

# 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

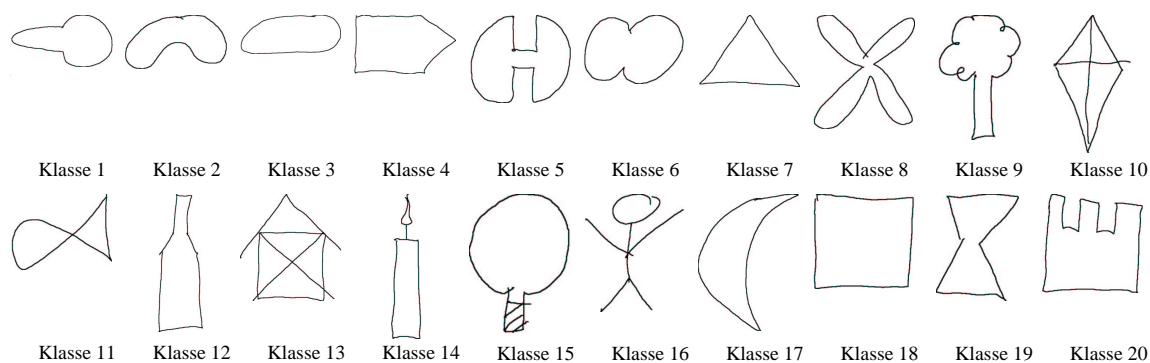


Abbildung 3.1: Beispiele aus den verwendeten 20-Klassen-Piktogramm-Datenbasen. Die ersten acht Klassen werden auch in [He91] und [Lee94] verwendet.

Art der Merkmalsextraktion 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 Merkmalsextraktion 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 Schlüsselwörtern 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 Merkmalsextraktion beschrieben.

## 3.2 Merkmalsextraktion

Eine Merkmalsextraktion 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 Merkmalsextraktion 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  $\vec{O} = \{\vec{o}_1, \dots, \vec{o}_T\}$  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:

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

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:

$$I_s(m, n) = I(m \cdot \Delta r, n \cdot \Delta \varphi) \quad (3.2)$$

$$m = 0, \dots, M - 1$$

$$n = 0, \dots, N - 1$$

- 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

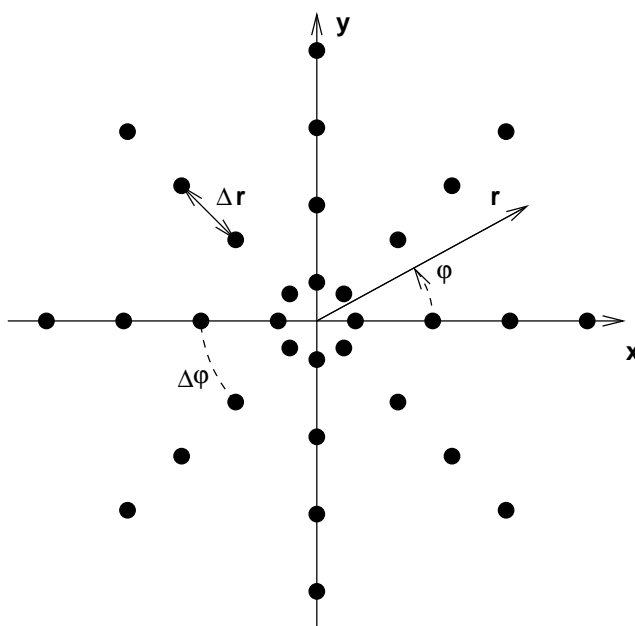


Abbildung 3.2: Schematische Darstellung der polaren Abtastung

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:

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

In obiger Gleichung bezeichnet der Operator  $\text{round}(\ )$  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:

$$R = \frac{\text{Anzahl an Abtastwerten}}{\text{Anzahl an Bildpunkten}} \quad (3.4)$$

$$R = \frac{\text{Anzahl an Abtastwerten}}{\text{Umfang des Kreises}} \quad (3.5)$$

$$R = \frac{c}{\text{Umfang des Kreises}} \quad (3.6)$$

$$R = \frac{c}{2\pi \cdot r} \quad (3.7)$$

$$R = \frac{c^*}{r} \quad (3.8)$$

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

$$R \propto \frac{1}{r} \quad (3.9)$$

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 ( $n = 0, \dots, N - 1$ ). 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:

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

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  $\vec{O} = \{\vec{o}_1, \dots, \vec{o}_T, \vec{o}_{T+1}, \dots, \vec{o}_{2T}\}$  mit  $\vec{o}_i = \vec{o}_{i+T}$  für  $i = 1, \dots, T$  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  $\varphi^*$  in Grad zu

$$\varphi^* = \frac{f_1}{f_1 + f_2} \cdot 360^\circ \quad (3.11)$$

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

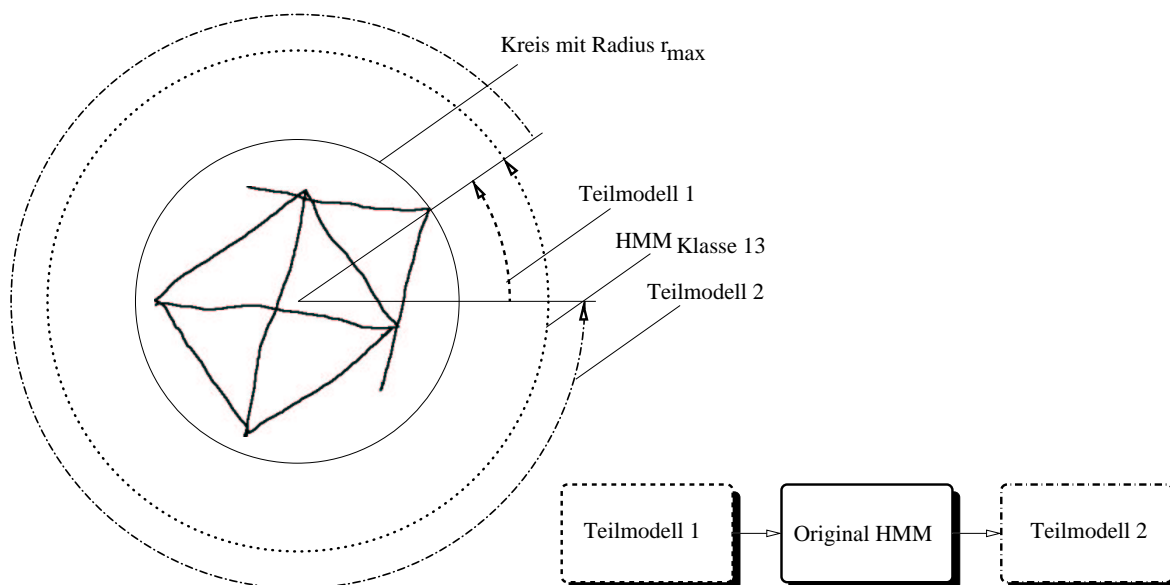


Abbildung 3.3: Zuordnung der Elemente der zweimalig präsentierten Merkmalsequenz zu den beiden Teilmodellen und dem Hidden-Markov-Modell für die Klasse 13

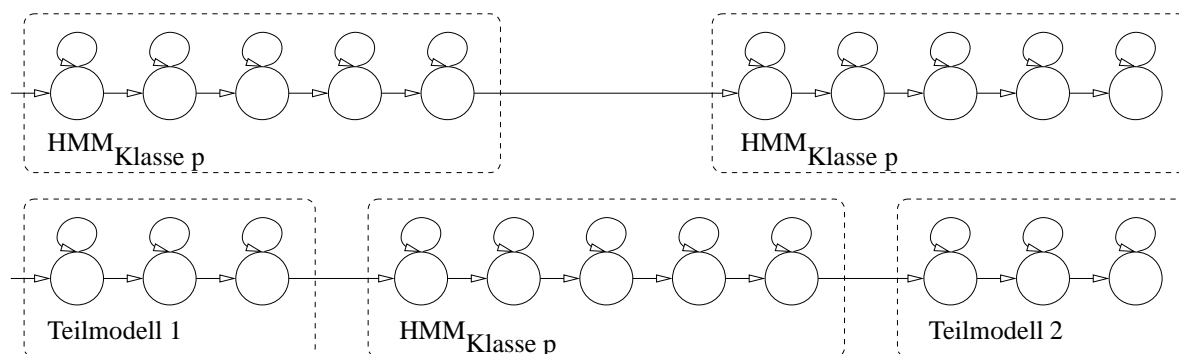


Abbildung 3.4: Zwei Hidden-Markov-Modelle repräsentieren eine einzige Piktogramm-Klasse. Das obere HMM modelliert unrotierte, das untere rotierte Piktogramme.

[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  $\{\vec{o}_1, \dots, \vec{o}_{T/2}\}$  und  $\{\vec{o}_{T/2+1}, \dots, \vec{o}_T\}$  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

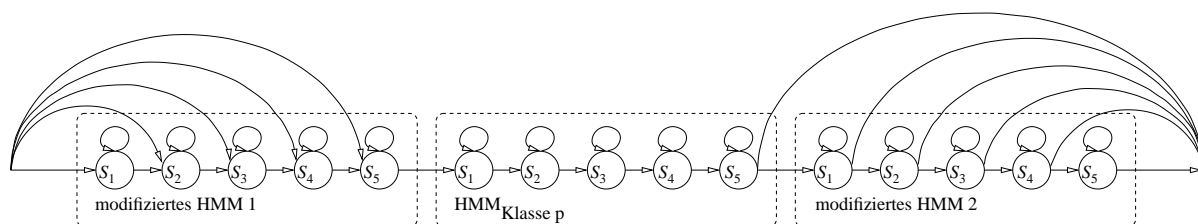


Abbildung 3.5: Bei der zweiten vorgestellten Methode werden drei identische, jedoch bezüglich der Wahrscheinlichkeiten für die Anfangs- und Endzustände veränderte, Hidden-Markov-Modelle verwendet.

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  $\{\vec{o}_1, \dots, \vec{o}_T\}$ . 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  $0^\circ$ , das Modell darunter einen Winkel von  $360^\circ/N$ , usw. Die Winkel können bei diesem Ansatz also nur in Vielfachen von  $360^\circ/N$  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.

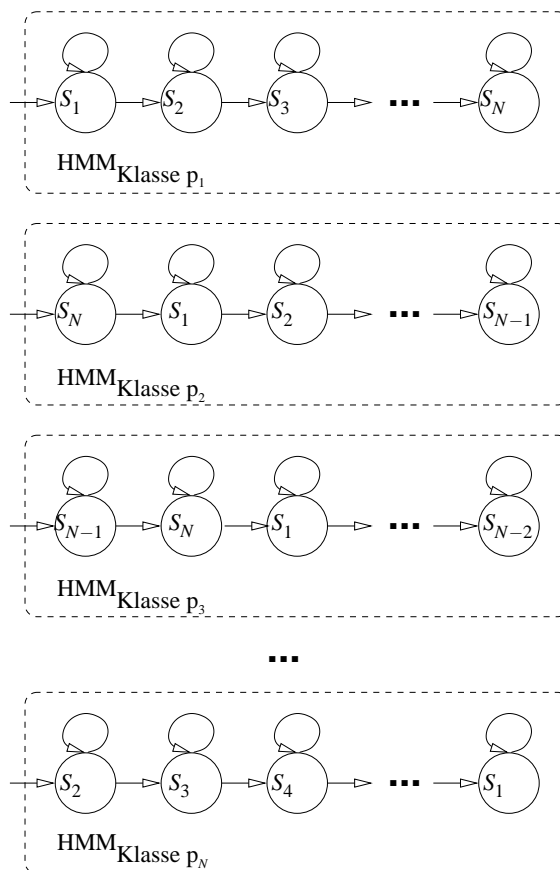


Abbildung 3.6: Zyklisches Vertauschen der HMM-Zustände

### 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.

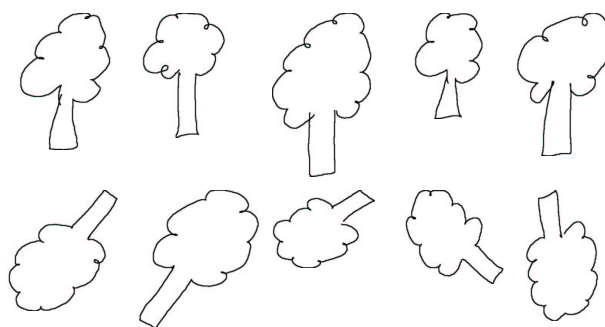


Abbildung 3.7: Fünf unrotierte und fünf rotierte Piktogramme der Klasse 9 aus der *stm*-Datenbank.

### 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	<i>stm</i>	<i>dib</i>	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	<b>Kap. 3.3.3</b>	<b>99,5%</b>	<b>99,5%</b>	<b>99,5%</b>
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)	<b>Kap. 3.3.3</b>	<b>97,5%</b>	<b>98,5%</b>	<b>98,0%</b>

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 rotationsinvarianten 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 gesehene 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 ungenaueren 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:

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

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

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

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

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

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:

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

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

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

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	<i>stm</i>	<i>dib</i>	Mittelwert
7 Hu-Momente	52,0%	46,5%	49,25%
<b>52 Li-Momente</b>	<b>99,0%</b>	<b>99,0%</b>	<b>99,0%</b>
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 rotationsinvarianter 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 Bilddatenbanken 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, Computermaus 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.

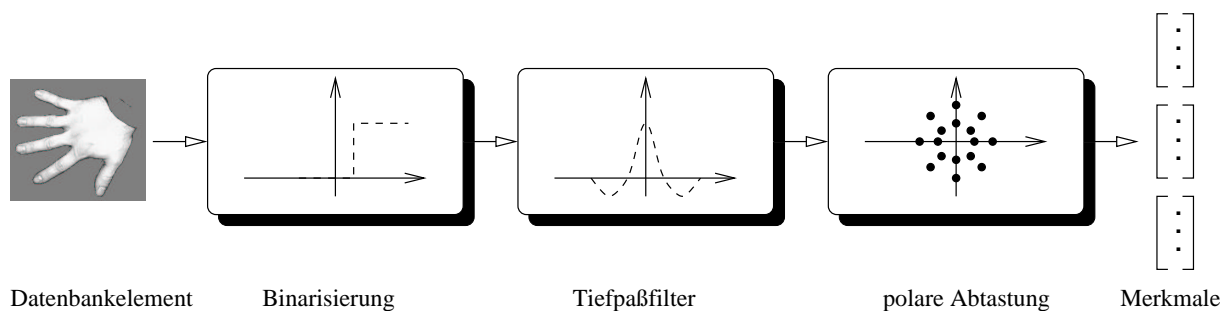


Abbildung 3.8: Vorverarbeitung und Erzeugung der Merkmalsequenz

### 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 Merkmalsextraktion 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 Merkmalsextraktion kann im wesentlichen wie in Kapitel 3.2 dargestellt durchgeführt wurden. Dabei ist jedoch zu beachten, daß nun Grauwertbilder anstelle 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-

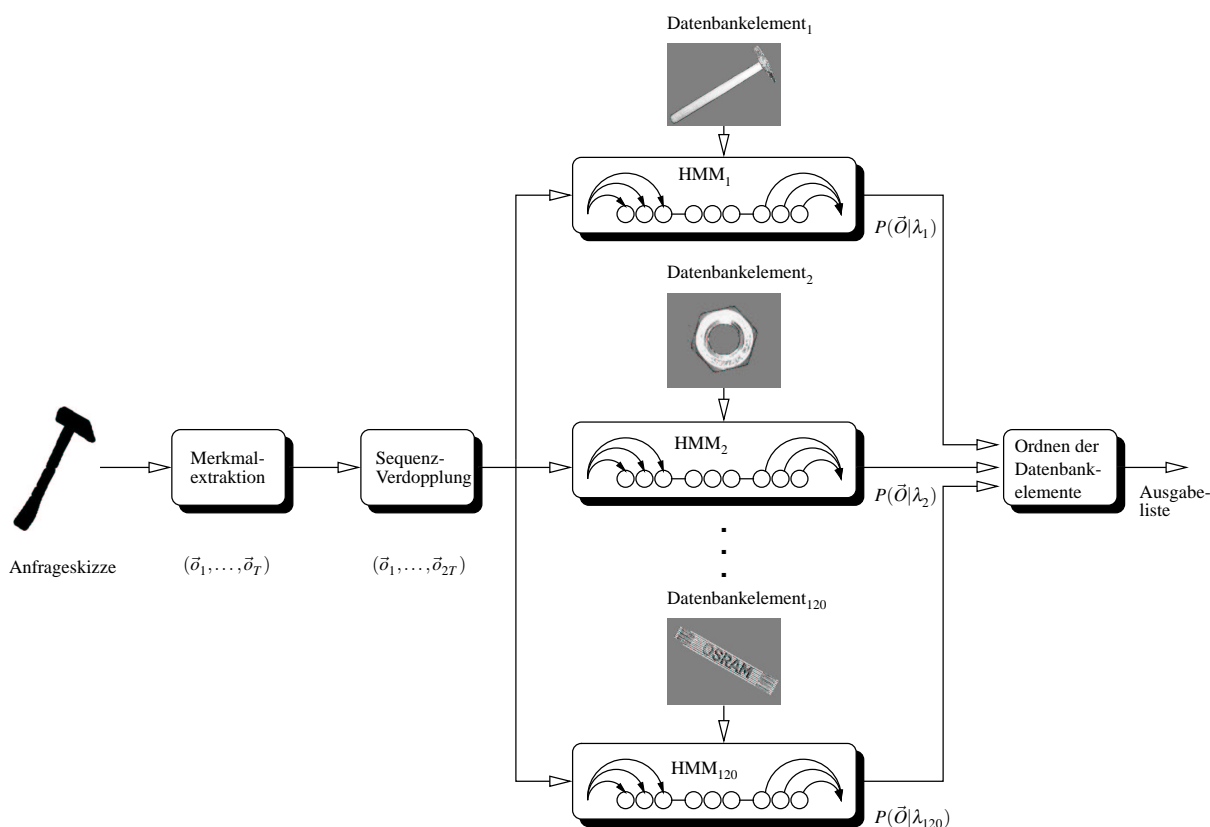


Abbildung 3.9: Schematische Darstellung des skizzenbasierten Bilddatenbankabfragesystems

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 Merkmalsextraktion 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-

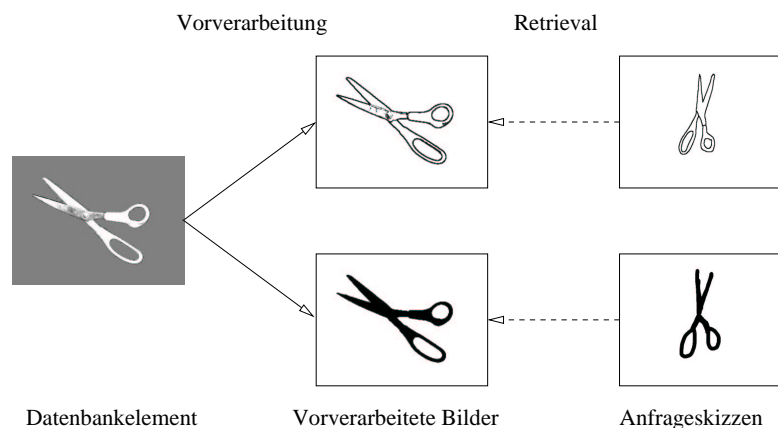


Abbildung 3.10: Abfrageskizzen und entsprechend vorverarbeitete Datenbankelemente

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  $\vec{O} = \{\vec{o}_1, \dots, \vec{o}_{2T}\}$  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]

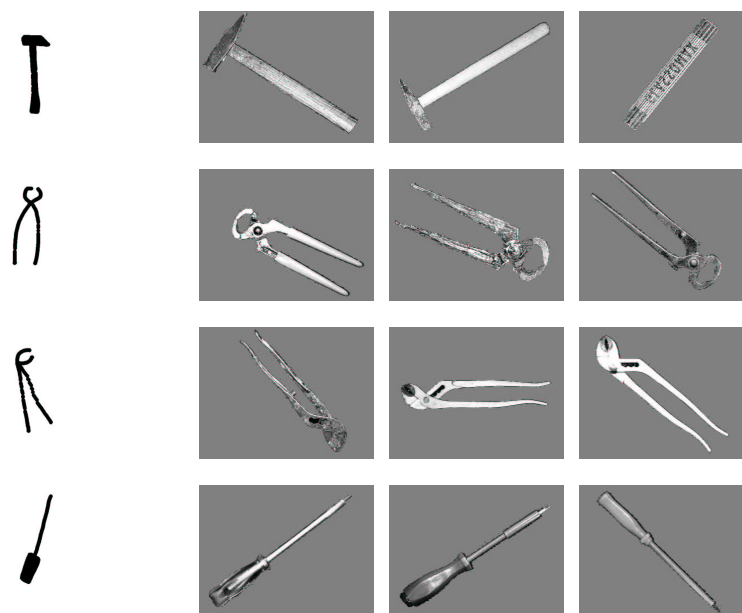


Abbildung 3.11: Anfrage-Skizzen und die vom Datenbanksystem ermittelten drei ähnlichsten Grauwertbilder

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 Abfrage entsprechen: *Zeige alle Datenbankelemente, die eine Zange enthalten, die blaue Griffe und einen grauen Kopf hat.*

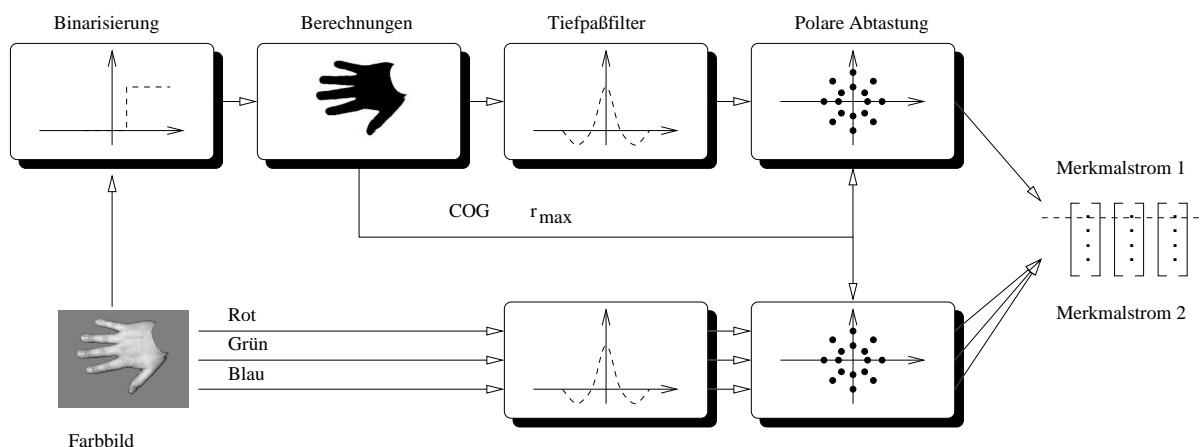


Abbildung 3.12: Vorverarbeitung und Erzeugung der Merkmalsequenz mit zwei Merkmalströmen

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 Farbkanaäle 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 Farbkanaäle 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 Merkmalsextraktion 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 Bild-datenbank 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 rotationsinvariante, statistische Modellierung. Bei Betrachtung der aneinandergefügt 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  $\Delta\phi = 10^\circ$ ). 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  $256 \times 256$  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.






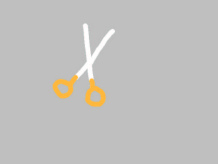









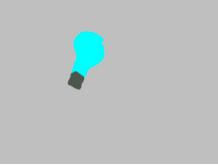











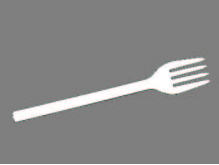


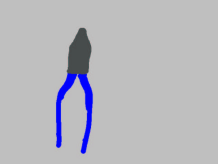


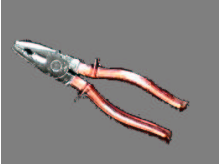

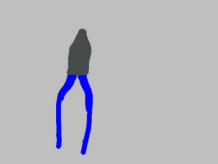




				
Skizze 1	( 22, 14, 220° )	( 28, 8, 280° )	( 31, 5, 310° )	( 11, 25, 110° )
				
Skizze 2	( 8, 28, 80° )	( 1, 35, 10° )	( 30, 6, 300° )	( 7, 29, 70° )
				
Skizze 3	( 6, 30, 60° )	( 28, 8, 280° )	( 28, 8, 280° )	( 5, 31, 50° )
				
Skizze 4	( 26, 10, 260° )	( 19, 17, 190° )	( 11, 25, 110° )	( 2, 34, 20° )
				
Skizze 5	( 14, 22, 140° )	( 35, 1, 350° )	( 2, 34, 20° )	( 14, 22, 140° )
				
Skizze 6	( 35, 1, 350° )	( 28, 8, 280° )	( 10, 26, 100° )	( 28, 1, 347° )
				
Skizze 7	( 24, 12, 240° )	( 32, 4, 320° )	( 6, 30, 60° )	( 22, 14, 220° )
				
Skizze 7	( 24, 12, 240° )	( 25, 11, 250° )	( 11, 25, 110° )	( 18, 18, 180° )

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 misslungen, 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 Beispielen 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 Glühlampe 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 widerspiegeln. Die erste in Abbil-

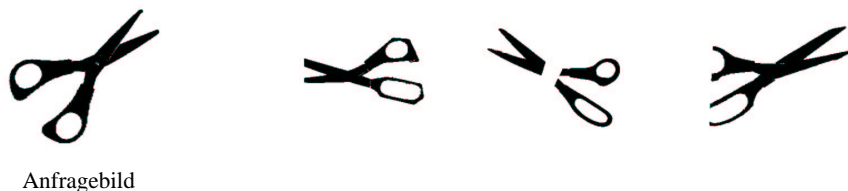


Abbildung 3.15: Künstlich deformierte Formen

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:

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

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 Merkmalsextraktion, 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 an 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.