Das Alphabet $V$ ist formal folgendermaßen gegeben:

$$V = \{v_1, v_2, \ldots, v_M\} \tag{2.8}$$

Damit ist die Wahrscheinlichkeit für die Ausgabe des Symbols $v_k$ im Zustand $S_j$

$$b_j(k) = P(v_k | q_t = S_j) \quad \text{für} \quad \begin{array}{c} 1 \leq j \leq N \\ 1 \leq k \leq M \end{array} \tag{2.9}$$

Sind diese Wahrscheinlichkeiten für alle Zustände und Symbole bekannt, so kann der Vektor der Ausgabeverteilungsfunktionen definiert werden als

$$\vec{b} = (b_1(k), \ldots, b_N(k))^T \quad \text{für} \quad 1 \leq k \leq M \tag{2.10}$$

Die vollständige Definition eines Markov-Modells erfordert also, neben der Festlegung der Quantitäten $N$ und $M$ und des Ausgabealphabets, im wesentlichen die Angabe der Elemente in $\vec{\pi}$,$\vec{b}$ und $A$. Eine übliche Kurzschreibweise für ein solches Modell ist

$$\lambda = (\vec{\pi}, A, \vec{b}) \tag{2.11}$$

Solch eine Modellfestlegung erfolgt jedoch, ähnlich wie bei künstlichen neuronalen Netzen (KNN) nicht *direkt*, sondern über eine Parameterbestimmung anhand von Beispielen. Dies wird bei KNNs mit Lernphase oder Training bezeichnet, während die Verwendung von trainierten Netzen mit Test oder Recall bezeichnet wird. Rabiner definiert in seinem Tutorial [Rab89] drei Aufgabenstellungen, deren Lösung effiziente Algorithmen für die Trainings- und Testphase liefern. Diese drei Aufgaben sind:

1) Finden einer effizienten Methode zur Berechnung von $P(O|\lambda)$. Dies ist die sog. *Produktionswahrscheinlichkeit*, also die Wahrscheinlichkeit dafür, daß eine Symbolsequenz $O = \{o_1, \ldots, o_T\}$ bei gegebenem Modell $\lambda = (\vec{\pi}, A, \vec{b})$ von diesem Modell ausgegeben wird.

2) Aufdecken der wahrscheinlichsten Zustandssequenz $Q$, unter der Annahme, daß die Symbolsequenz $O$ von dem Markov-Modell erzeugt wurde.

3) Finden von Parameteradaptionsalgorithmen für $\lambda = (\vec{\pi}, A, \vec{b})$, die $P(O|\lambda)$ maximieren.

Zu allen drei Aufgaben wurden effiziente Lösungen gefunden, die in den folgenden beiden Kapiteln vorgestellt werden. Zunächst werden die Lösungen zu den Aufgabenstellungen 1) und 2) präsentiert, die die wesentlichen Algorithmen für eine Verwendung der HMMs in der Test- bzw. Klassifikationsphase liefern. Die Lösung der Aufgabe 2) wird zudem die Grundlagen für den integrierten Klassifikations- und Segmentierungsansatz bereitstellen, der in den Kapiteln 3 und 5 vorgestellt wird.

### 2.2.2 Klassifikation

Die Lösung der ersten zuvor definierten Aufgabe, nämlich dem Finden eines effizienten Berechnungsverfahrens für $P(O|\lambda)$, wird es ermöglichen, die Markov-Modelle als Klassifikatoren einzusetzten. Die Produktionswahrscheinlichkeit kann als ein Maß dafür angesehen werden, wie gut eine Symbolsequenz bzw. Observationssequenz[2] zu einem gegebenen Modell paßt. Liegen mehrere Modelle vor, so kann durch die Berechnung der Wahrscheinlichkeiten $P(O|\lambda)$ für die konkurrierenden Modelle das am besten zu der Observation passende Modell ausgewählt werden. Dies erfolgt durch die Entscheidung

$$\mathrm{p}^\star = \underset{\mathrm{p}}{\operatorname{argmax}} \left( P\left( O \,|\, \lambda_{\mathrm{p}} \right) \right). \tag{2.12}$$

Durch Gleichung 2.12 wird die unbekannte Observationssequenz $O$ der Klasse $\mathrm{p}^\star$ zugeordnet.

Der direkte Weg um die gesuchte Wahrscheinlichkeit bei gegebener Observationssequenz $O = \{o_1, \ldots, o_T\}$ zu berechnen ist, die Wahrscheinlichkeiten über alle möglichen Zustands-

[2] Die Bezeichnungen Symbol- und Observationssequenz werden im folgenden gleichwertig verwendet.

sequenzen der Länge $T$ aufzusummieren. Dies ergibt (vgl. auch Gleichung 2.7):

$$P(O|\lambda) = \sum_{\text{alle } Q} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \tag{2.13}$$

Die so gefundene Lösung ist sehr rechenaufwendig, sie erfordert $2T \cdot N^T$ Berechnungen [Rab89]. Der Aufwand der Berechnung steigt also mit der Sequenzlänge $T$ exponentiell an. Der sog. *Forward-Backward-Algorithmus* ermöglicht eine Berechnung der Produktionswahrscheinlichkeit, deren Aufwand linear mit der Sequenzlänge ansteigt. Es handelt sich um einen rekursiven Algorithmus, der die wie folgt definierten *Vorwärtswahrscheinlichkeiten* $\alpha_t(j)$ verwendet:

$$\alpha_t(j) = P(o_1 o_2 \ldots o_t, q_t = S_j|\lambda) \tag{2.14}$$

$P(O|\lambda)$ kann unter Verwendung der Wahrscheinlichkeiten $\alpha_t(j)$ folgendermaßen berechnet werden. Zunächst erfolgt eine Initialisierung durch

$$\alpha_1(j) = \pi_j \cdot b_j(o_1) \quad \text{für} \quad 1 \leq j \leq N \tag{2.15}$$

Anschließend werden iterative Berechnungen mit der folgenden Gleichung durchgeführt:

$$\alpha_{t+1}(j) = \left( \sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \right) b_j(o_{t+1}) \quad \text{für} \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq t \leq T-1 \end{array} \tag{2.16}$$

Die gesuchte Wahrscheinlichkeit kann schließlich durch das Summieren der berechneten Vorwärtswahrscheinlichkeiten zum Zeitpunkt $T$ ermittelt werden:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j) \tag{2.17}$$

Der Forward-Backward-Algorithmus erfordert lediglich $N^2T$ Rechenoperationen und stellt somit ein effizientes Berechnungsverfahren zur Bestimmung der Produktionswahrscheinlichkeit dar. Alternativ kann die Produktionswahrscheinlichkeit jedoch auch mit Hilfe der *Rückwärtswahrscheinlichkeiten* $\beta_t(j)$ berechnet werden. Diese sind definiert als:

$$\beta_t(j) = P(o_{t+1}, o_{t+2}, \ldots, o_T|q_t = S_j, \lambda) \tag{2.18}$$

Da bereits ein Algorithmus existiert, der $P(O|\lambda)$ berechnet, müßte dieses Alternativverfahren eigentlich nicht vorgestellt werden. Diese Rückwärtswahrscheinlichkeiten erklären jedoch zum einen den Namen dieses Algorithmus und werden andererseits für das in Kapitel 2.2.3 beschriebene Trainingsverfahren benötigt. Die Rückwärtswahrscheinlichkeiten $\beta_T(j)$ werden initialisiert durch:

$$\beta_T(j) = 1 \quad \text{für} \quad 1 \leq j \leq N \tag{2.19}$$

Die iterative Berechnung ist gegeben durch:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j) \quad \text{für} \quad \begin{array}{l} 1 \leq i \leq N \\ t = T-1, T-2, \ldots, 1 \end{array} \tag{2.20}$$

Und schließlich kann $P(O|\lambda)$ durch folgende Gleichung berechnet werden:

$$P(O|\lambda) = \sum_{j=1}^{N} \beta_1(j) \tag{2.21}$$

Es stehen somit zwei effiziente Verfahren zur Verfügung, die, basierend auf den Vorwärts- und Rückwärtswahrscheinlichkeiten, die Produktionswahrscheinlichkeit mit einem linear mit der Sequenzlänge ansteigenden Aufwand berechnen. Dennoch werden diese Verfahren zur Bestimmung von $P(O|\lambda)$ im Klassifikationsschritt nur selten verwendet. Stattdessen wird der sog. *Viterbi-Algorithmus* ([For73]) verwendet, der eine effiziente Lösung der zweiten formulierten Aufgabenstellung darstellt. Der Viterbi-Algorithmus ermittelt die wahrscheinlichste Zustandssequenz $Q^\star$, unter der Annahme, daß die Observationssequenz von dem Modell $\lambda$ erzeugt wurde und stellt eine Variante des Verfahrens zur Berechnung der Vorwärtswahrscheinlichkeiten dar. Diese *optimale Zustandssequenz* ergibt sich aus

$$P(O, Q^\star|\lambda) = \max_Q P(O, Q|\lambda) \tag{2.22}$$

Die Wahrscheinlichkeit $P(O, Q^\star|\lambda) = P^\star(O|\lambda)$ kann als ein Näherungswert für die Produktionswahrscheinlichkeit verwendet werden. Bei dem Viterbi-Algorithmus werden statt der Vorwärtswahrscheinlichkeiten $\alpha_t(j)$ die maximal erzielbaren Wahrscheinlichkeiten

$$\vartheta_t(j) = \max_Q (P(o_1 \ldots o_t, q_1 \ldots q_t = S_j|\lambda)) \tag{2.23}$$

definiert (vgl. auch Gleichung 2.14). $\vartheta_t(j)$ ist die höchste Wahrscheinlichkeit der Zustandssequenzen, die im Zustand $S_j$ enden, für die ersten $t$ Observationen. Um die optimale Zustandssequenz zurückverfolgen zu können, wird zudem die Matrix $\psi_t(j)$ verwendet. Der Viterbi-Algorithmus beginnt mit der Initialisierung von $\vartheta$ und $\psi$:

$$\begin{array}{l} \vartheta_1(i) = \pi_i \cdot b_i(o_1) \\ \psi_1(i) = 0 \end{array} \quad \text{für} \quad 1 \leq i \leq N \tag{2.24}$$

Der Rekursionsschritt ist gegeben durch

$$\begin{array}{l} \vartheta_t(j) = \max\limits_{1 \leq i \leq N} (\vartheta_{t-1}(i) \cdot a_{ij})\, b_j(o_t) \\ \psi_t(j) = \operatorname*{argmax}\limits_{1 \leq i \leq N} (\vartheta_{t-1}(i) \cdot a_{ij}) \end{array} \quad \text{für} \quad \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \tag{2.25}$$

Schließlich ergibt sich der Näherungswert für die Produktionswahrscheinlichkeit aus

$$\begin{array}{l} P^\star = \max\limits_{1 \leq j \leq N} (\vartheta_T(j)) \\ q_T^\star = \operatorname*{argmax}\limits_{1 \leq i \leq N} (\vartheta_T(i)) \end{array} \tag{2.26}$$

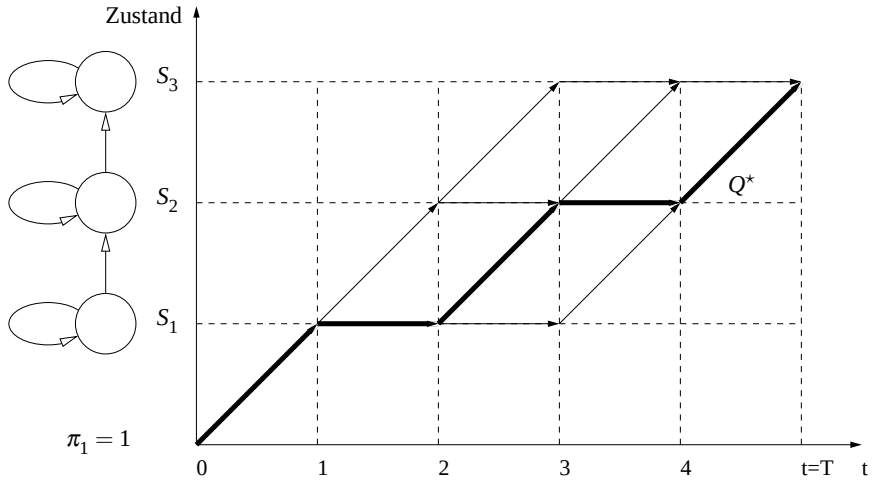


Abbildung 2.3: Mögliche Zustandssequenzen als Pfade in einem Trellis-Diagramm. Der hervorgehobene Pfad repräsentiert die wahrscheinlichste Zustandssequenz $Q^\star$.

Die gesuchte Zustandssequenz $Q^\star$ kann folgendermaßen zurückverfolgt werden:

$$q_t^\star = \psi_{t+1}(q_{t+1}^\star) \quad \text{für} \quad t = T-1, \ldots, 1 \tag{2.27}$$

Bei Betrachtung der Gleichungen für $\vartheta_t(j)$ (2.24 bis 2.26) fällt die große Ähnlichkeit zur Berechnung der Produktionswahrscheinlichkeit mit Hilfe der Vorwärtswahrscheinlichkeit $\alpha_t(j)$ auf (siehe Gleichungen 2.15 bis 2.17). Der Hauptunterschied besteht darin, daß in den Gleichungen für den Rekursionsschritt beim Viterbi-Algorithmus der Maximum-Operator anstelle der Summation verwendet wird (vgl. Gleichung 2.25 und 2.16).

Der Viterbi-Algorithmus bzw. die ermittelte optimale Zustandssequenz $Q^\star$ kann in einem sog. *Trellis*-Diagramm veranschaulicht werden ([For73]). Die Abb. 2.3 zeigt beispielhaft für ein sog. Links-Rechts-Modell ein solches Diagramm, das alle möglichen Zustandssequenzen als Pfad darstellt. Nach [Lev83] ist ein Links-Rechts-Modell durch folgende Eigenschaften gekennzeichnet: Die erste Ausgabe erfolgt im ersten Zustand des Modells oder anders formuliert gilt $\pi_1 = P(q_1 = S_1) = 1$. Die Observation zum Zeitpunkt $t = T$, also die letzte Observation der Sequenz wird vom Zustand $q_T = S_N$ des Markov-Modells emittiert. Dieser letzte Zustand $S_N$ wird auch als *absorbierender* Zustand bezeichnet, da er nach dem erstmaligen Erreichen nicht mehr verlassen werden kann ($a_{NN} = 1$). Die Topologie des Links-Rechts-Modells ist, wie in Abb. 2.3 dargestellt, dadurch gekennzeichnet, daß ein einmal verlassener Zustand nicht wieder erreicht werden kann. Diese speziellen Markov-Modelle werden in dieser Arbeit häufig verwendet. Der hervorgehobene Pfad in Abb. 2.3 repräsentiert die, durch den Viterbi-Algorithmus bestimmte, wahrscheinlichste Zustandssequenz.

### 2.2.3 Training

Die Klassifikation mit den Algorithmen aus Unterkapitel 2.2.2 ist nur dann sinnvoll, wenn die Parameter der Markov-Modelle zuvor auf die Trainingsdaten der jeweiligen Klassenelemente angepaßt wurden. Dies wurde schon in Unterkapitel 2.2.1 als eines der zu lösenden Aufgabenstellungen formuliert. Da es bisher keine analytische Methode gibt, um die Modellparameter direkt zu bestimmen, wird üblicherweise ein iteratives Verfahren verwendet, das als *Baum-Welch-Algorithmus* bekannt ist. Der Baum-Welch-Algorithmus basiert auf der Maximum-Likelihood(ML)-Schätzung der Modellparameter. Es werden, beginnend mit einer Initialisierung, die Modellparameter mit Hilfe des Forward-Backward-Algorithmus neu geschätzt, so daß die Produktionswahrscheinlichkeit $P(O|\lambda)$ in jeder Iteration vergrößert wird. Der Algorithmus verwendet die Rechengröße $\xi_t(i,j)$, die die Wahrscheinlichkeit darstellt, daß sich das Modell zum Zeitpunkt $t$ im Zustand $S_i$ befindet und zum darauf folgenden Zeitschritt in den Zustand $S_j$ wechselt. Dies kann formal folgendermaßen angegeben werden:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{2.28}$$

Die Größe $\xi$ kann unter Verwendung der Vorwärts- und Rückwärtswahrscheinlichkeiten auch dargestellt werden als:

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{2.29}$$

Dabei wurden die folgenden zwei Beziehungen verwendet:

$$\alpha_t(j) \cdot \beta_t(j) = P(O, q_t = S_j | \lambda) \tag{2.30}$$

und

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j) \tag{2.31}$$

Ferner wird die Wahrscheinlichkeit $\gamma_t(i) = P(q_t = S_i | O, \lambda)$ definiert als die Wahrscheinlichkeit, daß sich bei gegebener Observationssequenz das Modell zum Zeitpunkt $t$ im Zustand $S_i$ befindet. Diese Wahrscheinlichkeit kann unter Verwendung der Größe $\xi$ angegeben werden als:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.32}$$

Werden diese Wahrscheinlichkeiten $\gamma_t(i)$ über der Zeit $t$ aufsummiert, so ergibt sich eine Größe, die als die Anzahl der Einnahmen des Zustandes $S_i$ interpretiert werden kann. Eine andere Interpretation ist, diese Größe bei einer Summation über der Zeit von $t = 1$ bis $t = T - 1$ als die Anzahl der Übergänge, die vom Zustand $S_i$ ausgingen, anzusehen. In analoger Weise kann die Summation über $(0 \leq t \leq T - 1)$ von $\xi_t(j,i)$ als die Anzahl der Übergänge vom Modellzustand $S_i$ zum Zustand $S_j$ interpretiert werden. Zusammenfassend ergeben sich daraus die folgenden Schätzformeln für die Modellparameter:

$$\begin{aligned}\hat{\pi}_i &= \text{Häufigkeit der Einnahme des Zustands } S_i \text{ zum Zeitpunkt } t = 1 \\ &= \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum\limits_{t=1}^{T} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{2.33}$$

$$\begin{aligned}\hat{a}_{ij} &= \frac{\text{Übergänge vom Zustand } S_i \text{ in den Zustand } S_j}{\text{Alle Übergänge aus Zustand } S_i} \\ &= \frac{\sum\limits_{t=1}^{T-1} \xi_t(i,j)}{\sum\limits_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum\limits_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum\limits_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{2.34}$$

$$\begin{aligned}\hat{b}_j(y) &= \frac{\text{Aufenthaltswahrscheinlichkeit im Zustand } S_j \text{ wenn } y \text{ emittiert wird}}{\text{Aufenthaltswahrscheinlichkeit im Zustand } S_j} \\ &= \frac{\sum\limits_{\substack{t=1 \\ o_t=y}}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)} = \frac{\sum\limits_{\substack{t-1 \\ o_t=y}}^{T} \alpha_t(j)\beta_t(j)}{\sum\limits_{t=1}^{T} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{2.35}$$

Eine Alternative zur Berechnung mit dem Baum-Welch-Algorithmus ist der sog. Segmental-KMeans-Algorithmus, der auch Viterbi-Trainings-Algorithmus genannt wird. Beim Viterbi-Training wird anders als beim Baum-Welch-Algorithmus nicht die Größe $P(O|\lambda)$ bei jedem Iterationsschritt vergrößert, sondern die Zielgröße $P^\star(O|\lambda)$ (siehe auch Gleichung 2.22). Diese Größe ist der bereits eingeführte Näherungswert für die Produktionswahrscheinlichkeit, der auf der optimalen Zustandsfolge $Q^\star$ basiert. Der Algorithmus beginnt mit der Auswahl der Modellparameter für ein Startmodell $\lambda^0$. Anschließend werden Iterationen mit folgenden Einzelschritten durchgeführt:

1) Bestimmen der Zustandsfolge $Q^\star$ mit dem Viterbi-Algorithmus

$$P(O, Q^\star|\lambda^{(n-1)}) = \max_Q P(O, Q|\lambda^{(n-1)}) \tag{2.36}$$

2) Es werden die folgenden Häufigkeiten bestimmt:

$$\overline{\pi}_i = \chi_{[q_1^\star = S_i]} \tag{2.37}$$

$$\overline{a}_{ij} = \sum_{t=1}^{T-1} \chi_{[q_t^\star = S_i, q_{t+1}^\star = S_j]} \tag{2.38}$$

$$\overline{b}_{jk} = \sum_{t=1}^{T} \chi_{[q_t^\star = S_j, o_t = v_k]} \tag{2.39}$$

3) Normierung durch

$$\hat{\pi}_i = \frac{\overline{\pi}_i}{\sum_{i=1}^{N} \overline{\pi}_i} \tag{2.40}$$

$$\hat{a}_{ij} = \frac{\overline{a}_{ij}}{\sum_{i=1}^{N} \overline{a}_{ij}} \tag{2.41}$$

$$\hat{b}_j(v_k) = \frac{\overline{b}_j(v_k)}{\sum_{k=1}^{K} \overline{b}_j(v_k)} \tag{2.42}$$

4) Übernehmen der neuen Modellparameter

$$\lambda^{(n)} = (\hat{\pi}_i, \hat{a}_{ij}, \hat{b}_j(v_k)) \tag{2.43}$$

5) Gehe zu Schritt 1)

Die in den Gleichungen zur Bestimmung der Häufigkeiten (2.37 bis 2.39) verwendete Funktion $\chi$ ist die sog. Kronecker-Delta-Funktion. Eine detailliertere Darstellung des Segmental-KMeans-Algorithmus, einschließlich der Untersuchung des Konvergenzverhaltens, findet sich in den Arbeiten [Jua90] und [Rab76].

### 2.2.4 Kontinuierliche Ausgabefunktionen

Bisher wurden diskrete Markov-Modelle betrachtet, die Observationssequenzen der Form $O = \{o_1, \ldots, o_T\}$ emittieren, mit Elementen $o_t$, die aus einem festgelegten Alphabet $V$ stammen. Sollen reellwertige Vektorsequenzen mit Markov-Modellen trainiert bzw. klassifiziert werden, so können die Vektoren der Sequenz $\{\vec{o}_1, \ldots, \vec{o}_T\}$ durch eine Vektorquantisierung in ein diskretes Symbolalphabet z.B. durch den K-Means Algorithmus überführt werden und somit die Algorithmen der Kapitel 2.2.2 und 2.2.3 weiter verwendet werden. Alternativ können die Vektorsequenzen durch Markov-Modelle mit kontinuierlichen Ausgabeverteilungen auch direkt modelliert werden. Der Vorteil hierbei liegt in dem Wegfall des Quantisierungsschrittes, der stets zu einer Verzerrung der Eingabedaten und einem damit verbundenen

Informationsverlust führt ([ST95]). Aus diesem Grund wurden in der vorliegenden Arbeit kontinuierliche Modelle verwendet. Die Dichtefunktionen eines Zustands $S_j$ der kontinuierlichen Modelle sind üblicherweise als Gaußsche Mischverteilungsdichten der Form

$$b_j(\vec{o}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\vec{o}, \vec{\mu}_{jm}, \Sigma_{jm}) \quad \text{mit} \quad \sum_{m=1}^{M} c_{jm} = 1 \tag{2.44}$$

gegeben. Dabei ist $c_{jm}$ die Gewichtung der $m$-ten Mischungskomponente, $M$ die Anzahl der Mischungskomponenten und $\mathcal{N}$ eine multivariate Gaußdichte. Mit einer solchen Überlagerung von hinreichend vielen Gaußdichten können beliebige Dichtefunktionen approximiert werden. Die Gaußdichte ist für $D$-dimensionale Größen folgendermaßen angebbar:

$$\mathcal{N}(\vec{o}, \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \cdot e^{-\frac{1}{2}(\vec{o}-\vec{\mu})^T \Sigma^{-1} (\vec{o}-\vec{\mu})} \tag{2.45}$$

Der Vektor der Mittelwerte ist in obiger Gleichung mit $\vec{\mu}$ bezeichnet, die Kovarianzmatrix bzw. deren Inverse mit $\Sigma$ und $\Sigma^{-1}$.

Die Gleichungen für die Parameteranpassungen an die Trainingssequenzen können unter Verwendung der Wahrscheinlichkeit $\zeta_t(j,k)$, daß die Mischungskomponente $m_t = k$ im Zustand $S_j$ zur Zeit $t$ ausgewählt wurde, abgeleitet werden. Dies entspricht der folgenden formalen Formulierung

$$\zeta_t(j,k) = P(q_t = S_j, m_t = k | \vec{O}, \lambda) \tag{2.46}$$

Die Wahrscheinlichkeiten $\zeta_t(j,k)$ können ähnlich wie die Größen $\gamma$ und $\xi$ durch die Vorwärts- und Rückwärtswahrscheinlichkeiten $\alpha$ und $\beta$ bestimmt werden. Schließlich ergeben sich die folgenden Schätzformeln (aus [ST95]):

$$\hat{c}_{jk} = \frac{1}{\sum_{t=1}^{T} \sum_{k=1}^{M} \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \tag{2.47}$$

$$\hat{\mu}_{jk} = \frac{1}{\sum_{t=1}^{T} \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \cdot \vec{o}_t \tag{2.48}$$

$$\hat{\Sigma}_{jk} = \frac{1}{\sum_{t=1}^{T} \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \cdot (\vec{o}_t - \mu_{jk})(\vec{o}_t - \mu_{jk})^T \tag{2.49}$$

Bei dem Training eines kontinuierlichen Markov-Modells werden neben den Gleichungen 2.47 bis 2.49 auch in unveränderter Weise die Baum-Welch-Formeln für die Größen $\hat{\pi}$ (Gleichung 2.33) und $\hat{a}_{ij}$ (Gleichung 2.34) verwendet.

In späteren Kapiteln dieser Arbeit wird auch die folgende Variante der Modellierung der Dichtefunktion der Modellzustände (vgl. Gleichung 2.44) verwendet:

$$b_j(\vec{o}) = \prod_{s=1}^{S} b_{js}(\vec{o}_s)^{\gamma_s} \tag{2.50}$$

In der Gleichung 2.50 werden $S$ sog. Merkmalströme (engl. Streams) verwendet, die bei großen, inhomogenen Merkmalvektoren Vorteile aufweisen ([Gup97]). Merkmalströme sind ein oder mehrere Komponenten des Merkmalvektors, die als statistisch unabhängig angenommen werden und denen sog. Merkmalstrom-Gewichtungen $\gamma_s$ zugeordnet werden. Es wird beispielsweise oft in der automatischen Spracherkennung der vieldimensionale Merkmalvektor unterteilt in Komponenten, die auf die gleiche Weise berechnet wurden. Ein Merkmalstrom enthält nach dieser Unterteilung ausschließlich cepstrale Koeffizienten, zwei weitere die Differenzen dieser Koeffizienten bzw. Differenzen höherer Ordnung und schließlich ein Merkmalstrom die Komponenten, die aus der Signalenergie berechnet wurden. Eine detaillierte Beschreibung dieser Aufteilung in Merkmalströme findet sich in [Neu98].

### 2.2.5 Aspekte der Implementierung

Im Kontext dieser Arbeit wurde überwiegend das sog. Hidden Markov Toolkit (HTK) der Cambridge University verwendet (siehe z.B. [You94]). Diese Software ist auf die Verwendung für die automatische Spracherkennung ausgerichtet und mußte mithin für die Erkennung von Bildern und Bildinhalten angepaßt werden. Aus der Ausrichtung auf die Spracherkennung ergibt sich der Bedarf, aus einzelnen Markov-Modellen komplexere Strukturen aufbauen zu können. So sollen beispielsweise aus *Phonem*-basierten Markov-Modellen mittels einer Phonemisierungstabelle Worte gebildet werden können und aus diesen dann wiederum ganze Sätze. Dieses Aneinanderhängen von Modellen wird durch die Einführung von nichtemittierenden Zuständen jeweils *vor* und *hinter* dem eigentliche Modell ermöglicht. Die Übergangswahrscheinlichkeiten vom ersten nichtemittierenden Zustand zu den emittierenden Zuständen stellen eine alternative Formulierung der schon erwähnten Wahrscheinlichkeiten der Anfangszustände $\pi_j = P(q_1 = S_j)$ dar. Die Wahrscheinlichkeit $\pi_1$ für die Einnahme des Zustandes $S_1$ zum Zeitpunkt $t = 1$ kann beispielsweise bei Verwendung des nichtemittierenden Zustandes $S_0$ als Übergangswahrscheinlichkeit folgendermaßen dargestellt werden:

$$\pi_1 = P(q_1 = S_1) = a_{01} = P(q_1 = S_1 | q_0 = S_0) \tag{2.51}$$

Trotz der Verwendung zweier zusätzlicher Zustände soll in den folgenden Kapiteln jedoch wie bisher unter einem Modell mit $N$ Zuständen ein Modell mit $N$ *emittierenden* Zuständen verstanden werden. Die nichtemittierenden Zustände und damit die Möglichkeiten zur Verkettung von Modellen sind für die in den folgenden Kapiteln dargestellten Methoden wichtig.

### 2.2.6 Bayes Netze

Die in der Literatur am häufigsten gewählte graphische Darstellungsweise von Markov-Modellen ist die Darstellung als finiter statistischer Automat (siehe auch Abb. 2.2). Einer

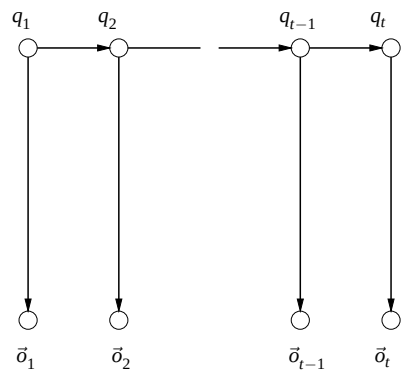


Abbildung 2.4: Darstellung des Hidden-Markov-Modells als dynamisches Bayes-Netz

solchen Darstellung ist vor allem die Topologie des Modells zu entnehmen und mithin die vorhandenen Parameter. So ist zum Beispiel in Abb. 2.2 zu sehen, daß es sich bei dem dargestellten Markov-Modell um ein sog. Links-Rechts-Modell mit drei Zuständen handelt. Impliziert wird zudem, daß es sich um ein Markov-Modell erster Ordnung handelt, da die Verbindungspfeile üblicherweise die Übergangswahrscheinlichkeiten und somit die Wahrscheinlichkeiten $P(q_t = S_j | q_{t-1} = S_i)$ repräsentieren. Eine alternative graphische Darstellung der Markov-Modelle bei der die statistischen Abhängigkeiten verdeutlicht werden, ist in Abb. 2.4 gezeigt und findet sich ebenfalls in [Smy97, Luc95, Mur00]. Die Abb. 2.4 stellt das Hidden-Markov-Modell als gerichtetes graphisches Modell oder dynamisches Bayes-Netz (DBN, engl. Dynamic Bayesian Network) dar. DBNs sind graphische Modelle von statistischen Prozessen und stellen die Abhängigkeiten zwischen der Observationssequenz und den Zufallsvariablen des Modells dar. Die Zufallsvariablen sind in Abb. 2.4 durch die Knoten des Graphen gegeben. Die horizontalen und vertikalen Verbindungslinien in der Abbildung repräsentieren die statistischen Abhängigkeiten, die durch die Annahme eines Markov-Prozesses erster Ordnung impliziert werden. Insbesondere stellt bei DBNs die Abwesenheit von Verbindungen in den Graphendarstellungen statistische Unabhängigkeitsannahmen dar. Dynamische Bayes-Netze stellen eine Obermenge einer Vielzahl statistischer Modelle dar, die aus unterschiedlichen wissenschaftlichen Disziplinen stammen. Beispiele für solche statistischen Modelle sind neben den Markov-Modellen, Kalman-Filter und probabilistische Experten-Systeme ([Smy97]). Die DBNs wiederum gehören der übergeordneten Klasse von sog. *graphischen Modellen* an, bei denen wahrscheinlichkeitstheoretische Ansätze mit der Graphentheorie verbunden werden. Neben den Bayes-Netzen sind die in der Physik und der Computer-Vision sehr populären ungerichteten Graphen Mitglieder dieser übergeordneten Familie. Zur Klasse der ungerichteten Graphen gehören z.B. die *Markov-Random-Fields* und die *Boltzmann-Maschine*. Mit Hilfe der graphischen Modelle ist es möglich, Gemeinsamkeiten zwischen den in unterschiedlichen wissenschaftlichen Disziplinen entwickelten Modellen und Algorithmen zu entdecken und zu nutzen. Dies wird sehr ausführlich in dem Übersichtsartikel von Murphy in [Mur00] behandelt.

In der vorliegenden Arbeit werden die graphischen Modelle genutzt, um die Zusammenhänge zwischen den in Kapitel 4.1 vorgestellten Markovschen Zufallsfeldern (engl. Markov-Random-Fields), dem in Kapitel 5.4.2 verwendeten Kalman-Filter und den bereits vorgestellten Hidden-Markov-Modellen (einschließlich der zweidimensionalen Variante in Kapitel 4.3) zu erläutern.

## 2.3 Kapitelzusammenfassung

Es wurde in die Theorie der eindimensionalen Hidden-Markov-Modelle eingeführt. Diese Modelle stellen die dominierende Klassifikationsmethode für zeitlich veränderliche Muster, insbesondere Sprachmuster, dar. Es stehen sehr effiziente Algorithmen für die Berechnung der Produktionswahrscheinlichkeiten, die für die Klassifikation benötigt werden, und das Modelltraining zur Verfügung. Diese Berechnungsvorschriften sind der Viterbi- bzw. der Baum-Welch-Algorithmus, die beide in diesem Kapitel vorgestellt wurden. Der Viterbi-Algorithmus berechnet nicht die Produktionswahrscheinlichkeit selbst, sondern einen Approximationswert, der auf der wahrscheinlichsten Zustandssequenz basiert. Der Algorithmus hat für die folgenden Kapitel eine wichtige Bedeutung, da er die Grundlage für integrierte Segmentierungs- und Klassifikationsverfahren darstellt. Dies ergibt sich aus der Tatsache, daß der Viterbi-Algorithmus die Klassifikation durch die Bestimmung eines Schätzwertes für die Produktionswahrscheinlichkeit erlaubt und im selben Schritt eine Segmentierung durch das Aufdecken der wahrscheinlichsten Zustandsabfolge ermöglicht. Diese kombinierte Segmentierung und Klassifikation wird im folgenden Kapitel genutzt, um bei gedrehten Objekten in Bildern die Orientierung herauszufinden und diese zu erkennen.

# Kapitel 3

# Statistische Modellierung von Objekten in Bildern mit eindimensionalen Hidden-Markov-Modellen

Wie in vorhergehenden Kapiteln erwähnt wurde, wurden Hidden-Markov-Modelle ursprünglich bevorzugt bei der Zeitreihen-Klassifikation verwendet. In diesem Modellierungskontext konnte zunächst die große Flexibilität der Modelle effizient eingesetzt werden. In diesem Kapitel wird dargestellt, wie diese guten Modellierungseigenschaften der eindimensionalen Markov-Modelle erfolgreich auf Probleme der Bildklassifikation übertragen werden können und dies, obgleich Bilder eigentlich zweidimensionale Methoden erfordern.

## 3.1 Invariante Modellierung von Objektformen

Eindimensionale Hidden-Markov-Modelle können eingesetzt werden, um Formen von Objekten in natürlichen Bildern oder von handskizzierten Piktogrammen zu erkennen und dies selbst bei einer großen Formvariation innerhalb einer Objektklasse. Die Definition der *Form* eines Objekts schließt die Unabhängigkeit gegenüber den affinen Abbildungen Rotation, Translation und Größenskalierung ein. Eine derartige Aufgabe wird auch als *invariante* Erkennung bezeichnet. Die invariante Erkennung von Mustern wird als komplexe Aufgabe angesehen, die viele Anwendungen, wie z.B. maschinelle Zeichenerkennung (OCR), Zielidentifikation und industrielle Produktinspektion, ermöglicht. Die zur Lösung dieses Problems vorgeschlagenen Methoden reichen von (geometrischen-) Momenten und Integraltransformationen bis hin zu invarianten Klassifizierern (siehe auch den Übersichtsartikel [Woo96]).

Die Aufgabe, handskizzierte Piktogramme invariant zu erkennen, stellt aufgrund der Einbeziehung großer Variationen innerhalb der einzelnen Klassen durch die handschriftliche Eingabe zusätzliche, hohe Anforderungen. So variieren beispielsweise handschriftlich ein-

gegebene Buchstaben wesentlich mehr als maschinell reproduzierte Buchstaben und dies selbst bei einem einzelnen Schreiber. Dies führt zu besseren Erkennungsergebnissen bei Problemen der OCR im Vergleich zur schreiberabhängigen Handschrifterkennung. Somit muß die invariante Erkennung von handschriftlich eingegebenen Piktogrammen als anspruchsvolle Aufgabe angesehen werden. Dies gilt insbesondere, da gleichzeitig eine elastische Musterzuordnung, die aufgrund der handschriftlichen Eingabe erforderlich ist, und eine (rotations-)invariante Erkennung durchgeführt werden soll. Genau dies ist die Aufgabe, mit der sich ein großer Teil dieses Kapitels befaßt. Die im folgenden beschriebenen Arbeiten anderer Autoren behandeln ebenfalls diese Thematik.

He und Kundu beschreiben in [He91] eine Methode, die Konturen mittels Markov-Modellen und autoregressiven (AR)-Modellen klassifiziert. Sie verwenden lediglich eine 8-Klassen-Datenbasis, die einen Teil der in diesem Kapitel zur Evaluierung verwendeten und in Abb. 3.1 vorgestellten 20-Klassen-Datenbasis bildet. Da die Merkmale in [He91] aus Radien zwischen dem Schwerpunkt des Piktogramms und dessen Randkurve bestehen, können nur Symbole klassifiziert werden, die aus einer geschlossenen Randkurve bestehen und keine Piktogramme, wie die in Abb. 3.1 dargestellten Klassen 10, 11, 13 oder 16. Diese

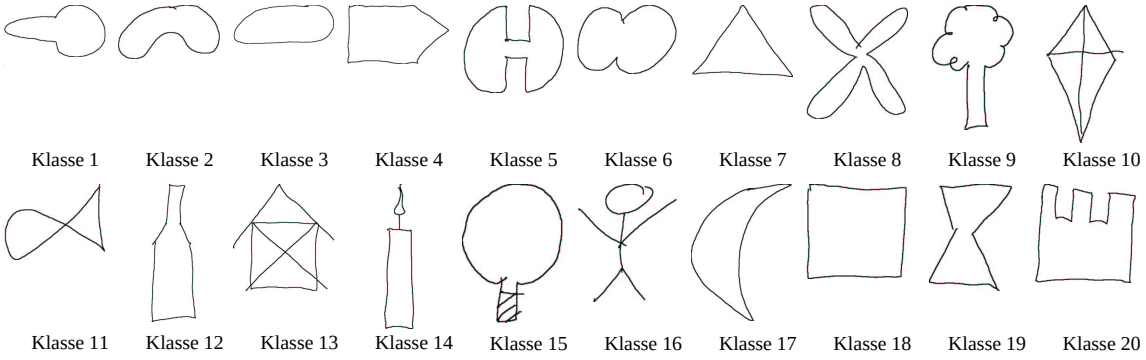


Abbildung 3.1: Beispiele aus den verwendeten 20-Klassen-Piktogramm-Datenbasen. Die ersten acht Klassen werden auch in [He91] und [Lee94] verwendet.

Art der Merkmalextraktion ist auch als sog. *Signature* in dem Buch von Gonzales und Woods [Gon92] beschrieben worden und ermöglicht bereits eine translations- und größenunabhängige Erkennung. Die angestrebte Rotationsinvarianz wird durch einen Vorverarbeitungsprozeß realisiert, der sukzessive in drei Schritten mittels geometrischer Betrachtungen sowie daraus abgeleiteten Entscheidungen die Piktogramme relativ zu einer festgelegten Achse ausrichtet. In einem ersten Schritt wird die Achse mit der längsten Ausdehnung (engl. elongation axis) basierend auf den zweiten Momenten des betrachteten Piktogramms bestimmt. Werden die Piktogramme unter Verwendung dieser berechneten Achse ausgerichtet, so ergeben sich häufig für Piktogramme derselben Klasse ähnliche Radiensequenzen. Dies ist jedoch nicht für alle Klassen der Fall. In [He91] wird beschrieben, daß diese Methode z.B. für Klasse 8 in Abb. 3.1 keine guten Ergebnisse zeigt. Daher wird ein weiterer Schritt basierend auf dem minimalen Radius eingeführt. Zusätzlich ist ein dritter Verarbeitungsschritt notwendig,

da die Ausrichtung der Piktogramme mit Hilfe der ersten beiden Schritte in einigen Fällen nicht eindeutig ist. Zusammenfassend kann gesagt werden, daß diese Vorverarbeitung, also das Ausrichten der Piktogramme zu einer vorgegebenen Achse, ein kompliziertes Verfahren darstellt. Lee und Lovell führen in [Lee94] Experimente auf einer sehr ähnlichen Piktogramm-Datenbasis durch und verwenden ebenfalls dieselben acht Klassen. Wie auch in [He91] werden als Merkmale die Radien zwischen dem Schwerpunkt und der Randkurve verwendet und die Rotationsinvarianz wird mit denselben Methoden ermöglicht. Der größte Unterschied zu [He91] liegt in der Verwendung eines Vektorquantisierers zur Klassifikation anstelle der Markov-Modelle.

In den folgenden Kapiteln wird eine für die invariante Erkennung geeignete und auf sog. Form-Matrizen basierende Merkmalextraktion eingeführt und daran anschließend eine HMM-Klassifizierungsmethode vorgestellt, die es ermöglicht, Piktogramme rotationsinvariant zu erkennen, ohne mittels komplizierter Vorverarbeitung die Piktogramme auszurichten. Die Hidden-Markov-Modelle sind in diesem Kontext auf spezielle Weise modifiziert worden, so daß sie zur rotationsinvarianten Erkennung eingesetzt werden können. Diese Rotationsinvarianz wird durch die Eigenschaft der Hidden-Markov-Modelle erreicht, gleichzeitig Klassifizieren und Segmentieren zu können. Die Grundidee ist dabei, eine Merkmalsequenz, die durch eine zweimalige polare Abtastung entstanden ist, mittels der Markov-Modelle in einen das Objekt unrotiert und vollständig repräsentierenden Anteil und in unvollständige Anteile zu unterteilen. Dieser Ansatz wurde durch Methoden, die in einer Unterdisziplin der Spracherkennung Verwendung finden, nämlich dem Auffinden von Schlüsselworten in fließend gesprochener Sprache, motiviert (siehe dazu [Ros90]). Dem Schlüsselwort entspricht in diesem Zusammenhang der das Objekt unrotiert und vollständig beschreibende Anteil der zweimalig abgetasteten Merkmalsequenz. Bevor diese Modellierung vorgestellt wird, wird zunächst die verwendete Merkmalextraktion beschrieben.

## 3.2 Merkmalextraktion

Eine Merkmalextraktion hat im wesentlichen die folgenden zwei Aufgaben zu erfüllen: Die Datenmenge, die zu einem Muster gehört, ist zu reduzieren, und ferner sind für die Klassifikationsaufgabe relevante Merkmale zu extrahieren. Der Nutzen der Datenreduktion liegt vor allem in einer kürzeren Laufzeit der Algorithmen und in der besseren Handhabbarkeit durch Digitalrechner. Das Extrahieren von relevanten Merkmalen kann auch als eine Irrelevanzreduktion angesehen werden. Die Art der Merkmalextraktion hat einen wichtigen Einfluß auf die Wahl eines geeigneten Klassifizierers.

Für den hier verwendeten HMM-Klassifizierer ist es, wie in Kapitel 2 dargestellt wurde, wichtig, eine Merkmalsequenz der Form $\vec{O} = \{\vec{o}_1, \ldots, \vec{o}_T\}$ zu erzeugen. In diesem Kapitel wird dieser Schritt mittels sog. Form-Matrizen (engl. shape matrix) durchgeführt. Diese Form-Matrizen basieren im wesentlichen auf einem polaren Abtastschema, welches von

Goshtasby in der Arbeit [Gos85] für binarisierte Objektformen eingeführt und evaluiert wurde. Dieser Ansatz wurde später von Taza und Suen methodisch verfeinert (siehe [Taz89]). Er wurde später auch in [Sab97] für die Unterschriftenverifikation eingesetzt und findet auch Erwähnung im Übersichtsartikel über Formanalyse-Techniken von Loncaric [Lon98].

Die polare Abtastung erfolgt auf adaptive Weise, was bereits zu Größen- und Translationsinvarianz führt. Eine Form-Matrix der Dimension $M \times N$ wird auf folgende Weise berechnet:

1: Bestimmung des Flächenschwerpunktes $\vec{S} = (x_s, y_s)^T$ der Objektform sowie des maximalen Radius $r_{max}$ des Objekts. Dabei wird $\vec{S}$ für ein gegebenes Bild $I(x,y)$ folgendermaßen bestimmt:

$$x_s = \frac{\sum_x \sum_y I(x,y) \cdot x}{\sum_x \sum_y I(x,y)} \qquad y_s = \frac{\sum_x \sum_y I(x,y) \cdot y}{\sum_x \sum_y I(x,y)} \tag{3.1}$$

Ist nun der Punkt $\vec{A} = (x_a, y_a)^T$ der von $\vec{S}$ am weitesten entfernte Bildpunkt, der noch Teil des Objekts ist, so ist $r_{max}$ der Betrag der Differenz von $\vec{S}$ und $\vec{A}$.

2: Setzen des Abtastintervalls in radialer Richtung auf $\Delta r = r_{max}/(M-1)$

3: Setzen des Abtastwinkels auf $\Delta\varphi = 2\pi/N$

4: Abtasten des Bildes $I$ bei Verwendung der Polarkoordinaten $(r, \varphi)$ nach folgendem Schema:

$$\begin{aligned} I_s(m,n) &= I\left(m \cdot \Delta r, n \cdot \Delta\varphi\right) \\ m &= 0, \ldots, M-1 \\ n &= 0, \ldots, N-1 \end{aligned} \tag{3.2}$$

5: $I_s(m,n)$ besteht nun aus $(M \cdot N)$ Abtastwerten und kann als Formmatrix angesehen werden.

Zu diesem Verfahren ist folgendes anzumerken: Bei der Berechnung des Flächenschwerpunktes $\vec{S}$ ist es zweckmäßig, eine Rundung der Werte $(x_s, y_s)^T$ vorzunehmen, da diese entsprechend Gleichung 3.1 im allgemeinen nicht ganzzahlig sind. Nach einer Rundung auf ganzzahlige Werte, definiert $\vec{S}$ einen Bildpunkt und keinen Punkt im *Zwischenpixelbereich*. Ähnlich wie bei der Bestimmung von $\vec{S}$ sind auch bei der polaren Abtastung in Schritt 4 wiederholt Rundungen vorzunehmen. Diese Abtastung ist zur Veranschaulichung in Abb. 3.2 schematisch dargestellt. Der Abb. 3.2 kann entnommen werden, daß die Dichte der Abtastwerte mit der radialen Entfernung vom Flächenschwerpunkt abnimmt. Dieses Phänomen, wurde in [Taz89] quantitativ untersucht mit der Absicht, ein Korrekturschema zu ermitteln, welches sich beim Vergleich zweier Form-Matrizen als vorteilhaft erweist. In [Taz89] wird

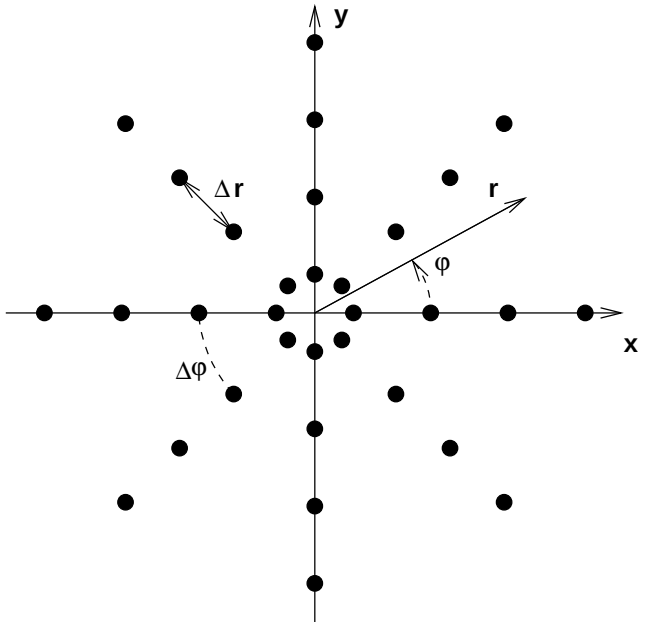


Abbildung 3.2: Schematische Darstellung der polaren Abtastung

von einer Form-Matrix der Größe $(M \times N)$ ausgegangen. Ferner ist dort die Anzahl der radialen Abtastwerte gleich dem (gerundeten) Radius. Eine weitere Annahme betrifft die Anzahl der Abtastpunkte auf dem Kreis mit dem Radius $r_{max}$, welche gleich der Anzahl der Bildpunkte auf diesem Kreis gesetzt wird. Zusammenfassend kann dies durch die folgenden Gleichungen beschrieben werden:

$$\begin{aligned} M &= \texttt{round}(r_{max}) \\ N &= \texttt{round}(2\pi \cdot r_{max}) \end{aligned} \tag{3.3}$$

In obiger Gleichung bezeichnet der Operator `round()` die Rundung auf die nächste ganze Zahl. Es gilt für diese Art der Abtastung allgemein, daß die Anzahl an Abtastwerten auf allen Kreisen konstant ist. Desweiteren nimmt die Anzahl an Bildpunkten auf den Kreisen nach außen hin zu. Aus diesen Betrachtungen, zusammen mit der Gleichung 3.3, kann geschlossen werden, daß die Abtastwerte auf dem Kreis mit dem Radius $r_{max}$ keine Redundanz enthalten, während für alle anderen Kreise die Redundanz mit abnehmendem Radius zunimmt. Diese Redundanz $R$ ist in [Taz89] für einen Kreis folgendermaßen definiert worden:

$$R = \frac{\text{Anzahl an Abtastwerten}}{\text{Anzahl an Bildpunkten}} \tag{3.4}$$

$$R = \frac{\text{Anzahl an Abtastwerten}}{\text{Umfang des Kreises}} \tag{3.5}$$

$$R = \frac{c}{\text{Umfang des Kreises}} \tag{3.6}$$

$$R = \frac{c}{2\pi \cdot r} \tag{3.7}$$

$$R = \frac{c^{\star}}{r} \tag{3.8}$$

Das Ergebnis der Überlegungen ist also, daß die Redundanz für einen gegebenen Kreis umgekehrt proportional ist zum Radius.

$$R \propto \frac{1}{r} \tag{3.9}$$

Weiterhin wurde das Gewicht $W$ als das Inverse zur Redundanz definiert und für weitere Betrachtungen verwendet.

Form-Matrizen wurden in [Gos85, Taz89] eingesetzt, um Objektklassen diskriminieren zu können. Um dies zu erreichen, wurden Abstandsmaße zwischen zwei Matrizen definiert, die den Vorteil aufweisen, wenig rechenintensiv zu sein. Da die Form-Matrizen nur binäre Elemente enthalten, wurden Matrixvergleiche vorgeschlagen, die im wesentlichen auf der XOR Operation basieren. Trotz der Möglichkeit, ein effizient zu berechnendes Abstandsmaß zu bieten, ist diese *direkte* Anwendung der Form-Matrizen für handschriftliche Muster wenig geeignet. So gibt es beispielsweise nur die Möglichkeit zwei Matrizen zu vergleichen und somit keinen echten Repräsentanten für eine Objektklasse. Betrachtet man beispielsweise die Objektklasse 13 in Abb. 3.1 und wurden während eines Trainings mehrere Beispiele für diese Klasse gesammelt, so gibt es bei der Form-Matrix-Methode keine Möglichkeit daraus einen einzelnen Repräsentanten zu berechnen. Die Berechnung eines einzelnen Repräsentanten oder Modells, welches auch die Variationen innerhalb einer Klasse berücksichtigt, ist hingegen sehr gut möglich bei der Verwendung von Hidden-Markov-Modellen. Um diese verwenden zu können, muß die Form-Matrix jedoch zuerst in eine Sequenz umgewandelt werden. Die für die Klassifikation mit Markov-Modellen benötigte Sequenz $\vec{O}$ kann durch die folgenden einfachen Schritte aus der Form-Matrix $I_s$ bestimmt werden. Jedem Vektor $\vec{o}$ werden die Elemente der Matrix $I_s(m,n)$ zugewiesen, für die der Index $n$ konstant ist. Die Sequenz $\vec{O}$ wird dann durch die Anordnung der Vektoren $\vec{o}$ derart erzeugt, daß die Werte von $n$ größer werden $(n = 0,\ldots,N-1)$. Um die Form-Matrizen besser als Merkmalextraktoren verwenden zu können, wurden folgende Veränderungen an dem Berechnungsschema vorgenommen: Statt des in Gleichung 3.2 angegebenen Abtastschemas wurde dieses Schema verwendet:

$$\begin{aligned} I_s(m,n) &= I\left(\Delta r\left(m+\tfrac{1}{2}\right), n\cdot\Delta\varphi\right) \\ m &= 0,\ldots,M-1 \\ n &= 0,\ldots,N-1 \end{aligned} \tag{3.10}$$

Dies entspricht einer Verschiebung der Abtastwerte in radialer Richtung um $\Delta r/2$. Der Grund hierfür ist eine Vermeidung der hohen Redundanz, wenn bei $r = 0$ wiederholt, wie in Gleichung 3.2 angegeben, abgetastet wird. Weiterhin wurden nicht die Bedingungen von [Taz89], nämlich die Gleichungen 3.3, die zur Ableitung von 3.9 verwendet wurden, eingehalten, sondern die Parameter $M$ und $N$ als variabel angesehen. Dies führte zu einem Unterabtastverhalten und somit wurde eine vorhergehende Tiefpaßfilterung erforderlich. Durch den hier beschriebenen Adaptionsvorgang sowie den Bezug zum Flächenschwerpunkt wird bereits eine Größen- und Translationsinvarianz erreicht, wohingegen die Rotationsinvarianz erst durch

Maßnahmen bei der Modellierung erreicht wird. Diese rotationsinvariante Modellierung ist Gegenstand des nächsten Unterkapitels.

## 3.3 Rotationsinvariante Modellierung

Die grundlegende Idee der rotationsinvarianten Modellierung mit Hidden-Markov-Modellen ist es, die entsprechend Kapitel 3.2 erzeugte Merkmalsequenz zu duplizieren und in dieser Sequenz mittels der kombinierten Segmentierungs- und Klassifizierungsfähigkeiten der Hidden-Markov-Modelle den Anteil der Sequenz zu erkennen, der dem unrotierten Anteil des Objekts bzw. des Piktogramms entspricht. Die duplizierte Merkmalsequenz kann als $\vec{O} = \{\vec{o}_1, \ldots, \vec{o}_T, \vec{o}_{T+1}, \ldots, \vec{o}_{2T}\}$ mit $\vec{o}_i = \vec{o}_{i+T}$ für $i = 1, \ldots, T$ dargestellt werden. Nach der Viterbi-Erkennung wird diese Sequenz Modellen zugeordnet, die einen ersten unvollständigen Teil eines Objekts oder eines Piktogramms, gefolgt von der Objektklasse selbst und schließlich den verbleibenden Anteil des Objekts repräsentieren. Die beiden unvollständigen Anteile des Objekts können dabei entweder von Modellen, die auf spezielle Weise trainiert wurden, oder mittels kopierter und modifizierter Piktogramm-HMMs beschrieben werden. Die Modelle, die ein spezielles Training mit Teilen der Merkmalsequenz durchlaufen haben, werden im folgenden mit *Teilmodell* bezeichnet.

Die rotationsinvariante Modellierung ist schematisch in Abb. 3.3 dargestellt. Abgebildet ist ein Piktogramm der Klasse 13 und dessen Segmentierung in Anteile, die den beiden Teilmodellen und dem Modell für Klasse 13 zugeordnet werden. Nach der Viterbi-Dekodierung werden die Merkmale, die entlang der gestrichelten Linie berechnet wurden, dem ersten Teilmodell zugeordnet. Die gepunktete Linie deutet die Merkmale an, die einem vollständigen Piktogramm, bzw. der Klasse 13 zugeordnet werden. Schließlich werden die Merkmale, die entlang der gestrichpunkteten Linie berechnet werden als vom zweiten Teilmodell generiert angesehen. Dabei ist anzumerken, daß die in der Abbildung dargestellten Hidden-Markov-Modelle entsprechend trainiert wurden. So wurde das Markov-Modell *Klasse 13* auf unrotierte Piktogramme trainiert und die Teilmodelle auf unvollständige Anteile derselben Klasse. Der Abb. 3.3 kann entnommen werden, daß, wenn die Zuordnung zwischen den Vektoren der Merkmalsequenz und den Hidden-Markov-Modellen bekannt ist, der Rotationswinkel über die Anzahl an Merkmalvektoren, die dem ersten oder zweiten Teilmodell zugeordnet wurden, bestimmt werden kann. Seien beispielsweise $f_1$ Merkmalvektoren dem ersten Teilmodell zugeordnet worden und weiterhin $f_2$ Merkmalvektoren dem zweiten Teilmodell, so berechnet sich der Rotationswinkel $\varphi^\star$ in Grad zu

$$\varphi^\star = \frac{f_1}{f_1 + f_2} \cdot 360^\circ \tag{3.11}$$

Diese Kombination aus Winkelbestimmung und Klassifikation während der Erkennungsphase ist wesentlich effizienter als die komplizierten Vorverarbeitungsschritte, die in [He91] und

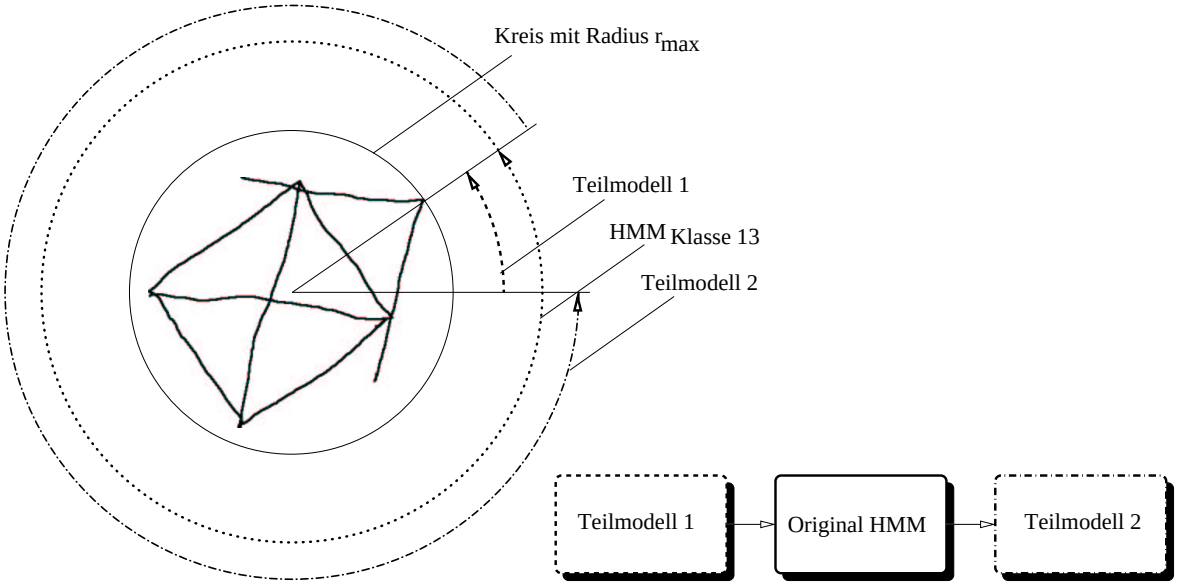


Abbildung 3.3: Zuordnung der Elemente der zweimalig präsentierten Merkmalsequenz zu den beiden Teilmodellen und dem Hidden-Markov-Modell für die Klasse 13

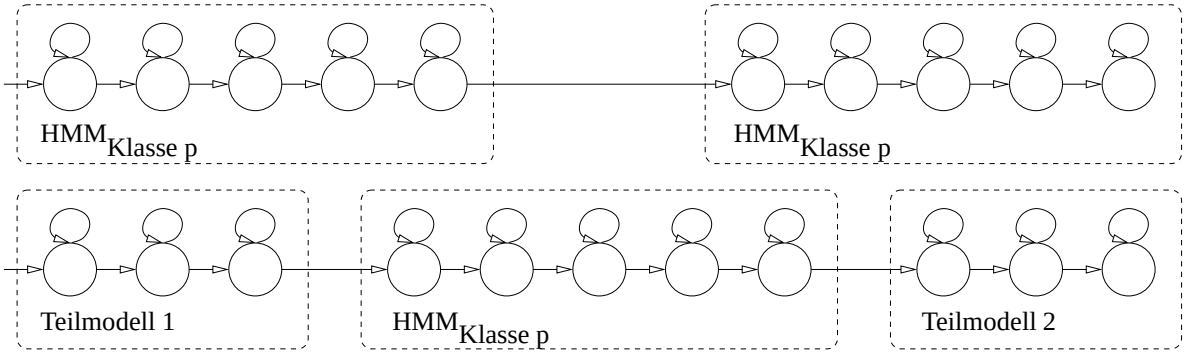


Abbildung 3.4: Zwei Hidden-Markov-Modelle repräsentieren eine einzige Piktogramm-Klasse. Das obere HMM modelliert unrotierte, das untere rotierte Piktogramme.

[Lee94] verwendet wurden und die in der Literaturübersicht in Unterkapitel 3.1 vorgestellt sind. Der Aufbau der zur rotationsinvarianten Erkennung verwendeten Hidden-Markov-Modelle wird im folgenden detailliert beschrieben. Es sind drei unterschiedliche Varianten im Rahmen dieser Arbeit entwickelt worden.

### 3.3.1 Modellierung mit Teilmodellen

Der erste vorgestellte Ansatz für die Modellierung rotierter Piktogramme besteht darin, daß das unrotierte Piktogramm-HMM mit Teilmodellen umgeben wird, die durch ein erstes Training mit den halbierten Merkmalsequenzen $\{\vec{o}_1, \ldots, \vec{o}_{T/2}\}$ und $\{\vec{o}_{T/2+1}, \ldots, \vec{o}_T\}$ aller Klassen initialisiert sind. Anschließend kann mit dem Baum-Welch-Algorithmus ein gemein-

sames Training der aneinandergehängten HMMs, wie in Abb. 3.4 dargestellt, erfolgen. In Abb. 3.4 wird ein zusätzliches Modell gezeigt, das aus der Verkettung zweier Piktogramm-HMMs besteht und speziell für unrotierte Piktogramme verwendet wird. Wenn eines dieser beiden HMMs für eine gegebene Merkmalsequenz die maximale Wahrscheinlichkeit $P(\vec{O}|\lambda)$ produziert, wird das Muster als zur *Klasse p* zugehörig klassifiziert. Die in den Experimenten in Unterkapitel 3.4 verwendeten Teilmodelle sind aus einer geringen Anzahl von Zuständen (3–5) aufgebaut, was zu einer schnellen Erkennung führt. Bedingt durch den Trainingsprozeß mit einer festen Anzahl von Beispielen werden von den Teilmodellen die während des Trainings präsentierten Rotationswinkel bevorzugt. Dies kann bei einigen Anwendungen erwünscht sein, beispielsweise wenn in geringem Maße rotierte Ziffern wie ”6” und ”9” unterschieden werden sollen. Um eine vollständig rotationsinvariante Erkennung zu ermöglichen, kann das im nächsten Abschnitt beschriebene Verfahren eingesetzt werden.

### 3.3.2 Modellierung mit modifizierten Wahrscheinlichkeiten für die Anfangs- und Endzustände

Eine weitere Möglichkeit für eine rotationsinvariante Modellierung besteht darin, das Modell für die unrotierten Piktogramme zweimal zu duplizieren, und das Original-HMM mit diesen Modellkopien zu umgeben. Zusätzlich werden die Wahrscheinlichkeiten für die Anfangszustände des ersten Modells ($\vec{\pi}$) sowie die Wahrscheinlichkeiten für die Endzustände des dritten HMM entsprechend Abb. 3.5 verändert. Die Komponenten von $\vec{\pi}$ des ersten Modells werden alle auf den gleichen Wert $1/N$ gesetzt, wobei $N$ die Dimension des Vektors $\vec{\pi}$ bzw. die Anzahl von HMM-Zuständen des original Modells bezeichnet. Dieser Schritt ist durch die Annahme motiviert, daß alle möglichen Drehungswinkel gleichwahrscheinlich sind. Bei dem Versuch dies auf die Endzustände zu übertragen, sieht man sich dem Problem konfrontiert, daß die Übergänge von einem bestimmten Zustand aus sich immer zu *eins* aufsummieren müssen ($\sum_j a_{ij} = 1$). Die beste Möglichkeit diesem Problem zu begegnen, ist es, die Endzustandswahrscheinlichkeit auf einen beliebigen Wert zu setzen (z.B. ebenfalls $1/N$) und die *Größenverhältnisse* der anderen Übergänge desselben Zustandes zu erhalten. Diese Maßnahme ermöglicht die rotationsinvariante Erkennung von Mustern ohne eine Bevorzugung eines bestimmten Winkels. Die Endzustände des dritten Modells in Abb. 3.5 werden entsprechend verändert und somit ist es möglich, daß sich das Markov-Modell in jedem seiner Zustände befinden kann, wenn der letzte Merkmalvektor einer Sequenz präsentiert wurde. Dies wurde bei den bisher beschriebenen Modellen (z.B. Abb. 3.4) nicht gefordert. Diese Modellierungstechnik kann auch als ein Nachbearbeitungsschritt nach einem Training entsprechend dem in Kap. 3.3.1 beschriebenen Verfahren angewendet werden. In diesem Fall werden die Teilmodelle nach dem ersten Training entfernt und die Modelltopologie entsprechend Abb. 3.5 wird für das Piktogramm-HMM erzeugt. Alternativ können Modelle einer solchen Topologie auch direkt trainiert werden, in diesem Fall würde jedoch der

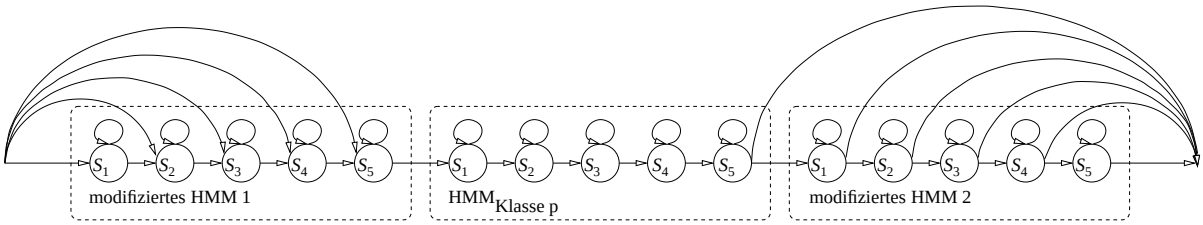


Abbildung 3.5: Bei der zweiten vorgestellten Methode werden drei identische, jedoch bezüglich der Wahrscheinlichkeiten für die Anfangs- und Endzustände veränderte, Hidden-Markov-Modelle verwendet.

Vektor der Anfangszustände $\vec{\pi}$ geändert werden und sich somit bevorzugte Rotationswinkel eintrainieren. Es sei hier noch angemerkt, daß die Viterbi-Dekodierung für diese Methode langsamer abläuft, als die zuvor präsentierte Methode aus Unterkapitel 3.3.1, da die Anzahl an Modellzuständen für die einzelnen Modelle höher ist. Nach einer Dekodierung kann der Rotationswinkel entsprechend Gleichung 3.11 bestimmt werden.

### 3.3.3 Zyklische Vertauschung der HMM-Zustände

Die dritte Modellierungstechnik, die in diesem Kapitel vorgestellt wird, verwendet eine erheblich höhere Anzahl von Hidden-Markov-Modellen pro Klasse als die bisher beschriebenen Methoden, erfordert jedoch lediglich die Präsentation der nicht duplizierten Merkmalsequenz $\{\vec{o}_1, \ldots, \vec{o}_T\}$. Für jede Klasse wird das HMM für unrotierte Piktogramme $N$-mal kopiert und die Zustände, wie in Abb. 3.6 angedeutet, sukzessive zyklisch permutiert. Der Rotationswinkel kann nun nicht mehr wie in Unterkapitel 3.3 angedeutet, ermittelt werden, ist jedoch über die Konfiguration desjenigen Hidden-Markov-Modells mit der maximalen Wahrscheinlichkeit für eine gegebene Sequenz bestimmbar. Beispielsweise repräsentiert das Modell in Abb. 3.6 oben die Piktogramme der *Klasse p* mit einem Rotationswinkel von $0°$, das Modell darunter einen Winkel von $360°/N$, usw. Die Winkel können bei diesem Ansatz also nur in Vielfachen von $360°/N$ angegeben werden und sind damit weniger genau bestimmbar als bei den in den Abschnitten 3.3.1 und 3.3.2 beschriebenen Methoden, wo die Quantisierung auf der Zuordnung der größeren Anzahl von Merkmalvektoren zu den einzelnen Modellen beruht. Dieses Verfahren wurde ausschließlich als ein zusätzlicher Schritt im Anschluß an ein Training entsprechend Kapitel 3.3.1 angewendet. In diesem Fall ist eine rotationsinvariante Erkennung ohne bevorzugte Rotationswinkel möglich. Es ist ebenfalls möglich, dieses Verfahren direkt anzuwenden. In diesem Fall ist dann eine erheblich aufwendigere manuelle Indexierung der Trainingsdaten erforderlich. Hervorgerufen durch die im Vergleich zu den anderen Verfahren erheblich gestiegene Anzahl an Modellen, ist die Viterbi-Dekodierung recht langsam. Um diese Dekodierung zu beschleunigen, können die

Parameter der Modellzustände innerhalb einer Klasse verknüpft werden und dadurch die Zeit, die für die Berechnung von Ausgabewahrscheinlichkeiten benötigt wird, erheblich reduziert werden.

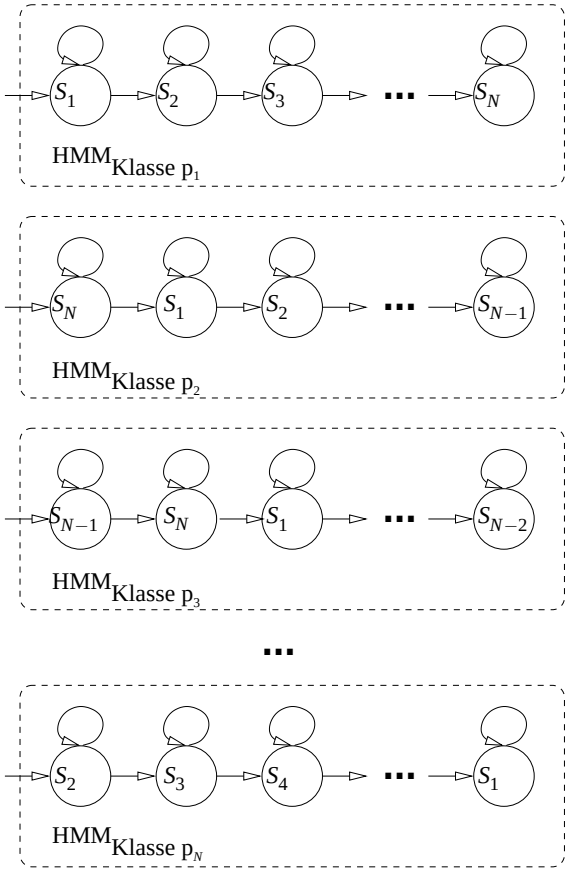


Abbildung 3.6: Zyklisches Vertauschen der HMM-Zustände

## 3.4 Experimentelle Ergebnisse und Vergleich mit Momentenmethoden

Die Evaluierung der vorgestellten Methoden erfolgt mit Hilfe zweier großer Piktogramm-Datenbasen. Nachdem diese im folgenden Abschnitt kurz vorgestellt werden, sind im Anschluß daran quantitative Ergebnisse angegeben. Diese Erkennungsergebnisse werden dann verglichen mit konventionellen Erkennungsmethoden, basierend auf invarianten Merkmalen wie geometrischen und Zernike-Momenten zusammen mit künstlichen neuronalen Netzen als Klassifikatoren.

### 3.4.1 Datenbasis mit rotierten Piktogrammen

Die zwei Datenbasen wurden von zwei verschiedenen Personen erstellt, die im folgenden mit *stm* und *dib* bezeichnet werden. Beide Datenbasen bestehen jeweils aus 10 unrotierten und 20 rotierten handskizzierten Piktogrammen für jede der 20 in Abb. 3.1 vorgestellten Klassen. Die Handskizzen wurden mit einem Grafiktableaus aufgenommen und als zweiwertige Bilder abgespeichert. Die 20 rotierten Piktogramme sind zu gleichen Teilen in einen Test- und einen Trainingsdatensatz aufgeteilt worden. Diese Aufteilung erfolgte nach dem Zufallsprinzip und wurde zu Beginn der Experimente festgelegt und nicht mehr verändert. Um die großen Formvariationen innerhalb der einzelnen Klassen zu illustrieren, sind 10 Beispiele der Klasse 9 aus der *stm*-Datenbasis in Abb. 3.7 dargestellt.

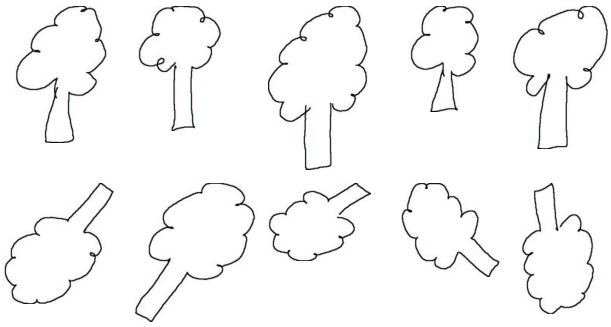

Abbildung 3.7: Fünf unrotierte und fünf rotierte Piktogramme der Klasse 9 aus der *stm*-Datenbank.

### 3.4.2 Quantitative Ergebnisse mit rotationsinvarianten HMMs auf einer Piktogramm-Datenbasis

Die verwendeten kontinuierlichen Hidden-Markov-Modelle bestehen aus 30 Zuständen für die Piktogramm-HMMs und fünf Zuständen für die Teilmodelle. Jede der Ausgabedichten der Modellzustände ist aus vier multivariaten Gaußverteilungen zusammengesetzt. Die Merkmalextraktion wurde mit fünf Abtastwerten bei jeweils $\Delta\varphi = 10^\circ$ durchgeführt (siehe auch Gleichung 3.11). Die mit diesen Parametern erzielten Erkennungsgenauigkeiten sind in Tabelle 3.1, getrennt für die *stm* und *dib*-Datenbasen, aufgeführt. In der ersten Zeile der Tabelle sind die Erkennungsergebnisse für die Modellierungstechnik entsprechend Kapitel 3.3.1, also der Modellierung mit den Teilmodellen, angegeben. Die folgenden beiden Zeilen enthalten die Ergebnisse für die Modellierung mit modifizierten Wahrscheinlichkeiten für die Anfangs- und Endzustände (siehe Kapitel 3.3.2). Dabei werden die Ergebnisse getrennt angegeben für die Verwendung dieser Modellierung als einen zusätzlichen Schritt nach einem Training mit Teilmodellen und für den Fall, daß direkt ein gemeinsames Training mit diesen modifizierten Modellen durchgeführt wird. In Zeile 4 sind die Erkennungsergebnisse für die Experimente mit zyklisch vertauschten Modellen angegeben. Die etwas abgesetzten

| Verwendete Methode | Siehe auch | *stm* | *dib* | Mittelwert |
|---|---|---|---|---|
| Gemeinsames Training mit Teilmodellen | Kap. 3.3.1 | 99,5% | 98,5% | 99,0% |
| Gem. Training mit Teilmodellen + modifizierte HMMs | Kap. 3.3.2 | 99,5% | 98,5% | 99,0% |
| Modifizierte HMMs ohne Teilmodelle | Kap. 3.3.2 | 97,5% | 92,5% | 95,0% |
| Gem. Training mit Teilmodellen + zyklische Vertauschung | **Kap. 3.3.3** | **99,5%** | **99,5%** | **99,5%** |
| Nur HMMs mit modifiziertem $\vec{\pi}$ (ohne gem. Training) | Kap. 3.3.2 | 96,5% | 95,0% | 95,8% |
| Zyklisch permutierte HMMs (ohne gem. Training) | **Kap. 3.3.3** | **97,5%** | **98,5%** | **98,0%** |

Tabelle 3.1: Erkennungsergebnisse, die mit den vorgestellten HMM-basierten Methoden erzielt wurden

Zeilen 5 und 6 zeigen die Erkennungsgenauigkeiten für die Modellierung entsprechend den Kapiteln 3.3.2 und 3.3.3 für den Fall, das kein Vortraining mit den Teilmodellen durchgeführt wurde. Dies bedeutet, das keine rotierten Piktogramme zum Training verwendet wurden und somit auch die Anzahl der insgesamt für das Training verwendeten Beispiele halbiert wurde. Die Modelle wurden ausschließlich mit unrotierten Mustern trainiert und anschließend entsprechend der jeweiligen Methode modifiziert um die rotationsinvarinaten Eigenschaften zu erzielen.

Insgesamt zeigen die vorgestellten Methoden gute Ergebnisse mit hohen Erkennungsgenauigkeiten, die vergleichbar sind mit denen, die in den Arbeiten [He91] und [Lee94] veröffentlicht wurden. Es muß jedoch beachtet werden, daß die hier verwendeten Datenbasen mehr als doppelt so viele Klassen aufweisen als die Datenbasen in den referenzierten Arbeiten. Wenn die Modelle mit den modifizierten Anfangs- und Endzuständen während eines gemeinsamen Trainings verwendet werden, so werden vergleichsweise schlechte Erkennungsergebnisse erzielt. Dies liegt an einer Überanpassung an die während des Trainings gesehenen Piktogramme und deren Rotationswinkel. Die Komponenten des Vektors der Wahrscheinlichkeiten für die Einnahme eines Anfangszustandes ($\vec{\pi}$) nehmen in diesem Fall Werte an, die von $1/N$ verschieden sind und somit werden bestimmte Rotationswinkel bevorzugt. Die anderen Methoden zeigen sehr viel bessere Ergebnisse, wobei die zyklisch permutierten Modelle mit einem sehr kleinen Abstand die besten Erkennungsraten erzielt haben. Es sei jedoch darauf hingewiesen, daß die letztgenannte Methode die meiste Rechenzeit benötigt und ferner die ungenausten Schätzungen für den Rotationswinkel liefert.

### 3.4.3 Quantitative Ergebnisse bei Verwendung von Momentenmethoden

Um eine detailliertere Bewertung der vorgestellten Methoden zu ermöglichen, wurden Experimente auf denselben Datenbasen mit invarianten Momenten und einem neuronalen Klassifizierer durchgeführt. Translations-, rotations- und skalierungsinvariante Merkmale, die auf geometrischen Momenten basieren, wurden von Hu in [Hu62] eingeführt. Diskrete geometrische Momente eines Bildmusters $f(x,y)$ der Ordnung $(p+q)$ sind durch folgende Gleichung gegeben:

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x,y) \tag{3.12}$$

Diese Momente sind nicht translationsinvariant und daher werden Zentalmomente der folgenden Form verwendet

$$v_{p,q} = \sum_x \sum_y x^p y^q f(x+x_0, y+y_0) \tag{3.13}$$

Dabei ist der Punkt $(x_0,y_0)^T$ der Flächenschwerpunkt entsprechend Gleichung 3.1. Sogenannte normalisierte Momente können mit

$$\mu_{p,q} = \frac{v_{p,q}}{v_{0,0}^{1+(p+q)/2}} \tag{3.14}$$

berechnet werden. Diese sind sowohl translations- als auch skalierungsinvariant. Durch eine nichtlineare Kombination dieser normalisierten Momente bis zur Ordnung drei konnte Hu sieben invariante Momente erzeugen. Diese Hu-Momente sind z.B. in [Hu62] oder [Woo96] aufgelistet. Später listete Li in [Li92] 52 invariante Momente bis zur Ordnung neun auf. Andere Momente wurden verwendet, wie beispielsweise Legendre-, komplexe, oder Zernike-Momente. In [Teh88] wird gezeigt, wie Zernike-Momente aus geometrischen Momenten berechnet werden können:

$$A_{n,l} = \frac{n+1}{\pi} \sum_{\substack{k=|l| \\ n-k=even}}^{n} \sum_{j=0}^{q} \sum_{m=0}^{|l|} w^m \cdot \binom{q}{j} \binom{|l|}{m} B_{n|l|k} \mu_{k-2j-m,2j+m} \tag{3.15}$$

Dabei ist $A_{n,l}$ das komplexe Zernike-Moment, $w = \begin{cases} -i & : \quad l > 0 \\ +i & : \quad l \leq 0 \end{cases}$, $q = \frac{1}{2}(k-|l|)$ und $i = \sqrt{-1}$. $B_{n|l|k}$ ist gegeben durch

$$B_{n|l|k} = \frac{(-1)^{(n-k)/2}((n+k)/2)!}{((n-k)/2)!((|l|+k)/2)!((k-|l|)/2)!} \tag{3.16}$$

Experimente wurden auf beiden Piktogramm-Datenbasen mit den sieben Hu-Momenten, den 52 Li-Momenten sowie den Zernike-Momenten als invariante Merkmale durchgeführt. Als Klassifizierer wurde ein künstliches neuronales Netz vom Typ mehrschichtiges Perzeptron

| Verwendete Methode | *stm* | *dib* | Mittelwert |
|---|---|---|---|
| 7 Hu-Momente | 52,0% | 46,5% | 49,25% |
| **52 Li-Momente** | **99,0%** | **99,0%** | **99,0%** |
| Zernike-Momente | 99,5% | 96,5% | 98,0% |

Tabelle 3.2: Erkennungsergebnisse, die mit Momenten in Kombination mit neuronalen Netzen erzielt wurden

mit einer verdeckten Schicht verwendet. Die Ergebnisse sind in Tabelle 3.2 wiedergegeben. Tabelle 3.2 kann entnommen werden, daß obwohl die erzielten Erkennungsgenauigkeiten insbesondere bei Verwendung der Li-Momente sehr hoch sind, diese nicht ganz an die Genauigkeit der HMM-basierten Methode heranreichen (vgl. Tabelle 3.1, Zeile 4). Es existieren jedoch weitere Argumente, die die Verwendung der rotationsinvarianten Markov-Modelle nahelegen. So ist ein detailliertes Mustermatching möglich, welches auch die Anwendung auf natürlichen Bildern ermöglicht (siehe Kapitel 3.5). Außerdem wird bei der Viterbi-Dekodierung zusätzlich ein Schätzwert für den Drehungswinkel ausgegeben. Dies ist bei den Methoden die auf Momenten basieren nicht der Fall.

## 3.5 Inhaltsbasierter Zugriff auf Objekte in Bilddatenbanken

Die guten Ergebnisse, die mit den neuartigen Methoden im vorangegangenen Unterkapitel erzielt wurden, motivierten die Durchführung weiterer Experimente, allerdings mit natürlichen Bildern. In der Arbeit [Wal98] sind erste Experimente vorgestellt, die die Klassifizierung von natürlichen Bildern mit den Methoden der Kapitel 3.2 und 3.3 beschreiben. Die rotationsinvarinate Modellierung kann also erfolgreich auf natürliche Bilder übertragen werden. Ebenfalls in [Wal98] sind erste Versuche unternommen worden, Merkmalextraktion und Modellierung für ein experimentelles Bilddatenbanksystem, das mit Skizzen abgefragt werden kann, einzusetzen. Diese Experimente und die daran anschließenden Versuche werden in diesem Kapitel vorgestellt. Zunächst ist jedoch eine kurze Begriffsbestimmung sowie eine Einführung in die Literatur zu diesem Thema erforderlich.

### 3.5.1 Relevante Arbeiten anderer Autoren zum Thema inhaltsbasierte Bilddatenbankabfragen

Auf Bildern in Bilddatenbasen wird herkömmlicherweise über textuelle Anfragen zugegriffen. Dies erfordert die Indexierung des gesamten Bildbestandes und ist eine zeitaufwendige Arbeit. Zusätzlich können durch unterschiedliche Interpretationen des Bildinhaltes bei dem

Ersteller und dem Benutzer der Datenbank unbefriedigende Abfrageergebnisse entstehen. Neben diesen Nachteilen ist auch die textuelle Anfrage wenig intuitiv für den Benutzer. Dieser bevorzugt oftmals benutzerfreundliche Mensch-Maschine-Schnittstellen wie beispielsweise Graphiktableau, Computermouse oder sensitive Bildschirme (touch screen) anstelle von Tastaturen. Durch das Aufkommen von großen Bilddatenbasen in den verschiedensten Anwendungsbereichen entstand der Bedarf nach automatischen Abfragesystemen, die einen einfachen und intuitiven Zugriff auf die Bilder ermöglichen. Aus diesen Gründen wurden verschiedene Systeme entwickelt, die die Abfrage von Bilddatenbanken entweder über ein Beispielbild oder über visuelle Bildmerkmale ermöglichen ([Pen94, Fli95, Lin97, Smi97]). Eine verbreitete Anfrage an ein solches System ist *welches Objekt (Form oder Bild) in der Datenbasis gleicht oder ähnelt dem vorgegebenen Objekt oder Bild ?* (aus [Meh97]). Diese Art der Anfrage wird inhaltsbasierte Bilddatenbankabfrage genannt (engl. Content-Based Image Retrieval, CBR).

In der Arbeit [Smi97] beschreiben Smith und Chang ein Bildabfragesystem für das World-Wide-Web (WWW). Die inhaltsbasierte Abfrage von Bilddaten im WWW stellt eine anspruchsvolle Aufgabe dar, da für die große Datenmenge des WWW effiziente und skalierbare Algorithmen erforderlich sind. Das System in [Smi97] benutzt jedoch Histogramme von Anfragebildern, um Bilder im WWW mit ähnlichem Bildinhalt aufzuspüren. Dies hat zum einen den Nachteil, daß möglicherweise kein Beispielbild vorliegt und methodisch das Problem, daß nur globale Merkmale verwendet werden und keine detaillierten Übereinstimmungen geprüft werden. Anfragen über Skizzen bzw. Objektformen werden in [Pen94, Fli95, Del97] verwendet und somit können auch Anfragen an diese Systeme gestellt werden, ohne daß ein Beispielbild vorliegen muß. In den referenzierten Arbeiten wurden jedoch wichtige Eigenschaften, wie z.B. Rotations- und Skalierungsinvarianz noch nicht erreicht. Rotationsinvarianz ist eine wichtige Eigenschaft, wenn die Anfrage eine handschriftlich eingegebene Skizze ist, da wie bei der Handschrift selbst, die Skizzen oft Schräglagen aufweisen. Del Bimbo und Pala beschreiben in [Del97] einen Algorithmus (*elastic matching algorithm*), der in der Lage ist, auch bei geringen Drehungen von 12 bis 15 Grad noch zu funktionieren. Die Fähigkeit solche geringen Drehungen verarbeiten zu können, sollte ausreichen, um die Drehungen die durch die handschriftliche Eingabe verursacht werden, verarbeiten zu können. Es reicht jedoch nicht aus, um Datenbasen mit beliebig orientierten Objekten bearbeiten zu können. Die hier verwendete Datenbasis besteht überwiegend aus Abbildungen von Werkzeugen, die keine fest vorgegebenen Orientierungen aufweisen. Ein Abfragesystem für diese Datenbasis erfordert also einen vollständig rotationsinvarianten Abfragemodus. Aus diesem Grund können die Methoden der Kapitel 3.2 und 3.3 für diese Aufgabe verwendet werden.

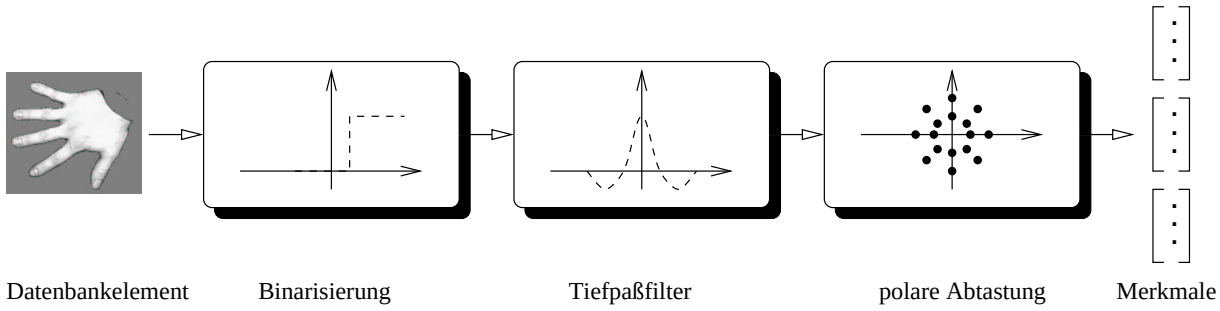


Abbildung 3.8: Vorverarbeitung und Erzeugung der Merkmalsequenz

### 3.5.2 Skizzenbasierte Bilddatenbankabfrage

In der Arbeit [San96] geben Santini und Jain an, daß bei Bilddatenbankabfragen die Ergebnisbilder entsprechend eines Ähnlichkeitsmaßes bezüglich des Anfragebildes geordnet ausgegeben werden sollten. Eine solche auf einem Ähnlichkeitsmaß basierende Ergebnisliste kann sehr effizient durch die Verwendung von Hidden-Markov-Modellen und deren Wahrscheinlichkeitsmaße erzeugt werden. Aus diesem Grund bietet sich die Verwendung der rotationsinvarianten Hidden-Markov-Modelle aus den vorherigen Kapiteln zusammen mit der beschriebenen Merkmalextraktion für eine formbasierte Bilddatenbankabfrage an.

Es ist jedoch noch eine Auswahl aus den drei, in den Kapiteln 3.3.1 bis 3.3.3 vorgestellten, Modellierungsvarianten zu treffen. Da keine Daten für ein gemeinsames Training von Objektmodell und den Teilmodellen vorliegen, können die Methoden, die zu den Ergebnissen in Tabelle 3.1 in den Zeilen 1–4 geführt haben, nicht verwendet werden. Für ein solches gemeinsames Training würde eine große Anzahl von gedrehten Beispielen benötigt. Da die einzelnen Datenbankelemente durch die Hidden-Markov-Modelle repräsentiert werden sollen, liegt nur ein einziges Trainingsbeispiel vor. Bei den verbleibenden Modellierungen bietet sich die Modellierung mit den modifizierten Wahrscheinlichkeiten für die Anfangs- und Endzustände aufgrund der günstigeren Rechenzeiten an, da nur ein einzelnes Hidden-Markov-Modell die Objekte beschreibt und nicht eine Vielzahl von Modellen.

Die Vorverarbeitung und Merkmalextraktion kann im wesentlichen wie in Kapitel 3.2 dargestellt durchgeführt wurden. Dabei ist jedoch zu beachten, daß nun Grauwertbilder anstellte von binären Bildern bzw. Piktogrammen vorliegen. Daher ist vor den schon vorgestellten Verarbeitungsschritten, nämlich der Tiefpaßfilterung, der Unterabtastung und der Sequenzerzeugung, eine histogrammbasierte Trennung von Objekt und Hintergrund vorzunehmen. Dieser Schritt kann bei der vorliegenden Bilddatenbank auf einfache Weise durchgeführt werden, da sich die Objekte auf einem weitgehend homogenen Untergrund befinden. Die durchzuführenden Schritte bei der Merkmalerzeugung sind in der Abbildung 3.8 zusammen mit einem Objekt der Datenbasis dargestellt. Die so für jedes Datenbankelement erzeugten Merkmalsequenzen werden anschließend für das Baum-Welch-Training von Links-Rechts-Modellen entsprechend der Abbildung 2.1 verwendet. Da solche Links-

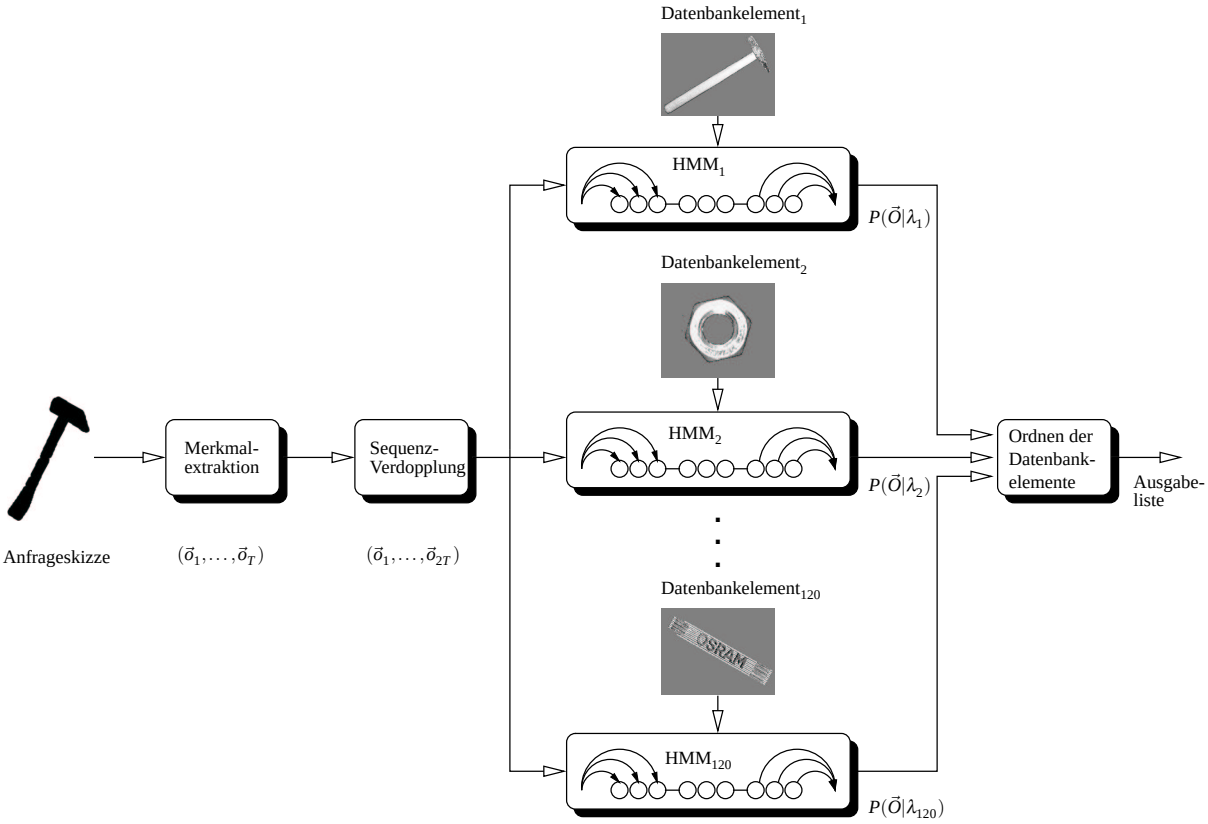


Abbildung 3.9: Schematische Darstellung des skizzenbasierten Bilddatenbankabfragesystems

Rechts-Modelle die Datenbankobjekte jedoch nicht rotationsinvariant beschreiben, sind diese Modelle zweimal zu duplizieren, an das lineare Modell anzuhängen und die Wahrscheinlichkeiten für die Anfangs- und Endzustände entsprechend Abbildung 3.5 zu modifizieren. Wenn für jedes Datenbankelement ein solches modifiziertes Hidden-Markov-Modell erzeugt wurde, kann dem System eine Skizze gezeigt werden. Anschließend können die berechneten Produktionswahrscheinlichkeiten der Modelle verwendet werden, um die Datenbankelemente entsprechend ihrer Ähnlichkeit mit der Anfrageskizze zu sortieren. Falls der Benutzer nur an den fünf ähnlichsten Bildern interessiert ist, können Pruning-Techniken eingesetzt werden, um die Suche entsprechend zu beschleunigen. Dies ist ein weiterer Vorteil bei der Verwendung von Hidden-Markov-Modellen. Es ist zu beachten, daß die Merkmalsequenz der Anfrageskizze entsprechend den Ausführungen in Unterkapitel 3.3.2 zu duplizieren ist. Der Aufbau des skizzenbasierten Bilddatenbankabfragesystems ist schematisch in Abbildung 3.9 dargestellt.

Das Vorgehen bei der Vorverarbeitung der Datenbankelemente, vor der Merkmalextraktion und der statistischen Modellierung, beeinflußt die Art in der die Anfrageskizzen erstellt werden sollten. Zwei verschiedene Anfrageskizzen sind in Abbildung 3.10 zusammen mit den entsprechend vorverarbeiteten Datenbankbildern gezeigt. Die obere Skizze in der Ab-

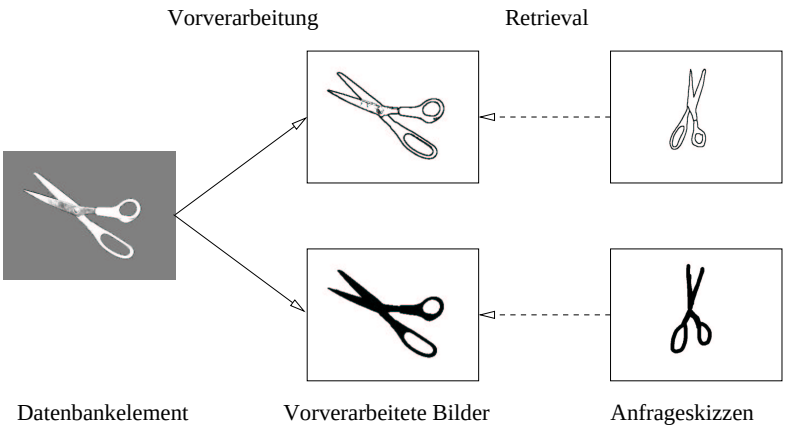


Abbildung 3.10: Abfrageskizzen und entsprechend vorverarbeitete Datenbankelemente

bildung sollte zu einem hohen Ähnlichkeitsmaß bei kantenverstärkten und binarisierten Abbildungen von Scheren führen, wohingegen die untere Skizze ein hohes Ähnlichkeitsmaß bei schwellwertbasiert binarisierten Bildern von Scheren ergibt. Obwohl die vorgestellten Merkmalextraktions- und Modellierungsmethoden in beiden Fällen gute Ergebnisse zeigen, wurde für die folgenden Experimente der untere Fall in Abbildung 3.10 gewählt. Diese Art von Skizzen sind mit der Computermouse oder mit einem Graphiktableau erheblich leichter anzufertigen. Zusätzlich hat die Kantenextraktionsmethode den Nachteil, daß ohne weitere algorithmischen Maßnahmen, die Liniendicke bei der Skizze und den vorverarbeiteten Bildern übereinstimmen sollte, um ein hohes Ähnlichkeitsmaß zu erzeugen. Skeletonisierung kann eingesetzt werden, um diesem Problem zu begegnen. Dies führt jedoch zu einem erhöhten Rechenaufwand.

Die in den Experimenten verwendete und in der Arbeit [Wal98] vorgestellt Bilddatenbasis besteht aus 120 Objekt-Bildern, überwiegend von Werkzeugen. Diese Objekte sind nicht ausgerichtet und haben eine zufällige Orientierung. Die Größe der Bilder entspricht $352 \times 264$ Bildpunkten. Die in den Experimenten verwendete Modellgröße beträgt 30 Zustände für die Links-Rechts-Modelle und somit 90 Zustände für die Strukturen entsprechend Abb. 3.5. Die Merkmalextraktion wurde mit fünf Abtastwerten bei $\Delta\varphi = 10^\circ$ durchgeführt. Somit ergibt sich für die Sequenz $\vec{O} = \{\vec{o}_1, \ldots, \vec{o}_{2T}\}$ eine Gesamtlänge von 72. Die Anfrageskizzen wurden mit der Computermouse und dem Zeichenprogramm *XPaint* bei einer Größe von $640 \times 480$ Bildpunkten erzeugt. Bei den Skizzen wird eine ähnliche Merkmalextraktion wie in Abbildung 3.8 dargestellt verwendet, jedoch kann die Trennung von Objekt und Bildhintergrund entfallen. Die Abbildung 3.11 zeigt Ergebnisse, die mit diesen Parametern erzielt wurden. Jeweils in der ersten Spalte befindet sich die Anfrageskizze und in den folgenden drei Spalten sind die Datenbankelemente mit dem höchsten Ähnlichkeitsmaß (von links nach rechts abnehmend) gezeigt. Der Abbildung kann entnommen werden, daß es mit diesem System möglich ist, Bildobjekte mit einfachen Skizzen abzufragen. Weitere Abfrageergebnisse, die mit diesem System erzielt wurden, sind in den Arbeiten [Mul98a, Mul98c]

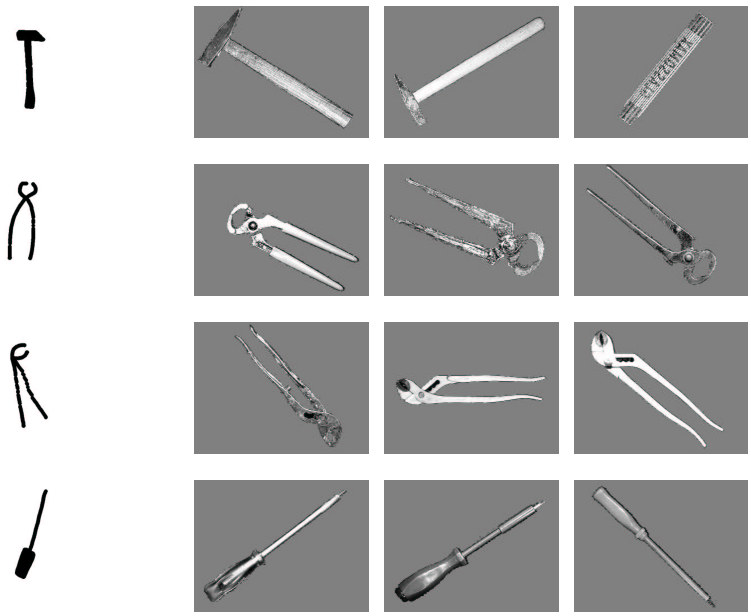

Abbildung 3.11: Anfrage-Skizzen und die vom Datenbanksystem ermittelten drei ähnlichsten Grauwertbilder

und [Mul99g] zu finden. Um eine weitergehende Evaluierung auch mit verschiedenen Benutzern zu ermöglichen, wurde ein Internet-basiertes Bilddatenbanksystem entwickelt. Dieses Internet-basierte System wird in Kapitel 3.5.6 ausführlich beschrieben, allerdings für ein kombiniertes Farb- und Formretrieval. Diese kombinierte Abfrage ist Gegenstand des nun folgenden Unterkapitels.

### 3.5.3 Integrierter Ansatz zur farb- und formbasierten Bilddatenbankabfrage

Eine Möglichkeit, das in Unterkapitel 3.5.2 vorgestellte System zu erweitern, besteht darin, Farbe in den Abfrageprozess einzubeziehen (siehe auch [Mul99b] und [Mul01]). Durch die Verwendung der Markov-Modelle ist es möglich, Farb- und Formmerkmale in ein einzelnes statistisches Modell zu integrieren. Dabei wird die Farbinformation mit ihrem lokalen Bezug im Modell verwendet und nicht als globales Merkmal, wie bei histogrammbasierten Systemen. Global beschreibende Form- und Farbmerkmale, wie etwa Farbhistogramme und geometrische Momente, sind in [Jai96] verwendet worden, um ein kombiniertes Farb- und Formretrieval zu ermöglichen. Dies läßt jedoch keine detaillierte Übereinstimmungsbestimmung zwischen Anfragebild (oder -skizze) und Datenbankelement zu. Eine solche detaillierte Übereinstimmung ist jedoch erforderlich, falls ein Abfragesystem Skizzen bearbeiten soll, die etwa der folgenden textuellen Anfrage entsprechen: *Zeige alle Datenbankelemente, die eine Zange enthalten, die blaue Griffe und einen grauen Kopf hat.*

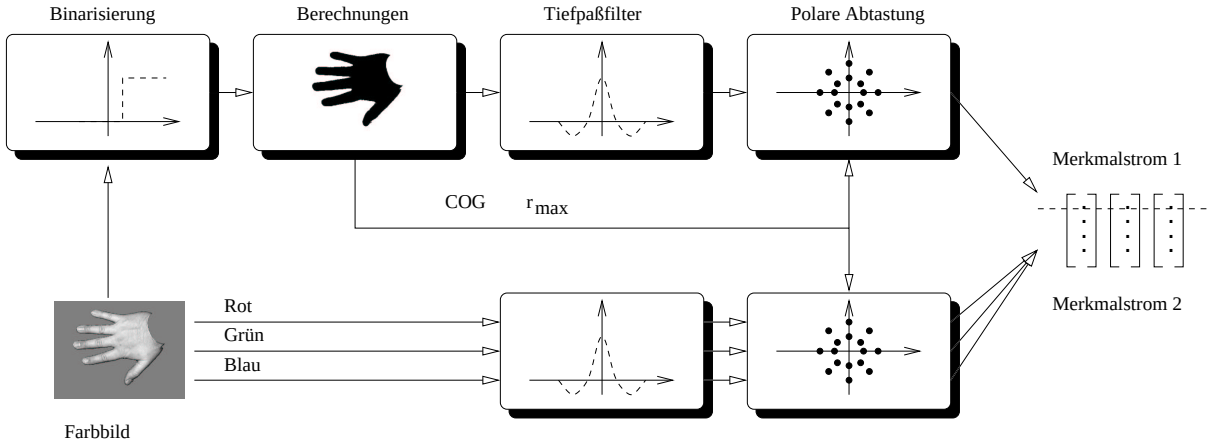


Abbildung 3.12: Vorverarbeitung und Erzeugung der Merkmalsequenz mit zwei Merkmalströmen

Wie schon in Abbildung 3.9 illustriert wurde, soll auch bei der kombinierten form- und farbbasierten Bilddatenbankabfrage jedes Datenbankelement durch ein Hidden-Markov-Modell repräsentiert werden. Die Parameter des Modells werden vorher mit einer Merkmalsequenz trainiert. Es ist daher zunächst eine solche Sequenz für jedes Bild der Datenbank zu berechnen. Die Abbildung 3.12 illustriert die Erzeugung der Form- und Farbmerkmale. Die Komponenten des Merkmalvektors, die die Form des Objekts charakterisieren, werden entsprechend den Bearbeitungsschritten im oberen Pfad in Abbildung 3.12 berechnet. Diese Schritte entsprechen denen, die in Unterkapitel 3.5.2 für die ausschließlich formbasierte Bilddatenbankabfrage vorgestellt wurden. Um die Farbmerkmale zu berechnen, wird das Bild im RGB-Farbraum betrachtet und der Binarisierungsschritt weggelassen. Es werden für jeden der drei Farbkanäle die in der Abb. 3.12 illustrierten und schon bekannten Verarbeitungsschritte, nämlich Tiefpaßfilterung und polare Unterabtastung durchgeführt. Die für die polare Abtastung erforderlichen Parameter, nämlich der Flächenschwerpunkt $\vec{S}$ und der Radius $r_{max}$, werden auf dem binarisierten Bild berechnet und sind somit für alle drei Farbkanäle identisch. In einem letzten Schritt werden die Abtastwerte, wie in Unterkapitel 3.5.2 beschrieben, zu einer Merkmalsequenz zusammengefaßt. Zu beachten ist hierbei, daß die Merkmalvektoren nun sowohl Informationen über die Form als auch über die Farbe des Objekts enthalten und ihre Dimension gegenüber den ausschließlich formbeschreibenden Vektoren vervierfacht ist. Obwohl es möglich wäre, die Merkmalvektoren ohne eine weitere Veränderung direkt zu verwenden, werden diese in zwei sogenannte *Merkmalströme* unterteilt. Merkmalströme sind Komponenten des Merkmalvektors, die als statistisch unabhängig angesehen werden. Mit Hilfe dieser Merkmalströme ist es möglich, verschiedene visuelle Merkmale, wie z.B. Farbe, Form oder Textur in einem einzelnen Modell zu integrieren und dabei die Merkmalstrom-Gewichte $\gamma_s$ zu verwenden, um den Einfluß der verschiedenen visuellen Merkmale zu verändern (siehe auch Kapitel 2). Der Benutzer eines solchen Systems wird also in die Lage versetzt, mittels der Gewichte $\gamma_{s1}$ und $\gamma_{s2}$ die Bedeutung der Farbe in

seiner Anfrageskizze zu erhöhen oder zu reduzieren. Es ist auch die Verwendung von mehr als zwei Merkmalströme denkbar. Beispielsweise kann der Farb-Merkmalstrom weiter unterteilt werden in eine Chrominanz und eine Luminanzkomponente im YIQ-Farbraum. Durch diese zusätzliche Unterteilung könnte ein Benutzer den Einfluß von Bildhelligkeit gegenüber der Farbigkeit steuern.

Mit den in zwei Merkmalströmen unterteilten Merkmalsequenzen werden wiederum die modifizierten und die Objekte rotationsinvariant beschreibenden Hidden-Markov-Modelle trainiert. Anschließend kann dem System eine farbige Skizze zusammen mit den Merkmalstrom-Gewichten $\gamma_{s1}$ und $\gamma_{s2}$ präsentiert werden. Das System gibt daraufhin eine Liste mit den ähnlichsten Objekten aus. Die hier vorgestellten Methoden wurden mittels der schon in Unterkapitel 3.5.2 verwendeten Werkzeugdatenbank evaluiert. Die Bilder der Werkzeuge sind im RGB-Farbraum mit jeweils 8 Bits pro Farbkanal und Bildpunkt aufgenommen worden. Für die Experimente in Kapitel 3.5.2 wurden die Bilder in Grauwertbilder umgewandelt, während die Farbinformation in den im folgenden beschriebenen Experimenten mitverwendet werden. Die Gesamtgröße der Markov-Modelle beträgt wiederum 90 Zustände und die Merkmalextraktion wurde ebenfalls mit fünf Abtastwerten bei $\Delta\varphi = 10^\circ$ für jeden Farbkanal durchgeführt. Die resultierende Vektorgröße beträgt also 20 (5 Komponenten für den Farb-Merkmalstrom und 15 Komponenten für den Form-Merkmalstrom).

### 3.5.4 Qualitative Ergebnisse

Die Abbildung 3.13 zeigt experimentelle Ergebnisse, die mit den in diesem Kapitel beschriebenen Methoden erzielt wurden (siehe auch [Rig00] für weitere Ergebnisse). In jeder Zeile wird zunächst die Anfrageskizze gezeigt (hellgrauer Hintergrund) und anschließend die vier Datenbankbilder mit dem größten Ähnlichkeitsmaß (dunkelgrauer Hintergrund). Die Anzahl der den einzelnen Teilmodellen zugeordneten Merkmalvektoren ($f_1$ bzw. $f_2$ in Gleichung 3.11) ist unter den Datenbankelementen angegeben. Aus diesen lassen sich entsprechend Gleichung 3.11 geschätzte Rotationswinkel berechnen, die ebenfalls angegeben sind. Mit Ausnahme der untersten Zeile in Abb. 3.13 wurden die Ergebnisse mit den Merkmalstrom-Gewichten $\gamma_{s1} = 1.0$ und $\gamma_{s2} = 0.1$ erzielt. Dabei ist $\gamma_{s1}$ das Gewicht der von der Form abgeleiteten Merkmale und $\gamma_{s2}$ das Gewicht der farbbasierten Merkmale. Diese Konfiguration führt zu einem Abfragemodus, bei dem die Form der Anfrageskizze stärker berücksichtigt wird als deren Farbe. Falls also beispielsweise die Skizze eines roten Schraubenziehers eingegeben wird, wird das System rote und andersfarbige Schraubenzieher ausgeben und nicht rote Schraubenzieher und andere rote Objekte. Der Einfluß der Merkmalstrom-Gewichte wird ebenfalls in Abb. 3.13 verdeutlicht. So wurde in der achten Zeile dieselbe Skizze wie in Zeile 7 präsentiert, die Merkmalstrom-Gewichte jedoch auf $\gamma_{s1} = \gamma_{s2} = 1.0$ geändert. Durch die Änderung des Gewichts $\gamma_{s2}$ erscheinen nun mehr blaue bzw. dunkle Objekte unter den ersten fünf Rängen und nicht mehr ausschließlich Zangen, wie in Zeile 7 der

Abbildung. Die Merkmalstrom-Gewichte werden somit zu einem wichtigen Bestandteil der Anfrage.

Wie Abb. 3.13 entnommen werden kann, funktioniert das System gut. Bilder einer Bilddatenbank können durch die intuitive Formulierung einer farbigen Skizze gefunden werden. Bei der Erstellung der Skizze ist es durch die rotationsinvariante Repräsentation der Bildinhalte mit Hilfe der Hidden-Markov-Modelle nicht notwendig, die Orientierung der gesuchten Objekte *a priori* zu kennen. Die berechneten Rotationswinkel in Abb. 3.13 sind gute Schätzwerte für die Drehungen der Objekte, bezogen auf die Orientierung der Anfrageskizzen. Obwohl diese angegebenen Winkel in dem hier dargestellten Anwendungsszenario keine große Bedeutung haben, bieten diese Winkel dennoch eine weitere wichtige Evaluierungsmöglichkeit für die rotationsinvarinate, statistische Modellierung. Bei Betrachtung der aneinandergefügten Hidden-Markov-Modelle in Abb. 3.5 und des Zuordnungsschemas von Merkmalen zu den Modellen in Abb. 3.3 kann eingewendet werden, daß es nicht garantiert ist, daß die Merkmale, die ein Objekt vollständig und unrotiert beschreiben, auch dem entsprechenden Modell zugeordnet werden. Die Anzahl der den Teilmodellen zugeordneten Merkmale ist, wie in Abb. 3.13 zu sehen ist, jedoch bei nahezu allen Anfragen gleich 36 (dies Entspricht einer Sequenz von $360^\circ$ bei $\Delta\varphi = 10^\circ$). Diese Beobachtung belegt, daß die Merkmale, die ein Objekt vollständig beschreiben, in den meisten Fällen auch dem entsprechenden Modell zugeordnet werden. In Abb. 3.13 ist auch ein vom System gemachter Fehler zu sehen. Das höhere Ähnlichkeitsmaß für den Schraubenschlüssel in Zeile 5, Spalte 4 im Vergleich zu der Gabel in Zeile 5, Spalte 5 entspricht nicht den Erwartungen bzw. der menschlichen Wahrnehmung.

Neben guten Anfrageergebnissen werden von einem Retrievalsystem auch kurze Reaktionszeiten erwartet. Die Zeit, die für eine Verarbeitung einer Anfrage benötigt wird, beträgt sechs Sekunden bei 120 Datenbankelementen. Dabei wurde ein handelsüblicher 300 MHz Personal Computer verwendet. Um die Geschwindigkeit mit einem Wert aus der Literatur, nämlich 14-15 Minuten bei 200 Bildern der Größe $256\times256$ vergleichen zu können ([Lin97]), wurde die hier verwendete Datenbasis um weitere 80 Bilder erweitert. Diese sind nicht neu aufgenommen worden, sondern entstanden durch Spiegelung von vorhandenen Datenbankelementen. Das System benötigt nun 10 Sekunden bei 200 Datenbankelementen. Diese Reaktionszeit kann weiter verkürzt werden durch die Verknüpfung von Modellparametern (engl. Tying). Dies ist eine in der automatischen Spracherkennung weitverbreitete Technik ([You92]). Es ist dabei vorteilhaft, die Parameter derjenigen Zustände in Abb. 3.5 zu verknüpfen, die dieselbe Position im Modell haben (gleiche Zustandsnummer in Abb. 3.5). Diese Technik führt zu einer Reduktion der Antwortzeit um 1–2 Sekunden. Der Grund für die Beschleunigung liegt in der Wiederverwendung von bereits berechneten Ausgabewahrscheinlichkeiten während der Durchführung des Viterbi-Algorithmus.

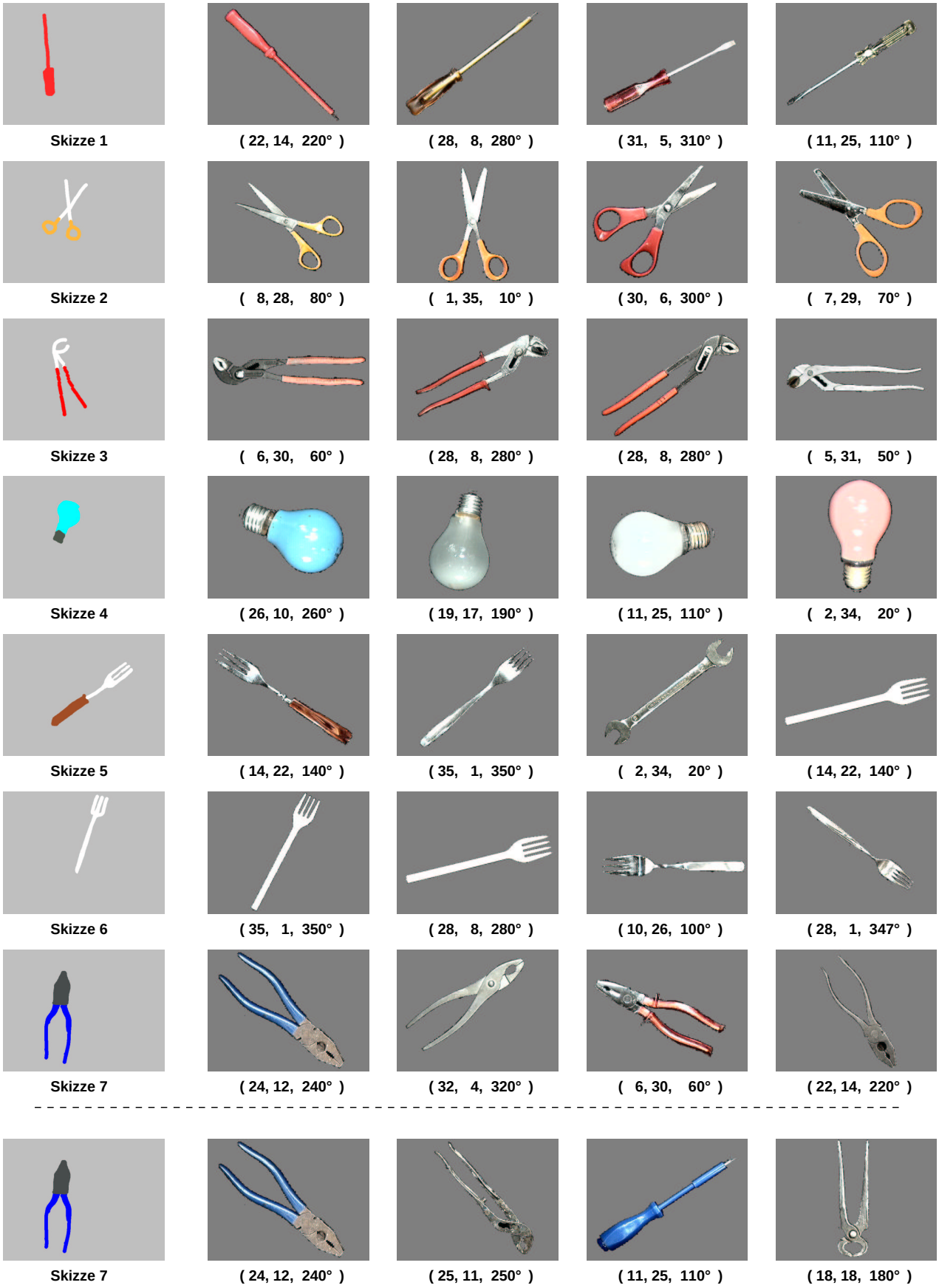


Abbildung 3.13: Anfrageskizzen und die vom Datenbanksystem ermittelten vier ähnlichsten Farbbilder

### 3.5.5 Quantitative Ergebnisse

In den vorangegangenen Unterkapiteln wurden gute qualitative Ergebnisse, die mit den modifizierten Hidden-Markov-Modellen erzielt wurden, präsentiert. Die Verwendung von Anfrageskizzen erschwert es erheblich, die in der Mustererkennung üblichen quantitativen Ergebnisse zu präsentieren. Der Grund hierfür ist, daß die Qualität der Anfrageskizze einen Einfluß auf das Abfrageergebnis hat. Ist beispielsweise eine Skizze derart mislungen, daß ein menschlicher Betrachter das Objekt nicht mehr eindeutig bestimmen kann, so sind auch keine guten Ergebnisse mehr von den vorgestellten Algorithmen zu erwarten. Ferner macht es die Abhängigkeit von der Person, die die Skizze anfertigt, sehr schwer für andere Forschergruppen, quantitative Ergebnisse, die mit Skizzen erzielt wurden zu verifizieren. Aus diesen Gründen werden im folgenden quantitative Ergebnisse präsentiert, die mit Beispielbildern erzielt wurden. Um die Aufgabe ähnlich anspruchsvoll zu definieren, wie bei einer Verwendung von Anfrageskizzen, werden stark deformierte Objekte als Datenbankelemente verwendet. Dieses Szenario ähnelt dem der Anfrage mit Skizzen und ist zudem ein wichtiger Schritt auf dem Weg zur Lösung des Problems der Bilddatenbankabfrage von sich berührenden oder überlappenden Objekten. Dieses Problem soll anhand von Abbildung 3.14 analysiert werden. In der Abbildung ist zu sehen, daß durch eine Segmentierung in Einzelobjekte

Abbildung 3.14: Sich teilweise überlappende Objekte

zusätzliche Deformationen unvermeidbar sind. So ist beispielsweise die Fassung der Glühbirne in Abbildung 3.14 von dem Schraubenschlüssel verdeckt und die genaue Form der Fassung somit nicht rekonstruierbar. Die quantitativen Untersuchungen ermöglichen auch eine detaillierte Evaluierung des Einflusses der von Taza und Suen [Taz89] vorgestellten und in Unterkapitel 3.2 beschriebenen Gewichtungsfaktoren bei der Berechnung von Form-Matrizen. Die Einführung dieser Gewichtungsfaktoren war in der abnehmenden Abtastdichte mit zunehmendem Radius begründet. Wie in Gleichung 3.9 in Unterkapitel 3.2 gezeigt wurde, sind die Gewichte direkt proportional zum Radius zu wählen. Diese Gewichte können auf effiziente Weise in die Markov-Modelle integriert werden, indem die Elemente des

Merkmalvektors in Merkmalströme unterteilt werden und die Merkmalstrom-Gewichte $\gamma_s$ entsprechend dem in [Taz89] vorgestellten Schema gewählt werden.

Die Experimente sind auf einer Datenbasis mit künstlich deformierten Objekten durchgeführt worden. Die Abb. 3.15 zeigt drei Beispiele von künstlich deformierten Scheren, die einen Teil der verwendeten Datenbank darstellen. Die Art der Deformationen sollen die Effekte von Teilüberdeckungen mit anderen Objekten wiederspiegeln. Die erste in Abbil-

Anfragebild

Abbildung 3.15: Künstlich deformierte Formen

dung 3.15 gezeigte Form ist nicht deformiert und wird während der Experimente als Anfragebild verwendet. Die Datenbasis besteht aus 12 Klassen von Werkzeugen. Jede Klasse enthält vier deformierte Objekte und ein undeformiertes Anfrageobjekt. Während der Experimente werden die 12 Anfrageobjekte präsentiert und die sechs Bilder mit dem höchsten Übereinstimmungsmaß weiter analysiert. Tabelle 3.3 zeigt die Retrieval-Effizienz $\eta_T$. Wie in [Meh97] angegeben ist, ist die Retrieval-Effizienz für eine definierte Liste der Größe $T$ folgendermaßen definiert:

$$\eta_T = \begin{cases} \frac{n}{N} & \text{für} \quad N \leq T \\ \frac{n}{T} & \text{für} \quad N > T \end{cases} \tag{3.17}$$

Dabei ist $n$ die Anzahl an richtig zurückgelieferten Bildern und $N$ ist die Anzahl an richtigen Bildern in der Datenbank. In den durchgeführten Experimenten ist die maximale Anzahl an richtigen Bildern $N = 4$ und die Größe der Liste beträgt $T = 6$. Die durchschnittliche Retrieval-Effizienz bezogen auf zwölf Experimente (jeweils ein Experiment für jede Objektklasse) ist in Tabelle 3.3 angegeben. In der ersten Zeile der Tabelle ist das Ergebnis für

| Verwendete Methode | Retrieval-Effizienz $\eta_T$ |
|---|---|
| HMM-Ansatz | 75,00% |
| HMM-Ansatz mit Merkmalstrom-Gewichten | 83,33% |
| Pseudo zweidimensionale HMMs | 79,17% |

Tabelle 3.3: In den Experimenten erzielte Retrieval-Effizienz

den im Kapitel 3.5.2 vorgestellten Ansatz angegeben. Die zweite Zeile enthält das durch die Merkmalstrom-Gewichte verbesserte Ergebnis. Ebenfalls in der Tabelle angegeben ist die Retrieval-Effizienz für pseudo zweidimensionale Markov-Modelle. Diese Modelle werden im folgenden Kapitel 4 vorgestellt und somit sei an dieser Stelle auf Unterkapitel 5.2 verwiesen. Unterkapitel 5.2 erklärt detailliert die Topologie und die Merkmalextraktion, die zu

den in Tabelle 3.3 angegebenen Ergebnissen geführt haben. Details zu dieser Methode sind ebenfalls in der Arbeit [Mul99d] zu finden.

### 3.5.6 Skizzenbasierte Datenbankabfrage im Internet

Eine weitere Möglichkeit, das vorgestellte Datenbanksystem zu evaluieren, ist, das System einer großen Anzahl von Benutzern über das World-Wide-Web (WWW) zugänglich zu machen und deren Erfahrungen mit dem System auszuwerten. Aus diesem Grunde wurde eine auf der Programmiersprache Java basierende Version entwickelt und unter `http://www.fb9-ti.uni-duisburg.de/demos/query.html` zugänglich gemacht. Abb. 3.16 zeigt das Java-Applet, welches für die Eingabe von Anfrageskizzen ent-


Abbildung 3.16: Java-Applet für die Eingabe von Anfrageskizzen

wickelt wurde. Nachdem eine Anfrageskizze mit dem Applet und einer Computermouse erstellt wurde, werden Daten, die die Skizze repräsentieren, über das WWW zu einem Server übermittelt, auf dem sich das Bilddatenbanksystem befindet. Der Server berechnet die vier ähnlichsten Bilder und übermittelt diese als Abfrageergebnis zurück and den Client, der den Anfrageprozeß eingeleitet hat. Bei der Benutzung des Systems kann eine Anfrage sukzessive verfeinert werden, bis die gewünschten Bilder in der Datenbank gefunden wurden. Dabei können Zwischenergebnisse gespeichert werden und somit eine Historie der Anfragen erstellt und ausgewertet werden.

## 3.6 Kapitelzusammenfassung

In diesem Kapitel wurden neuartige Hidden-Markov-Modell Topologien vorgestellt, die zusammen mit den Form-Matrizen eine translations-, skalierungs- und rotationsunabhängige Modellierung von Objektformen bzw. handskizzierten Piktogrammen ermöglicht. Da-

bei wurden die integrierten Segmentierungs- und Klassifizierungseigenschaften der Hidden-Markov-Modelle dazu genutzt, die Orientierung der gedrehten Objekte herauszufinden und das Objekt zu erkennen. Drei Varianten des HMM-Klassifizierers wurden vorgestellt, die neben der vollständig rotationsunabhängigen Erkennung auch eine Erkennung mit eingeschränkten Winkelbereichen ermöglichen. In den Experimenten wurden Erkennungsgenauigkeiten von bis zu 99,5% mit Piktogramm-Datenbasen erreicht, die aus 20 Klassen bestehen. Die Erkennungsergebnisse lagen über denen, die mit konventionellen Erkennungsmethoden, nämlich Momentenmethoden in Kombination mit künstlichen neuronalen Netzen erzielt wurden. Die vorgestellten Methoden konnten erfolgreich auf die Erkennung natürlicher Bilder erweitert werden. Es wurden Ergebnisse präsentiert, die mit einem experimentellen Bilddatenbanksystem, das intuitiv über Skizzen des Benutzers abgefragt werden kann und das die in diesem Kapitel vorgestellten Methoden verwendet, erzielt wurden. In dieses Basissystem wurden zusätzlich Farbmerkmale integriert. Dadurch können die Elemente der Bilddatenbank über farbige Skizzen abgefragt werden. Qualitative und quantitative Ergebnisse wurden angegeben und ein internetbasierter Demonstrator entwickelt.

Die Erkenntnisse, die durch die Experimente mit den eindimensionalen Markov-Modellen gewonnen wurden, konnten genutzt werden, um die kombinierten Segmentierungs- und Klassifizierungseigenschaften der HMMs auch im zweidimensionalen Fall nutzen zu können. Kapitel 5 stellt eine Technik vor, um zweidimensionale Muster in komplexen Umgebungen HMM-basiert aufzufinden und zu klassifizieren. Bevor dieses Verfahren vorgestellt werden kann, wird im nächsten Kapitel zunächst in die Theorie der zweidimensionalen statistischen Modellierung eingeführt.

# Kapitel 4

# Statistische Modellierung in zwei Dimensionen

Die in diesem Kapitel vorgestellten statistischen Modellierungsverfahren sind aufgrund ihrer zweidimensionalen Struktur ideal geeignet, um Bilder zu modellieren. Das theoretisch geeignetste Modell, welches Gegenstand des folgenden Unterkapitels ist, basiert auf den sog. Markov-Random-Fields (MRFs), die eine mehrdimensionale Erweiterung der Markov-Quellen darstellen.

## 4.1 Markov-Random-Fields

In Kapitel 2 wurden die Hidden-Markov-Modelle als eine Erweiterung der Markov-Quellen vorgestellt, die einen *einfachen* statistischen Prozeß darstellen. Der Übergang zwischen den Modellzuständen ist bei der Markov-Quelle erster Ordnung durch die Übergangsmatrix $A$ gegeben, deren Elemente durch $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$ darstellbar sind. Diese Definition des Zustandsübergangs charakterisiert die Markov-Quelle als einen eindimensionalen Prozeß, der zudem kausal ist. Es handelt sich mithin um ein Modell, das sehr gut geeignet ist, um zeitabhängige Prozesse zu beschreiben.

Kartesisch abgetastete Bilder sind von zweidimensionaler Art und ihre Elemente (Pixel) verfügen nicht über eine natürliche kausale Ordnung. Aus diesen Gründen besitzt das den Markov-Modellen entsprechende statistische Modell für zweidimensionale, nichtkausale Prozesse eine erheblich komplexere Struktur. Das in der wissenschaftlichen Literatur gebräuchlichste statistische Modell für Bilder ist das zweidimensionale Wahrscheinlichkeitsfeld mit lokalen Abhängigkeiten, das als *Markov-Random-Field* (MRF) bekannt ist (siehe z.B. [Gem85, Der89] und [Li95]). Die Theorie der MRFs wurde ursprünglich in der statistischen Physik für interagierende Partikel wie z.B. Moleküle oder atomare Magnete entwickelt. Mit Hilfe der Markov-Random-Fields konnten Vorgänge beim Ferromagnetismus (Ising-Modell), in idealen Gasen, sowie in zweiwertigen Metallegierungen modelliert und

erklärt werden. Später wurden die MRFs in der Bildverarbeitung, beispielsweise für die Restaurierung von verrauschten oder optisch verzerrten Bildern ([Gem85]), der Segmentierung von Bildregionen mit homogener Textur ([Sim88]) oder der Oberflächenrekonstruktion aus Abstandsdaten eingesetzt.

Im folgenden wird angenommen, daß das MRF bei der Modellierung von Bildern eingesetzt wird und die Beispiele werden entsprechend gewählt. Bei Markov-Random-Fields werden Aufgabenstellungen der Bildverarbeitung umformuliert und als die Aufgabe dargestellt, Bildpunkten *Label* zuzuordnen. Solche Label können beispielsweise Indizes sein, die Texturklassen oder farblich homogenen Bildregionen entsprechen. Formal wird die Label-Aufgabe mit Hilfe von *Sites* und einer Label-Menge definiert. Die Menge der $N$ Sites sei gegeben durch

$$S = \{s_1, s_2, \ldots, s_N\} \tag{4.1}$$

Sites repräsentieren Punkte oder Regionen im Bildraum, wie z.B. Pixel oder Bildmerkmale. Für den Fall, daß die Sites Pixel eines $m \times m$ dimensionalen Bildes repräsentieren, entspricht die folgende Indexierung eher den üblichen Notationen in der Bildverarbeitung:

$$S = \{s_{11}, s_{12}, \ldots, s_{mm}\} \tag{4.2}$$

Es geht die in der Literatur zu MRFs gebräuchlichere Gleichung 4.1 aus Gleichung 4.2 durch Umindizierung hervor. Die Sites $S$ interagieren miteinander durch das *Nachbarschaftssystem*. Ein Nachbarschaftssystem ist folgendermaßen gegeben:

$$\mathscr{N} = \{\mathscr{N}_i, s_i \in S\} \tag{4.3}$$

Die Sites $\mathscr{N}_i$ sind die Nachbarn der Site $s_i \in S$. Von besonderem Interesse für die Bildverarbeitung sind homogene Nachbarschaftssysteme auf rechteckigem Raster der Form

$$\mathscr{N}_{ij} = \{s'_{ij} \in S, s_{kl} \in S, 0 < (k-i)^2 + (l-j)^2 \leq r\} \tag{4.4}$$

Es ist dabei zu beachten, daß an den Randpunkten der Bilder noch entsprechende Korrekturen vorzunehmen sind. Für $r = 1$ liegt eine sog. Nachbarschaftsbeziehung erster Ordnung vor, bei der die Nachbarn der Site $s_{ij}$ durch die Menge

$$\mathscr{N}_{ij} = \{s_{i,j-1}, s_{i,j+1}, s_{i-1,j}, s_{i+1,j}\} \tag{4.5}$$

gegeben ist. Den Sites können diskrete Label aus der Menge

$$\mathscr{L} = \{l_1, \ldots, l_M\} \tag{4.6}$$

zugeordnet werden.

Unter Verwendung der bisher definierten Größen kann nun das MRF formal definiert werden. Seien $F = \{F_1, \ldots, F_N\}$ Zufallsvariablen über einer Menge von Sites $S = \{s_1, \ldots, s_N\}$

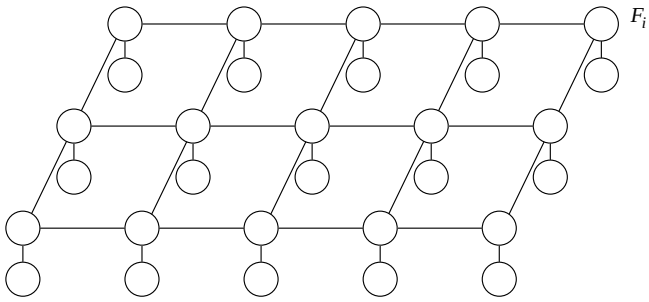


Abbildung 4.1: Darstellung eines Markov-Random-Fields mit Nachbarschaftsbeziehung erster Ordnung als ungerichteten Graph (aus [Luc95])

unter Verwendung eines Nachbarschaftssystems $\mathscr{N}$, so ist der statistische Prozeß $F$ ein Markov-Random-Field, falls

$$P(F = f) > 0 \quad \text{und}$$
$$P(F_i = f_i | F_k = f_k, k \neq i) = P(F_i = f_i | F_k = f_k, s_k \in \mathscr{N}_i) \tag{4.7}$$

In der obigen Gleichung ist mit $f_i$ eine Realisierung der Zufallsvariablen $F_i$ bezeichnet. $f_i$ nimmt Werte aus der Menge der Label $\mathscr{L}$ an. Gleichung 4.7 stellt eine mehrdimensionale und nicht notwendigerweise kausale Verallgemeinerung der in Gleichung 2.1 für Markov-Quellen vorgestellten eindimensionalen Markovbedingung dar. Das einer Site zugeordnete Label hängt ausschließlich von den Labels der benachbarten Sites ab. Diese Abhängigkeit kann für den Fall einer Nachbarschaftsbeziehung erster Ordnung (Gleichung 4.5) in Form eines graphischen Modells visualisiert werden (siehe auch Abb. 2.4). Abb. 4.1 zeigt die zweidimensionale Anordnung der Sites und die den Sites zugeordneten Labels als ungerichteten Graphen. Wie schon in Unterkapitel 2.2.6 bzw. in Abb. 2.4 deutet die Abwesenheit von Verbindungen in Abb. 4.1 auf statistische Unabhängigkeit hin. An dieser Stelle sei angemerkt, daß sowohl für Markov-Quellen bzw. Hidden-Markov-Modelle als auch für MRFs graphische Modelle existieren (vgl. Abb. 2.4 und Abb. 4.1), eine Darstellung als endlicher stochastischer Automat ist jedoch nur für die erstgenannten, eindimensionalen Modelle möglich (siehe Abb. 2.1). Der Grund hierfür liegt in der fehlenden Kausalität der Markov-Random-Fields. Die Abb. 4.1 und die Markovbedingung in Gleichung 4.7 beschreiben die *lokale* Charakteristik der MRFs. Neben der Möglichkeit, ein MRF über die lokalen Eigenschaften und mithin über die bedingten Wahrscheinlichkeiten $P(F_i = f_i | F_k = f_k, s_k \in \mathscr{N}_i)$ zu definieren, existiert auch die Möglichkeit die globalen Eigenschaften zur Definition zu verwenden. Das Hammersly-Clifford Theorem besagt, daß jedes Wahrscheinlichkeitsfeld $F$ auf $S$ unter Verwendung des Nachbarschaftssystems $\mathscr{N}$ genau dann ein MRF ist, falls $P(F = f)$ in Form einer Gibbs'schen Verteilung darstellbar ist (Beweis siehe [Li95]). Die Gibbs'sche

Verteilung ist gegeben durch:

$$P(F = f) = \frac{1}{Z} e^{-\frac{U(f)}{T}} \quad \text{mit}$$
$$Z = \sum_{\text{alle } f} e^{-\frac{U(f)}{T}} \tag{4.8}$$

In Gleichung 4.8 ist $Z$ die sog. *Partition Function*, $T$ eine Konstante, die auch Temperatur genannt wird und $U(f)$ die Energiefunktion. Die Energie kann berechnet werden durch

$$U(f) = \sum_{c \in C} V_c(f) \tag{4.9}$$

$U(f)$ ist die Summe über alle sog. *Clique-Potentiale* $V_c(f)$. Cliques sind Untermengen der Sites ($C \subseteq S$) und sie sind durch die Eigenschaft charakterisiert, daß jedes Site-Paar in einem Clique ein Nachbarpaar ist. Dies soll am Beispiel der Nachbarschaftbeziehung erster Ordnung (siehe Gleichung 4.5) verdeutlicht werden. Die Cliques für die Site $s_{ij}$ und dem letztgenannten Nachbarschaftssystem bilden die folgende Menge von Sites:

$$C = \{\{s_{i,j}\}, \{s_{i,j}, s_{i,j+1}\}, \{s_{i,j}, s_{i,j-1}\}, \{s_{i,j}, s_{i+1,j}\}, \{s_{i,j}, s_{i-1,j}\}\} \tag{4.10}$$

Die Clique-Potentiale sind ein Bestandteil des festgelegten Modells und charakterisieren die lokalen Interaktionen der Sites. In [Li95] ist aus Gleichung 4.8 eine Gleichung für die Wahrscheinlichkeit $P(F_i = f_i | F_k = f_k, s_k \in \mathcal{N}_i)$, also die Wahrscheinlichkeit, daß die Site $s_i$ den *Zustand* $f_i$ einnimmt, unter der Annahme, daß die Nachbarsites festgelegt sind, abgeleitet worden. Diese Gleichung ist:

$$P(F_i = f_i | F_k = f_k, s_k \in \mathcal{N}_i) = \frac{e^{-\sum_{c \in C} V_c(f)}}{\sum_{f_k; s_k \in \mathcal{N}_i; k \neq i} e^{-\sum_{c \in C} V_c(f')}} \tag{4.11}$$

Diese bedingte Wahrscheinlichkeit entspricht einer *nichtkausalen* Verallgemeinerung der Übergangswahrscheinlichkeit $a_{ij}$ bei der Markov-Quelle bzw. dem Markov-Modell (vgl. die Gleichungen 2.2 und 2.3).

Zusammenfassend kann festgestellt werden, daß die Markov-Random-Fields eine nichtkausale, multidimensionale Erweiterung der Markov-Quellen sind. In Gleichung 4.11 ist eine den Übergangswahrscheinlichkeiten der Markov-Quellen entsprechende Größe definiert worden und somit ist ein dem *ersten* stochastischen Prozeß der Hidden-Markov-Modelle entsprechender Prozeß vorhanden. Auf ähnliche Weise, wie bei den HMMs kann nun auch bei den MRFs ein zweiter stochastischer Prozeß eingeführt werden. Dies geschieht auf folgende Weise: Die Label entsprechen nicht mehr direkt z.B. einem Grauwert oder allgemeiner formuliert, einem Merkmal in einem Bild, sondern stellen einen Index einer Wahrscheinlichkeitsverteilung bzw. -dichte dar. Die Ausgabe von Grauwerten erfolgt dann wie bei den Hidden-Markov-Modellen über diesen zweiten stochastischen Prozeß. Ein solches *Hidden*

Markov-Random-Field stellt ein sehr geeignetes Modell für Bilder dar, da der nichtkausale Aufbau der Bilder modelliert werden kann und durch die Ausgabe von Merkmalen über Wahrscheinlichkeitsverteilungen zusätzliche Variationen berücksichtigt werden. Für die praktische Verwendung in der Mustererkennung erweist sich jedoch das Fehlen von effizienten Algorithmen zur Parameterbestimmung bzw. zur Bestimmung der Produktionswahrscheinlichkeiten als Hindernis. Durch die fehlende Kausalität sind rekursive Verfahren, wie Baum-Welch- und Viterbi-Algorithmus nicht anwendbar ([Luc95]) und es müssen sehr rechenaufwendige Verfahren für die Bestimmung der Produktionswahrscheinlichkeit eingesetzt werden, wie z.B. *Stochastic Relaxation* ([Gem85]). Dieses Fehlen effizienter Algorithmen hat dazu geführt, daß die Modellierung mit MRFs im Bereich der zwei- und mehrdimensionalen Muster nicht dieselbe, dominierende Verbreitung gefunden hat, wie die Hidden-Markov-Modelle in der Modellierung eindimensionaler Muster und insbesondere von Sprachmustern. Da für die nichtkausalen MRFs keine effizienten Algorithmen existieren, wurde in der wissenschaftlichen Literatur eine Vielzahl von kausalen Variationen untersucht. Ein wichtiger Beitrag im Kontext dieser Arbeit ist die Betrachtung von zweidimensionalen Hidden-Markov-Modellen, die von Levin und Pieraccini in der Arbeit [Lev92] vorgestellt wurden.

## 4.2 Zweidimensionale Hidden-Markov-Modelle

Zweidimensionale Hidden-Markov-Modelle (2DHMMs), die auch planare HMMs genannt werden, sind kausale MRFs mit einem zusätzlichen zweiten stochastischen Prozeß, der durch die Ausgabe von Observationen über Ausgabeverteilungen bzw. Ausgabedichten gegeben ist. Die Observationen $o_{xy}$, beispielsweise Grauwerte eines Bildes, werden als zweidimensionale Matrix, die in den Bilddimensionen $x$ mit $(1 \leq x \leq X)$ und $y$ mit $(1 \leq y \leq Y)$ indiziert sind, betrachtet. Statistische Abhängigkeiten bestehen beim 2DHMM zu den beiden *Vorgängern* in beiden Dimensionen. Der Zustand $q_{(x,y)}$ am Ort $(x,y)$ hängt ausschließlich von den Zuständen $q_{(x-1,y)}$ und $q_{(x,y-1)}$ ab. Dies ist eine kausale Markovbedingung erster Ordnung für zwei Dimensionen und kann formal folgendermaßen angegeben werden:

$$\begin{aligned} P(q_{(x,y)} = S_{(i,j)} | q_{(\tilde{x},\tilde{y})} = S_{(\tilde{k},\tilde{l})}; \begin{matrix} 1 \leq \tilde{x} \leq (x-1) \\ 1 \leq \tilde{y} \leq (y-1) \end{matrix}; \begin{matrix} 1 \leq \tilde{k} \leq N_x \\ 1 \leq \tilde{l} \leq N_y \end{matrix}) = \\ P(q_{(x,y)} = S_{(i,j)} | q_{(x-1,y)} = S_{(k,l)}, q_{(x,y-1)} = S_{(m,n)}) \end{aligned} \tag{4.12}$$

In der obigen Gleichung ist mit $S_{(i,j)}$ der Zustand mit Index $(i,j)$ auf einem kartesischen Raster von Zuständen bezeichnet, während $q_{(x,y)}$ die Zufallsvariable für die Einnahme eines Zustandes aus der Menge der möglichen Zustände $(S_{(1,1)}, S_{(1,2)}, \ldots, S_{(2,1)}, \ldots, S_{(N_x,N_y)})$ am Ort $(x,y)$ darstellt (vgl. Gleichung 2.1 für den eindimensionalen Fall). Wie schon bei den eindimensionalen HMMs, gehört zu jedem Modellzustand $S_{(i,j)}$ eine z.B. diskrete Ausgabeverteilung $b_{(i,j)}(k)$ über einem festgelegten Alphabet. Aus Gleichung 4.12 lassen sich Übergangs-

wahrscheinlichkeiten zwischen Modellzuständen der folgenden Form ableiten (aus [Lev92]; Siehe auch [Mer00]):

$$A_{ij,kl,mn} = P(q_{(x,y)} = S_{(m,n)} | q_{(x-1,y)} = S_{(i,j)}, q_{(x,y-1)} = S_{(k,l)}) \tag{4.13}$$

Zusätzlich sind zwei Ränder des Bildes ($x = 1$ bzw. $y = 1$) als Sonderfälle zu betrachten. Hier können die folgenden Übergangswahrscheinlichkeiten verwendet werden, die den eindimensionalen Übergangswahrscheinlichkeiten über den diskreten Werten $x$ und $y$ entsprechen (vgl. Gleichung 2.2):

$$a^V_{ij,kl} = P(q_{(x=1,y)} = S_{(k,l)} | q_{(x=1,y-1)} = S_{(i,j)}) \quad \text{für} \quad (x = 1, y) \tag{4.14}$$

$$a^H_{ij,kl} = P(q_{(x,y=1)} = S_{(k,l)} | q_{(x-1,y=1)} = S_{(i,j)}) \quad \text{für} \quad (x, y = 1) \tag{4.15}$$

Zusätzlich zu den Übergangswahrscheinlichkeiten und den Ausgabewahrscheinlichkeiten fehlt für die vollständige Definition eines Modells noch die Angabe der Wahrscheinlichkeiten für die Einnahme des Anfangszustands, die gegeben sind durch:

$$\pi_{ij} = P(q_{(1,1)} = S_{(i,j)}) \tag{4.16}$$

Nach einer vollständigen Definition eines Modells $\lambda$ kann die Produktionswahrscheinlichkeit für eine gegebene Symbolmatrix $O_{XY}$ berechnet werden. Das Finden einer effizienten Methode zur Berechnung der Produktionswahrscheinlichkeiten war in Kapitel 2 als eines der zu lösenden Aufgaben definiert worden, um die Markov-Modelle zur Musterklassifikation verwenden zu können. Der direkte Weg, um die gesuchte Wahrscheinlichkeit bei gegebener Matrix $O_{XY}$ zu berechnen, führt zunächst über die Wahrscheinlichkeit $P(Q|\lambda)$, der Wahrscheinlichkeit, daß eine Zustandsfolge $Q$ durchlaufen wurde. Diese ist in [Lev92] angegeben als:

$$P(Q|\lambda) = \pi_{q_{(1,1)}} \prod_{x=2}^{X} a^H_{q_{(x-1,1)},q_{(x,1)}} \prod_{y=2}^{Y} a^V_{q_{(1,y-1)},q_{(1,y)}} \prod_{y=2}^{Y} \prod_{x=2}^{X} A_{q_{(x-1,y)},q_{(x,y-1)},q_{(x,y)}} \tag{4.17}$$

Die Verbundwahrscheinlichkeit für das gemeinsame Eintreten der Zustandsfolge $Q$ und der Observation $O_{XY}$ ist

$$\begin{aligned} P(O_{XY}, Q|\lambda) &= P(O_{XY}|Q,\lambda) \cdot P(Q|\lambda) = \\ \prod_{x=1}^{X} \prod_{y=1}^{Y} b_{q_{x,y}}(o_{xy}) \pi_{q_{(1,1)}} \prod_{x=2}^{X} a^H_{q_{(x-1,1)},q_{(x,1)}} \prod_{y=2}^{Y} a^V_{q_{(1,y-1)},q_{(1,y)}} \prod_{y=2}^{Y} \prod_{x=2}^{X} A_{q_{(x-1,y)},q_{(x,y-1)},q_{(x,y)}} \end{aligned} \tag{4.18}$$

Schließlich ergibt sich die gesuchte Produktionswahrscheinlichkeit als die Summation dieser Verbundwahrscheinlichkeit über alle möglichen Zustandsfolgen zu:

$$P(O|\lambda) = \sum_{\text{alle } Q} P(O, Q|\lambda) \tag{4.19}$$

Es ist ebenso wie bei den eindimensionalen Hidden-Markov-Modellen möglich, diese Produktionswahrscheinlichkeit unter Verwendung der wahrscheinlichsten Zustandsfolge $Q^\star$ anzunähern. Die Bestimmung beider Wahrscheinlichkeiten erfordert jedoch $(N_X N_Y)^{XY}$ Berechnungen (siehe [Lev92]). Die Anzahl an erforderlichen Berechnungen wächst also mit der Bildgröße exponentiell an. Diese Komplexität läßt sich nicht auf die gleiche Weise, wie im eindimensionalen Fall reduzieren, da sowohl der Baum-Welch-Algorithmus, als auch der Viterbi-Algorithmus nur bei den einfachen statistischen Abhängigkeiten der eindimensionalen Modelle anwendbar sind ([Lev92, Li99, Mer00]). Die Komplexität der Algorithmen kann erheblich reduziert werden, wenn die nichtlinearen Verzerrungen der Zeilen und Spalten eines Bildes als unabhängig voneinander angesehen werden. Diese Annahme führt zu den im nächsten Unterkapitel vorgestellten sog. pseudo zweidimensionalen Hidden-Markov-Modellen.

## 4.3 Pseudo zweidimensionale Hidden-Markov-Modelle

Pseudo zweidimensionale Hidden-Markov-Modelle (P2DHMMs) wurden von Agazzi und Kuo in ([Aga93b]) vorgestellt. Durch den Verzicht auf eine Modellierung, die Musterverzerrungen in beiden Dimensionen gemeinsam betrachtet, also ein Verzicht auf Zustandsübergänge wie die in Gleichung 4.13 für zweidimensionale Hidden-Markov-Modelle vorgestellten Modellgrößen $A_{ij,kl,mn}$, konnten für das Training und die Erkennung effiziente Algorithmen gefunden werden. Diese effizienten Algorithmen basieren auf dem in [Kuo94] vorgestellten sog. zweifachverschachtelten Viterbi-Algorithmus, der eine Variante des in Unterkapitel 2.2.2 für eindimensionale HMMs beschriebenen Algorithmus darstellt. Durch das Vorhandensein effizienter Trainings- und Erkennungsalgorithmen wurden die P2DHMMs, im Gegensatz zu den bisher in diesem Kapitel vorgestellten statistischen Modellen (MRF bzw. 2DHMM) vielfach für die Erkennung von zweidimensionalen Mustern eingesetzt.

Anwendungsbeispiele für pseudo zweidimensionale Hidden-Markov-Modelle sind die Gesichtserkennung [Sam94b, Eic00], inhaltsbasierte Bilddatenbankabfragen [Lin97], die Erkennung von gedruckter Schrift (OCR) [Aga93a] und von Handschrift [Bip97, Bip00]. In der vorliegenden Arbeit werden ebenfalls die P2DHMMs verwendet und um die Fähigkeit erweitert, in komplexe Umgebungen eingebundene zweidimensionale Muster, auf integrierte Weise, zu Segmentieren und zu Klassifizieren. Dieses integrierte Auffinden und Erkennen wird durch die Modellierung mit zusätzlichen Umgebungszuständen ermöglicht und ist in dem sich anschließenden Kapitel 5 ausführlich beschrieben. Eine weitere Bedeutung kommt den P2DHMMs in Kapitel 6 zu, da sie hier als Baustein einer im Rahmen dieser Arbeit entwickelten pseudo dreidimensionalen Hidden-Markov-Modell-Struktur verwendet werden. Zunächst ist es jedoch erforderlich, die P2DHMMs formal zu definieren.

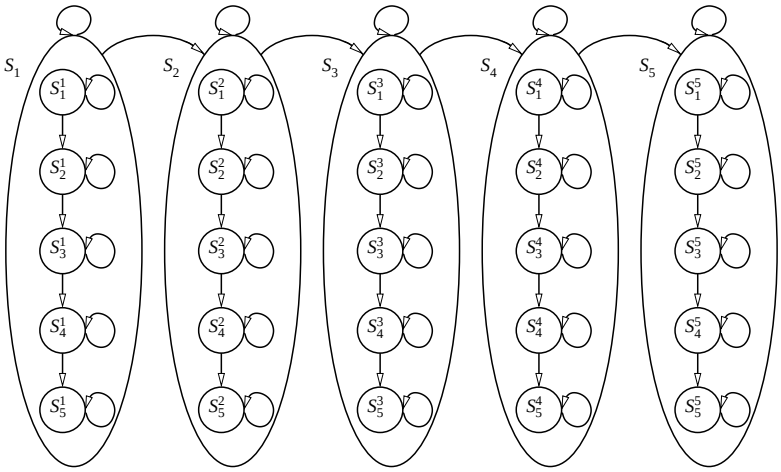


Abbildung 4.2: Pseudo zweidimensionales Hidden-Markov-Modell

### 4.3.1 Modelldefinition der pseudo zweidimensionalen Hidden-Markov-Modelle

Pseudo zweidimensionale HMMs modellieren, im Gegensatz zu den MRFs und den 2DHMMs, die Abhängigkeiten von räumlich benachbarten Merkmalen durch einen zweistufigen, hierarchischen Prozeß. Der übergeordnete statistische Prozeß modelliert die Abhängigkeiten von beispielsweise Bildspalten mit einem Markov-Modell erster Ordnung. Die Bildspalten selbst werden mit weiteren Markov-Modellen modelliert, die in das übergeordnete Modell eingebunden sind. Die Abb. 4.2 zeigt die Darstellung eines P2DHMMs, die das Modell als zweistufigen statistischen Automaten interpretiert. Dargestellt sind fünf, mit $S_1,\ldots,S_5$ bezeichnete, übergeordnete Zustände, die üblicherweise in der englischsprachigen Literatur mit *Superstates* bezeichnet werden ([Aga93b, Aga93a, Kuo94]). In dieser Arbeit wird für die übergeordneten Zustände die in [Bip00] eingeführte Bezeichnung *Metazustand* verwendet. In Abb. 4.2 sind ebenfalls die den Metazuständen zugeordneten eindimensionalen HMMs mit jeweils fünf Zuständen $(S_1^n,\ldots,S_5^n; 1 \leq n \leq 5)$ dargestellt. Liegt nun ein zweidimensionales Muster $O_{XY}$ in Form einer $X \times Y$ Matrix vor, so kann dieses von dem P2DHMM auf folgende Weise modelliert werden: Jede Bildspalte $(o_{xy}, 1 \leq y \leq Y, x = \text{const.})$ wird einem Metazustand zugeordnet, was eine nichtlineare Musterverzerrung in horizontaler Richtung ermöglicht. Zusätzlich werden die Bildspalten selbst von eindimensionalen Hidden-Markov-Modellen modelliert, und dies ermöglicht eine nichtlineare Musterverzerrung in vertikaler Richtung. Auf die gleiche Weise können auch Bildzeilen den Metazuständen zugeordnet werden. Die Unterscheidung in entweder zeilen- oder spaltenbasiert Modellierung verdeutlicht den hierarchischen Aufbau der P2DHMMs.

Formal wird ein pseudo zweidimensionales Hidden-Markov-Modell ($\Lambda$) durch die folgenden Parameter beschrieben. In der höheren hierarchischen Modellebene ist die Anzahl $N$ der Metazustände $(S_1,\ldots,S_N)$ zu spezifizieren. Diesen Zuständen sind Übergangswahr-

scheinlichkeiten $a_{ij}$ zugeordnet, die definiert sind durch:

$$a_{ij} = P(q_x = S_j | q_{x-1} = S_i) \tag{4.20}$$

In Gleichung 4.20 ist mit $q_x$ die Zufallsvariable für die Einnahme eines Metazustands für die Bildspalte $x$ bezeichnet. Es wird an dieser Stelle deutlich, daß bei der Modellfestlegung auch die *Ausrichtung* des P2DHMMs angegeben werden muß. Die Wahrscheinlichkeit für die Anfangszustände der höheren Hierarchieebene ist gegeben durch:

$$\pi_j = P(q_1 = S_j) \tag{4.21}$$

Die Modellgrößen der Gleichungen 4.20 und 4.21 können zu Matrizen bzw. Vektoren zusammengefaßt werden. Jedem Metazustand $S_j$ ist ein eindimensionales HMM zugeordnet. Diese können wie in Kapitel 2 dargestellt, definiert werden. Jeder Modellparameter erhält einen zusätzlichen Index, der die Zugehörigkeit zum entsprechenden Metazustand angibt. Es ergeben sich mithin die folgenden Parameter für die Modelle $\lambda_1, \ldots, \lambda_N$:

- $N^j$ ist die Anzahl der Zustände $(S_1^j, \ldots, S_{N^j}^j)$ des dem $j$-ten Metazustand $S_j$ zugeordneten Modells.

- Die Übergangswahrscheinlichkeiten $a_{kl}^j$ sind folgendermaßen definiert:

  $$a_{kl}^j = P(q_{x,y} = S_l^j | q_{x,y-1} = S_k^j) \tag{4.22}$$

  Dabei ist mit $q_{x,y}$ die Zufallsvariable für die Einnahme eines Zustands für die Bildspalte $x$ und die Bildzeile $y$ bezeichnet.

- Die Wahrscheinlichkeiten für die Anfangszustände sind gegeben durch:

  $$\pi_i^j = P(q_{x,1} = S_i^j) \tag{4.23}$$

- Die Ausgabeverteilungen können wie im eindimensionalen Fall diskret oder kontinuierlich sein. Hier wird ein diskretes Hidden-Markov-Modell angenommen und somit ergibt sich die diskrete Ausgabeverteilung $b_i^j(k)$ über einem für das gesamte P2DHMM festgelegten Alphabet zu:

  $$b_i^j(k) = P(v_k | q_{x,y} = S_i^j) \tag{4.24}$$

  Die in Kapitel 2.2.4 dargestellten Gaußschen Mischverteilungen können alternativ verwendet werden um kontinuierliche P2DHMMs zu erhalten.

Es sind bei den vorgestellten Modellgrößen die üblichen statistischen Randbedingungen einzuhalten (vgl. Gleichung 2.4). Somit ist der Aufbau des pseudo zweidimensionalen Hidden-Markov-Modells mit seinen Parametern vollständig definiert. Im eindimensionalen Fall (Kapitel 2) konnte gezeigt werden, daß durch den Viterbi-Algorithmus die Möglichkeiten gegeben sind, HMMs für die Klassifikation einzusetzen und zudem auch die HMMs an Trainingsdaten anzupassen. Letzteres wird durch den in Unterkapitel 2.2.3 vorgestellten Viterbi-Trainingsalgorithmus ermöglicht, der aufbauend auf einer Viterbi-Segmentierung die sich

daraus ergebenden Häufigkeiten, z.B. der Verwendung einer bestimmten Übergangswahrscheinlichkeit $a_{ij}$ (vgl. Gl. 2.38), bestimmt und nachfolgend die Parameter neu schätzt. Für den Fall, daß ein verallgemeinerter Viterbi-Algorithmus existiert, ist es möglich, die P2DHMMs zu trainieren und Bildmuster mit den Modellen zu klassifizieren. Solch ein verallgemeinerter Viterbi-Algorithmus existiert und wird als zweifachverschachtelter Viterbi-Algorithmus bezeichnet, der im folgenden vorgestellt wird.

### 4.3.2 Zweifachverschachtelter Viterbi Algorithmus

Der zweifachverschachtelte Viterbi-Algorithmus wurde in [Kuo94] von Agazzi und Kuo vorgestellt. Es wird wie im eindimensionalen Fall die wahrscheinlichste Zustandszuordnung $Q^{\star}_{XY}$ bestimmt und basierend auf dieser Kenntnis ein Schätzwert für die Produktionswahrscheinlichkeit $P(O_{XY}, Q^{\star}_{XY}|\lambda) = P^{\star}(O_{XY}|\lambda)$ ermittelt. Da es sich bei der Observation $O_{XY}$ um eine $X \times Y$ Matrix handelt, besteht auch die Größe $Q^{\star}$ im zweidimensionalen Fall aus Elementen, die in Form einer $X \times Y$-Matrix angeordnet sind.

Bedingt durch die Struktur der P2DHMMs, die ein hierarchisches Modell darstellen, wird die Ermittlung der wahrscheinlichsten Zustandssequenz durch zwei ineinander verschachtelte Berechnungen durchgeführt. Zunächst werden die Schätzwerte der Produktionswahrscheinlichkeiten der Bildspalten mit den den Metazuständen zugeordneten HMMs ($\lambda_i$ für $1 \leq i \leq N$) bestimmt. Diese Wahrscheinlichkeiten $P_i(O_x) = P(O_x|\lambda_i)$ werden durch den schon bekannten eindimensionalen Viterbi-Algorithmus berechnet. Im nachfolgenden Schritt werden diese Wahrscheinlichkeiten als Ausgabewahrscheinlichkeiten der Zustände des übergeordneten Markov-Modells ($S_i$ für $1 \leq i \leq N$) verwendet. Auf der Ebene des übergeordneten Markov-Modells werden ebenfalls dem eindimensionalen Fall entsprechende Viterbi-Berechnungen durchgeführt. Der Algorithmus wird im folgenden formal dargestellt, wobei mit den Berechnungen der Zustandszuordnungen auf Bildspaltenebene begonnen wird:

- Initialisierung

$$\begin{aligned} \vartheta^j_{x1}(i) &= \pi^j_i b^j_i(O_{x1}) \\ \psi^j_{x1}(i) &= 0 \end{aligned} \tag{4.25}$$

- Rekursionsschritt

$$\begin{aligned} \vartheta^j_{xy}(i) &= \max_{i-2 \leq k \leq i} (\vartheta^j_{x,y-1}(k) \cdot a^j_{ki}) b^j_i(O_{xy}) \\ \psi^j_{xy}(i) &= \arg\max_{i-2 \leq k \leq i} (\vartheta^j_{x,y-1}(k) \cdot a^j_{ki}) \end{aligned} \tag{4.26}$$

- Terminierung

$$\begin{aligned} P_j(O_x) &= \max_{1 \leq i \leq N^j} (\vartheta^j_{xY}(i)) \\ N_j(O_x) &= \operatorname{argmax}_{1 \leq i \leq N^j} (\vartheta^j_{xY}(i)) \end{aligned} \tag{4.27}$$

Diese Darstellung des Viterbi-Algorithmus unterscheidet sich von der Darstellung in Kapitel 2 in der Verwendung des diskretisierten Ortes $y$ anstelle des Zeitschritts $t$ (vgl. Gleichungen 2.24 bis 2.26). Ferner sind die Größen mit dem hochgestellten Index $j$ versehen, der den Bezug zum Markov-Modell $\lambda_j$ und somit zum $j$-ten Metazustand anzeigt, versehen. Die Erläuterungen zu den Größen $\vartheta$ und $\psi$ sind in Kapitel 2 gegeben. Der zweite Durchlauf des Viterbi Algorithmus kann wie folgt dargestellt werden:

- Initialisierung

$$\begin{aligned} D_1(j) &= \pi_j P_j(1) \\ \gamma_1(j) &= 0 \end{aligned} \tag{4.28}$$

- Rekursionsschritt

$$\begin{aligned} D_x(j) &= \max_{j-2 \leq k \leq j} (D_{x-1}(k) \cdot a_{kj}) P_j(O_x) \\ \gamma_x(j) &= \arg\max_{j-2 \leq k \leq j} (D_{x-1}(k) \cdot a_{kj}) \end{aligned} \tag{4.29}$$

- Terminierung

$$\begin{aligned} P^\star &= \max_{1 \leq j \leq N} (D_X(j)) \\ q_x &= \operatorname*{argmax}_{1 \leq j \leq N} (D_X(j)) \end{aligned} \tag{4.30}$$

In den unterschiedlichen hierarchischen Stufen entsprechen die Größen $D$ und $\vartheta$ bzw. $\gamma$ und $\psi$ einander. Nach einem vollständigen Durchlauf des zweifachverschachtelten Viterbi Algorithmus ist der Schätzwert für die Produktionswahrscheinlichkeit durch die Gleichung 4.30 gegeben. Somit ist es möglich, zweidimensionale Muster mit den P2DHMMs zu klassifizieren. Soll auch die optimale Zustandssequenz ermittelt werden, die verwendet werden kann um ein Viterbi-Training durchzuführen, so kann diese Sequenz mit folgender Rechenvorschrift ermittelt werden:

$$\begin{aligned} q_x &= \gamma_{x+1}(q_{x+1}) \\ q_{xy} &= \psi_{x,y+1}(q_{x,y+1}) \end{aligned} \tag{4.31}$$

Neben der Möglichkeit die Parameter des P2DHMMs mit einem Viterbi-Training zu bestimmen, existiert auch ein zweifachverschachtelter Baum-Welch-Trainingsalgorithmus. Dieser ist in [MM99] beschrieben. Anstelle der zweifachverschachtelten Trainings- bzw. Klassifikationsalgorithmen können jedoch auch die Algorithmen für eindimensionale Markov-Modelle verwendet werden, falls die im folgenden Abschnitt beschriebenen Modellierungsverfahren eingesetzt werden.

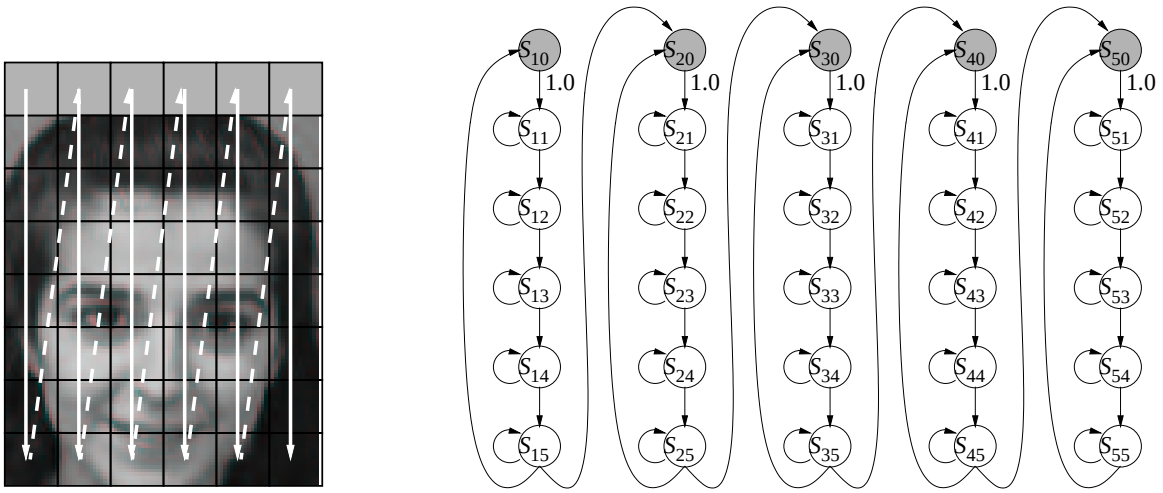


Abbildung 4.3: Eine den P2DHMMs gleichwertige Modellierung, die eindimensionale Hidden-Markov-Modelle verwendet.

### 4.3.3 Umformung in gleichwertige eindimensionale Hidden-Markov-Modelle

Samaria beschreibt in [Sam94b] eindimensionale Hidden-Markov-Modelle, die eine im Vergleich zu den P2DHMMs gleichwertige Modellierung ermöglichen. Dies wird durch das Einfügen von sog. *Markierungszuständen* in die eindimensionalen HMMs und durch das Hinzufügen von korrespondierenden *Markierungsmerkmalen* erreicht. Diese Modellierung ist in Abb. 4.3 schematisch dargestellt. Auf der linken Seite in Abb. 4.3 ist das zu modellierende zweidimensionale Muster gezeigt, bei dem es sich um ein Bild aus einer Gesichtsdatenbank handelt. Um eine Modellierung mit eindimensionalen HMMs zu ermöglichen, müssen zunächst die Merkmale des Bildes in eine Merkmalsequenz überführt werden. Dies geschieht durch die Abtastung des Bildes auf folgende Weise: Merkmale werden, wie in Abb. 4.3 durch weiße Pfeile visualisiert, mit einem Abtastfenster von oben nach unten und links nach rechts entnommen. Der Anfang einer jeden Bildspalte wird durch ein sog. Markierungsmerkmal angezeigt. Diese hinzugefügten Merkmale sind in der Abb. 4.3 grau dargestellt.

Die Topologie des eindimensionalen Modells ist auf der rechten Seite der Abb. 4.3 illustriert. Die grau-schattierten Zustände sind die sog. Markierungszustände, die bei der Emission von Markierungsmerkmalen hohe Wahrscheinlichkeiten ausgeben. Da dem Markierungsmerkmal unmittelbar kein weiteres folgt, sind die Zustandsübergänge von einem Markierungszustand auf sich selbst auf Null gesetzt (z.B. $a_{10,10} = P(q_k = S_{10}|q_{k-1} = S_{10}) = 0$, $a_{20,20} = 0,\ldots$). Die Übergangswahrscheinlichkeiten, die zu den Markierungszuständen führen, also z.B. $a_{15,10}$ und $a_{15,20}$ modellieren die statistischen Abhängigkeiten aufeinanderfolgender Bildspalten. Sie entsprechen somit den Übergangswahrscheinlichkeiten des übergeordneten Markov-Modells bei den pseudo zweidimensionalen Modellen (vgl. Gleichung 4.20).

Bei der Verwendung dieser Modellierungsmethode muß darauf geachtet werden, daß die Markierungsmerkmale ausschließlich den Markierungszuständen zugeordnet werden. Dies kann bei der Verwendung von diskreten Ausgabeverteilungen durch das Reservieren eines Ausgabesymbols $v_m$ geschehen. Die Wahrscheinlichkeitsverteilung der Markierungszustände wird für alle Ausgabesymbole $v_i$ mit $i \neq m$ zu Null gesetzt und für das Symbol $v_m$ zu Eins. Für alle anderen Zustände ist die Wahrscheinlichkeit für die Ausgabe des Markierungssymbols auf Null zu setzen. Durch diese Maßnahmen ist gesichert, daß die vorgestellte eindimensionale Modelltopologie die Abfolge von Bildspalten richtig modelliert und somit mit den P2DHMMs vergleichbare Eigenschaften aufweist. Sollen kontinuierliche Ausgabeverteilungen verwendet werden, so erweist sich die Tatsache als problematisch, daß die verwendeten Gaußfunktionen (siehe Unterkapitel 2.2.4) theoretisch auch im beliebig großen Abstand zum Mittelwert noch von Null verschiedene Werte aufweisen. Somit wäre eine fälschliche Zuordnung von Markierungsmerkmalen und gewöhnlichen Zuständen sowie die Zuordnung von gewöhnlichen Merkmalen und Markierungszuständen möglich. Dies kann jedoch durch programmtechnische Maßnahmen verhindert werden, indem z.B. auf ein festgelegtes Markierungsmerkmal abgefragt wird und die Wahrscheinlichkeit Null ausgegeben wird. Dabei darf das gewählte Markierungsmerkmal nicht im Wertebereich der gewöhnlichen Merkmale vorhanden sein, was bei natürlichen Bildern stets möglich ist. Falls die Merkmale z.B. Grauwerte, die mit 8bit kodiert wurden, repräsentieren, so ist der Wertebereich für die Merkmale $0, \ldots, 255$ und eine geeignete Wahl für das Markierungsmerkmal wäre somit ein Wert außerhalb des 8bit-Wertebereichs.

Diese Modellierungsmethode erlaubt die direkte Verwendung der in Kapitel 2 vorgestellten Trainings- und Klassifikationsalgorithmen für eindimensionalen Hidden-Markov-Modelle. Aus diesem Grund kann somit Software, die für den Einsatz in der automatischen Spracherkennung entwickelt wurde, wie etwa das Hidden-Markov-Toolkit (HTK), unter den erwähnten Voraussetzungen verwendet werden. Daher wurde diese Modellierungstechnik, bei den in dem folgenden Kapitel 5 vorgestellten Experimenten, eingesetzt.

## 4.4 Kapitelzusammenfassung

In diesem Kapitel wurde in die Theorie der zweidimensionalen statistischen Modellierung eingeführt. Zunächst wurden die Markov-Random-Fields vorgestellt, die ein nichtkausales Modell darstellen, welches in der statistischen Physik entwickelt wurde. Aus diesen gehen durch die Einführung einer kausalen Abhängigkeit von benachbarten Bildelementen und eines zweiten statistischen Prozesses die zweidimensionalen Hidden-Markov-Modelle hervor. Dieser zweite statistische Prozeß ist wie im eindimensionalen Fall die Ausgabe von Merkmalen durch Ausgabeverteilungen bzw. -dichten. Sowohl die Markov-Random-Fields, als auch die zweidimensionalen HMMs sind für die Mustererkennung wenig geeignet, da keine effizienten Algorithmen existieren, um ein Training der Modelle anhand von Musterbeispie-

len zu ermöglichen. Durch weitere Vereinfachungen wurden die pseudo-zweidimensionalen Hidden-Markov-Modelle abgeleitet, für die ein erweiterter Viterbi-Algorithmus existiert. Somit existiert ein effizienter Algorithmus für diese Modelle, der eine Parameterbestimmung durch ein Training mit Beispielen ermöglicht. Zusätzlich kann durch diesen Viterbi-Algorithmus eine Segmentierung eines Musters in Kombination mit einer Klassifikation erfolgen. Dies wird im folgenden Kapitel verwendet, um zweidimensionale Muster in einer komplexen Szene aufzufinden und zu erkennen.

# Kapitel 5

# Ein integrierter Ansatz zur Klassifizierung und Segmentierung mit pseudo zweidimensionalen Hidden-Markov-Modellen

Dieses Kapitel beschreibt ein neuartiges Verfahren, mit dem zweidimensionale Muster in komplexen Umgebungen aufgefunden und klassifiziert werden können. Durch das Einbringen von an die Umgebung adaptierten Zuständen in die in Kapitel 4 vorgestellten pseudo zweidimensionalen Hidden-Markov-Modelle, kann die Segmentierung und Klassifizierung auf integrierte Weise erfolgen. Nach einem Training der Muster-HMMs und der Umgebungszustände kann durch die vom Viterbi-Algorithmus festgelegte Merkmal-Zustandszuordnung eine Segmentierung der Bildszene erfolgen. Gleichzeitig gibt der Viterbi-Algorithmus einen Schätzwert für die Produktionswahrscheinlichkeit des Musters aus, mit dem eine Klassenzugehörigkeit ermittelt werden kann. Bevor dieses Verfahren detailliert vorgestellt wird, gehen die folgenden beiden Unterkapitel zunächst auf die zweidimensionale Musterklassifikation bzw. die rotationsinvariante Modellierung von Objektformen mit P2DHMMs ein.

## 5.1 Klassifizierung von Bildern mit P2DHMMs

Die Klassifizierung von Mustern mit P2DHMMs soll hier kurz am Beispiel der Erkennung von Gesichtern dargestellt werden. Die Hidden-Markov-Modelle sind in das Gebiet der automatischen Gesichtsklassifikation von Samaria von der Cambridge University eingeführt worden (siehe [Sam94b]). Zunächst wurde die eindimensionale, aus der Spracherkennung bekannte Variante verwendet und später die pseudo zweidimensionale, welche eine signifikante Fehlerreduktion erreichte ([Sam94a]).

Die Klassifizierung von Gesichtern mit P2DHMMs erfolgt durch die folgenden beiden Schritte: Der erste Schritt, nämlich die Merkmalextraktion, basiert auf der Diskreten-Cosinus-Transformation (DCT). Ein gegebenes Bild wird mit einem Abtastfenster, entsprechend dem in Abb. 4.3 illustrierten Schema, von oben nach unten und links nach rechts abgetastet. Die Bildpunkte in dem Abtastfenster mit der festen Größe $N \times N$ werden durch die Anwendung der DCT transformiert. Die Transformationsvorschrift lautet (siehe auch [Gon92]):

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad \text{für} \quad u,v = 0,\ldots,N-1 \tag{5.1}$$

Die Werte für $\alpha(u)$ ergeben sich aus

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{für} \quad u = 0 \\ \sqrt{\frac{2}{N}} & \text{für} \quad u = 1,\ldots,N-1 \end{cases} \tag{5.2}$$

Es hat sich anhand von experimentellen Ergebnissen gezeigt, daß es vorteilhaft ist, nicht alle Koeffizienten zu verwenden, sondern die 15 Koeffizienten mit dem höchsten Betrag auszuwählen. Dies sind die Koeffizienten $C(u,v)$ für die die Bedingung $(u+v \leq 4)$ gilt. Ebenfalls als vorteilhaft hat sich die Verwendung einer Blocküberlappung von 75% in beiden Bilddimensionen erwiesen. Diese Überlappung benachbarter Abtastblöcke ermöglicht die Modellierung von Kontext und ist vergleichbar mit den sog. Delta-Merkmalen in der automatischen Spracherkennung. Die Experimente, die zu dieser Konfiguration geführt haben, sind in den Arbeiten [Eic99b] und [Eic00] dokumentiert.

Der zweite Schritt ist die statistische Klassifikation mit den pseudo zweidimensionalen Hidden-Markov-Modellen. Jeweils ein P2DHMM mit einer Struktur wie sie in Abb. 4.3 dargestellt ist, wird für jede Person in der Trainingsdatenbasis mit dem Baum-Welch-Algorithmus trainiert. Die Klassifikation erfolgt dann durch die Maximum-Likelihood-Entscheidung auf Basis der mit dem Viterbi-Algorithmus ermittelten Produktionswahrscheinlichkeiten. Das zu testende Bild wird als diejenige Person erkannt, deren zugeordnetes Modell die höchste Produktionswahrscheinlichkeit ausgibt (siehe auch Gleichung 2.12).

Experimente wurden mit diesem Verfahren auf der Olivetti Research Laboratory (ORL) Datenbasis durchgeführt und sind in [Eic00] und [Eic99b] beschrieben. Die ORL Datenbasis besteht aus je zehn Frontalansichten von 40 verschiedenen Personen, wobei die Aufnahmen sowohl Beleuchtungsvariationen als auch Variationen in der Mimik aufweisen. Die Abbildungen 4.3 und 5.1 zeigen zwei Beispielbilder aus dieser Datenbasis. Diese Datenbasis wurde in der Literatur intensiv eingesetzt (siehe z.B. [Sam94a, Sam94b] und [Che95]) und es wurden mit verschiedenen Klassifizierungsmethoden Erkennungsgenauigkeiten zwischen 80 und 99,5% erreicht. Die beschriebene Methode, die auf DCT-Koeffizienten und P2DHMMs basiert, erreichte eine Genauigkeit von 100% ([Eic99b]).

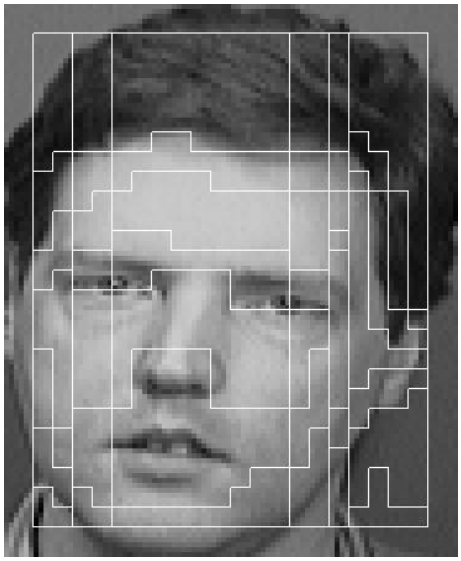

Abbildung 5.1: Zuordnung von Merkmalen und Modellzuständen durch den Viterbi-Algorithmus (aus [Eic00])

Obwohl diese optimale Erkennungsgenauigkeit erreicht wurde, so kann die Aufgabe der automatischen Gesichtserkennung für reale Anwendungen nicht als gelöst gelten. Bei der ORL-Datenbasis sind die zu klassifizierenden Bilder, wie auch die Testbilder, sowohl geeignet skaliert, als auch optimal ausgeschnitten. In Abb. 5.1 läßt sich erkennen, daß das Gesicht (inklusive der Haare) nahezu das gesamte Bild ausfüllt. Dies ist für alle Bilder dieser Datenbasis der Fall. Somit entfällt der Prozeß des Auffindens des Gesichts in dem vorgegebenen Bild. Durch diesen Auffindungsprozeß und die damit verbundene mögliche Fehlsegmentierung ergeben sich im Klassifikationsschritt zusätzliche Fehler.

Kapitel 5.3 beschreibt eine im Rahmen dieser Arbeit entwickelte Modellierungstechnik, die es ermöglicht, das Auffinden und die Klassifikation eines zweidimensionalen Musters in einem Verarbeitungsschritt mit den pseudo zweidimensionalen Hidden-Markov-Modellen durchzuführen. Diese Methode basiert auf der Fähigkeit des Viterbi-Algorithmus, die Merkmale den Modellzuständen automatisch zuzuordnen. Die Abb. 5.1 visualisiert, für ein in Merkmalblöcke der Größe $8 \times 8$ zerlegtes Bild, solch eine Merkmals-Zustandszuordnung unter Verwendung eines P2DHMM, das aus $6 \times 7$ Zuständen besteht. Diejenigen Bildbereiche, deren Merkmale einem einzigen Zustand zugeordnet wurden, sind in der Abbildung von einer weißen Linie umgeben. In der Abb. 5.1 ist gut zu erkennen, daß die Ausgabeverteilungen der Modellzustände jeweils auf bestimmte Bereiche des Bildes adaptiert sind. So wurde der Mund der abgebildeten Person beispielsweise einem einzigen Zustand zugeordnet. Ebenso gibt es Zustände, die ausschließlich dem Hintergrund oder den Haaren zugeordnet wurden. Sind solche Zustände entsprechend identifiziert, so könnte das Bild durch den Viterbi-Algorithmus in verschiedene Bereiche, wie z.B. die Haarregion, den Mundbereich oder den Hintergrund segmentiert werden. Dies ist eine zusätzliche Information, die sich

während des Klassifizierungsschrittes ergibt. Gelingt es, die Merkmale der Umgebung eines zu erkennenden Musters zu modellieren, so könnte durch den Viterbi-Algorithmus, in einem Schritt, ein Muster in seiner Umgebung aufgefunden werden und das Muster klassifiziert werden.

Bevor die gemeinsame Klassifizierung und Segmentierung mit P2DHMM und Umgebungsmodell ausführlich in Unterkapitel 5.3 vorgestellt wird, geht das folgende Unterkapitel auf die rotationsinvariante Modellierung von Objektformen mit P2DHMMs ein.

## 5.2 Rotationsinvariante Modellierung von Objektformen mit P2DHMMs

Die rotationsinvariante Modellierung mit P2DHMMs wurde im Rahmen dieser Arbeit entwickelt. Sie ermöglicht eine Modellierung von gedrehten Objekten, die eine hohe Toleranz gegenüber Deformationen, die z.B. von Überdeckungen mehrerer Objekte hervorgerufen wurden, aufweist. Insbesondere ist es mit diesem Ansatz möglich, polar abgetastete Muster sowohl in radialer, als auch in zirkularer Richtung verzerrungstolerant zu modellieren. Dies stellt bei stark deformierten Mustern, die etwa von anderen Objekten überdeckt waren, einen Vorteil dar (siehe auch Abb. 3.14). In Kapitel 3 wurde der hier vorgestellte Ansatz kurz erwähnt und in Tabelle 3.3 im Unterkapitel 3.5.5 bereits quantitative Ergebnisse präsentiert, die mit einer Datenbank, die aus deformierten Objekten besteht, erzielt wurden. Da sich Kapitel 3 mit der translations-, skalierungs- und rotationsinvarianten Modellierung von Objektformen bzw. handskizzierten Piktogrammen mit *eindimensionalen* HMMs befaßt und zudem in die Theorie der P2DHMMs erst in Kapitel 4 eingeführt wurde, erfolgt die detaillierte Darstellung des Ansatzes an dieser Stelle.

Abb. 5.2 zeigt die P2DHMM-Topologie, die für die rotationsinvariante Modellierung verwendet werden kann. Auf ähnliche Weise wie in Abb. 3.5 wird das Originalmodell zweimal dupliziert und von den Modellkopien umgeben. Zusätzlich werden die Wahrscheinlichkeiten für die Anfangszustände des ersten Modells sowie die Wahrscheinlichkeiten für die Endzustände des dritten P2DHMM entsprechend Abb. 5.2 verändert. Die Wahrscheinlichkeiten für die Anfangszustände des ersten Modells werden alle auf den Wert $1/N$ gesetzt, wobei $N$ die Anzahl der Metazustände des Originalmodells bezeichnet. Dieser Schritt ist durch die Annahme motiviert, daß alle Drehungswinkel gleichwahrscheinlich sind. Bei dem Versuch, dies auf die Endzustände zu übertragen, treten hingegen Schwierigkeiten auf. Auf diesen Punkt wurde ausführlich in Unterkapitel 3.3.2 eingegangen.

Trotz der offensichtlichen Ähnlichkeiten der Modellstruktur in Abb. 5.2 und der eindimensionalen Struktur in Abb. 3.5 ist dennoch zu beachten, daß die hierarchische Struktur des P2DHMM-Ansatzes zu einer veränderten Modellierung der Merkmale führt. So werden entsprechend Abb. 3.2 und Gleichung 3.2 polar abgetastete Muster auf folgende Weise

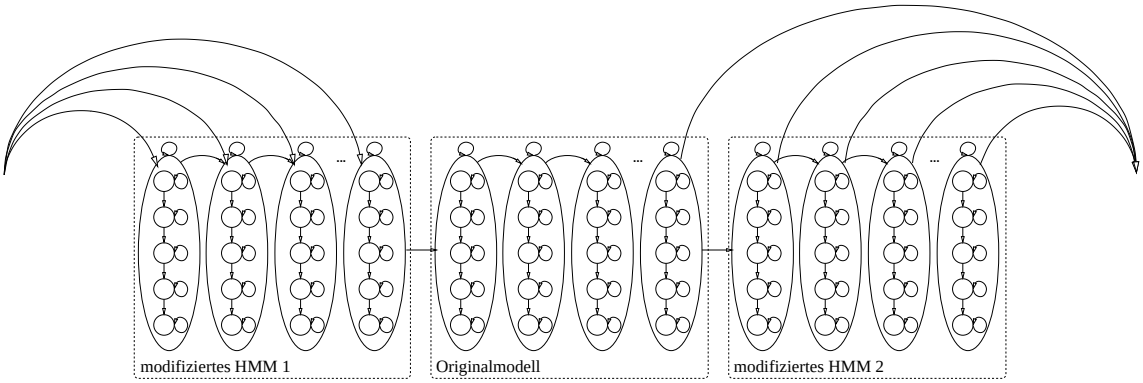


Abbildung 5.2: P2DHMM, das für die Modellierung von deformierten Objektformen verwendet werden kann

modelliert: Die für einen konstanten Winkel bestimmten Abtastwerte ($\varphi =$ const.) werden einem Metazustand zugeordnet, was eine nichtlineare Musterverzerrung in zirkularer Richtung ermöglicht. Zusätzlich werden diese Abtastwerte selbst von eindimensionalen Hidden-Markov-Modellen modelliert. Dies ermöglicht eine nichtlineare Musterverzerrung in radialer Richtung.

Die mit diesem Ansatz erzielten Ergebnisse belegen die verformungstoleranten Modellierungseigenschaften der P2DHMMs. Die Experimente wurden mit einer Datenbasis, die aus künstlich deformierten Objekten besteht, durchgeführt (siehe auch Unterkapitel 3.5.5). Die künstlichen Deformationen sollen die Effekte von Teilüberdeckungen mit anderen Objekten wiederspiegeln. Wie Tabelle 3.3 entnommen werden kann, ist die Retrieval-Effizienz gegenüber der Modellierung mit eindimensionalen HMMs von $\eta_T$ =75,00% auf $\eta_T$ =79,17% verbessert worden. Die im Vergleich mit den eindimensionalen Modellen gesteigerte Toleranz bei Musterverzerrungen in radialer Richtung hat sich hier positiv ausgewirkt. Es sei jedoch auch darauf hingewiesen, daß die Methode mit den eindimensionalen HMMs in Kombination mit den Merkmalstrom-Gewichten nochmals bessere Ergebnisse lieferte.

Nach dieser Beschreibung der rotationsinvarianten Modellierung mit P2DHMMs wird nun im folgenden Kapitel der schon zu Beginn des Kapitels erwähnte Ansatz zur integrierten Klassifizierung und Segmentierung mit pseudo zweidimensionalen HMMs vorgestellt.

## 5.3 Klassifizierung und Segmentierung mit P2DHMMs und Umgebungsmodell

Ein grundlegendes Problem der klassischen Bildverarbeitung, das bei Mustererkennungsaufgaben in der realen Welt auftritt, ist die große Abhängigkeit von guten Segmentierungsergebnissen. Die Bildsegmentierung wird für die darauf folgende Objektklassifizierung zu einem entscheidenden Schritt: Zunächst muß das Objekt vom Hintergrund isoliert werden,

um dann anschließend klassifiziert zu werden. Diese Trennung vom Hintergrund ist jedoch vielfach nur in Spezialfällen möglich, zum Beispiel falls das zu klassifizierende Objekt im Vergleich zum Hintergrund unterschiedliche Farb- oder Grauwerte aufweist. Bei vielen Mustererkennungsaufgaben sind solche idealen Bedingungen jedoch nicht vorhanden. So ist es kaum möglich, bei Aufnahmen von Straßenszenen, die Passanten, Fahrzeuge, Verkehrsschilder und Häuser zeigen, aufgrund von Farbinformationen Personen zu finden bzw. zu segmentieren. Eine solche Trennung in Objekt und Hintergrund ist ebenfalls nicht möglich, falls es keine Farbinformationen gibt, wie im Fall von Zeichnungen.

Die Abb. 1.1 in der Einleitung z.B. zeigte eine Schwarz-Weiss Skizze, bei der ein Piktogramm in eine komplexe Umgebung[1] eingebettet ist. Bemerkenswert ist, daß menschliche Betrachter ohne Mühe solche komplizierten Segmentierungsaufgaben bewältigen können, da sie in einem einzigen Schritt das Objekt vom Hintergrund trennen und gleichzeitig klassifizieren können. Dies führt zu der Annahme, daß es in vielen Anwendungen nur möglich ist, ein Objekt zu segmentieren, falls es im gleichen Verarbeitungsschritt erkannt und segmentiert wird. Genau diese Anforderung wird von den Hidden-Markov-Modellen erfüllt. Falls beispielsweise der Hintergrund in einem Bild von einem HMM modelliert wird und ein Objekt in dem Bild von einem weiteren Markov-Modell modelliert wird und sind ferner beide auf geeignete Weise miteinander verbunden, so kann mit dem Viterbi-Algorithmus eine Zuordnung der Merkmale von Hintergrund und Objekt zu den entsprechenden Modellen bestimmt werden. Das Ergebnis dieser Zuordnung ist also eine automatische Segmentierung des Bildes in einen Objektanteil und in einen Bildhintergrund. Zusätzlich zu dieser Segmentierung liefert der Viterbi-Algorithmus ein implizites Erkennungsergebnis, da der Viterbi-Algorithmus einen Näherungswert für die Wahrscheinlichkeit ausgibt, daß das Objekt von dem Hidden-Markov-Modell produziert wird.

### 5.3.1 Objekt-HMM mit Umgebungszuständen

Bei Betrachtung von Abb. 1.1 wird es offensichtlich, daß zweidimensionale Modellierungstechniken verwendet werden müssen, um die Skizze bearbeiten zu können. Daher bietet sich die Modellierung mit pseudo zweidimensionalen Hidden-Markov-Modellen an. Bei der integrierten Segmentierung und Klassifizierung mit P2DHMMs wird auf folgende Weise verfahren:

Schritt 1: Training von Objekt-HMMs für jede einzelne Klasse. Dabei können die in Unterkapitel 5.1 beschriebenen Schritte verwendet werden.

Schritt 2: Bestimmung der Wahrscheinlichkeitsverteilung der Merkmale des Hintergrunds.

[1] Die Begriffe *Hintergrund* und *Umgebung* werden im folgenden auf gleiche Weise verwendet. Der Grund hierfür ist die gemeinsame Betrachtung von künstlich erzeugten und natürlichen Bildern.

Schritt 3: Umgeben der bereits trainierten Objekt-Modelle mit Zuständen, die die Wahrscheinlichkeitsverteilung der Hintergrundmerkmale als Ausgabewahrscheinlichkeiten verwenden. Diese Zustände werden im folgenden *Umgebungszustände* genannt.

Schritt 4: Anwendung des Viterbi-Algorithmus und anschließende Analyse der Merkmal-Modellzustandszuordnung, sowie Maximum-Likelihood-Entscheidung

Zu dieser Verfahrensweise ist folgendes anzumerken: Das Training von Objekt-Modellen geschieht auf analoge Weise zu dem Training, wie es in Kapitel 5.1 beschrieben wurde. Für jede a-priori festgelegte Klasse wird ein entsprechendes P2DHMM trainiert.

Die Bestimmung der Wahrscheinlichkeitsverteilung der Umgebungsmerkmale stellt hingegen ein noch zu lösendes Problem dar. Dabei sind grundsätzlich zwei verschiedene Situationen zu unterscheiden: Ein mögliches Szenario besteht darin, daß a-priori Wissen über den Hintergrund vorliegt. In diesem Fall können die Wahrscheinlichkeitsverteilungen entsprechend adaptiert werden. Liegt das Wissen über den Hintergrund beispielsweise als Beispiel- oder Trainingsmuster vor, so kann mit diesem Beispielmuster direkt ein Training erfolgen.

Der sehr viel schwierigere Fall besteht darin, daß kein a-priori Wissen über den Hintergrund vorliegt. In diesem Fall kann keine allgemeingültige Methode angegeben werden, um die Verteilung der Merkmale des Hintergrunds zu schätzen. Beispiele, wie für diese schwierige Aufgabe dennoch praxisrelevante Lösungen gefunden werden können, sind in den folgenden Unterkaptiteln beschrieben. Das Bestimmen dieser Wahrscheinlichkeitsverteilungen kann auch als das Training eines Markov-Modells, welches nur aus einem einzigen Zustand besteht, angesehen werden. Die in der Literatur häufig zu findende Bestimmung von sog. Gaußschen-Mischverteilungen (GMM) ist identisch zu dem Training eines Hidden-Markov-Modells mit einem Zustand. Der dritte Schritt, der in obiger Aufzählung angegebenen wurde, ist das Zusammenfügen der vorher bestimmten Modelle. Dabei ist das Objekt-P2DHMM vollständig mit Umgebungszuständen zu umgeben. Der letzte Schritt besteht in der Anwendung des Viterbi-Algorithmus und der schon beschriebenen Analyse der Merkmal-Modellzustandszuordnung und der mit der Wahrscheinlichkeitsschätzung verbundenen Klassifikation. Anwendungen und Evaluierungen der hier vorgestellten Methode werden in den folgenden Kapiteln vorgestellt.

### 5.3.2 Erkennen von handskizzierten Piktogrammen in komplexen Szenen

Die Aufgabe, handskizzierte Piktogramme zu erkennen stellt eine große Herausforderung dar (siehe auch Kapitel 3.3). Der Grund hierfür liegt in den großen Variationen innerhalb der Klassen, selbst bei einem einzigen Schreiber. Sind diese Piktogramme zusätzlich in verschiedene Umgebungen eingebettet, so wird die Aufgabe noch erheblich erschwert. In diesem Fall muß zusätzlich zu der Diskriminierung der einzelnen Piktogrammklassen auch das Pik-

togramm vom Hintergrund unterschieden, bzw. lokalisiert werden. In diesem Kapitel besteht der Hintergrund aus denselben Konstruktionselementen wie die Piktogramme selbst, nämlich aus handschriftlich angefertigten Linien und Schraffuren. Die Piktogramme, die in den künstlich erstellten Szenen eingebettet sind, sind dieselben, die in Kapitel 3.3 bzw. Abb. 3.1 vorgestellt wurden. Abbildung 5.3 zeigt sechs Bilder aus der verwendeten Datenbasis. Es

Abbildung 5.3: Handschriftlich skizzierte Piktogramme derselben *Klasse 16* eingebettet in drei verschiedene Umgebungen.

wurden Piktogramme derselben Klasse, nämlich Klasse 16 aus Abb. 3.1 in drei verschiedene Umgebungen eingebettet. Die Aufgabe besteht nun darin, die Piktogrammklasse 16 von neunzehn konkurrierenden Klassen zu unterscheiden und das bei einer großen Variation in Größe, Form und Position. Zusätzlich sind die Linien, die den Hintergrund bilden von dem Piktogramm zu unterscheiden. Es kann dabei bei einigen Elementen der erstellten Datenbasis durchaus vorkommen, daß die Hintergrundschraffur in das zu bestimmende Piktogramm hineinragt, oder es berührt (siehe z.B. das untere linke Bild in Abb. 5.3). Insbesondere in diesen Fällen ist es mit konventionellen Methoden nahezu unmöglich, diese Szenen vorzusegmentieren.

Diese Aufgabe kann mittels pseudo zweidimensionaler Hidden-Markov-Modelle gelöst werden. Zu diesem Zweck wird das in Abb. 5.4 dargestellte Modell entsprechend den Ausführungen in Kapitel 5.3.1 verwendet. Die weiß dargestellten Zustände repräsentieren die mit Piktogrammerkmalen trainierten Zustände, wohingegen die grau dargestellten Zustände für die mit Hintergrundmerkmalen trainierten Zustände stehen, also Umgebungszustände sind. Die gepunkteten Modellübergänge in Abb. 5.4 wurden im Gegensatz zu den durchgezogenen Übergänge nicht auf Daten trainiert, sondern willkürlich ausgewählt. Der Grund hierfür ist, daß keine Trainingsdaten zur Verfügung stehen. Durch das Festlegen der in Abb. 5.4 gepunktet dargestellten Übergangswahrscheinlichkeiten wird das a-priori erwartete Flächenverhältnis zwischen Piktogramm und Bildhintergrund einer zu analysierenden Szene modelliert. Da dieses Verhältnis für die zu testenden Bilder nicht als bekannt vorausgesetzt wird, ste-

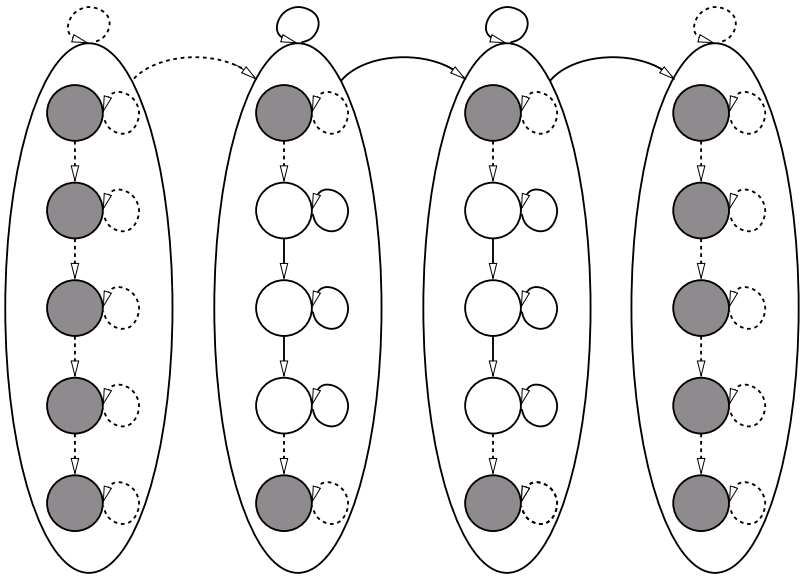

Abbildung 5.4: Pseudo zweidimensionales Hidden-Markov-Modell mit Umgebungszuständen

hen somit keine Trainingsdaten zur Verfügung. Die grau dargestellten Modellzustände teilen alle dieselbe Ausgabewahrscheinlichkeitsdichte. Dies wird im allgemeinen als Parameter-Verknüpfung (engl. tying) bezeichnet (siehe [You92]).

### 5.3.3 Bestimmung der Parameter der Umgebungszustände unter Verwendung von Vorwissen

Um die Ausgabewahrscheinlichkeiten der Umgebungszustände zu schätzen, wurden im Rahmen dieser Arbeit zwei verschiedene Strategien entwickelt. Die erste bezieht Vorwissen über die zu analysierende Szene in die Bestimmung der Ausgabeverteilung ein, während die zweite Strategie ohne dieses Vorwissen auskommt. Die erste entwickelte Methode verwendet eine große Anzahl von Bildern, die Piktogramme in derselben Umgebung, die auch für das Testbild erwartet wird, enthalten. Idealerweise würden Bilder, die ausschließlich Hintergründe enthalten, verwendet. Da solche Bilder jedoch erst hätten erstellt werden müssen, wurden stattdessen die bereits existierenden Bilder der Datenbasis verwendet. Da in diesen Bildern die eingebetteten Piktogramme aus verschiedenen Klassen stammen und sich ferner stets an anderen Positionen im Bild befinden, werden in die Umgebungszustände hauptsächlich die Eigenschaften des Hintergrundes eintrainiert. Der erste Schritt bei beiden Verfahren ist entsprechend der Darstellung in Kapitel 5.3.1 die Erstellung von klassenrepräsentierenden pseudo zweidimensionalen Hidden-Markov-Modellen. Dieser Schritt ist in Abb. 5.5 illustriert. Die Abbildung zeigt, wie mit Beispielbildern der *Klasse 9* das P2DHMM $\lambda_9$ trainiert wird. Die dabei verwendete Merkmalextraktion ist eine einfache Unterabtastung der Beispielbilder. Es wurde beispielhaft in Abb. 5.5 von einem Modell der Größe $2 \times 3$ ausgegangen.

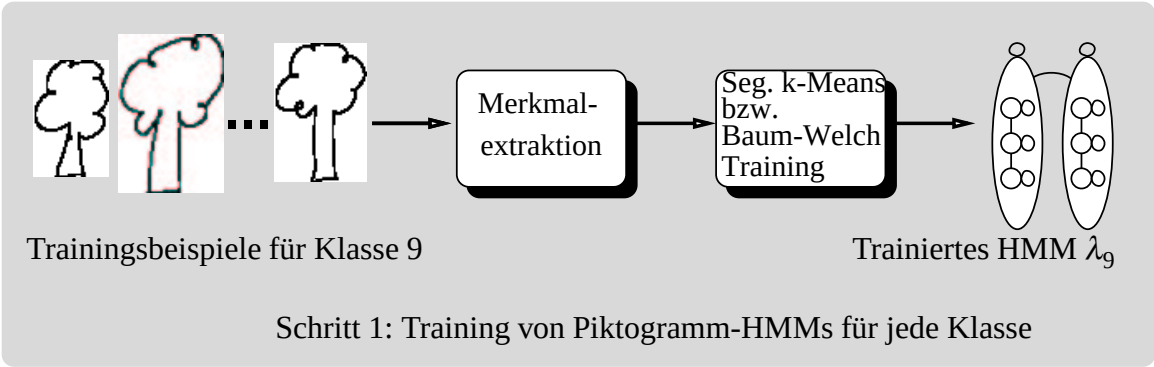


Abbildung 5.5: Training des P2DHMMs für *Klasse 9*

Die Abb. 5.6 zeigt schematisch die weiteren Bearbeitungsschritte, nämlich die Schätzung der Parameter der Umgebungszustände und den integrierten Segmentierungs- und Klassifizierungsschritt. Nachdem ein zu testendes Bild dem System präsentiert wurde, wird für jedes erweiterte Markov-Modell der Viterbi-Algorithmus durchgeführt. Die vergrößerten Modelle haben entsprechend Abb. 5.6, unter Bezugnahme auf die in Abb. 5.5 gezeigte Größe, eine Größe von $4 \times 5$ Zuständen. Die vom Viterbi-Algorithmus ausgegebenen Schätzwerte für die Produktionswahrscheinlichkeiten werden verwendet, um mittels der Maximum-Likelihood(ML)-Entscheidung das Eingangsmuster zu klassifizieren. Dabei wurden die Schätzwerte der Produktionswahrscheinlichkeiten sowohl auf den Hintergrundmerkmalen, als auch auf den Piktogrammerkmalen berechnet, die durch den Viterbi-Algorithmus den Umgebungs- bzw. Piktogrammzuständen zugeordnet werden. Eine mögliche Nachbearbeitung, wie etwa das Herausrechnen von Wahrscheinlichkeiten, die auf den Hintergrundmerkmalen berechnet wurden, ist nicht notwendig, da der Hauptanteil der Produktionswahrscheinlichkeit auf den Piktogrammerkmalen berechnet wird. Ein weiterer Grund ist, daß die Parameter der in Abb. 5.6 dargestellten Umgebungszustände auch über die verschiedenen Modelle hinweg miteinander verknüpft sind. Sie teilen sich alle dieselbe Ausgabeverteilung. Ist also die Anzahl der Merkmale, die den Umgebungszuständen zugeordnet werden für die verschiedenen Modelle in etwa gleich, so ergeben sich aufgrund der gleichen Ausgabeverteilungen auch gleiche Produktionswahrscheinlichkeiten.

### 5.3.4 Adaptive Bestimmung der Parameter der Umgebungszustände

Die Erfahrungen, die durch die Experimente gesammelt wurden, die mit der im vorherigen Unterkapitel beschriebenen Methode durchgeführt wurden, konnten genutzt werden, um eine verbesserte Methode zu entwerfen. Dieses neue Verfahren verwendet eine Parameterschätzung für den Umgebungszustand, welches kein Vorwissen über das zu klassifizierende Bild erfordert. Dieses adaptive Verfahren ist in Abb. 5.7 schematisch dargestellt. Die Bestimmung der Parameter des Umgebungszustands wird hierbei direkt auf dem unsegmentierten Testbild

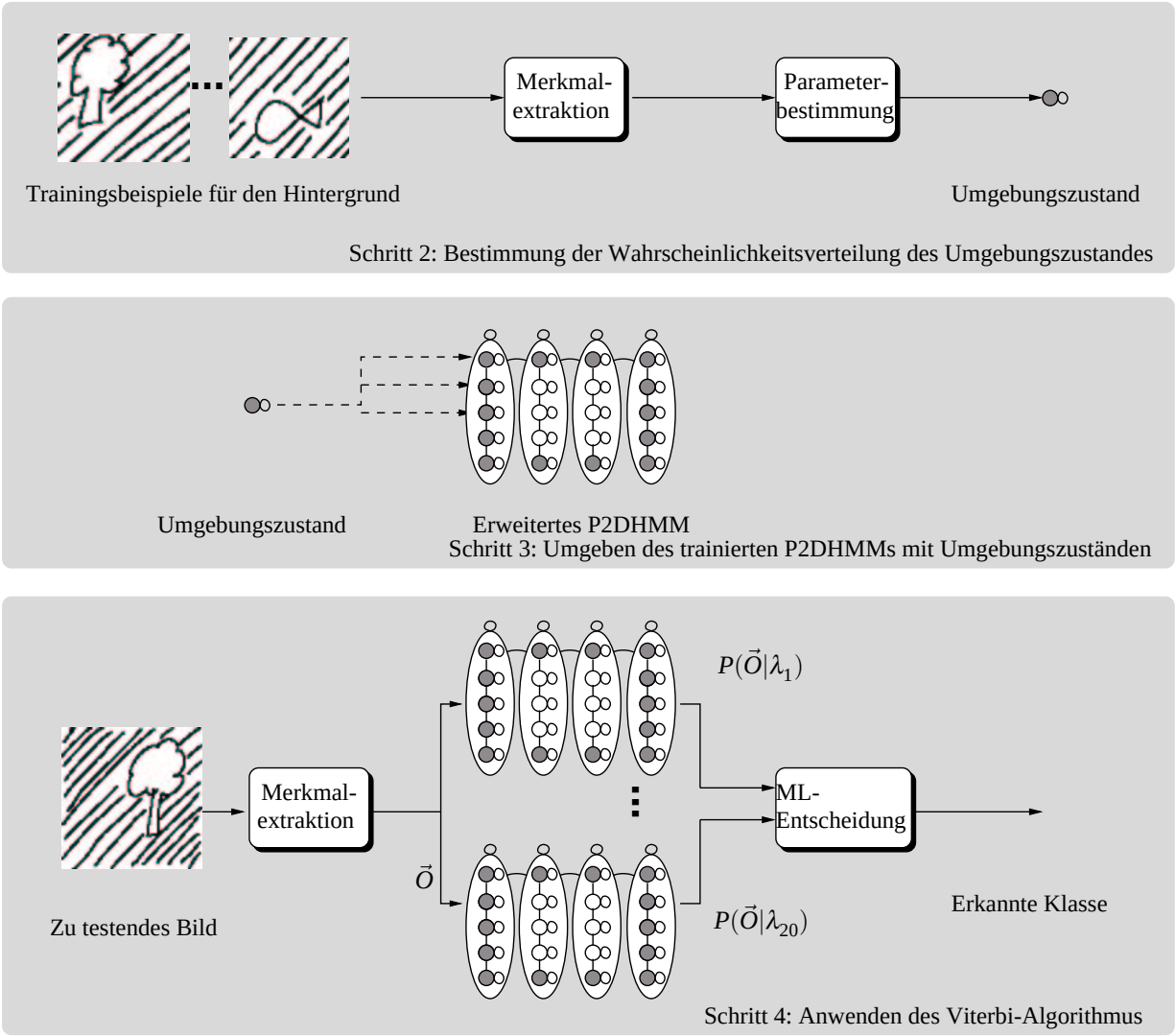


Abbildung 5.6: Schematische Darstellung der auf a-priori Wissen basierenden Parameterschätzung für den Umgebungszustand sowie die integrierte Segmentierung und Klassifizierung.

durchgeführt. Anders formuliert wird die vollständige Sequenz der Merkmalvektoren, die dem Eingangsbild entnommen wird, zum Training des Umgebungszustandes verwendet. Da bei der verwendeten Datenbasis der Hintergrund die größte Fläche ausfüllt und mithin auf diesem Hintergrund die meisten Merkmalvektoren berechnet werden, wird der Umgebungszustand überwiegend auf die Eigenschaften der Hintergrundmerkmale adaptiert.

### 5.3.5 Experimentelle Ergebnisse

Die beiden vorgestellten Methoden wurden auf einer Piktogramm-Datenbasis evaluiert, die in der Arbeit [Mul99a] vorgestellt wurde und folgendermaßen aufgebaut ist: Es sind 200 isolierte Piktogramme vorhanden, jeweils 10 verschiedene für jede der Klassen in Abb. 3.1. Weiterhin sind 300 Bilder mit Piktogrammen in drei unterschiedlichen Umgebungen (vgl. Abb. 5.3) vorhanden. Es handelt sich dabei um fünf Beispiele pro Klasse und Hintergrund. Die drei verschiedenen Hintergründe sind sehr gut durch die Beispiele in Abb. 5.3 repräsentiert. Wie der Abbildung ebenfalls entnommen werden kann, variiert neben der Gesamtgröße der Bilder auch die Position der Piktogramme innerhalb des Bildes und zudem auch das Flächenverhältnis zwischen dem Piktogramm selbst und dem Hintergrund. Dieses Flächenverhältnis variiert in der verwendeten Datenbasis zwischen 11 und 41%. Die Bilder wurden auf einem Graphiktableau vom Autor gezeichnet. Es wurde für die Experimente die folgende Merkmalextraktion, die eine Unterabtastung des Bildes darstellt, verwendet: Jedes Bild wurde in Blöcke der Größe $30 \times 30$ Bildpunkte, unter Verwendung einer Blocküberlappung von 75%, eingeteilt. Danach wurde jeder Bildblock in neun weitere Teilblöcke geteilt und auf diesen dann der mittlere Grauwert bestimmt. Nach der Merkmalextraktion wurde analog zu Abb. 5.5 jeweils ein P2DHMM für jede Piktogramm-Klasse auf zehn isolierten Beispielbildern trainiert. Diese pseudo zweidimensionalen Modelle hatten eine Größe von fünf Metazuständen mit jeweils fünf Modellzuständen. Nachdem die Ausgabeverteilungen der Umgebungszustände mit Hilfe einer der beiden vorgestellten Verfahren bestimmt waren, wurden die trainierten P2DHMM mit den Umgebungszuständen umgeben und somit auf die Größe $7 \times 7$ erweitert. Der Viterbi-Algorithmus wird dann verwendet, um die wahrscheinlichste Zustandssequenz zu bestimmen, die das aktuelle Bild generieren kann. Das Ergebnis des Viterbi-Algorithmus ist die Zuordnung der Merkmale zu den Piktogramm- und Umgebungszuständen. Eine solche Zuordnung ist in Abb. 5.8 für eines der in Abb. 5.3 vorgestellten Beispiele, angegeben. In der Abb. 5.8 steht die graue Schattierung für Merkmalblöcke, die den in Abb. 5.4 ebenfalls grau dargestellten Umgebungszuständen zugeordnet wurden und die weiße Fläche für Merkmale, die dem Piktogrammodell zugeordnet wurden.

Die Segmentierung isoliert betrachtet stellt kein optimales Ergebnis dar. Die helle Fläche in Abb. 5.8, die dem Piktogramm zugeordnet ist, scheint aus vertikal gegeneinander verschobenen Streifen unterschiedlicher Länge zu bestehen. Somit kann gefolgert werden, daß eine Modell-Topologie verwendet wurde, bei der die Bildspalten von Metazuständen model-

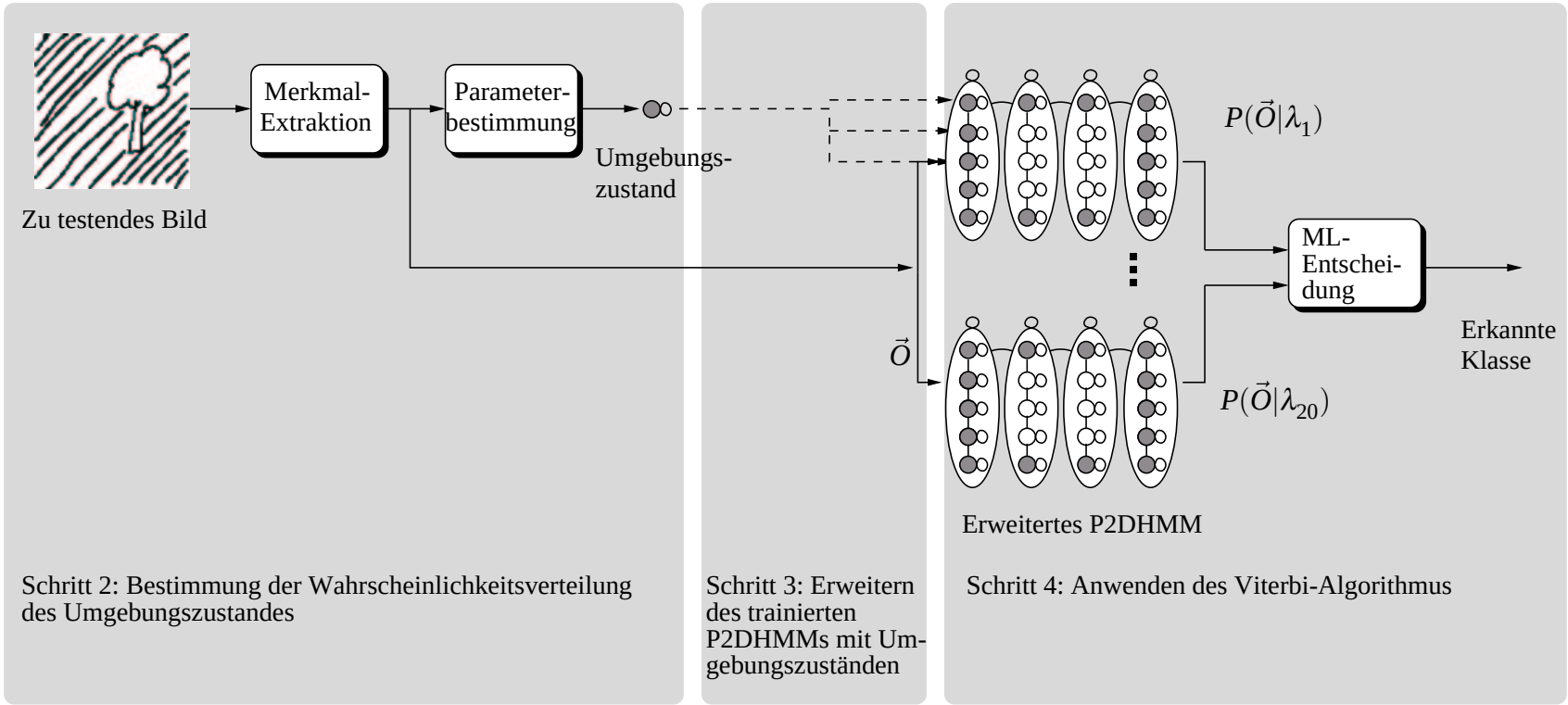


Abbildung 5.7: Adaptive Parameterschätzung

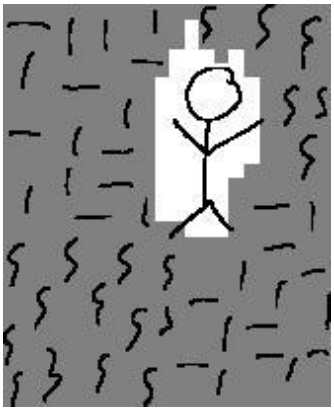

Abbildung 5.8: Segmentierungsergebnis nach Anwendung des Viterbi-Algorithmus

| Richtung | Hintergrund 1 | Hintergrund 2 | Hintergrund 3 | Adaptives Verfahren |
|---|---|---|---|---|
| | 91,0% | 90,0% | 90,0% | 90,3% |
| | 91,0% | 89,0% | 91,0% | 90,0% |

Tabelle 5.1: Erkennungsgenauigkeiten, die in den Experimenten erzielt wurden

liert werden. Eine Segmentierung bei Verwendung der alternativen Topologie, nämlich der Modellierung von Bildzeilen mit den Metazuständen, würde eine dem Piktogramm zugeordnete Fläche entstehen lassen, die aus gegeneinander verschobenen Zeilen aufgebaut ist. Dieser Aufbau aus vertikalen oder horizontalen Streifen ist eine unmittelbare Folge aus der Vernachlässigung von Abhängigkeiten bei den pseudo zweidimensionalen Modellen.

Wie bereits erwähnt wurde, liefert der Viterbi-Algorithmus neben der wahrscheinlichsten Merkmal-Zustandszuordnung auch einen Schätzwert für die Produktionswahrscheinlichkeit. Aufgrund dieser Wahrscheinlichkeiten, daß das Gesamtbild von dem Hidden-Markov-Modell generiert wurde, erfolgt die Klassifikation. Diese Klassifikationsergebnisse sind in Tabelle 5.1 zusammengefaßt. Die beiden Zeilen der Tabelle geben die Ergebnisse jeweils für unterschiedliche Orientierungen der P2DHMMs an. Die obere Zeile entspricht dabei der Modellierung von Bildspalten mit den Metazuständen und in der unteren Zeile modellieren die Metazustände die Bildzeilen. Dies ist in Tab. 5.1 durch die Verwendung entsprechend orientierter Modelle illustriert. Die ersten drei Spalten in der Tabelle zeigen getrennt nach dem verwendeten Hintergrund die Ergebnisse für die erste vorgestellte Methode (siehe auch Abb. 5.6). Die Erkennungsergebnisse für die adaptive Methode sind in der vierten Spalte angegeben. Bei dieser Methode wurden alle drei Hintergründe gemeinsam präsentiert.

### 5.3.6 Retrieval von Formen in technischen Zeichnungen

In Kapitel 3 konnten Methoden, die mit Hilfe von Piktogrammdatenbasen evaluiert wurden, erfolgreich auf praxisrelevantere Aufgaben, wie etwa der inhaltsbasierten Abfrage von Bilddatenbasen (Kapitel 3.5), übertragen werden. Auf ähnliche Weise können die Algorithmen der Kapitel 5.3.1 bis 5.3.4 auf natürliche Bilder übertragen werden. Anwendungsbeispiele, bei denen natürliche Bilder verarbeitet werden, sind z.B. das Erkennen von Personen in komplexen (Straßen-)Szenen, sowie das Auffinden von Objektformen in technischen Zeichnungen. Die letztgenannte Aufgabe wird in den folgenden Abschnitten behandelt.

Es wurde in Kapitel 3.5 bereits erwähnt, daß inhaltsbasierte Abfragen von Bilddatenbasen eine erhebliche Reduktion des Indexierungsaufwandes ermöglichen und ferner dem Benutzer eines solchen Systems eine intuitive Benutzerschnittstelle zur Verfügung stellen. In diesem Kapitel wird ein experimentelles Datenbanksystem vorgestellt, das es ermöglicht, technische Zeichnungen inhaltsbasiert abfragen zu können. Durch den Einsatz der pseudo zweidimensionalen Markov-Modelle mit Umgebungsmodell können mittels einer Skizze spezifizierte Details in den technischen Zeichnungen gefunden und lokalisiert werden. Eine solche skizzenbasierte Anfrage könnte etwa der folgenden textuellen Beschreibung entsprechen: *Zeige alle technischen Zeichnungen, die eine oder mehrere Schrauben enthalten.* Bei der textuellen Beschreibung fehlt jedoch noch eine Angabe über die spezielle Form der gesuchten Schraube. Dies ist mit Hilfe einer Skizze sehr viel intuitiver zu spezifizieren. Abb. 5.9 zeigt eine Anfrageskizze und die dazu passenden technischen Zeichnungen aus der verwen-

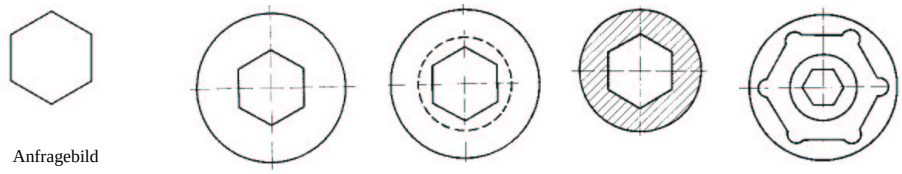


Abbildung 5.9: Anfrageskizze und vier Elemente aus der verwendeten Datenbasis. Die technischen Zeichnungen wurden [Bod88] entnommen.

deten Datenbasis. Ein solches Abfrageergebnis ist sehr schwer zu erzielen, da das gesuchte Objekt (eine hexagonale Form) mit anderen Objekten in der technischen Zeichnung verbunden ist und auch in seiner Größe stark variiert. Da diese Aufgabe sehr ähnlich zu der in Kapitel 5.3.2 vorgestellten Aufgabe ist, nämlich der Klassifikation von Piktogrammen in komplexen Umgebungen, ist es offensichtlich, daß die vorgestellte Modellierung mit zweidimensionalen Markov-Modellen hier einen Lösungsansatz darstellt. Diese Methode unterscheidet sich von konventionellen mehrstufigen Ansätzen, die zunächst Bildelemente wie beispielsweise Linien, Kreisbögen und Schnittpunkte bestimmen und anschließend eine darauf aufbauende Bildinterpretation durchführen (siehe z.B. [Abl97] und [Kas90]). Solch ein mehrstufiger Ansatz ist jedoch immer dann schwer anzuwenden, wenn es nicht möglich ist, die Zeichnung in einfache Elemente zu unterteilen. Genau dieses Problem tritt jedoch bei den

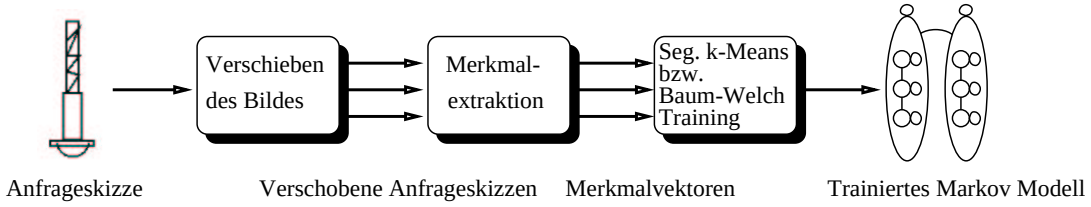


Abbildung 5.10: Training eines pseudo zweidimensionalen Modells auf dem Anfragebild

technischen Zeichnungen auf, da hier Bildelemente oft durch Schraffuren (vgl. Abb. 5.9) verbunden sind und somit beispielsweise eine Zerlegung in nichtverbundene Einzelkomponenten (engl. Connected Component Analysis, siehe auch [Gon92]) scheitert. Aus diesem Grund wird in diesem Kapitel der integrierte Segmentierungs- und Klassifikationsansatz verwendet. Grundsätzlich kommt die gleiche Modellierung, wie sie in Unterkapitel 5.3.1 beschrieben wurde, zum Einsatz.

Da das P2DHMM in Abb. 5.4 das gesuchte Objekt in einem unbekannten Kontext modelliert, muß mit dem Anfragebild ein entsprechendes Objekt-Modell trainiert werden. Hierfür steht lediglich ein einziges Bild zur Verfügung. Aus diesem Grund wird das Anfragebild um einige Bildpunkte in jede Richtung verschoben, um so künstlich die Anzahl an Trainingsbildern zu erhöhen und eine robustere Modellierung zu erreichen. Dies ist in Abb. 5.10 illustriert. Die Umgebung, in die das gesuchte Objekt eingebettet ist, wird mit Umgebungszuständen modelliert, deren Ausgabeverteilungen auf den einzelnen Datenbankelementen ermittelt wurden. Das mit den Umgebungszuständen erweiterte Modell ist, wie in Abb. 5.4 dargestellt, aufgebaut. Abbildung 5.11 zeigt die Bestimmung der Parameter der Wahrscheinlichkeitsdichten der Umgebungszustände. Jedes Bild der Datenbank wird durch eine Merkmalsequenz und deren modellierte Verteilung repräsentiert. Sobald ein Anfragebild vorliegt, wird ein Objektmodell trainiert (siehe Abb. 5.10) und dieses Modell mit Umgebungszuständen erweitert. Anschließend werden sukzessive für jedes Datenbankelement die Produktionswahrscheinlichkeiten mit dem Viterbi-Algorithmus bestimmt. Dabei werden die jeweiligen Merkmalsequenzen und Wahrscheinlichkeitsverteilungen für die Umgebungszustände (siehe Abb. 5.10) verwendet.

Die wesentlichen Schritte bei dem Verfahren zum Auffinden des Anfrageobjekts in Bilddatenbanken mit pseudo zweidimensionalen Modellen werden im folgenden zusammengefaßt:

Schritt 1: Merkmalextraktion und Parameterbestimmung der Umgebungszustände für jedes Bild der Datenbank (Abb. 5.11)

Schritt 2: Präsentation eines Anfragebildes und Training eines Hidden-Markov-Modells für dieses Bild (Abb. 5.10)

Schritt 3: Erweiterung des in Schritt 2 bestimmten P2DHMM mit den Umgebungszuständen (vgl. Abb. 5.4)

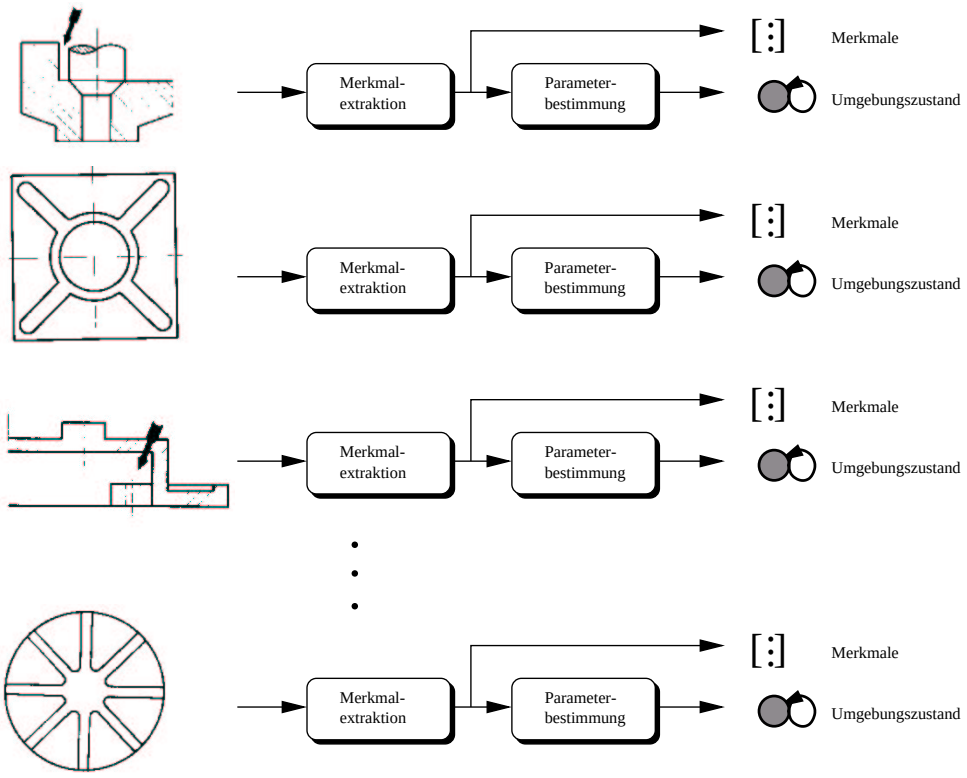


Abbildung 5.11: Bearbeitung der Datenbankelemente

Schritt 4: Für jedes Datenbankelement wird mit dem Viterbi-Algorithmus die Merkmals-Modellzustandszuordnung und die Produktionswahrscheinlichkeit berechnet. Dabei werden sowohl die in Schritt 1 berechneten Merkmale als auch deren ebenfalls schon berechneten Wahrscheinlichkeitsverteilungen verwendet. Die grundlegende Modellstruktur nach Abb. 5.4 bleibt jedoch bestehen.

Schritt 5: Unter Verwendung der in Schritt 4 bestimmten Segmentierungen werden die Produktionswahrscheinlichkeiten neu berechnet. Diese Neuberechnung erfolgt ausschließlich auf den Merkmalen, die den Objektzuständen zugeordnet wurden.

Schritt 6: Ordnen der Datenbankelemente nach den neuberechneten Produktionswahrscheinlichkeiten und Anzeigen der Bilder auf den höchsten Rängen.

Schritt 7: Zurück zu Schritt 2 und präsentieren der nächsten Anfrage.

Zu Schritt 2 ist anzumerken, daß im Gegensatz zu dem in Kapitel 3.5 dargestellten Retrievalverfahren das Hidden-Markov-Modell mit dem Anfragebild trainiert wird. Bei dem Verfahren in Kapitel 3.5 wird eine Merkmalsequenz auf dem Anfragebild berechnet und die Datenbankelemente durch Markov-Modelle repräsentiert. Eine weitere Anmerkung betrifft Schritt 1 in obiger Aufzählung: Dieser Schritt muß lediglich einmal ausgeführt werden, solange die Datenbank nicht verändert wird. Falls eine technische Zeichnung der Datenbank

zugefügt wird, muß die Merkmalextraktion und die Bestimmung der Wahrscheinlichkeitsverteilung nur für dieses neue Element durchgeführt werden. Die verwendete Datenbank besteht aus 56 technischen Zeichnungen, die mit Hilfe eines Scanners digitalisiert wurden und alle dem Konstruktionsatlas von Bode [Bod88] entstammen. Die Bilder wurden mit 300dpi eingescanned und variieren in der Größe zwischen $180 \times 134$ und $256 \times 240$ Bildpunkten. Die Bilder werden direkt, d.h. ohne Verwendung von typischen Vorverarbeitungsschritten wie z.B. dem Entfernen isolierter Bildpunkte (Despecle) oder einer Liniendickennormalisierung (Linethinning) verarbeitet.

Testdurchläufe mit dem experimentellen System wurden mit zehn verschiedenen Anfrageskizzen, die mit dem verbreiteten UNIX-Programm *xfig* erstellt wurden, durchgeführt. Drei dieser Anfragen sind zusammen mit den vom System zurückgelieferten technischen Zeichnungen in den Abb. 5.12 bis 5.14 gezeigt. Die Anfrageskizzen sind in den Abbildungen jeweils im linken oberen Teil dargestellt. Es sind die ähnlichsten fünf Elemente der Datenbank, geordnet nach ihrem Ähnlichkeitsmaß von links nach rechts und oben nach unten, dargestellt. Bemerkenswert sind die guten Abfrageergebnisse z.B. in Abb. 5.12, wo mit der groben Skizze einer Schraube zwei technische Zeichnungen gefunden werden konnten, die eine solche Schraube enthalten und dies obwohl beide Objekte in Schraffuren eingebettet sind. Weitere qualitative Ergebnisse, die mit dem in diesem Kapitel dargestellten System erzielt wurden, sind in den Arbeiten [Mul99e] und [Mul00b] zu finden. Mit einem algorithmisch identischen Verfahren, jedoch in dem Anwendungskontext der Kunstarchive wurde ein System zum Auffinden von Wasserzeichen in [Mul99g] vorgestellt.

In den Abb. 5.12 bis 5.14 sind die Rangfolgen, geordnet nach den berechneten Produktionswahrscheinlichkeiten angegeben. Es wurden während der Anwendung des Viterbi-Algorithmus jedoch auch die gesuchten Objekte in den Zeichnungen lokalisiert. Eine solche Lokalisierung, die durch die Zuordnung der Merkmale zu den Modellzuständen ermittelt wird, ist exemplarisch in Abb. 5.15, für die skizzierte Schraube und das Datenbankelement mit dem höchsten Ähnlichkeitsmaß (vgl. Abb. 5.12), angegeben. Das Raster, das in Abb. 5.15 über die technische Zeichnung gelegt wurde, visualisiert die einzelnen Blöcke, auf denen die Merkmalvektoren berechnet werden. Die Merkmale, die den grau schattierten Flächen entnommen wurden, sind durch den Viterbi-Algorithmus den Umgebungszuständen zugeordnet worden (vgl. Abb. 5.4), wohingegen die weiß dargestellten Flächen zu Merkmalen gehören, die dem Objektmodell (Anfragemodell) zugeordnet wurden. Obwohl die dargestellte Segmentierung kein optimales Ergebnis darstellt, wird sich dennoch durch die Zuordnung der weiß dargestellten Merkmale zu den Objektzuständen eine hohe Produktionswahrscheinlichkeit ergeben.

Die im Rahmen dieser Arbeit entwickelte Methode, nämlich die integrierte Segmentierung und Klassifikation mit P2DHMMs zeigt gute Ergebnisse auf dem Gebiet des Retrievals von technischen Zeichnungen. Um die Leistungsfähigkeit dieses Ansatzes weiter testen zu können, wurde nach einer anspruchsvollen und gleichzeitig praxisrelevanten Anwendung

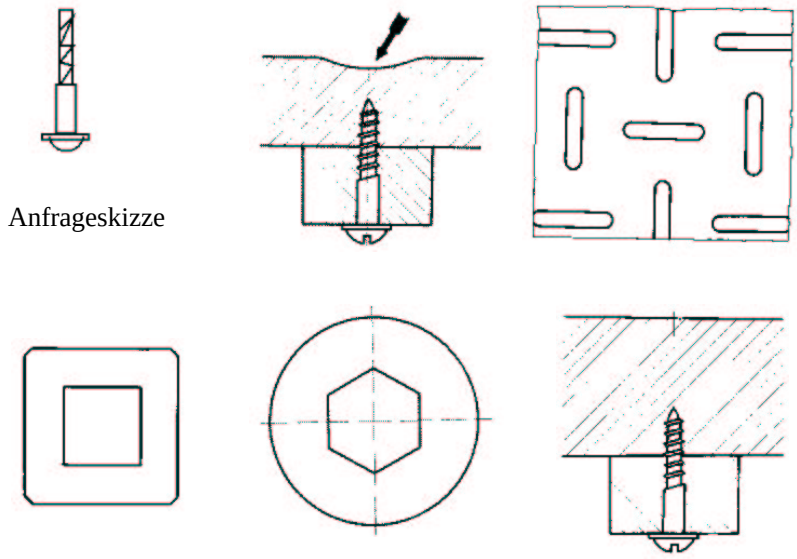


Abbildung 5.12: Anfrageskizze und fünf zurückgelieferte technische Zeichnungen (1)

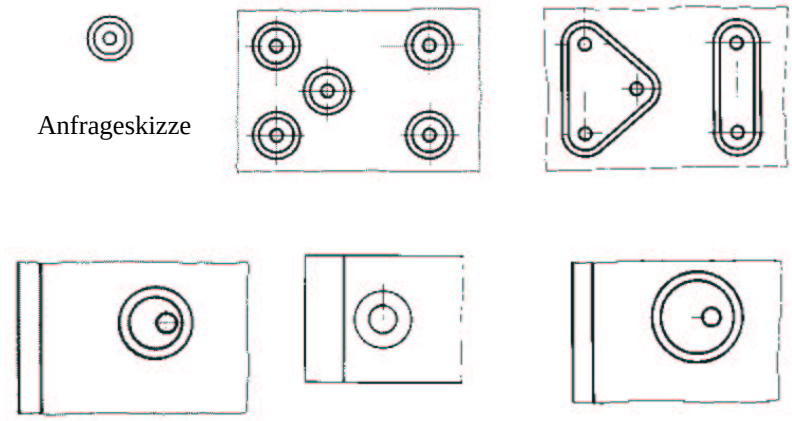


Abbildung 5.13: Anfrageskizze und fünf zurückgelieferte technische Zeichnungen (2)

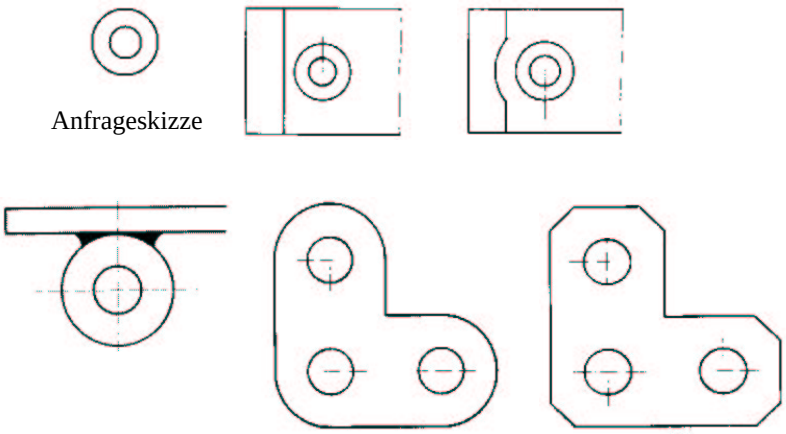


Abbildung 5.14: Anfrageskizze und fünf zurückgelieferte technische Zeichnungen (3)

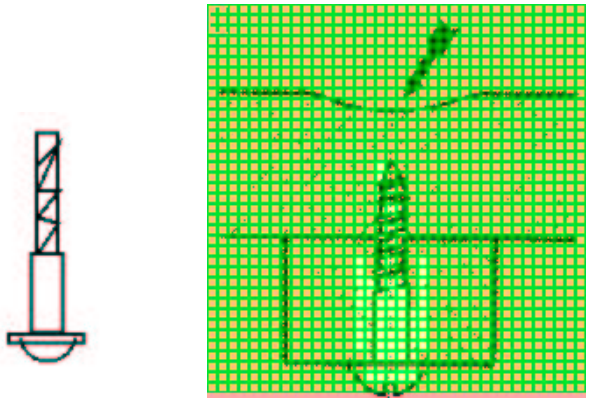

Abbildung 5.15: Anfrageskizze und zugeordnete Segmentierung des Datenbankelements

gesucht, die nicht ausschließlich binäre Bilder, wie technische Zeichnungen verwendet. Eine solche Aufgabe wurde in Form des im folgenden Kapitel beschriebenen Personen-Trackings gefunden.

## 5.4 Tracking von Personen

Das Tracking von Personen stellt ein sehr interessantes Anwendungsszenario für die P2DHMMs mit Umgebungsmodell dar. So kann untersucht werden, ob die Methode auch auf Grauwertbilder bzw. Farbbilder übertragbar ist. Zudem kann überprüft werden, ob die Erkennung von Personen in Einzelbildern in geeigneter Weise mit einem Algorithmus kombiniert werden kann, mit dem ganze Bildsequenzen bearbeiten werden können. Bevor das Personen-Tracking mit P2DHMM ausführlich beschrieben wird, wird zunächst kurz in dieses Anwendungsszenario eingeführt.

Das Tracking von Personen kann als eine Basistechnik angesehen werden, die eine Vielzahl von Applikationen, insbesondere in der Mensch-Maschine-Kommunikation und im Multimediabereich ermöglicht. Bei der Mensch-Maschine-Kommunikation ist die Information von großer Relevanz, ob sich eine oder mehrere Personen in einer definierten Umgebung befinden. Zusätzlich muß oft herausgefunden werden, wer die Personen sind, welche Handlungen diese ausführen und wo die genauen Positionen der Personen sind. Das Personen-Tracking löst dabei zunächst die folgenden beiden Unteraufgaben: Die Anwesenheit von Personen wird festgestellt und deren Positionen werden über eine Sequenz von Bildern verfolgt. Die Kenntnis der Position der Personen kann genutzt werden, um in einem weiteren Schritt die Personen zu identifizieren. Dies kann durch das Ausschneiden des Gesichts und durch die Verwendung der in Unterkapitel 5.1 vorgestellten Klassifikationsmethoden erfolgen. Weiterhin gibt die Position, bzw. die Trajektorie einer Person einen wichtigen Hinweis auf ausgeführte Aktionen oder Gesten.

Die Analyse und Erkennung von menschlichen Handlungen in Bildsequenzen wird auch bei einer kommerziell sehr bedeutenden Anwendung, nämlich der automatischen Überwachung benötigt. Beispiele für Überwachungsszenarien schließen Tankstellen, Kaufhäuser, Geldautomaten oder Verkehrskreuzungen ein.

Im Bereich der Multimediaanwendungen kann das Personen-Tracking einen wichtigen Beitrag zum automatischen Indexieren von Videoaufnahmen liefern. Insbesondere bei Sportaufnahmen stellt die Position der Spieler eine wichtige Information dar. Läuft beispielsweise ein Tennisspieler in die Nähe des Netzes, so erfolgt sehr wahrscheinlich ein Angriff und es wird im folgenden mit hoher Wahrscheinlichkeit ein Volley geschlagen. Eine ähnlich wichtige Information stellt die Position der Spieler beim Fußball dar, da auch hier aus der Verteilung der Spieler auf dem Feld die jeweilige Spielsituation ermittelt werden kann. Ein umfangreicher und aktueller Übersichtsartikel zu dem Thema der menschlichen Aktionsanalyse, die das Personen-Tracking einschließt, ist von Gavrila in der Arbeit [Gav99] zusammengestellt worden. In [Gav99] findet sich eine Vielzahl weiterer Einsatzmöglichkeiten des Personen-Trackings und eine Vorstellung der am häufigsten eingesetzten Verfahren.

Wie schon erwähnt wurde, teilt sich das Personen-Tracking in die beiden Teilaufgaben, nämlich die Anwesenheit von Personen zu detektieren und die Dynamik der Bewegung zu erfassen. Das folgende Unterkapitel befaßt sich mit der ersten genannten Teilaufgabe, nämlich dem Auffinden von Personen in natürlichen Bildern. Die hierbei eingesetzten Methoden basieren auf der in Kapitel 5.3 eingeführten Modellierung mit pseudo zweidimensionalen Hidden-Markov-Modellen und Umgebungszuständen.

### 5.4.1 Auffinden von Personen in natürlichen Bildern mit pseudo zweidimensionalen Hidden-Markov-Modellen

Das Auffinden von Personen in Bildern stellt aufgrund der großen Formvariation eine anspruchsvolle Aufgabe dar. Dies kann anhand von Abb. 5.16 erläutert werden. In der Abbildung sind zwei Umrisse derselben sich bewegenden Person gezeigt. Die beiden Umrisse haben sehr unterschiedliche Formen und bei einer weiteren Analyse der Bewegung dieser Person werden zusätzliche Variationen in der Form auftreten. Das Auffinden von Personen

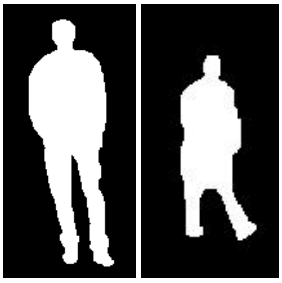

Abbildung 5.16: Umrisse einer sich bewegenden Person

Abbildung 5.17: Beispiele für das Auffinden von Personen in komplexen Umgebungen

ist somit erheblich anspruchsvoller, als das Auffinden von starren Objekten, wie beispielsweise von Fahrzeugen oder das Auffinden von Gesichtern. Rowley beschreibt in [Row96] eine auf künstlichen neuronalen Netzen basierende Methode, mit der frontale Ansichten von Gesichtern sehr zuverlässig in Bildern gefunden werden können. Aufgrund der großen Formvariationen bietet sich diese Methode jedoch nicht zum Auffinden von Personen an. Stattdessen kann die schon beschriebene Methode, die auf P2DHMMs in Kombination mit Umgebungszuständen basiert, verwendet werden, da sie die folgenden Vorteile bietet:

- Die elastischen Modellierungseigenschaften der Hidden-Markov-Modelle passen sehr gut zu den stark variierenden Personenabbildungen.
- Die P2DHMMs ermöglichen eine elastische Modellierung sowohl in vertikaler, als auch in horizontaler Richtung.
- Durch den Viterbi-Algorithmus und die Merkmal-Zustandszuordnung kann eine integrierte Segmentierung und Klassifikation erfolgen. Dies ermöglicht es, nicht nur die Personen aufzufinden, sondern diese im gleichen Schritt auch zu erkennen.
- Da einem Modellzustand unterschiedlich viele Merkmale zugeordnet werden können, ist es möglich, verschieden große zweidimensionale Muster mit einem P2DHMM zu verarbeiten. Dies ist ein Vorteil bei unterschiedlich großen Abbildungen von Personen, die sich aus der Aufnahmesituation oder durch verschieden große Personen ergeben können. Ein auf KNNs basierendes System, wie etwa das in [Row96] beschriebene, ist stets auf die Analyse einer festen Mustergröße beschränkt.
- Es stehen effiziente Algorithmen für das Training und die Klassifikation zur Verfügung.

Die Abb. 5.17 zeigt Ergebnisse, die mit dem im Rahmen dieser Arbeit entwickelten Verfah-

ren erzielt wurden. Die auf der linken Seite in Abb. 5.17 dargestellte Abbildung einer Person wurde zusammen mit anderen, ähnlichen Abbildungen zum Training eines P2DHMMs verwendet. Dabei wurde eine blockweise Merkmalextraktion zusammen mit einer diskreten Cosinus-Transformation verwendet (siehe Gleichung 5.1 und Abb. 4.3). Anschließend wurde das in Kapitel 5.3.2 vorgestellte, adaptive Parameterschätzungsverfahren für die Umgebungszustände angewendet. Für die beiden Testbilder in Abb. 5.17 wurden jeweils unter Verwendung aller Merkmale Verteilungen bestimmt, die als Ausgabeverteilungen der Umgebungszustände verwendet werden (siehe auch Abb. 5.7). Anschließend wird mit dem Viterbi-Algorithmus die Merkmal-Zustandszuordnung bestimmt und somit die Aufteilung der Merkmale in Umgebung und Person ermittelt. In der Abb. 5.17 ist für die beiden Testbilder die so ermittelte Grenze zwischen der Person und der Umgebung eingezeichnet. Der Abbildung kann entnommen werden, daß bei Verwendung der P2DHMMs mit Umgebungszuständen die Person in komplexen Szenen lokalisiert werden kann.

Neben der adaptiven Methode zur Bestimmung der Verteilung der Umgebungszustände kann auch die nicht-adaptive Methode bei der Personendetektion verwendet werden (siehe Abb. 5.6). Diese bietet Vorteile bei fest installierten Kameras und somit z.B. bei der Überwachung von Tankstellen, da bedingt durch den statischen Hintergrund die Verteilungen der Umgebungszustände sehr zuverlässig geschätzt werden können. Zudem steht oft eine große Anzahl von Trainingsbildern zur Verfügung, bei denen sich keine Person im Bild befindet und somit *ausschließlich* der Hintergrund eintrainiert wird. Es sei jedoch an dieser Stelle angemerkt, daß konventionelle Verfahren, die auf der Subtraktion des Bildhintergrundes basieren, bei fest installierten Kameras geeigneter sind.

Die vorgestellte Personendetektion arbeitet auf einzelnen Bildern und berücksichtigt somit ausschließlich statische Informationen. Um die Dynamik der Bewegung einer Person modellieren zu können, ist somit ein weiterer Verarbeitungsschritt, der Bildsequenzen verwendet, erforderlich. Dies erfolgt mit dem im folgenden betrachteten sog. Kalman-Filter.

### 5.4.2 Kalman-Filter

Der Kalman-Filter-Algorithmus wurde 1960 von R.E. Kalman in der Arbeit [Kal60] veröffentlicht und stellt eine rekursive Lösung der folgenden Aufgabe dar. Gegeben sei ein diskreter, zeitabhängiger Prozeß, der durch die folgende lineare Differenzengleichung beschrieben wird:

$$\vec{x}_{k+1} = A \cdot \vec{x}_k + \vec{w} \tag{5.3}$$

Dabei ist $\vec{x}_k$ ein Zustandsvektor zum Zeitpunkt $t_k$, $A$ eine Übergangsmatrix und $\vec{w}$ ein Vektor mit Zufallsvariablen, der das Prozeßrauschen charakterisiert. Es wird angenommen, daß $\vec{w}$ durch eine multivariate Gaußverteilung mit dem Mittelwert 0 und der Kovarianzmatrix $Q$ beschrieben werden kann. Ferner ist die folgende sog. Meßgleichung gegeben, die die nicht

meßbaren Systemzustände $\vec{x}_k$ mit dem beobachtbaren Meßvektor $\vec{z}_k$ verknüpft:

$$\vec{z}_k = H_k \cdot \vec{x}_k + \vec{v} \tag{5.4}$$

Die Matrix $H$ gibt die Beziehung zwischen dem Zustandsvektor $\vec{x}_k$ und dem Vektor der meßbaren Größen $\vec{z}_k$ an. $\vec{v}$ ist ein Vektor aus Zufallsvariablen, der das Meßrauschen charakterisiert. Der Vektor $\vec{v}$ wird wiederum als eine, durch eine mutivariate Gaußverteilung mit dem Mittelwert 0 charakterisierte Größe, angesehen. Die Kovarianzmatrix der Gaußverteilung sei mit $R$ bezeichnet. Der Kalman-Filter-Algorithmus löst nun bei einem durch die Gleichungen 5.3 und 5.4 beschriebenen System die Aufgabe, die Zustandsvektoren $\vec{\hat{x}}_k$, bei Kenntnis der Meßwerte $\vec{z}_k$, zu schätzen. Im Gegensatz zu der Größe $\vec{x}_k$, die den tatsächlichen Zustand des Systems zum Zeitpunkt $t_k$ darstellt, bezeichnet $\vec{\hat{x}}_k$ den Schätzwert für diesen Zustand bei bekanntem Meßwert $\vec{z}_k$. $\vec{\hat{x}}_k$ wird, da der Meßwert bekannt ist, als *a posteriori* Schätzwert bezeichnet, während $\vec{\hat{x}}_k^-$ den *a priori* Schätzwert, also den Schätzwert für $\vec{x}_k$, der ohne Berücksichtigung der Messung $\vec{z}_k$ ermittelt wird, bezeichnet.

Der Kalman-Filter-Algorithmus stellt eine rekursive Lösung dar, der aus den folgenden drei Verarbeitungsstufen besteht: der Initialisierung, der Vorhersage der Parameter und dem Abgleich mit der Messung (siehe auch [Wel95]). Die beiden letztgenannten Schritte werden aufeinanderfolgend für jeden Zeitschritt wiederholt. Bei der Initialisierung wird der Anfangszustand $\vec{\hat{x}}_0^-$, sowie die Kovarianzmatrix des Fehlers bei der Bestimmung der Zustände festgelegt. Diese Kovarianzmatrix wird mit $P_0^-$ bezeichnet und berechnet sich zu:

$$P_k^- = E\{e_k^- e_k^{-T}\} \quad \text{mit} \quad e_k^- = \vec{x}_k - \vec{\hat{x}}_k^- \tag{5.5}$$

Die Kovarianzmatrix des Fehlers unter Berücksichtigung der Messung $z_k$ ergibt sich zu:

$$P_k = E\{e_k e_k^T\} \quad \text{mit} \quad e_k = \vec{x}_k - \vec{\hat{x}}_k \tag{5.6}$$

Der Schritt der *Vorhersage der Parameter* ermöglicht das zeitliche Vorwärtsprojezieren des aktuellen Zeitpunktes und der aktuellen Kovarianzmatrizen des Fehlers. Das Ergebnis dieses Schritts sind a-priori-Schätzwerte für den folgenden Zeitschritt. Das Kalman-Filter verwendet hierbei die folgenden Gleichungen:

$$\vec{\hat{x}}_{k+1}^- = A \cdot \vec{\hat{x}}_k \tag{5.7}$$

$$P_{k+1}^- = A \cdot P_k \cdot A^T + Q \tag{5.8}$$

Der Schritt der mit *Abgleich mit der Messung* bezeichnet wurde, verwendet eine neue Messung, um zusammen mit der a-priori-Schätzung eine verbesserte a-posteriori-Schätzung zu erhalten. Dabei kommen die folgenden Gleichungen zum Einsatz:

$$K_k = \frac{P_k^- H^T}{H P_k^- - H^T + R} \tag{5.9}$$

$$\vec{\hat{x}}_k = \vec{\hat{x}}_k^- + K_k \cdot \left[\vec{z}_k - H \cdot \vec{\hat{x}}_k^-\right] \tag{5.10}$$

$$P_k^+ = \left[I - K_k \cdot H\right] \cdot P_k^- \tag{5.11}$$

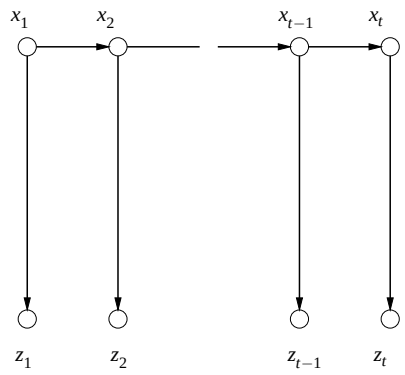


Abbildung 5.18: Darstellung des dem Kalman-Filter zugrundeliegenden Modells als dynamisches Bayes-Netz (aus [Mur00])

In Gleichung 5.11 bezeichnet $I$ die Einheitsmatrix und die Größe $K_k$ die sog. *Kalman-Verstärkung*. Nach der Durchführung der beiden durch die Gleichungen 5.7 und 5.8 bzw. 5.9 bis 5.11 beschriebenen Schritte werden diese für den folgenden Zeitschritt wiederholt. Dabei werden die a-posteriori-Schätzwerte des aktuellen Zustandes für die Vorhersage der a-priori-Schätzwerte des folgenden Zeitschrittes verwendet. Diese rekursive Arbeitsweise des Kalman-Filters hat zu einer weiten Verbreitung des Algorithmus geführt, da sie den Echtzeiteinsatz ermöglicht. Die Gleichungen des Kalman-Filters wurden in diesem Unterkapitel ohne Herleitung angegeben. Diese Herleitungen finden sich z.B. in dem Buch [Gre93].

Das Kalman-Filter stellt nicht nur einen Algorithmus dar, der aus den rauschbehafteten Messungen die tatsächlichen System-Zustände schätzt, bzw. *filtert*, sondern es kann auch als ein Modell für Vorgänge und Prozesse angesehen werden. Dies wird vor allem anhand der Gleichungen 5.3 und 5.4 deutlich. Die Modellvorstellung, die das Kalman-Filter zugrundelegt, kann ähnlich wie bei Hidden-Markov-Modellen basierend auf dynamischen Bayes-Netzen weiter analysiert werden (siehe auch Unterkapitel 2.2.6). Abb. 5.18 stellt das Kalman-Filter als dynamisches Bayes-Netz dar. Bei dem Vergleich zwischen den Abb. 2.4 und 5.18 fällt die große Ähnlichkeit zwischen beiden Bayes-Netzen auf. Dies deutet auf ähnliche Abhängigkeitsbeziehungen hin, die sowohl beim Hidden-Markov-Modell, als auch beim Kalman-Filter vorausgesetzt werden. Diese gemeinsamen Abhängigkeitsbeziehungen können auch den Gleichungen 2.2 und 5.3 entnommen werden, die beide die jeweiligen Modelle als kausal und als beschränkt gedächtnisbehaftet charakterisieren. Ferner liegt bei beiden Modellen eine statistischen Abhängigkeit erster Ordnung der nicht beobachtbaren Variablen vor. Diese nicht beobachtbaren Zufallsvariablen sind beim Kalman-Filter durch die Systemzustände $\vec{x}_k$ und beim Hidden-Markov-Modell durch die Modellzustände $q_t$ gegeben. Der wichtigste Unterschied zwischen beiden Modellen besteht darin, daß alle Knoten des Bayes-Netzes, die den Zufallsvariablen des Modells entsprechen, im Fall des Kalman-Filters kontinuierliche Werte annehmen können. Dies ist beim HMM nicht der Fall, da die Zufallsvariable für die Einnahme eines Modellzustands $q_t$ nur diskrete Wer-

te aus der Menge $\{S_1, \ldots, S_N\}$ annehmen kann. Zusammenfassend läßt sich somit feststellen, daß das dem Kalman-Filter zugrundeliegende Modell ein Hidden-Markov-Modell mit nicht beobachtbaren Zufallsvariablen ist, die beliebige Werte annehmen können (siehe auch [Min96, Smy97, Min99, Mur00]).

### 5.4.3 Interaktion zwischen Kalman-Filter und P2DHMM

Mit dem P2DHMM-basierten Ansatz, der in diesem Kapitel 5 ausführlich beschrieben wurde, ist es möglich, Personen in komplexen Bildszenen aufzufinden (siehe auch Abb. 5.17). Dabei werden ausschließlich Einzelbilder betrachtet. Zusätzlich steht mit dem im vorherigen Unterkapitel vorgestellten Kalman-Filter ein Algorithmus zur Verfügung, der die dynamische Modellierung von Bildsequenzen ermöglicht. Die Kombination beider Methoden erlaubt es somit, Personen in einer Bildsequenz zu verfolgen. Um dies zu erreichen, wird das im folgenden beschriebene dynamische Modell für die Bewegung einer Person angenommen. Der Zustandsvektor $\vec{x}$ sei folgendermaßen gegeben:

$$\vec{x} = (x_s, y_s, v_x, v_y, w, h)^T \tag{5.12}$$

In obiger Gleichung steht das Wertepaar $(x_s, y_s)$ für die Koordinaten des Schwerpunkts einer Person, $(v_x, v_y)$ bezeichnen die horizontale bzw. die vertikale Geschwindigkeit des Schwerpunktes und $(w, h)$ bezeichnen die Breite und Höhe eines die Person umschreibenden Rechtecks. Die Übergangsmatrix $A$ wird folgendermaßen gewählt und stellt zusammen mit Gleichung 5.3 ein einfaches dynamisches Modell dar:

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.13}$$

Unter Verwendung des Segmentierungsergebnisses, welches durch die pseudo zweidimensionalen Hidden-Markov-Modelle erzeugt wird, wird der folgende Meßvektor bestimmt:

$$\vec{z} = (x_s, y_s, w, h)^T \tag{5.14}$$

Dabei steht das Wertepaar $(x_s, y_s)$ für die aus dem Segmentierungsergebnis berechneten Koordinaten des Schwerpunktes der Person. Die Größen $(w, h)$ in Gleichung 5.14 sind die Breite, bzw. die Höhe eines die segmentierte Person umschließenden Rechtecks. Sowohl der Schwerpunkt $(x_s, y_s)$ als auch die Größen $(w, h)$ können z.B. auf der eingerahmten Fläche in Abb. 5.17 berechnet werden. Durch diese Festlegegung des Meßvektors ergibt sich die

folgende Matrix $H$:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.15}$$

Unter Verwendung der P2DHMMs und des Kalman-Filters, sowie des bisher definierten Zustandsvektors, des Meßvektors und der beiden Matrizen $A$ und $H$, können Personen in Bildsequenzen verfolgt werden. Es ist jedoch zweckmäßig, zusätzlich eine Interaktion zwischen dem Markov-Modell und dem Kalman-Filter einzuführen. Dieser Schritt erhöht die Robustheit des Tracking-Verfahrens und funktioniert auf folgende Weise: Bei der Bestimmung der Segmentierung mit dem P2DHMM wird die zu verfolgende Person nicht im gesamten Bild gesucht, sondern lediglich in dem durch das Kalman-Filter vorhergesagten Bereich. Dieser Bereich ist durch die Komponenten des a-priori Schätzwertes $\vec{\hat{x}}_{k+1}^{-}$ gegeben. Zusätzlich wird eine Erweiterung des Suchbereiches um den Faktor 1,5 vorgenommen, da somit sichergestellt wird, daß es genügend Merkmale gibt, die den Umgebungszuständen zugeordnet werden können. Unter Verwendung der aktuellen Meßwerte, die dem Segmentierungsergebnis entnommen werden, wird der a-posteriori-Schätzwert $\vec{\hat{x}}_k$ für die Systemzustände ermittelt. Die Komponenten $x_s$ und $y_s$ des Vektors $\vec{\hat{x}}_k$, die den Schwerpunkt der Person angeben, werden verwendet, um die Position der Person in den einzelnen Bildern einer Sequenz zu bestimmen, bzw. zu markieren. Dies wird an einer Beispielsequenz im folgenden Unterkapitel gezeigt. Vorher werden jedoch die wesentlichen Schritte bei dem Trackingverfahren mit P2DHMM und Kalman-Filter im folgenden zusammenfassend dargestellt.

Das Tracking-Verfahren beginnt mit dem manuellen Ausschneiden einer rechteckigen Region aus dem ersten Bild einer Sequenz. Diese Bildregion soll die zu trackende Person möglichst vollständig enthalten. Das mit Beispielbildern einer großen Zahl von Personen vortrainierte P2DHMM wird an die zu trackende Person unter Verwendung eines Trainingsschrittes adaptiert. Anschließend wird mit dem Viterbi-Algorithmus die Merkmal-Zustandszuordnung bestimmt und somit die Modellzustände identifiziert, die die Merkmale der Person bzw. den Bildhintergrund modellieren. Schließlich werden sukzessive die folgenden Einzelschritte durchgeführt:

Schritt 1: Mit dem Viterbi-Algorithmus und unter Verwendung des adaptierten P2DHMMs wird ein Bild der Sequenz in eine Personenregion und in einen Bildhintergrund segmentiert (vgl. Abb. 5.17).

Schritt 2: Ausgehend von dieser Segmentierung werden die Komponenten des Meßvektors bestimmt. Die Koordinaten des Schwerpunktes der Person $(x_s, y_s)$ werden als Flächenschwerpunkt der der Person zugeordneten Bildregion bestimmt. Anschließend wird ein die Personenregion umschreibendes Rechteck bestimmt, dessen Breite bzw. Höhe das Wertepaar $(w, h)$ bestimmt.

Schritt 3: Der Meßvektor $\vec{z} = (x_s, y_s, w, h)^T$ wird dem Kalman-Filter zugeführt. Das Kalman-Filter führt den Schritt *Abgleich mit der Messung* aus. Hierzu werden die Gleichungen 5.9 bis 5.11 verwendet.

Schritt 4: Die a-priori-Schätzwerte für den folgenden Zeitschritt $\vec{\hat{x}}_{k+1}^{-}$, die aus dem Schritt *Vorhersage der Parameter* stammen, werden verwendet, um den Suchbereich für das folgende Bild der Sequenz zu bestimmen (Gleichungen 5.7 und 5.8). Dieser Suchbereich wird um den Faktor 1,5 erweitert.

Schritt 5: Die Segmentierung des folgenden Bildes der Sequenz in einen Hintergrund und eine Personenregion wird auf diesen Suchbereich eingeschränkt.

Schritt 7: Zurück zu Schritt 1 und bearbeiten des folgenden Einzelbildes.

### 5.4.4 Experimentelle Ergebnisse

Abb. 5.19 zeigt einen Ausschnitt aus einer Sequenz, die eine sich bewegende Person zeigt. Zusätzlich sind die Ergebnisse des Personen-Trackings durch die blauen Rechtecke visualisiert. Diese Rechtecke repräsentieren die Komponenten $x_s, y_s$ und $w, h$ des a-posteriori Schätzwertes für den Zustandsvektor $\vec{\hat{x}}_k$. Die Komponenten des dabei verwendeten Meßvektors $\vec{z}$ werden aus dem Segmentierungsergebnis berechnet, das mit einem P2DHMM, das aus $6 \times 6$ Zuständen besteht, erzeugt wird. Das Training des Modells erfolgt mit handsegmentierten Abbildungen der gleichen Person, die aus einer anderen Sequenz stammen. Das gute Trackingergebnis wurde trotz der teilweisen Verdeckung der Person durch Stuhlreihen und Tische erzielt. Die starke Schattenbildung in Teilen der Sequenz stellt für konventionelle Verfahren ein großes Problem dar, da z.B. Differenzbildmethoden, die in [Jan97] verwendet wurden, auch in dem Schattenbereich die Anwesenheit einer Person signalisieren würden. Das hier verwendete Markov-Modell, das unter Verwendung von Trainingsdaten erstellt wurde, lokalisiert die Person jedoch auf robuste Weise. Weitere Ergebnisse, die mit dem vorgestellten Verfahren erzielt wurden, sind in [Rig99b, Rig99a] zu finden. In diesen Arbeiten sind auch Sequenzen zu finden, die mit einer bewegten Kamera aufgenommen wurden.

Abbildung 5.19: Ausschnitt aus einer Bildsequenz, die eine sich bewegende Person zeigt. Die blau eingezeichneten Rechtecke visualisieren das Trackingergebnis (aus [Win98]).

## 5.5 Kapitelzusammenfassung

Ein im Rahmen dieser Arbeit entwickelter Ansatz, der die integrierte Segmentierung und Klassifikation von in komplexen Umgebungen eingebetteten Mustern ermöglicht, wurde vorgestellt. Der Ansatz verwendet pseudo zweidimensionale Hidden-Markov-Modelle in Kombination mit an die Umgebung angepaßten Umgebungszuständen. Nach der Anwendung des Viterbi-Algorithmus liegt eine Merkmal-Zustandszuordnung, die als Segmentierung interpretiert werden kann, sowie ein Schätzwert für die Produktionswahrscheinlichkeit vor. Unter Verwendung dieses Schätzwertes erfolgt die Musterklassifikation.

Die Anpassung der Parameter der Umgebungszustände kann auf verschiedene Weisen erfolgen, je nachdem, ob Vorwissen über die zu analysierende Szene vorliegt oder nicht. Für den letztgenannten Fall wurde ein Verfahren entwickelt, bei dem die Parameter der Umgebungszustände auf allen Merkmalen des zu analysierenden Bildes bestimmt wurden.

Das Verfahren wurde zunächst mit Hilfe einer Piktogrammdatenbasis, die aus 20 Klassen besteht, evaluiert. Weitere Experimente sind beschrieben, die die Eignung des P2DHMM-Ansatzes für das Retrieval von Formen in technischen Zeichnungen belegen. Somit ist es möglich, z.B. eine mit einer Skizze spezifizierte Schraube in komplexen technischen Zeichnungen aufzufinden. Schließlich wurde der P2DHMM-Ansatz für das Personen-Tracking eingesetzt. Dabei konnte gezeigt werden, daß Muster bzw. Personen auch in Grauwertbildern bzw. Farbbildern mit dem vorgestellten Ansatz gefunden werden können. Es zeigte sich ebenfalls, daß der Ansatz gut kombinierbar ist mit einem sog. Kalman-Filter, das die Dynamik der Bewegung der Person modelliert.

# Kapitel 6

# Neuartige statistische Modellierung für die Klassifikation von Bildsequenzen

Dieses Kapitel beschreibt die Bildsequenzerkennung mit neuartigen pseudo dreidimensionalen Hidden-Markov-Modellen. Anders als im vorhergehenden Kapitel 5 soll hier die Bildfolge in ihrer Gesamtheit analysiert und schließlich klassifiziert werden. Der hierarchische Ansatz des Kapitels 5, der P2DHMMs in Kombination mit einem Kalman-Filter verwendet, dient hingegen lediglich dem Tracking von Personen und nicht der Klassifikation der Bewegungen. Genau dies, nämlich die Klassifikation von menschlichen Aktionen, bzw. Gesten stellt das Anwendungsszenario für die in diesem Kapitel beschriebenen pseudo dreidimensionalen Modelle dar.

Auf Markov-Random-Fields mit einer Nachbarschaftsbeziehung, die die drei Dimensionen einer Bildsequenz $(x,y,t)$ berücksichtigt, sowie auf daraus abgeleiteten *echten* dreidimensionalen Hidden-Markov-Modellen wird in diesem Kapitel nicht ausführlich eingegangen. Die Theorie der MRFs ist allgemein formuliert und schließt den dreidimensionalen Fall ein (siehe [Li95]). Durch die Einführung einer kausalen Nachbarschaftsbeziehung und eines zweiten statistischen Prozesses, der zur Ausgabe von Merkmalen führt, geht das dreidimensionale Hidden-Markov-Modell aus dem Markov-Random-Field hervor (vgl. Kapitel 4.1 und 4.2). Beim dreidimensionalen HMM werden somit Zustandsübergänge mit folgender Definition verwendet:

$$A_{ijk,lmn,opq,uvw} = P(q_{(x,y,t)} = S_{(u,v,w)} | q_{(x-1,y,t)} = S_{(i,j,k)}, q_{(x,y-1,t)} = S_{(l,m,n)}, q_{(x,y,t-1)} = S_{(o,p,q)}) \tag{6.1}$$

Obwohl mit den dreidimensionalen Hidden-Markov-Modellen eine theoretisch sehr geeignete Modellierungsmethode zur Verfügung steht, ist es wiederum sehr problematisch, daß keine effizienten Trainings- und Klassifizierungsalgorithmen bekannt sind. Somit wird wie im zweidimensionalen Fall auf eine Modellierung, die Musterverzerrungen in allen Dimensionen gemeinsam betrachtet, verzichtet (siehe auch Kapitel 4.2). Dies führt zu der Einführung der *pseudo* dreidimensionalen Modelle. Durch den Verzicht auf die in Gleichung 6.1

vorgestellten Modellgrößen können für diese Modelle effiziente Algorithmen für das Training und die Erkennung gefunden werden. Die P3DHMMs werden im folgenden ausführlich vorgestellt.

## 6.1 Pseudo dreidimensionale Hidden-Markov-Modelle

Pseudo dreidimensionale Hidden-Markov-Modelle wurden im Rahmen dieser Arbeit entwickelt und sind erstmalig in den Arbeiten [Mul99c] und [Mul00a] erwähnt und verwendet worden. Die Bezeichnung *pseudo dreidimensionales Hidden-Markov-Modell* wurde vom Autor gewählt, da es sich um einen leicht zu merkenden Begriff handelt und zudem die methodische Verwandtschaft zu dem pseudo zweidimensionalen HMM betont wird. Es sei an dieser Stelle jedoch darauf hingewiesen, daß der Begriff P3DHMM die Existenz ähnlicher statistischer Vereinfachungen impliziert, wie dies bei P2DHMMs der Fall ist. Dies trifft jedoch nicht zu, denn die bei den P3DHMMs gemachten Vereinfachungen sind schwerwiegender als bei den P2DMMs. Bei den P3DHMMs wird neben dem Verzicht auf eine Modellierung, die Musterverzerrungen in beiden Bilddimensionen gemeinsam betrachtet, auch eine Unabhängigkeit zeitlich benachbarter Bildpunkte angenommen. Die statistische Modellierung besser beschreibende Bezeichnungen wären somit *pseudo pseudo dreidimensionales HMM* oder *doppelt pseudo dreidimensionales HMM*. Es ist offensichtlich, daß diese Bezeichnungen aufgrund ihrer Länge ungeeignet sind.

### 6.1.1 Modelldefinition

Pseudo dreidimensionale HMMs modellieren die Abhängigkeiten von Merkmalen einer Bildsequenz durch einen dreistufigen, hierarchischen Prozeß. Der in dieser Hierarchie am höchsten stehende Prozeß modelliert die Abhängigkeiten von aufeinanderfolgenden Bildern mit einem Markov-Modell erster Ordnung. Die Bilder selbst werden mit pseudo zweidimensionalen HMMs modelliert (siehe auch Kap. 4.3), die in das übergeordnete Modell eingebunden sind. Abb. 6.1 zeigt die Darstellung eines P3DHMMs, die das Modell als dreistufigen statistischen Automaten interpretiert. Dargestellt sind drei mit $S_1,\ldots,S_3$ bezeichnete übergeordnete Zustände, sowie den übergeordneten Zuständen zugeordnete pseudo zweidimensionale Modelle. Die P2DHMMs in Abb. 6.1 bestehen jeweils aus drei Metazuständen (z.B. $S_1^1,\ldots,S_3^1$ für das $S_1$ zugeordnete Modell), die wiederum aus jeweils vier Zuständen bestehen (z.B. $S_{\substack{1\\1}}^{\substack{1\\1}},\ldots,S_{\substack{1\\4}}^{\substack{1\\1}}$ für den Metazustand $S_1^1$). Die Zustände $S_1,\ldots,S_3$ werden im folgenden als *Hyperzustände* des P3DHMMs bezeichnet. Ein dreidimensionales Muster $O_{XYT}$, das in Form einer $X \times Y \times T$ Matrix vorliegt, kann auf folgende Weise modelliert werden: Jedes Einzelbild des Musters $(o_{xyt}, t = \text{const.})$ wird einem Hyperzustand zugeordnet. Dies ermöglicht eine nichtlineare Musterverzerrung in der Zeitdimension. Darüber hinaus werden die Einzelbilder selbst von pseudo zweidimensionalen Hidden-Markov-Modellen modelliert, was

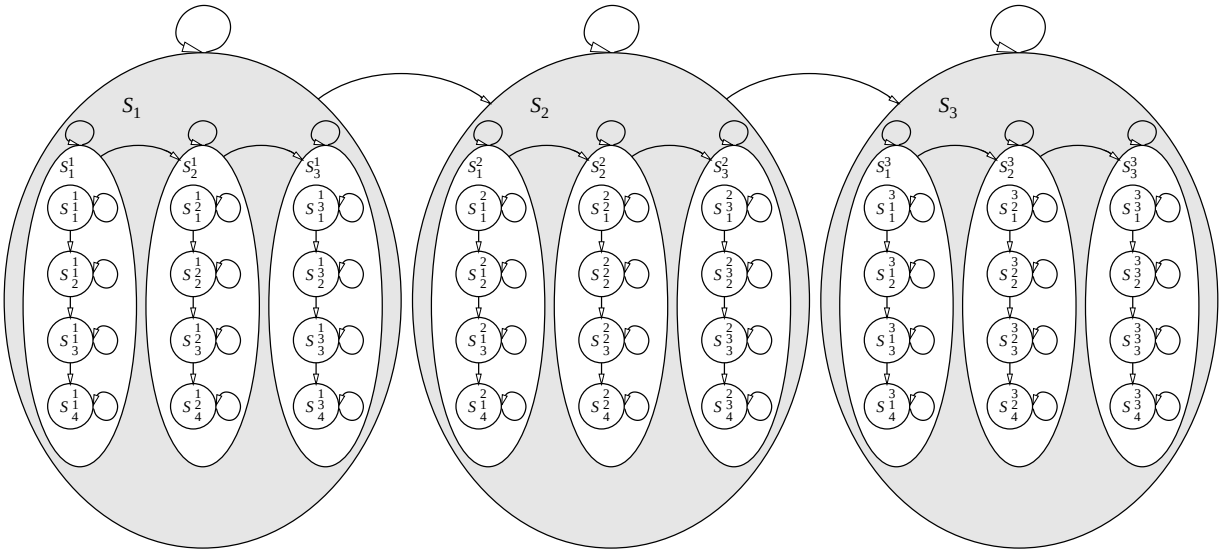


Abbildung 6.1: Pseudo dreidimensionales Hidden-Markov-Modell

eine nichtlineare Musterverzerrung in den beiden Bilddimensionen ermöglicht (siehe auch Kapitel 4.3).

Ein pseudo dreidimensionales HMM ($L$) ist durch die folgenden Parameter bestimmt: Zunächst ist die Anzahl $K$ der Hyperzustände festzulegen. Diesen Zuständen sind Übergangswahrscheinlichkeiten $a_{ij}$ zugeordnet, die in gleicher Weise, wie im eindimensionalen Fall, definiert sind (siehe Gleichung 2.2). Ebenso entspricht die Definition der Wahrscheinlichkeit für den Anfangszustand ($\pi_j$) der Gleichung 2.5. Jedem Hyperzustand des P3DHMMs ist ein P2DHMM zugeordnet. Diese können wie in Kapitel 4.3 dargestellt, definiert werden. Jeder Modellparameter erhält einen zusätzlichen hochgestellten Index, der die Zugehörigkeit zum entsprechenden Hyperzustand angibt (siehe auch Abb. 6.1). Es ergeben sich somit die folgenden Parameter für die pseudo zweidimensionalen Modelle $\Lambda_1, \ldots, \Lambda_K$, die den $K$ Hyperzuständen zugeordnet sind:

- $L^j$ ist die Anzahl der Metazustände $(S^j_1, \ldots, S^j_{L^j})$ des dem $j$-ten Hyperzustand zugeordneten P2DHMMs.

- Diesen Metazuständen sind Übergangswahrscheinlichkeiten $a^j_{kl}$ zugeordnet:

$$a^j_{kl} = P(q_{x,t} = S^j_l | q_{x-1,t} = S^j_k) \tag{6.2}$$

  Dabei ist mit $q_{x,t}$ die Zufallsvariable für die Einnahme eines Metazustandes für die Bildspalte $x$ zum Zeitpunkt $t$ bezeichnet.

- Die Wahrscheinlichkeit für die Anfangszustände sind gegeben durch:

$$\pi^j_i = P(q_{1,t} = S^j_i) \tag{6.3}$$

Jedem Metazustand ist wiederum ein eindimensionales HMM zugeordnet. Dieses wird durch die folgenden Parameter bestimmt:

- $M_i^j$ ist die Anzahl der Zustände $(S_{\ 1}^{\ j}_{\ i}, \ldots, S_{\ M_i^j}^{\ j}_{\ i})$ des dem $j$-ten Hyperzustand und dem $i$-ten Metazustand zugeordneten eindimensionalen Hidden-Markov-Modells.

- Diesen Zuständen sind die folgenden Übergangswahrscheinlichkeiten $a_{\ kl}^{\ j}_{\ i}$ zugeordnet:

$$a_{\ kl}^{\ j}_{\ i} = P(q_{x,y,t} = S_{\ l}^{\ j}_{\ i} | q_{x,y-1,t} = S_{\ k}^{\ j}_{\ i}) \tag{6.4}$$

  Dabei ist mit $q_{x,y,t}$ die Zufallsvariable für die Einnahme eines Zustandes für die Bildspalte $x$ und die Bildzeile $y$ zum Zeitpunkt $t$ bezeichnet.

- Die Wahrscheinlichkeit für die Anfangszustände sind gegeben durch:

$$\pi_{\ k}^{\ j}_{\ i} = P(q_{x,1,t} = S_{\ k}^{\ j}_{\ i}) \tag{6.5}$$

- Die Ausgabeverteilungen können wiederum diskret oder kontinuierlich sein. Eine diskrete Ausgabeverteilung $b_{\ k}^{\ j}_{\ i}(l)$ über einem für das gesamte P3DHMM festgelegten Alphabet kann angegeben werden als:

$$b_{\ k}^{\ j}_{\ i}(l) = P(v_l | q_{x,y,t} = S_{\ k}^{\ j}_{\ i}) \tag{6.6}$$

  Die in Kapitel 2.2.4 dargestellten Gaußschen Mischverteilungen können alternativ verwendet werden, um kontinuierliche P3DHMMs zu erhalten.

Es sei an dieser Stelle angemerkt, daß die üblichen statistischen Randbedingungen für die vorgestellten Modellgrößen eingehalten werden müssen (siehe auch Gleichung 2.4). Durch die Spezifizierung der angegebenen Modellgrößen wird ein P3DHMM vollständig beschrieben. Es wurde in dieser Arbeit bereits an mehreren Stellen darauf hingewiesen, daß durch den Viterbi-Algorithmus die Möglichkeiten gegeben sind, HMMs für die Klassifikation einzusetzen und zudem auch die HMMs an Trainingsdaten anzupassen (siehe auch Kapitel 2 und 4). Für den Fall, daß ein verallgemeinerter Viterbi-Algorithmus existiert, ist es möglich, die P3DHMMs zu trainieren und Bildsequenzen mit den Modellen zu klassifizieren. Solch ein verallgemeinerter Viterbi-Algorithmus existiert in Form des dreifachverschachtelten Viterbi-Algorithmus. Dieser Algorithmus basiert auf dem zweifachverschachtelten Viterbi-Algorithmus für P2DHMMs (siehe Kapitel 4.3.2) und geht aus diesem hervor, indem ein weiterer, übergeordneter Viterbi-Durchlauf für die Zeitdimension hinzugefügt wird. Da es wie schon im zweidimensionalen Fall wiederum möglich ist, eine den P3DHMMs gleichwertige eindimensionale HMM-Struktur zu finden, wird an dieser Stelle auf die ausführliche Darstellung des dreifachverschachtelten Viterbi-Algorithmus verzichtet. Das folgende Unterkapitel beschreibt eine gleichwertige eindimensionale Modellierung, die es ermöglicht, die in Kapitel 2 vorgestellten Trainings- und Klassifikationsalgorithmen zu verwenden.

### 6.1.2 Umformung in gleichwertige eindimensionale Hidden-Markov-Modelle

Auf ähnliche Weise, wie im zweidimensionalen Fall (siehe Kapitel 4.3.3) ist im Rahmen dieser Arbeit mit Hilfe von Markierungszuständen und -merkmalen eine eindimensionale Modellierung entwickelt worden, die gleichwertig mit der pseudo dreidimensionalen Modellierung ist. Die grundlegende Idee dabei ist, die von Samaria in [Sam94b] vorgeschlagene Modellierungstechnik zweimal anzuwenden. Dies bedeutet, daß außer den Markierungsmerkmalen für den Anfang einer Bildspalte auch Merkmale eingefügt werden müssen, die den Anfang eines Einzelbildes markieren. Zusätzlich sind auch Markierungszustände dem eindimensionalen HMM hinzuzufügen, deren Ausgabeverteilungen mit den Markierungsmerkmalen korrespondieren.

Die Abbildungen 6.2 und 6.3 stellen schematisch die gleichwertige eindimensionale Modellierung dar. In Abb. 6.2 ist das zu modellierende dreidimensionale Muster gezeigt, bei dem es sich um einen Ausschnitt aus einer Bildsequenz, die einer Gestendatenbasis entnommen wurde, handelt. Um die Modellierung mit eindimensionalen HMMs zu ermöglichen, müssen die Merkmale der Bildsequenz in eine Merkmalsequenz überführt werden. Dies geschieht in der gleichen Weise, wie im zweidimensionalen Fall: Merkmale werden mit einem Abtastfenster für jedes Einzelbild von oben nach unten und links nach rechts entnommen (vgl. auch Abb. 4.3). Der Anfang einer jeden Bildspalte wird durch ein in Abb. 6.2 grau dargestelltes Markierungsmerkmal angezeigt. Die auf diese Weise erhaltenen Merkmale der Einzelbilder werden unter Berücksichtigung der zeitlichen Reihenfolge und unter Verwendung von weiteren Markierungsmerkmalen aneinandergehängt. Diese zusätzlichen Markierungsmerkmale sind in Abb. 6.2 schwarz dargestellt und zeigen den Anfang eines Einzelbildes an.

Abb. 6.3 illustriert die eindimensionale Modelltopologie, mit der ganze Bildsequenzen modelliert werden können. Die grau-schattierten Zustände sind die Markierungszustände, die beim Auftreten eines Markierungsmerkmals, das den Anfang einer Bildspalte anzeigt, hohe Wahrscheinlichkeiten ausgeben. Die schwarz dargestellten Markierungszustände geben hohe Wahrscheinlichkeiten beim Auftreten eines Markierungszustandes, das den Anfang eines Einzelbildes anzeigt, aus. Da die Markierungsmerkmale sowohl für die Bildspalten als auch für die Einzelbilder nur einzeln in der Merkmalsequenz auftreten, sind die Selbstübergänge der zugehörigen Markierungszustände auf Null zu setzen (z.B. $a_{100,100} = P(q_k = S_{100}|q_{k-1} = S_{100}) = 0, a_{110,110} = 0, a_{120,120} = 0, \ldots$). Die Übergangswahrscheinlichkeiten, die zu den Markierungszuständen führen, also z.B. $a_{114,110}$ und $a_{134,100}$ modellieren die statistischen Abhängigkeiten aufeinanderfolgender Bildspalten und Einzelbilder. Sie entsprechen somit den Übergangswahrscheinlichkeiten der Metazustände, bzw. der Hyperzustände.

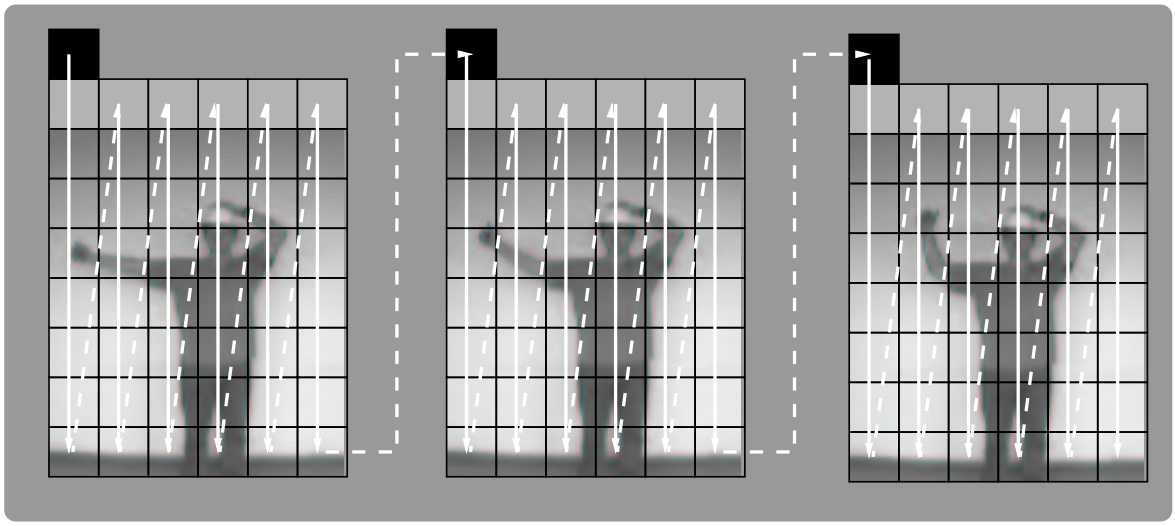

Abbildung 6.2: Überführung der Merkmale einer Bildfolge in eine eindimensionale Sequenz

Obwohl es die unterschiedliche farbliche Gestaltung der Markierungszustände und -merkmale in den Abb. 6.2 und 6.3 vermuten läßt, müssen für diese keine unterschiedlichen Ausgabefunktionen bzw. Werte gewählt werden. Sowohl für die Markierungsmerkmale der Einzelbilder, als auch der Bildspalten können dieselben Werte verwendet werden und somit die Parameter der Markierungszustände in Abb. 6.3 alle miteinander verknüpft werden. Dies ist möglich, da der Anfang eines Einzelbildes durch zwei aufeinanderfolgende Markierungsmerkmale stets eindeutig gekennzeichnet ist. Wie schon im zweidimensionalen Fall ist bei dieser Modellierungsmethode darauf zu achten, daß die Markierungsmerkmale ausschließlich den Markierungszuständen zugeordnet werden. Um dies zu erreichen, können unverändert die in Unterkapitel 4.3.3 beschriebenen Methoden verwendet werden.

Da es sich bei der in diesem Unterkapitel beschriebenen Modelltopologie um eine eindimensionale handelt, können die Trainings- und Klassifikationsalgorithmen des Kapitels 2 verwendet werden. Im folgenden werden nach einer kurzen Einführung in das Gebiet der Bildsequenzklassifikation experimentelle Ergebnisse präsentiert, die mit der vorgestellten Modellierungstechnik erreicht wurden.

## 6.2 Klassifikation von Bildsequenzen

Die populärste Anwendung von Hidden-Markov-Modellen ist der Bereich der Klassifikation von sich zeitlich ändernden Mustern. Es liegt somit nahe, diese Modelle auch für die Klassifikation von Bildsequenzen einzusetzen. Dies erfolgte in den verschiedensten Anwendungsszenarien, wie beispielsweise Videoindexierung oder Gestikerkennung. Der letztgenannte Bereich, die Gestikerkennung, ist als Anwendungsszenario für die Evaluierung der

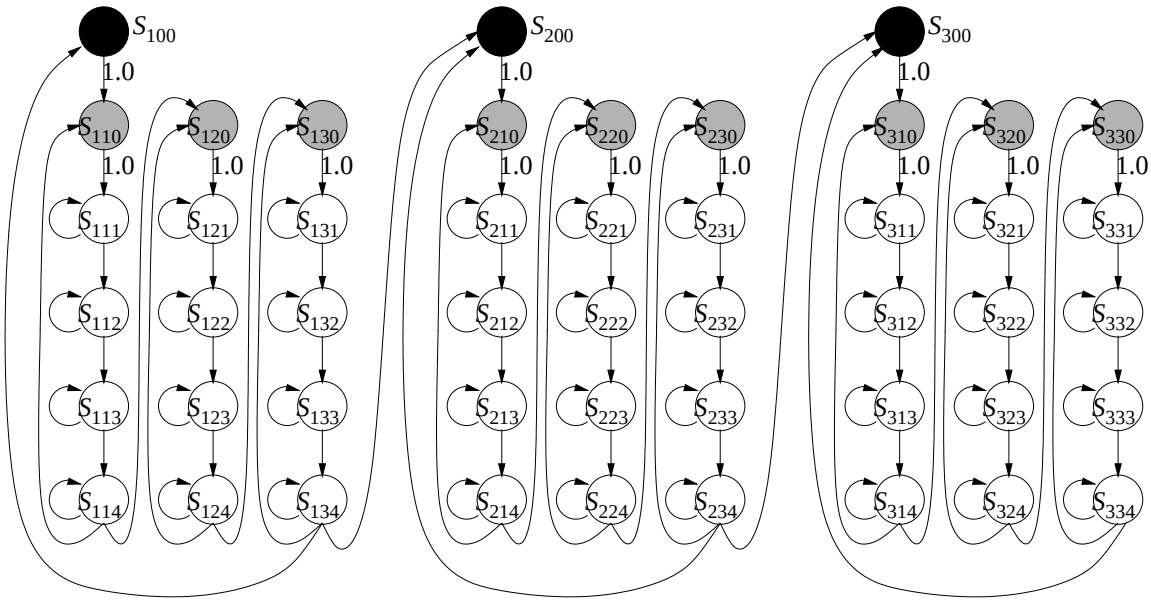


Abbildung 6.3: Eindimensionale HMM-Struktur mit Markierungszuständen, die eine im Vergleich mit den P3DHMMs gleichwertige Modellierung ermöglicht.

neuartigen pseudo dreidimensionalen Hidden-Markov-Modelle gewählt worden, da bereits Ergebnisse mit einem alternativen Verfahren am Lehrstuhl Technische Informatik vorliegen und sich somit eine Vergleichsmöglichkeit bietet. Bevor der Einsatz der P3DHMMs auf dem Gebiet der Gestikerkennung beschrieben wird, werden im folgenden Unterkapitel zunächst relevante Arbeiten anderer Autoren vorgestellt.

### 6.2.1 Relevante Arbeiten anderer Autoren

Es existiert eine große Zahl von Arbeiten, vorwiegend aus den 90er Jahren, die über die Erkennung von Gesten mit Hidden-Markov-Modellen berichten. Die wahrscheinlich erste Arbeit zu diesem Thema stammt von Yamato et al. [Yam92] und beschreibt Experimente mit diskreten eindimensionalen Hidden-Markov-Modellen, die für die Klassifikation von Tennis-Schlagtechniken eingesetzt werden. Es wurden die folgenden sechs Tennis-Schlagtechniken verwendet, die die zu erkennenden Klassen bilden: Rückhand-Volley, Rückhand-Schlag, Vorhand-Volley, Vorhand-Schlag, Schmettern und Aufschlag. Es wird eine Vielzahl von Vorverarbeitungsschritten, wie z.B. Tiefpaßfilterung, Hintergrundsubtraktion und Binarisierung auf jedes einzelne Bild der Sequenz angewendet. Das Ergebnis dieser Vorverarbeitung ist ein zweiwertiges Bild, das im wesentlichen die extrahierte Pose der Person darstellt. Vor der Berechnung der Merkmale wird zusätzlich eine Größennormalisierung und eine Zentrierung vorgenommen. Die Merkmale sind die Anzahlen von schwarzen Bildpunkten in den Segmenten eines Abtastrasters. Diese Merkmale werden anschließend vektorquantisiert und somit ergibt sich eine Sequenz von Symbolen, die die Bildsequenz repräsentiert. Diese Se-

quenz von Symbolen kann mit einem eindimensionalen diskreten Hidden-Markov-Modell weiterverarbeitet werden.

Schuster und Rigoll verwenden in der Arbeit [Sch96] ebenfalls diskrete Hidden-Markov-Modelle für die Erkennung von Bildsequenzen. Der Hauptunterschied zu [Yam92] liegt in der Verwendung einer wesentlich simpleren Vorverarbeitung, was den Echtzeiteinsatz ermöglicht. Die Vorverarbeitung in [Sch96] besteht lediglich aus der Unterabtastung der RGB-Kanäle der einzelnen Farbbilder einer Sequenz. Horizontale bzw. vertikale Streifen dieser Abtastwerte werden anschließend, ohne weitere Verarbeitungsschritte, vektorquantisiert und zusammen mit den diskreten HMMs für die Klassifikation eingesetzt. Alternativ wurden dieselben Schritte auf Differenzbildern angewendet. Das echtzeitfähige System wurde auf einer Gestendatenbasis evaluiert, die aus zehn selbstdefinierten Klassen besteht. Beispiele für diese Klassen sind: klatschen, sich verbeugen, nicken und den Kopf schütteln.

Das oben beschriebene System ist durch die Verwendung von kontinuierlichen Hidden-Markov-Modellen zusammen mit geometrischen Momenten, die auf Differenzbildern berechnet werden, weiter verbessert worden. Dieses System verwendet, wie in [Rig96] berichtet wurde, 24 verschieden Klassen, die mit einer Genauigkeit von mehr als 90% erkannt werden.

Die Kombination aus kontinuierlichen Hidden-Markov-Modellen und geometrischen Momenten wurde ebenfalls von Starner et al. in der Arbeit [Sta98] verwendet. Das System in [Sta98] erkennt amerikanische Zeichensprache und verwendet die folgenden Vorverarbeitungschritte: Die Hände der Person werden in den einzelnen Bildern der Sequenz lokalisiert und basierend auf diesen Regionen werden Momente berechnet. Neben diesen Merkmalen werden dynamische Merkmale, wie die Positionsveränderung der Hände zwischen den Einzelbildern verwendet.

Die bisher kurz vorgestellten Systeme sind sehr stark abhängig von der Existenz von Bewegung, da sehr oft Merkmale, die auf Differenzbildern basieren, eingesetzt werden. Diese Einschränkung kann durch den Einsatz von pseudo dreidimensionalen Hidden-Markov-Modellen überwunden werden. Dies wird im folgenden Kapitel erläutert.

### 6.2.2 Klassifikation von Bildsequenzen mit P3DHMMs

Mit Hilfe der pseudo dreidimensionalen Hidden-Markov-Modelle können sowohl auf Bewegung basierende Merkmale als auch statische, auf den Einzelbildern berechnete Merkmale gemeinsam verwendet werden. Die Integration in ein einzelnes Modell erfolgt über die Merkmalströme (siehe auch Kapitel 2). Zusätzlich ermöglicht die Modellierung mit P3DHMMs ein flexibles Erkennungsverhalten auf den einzelnen Bildern der Sequenz. Dies ist ein großer Unterschied zu den im vorhergehenden Unterkapitel vorgestellten Ansätzen, da hier entweder VQ-Indices ([Yam92, Sch96]) oder globale Merkmale ([Rig96, Sta98]) auf den einzelnen Bildern der Sequenz berechnet werden und somit die Bilder auf starre Weise

modelliert werden. Der Vorteil des flexiblen Erkennungsverhaltens auf den Einzelbildern bei Verwendung der P3DHMMs ist, neben der besseren Erkennungsleistung, die Toleranz gegenüber Positionsveränderungen. Ist die Position einer gestikulierenden Person in einer Bildsequenz beispielsweise relativ zu der Position in einer Trainingssequenz verändert, so wird dies durch die flexible Modellierung der Einzelbilder durch die P2DHMMs ausgeglichen. Die P2DHMMs ermöglichen eine nichtlineare Musterverzerrung in beiden Bilddimensionen und daher wird insbesondere die Verschiebung in $x$-Richtung, also eine translatorische Positionsveränderung, kompensiert. Dies erlaubt die Erkennung von Gesten auch für den Fall, daß sich die gestikulierende Person selbst in einer translatorischen Bewegung befindet und somit die Position im Bild verändert.

Die zusammen mit den P3DHMMs verwendete Merkmalextraktion basiert, wie im Fall der Bildklassifikation mit P2DHMMs (siehe Unterkapitel 5.1), auf der Diskreten-Kosinus-Transformation. Die DCT-Koeffizienten werden sowohl auf den Einzelbildern der Sequenz als auch auf den Differenzbildern berechnet und somit ergeben sich statische und dynamische Merkmale. Durch die Verwendung der Merkmalströme können diese Merkmale zu heterogenen Merkmalvektoren zusammengefaßt werden. Die Merkmalstrom-Gewichte ermöglichen ferner, den Einfluß der statischen und der dynamischen Merkmale zu kontrollieren.

## 6.3 Experimentelle Ergebnisse

Die P3DHMMs wurden anhand einer aus 12 Klassen bestehenden Gesten-Datenbasis evaluiert. Zusätzlich sind die erzielten Ergebnisse mit denen, die mit einem alternativen Verfahren erreicht wurden, verglichen worden. Bevor diese Ergebnisse detailliert vorgestellt werden, wird zunächst der Aufbau der verwendeten Datenbasis erläutert.

### 6.3.1 Gesten-Datenbasis

Die in den Experimenten verwendete Gesten-Datenbasis wurde am Fachgebiet Technische Informatik vom Autor erstellt. Die Datenbasis besteht aus 12 verschiedenen Gesten, die der Steuerung von Baukränen dienen. Diese Gesten ermöglichen das Manövrieren von Kränen in vom Kranführer schwer einsehbarem Gelände. Eine zweite Person, deren Sicht auf eine zu positionierende Last besser ist, kann durch Ausführen der Gesten dem Kranführer assistieren. Dieses Gestenvokabular ist wohldefiniert und ist z.B. in dem Nachschlagewerk für Mechanik [Par80] zu finden. Abb. 6.4 illustriert die verschiedenen Klassen, die folgendermaßen benannt sind: links-herumdrehen, rechts-herumdrehen, näherkommen, entfernen, Lastarm ausfahren, Lastarm einziehen, Lastarm hoch, Lastarm runter, hochwinden, herunterlassen, halt, nothalt. Die beiden letztgenannten Klassen stellen Beispiele für statische Gesten dar, da sie durch Körperhaltungen und nicht durch Bewegungsabläufe definiert sind, was Abb. 6.4 entnommen werden kann. Fünf Personen führten die in Abb. 6.4 definierten

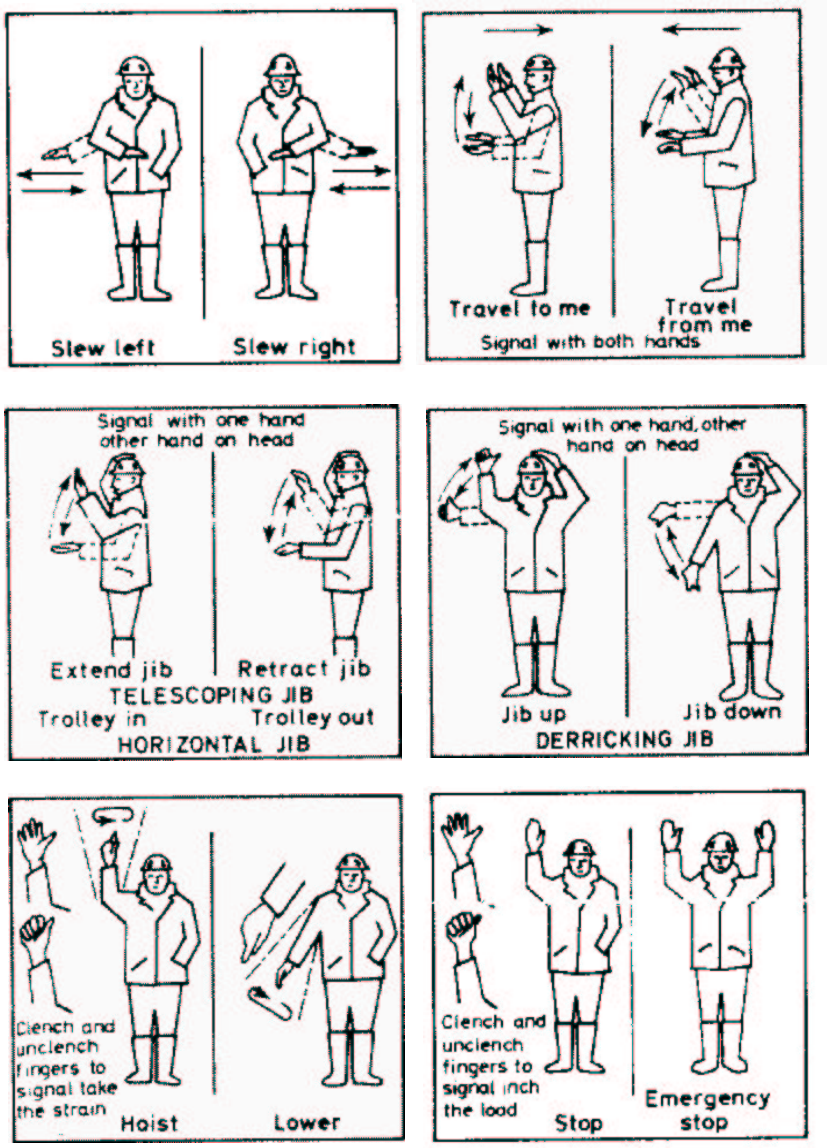


Abbildung 6.4: Darstellung der 12 Gesten, mit denen Baukräne manövriert werden können. Ins Deutsche übersetzt ergeben sich die folgenden Klassenbezeichnungen (von links nach rechts und oben nach untern): links-herumdrehen, rechts-herumdrehen, näherkommen, entfernen, Lastarm ausfahren, Lastarm einziehen, Lastarm hoch, Lastarm runter, hochwinden, herunterlassen, halt, nothalt (aus [Par80])

Abbildung 6.5: Ausschnitte aus Sequenzen, die die Gesten *Lastarm hoch* und *Lastarm runter* darstellen

Bewegungsabläufe mehrere Male durch. Zwei Beispiele für jede Gesten-Klasse dienen als Trainingsdatensatz, die verbleibenden Beispiele werden in der Erkennungsphase verwendet. Die Abb. 6.5 zeigt zwei Ausschnitte aus Sequenzen, die zu den Klassen *Lastarm hoch* (obere Zeile) und zur Klasse *Lastarm runter* (untere Zeile) gehören. Die Bildsequenzen sind mit einer Auflösung von $192 \times 144$ Bildpunkten und einer Bildwiederholrate von 25 Bildern pro Sekunde digitalisiert worden. Die Aufnahme selbst erfolgte mit einer analogen Videokamera.

### 6.3.2 Quantitative Ergebnisse

Es wurde bei der Durchführung der Experimente die folgende P3DHMM-Topologie verwendet: Das Modell bestand aus 4 Hyperzuständen, denen jeweils ein $(5 \times 5)$ P2DHMM zugeordnet war. Die Größe der Abtastfenster, auf die eine diskrete Cosinustransformation angewendet wurde, betrug $16 \times 16$ Bildpunkte. Die DCT-Koeffizienten wurden sowohl auf Abtastfenstern, die Grauwerte enthalten, als auch auf Abtastfenstern, die Differenzen benachbarter Einzelbilder enthalten, berechnet. Die dabei entstehenden Merkmalströme wurden auf gleiche Weise mit den Werten $\gamma_1 = \gamma_2 = 1$ gewichtet. In Tabelle 6.1 sind in der zweiten Spalte die mit dieser Konfiguration erzielten Erkennungsgenauigkeiten angegeben. Diese Erkennungsgenauigkeiten wurden jeweils getrennt für die einzelnen Personen ermittelt, die in Tabelle 6.1 in der ersten Spalte aufgelistet sind. Zusätzlich sind zur besseren Bewertbarkeit dieser Ergebnisse in der dritten Spalte Erkennungsgenauigkeiten angegeben, die mit einem alternativen Ansatz erzielt worden sind. Dieser alternative Ansatz verwendet geometrische Momente, die auf den Differenzbildern berechnet werden, sowie eindimensionale kontinuierliche Hidden-Markov-Modelle zur zeitlichen Modellierung. Eine ausführliche Darstellung dieses Ansatzes ist in [Rig97] zu finden. Der Tabelle 6.1 kann entnommen werden, daß der neuartige P3DHMM basierte Ansatz im Vergleich zur eindimensionalen Modellierung eine höhere durchschnittliche Erkennungsgenauigkeit erzielt hat. Darüber hinaus existieren noch zwei weitere Vorteile der pseudo dreidimensionalen Hidden-Markov-Modelle: Zum einen können mit dieser Methode statische und dynamische Gesten gemeinsam und unter Verwendung eines Ansatzes modelliert und klassifiziert werden. Ein zweiter Vorteil ist, daß durch die elastische Modellierung der Einzelbilder durch die P3DHMMs eine positions- und größen-

| Person | P3DHMM | 1DHMM |
|---|---|---|
| ste | 88,6% | 100% |
| stm | 91,2% | 85,3% |
| ank | 100% | 100% |
| bw | 94,1% | 88,2% |
| jmr | 80,5% | 80,5% |
| Durchschnitt | 90,88% | 90,74% |

Tabelle 6.1: In den Experimenten erzielte Erkennungsgenauigkeiten

tolerante Erkennung durchgeführt werden kann. Dies wird durch Experimente belegt, die in [Yal00b] und [Yal00a] dokumentiert sind. Neben der Anwendung in der Gestikerkennung können P3DHMMs auch auf anderen Gebieten eingesetzt werden. So sind z.B. in [Hue01] Experimente beschrieben, die die Eignung der P3DHMMs für die Erkennung menschlicher Gesichtsausdrücke bzw. Gemütszustände belegen.

## 6.4 Ausblick auf einen integrierten Ansatz zur Klassifikation und Segmentierung mit P3DHMMs

In Kapitel 5 wurde für den zweidimensionalen Fall zunächst die Klassifikation von Einzelbildern mittels pseudo zweidimensionaler Hidden-Markov-Modelle beschrieben und, im Anschluß daran, die integrierte Segmentierung und Klassifikation von komplexen Szenen durch erweiterterte P2DHMMs vorgestellt. Für den in diesem Kapitel betrachteten dreidimensionalen Fall wurde bisher die Klassifikation von Bildsequenzen mit neuartigen pseudo dreidimensionalen HMMs vorgestellt. Obwohl die in Kapitel 5 vorgestellten Modellierungstechniken nahezu unverändert auf den dreidimensionalen Fall übertragen werden können, werden in dieser Arbeit keine Experimente mit den um Umgebungszustände erweiterten P3DHMMs vorgestellt. Der Grund hierfür ist der sehr hohe Rechenbedarf, der für solche Experimente benötigt wird. Als Ausblick soll an dieser Stelle auf die möglichen Einsatzgebiete für die integrierte Segmentierung und Klassifikation mit P3DHMMs hingewiesen werden. Die Modelltopologie für diesen Ansatz ist in Abb. 6.6 dargestellt. Wie schon in Abbildung 5.4 sind die Umgebungszustände grau schattiert, während die klassenbeschreibenden Zustände weiß ausgefüllt sind. Die Parameter der Umgebungszustände können, wie es in Kapitel 5.3.1 dargestellt wurde, z.B. unter Verwendung aller Merkmale einer zu analysierenden Sequenz bestimmt werden. Der integrierte Segmentierungs- und Klassifikationsansatz mit P3DHMM kann für die folgenden Aufgaben eingesetzt werden:

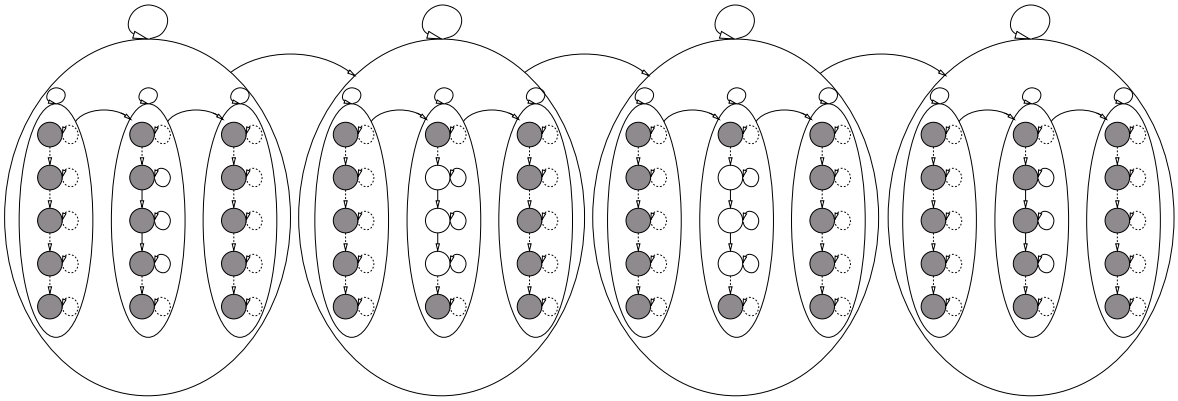

Abbildung 6.6: Pseudo dreidimensionales Hidden-Markov-Modell mit Umgebungszuständen

- Auffinden von Personen in Bildsequenzen aufgrund charakteristischer Bewegungen und des Aussehens
- Erkennen von Gesten einer sich bewegenden Person
- Abfrage von Filmdatenbanken mit folgenden Aufgaben
  - Auffinden von Actionszenen in Spielfilmen
  - Erkennen von Bewegungsabläufen im Sport
  - Anfrage an Filmdatenbanken mit Beispielgesten

Das erstgenannte Beispiel stellt eine um die Zeitdimension erweiterte Version des bereits in der Einleitung vorgestellten Problems dar, ein bekanntes Muster zu erkennen, das in eine komplexen Umgebung eingebettet ist (siehe Abb. 1.1). Da nun das zu erkennende Muster auch in der Zeitdimension aufzufinden ist, besteht der erste und der letzte Hyperzustand des erweiterten P3DHMM in Abb. 6.6 ausschließlich aus Umgebungszuständen. Durch diese Maßnahme wird es ermöglicht, ein Muster zu erkennen, welches am Anfang bzw. am Ende einer Bildsequenz möglicherweise *nicht* vorkommt.

## 6.5 Kapitelzusammenfassung

Es wurde die Klassifikation von Bildfsequenzen mit neuartigen pseudo dreidimensionalen Hidden-Markov-Modellen vorgestellt. Diese Modellierung ermöglicht es, Merkmale, die auf Einzelbildern berechnet wurden, gemeinsam mit Merkmalen zu verwenden, die aus der temporalen Abfolge der Bilder bestimmt wurden. Somit können dynamische und statische Gesten gemeinsam mit einem Modell erkannt werden.

Es wurde in die Theorie der P3DHMMs eingeführt und dargestellt, wie gleichwertige eindimensionale Modelle konstruiert werden können. Die Klassifikation von Bildsequenzen wurde anhand einer Gestendatenbasis demonstriert. Die verwendete Datenbasis besteht

aus 12 Gesten, die zur Steuerung von Baukränen dienen. In Experimenten wurden auf dieser Datenbasis höhere Erkennungsgenauigkeiten erzielt, als mit einem alternativen Ansatz, der eindimensionale Hidden-Markov-Modelle in Kombination mit geometrischen Momenten verwendet. Als Ausblick wurde auf die integrierte Segmentierung und Klassifikation von Bildsequenzen hingewiesen, die z.B. eine positionsunabhängige und gleichzeitig hintergrundunabhängige Gestenerkennung ermöglicht.

# Kapitel 7

# Zusammenfassung

Ziel dieser Arbeit war die Nutzung der integrierten Segmentierungs- und Klassifikationseigenschaften der Hidden-Markov-Modelle für die Erkennung von Mustern in Bildern und Bildsequenzen. Diese besondere Eigenschaft ist durch den Viterbi-Algorithmus gegeben, der eine Merkmal-Zustandszuordnung ausgibt, die als Segmentierung interpretiert werden kann. Ferner liefert der Viterbi-Algorithmus einen Schätzwert dafür, daß ein gegebenes Muster von einem HMM produziert wurde. Unter Verwendung dieses Schätzwertes erfolgt die Musterklassifikation. Obwohl Hidden-Markov-Modelle schon seit den 80er Jahren erfolgreich bei der Erkennung von zeitlich veränderlichen Mustern, wie beispielsweise Sprache oder Online-Handschrift, eingesetzt werden und in diesen Anwendungsszenarien somit schon lange Gebrauch gemacht wird von der Fähigkeit in einem Schritt segmentieren und klassifizieren zu können, so ist die Nutzung dieser Eigenschaften auf dem Gebiet der Bild- und Bildsequenzerkennung als neu anzusehen. Dabei ist die Verwendung der populären Hidden-Markov-Modelle bei der Erkennung von Bildern, also zweidimensionaler Muster, keinesfalls trivial, da eine Erweiterung der eindimensionalen Struktur dieser Modelle erforderlich ist. Gleiches gilt für Bildsequenzen, die dreidimensionale Muster darstellen.

In dieser Arbeit konnte zunächst ohne Verwendung der höherdimensionalen Hidden-Markov-Modelle die erfolgreiche Anwendung der integrierten Segmentierungs- und Klassifikationseigenschaften auf dem Gebiet der automatischen Bilderkennung demonstriert werden. Dazu wurden neuartige eindimensionale HMM-Topologien vorgestellt, die zusammen mit einer polaren Abtastung eine translations-, skalierungs- und rotationsunabhängige Modellierung von Objektformen bzw. handskizzierten Piktogrammen ermöglichen. Die integrierten Segmentierungs- und Klassifikationseigenschaften der HMMs wurden dazu genutzt, die Orientierung der gedrehten Objekte herauszufinden und das Objekt zu erkennen. In den Experimenten wurden Erkennungsgenauigkeiten von bis zu 99.5% mit Piktogramm-Datenbasen erreicht, die aus 20 Klassen bestehen. Die Erkennungsergebnisse lagen über denen, die mit konventionellen Erkennungsmethoden, nämlich Momenten in Kombination mit künstlichen neuronalen Netzen erzielt wurden. Die vorgestellten Methoden konnten erfolgreich auf die Erkennung natürlicher Bilder erweitert werden. Es wurden Ergebnisse präsen-

tiert, die mit einem Bilddatenbanksystem, das intuitiv über Skizzen des Benutzers abgefragt werden kann und das die neuartigen eindimensionalen Modelltopologien verwendet, erzielt wurden.

Die Erkenntnisse, die durch die Experimente mit den eindimensionalen Hidden-Markov-Modellen erzielt wurden, konnten genutzt werden, um die kombinierten Segmentierungs- und Klassifikationseigenschaften auch im zweidimensionalen Fall nutzen zu können. Die vorliegende Arbeit präsentierte einen Ansatz, der es ermöglicht, zweidimensionale Muster in komplexen Umgebungen aufzufinden und zu klassifizieren. Dabei wurden pseudo zweidimensionale Hidden-Markov-Modelle in Kombination mit an den Bildkontext angepaßten Umgebungszuständen verwendet. Pseudo zweidimensionale HMMs stellen eine hierarchische Erweiterung der eindimensionalen Modelle dar und sind geeignet, um zweidimensionale Muster zu modellieren. Es existieren effiziente Algorithmen für das Training und die Klassifikation mit diesen Modellen und somit bietet sich die Anwendung dieser Modelle an. Die Anpassung der Parameter der Umgebungszustände kann auf verschiedene Weisen erfolgen, je nachdem, ob Vorwissen über die zu analysierende Szene vorliegt oder nicht. Für den letztgenannten Fall wurde ein Verfahren entwickelt, bei dem die Parameter der Umgebungszustände auf allen Merkmalen des zu analysierenden Bildes bestimmt wurden. Nach der Ausführung des Viterbi-Algorithmus liegt eine Zuordnung der Merkmale zu den Umgebungszuständen und den Zuständen des gesuchten Musters vor, die als Segmentierung des Bildes in Muster und Umgebung interpretiert werden kann. Zusätzlich liefert der Viterbi-Algorithmus einen Schätzwert für die Produktionswahrscheinlichkeit, der zur Klassifikation genutzt werden kann. Das neuartige Verfahren wurde zunächst auf die Erkennung von handskizzierten Piktogrammen in komplexen Szenen angewendet. Dabei wurden Erkennungsgenauigkeiten von 90% auf einer Piktogrammdatenbasis erreicht, die aus 20 Klassen bestand. Weitere Experimente wurden beschrieben, die die Eignung des Ansatzes für das Auffinden von benutzerdefinierten Formen in technischen Zeichnungen belegen. Somit ist es z.B. möglich, eine durch eine Skizze spezifizierte Schraube in komplexen technischen Zeichnungen aufzufinden. Schließlich wurde der P2DHMM-Ansatz für das Personen-Tracking in Bildfolgen eingesetzt. Dabei konnte gezeigt werden, daß Muster bzw. Personen auch in Grauwert- und Farbbildern mit dem vorgestellten Ansatz gefunden werden können. Es zeigte sich ebenfalls, daß der Ansatz gut kombinierbar ist mit einem Kalman-Filter, das die Dynamik der Bewegung einer Person modelliert. Obwohl diese zweistufige Methode, die P2DHMMs für das Auffinden der Person und das Kalman-Filter für die Bewegungsmodellierung verwendet, gute Ergebnisse zeigte, so ist eine dreidimensionale Modellierung sehr viel geeigneter für die Erkennung von Bildfolgen.

Eine solche dreidimensionale Modellierung wurde im Rahmen dieser Arbeit entwickelt. Die neuartigen sog. pseudo dreidimensionalen Hidden-Markov-Modelle ermöglichen es, Merkmale, die auf Einzelbildern berechnet werden, gemeinsam mit Merkmalen zu modellieren, die aus der temporalen Abfolge der Bilder bestimmt werden. Somit können dynamische

und statische Bewegungsmuster gemeinsam mit einem Modell erkannt werden. Die Evaluierung des P3DHMM-Ansatzes erfolgte anhand einer selbsterstellten Gestendatenbank, die aus 12 Gesten besteht, die der Steuerung von Baukränen dienen. Es wurden experimentelle Ergebnisse mit diesen Modellen erzielt, die denen, die mit einem alternativen Ansatz, der eindimensionale HMMs in Kombination mit geometrischen Momenten verwendet, überlegen sind. Als Ausblick wurde auf die integrierte Segmentierung und Klassifikation von Bildsequenzen mit P3DHMMs und Umgebungsmodell hingewiesen, die z.B. eine positionsunabhängige Gestenerkennung ermöglicht, oder für die Abfrage von Filmdatenbanken mit Beispielgesten genutzt werden kann.

Die Methode der gemeinsamen Segmentierung und Klassifikation, die bei der Verwendung von Hidden-Markov-Modellen zur Verfügung steht, konnte erfolgreich für die Erkennung von Mustern in Bildern und Bildsequenzen genutzt werden. Es wurde eine Vielzahl von Experimenten mit neuartigen Modellierungsmethoden vorgestellt, die die Eignung dieser Methoden für die Mensch-Maschine-Kommunikation und verschiedenen multimedialen Anwendungen belegen. Die Arbeit zeigt somit das große Anwendungspotential der Hidden-Markov-Modelle im Bereich der Bild- und Bildsequenzerkennung.

# Literaturverzeichnis


[Abl97] S. Ablameyko, V. Bereishik, O. Frantskevich, M. Homenko und N. Paramonova. "Algorithms for Recognition of the Main Drawing Entities." In *Proc. Intern. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 776–779. Ulm, 1997.

[Aga93a] O. E. Agazzi und S.-S. Kuo. "Pseudo Two-Dimensional Hidden Markov Models for Document Recognition." *AT&T Technical Journal*, 72, Nr. 5, Seiten 60–72, September-Oktober 1993.

[Aga93b] O. E. Agazzi, S.-S. Kuo, E. Levin und R. Pieraccini. "Connected and Degraded Text Recognition Using Planar Hidden Markov Models." In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Band 5, Seiten 113–116, 1993.

[Bip97] R.-D. Bippus. "1-Dimensional and Pseudo 2-Dimensional HMMs for the Recognition of German Literal Amounts." In *Proceedings International Conference on Document Analysis and Recognition (ICDAR)*, Seiten 487–490, 1997.

[Bip00] R.-D. Bippus. *Stochastische Modelle zur off-line Fließschrifterkennung*. Shaker Verlag, Aachen, 2000.

[Bod88] K.-H. Bode. *Konstruktions-Atlas: Werkstoff- und verfahrensgerecht konstruieren*. Hoppenstedt Technik Tabellen Verl., 1988.

[Che95] R. Chellappa, C. L. Wilson und S. Sirohey. "Human and Machine Recognition of Faces: A Survey." *Proceedings of the IEEE*, 83, Nr. 5, Seiten 705–740, Mai 1995.

[Del97] A. Del Bimbo und P. Pala. "Visual Image Retrieval by Elastic Matching of User Sketches." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, Nr. 2, Seiten 121–132, Februar 1997.

[Der89] H. Derin und P. A. Kelly. "Discrete-Index Markov-Type Random Process." *Proceedings of the IEEE*, 77, Nr. 10, Seiten 1485–1510, Oktober 1989.

[Dic98] G. Dickopp. *Einführung in die Nachrichtencodierung*. Gerhard-Mercator-Universität, Duisburg, Oktober 1998.

[Eic99a] S. Eickeler und S. Müller. "Content-Based Video Indexing of TV Broadcast News Using Hidden Markov Models." In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 2997–3000. Phoenix, März 1999.

[Eic99b] S. Eickeler, S. Müller und G. Rigoll. "High Quality Face Recognition in JPEG Compressed Images." In *Proceedings IEEE Intern. Conference on Image Processing (ICIP)*, Seiten 672–676. Kobe, Japan, Oktober 1999.

[Eic00] S. Eickeler, S. Müller und G. Rigoll. "Recognition of JPEG Compressed Face Images Based on Statistical Methods." *Image and Vision Computing Journal, Special Issue on Facial Image Analysis*, 18, Nr. 4, Seiten 279–287, März 2000.

[Fli95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele und P. Yanker. "Query by Image and Video Content: The QBIC System." *IEEE Computer Magazine*, Seiten 23–32, 1995.

[For73] G. D. Forney. "The Viterbi Algorithm." *Proceedings of the IEEE*, 61, Nr. 3, Seiten 268–278, März 1973.

[Fri97] N. S. Friedland und A. Rosenfeld. "An Integrated Approach to 2D Object Recognition." *Pattern Recognition*, 30, Nr. 3, Seiten 525–535, 1997.

[Gav99] D. Gavrila. "The Visual Analysis of Human Movement: A Survey." *Computer Vision and Image Understanding*, 73, Nr. 1, Seiten 82–99, 1999.

[Gem85] S. Geman und D. Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, Nr. 6, Seiten 721–741, 1985.

[Gon92] R. C. Gonzalez und R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, Massachusetts, 1992.

[Gos85] A. Goshtasby. "Description and Discrimination of Planar Shapes Using Shape Matrices." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7, Nr. 6, Seiten 738–743, November 1985.

[Gre93] M. S. Grewal und A. P. Andrews. *Kalman filtering : theory and practice*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[Gup97] V. N. Gupta, M. Lenning und P. Mermelstein. “Integration of Acoustic Information in a Large Vocabulary Word Recognizer.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 697–700. Dallas, 1997.

[He91] Y. He und A. Kundu. “2-D Shape Classification Using Hidden Markov Model.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, Nr. 11, Seiten 1172–1184, 1991.

[Hu62] M. K. Hu. “Visual pattern recognition by moment invariants.” *IEEE Transactions on Information Theory*, 8, Seiten 178–187, 1962.

[Hue01] F. Huelsken, F. Wallhoff und G. Rigoll. “Facial Expression Recognition with Pseudo-3D Hidden Markov Models.” In *23. DAGM-Symposium, Tagungsband Springer-Verlag*. München, September 2001.

[Jai96] A. K. Jain und A. Vailaya. “Image Retrieval Using Color and Shape.” *Pattern Recognition*, 29, Nr. 8, Seiten 1233–1244, 1996.

[Jai00] A. K. Jain. “Statistical Pattern Recognition: A Review.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, Nr. 1, Seiten 4–37, 2000.

[Jan97] D.-S. Jang, G.-Y. Kim und H.-I. Choi. “Model-based Tracking of Moving Object.” 30, Nr. 6, Seiten 999–1008, 1997.

[Jua90] B.-H. Juang und L. R. Rabiner. “The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models.” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38, Nr. 9, Seiten 1639–1641, September 1990.

[Kal60] R. Kalman. “A New Approach to Linear Filtering and Prediction Problems.” *Transactions of the ASME – Journal of Basic Engineering*, Seiten 35–45, März 1960.

[Kas90] R. Kasturi, S. T. Bow, W. El-Masri, J. Shah, J. R. Gattiker und U. B. Mokate. “A System for Interpretation of Line Drawings.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, Nr. 10, Seiten 978–992, 1990.

[Kuo94] S.-S. Kuo und O. E. Agazzi. “Keyword Spotting in Poorly Printed Printed Documents Using Pseudo 2-D Hidden Markov Models.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, Nr. 8, Seiten 842–848, August 1994.

[Lee94] S. Lee und B. Lovell. “Modelling and Classification of Shapes in Two-Dimensions Using Vector Quantisation.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Band 5, Seiten 141–144, 1994.

[Lev83] S. E. Levinson, L. R. Rabiner und M. M. Sondhi. “An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition.” *The Bell System Technical Journal*, 62, Nr. 4, Seiten 1035–1055, April 1983.

[Lev92] E. Levin und R. Pieraccini. “Dynamic Planar Warping for Optical Character Recognition.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Band 3, Seiten 149–152, 1992.

[Li92] Y. Li. “Reforming the Theory of Invariant Moments for Pattern Recognition.” *Pattern Recognition*, 25, Nr. 7, Seiten 723–730, 1992.

[Li95] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, Tokyo, 1995.

[Li99] J. Li und A. Murua. “A 2D Extended HMM for Speech Recognition.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 349–352, 1999.

[Lin97] H.-C. Lin, L.-L. Wang und S.-N. Yang. “Color Image Retrieval Based on Hidden Markov Models.” *IEEE Transactions on Image Processing*, 6, Nr. 2, Seiten 332–339, Februar 1997.

[Lon98] S. Loncaric. “A Survey of Shape Analysis Techniques.” *Pattern Recognition*, 11, Nr. 8, Seiten 983–1001, 1998.

[Luc95] H. Lucke. “Improved Acoustic Modeling for Speech Recognition Using 2D Markov Random Fields.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 540–543, 1995.

[Meh97] B. M. Mehtre, M. S. Kankanhalli und W. F. Lee. “Shape Measures for Content Based Image Retrieval: A Comparison.” *Pattern Recognition*, 33, Nr. 3, Seiten 319–337, 1997.

[Mer00] B. Merialdo, S. Marchand-Maillet und B. Huet. “Approximate Viterbi Decoding For 2D-Hidden Markov Models.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 2147–2150, 2000.

[Min96] T. P. Minka. “Markovian Models for Sequential Data.” Technischer Bericht 1049, Dept. Informatique et Recherche Operationelle, Universite de Montreal, Montreal, Canada, 1996.

[Min99] T. P. Minka. “From Hidden Markov Models to Linear Dynamical Systems.” Technischer Bericht TR-531, MIT, 1999.

[Mul98a] S. Müller, S. Eickeler und G. Rigoll. “Image Database Retrieval of Rotated Objects by User Sketch.” In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL-98), in conjunction with CVPR ’98*, Seiten 40–44. Santa Barbara, USA, Juni 1998.

[Mul98b] S. Müller, G. Rigoll, A. Kosmala und D. Mazurenok. “Invariant Recognition of Hand-Drawn Pictograms Using HMMs with a Rotating Feature Extraction.” In *6th Intern. Workshop on Frontiers in Handwriting Recognition*, Seiten 25–34. Taejon, Korea, August 1998.

[Mul98c] S. Müller, G. Rigoll, D. Mazurenok und D. Willett. “Invariante Erkennung handskizzierter Piktogramme mit Anwendungsmöglichkeiten in der inhaltsorientierten Bilddatenbankabfrage.” In *20. DAGM-Symposium Tagungsband Springer-Verlag*, Seiten 271–279. Stuttgart, Germany, September 1998.

[Mul99a] S. Müller, S. Eickeler, C. Neukirchen und B. Winterstein. “Segmentation and Classification of Hand-Drawn Pictograms in Cluttered Scenes – An Integrated Approach.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 3489–3492. Phoenix, März 1999.

[Mul99b] S. Müller, S. Eickeler und G. Rigoll. “Multimedia Database Retrieval Using Hand-Drawn Sketches.” In *International Conference on Document Analysis and Recognition (ICDAR)*, Seiten 289–292. Bangalore, India, September 1999.

[Mul99c] S. Müller, S. Eickeler und G. Rigoll. “Pseudo 3-D HMMs for Image Sequence Recognition.” In *Proceedings IEEE Intern. Conference on Image Processing (ICIP)*, Seiten 237–241. Kobe, Japan, Oktober 1999.

[Mul99d] S. Müller und G. Rigoll. “Improved Stochastic Modeling of Shapes for Content-Based Image Retrieval.” In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL-99), in conjunction with CVPR ’99*, Seiten 23–27. Fort Collins, USA, Juni 1999.

[Mul99e] S. Müller und G. Rigoll. “Searching an Engineering Drawing Database for User-specified Shapes.” In *International Conference on Document Analysis and Recognition (ICDAR)*, Seiten 697–700. Bangalore, India, September 1999.

[Mul99f] S. Müller, G. Rigoll, A. Kosmala und D. Mazurenok. “Combining Shape Matrices and HMMs.” In S.-W. Lee, Herausgeber, *Advances in Handwriting Recognition*, Seiten 519–528. World Scientific Publishing, Berlin, Germany, 1999.

[Mul99g] S. Müller, F. Wallhoff, S. Eickeler und G. Rigoll. “Content-based Retrieval of Digital Archives Using Statistical Object Modeling Techniques.” In *Electronic Ima-*

*ging & the Visual Arts (EVA99)*, Seiten 12/1–12/4. Berlin, Germany, November 1999.

[Mul00a] S. Müller, S. Eickeler und G. Rigoll. "Crane Gesture Recognition Using Pseudo 3-D Hidden Markov Models." In *Proceedings IEEE Intern. Conference on Automatic Face and Gesture Recognition*, Seiten 398–402. Grenoble, France, März 2000.

[Mul00b] S. Müller und G. Rigoll. "Engineering Drawing Database Retrieval Using Statistical Pattern Spotting Techniques." In A. K. Chhabra und D. Dori, Herausgeber, *Graphics Recognition: Recent Advances*, Seiten 246–255. Springer Verlag, Berlin, 2000.

[Mul01] S. Müller, S. Eickeler und G. Rigoll. "An Integrated Approach to Shape and Color-Based Image Retrieval of Rotated Objects Using Hidden Markov Models." *International Journal of Pattern Recognition and Artificial Intelligence, Special Issue on Hidden Markov Models in Vision*, 15, Nr. 1, Seiten 223–238, Februar 2001.

[MM99] S. Marchand-Maillet. "1D and Pseudo-2D Hidden Markov Models for Image Analysis, Theoretical Introduction." Technischer Bericht MMWP-99xx, Department of Mulitmedia Communications, EURECOM Institute, Sophia-Antipolis, März 1999.

[Mur00] K. P. Murphy. "A Brief Introduction to Graphical Models and Bayesian Networks." Technischer Bericht, Berkeley University, 2000. http://www.cs.berkeley.edu/murphyk/Bayes/bayes.html.

[Nat93] K. S. Nathan, J. R. Bellegarda, D. Nahamoo und E. J. Bellegarda. "On-line Handwriting Recognition Using Continuous Parameter Hidden Markov Models." In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Band 5, Seiten 121–124. Minneapolis, 1993.

[Neu98] C. Neukirchen, D. Willett, S. Eickeler und S. Müller. "Exploiting Acoustic Feature Correlations by Joint Neural Vector Quantizer Design in a Discrete HMM System." In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 5–8. Seattle, Mai 1998.

[Pal93] N. R. Pal und S. K. Pal. "A Review on Image Segmentation Techniques." *Pattern Recognition*, 26, Nr. 9, Seiten 1277–1294, 1993.

[Par80] A. Parrish. *Mechanical Engineer's Reference Book*. Butterworth, London, 1980.

[Pen94] A. Pentland, R. W. Picard und S. Sclaroff. "Photobook: Content-Based Manipulation of Image Databases." In *Proc. SPIE Storage and Retrieval Image and Video Databases II*, Nummer 2185, Februar 1994.

[Plu91] M. D. Plumbley. *An Information-Theoretic Approach to Unsupervised Connectionist Models*. Doktorarbeit, Cambridge University, 1991.

[Rab76] L. R. Rabiner, J. G. Wilpon und B.-H. Juang. "A Segmental K-Means Training Procedure for Connected Word Recognition." *AT&T Technical Journal*, 64, Nr. 4, Seiten 21–40, Mai 1976.

[Rab89] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE*, 77, Nr. 2, Seiten 257–286, Februar 1989.

[Rig96] G. Rigoll, A. Kosmala und M. Schuster. "A New Approach to Video Sequence Recognition Based on Statistical Methods." In *Proceedings IEEE Intern. Conference on Image Processing (ICIP)*, Seiten 839–842. Lausanne, September 1996.

[Rig97] G. Rigoll und A. Kosmala. "New improved Feature Extraction Methods for Real-Time High Performance Image Sequence Recognition." In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 2901–2904. Munich, April 1997.

[Rig99a] G. Rigoll, S. Müller und B. Winterstein. "Robust Person Tracking with Non-Stationary Background Using a Combined Pseudo-2D-HMM and Kalman-Filter Approach." In *Proceedings IEEE Intern. Conference on Image Processing (ICIP)*, Seiten 242–246. Kobe, Japan, Oktober 1999.

[Rig99b] G. Rigoll, B. Winterstein und S. Müller. "Robust Person Tracking in Real Scenarios with Non-Stationary Background Using a Statistical Computer Vision Approach." In *IEEE International Workshop on Visual Surveillance in conjunction with CVPR-99*, Seiten 41–47. Fort Collins, USA, Juni 1999.

[Rig00] G. Rigoll und S. Müller. "Graphics-Based Retrieval of Color Image Databases Using Hand-Drawn Query Sketches." In A. K. Chhabra und D. Dori, Herausgeber, *Graphics Recognition: Recent Advances*, Seiten 256–265. Springer Verlag, Berlin, 2000.

[Ros90] R. C. Rose und D. B. Paul. "A Hidden Markov Model Based Keyword Recognition System." In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 129–132. Albuquerque, 1990.

[Row96] H. Rowley, S. Baluja und T. Kanade. "Neural Network-Based Face Detection." In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Seiten 203–207. San Francisco, CA, 1996.

[Sab97] R. Sabourin, J.-P. Drouhard und E. S. Wah. “Shape Matrices as a Mixed Shape Factor for Off-line Signature Verification.” In *Proceedings International Conference on Document Analysis and Recognition (ICDAR)*, Seiten 661–665. Ulm (Germany), 1997.

[Sam94a] F. Samaria und A. Harter. “Parameterisation of a Stochastic Model for Human Face Identification.” In *IEEE Workshop on Applications of Computer Vision*. Sarasota, Florida, Dezember 1994.

[Sam94b] F. S. Samaria. *Face Recognition Using Hidden Markov Models*. Doktorarbeit, Engineering Department, Cambridge University, Cambridge, Oktober 1994.

[San96] S. Santini und R. Jain. “Similarity Queries in Image Databases.” In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 646–651. San Francisco, Juni 1996.

[Sch96] M. Schuster und G. Rigoll. “Fast Online Video Image Sequence Recognition with Statistical Methods.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 3450–3453. Atlanta, 1996.

[Sim88] T. Simchony und R. Chellappa. “Stochastic and Deterministic Algorithms for MAP Texture Segmentation.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 1120–1123, 1988.

[Smi97] J. Smith und S.-F. Chang. “An Image and Video Search Engine for the World-Wide Web.” In *Symposium on Electronic Imaging: Science and Technilogy - Storage & Retrieval for Image and Video Databases V*, Seiten 84–95, Februar 1997.

[Smy97] P. Smyth. “Belief Networks, Hidden Markov Models, and Markov Random Fields: A Unified View.” *Pattern Recognition Letters*, , Nr. 18, Seiten 1261–1268, 1997.

[ST95] E. G. Schukat-Talamazzini. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*. Künstliche Intelligenz. Vieweg, Braunschweig, 1995.

[Sta98] T. Starner, J. Weaver und A. Pentland. “Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, Nr. 12, Seiten 1371–1375, 1998.

[Taz89] A. Taza und C. Y. Suen. “Discrimination of Planar Shapes Using Shape Matrices.” *IEEE Transactions on Systems, Man, and Cybernetics*, 19, Nr. 5, Seiten 1281–1289, Sep/Okt 1989.

[Teh88] C.-H. Teh und R. T. Chin. “On Image Analysis by the Methods of Moments.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, Nr. 4, Seiten 496–512, 1988.

[Wal98] F. Wallhoff. *Aufbau eines Systems zur rotationsinvarianten Erkennung von isolierten Objekten in natürlichen Bildern*. Studienarbeit, Fachbereich Elektrotechnik, Fachgebiet Technische Informatik, Gerhard-Mercator-Universität Duisburg, Januar 1998.

[Wel95] G. Welch und G. Bishop. “An Introduction to the Kalman Filter.” Technischer Bericht TR95-041, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC, USA, 1995.

[Win98] B. Winterstein. *Eine systematische Untersuchung von Pseudo-2D-HMM Techniken für das Auffinden von Personen in realen Bildszenen*. Studienarbeit, Fachbereich Elektrotechnik, Fachgebiet Technische Informatik, Gerhard-Mercator-Universität Duisburg, Oktober 1998.

[Woo96] J. Wood. “Invariant Pattern Recognition: A Review.” *Pattern Recognition*, 29, Nr. 1, Seiten 1–17, 1996.

[Yal00a] I. K. Yalcin. *Gesture Recognition Using Pseudo 2D and Pseudo 3D Hidden Markov Models*. Diplomarbeit, Fachbereich Elektrotechnik, Fachgebiet Technische Informatik, Gerhard-Mercator-Universität Duisburg, Mai 2000.

[Yal00b] I. K. Yalcin, A. T. Kilinc, S. Müller und G. Rigoll. “Gesture Recognition Using Pseudo 3D Hidden Markov Models.” In *22. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 420–427. Kiel, Germany, September 2000.

[Yam92] J. Yamato, J. Ohya und K. Ishii. “Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model.” In *Proc. IEEE Int. Conference on Computer Vision and Pattern Recognition*, Seiten 379–385, 1992.

[You92] S. J. Young. “The General Use of Tying in Phoneme-Based HMM Speech Recognisers.” In *Proceedings IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 569–572, 1992.

[You94] S. J. Young. “The HTK Hidden Markov Toolkit: Design and Philosophy.” Technischer Bericht, Cambridge University, September 1994.