Technische Universität Berlin
Institut für Angewandte Informatik
Fachgebiet Methoden der Künstlichen Intelligenz

Segmentierung mit Gaborfiltern zur Induktion struktureller Klassifikatoren auf Bilddaten

Diplomarbeit

Ralf Herbrich

Oktober 1997

Aufgabensteller
Prof. Dr. Fritz Wysotzki

Betreuer
Tobias Scheffer

Zweitgutachter
Dr. Ute Schmid

Zusammenfassung

In dieser Diplomarbeit wird die Möglichkeit untersucht, ein Bild mit Hilfe maschineller Lernverfahren überwacht zu segmentieren. Dazu werden aus einer vorgegebenen Menge von segmentierten Beispielbildern durch Gaborfilteranwendung Merkmalsbilder berechnet. Anschließend wird eine Segmentierungsprozedur in Form von Tests über die Merkmalsbilder gelernt, die das Segmentierungsproblem bestmöglich löst. Das Segmentierungsproblem ist durch die Problemverteilung in den Beispielbildern gegeben. Der präsentierte Ansatz unterscheidet sich von früheren Arbeiten (Michie et al. 1994) darin, daß er parameterlos ist. Dies bedeutet, daß für ein gegebenes Problem mit Hilfe von Merkmalsselektion das beste Modell zu dessen Lösung gesucht wird. Empirisch evaluiert wird dieser Segmentierungsalgorithmus am Problem der Textursegmentierung von Brodatztexturen sowie der Segmentierung von Röntgenbildern. Darüber hinaus wird eine mögliche Anwendungen der Segmentierung zur Induktion eines strukturellen Klassifikators zur Gefäßwanderkennung auf Zellbildern untersucht. Hierbei steht allerdings die Verwendung struktureller Merkmale bei gegebener optimaler Segmentierung im Vordergrund.

Ich versichere hiermit, daß ich die vorliegende Diplomarbeit selbständig verfaßt habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden.

Berlin, den 22. Oktober 1997

Danksagungen

Ich möchte allen an dieser Diplomarbeit Beteiligten danken. Besonderer Dank gebührt dabei Hauke Bartsch und Christian Piepenbrock für die unzähligen fruchtbaren Diskussionen und ihre ständige Hilfsbereitschaft. Meinen Betreuern Ute Schmid und Tobias Scheffer und meinem Aufgabensteller Prof. Wysotzki, der immer ein offenes Ohr für mich hatte, möchte ich ebenfalls meinen herzlichsten Dank aussprechen. Desweiteren bedanke ich mich bei Dr. Hufnagl für die nette Zusammenarbeit und die Bereitstellung des Zellbilderdatensatzes. Letztendlich möchte ich mich auch bei Eric Heymann, Jens Matthias, Jeannette Grossmann und Thomas Bendig für die vielen kleinen Ratschläge und die massive Unterstützung während der Endphase dieser Diplomarbeit bedanken.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

„Ein Bild sagt mehr als tausend Worte“. Jeder kennt diesen Satz. Für den Menschen ist es am einfachsten, Umweltinformationen durch Bilder wahrzunehmen. Versucht man Informationen dagegen mit Computern zu verarbeiten, so müßte die Aussage in „Tausend Worte sagen mehr als ein Bild“ umgekehrt werden. Informationen aus Bilder zu gewinnen — auch als Mustererkennung bezeichnet – ist eine der schwierigsten und am längsten untersuchten Aufgaben seit Anbeginn der Informatik in den 60–er Jahren (Pavlidis 1977). Aktuelle Anwendungsgebiete umfassen die Analyse medizinischer Bilddaten (Tönnies und Lemke 1994), Schrifterkennung (Seni et al. 1996), Gesichtererkennung (Wiskott und von der Malsburg 1995) oder Auswertung von Satellitenbildern (Michie et al. 1994). Allerdings ist bis heute noch keine dieser Aufgaben zufriedenstellend gelöst worden. So machen beispielsweise die besten Handschrifterkennungssysteme bis heute noch mehr als einen Fehler auf 100 Buchstaben handgeschriebene Blockschrift. Handgeschriebene Fließschrift zu erkennen ist mit bestehenden Systemen praktisch unmöglich.

Einer der wichtigsten Schritte in der Mustererkennung ist die Segmentierung des Bildes. Hier werden Bildbereiche mit einer gemeinsamen Eigenschaft zusammengefaßt. In den letzten Jahren sind eine Fülle von Segmentierungsalgorithmen entwickelt worden (Devijver 1986; Fiorio und Gustedt 1994; Campbell et al. 1996; Greenspan 1996; Weldon und Higgins 1997). Jeder Algorithmus macht zuerst eine Modellannahme über die Eigenschaften der Bildbereiche. Beispielsweise gehen die Algorithmen von Campbell et al. (1996) und Weldon und Higgins (1997) davon aus, daß diese Bereiche durch einige *Peaks* im Frequenzspektrum gekennzeichnet sind. Je nach gewählter Modellannahme besitzen die Verfahren dann Parameter, die für ein gegebenes Segmentierungsproblem angepaßt werden müssen. Algorithmen, die diese Parameter mit Hilfe gegebener Beispielsegmentierungen anpassen, werden überwachte Segmentierungsalgorithmen genannt.

Bei der Wahl des Modells ergibt sich ein Problem, welches im maschinellen Lernen als das *Bias*–Varianz Dilemma bekannt ist (Geman et al. 1992): Für ein einfaches Modell mit wenigen Parametern gibt es nur sehr wenige Segmentierungsprobleme, in denen mit Hilfe der angepaßten Parameter eine gute Segmentierung erreicht wird. Im allgemeinen wird die geringe Ausdrucksstärke des einfachen Modells zu einer schlechten Segmentierungsqualität führen. In komplexen Modellen mit vielen Parametern würde die große Ausdrucksstärke es ermöglichen, viele Segmentierungsprobleme bestmöglich zu lösen. Allerdings hat man in komplexen Modellen ein sehr große Varianz, d.h. mit hoher Wahrscheinlichkeit werden die Parameter aus den Beispielsegmentierungen sehr schlecht geschätzt. Alle bekannten Segmentierungsalgorithmen versuchen dieses Dilemma zu lösen, indem sie für ein spezielles Segmentierungsproblem das

einfachste Modell zu dessen Lösung benutzen. Diese Prinzip wird auch als *Occams razor* bezeichnet. Jeder Segmentierungsalgorithmus schränkt damit die Klasse der segmentierbaren Bilder sehr stark ein.

In dieser Arbeit wird ein überwachter Segmentierungsalgorithmuses vorgestellt, der über die Anpassung der Parameter hinaus aus einer gegebenen Menge von Modellen mit einem aus dem maschinellen Lernen bekannten Verfahren das beste Modell selektiert. Dazu wird zuerst die Bildinformation in einen hochdimensionalen Raum (komplexes Modell) transformiert. Das Ergebnis nach Parameteranpassung im besten Modell ist ein Regelbaum über Merkmale des hochdimensionalen Raumes. Dieser Regelbaum wird als Segmentierungsprozedur benutzt. In diesem Sinne ist der vorgestellte Segmentierungsalgorithmus ein Metaalgorithmus. Der Algorithmus in Greenspan (1996) verwendet ebenfalls einen Regelbaum zur Segmentierung. In diesem Algorithmus findet aber keine Suche nach dem besten Modell statt.

In dieser Arbeit wird sich auf Gaborfilterfaltungen zur Expansion der Bildinformation beschränkt. Gaborfilter wurden in der Vergangenheit erfolgreich zur Segmentierung texturierter Bilder eingesetzt (Bovik et al. 1990; Dunn et al. 1994; Bigun und Buf 1994; Weldon und Higgins 1997). Texturierte Bilder zeichnen sich dadurch aus, daß die Grauwerte in den einzelnen Bildbereichen sehr stark aber relativ regelmäßig variieren. In texturierten Bildern reicht die Grauwertinformation nicht zu Segmentierung aus. Durch Gaborfilter können aus dem Bild Informationen gewonnen werden, die nur im Frequenzspektrum sichtbar werden. Durch die große Anzahl an Parametern eines Gaborfilters müssen alle bekannten Algorithmen allerdings weitere, sehr restriktive, Modellannahmen treffen (auch als *texture models* bezeichnet (Vistnes 1988)). Dadurch nutzen diese Algorithmen die Mächtigkeit von Gaborfiltern zur Bildsegmentierung nicht vollständig aus. Texturierte Bilder sind vor allem in der praktischen Anwendung von großem Interesse, da reale Bilder oft aus Texturen zusammengesetzt sind.

Die Segmentierung zerlegt ein Bild in einzelne Bildbereiche. Diese Bereiche können als Knoten eines Graphen interpretiert werden. Damit kann der Bildinhalt durch einen Graphen beschrieben werden. Auf der anderen Seite kann der Bildinhalt aber auch durch einen Merkmalsvektor fester Länge beschrieben werden. Im einfachsten Fall repräsentiert jedes Pixel ein Merkmal (Lawrence et al. 1997). Aufgrund der verwendeten Repräsentation zur Mustererkennung unterscheidet man daher strukturelle und statistische Mustererkennung. Bei der strukturellen Mustererkennung hat man den Vorteil, Relationen zwischen den einzelnen Bereichen berücksichtigen zu können. Strukturelle Mustererkennungsverfahren sind in der Auswertung von Satellitenbildern (Eshera und Fu 1986) und Interpretation von Bauplänen (Bunke und Messmer 1995) erfolgreich angewendet worden. An einem gegebenem Problem — der Erkennung von Gefäßwandzellen auf Zellbildern — wird als Anwendungsgebiet der Segmentierung untersucht, ob strukturelle Mustererkennung mit den auf Graphen basierenden Regelbaumlernverfahren aus Scheffer et al. (1997b) und Geibel und Wysotzki (1997) zu besseren Ergebnissen als statistische Mustererkennung führt.

Die Arbeit ist in sechs Kapitel gegliedert. Nach dieser Einleitung wird im zweiten Kapitel der Prozeß der Mustererkennung vorgestellt. Besonderes Augenmerk wird auf die Segmentierung und Induktion struktureller Klassifikatoren gelegt. Da auf einem der verwendeten Datensätze versucht wurde, mit Hilfe struktureller Klassifikatoren ein gegebenes Problem zu lösen, liegt der Schwerpunkt dieses Kapitel darauf zu zeigen, in welchen Schritten strukturelle Klassifikatoren gefunden werden. Das Kapitel geht an notwendigen Stellen über die Grundlagen hinaus. Abgeschlossen wird es durch eine Einführung in Gaborfilter — ein Vorgriff auf Definitionen und Funktionen, die im Kapitel 4 benötigt werden. Das dritte Kapitel führt,

soweit als notwendig, ins Maschinelle Lernen ein. Dabei werden alle für den vorgestellten Segmentierungsalgorithmus irrelevanten Themen ausgeblendet. An einigen Stellen geht auch dieses Kapitel über Grundlagen hinaus. Im vierten Kapitel wird dann eine Kombination der in den beiden vorhergehenden Kapiteln vorgestellten Methoden zur parameterlosen Segmentierung präsentiert. Eine Beschreibung der Eigenschaften dieses Algorithmus, soweit diese theoretisch abgeleitet werden können, beenden dieses kurze Kapitel. Es schließt sich eine ausführliche Beschreibung und Auswertung der durchgeführten Experimente in Kapitel 5 an. Das letzte Kapitel faßt den Inhalt der Arbeit kurz zusammen und vergleicht die Ergebnisse mit denen aktueller Arbeiten. Außerdem werden zwei mögliche Erweiterungen des Segmentierungsalgorithmus vorgestellt. Beendet wird diese Arbeit mit einem Anhang, in dem neben einigen ausführlicheren mathematischen Formeln das implementierte System vorgestellt wird. Desweiteren finden sich hier alle durch Experimente erhaltenen Bilder. In Anhang C sind die in der Arbeit verwendeten Notationen in Listenform zusammengefaßt.

Kapitel 2

Mustererkennung und Gaborfilter

Dieses Kapitel beginnt mit einer allgemeinen Einführung in den Prozeß der Mustererkennung. Schwerpunkt wird auf die Teilprozesse Segmentierung und Formerkennung gelegt. Im weiteren werden zwei bekannte Verfahren der Primitivenextraktion aus Bildern, das Houghraumverfahren und das *Random minimal subset* Verfahren (RMS) (Roth und Levine 1993), vorgestellt. Eine Kombination dieser Methoden wird im Kapitel 5 an einem Datensatz untersucht. Vorgreifend auf Kapitel 4 werden Gaborfilter als eine spezielle Klasse von Filtern definiert und kurz deren Eigenschaften zur Texturanalyse vorgestellt.

2.1 Mustererkennung im Überblick

Der Prozeß der Mustererkennung ist durch gezielte Informationsreduktion gekennzeichnet. Der vollständige Prozeß ist in Abbildung 2.1 dargestellt. Die in dieser Abbildung dargestellten Funktionen und Daten sollen in den folgenden Abschnitten kurz vorgestellt werden. Der Schwerpunkt liegt dabei auf den beiden dunkleren Funktionsblöcken der Abbildung. Dieser Überblick lehnt sich an Pavlidis (1977, S. 6*ff*) an. Für eine tiefgreifende Beschäftigung wird auf Pavlidis (1977), Klette und Zamperoni (1995) und Pratt (1991) verwiesen.

Definition 2.1 (Bild). Ein Bild f ist eine partielle Abbildung des euklidischen zweidimensionalen Raumes.

$$f : \mathfrak{D} \to \mathfrak{R}$$

Bilder mit $\mathfrak{D} \subset I\!R^2$ und $\mathfrak{R} \subseteq I\!R^n$ werden analoge Bilder genannt. Gilt für Bilder $\mathfrak{D} \subset I\!N^2$, so spricht man von digitalen Bildern. Alle Bilder mit $n > 1$ werden als Mehrkanalbilder bezeichnet.

An erster Stelle steht bei der Mustererkennung immer die **Bildsynthese**. Analoge Bilder sind für die weitere Verarbeitung nicht geeignet, da ihr Definitionsbereich nicht endlich ist. Sie müssen in digitale Bilder transformiert werden. Dieser Prozeß wird Digitalisierung genannt. Andererseits können digitaler Bilder auch mit Verfahren der Computergrafik erzeugt werden. Aus einer zwei- bzw. dreidimensionalen Beschreibungen einer Szene wird mit Hilfe von Beleuchtungsmodellen die Intensität für jeden Pixel berechnet (Tönnies und Lemke 1994). Diese Bilder eignen sich besonders gut zur Mustererkennung, da sie im allgemeinen rauschfrei sind. In einigen Mustererkennungssystemen werden solche Bilder als Lerndaten benutzt (Schyns und Bulthoff 1993; Guerin-Dugue und Palagi 1994).

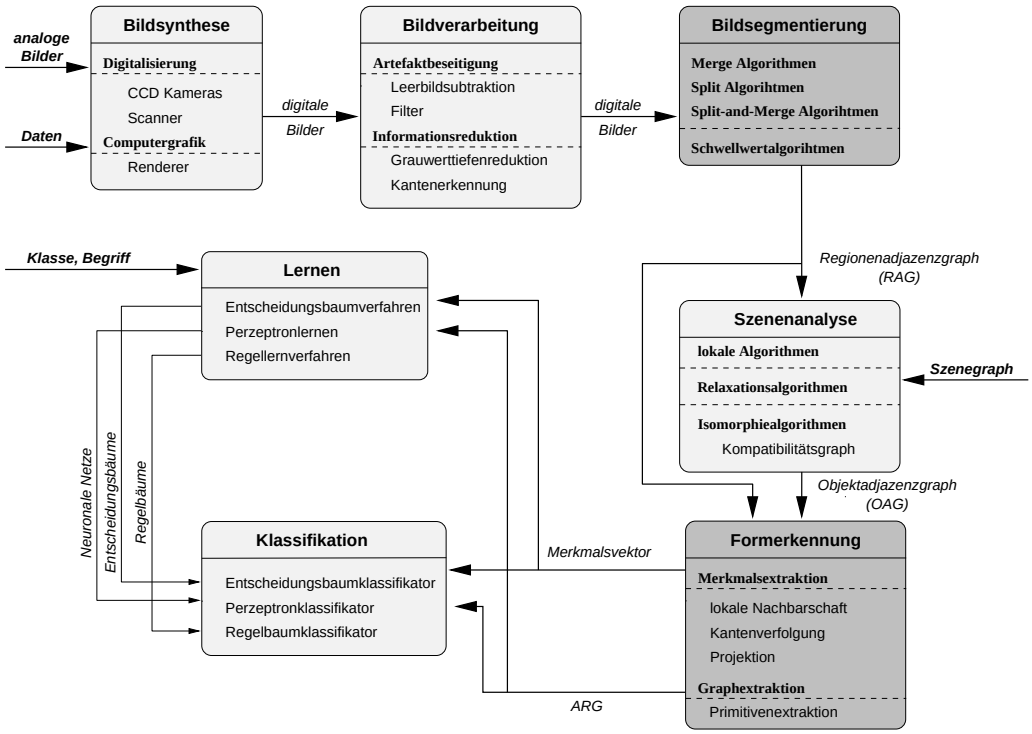


Abbildung 2.1: Allgemeine Arbeitsweise von Mustererkennungsverfahren. Durch Pfeile sind Datenströme und durch Kästen Funktionsblöcke dargestellt.

Nach der Bildsynthese werden in der **Bildverarbeitung** die Artefakte in den Bilddaten beseitigt. Artefakte entstehen aus Fehlern im Eingabegerät oder durch veränderte Lichteinflüße bei der Digitalisierung. Neben der Beseitigung von Artefakten kann aber auch die Informationsreduktion der Bilddaten von Interesse sein. Bekannte Funktionen zur Artefaktbeseitung und Informationsreduktion sind die Leerbildsubtraktion, Filterung (siehe hierzu auch Abschnitt 2.4.1), Grauwerttiefenreduktion sowie Kantenerkennung (Haralick und Lee 1988).

Aus den in Bildverarbeitung aufbereiteten Bilddaten wird in der **Bildsegmentierung** in eine einfachere Struktur, der Regionenadjazenzgraph (RAG), berechnet. Die Knoten dieses Graphen bilden Mengen von zusammenhängenden Pixeln (Regionen). Grenzen zwei Regionen im Bild aneinander, so existiert eine Kante im RAG. Im Hinblick auf das zu lernende Muster müssen die Regionen bei der Berechnung bestimmte Eigenschaften erfüllen (Segmentierungsprädikat). Dieser Teilprozeß wird im Abschnitt 2.2 eingehender vorgestellt.

Im nächsten Schritt der Mustererkennung, der **Szenenanalyse**, werden die Knoten des RAG mit Zusatzattributen versehen. Dazu muß explizit und datenabhängig ein sogenannter Szenegraph vorgegeben werden. Mit Hilfe von Relaxations- oder Isomorphiealgorithmen (Scheffer et al. 1997a) wird versucht, die Knoten des RAG und des Szenegraphen aufeinander abzubilden. Dadurch entsteht der sogenannte Objektadjazenzgraph (OAG). Die Szenenanalyse soll im Rahmen dieser Arbeit nicht weiter ausgebaut werden.

Im letzten Schritt der Mustererkennung, der **Formerkennung**, wird der RAG[1] in eine propo-

[1] bzw. OAG, wenn eine Szenenanalyse durchgeführt wurde

sitionale bzw. graphische Repräsentation überführt. Der RAG wird in einen n–dimensionalen Merkmalsvektor (propositionale Repräsentation) mittels Merkmalsextraktion überführt. Bekannte Extraktionsverfahren sind lokale Nachbarschaften (Lawrence et al. 1997), Kantenverfolgungsverfahren oder Projektionen. Wird der RAG dagegen in eine graphische Repräsentation überführt, entsteht der sogenannte attributierte Relationsgraph (ARG). Der ARG zeichnet sich dadurch aus, daß die Knoten und Kanten des RAG/OAG attributiert werden (Eshera und Fu 1986). Für die Knoten- und Kantenattribute werden Merkmalsvektoren verwendet. In (Eshera und Fu 1986) wird das Nachbarschafts- und Distanzattribut als Kantenmerkmal verwendet. In Abschnitt 2.3 werden zwei Verfahren zur Extraktion von Knotenmerkmalen vorgestellt. Bei der graphischen und propositionalen Repräsentation hängen die verwendeten Merkmale im allgemeinen sehr stark vom Datensatz ab.

Im Anschluß wird die extrahierte, stark informationsreduzierte Repräsentation des Bildes bzw. einer Region benutzt, um bei gegebener Klasse mit Hilfe maschineller **Lernverfahren** eine Wissensstruktur für die einzelnen Klassen aufzubauen. Nicht zum Lernen benutzte Bilder können dann mit Hilfe der gelernten Wissensstruktur automatisch klassifiziert werden. Die Beschreibung der Klasse (Muster) ist durch die gelernte Wissensstruktur gegeben. Eine detailierte Einführung in maschinelle Lernverfahren folgt in Kapitel 3.

Aufgrund der verwendeten Wissensstruktur unterscheidet man statistische und strukturelle Mustererkennungsverfahren. Bei den statistischen Mustererkennungsverfahren werden die n–dimensionalen, aus dem Bild extrahierten, Merkmalsvektoren benutzt, um mit Hilfe eines gelernten Klassifikators den Merkmalsvektor — und damit das Bild oder eine Region — einer Klasse zuzuordnen. Bei den strukturellen Mustererkennungsverfahren wird der RAG in eine graphische Repräsentation überführt. Eine adäquate Repräsentationsform stellen Klauseln des Prädikatenkalküls dar, welche als Lerndaten für ILP–System wie FOIL (Quinlan 1990) oder GOLEM (Muggleton und Cao 1990) dienen.

Der Nachteil der statistischen Mustererkennungsverfahren ist im allgemeinen die Vernachlässigung relationaler Beziehungen zwischen einzelnen Regionen des Bildes. Relationale Beziehungen können teilweise mit Hilfe von Komplexmerkmalen berücksichtigt werden. So sind in (Brunelli und Poggio 1993) 23 der 35 benutzten Merkmale zur Gesichtererkennung Komplexmerkmale, die Relationen zwischen den Regionen beschreiben. Der Vorteil bei statistischen Mustererkennungsverfahren ist die kleine Größe des dadurch entstehenden Hypothesenraumes — verglichen mit der Größe des Hypothesenraumes bei graphischer Repräsentation. Dadurch gilt, daß bei gleichem Fehler auf den Trainingsdaten die Hypothese aus dem kleineren Hypothesenraum mit höherer Wahrscheinlichkeit wirklich besser ist als die Hypothese mit gleichem Trainingsfehler aus dem größerem Hypothesenraum (Blumer et al. 1986) (eine Präzisierung dieser Aussage folgt in Kapitel 3).

In der Literatur findet sich auch noch eine dritte Gruppe von Mustererkennungsverfahren, die syntaktischen Mustererkennungsverfahren. Diese zeichnen sich dadurch aus, daß als Wissensstruktur eine Grammatik verwendet wird. Beschreibt man die ARGs aber über Klauseln im Prädikatenkalkül, dann kann die syntaktische Mustererkennung als Teilklasse der strukturellen Mustererkennung angesehen werden (Bunke 1986). Aus diesem Grunde soll nur zwischen statistischer und struktureller Mustererkennung unterschieden werden.

2.2 Bildsegmentierung

Die in der Bildverarbeitung aufbereiteten Daten sollen in der Bildsegmentierung in eine einfachere Struktur, den Regionenadjazenzgraph (RAG) transformiert werden. Der RAG definiert sich folgendermaßen.

Definition 2.2 (Regionenadjazenzgraph). Gegeben sei ein digitales Bild $f : \mathfrak{D} \to \mathfrak{R}$. Dann wird jeder Graph $R_f = (V, E, f_v)$ mit

$$f_v(v) \subseteq \mathfrak{D} \qquad \forall v \in V$$

$$f_v(v_1) \cap f_v(v_2) = \emptyset \qquad \forall v_1, v_2 \in V \wedge v_1 \neq v_2 \tag{2.1}$$

$$\bigcup_{v \in V} f_v(v) = \mathfrak{D} \tag{2.2}$$

$$(v_1, v_2) \in E \Leftrightarrow \exists p \in f_v(v_1) \bullet \mathcal{N}(p) \cap f_v(v_2) \neq \emptyset \tag{2.3}$$

Regionenadjazenzgraph des Bildes f genannt. Dabei ist $\mathcal{N} : \mathfrak{D} \to \mathcal{P}(\mathfrak{D})$ eine Abbildung jedes Punktes auf seine Nachbarn. Übliche Nachbarschaften sind die 4– und 8–Nachbarschaft. Jedes Element der Menge V wird Region genannt.

Die Bedingung (2.3) sagt aus, daß eine Kante zwischen zwei Knoten des RAG existiert gdw beide Regionen aneinandergrenzen. Die Bedingung (2.1) fordert eine eindeutige Zuordnung eines Bildpunktes zu genau einer Region. Außerdem muß im RAG jeder Punkt einer Region zugeordnet werden (Bedingung (2.2)). Bei der Berechnung des RAG werden die Kanten meist explizit nach der eigentlichen Segmentierung berechnet.

Ziel eines Segmentierungsverfahrens ist die Minimierung der Anzahl der Knoten des RAG unter Beachtung von Nebenbedingungen. Diese Nebenbedingungen sind Prädikate über Regionen.

Definition 2.3 (Segmentierung). Gegeben sei ein Segmentierungsprädikat $P : \mathcal{P}(\mathfrak{D}) \to$ *bool*. Gesucht ist ein RAG $R = (V, E, f_v)$ mit folgenden Eigenschaften:

$$P(f_v(v)) = true \qquad \forall v \in V \tag{2.4}$$

$$\|V\| \stackrel{!}{=} Min. \tag{2.5}$$

Das Problem der Segmentierung wird im allgemeinen iterativ gelöst. Je nachdem, mit welchem RAG initial begonnen wird, unterscheidet man folgende Segmentierungsverfahren.

Splitverfahren. Bei Splitverfahren beginnt man mit

$$R = (\{1\}, \emptyset, \{(1, \mathfrak{D})\})$$

Es wird ein RAG mit nur einem Knoten benutzt, um nun sukzessive die Regionen zu zerteilen. Dieser Prozeß wird solange wiederholt, bis die Bedingung (2.4) erfüllt ist. Da nicht jede mögliche Zerteilung der Segmente probiert werden kann — es existieren insgesamt 2^{NM} Möglichkeiten, terminiert das Verfahren in suboptimalen Lösungen.

Mergeverfahren. Bei Mergeverfahren beginnt man mit einem RAG in dem jeder Pixel als Region interpretiert wird. Nun werden sukzessive Regionen gemischt. Der Prozeß endet, wenn Bedingung (2.4) erfüllt ist. Auch dieses Verfahren terminiert in suboptimalen Lösungen, wobei dieses Verfahren die Bedingung (2.5) im allgemeinen schlechter erfüllt als Splitverfahren. Es ist anfälliger für Rauschen im Bild.

Split- und Mergeverfahren. Bei diesem Verfahren wird zuerst ein Splitschritt durchgeführt und anschließend das Mergeverfahren angewandt. Dies hat den Vorteil, daß durch das Mischen die Anzahl der Knoten des RAG verkleinert wird. Da im Mergeverfahren als initialer RAG nun schon ein RAG benutzt wird, bei dem Rauschen eliminiert wurde, ist das Mergeverfahren auch nicht mehr anfällig für das Rauschen im Bild. Auch dieses Verfahren liefert nicht die optimale Segmentierung. Trotzdem ist dieses Verfahren das beste, da es auf jeden Fall weniger (bzw. gleich viele) Regionen findet, als nur das Split- oder Mergeverfahren.

Ist das Segmentierungsprädikat P separierbar, d.h. das Segmentierungsprädikat ist für zwei vermischte Regionen genau dann wahr, wenn es auch für jede einzelne Region wahr ist, dann läßt sich das Problem der Segmentierung mittels pixelweiser Überprüfung von P in linearer Zeit lösen. Typischerweise fallen Schwellwertalgorithmen in diese Gruppe von Segmentierungsverfahren.

Definition 2.4 (Schwellwertalgorithmus). Gegeben sei ein Bild $f : \mathfrak{D} \to \mathfrak{R}$ und eine Menge $T = \{t_1, t_2, \ldots, t_k\}$ von k Intervallen in $\mathfrak{R}$, die vollständig $\mathfrak{R}$ abdecken. Dann ist das Ergebnis f_t eines Schwellwertalgorithmus definiert über

$$f_t(x, y) = i \quad \text{gdw } f(x, y) \in t_i$$

Der RAG wird aus f_t konstruiert, indem genau k Knoten erzeugt werden und der Bildpunkt (x, y) dem Knoten i zugeordnet wird gdw $f_t(x, y) = i$. Die Kanten des RAG können anschließend über die Nachbarschaft $\mathcal{N}$ berechnet werden.

2.3 Primitivenextraktion

Ein stark diskriminierendes Merkmal für die Knoten des ARG ist der Typ der geometrischen Primitive der entsprechenden Primitive. Im folgenden sollen deshalb kurz die beiden bekanntesten Verfahren zur Extraktion von Primitiven vorgestellt werden.

Houghraumverfahren Das Houghraumverfahren wurde von Hough (1962) erstmalig vorgestellt. Das Verfahren basiert auf der Houghtransformation des Bildes f.

Definition 2.5 (Houghtransformation). Sei eine geometrische Primitive gegeben durch $\boldsymbol{ax} = 0$. Dabei bestehen die Komponenten von $\boldsymbol{x}$ ausschließlich aus Termen über x und y. Dann bildet die Houghtransformation jeden Punkt (x, y) des Bildes f auf die Menge der Punkte $\boldsymbol{a}$ ab, die $\boldsymbol{ax} = 0$ erfüllen. Im diskretisierten Houghraum wird für jede Primitive $\boldsymbol{a}$ die Häufigkeit der Bildpunkte (x, y) gespeichert, die über die Houghtransformation auf $\boldsymbol{a}$ abgebildet werden.

Nachdem die Transformation des Bildes ausgeführt wurde, wird im diskretisierten Houghraum nach lokalen Maxima gesucht. Jedes lokale Maxima definiert eine zu extrahierende Primitive. Dieses Verfahren ist für höherdimensionale Primitiven (Kreis, Ellipse, usw.) sehr aufwendig, da der Aufwand exponentiell mit der Dimensionalität der Primitiven wächst. Desweiteren können durch schlechte Diskretisierung des Houghraums Primitiven extrahiert werden, die nicht im Bild enthalten waren (Stockman und Agrawala 1977).

RMS–Verfahren Das RMS–Verfahren wurde von Diaconis und Efron (1983) vorgeschlagen und von Roth und Levine (1993) auf das Problem der Primitivenextraktion angewandt. Im Gegensatz zum Houghraumverfahren wird beim RMS–Verfahren nicht für jeden Bildpunkt (x, y) die Menge aller Primitiven $\boldsymbol{a}$ berechnet, sondern eine minimale Teilmenge $\{(x, y)\}$ ausgewählt, die genau eine Primitive beschreiben. Dann läßt sich folgendes Lemma leicht zeigen (Roth und Levine 1993).

Lemma 2.1 (Anzahl minimaler Teilmengen). Gegeben seien N zweidimensionale Datenpunkte. Es werden jeweils minimale Teilmengen mit genau R Elementen gezogen. Es soll die beste Primitive — definiert über die Distanz jedes Datenpunktes zur Primitive — mindestens $\varepsilon \cdot N$ Punkte enthalten und mit einer Wahrscheinlichkeit von s aus allen Primitiven mit mindestens $\varepsilon \cdot N$ Punkten gefunden werden. Dann gilt für die Anzahl K von minimalen Teilmengen, die aus den N Datenpunkten zufällig ohne Zurücklegen gezogen werden müssen:

$$K = \frac{\ln(1-s)}{\ln(1-\varepsilon^R)}$$

Das kombinatorische Problem der Suche nach der besten Primitive $\boldsymbol{a}$ — im allgemeinen müßten $\binom{N}{R}$ unterschiedliche Teilmengen gezogen werden, um mit $s = 1.0$ die beste Primitive zu extrahieren — wird durch die *a priori* festgesetzte Wahrscheinlichkeit ε stark reduziert. In Abbildung 2.2 ist für $s = 0.95$ die Abhängigkeit dargestellt. Dabei reichen die R–Werte von 2 (Linie) bis 6 (allgmeine Ellipse). Wie man sieht, müssen selbst unter der Annahme, daß nur 50% der Datenpunkte auf der Ellipse liegen, maximal 180 Teilmengen zufällig gezogen werden. Ohne diese Annahme müßten für 100 Datenpunkte bereits 1 132 449 780 solcher Teilmengen zur Extraktion betrachtet werden!

2.4 Gaborfilter

2.4.1 Filter und Faltung

Filter spielen eine zentrale Rolle in der Bildverarbeitung. Über die Faltung eines Bildes f mit einem Filter ist eine direkte Manipulation des Bildes im Frequenzraum möglich. Dies ermöglicht z.B. die Reduktion von Rauschen oder die Maskierung einer Textur. Ein Filter wird im Ortsraum durch ein Bild repräsentiert.

Definition 2.6 (Faltung). Gegeben sei ein Bild f und ein Filter g. Dann ist die Faltung des Bildes f mit dem Filter g (geschrieben als $h = f \otimes g$) definiert als:

$$h(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) g(x - i, y - j)\, didj$$

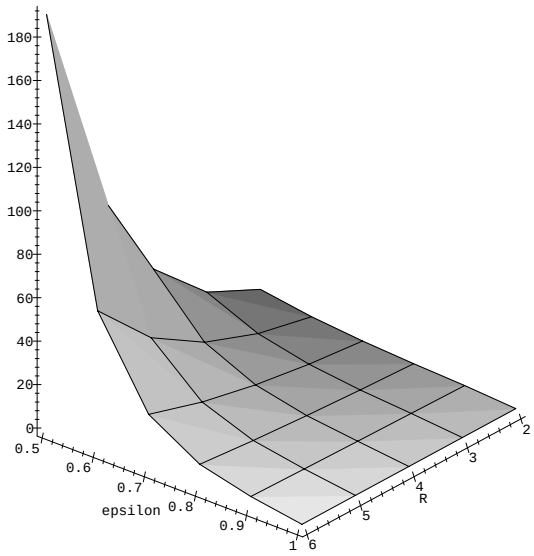


Abbildung 2.2: Abhängigkeit der Anzahl minimaler Teilmengen von ε und R bei $s = 0.95$

Das Filter g wird also, bildlich gesprochen, über das Bild f geschoben und maskiert die Funktionswerte von f über seine Funktion g. Ein besseres Verständnis der Faltung ergibt sich erst mit dem Faltungssatz (Gaskill 1978).

Satz 2.1 (Faltungssatz). Gegeben sei ein Bild f und ein Filter g. Dann gilt[2]:

$$h = f \otimes g = \mathfrak{F}_2^{-1}(\mathfrak{F}_2^1(f) \cdot \mathfrak{F}_2^1(g))$$

bzw.

$$H = F \cdot G$$

Betrachtet man also die Fouriertransformation G des Filters und des Bildes F, so entspricht eine Faltung einer komponentenweisen Multiplikation des Bildes im Frequenzraum. Durch die inverse Fouriertransformation erhält man dann das gefaltete Bild. Die Wirkungsweise von Filtern wird aus diesem Grund immer im Frequenzraum angegeben.

2.4.2 Motivation für Gaborfilter

Gaborfilter sind eine spezielle Klasse von Filtern, die erstmals in Gabor (1946) vorgestellt wurden. In dieser Arbeit wurde gezeigt, daß diese Funktionen die Eigenschaft haben, die Unschärfe c in der Relation $\Delta t \Delta f \geq c$ zu minimieren. In Daugman (1985) wurde diese Eigenschaft für die zweidimensionale Erweiterung dieser Filter bewiesen. Diese Eigenschaft

[2]Das Symbol $\mathfrak{F}_n^1$ bezeichnet die n–dimensionale Fouriertransformation. Für eine ausführliche Einführung in die Fouriertransformation wird auf Pratt (1991) verwiesen. Analog bezeichnet $\mathfrak{F}_n^{-1}$ die n–dimensionale inverse Fouriertransformation.

wird im folgenden Abschnitt detailierter erklärt und führt dazu, daß eine Signal schon mit sehr wenigen Gaborfiltern in der harmonischen Analyse approximiert werden kann.

Hubel und Wiesel (1962) fanden heraus, daß die rezeptiven Felder einzelner Zellen im V1, dem sogenannten primären visuellen Kortex, mathematisch mit Gaborfiltern modelliert werden können. In Abbildung 2.3 ist schematisch die Informationsverarbeitung vom Auge zum primären visuellen Kortex dargestellt. Jede Zelle im V1 reagiert auf Muster, die an bestimmten Stellen der Retina angeboten werden (linke Abbildung). Diese Zellen sind orientierungsselektiv und besitzen eine bevorzugte Frequenz im Frequenzraum [3].

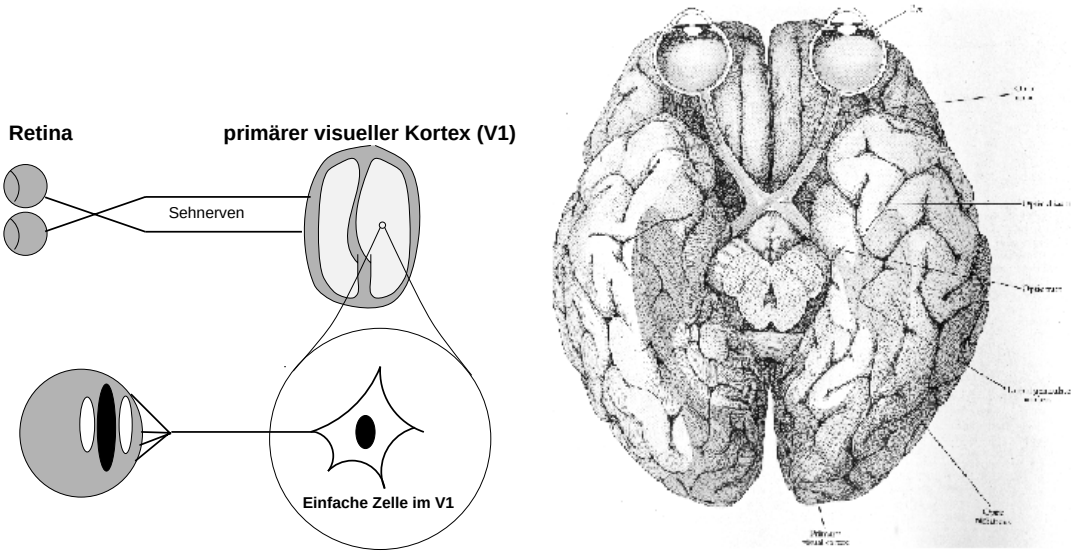


Abbildung 2.3: **Links:** Abstraktes Modell der Informationsverarbeitung in der menschlichen visuellen Wahrnehmung. Jede Zelle im V1 (unten rechts) ist mit retinalen Ganglienzellen verbunden, deren Verbindungsmuster sehr gut durch Gaborfilter approximiert werden kann (unten links). **Rechts:** Detailiertes Modell der Gehirnregionen, die an der menschlichen visuellen Wahrnehmung beteiligt sind (aus Hubel (1995)).

2.4.3 Grundlagen

Ein Gaborfilter ist eine komplexe Sinuswelle, die durch eine Gaußglocke moduliert wird. Ein zweidimensionales Gaborfilter hat somit die Gestalt

$$h(x,y) = \underbrace{g_{(\sigma_x,\sigma_y)}(x',y')}_{\text{Gaußglocke}} \cdot \underbrace{e^{2\pi \boldsymbol{j}(u_0 x + v_0 y)}}_{\text{komplexe Sinuswelle}}$$

[3] Darüber hinaus sind die Zellen des V1 auch augen- und richtungsselektiv (Hubel und Wiesel 1962).

wobei

$$g_{(\sigma_x,\sigma_y)}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)}$$
$$x' = (x-\mu_x)\cos(\theta) + (y-\mu_y)\sin(\theta)$$
$$y' = -(x-\mu_x)\sin(\theta) + (y-\mu_y)\cos(\theta)$$

Eine Faltung mit einem Gaborfilter transformiert das Bildsignal derart, daß die Unschärfe in Orts- und Frequenzraum minimiert wird. Zum besseren Verständnis dieser Aussage sollen folgende zwei Extremfälle betrachtet werden:

1. $\sigma_x, \sigma_y \to \infty$: In diesem Fall wird aus der Gaußglocke die konstante Funktion $g(x,y) = 1$ und das Gaborfilter $h(x,y)$ wird damit zur Funktion $h(x,y) = e^{2\pi \boldsymbol{j}(u_0 x + v_0 y)}$. Dies ist genau die Basisfunktion der Fouriertransformation. In diesem Extremfall berechnet das Gaborfilter die Koeffizienten der Fouriertransformation des Bildes bzw. die Repräsentation des Bildes im Frequenzraum.

2. $\sigma_x, \sigma_y \to 0$: In diesem Fall wird aus der Gaußglocke die sogenannte Diracfunktion $\delta(x-\mu_x, y-\mu_y)$, ein Impuls an der Stelle (μ_x, μ_y). Das Filter berechnet damit genau die Koeffizienten an diesen Stellen. Dies entspricht aber genau den Grauwerten bzw. der Ortsraumrepräsentation des Bildes.

In diesem beiden Extremfällen ist die Unschärfe in jeweils Orts- bzw. Frequenzraum unendlich groß. Jedes andere Filter verkleinert somit das Produkt der Unschärfen.

Unter Berücksichtigung von Eulers Formel

$$e^{2\pi \boldsymbol{j}(u_0 x + v_0 y)} = \cos(2\pi(u_0 x + v_0 y)) + \boldsymbol{j}\sin(2\pi(u_0 x + v_0 y))$$

kann das komplexe Filter auch als

$$h(x,y) = \underbrace{h_e(x,y)}_{\text{Realteil}} + \boldsymbol{j}\ \underbrace{h_o(x,y)}_{\text{Imaginärteil}}$$

mit dem Real- und Imaginärteil

$$h_e(x,y) = g_{(\sigma_x,\sigma_y)}(x',y') \cdot \cos(2\pi(u_0 x + v_0 y))$$
$$h_o(x,y) = g_{(\sigma_x,\sigma_y)}(x',y') \cdot \sin(2\pi(u_0 x + v_0 y))$$

geschrieben werden. Diese Eigenschaft kann zur schnelleren Berechnung benutzt werden. Außerdem ist diese Eigenschaft auch biologisch nachweisbar, denn es gibt immer ein korrespondierendes Paar von Zellen im V1, die jeweils Real- und Imaginärteil eines Filters darstellen (Tan 1995; MacLennan 1992).

Die besondere Eigenschaft dieser Filter im Kontext der Texturanalyse wird sichtbar, wenn man die Fouriertransformierte eines Gaborfilters betrachtet. O.b.d.A. kann $\mu_x = \mu_y = 0$ gesetzt werden.

$$\begin{aligned}\mathfrak{F}_2(h) &= \mathfrak{F}_2(g_{(\sigma_x,\sigma_y)}) \otimes \mathfrak{F}_2(e^{2\pi \boldsymbol{j}(u_0 x + v_0 y)}) \\ &= \frac{1}{\sigma_x\sigma_y} g_{(1/\sigma_x, 1/\sigma_y)}(u' - 2\pi u_0, v' - 2\pi v_0)\end{aligned}$$

wobei

$$u' = u\cos(\theta + \pi/2) + v\sin(\theta + \pi/2)$$
$$v' = -u\sin(\theta + \pi/2) + v\cos(\theta + \pi/2)$$

Die Fouriertransformierte ist eine Gaußglocke mit den inversen Standartabweichungen, deren Mittelpunkt im Punkt $(2\pi u_0, 2\pi v_0)$ liegt. In Abschnitt 2.4.1 wurde bereist gezeigt, daß die Wirkungsweise der Faltung der komponentenweisen Multiplikation von Filter- und Bildtransformation im Frequenzraum entspricht. Durch eine Faltung eines Bildes mit einem Gaborfilter ist man somit in der Lage eine Frequenz[4] $(2\pi u_o, 2\pi v_0)$ aus dem ursprünglichen Bild zu extrahieren, d.h. im gefalteten Bild gibt es an den Stellen im Bild einen signifikanten *Peak*, die zu einer Schwingung der Frequenz $(2\pi u_0, 2\pi v_0)$ gehören. Jedes Gaborfilter ist auf eine bevorzugte Frequenz $(2\pi u_0, 2\pi v_0)$ parametrisiert. Da das Bildsignals im Frequenzraum mit einer Gaußglocke moduliert wird, streut die „Antwort" des Filters im gefalteten Bild in Abhängigkeit von den Gaußglockenparametern. Die inverse Beziehung zwischen Streuung im Orts- und Frequenzraum, d.h. ein Gaborfilter, das sehr breit im Ortsraum ist, ist sehr lokal im Frequenzraum, ist schon in vorherigen Arbeiten gezeigt worden[5].

Unter der „Antwort" wird die Amplitude des im allgemeinen komplexen Faltungsbildes verstanden. Wenn f_h das aus f durch Faltung mit h entstandene Bild bezeichnet, dann ist die Amplitude von $f_h(x, y) = a + \boldsymbol{j}b$ als $m_{f_h} = \sqrt{a^2 + b^2}$ und die Phase als $\phi_{f_h} = \arctan b/a$ definiert. In (Dunn et al. 1994) wird gezeigt, daß die Amplitude ausreichende Information zur Texturanalyse liefert. Trotzdem zeigen sich an (künstlichen) Bildern auch Erfolge, wenn die Phase des Faltungsbildes benutzt wird (Landraud und Oh Yum 1995; Bovik et al. 1990).

Exemplarisch ist der Real- und Imaginärteil eines Gaborfilters in Abbildung 2.4 dargestellt. Die Fouriertransformation dieses Gaborfilters ist in Abbildung 2.5 dargestellt.

2.4.4 Gaborfilter in der Bildverarbeitung

In der Bildverarbeitung gibt es hauptsächlich zwei Anwendungen für Gaborfilter. In MacLennan (1991) und Lee (1996) werden Gaborfilter zur Bildkompression benutzt. Weitaus verbreiteter ist aber die Benutzung von Gaborfiltern zur Texturanalyse (Pichler et al. 1996), Textursegmentierung (Bovik et al. 1990; Dunn et al. 1994; Weldon und Higgins 1997) und Objekterkennung (Heidemann und Ritter 1996; Wiskott und von der Malsburg 1995).

Bildkompression Hier wird versucht, durch Faltung eines Bildes mit einer ausgewählten Menge von Filtern — einer sogenannten Filterbank — den Bildinhalt zu komprimieren. Jede orthogonale Menge von Basisfunktionen[6] ist geeignet, um ein Bildsignal verlustfrei zu transformieren. Gaborfilter sind zwar nicht orthogonal, aber durch ihre parametrisierbare Lokalität im Ortsraum gibt es Mengen von Filtern, die fast orthogonal sind. Solche Filterbänke werden *gabor wavelets* genannt. Die bei der Bildkompression verwendeten Filterbänke müssen nach dem Kriterium der geringsten Überlappung ausgewählt werden. Dazu wird der Frequenzraum

[4] Wegen des konstanten Faktors von 2π wird diese Frequenz auch als Radialfrequenz bezeichnet.

[5] Hier wird die sogenannte Nyquistfrequenz, die maximal abtastbare Frequenz in einem Bild, invers zur Größe des Bildes im Ortsraum definiert (Shannon 1948).

[6] Für eine Definition von Orthogonalität wird auf Gaskill (1978) oder andere Standartwerke der Signalverarbeitung verwiesen.

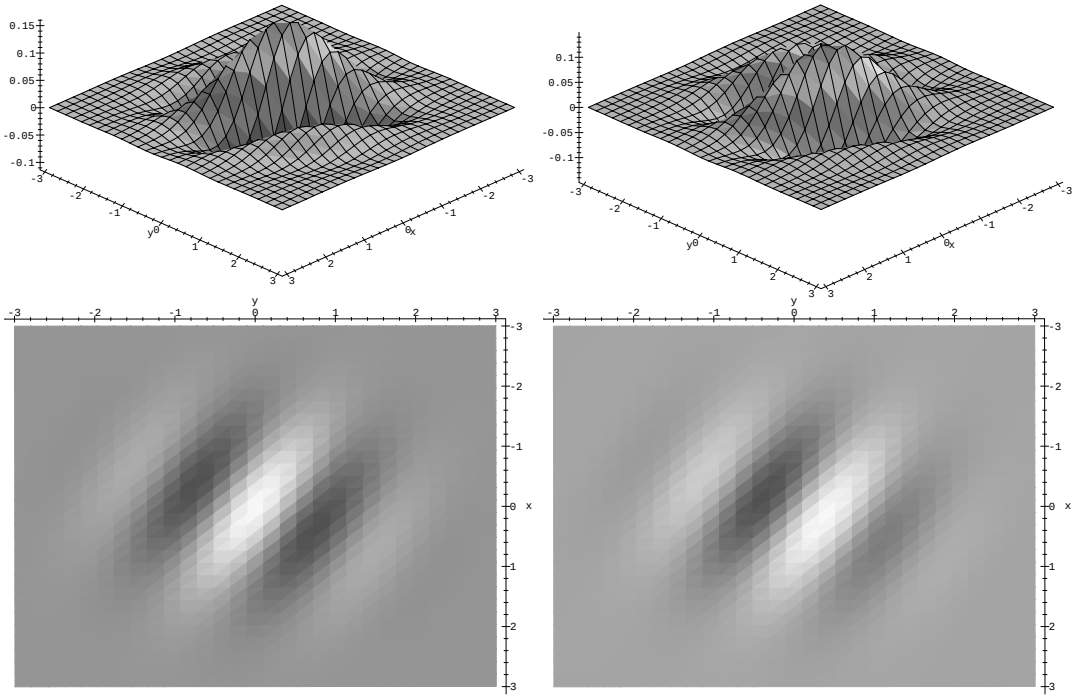


Abbildung 2.4: Real- und Imaginärteil eines Gaborfilters mit den Parametern $\mu_x = \mu_y = 0, \sigma_x = \sigma_y = 1, u_0 = 0.25, v_0 = 0.5, \theta = 0°$. **Oben:** Darstellung des Gaborfilters als zweidimensionale Funktion. **Unten:** Darstellung des Gaborfilters als digitales Bild (Draufsicht)

mit Hilfe der Filterbank systematisch abgetastet. Es bildet sich die sogenannte Rosette im Frequenzraum (siehe Abbildung 5.2).

Bild- oder Texturanalyse Hier wird durch Faltung eines Bildes mit einem Gaborfilter versucht, Regionen gleicher Textur hervorzuheben. Diese Faltung selbst wird meist auch als Texturanalyse bezeichnet (Pichler et al. 1996). Eine analysierbare Textur muß dabei dadurch gekennzeichnet sein, daß sie deutliche *Peaks* in ihrem Frequenzspektrum hat (siehe Abbildung 2.6). Faltet man nun das Bild mit einem Gaborfilter, das auf die Frequenz parametrisiert ist, in der es einen deutlichen *Peak* im Frequenzspektrum der Textur gibt, so „antwortet" dieses Filter an all den Stellen im Faltungsbild, an denen die Textur im ursprünglichen Bild enthalten ist (Bovik et al. 1990; Weldon und Higgins 1997). Im allgemeinen wird dann mit Hilfe eines Schwellwertalgorithmuses die Region im Bild ausmaskiert, an der im Faltungsbild hohe Amplitudenwerte vorlagen. Zu beachten bleibt hier allerdings, daß niedrige Frequenzen nur von im Ortsraum breiten Gaborfilter erkannt werden können, was selbst wieder dazu führt, das auch die Amplituden sehr breit streuen (Tan 1995). Um auch Texturen zu unterscheiden, die durch mehr als einen *Peak* gekennzeichnet sind, wird in Weldon und Higgins (1997) eine ganze Filterbank benutzt. Die Amplitudenkombinationen, die kennzeichnend für eine Textur sind, werden mit einem Bayesianischen Netz gelernt. Kennzeichnend für beide Ansätze ist die Tatsache, daß — im Gegensatz zur Bildkompression — dem Bildinhalt angepaßte Filter oder Filterbänke gesucht werden (sogenannte optimale Filter).

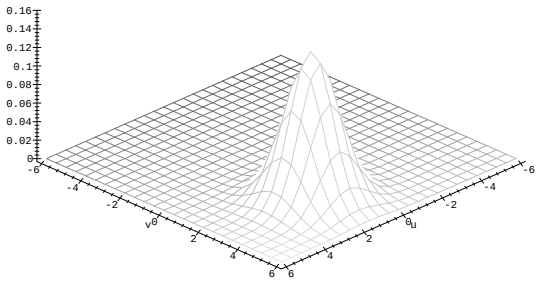
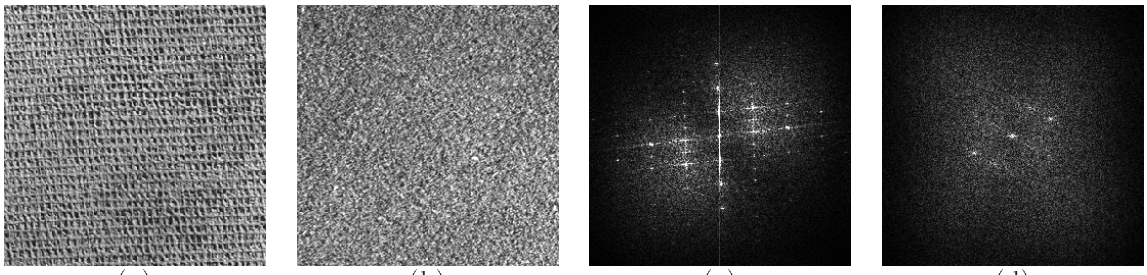


Abbildung 2.5: Fouriertransformation eines Gaborfilters mit den Parametern $\mu_x = \mu_y = 0, \sigma_x = \sigma_y = 1, u_0 = 0.25, v_0 = 0.5$.



(a) (b) (c) (d)

Abbildung 2.6: Zwei Texturen und deren Frequenzspektrum **(a)** `d17` aus Brodatz (1966). **(b)** `d18` aus Brodatz (1966) **(c)** Frequenzspektrum von (a). Deutlich zu sehen sind die kennzeichnenden *Peaks* im Frequenzspektrum. Die *Peaks* eines Strahls kennzeichnen die gleiche Strukturkomponente in höherer Auflösung. **(d)** Frequenzspektrum von (b). Auch hier sind deutlich die texturkennzeichnenden *Peaks* zu sehen.

In Guerin-Dugue und Palagi (1994) wird dagegen die Nichtorthogonalitätseigenschaft von Gaborfiltern benutzt, um auch Texturen mit einer Frequenz zu erkennen, auf die die benutzten Gaborfilter nicht parametrisiert waren. In dieser Arbeit wird eine SOFM (*self organization feature map*) (Kohonen 1984) verwendet, um typische Amplitudenkombinationen von solchen „Zwischenfrequenzen“ zu lernen. Diese Idee, daß auch eine systematische Abtastung des Frequenzraumes (wie bei der Bildkomprimierung) sehr gut zur Texturanalyse benutzt werden kann, liegt auch den Arbeiten von Wiskott und von der Malsburg (1995), Greenspan (1996) sowie Campbell et al. (1996) zu Grunde. Allerdings müssen hier — im Gegensatz zur Bildkomprimierung — sich teilweise überlappende Filterbänke benutzt werden. Eine Rechtfertigung dieses Ansatzes ist durch Pötzsch in der Arbeit Pötzsch et al. (1996) gegeben worden, in der aus den Amplituden von 32 Gaborfiltern, welche systematisch den Frequenzraum abtasten, das gelernte Muster als Bild rekonstruiert werden konnte. Es ist dort sehr schön zu sehen, daß mit diesem Vorgehen keine relevante Information verloren geht.

Kapitel 3

Maschinelles Lernen

Dieses Kapitel beginnt mit einem Überblick über die in dieser Arbeit verwendeten Lernverfahren. Schwerpunkt liegt hierbei auf überwachten Lernverfahren — im speziellen auf den in Kapitel 4 und bei der strukturellen Mustererkennung benutzten Regel- und Entscheidungsbaumlernverfahren. Im zweiten Teil wird das Modellselektionsproblem besprochen. Anschließend soll eine, ebenfalls in Kapitel 4 verwendete, Modellselektionsmethode vorgestellt werden.

3.1 Maschinelle Lernverfahren

Definitionen. Beim maschinellen Lernen geht es überwiegend darum, aus Daten eine Menge von Regeln zu gewinnen oder eine gegebene Regelmenge zu verbessern (Morik 1995, S. 251). Die Daten sind dem Lernverfahren im allgemeinen in Form einer endlichen Trainingsmenge $S = \{(x, y)\} \subset \mathcal{X} \times Y$ gegeben, wobei $\mathcal{X}$ hier für die Grundgesamtheit aller Instanzen steht (Instanzenraum). Die Elemente der Menge Y werden Klassen genannt. Die Verteilung der Klassen über dem Instanzenraum wird als Problemverteilung bezeichnet. Die Problemverteilung definiert bei gegebenem Instanzenraum $\mathcal{X}$ ein Konzept $\mathcal{C}$. Ein Beispiel: Über den Instanzenraum aller Autos kann mit Hilfe der Menge $Y = \{0, 1\}$ das Konzept $\mathcal{C}_1$ aller Benzinautos oder aber das Konzept $\mathcal{C}_2$ aller Personenkraftwagen spezifiziert werden. Ein Lernalgorithmus erzeugt dann aus S eine Hypothese[1] $h : \mathcal{X} \to Y$. Ein Hypothese ist eine Funktion, die jede Instanz auf eine Klasse abbildet. Lernen ist deshalb immer induktiv, d.h. man schließt von einigen Beispielen auf die Grundgesamtheit. Ziel im maschinellen Lernen ist es, aus der Trainingsmenge S eine Hypothese h zu konstruieren, die sich möglichst gut an die Problemverteilung anpaßt. Die Menge aller Hypothesen bildet den Hypothesenraum $\mathcal{H} = \{h\}$. Existiert eine Ordnung über den Hypothesen, so wird der dadurch entstehende Graph Refinementgraph genannt.

3.1.1 Überblick

In dieser Arbeit werden die Systeme C4.5 (Quinlan 1986), CAL5 (Unger und Wysotzki 1981; Michie et al. 1994), HERAKLES (Scheffer et al. 1997b) und TRITOP (Geibel und Wysotzki 1997) verwendet und sollen mit Hilfe der Definitionen aus dem vorherigen Abschnitt unterschieden werden. Darüber hinaus soll das mehrfach zitierte Lernsystem SOFM

[1] auch als Klassifikator bezeichnet

(Kohonen 1984) eingeordnet werden. Diese Differenzierung erhebt nicht den Anspruch der Vollständigkeit sondern der Übersicht!

Repräsentation von $\mathcal{X}$. In dieser Arbeit sollen nur zwei mögliche Typen von Instanzenräumen betrachtet werden. Alle Lernverfahren, die über dem Instanzenraum $\mathcal{X} = \mathbb{R}^n$ lernen, werden propositionale Lernverfahren genannt. Tests in den Hypothesen haben dann meist die Form $X_i \leq x_i$ oder im Falle diskreter Attribute $X_i = x_i$. Die Lernsysteme C4.5, CAL5, SOFM und HERAKLES sind in der Lage, in propositionalen Instanzenräumen zu lernen. $\mathcal{X}$ kann aber auch die Menge aller Formeln des Prädikatenkalküls erster Stufe sein. Lernverfahren in diesen Instanzenräumen werden ILP (*inductive logic programming*) Lernverfahren genannt. Man beschränkt sich meist auf Formeln über Hornklauseln. Hornklauseln sind Implikationen der Form $A \rightarrow B$. Für eine detailierte Einführung in die ILP wird auf Lavrač und Dzeroski (1994) verwiesen. Die Systeme HERAKLES und TRITOP sind ILP–Systeme.

Y wird zur Hypotheseninduktion verwendet? Werden die Klassen bei der Hypothesenerzeugung benutzt, so spricht man vom überwachten Lernen. Systeme, die überwacht lernen, sind C4.5, CAL5, HERAKLES und TRITOP. Lernen ohne Benutzung der Klassen bezeichnet man als unüberwachtes Lernen. In die Gruppe der unüberwachten Lernverfahren gehört SOFM. Hier wird die Verteilung, welche die Trainingsbeispiele im Instanzenraum haben, benutzt, um Regeln zu extrahieren.

Repräsentation der Hypothesen (Hypothesensprache). Die Lernsysteme können danach unterschieden werden, wie die Hypothesen repräsentiert werden. Übliche Repräsentationen sind Entscheidungs-, Regelbäume sowie neuronale Netze. Entscheidungsbäume sind Bäume, an deren Knoten Tests stehen und an deren Blättern die Klassen stehen. Entscheidungsbäume werden von C4.5, CAL5 oder TRITOP induziert. Demgegenüber stehen an den Knoten eines Regelbaumes Implikationen der Form $A \rightarrow B$, wobei A eine Konjunktion von Tests und $B \in Y$ ist. Regelbäume werden vom System HERAKLES als Hypothesen verwendet. Neuronale Netze repräsentieren die Hypothese (Wissen) in der Topologie und verteilt in den Netzgewichten (Rojas 1993). Beispielsweise induziert SOFM ein neuronales Netz.

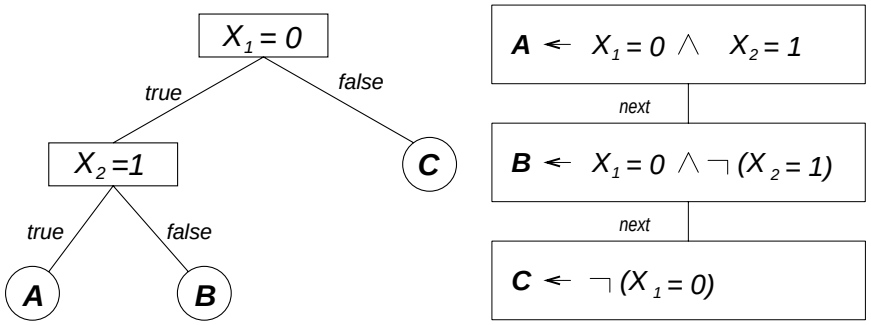
Bewegung im Refinementgraph. Beginnt ein Lernverfahren mit der allgemeinsten Hypothese und spezialisiert dann schrittweise seine Hypothese, wird ein solches Lernverfahren als *Top–Down* Lernverfahren bezeichnet. Im Gegensatz dazu wird bei *Bottom–Up* Lernverfahren mit der speziellsten Hypothese, den Trainingsbeispielen selbst, begonnen. Entscheidungsbaumlernverfahren sind üblicherweise *Top–Down* Lernverfahren (es werden schrittweise Tests eingeführt, was eine Spezialisierung bedeutet). Alle in dieser Arbeit verwendeten Lernsysteme bewegen sich *top–down* durch den Hypothesenraum.

3.1.2 Entscheidungsbaum- vs. Regelbaumlernverfahren

Um exemplarisch einen Entscheidungs- und Regelbaum zu zeigen, sei über dem Instanzenraum $\mathcal{X} = \{0;1\}^2$ und der Klassenmenge $Y = \{A, B, C\}$ folgendes Konzept $\mathcal{C}$ gegeben:

$$(X_1 = 0 \wedge X_2 = 1 \rightarrow A) \wedge (X_1 = 0 \wedge X_2 \neq 1 \rightarrow B) \wedge (X_1 \neq 0 \rightarrow C) \tag{3.1}$$

Entscheidungsbaum Die Konjunktion in der Formel (3.1) läßt sich in 3 Implikationen zerlegen[2]. Diese Implikationen repräsentieren die Pfade eines Entscheidungsbaumes. In Abbildung 3.1 (Teil (a)) ist ein möglicher Entscheidungsbaum für dieses Konzept $\mathcal{C}$ dargestellt. Um nun für ein gegebenes Beispiel die Klasse zu bestimmen, muß knotenweise der Test aus-



(a) (b)

Abbildung 3.1: **(a)** Beispiel eines Entscheidungsbaumes. **(b)** Äquivalente Regelliste.

gewertet werden. Ist der Test erfüllt, wird am linken Ast zum nächsten Test oder zur Klasse verzweigt. Anderenfalls wird der rechte Ast verfolgt.

Regelbaum Ein möglicher Regelbaum entsteht aus der Verkettung aller Implikationen der Formel (3.1). Dieser Regelbaum ist in Abbildung 3.1 (Teil (b)) dargestellt. Allerdings kann

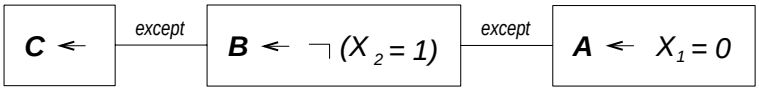


Abbildung 3.2: Zum Entscheidungsbaum in Abbildung 3.1 äquivalente Ausnahmenliste

diese Formel auch über eine Liste von Ausnahmen dargestellt werden. Eine solcher Regelbaum ist in Abbildung 3.2 dargestellt. Ein Beispiel wird nun klassifiziert, indem geprüft wird, ob der Test in einem Regelknoten erfüllt sind. In diesem Fall wird rekursiv geprüft, ob einer der Tests in den Ausnahmen (*except*–Äste) wahr ist. Das Beispiel wird dann von der entsprechenden Ausnahme klassifiziert. Ist keiner der Tests in den Ausnahmeregel wahr, wird das Beispiel durch die Regel klassifiziert. Ist der Test im Regelknoten falsch, wird versucht, das Beispiel durch die Nachfolgeregel (*next*–Äste) zu klassifizieren. Diese Regelbäume werden auch *Ripple Down Rules* (RDR) genannt (Gaines und Compton 1992). Algebraische Grundlagen und Transformationsregeln für RDRs werden in (Scheffer 1996) formuliert.

Wie in Abschnitt 3.1.1 festgestellt wurde, sind alle in dieser Arbeit verwendeten Lernverfahren *Top–Down* Lernverfahren. Im folgenden soll kurz und nur schematisch der Induktionsprozeß

[2]Es sei: $X \neq x \equiv \neg(X = x)$.

Eingabe : $S_{train} = \{(x, y)\}$
Ausgabe : $h = \text{TopDown}(S_{train})$

TopDown (S) *liefert eine Hypothese für S*

[1]	**if** $finish(S)$ **return** $close(S)$	*wenn kein Test notwendig → Rekursionsabbruch*
[2]	$t' = \text{argmax}_{t \in T(S)} Q(t, S)$	*Auswahl des Tests t' mit der höchsten Qualität $Q(t')$ aus allen durch T generierten Tests*
[3]	$S = S_0 \cup S_1 \cup \ldots$	*S_k entsteht durch k–te Testanwendung von t' auf S*
	for $S_k \subset S$ **do**	
[4]	$h_k = \text{TopDown}\ (S_k)$	*Teilhypothesen für alle S_k berechnen*
	end for	
[5]	**return** $\bigwedge_k combine(h_k, t'_k)$	*Konjunktion aller Teilhypothesen bilden*

Tabelle 3.1: Schema einer *Top–Down* Induktion

der vier verwendeten Systeme mit Hilfe der Tabelle 3.1 erklärt werden. Alle Systeme benutzen einen rekursiven Algorithmus zur Induktion der Hypothesen.

Schritt 1: Es wird geprüft, ob es bei gegebener Trainingsmenge S noch sinnvoll ist, einen Test zu lernen. Viele Systeme besitzen für dieses Kriterium einen Parameter, der die minimale Anzahl von Trainingsbeispielen angibt, die notwendig sind, um einen neuen Test zu lernen. Hat S weniger als die angegebene Anzahl von Beispielen, wird die häufigste Klasse in S (C4.5, CAL5, TRITOP) bzw. die leere Hypothese (HERAKLES) zurückgegeben (Funktion *close*). An dieser Stelle bricht der rekursive Algorithmus ab.

Schritt 2: Es wird mit Hilfe einer Funktion T aus den Beispielen S eine Menge von Tests generiert. Tests über nur eine Dimension des Instanzenraumes bzw. über ein Merkmal werden univariate Tests genannt. Die Funktion T liefert bei C4.5 alle aus den Trainingsbeispielen bildbaren Tests der Art $X_i \leq x_i$. Demgegenüber generiert CAL5 einen Test über eine Menge von Intervallen einer Dimension X_i. Tests über mehrere Dimensionen des Instanzenraumes bzw. über mehrere Merkmale nennt man multivariate Tests. So generiert die Funktion T beim System HERAKLES einen multivariaten Test der Art $X \in ([x_{1_l}, x_{1_u}], \ldots, [x_{n_l}, x_{n_u}])$. Beim ILP–System TRITOP erzeugt die Funktion T Tests durch schrittweises Einfügen eines aus der Trainingsmenge bildbaren Literals in den leeren Klauselkörper[3]. In diesem Sinne werden die Tests *top–down* generiert. Dagegen werden bei der ILP–Version von HERAKLES die Tests durch paarweise Generalisierung aus den Trainingsbeispielen gebildet. Die Funktion T bewegt sich in diesem Fall *bottom–up* durch den Hypothesenraum. Jedem generierten Test wird dann mittels der Funktion Q unter Benutzung der aktuellen Trainingsmenge eine Qualität (Funktion Q) zugewiesen. Alle Systeme benutzen hier den *information gain* (IG)[4] (Mingers 1988). Jeder Lernalgorithmus wählt im folgenden den Test t' mit der besten Qualität aus.

[3] In TRITOP sind auch andere, hier nicht benutzte, Testgenerierungsfunktionen implementiert.

[4] Darüber hinaus sind in HERAKLES noch drei andere, hier nicht verwendete Qualitätsfunktionen, verfügbar.

Schritt 3: Dieser beste Test t' wird im weiteren benutzt, um die Menge S der Trainingsbeispiele in disjunkte Teilmengen S_k zu zerlegen. Bei C4.5 und HERAKLES werden genau zwei Mengen erzeugt (ein Trainingsbeispiel kann nur den Gleichheitstest bzw. Enthaltenseintest erfüllen oder nicht). Bei CAL5 wird die Anzahl der Teilmengen S_k durch die Menge der Intervalle über die gewählte Dimension bestimmt. TRITOP zerlegt die Menge der Trainingsbeispiele nach der Anzahl der Vorkommen des Klauselkörpers t' in jedem Beispiel.

Schritt 4: Nun wird rekursiv für jede der so entstandenen Teilmengen S_k eine Hypothese h_k gelernt. Allerdings wurden beim System HERAKLES in Schritt 3 diejenigen Trainingsbeispiele in keiner der Mengen S_k eingeordnet, die schon vom gefundenen Test t' richtig klassifiziert werden.

Schritt 5: Um letztendlich eine Hypothese für die gesamte Trainingsmenge S zu erhalten, muß der gefundene Test t' noch mit den Hypothesen h_k verbunden werden. Dabei wird zur Hypothese h_k der Teil t'_k des Tests t' hinzugefügt, der die Menge S_k erzeugte (Funktion *combine*). Die Konjunktion aller so entstandenen Hypothesen liefert die Hypothese für S. Intuitiv gesprochen wird in diesem Schritt ein Testknoten im Baum erzeugt, an dessen Ausgängen die rekursiv gelernten Teilhypothesen angehängt werden.

Die von TRITOP induzierten Entscheidungsbäume, in deren Knoten Konjunktion von Literalen stehen, werden auch als strukturelle Entscheidungsbäume bezeichnet (Watanabe und Rendell 1991). Strukturelle Entscheidungsbäume und die von der ILP–Version von HERAKLES induzierten RDRs mit Konjunktionen von Literalen in den Klauselkörpern der Regelknoten können die gleiche Menge von Konzepten beschreiben und sind leicht ineinander überführbar.

3.2 Modellselektion

Die Menge aller möglichen Hypothesen bildet den Hypothesenraum $\mathcal{H}$. Der Hypothesenraum kann über Parameter des Lernverfahrens wie minimale Anzahl von Beispielen je Regel- bzw. Entscheidungsbaumknoten usw. vergrößert oder verkleinert werden. Dadurch wird der gesamte Hypothesenraum $\mathcal{H}$ in sich enthaltenden Hypothesenräume $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots \subset \mathcal{H}$ zerlegt. Die Hypothesenräume $\mathcal{H}_i$ werden im folgenden Modelle genannt. Ein Modell ist um so komplexer, je mehr Hypothesen es enthält[5]. Ziel eines Lernverfahrens ist es nun, bei gegebenem Modell $\mathcal{H}_i$ und bei gegebener Trainingsmenge S eine Hypothese h_i zu finden, welche die Problemverteilung über dem Instanzenraum bestmöglich approximiert, d.h. den echten Fehler minimiert. Der echte Fehler ist der Fehler, den die Hypothese h_i über dem gesamten Instanzenraum verursacht. Um dieses Kriterium zu erfüllen, versuchen die hier vorgestellten Lernverfahren eine Hypothese h_i^{min} zu finden, die den Trainingsfehler minimiert. Der Trainingsfehler ist der Fehler, den eine Hypothese auf der Trainingsmenge S verursacht. Betrachtet man den Erwartungswert des echten Fehlers dieser Hypothesen h_i^{min} über alle Trainingsmengen der Größe $\|S\|$ — auch als der echte Fehler des Modells $\mathcal{H}_i$ bezeichnet, ergibt sich typischerweise ein Kurvenverlauf wie in Abbildung 3.3. Der „echte Fehler“ wird dabei in einen *Bias*– und Varianzterm zerlegt. Der *Bias* (dt.: Verzerrung) ist der Fehler, der durch die beste Hypothese h_i^{min} im gegebenem Modell $\mathcal{H}_i$ verursacht wird. Ein *Bias* entsteht aus der Einschränkungen der repräsentierbaren Konzepte durch die Hypothesensprache (absoluter *Bias*)

[5]Diese Definition ist eine starke, in dieser Arbeit aber ausreichende, Vereinfachung der korrekten Aussage: „Ein Modell ist um so komplexer, je größer seine VC–Dimension ist.“. Die Behandlung der VC–Dimension würde allerdings den Rahmen dieser Arbeit sprengen.

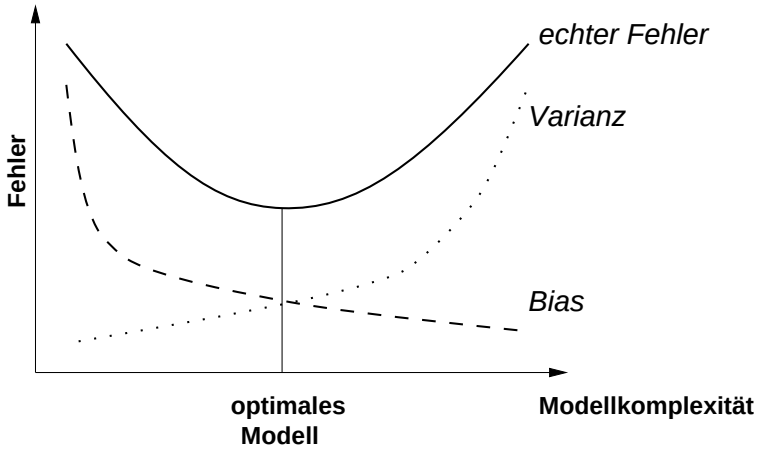


Abbildung 3.3: *Bias*–Varianz Dilemma im maschinellen Lernen

und die Tatsache, daß bei der Induktion bestimmte Hypothesen bevorzugt werden (relativer *Bias*) (Dietterich und Kong 1995). Der Varianzterm dagegen entsteht aus der Tatsache, daß der echte Fehler der den Trainingsfehler minimierenden Hypothese h_i^{min} mehr oder weniger stark von der zufällig gezogenen Trainingsmenge S abhängt. Im Übereinstimmung mit Dietterich und Kong (1995) soll dieser Term auch als Varianz des Lernverfahrens[6] bezeichnet werden.

Die in Abbildung 3.3 dargestellte Situation wird als *Bias*–Varianz Dilemma bezeichnet (Geman et al. 1992): In einfachen Modellen existieren nur wenige Hypothesen und mit hoher Wahrscheinlichkeit haben alle Hypothesen mit kleinem Trainingsfehler auch einen kleinen echten Fehler (kleine Varianz). Die geringe Anzahl an betrachteten Hypothesen erlaubt im allgemeinen aber nur eine schlechten Approximation der Problemverteilung (hoher *Bias*). Mit zunehmender Modellkomplexität steigt die Wahrscheinlichkeit für Hypothesen, die einen kleinen Trainingsfehler aber einen großen echten Fehler haben. Man bezahlt die Möglichkeit der guten Approximation der Problemverteilung (kleiner *Bias*) mit einer starken Abhängigkeit des echten Fehlers der induzierten Hypothese h^{min} von der Trainingsmenge (hohe Varianz). Die zu starke Anpassung der Hypothese an die Trainingsmenge wird auch als *overfitting* bezeichnet. Wie man in Abbildung 3.3 sieht, gibt es ein Minimum des echten Fehlers eines Modells. Dieses Minimum ist dasjenige Modell $\mathcal{H}_i$, in dem der Erwartungswert des echten Fehler einer in diesem Modell induzierte Hypothese h_i^{min} minimal ist.

Schätzung des echten Fehlers. Der echte Fehler eines Modells $\mathcal{H}_i$ kann durch Testen einer induzierten Hypothese h_i^{min} auf einer nicht zum Training benutzten Teilmenge der Trainingsmenge (Testmenge) geschätzt werden. Dieser einmalige Schätzvorgang besitzt natürlich eine große Varianz. Eine Möglichkeit, diese Varianz zu berücksichtigen und eine genauere Schätzung des echten Fehlers zu erhalten ist m–fache *cross validation* durchzuführen. Hier wird die Trainingsmenge in m disjunkte, gleich große, Mengen unterteilt. Das Lernverfahren lernt jeweils auf $m - 1/m$–tel der Trainingsmenge und schätzt den echten Fehler der gefunde-

[6] bei konstantem Modell

nen Hypothesen h^{min} auf dem verbleibendem $1/m$–tel der Trainingsmenge. Der Mittelwert der Fehler ist ein besserer Schätzer für den echten Fehler des Modells (Kohavi 1996). Allerdings besitzt diese Schätzung eine kleine Verzerrung, da die einzelne Trainingsmenge nur noch $m - 1/m$–tel so groß wie die ursprüngliche Trainingsmenge ist. Benutzt man daher m–fache *cross validation* zur Modellselektion, erhält man ein zu einfaches Modell; das beste Modell für eine Trainingsmenge der Größe $\|S\|$ ist im allgemeinen komplexer.

3.2.1 Merkmalsselektion

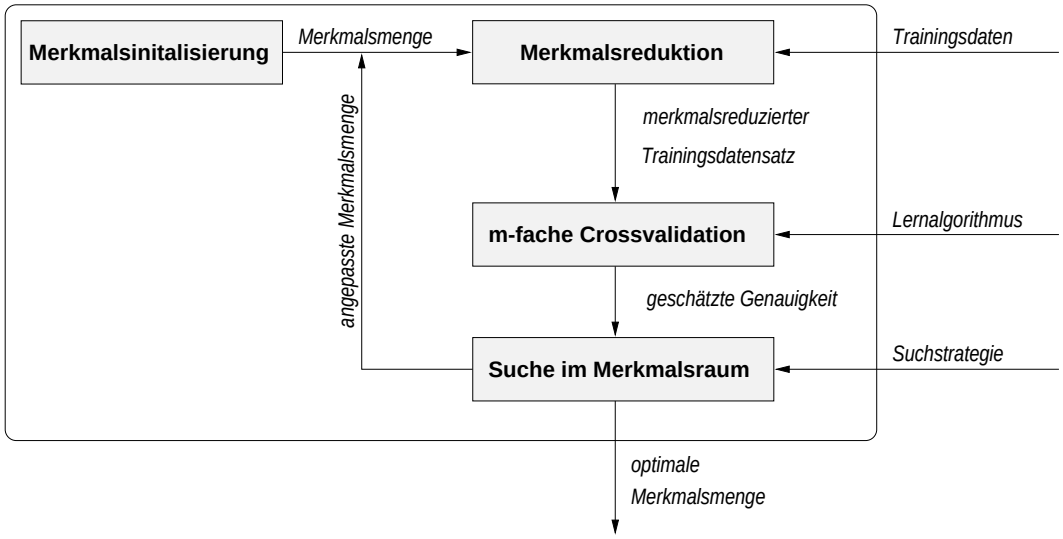


Abbildung 3.4: Schematische Ablauf der Merkmalsselektion

Im Falle propositionaler Lernverfahren werden die Hypothesen durch Tests über den n–dimensionalen Instanzenraum beschrieben (siehe Abschnitt 3.1.1). Jede der n Dimensionen des Instanzenraumes nennt man auch Merkmal (*feature*). Je mehr Merkmale der Instanzenraum besitzt, um so komplexer werden die dadurch entstehenden Modelle. Eine Möglichkeit der Modellselektion ist demnach die Selektion von Merkmalen. Dabei müssen die Merkmale bzw. Modelle in Bezug auf das verwendete Lernverfahren, welches die Hypothese h^{min} induziert, ausgewählt werden.

Eine Methode, welche die Merkmale in Abhängigkeit vom Lernalgorithmus selektiert, ist die von Kohavi (1996) vorgestellte *Wrapper* Methode. Der schematische Ablauf ist in Abbildung 3.4 dargestellt. Beginnend mit einer initialen Merkmalsmenge (initiales Modell) wird im ersten Schritt die Trainingsmenge bezüglich der aktuellen Merkmalsmenge reduziert. Diese merkmalsreduzierte Trainingsmenge wird dann zusammen mit dem zu verwendenden Lernalgorithmus benutzt, um den echten Fehler der Merkmalsmenge zu schätzen (m–fache *cross validation*). Besitzt der Lernalgorithmus Parameter, so müssen diese mit der von Scheffer und Herbrich (1997) vorstellten Methode angepaßt werden. Hier werden mit Hilfe einer zweiten, geschachtelten, k–fachen *cross validation* die Lernparameter für jede der m Trainingsmengen angepaßt. Dadurch ist sichergestellt, daß man trotz Parameteranpassung einen unverzerrten Schätzer des echten Fehler der Merkmalsmenge erhält. Anhand der Suchstrategie und

(1)	v sei der Startzustand
(2)	Expandiere v in die Nachfolger $\{w\}$
(3)	Berechnung der Qualität q_w aller Nachfolger von v
(4)	v' sei der Nachfolger mit der höchsten Qualität $q_{v'}$
(5)	Wenn $q_{v'} > q_v$ dann ist $v = v'$ und gehe zu Schritt 2
(6)	v ist der optimale Knoten

Tabelle 3.2: Schema einer *Hill–climbing* Strategie (aus Kohavi (1996))

des geschätzten echten Fehlers wird entschieden, ob weiter nach besseren Merkmalsmengen (Modellen) gesucht wird oder die bis dahin erhaltene Teilmenge als optimale Merkmalsmenge (optimales Modell) zurückgegeben wird. Dieses Verfahren bewegt sich durch Merkmalsselektion auf der Kurve des echten Fehlers in Abbildung 3.3 mit Hilfe der Suchstrategie aus einer Richtung auf das optimale Modell zu. Allerdings ist dieser Ansatz sehr rechenintensiv.

Der Raum möglicher Merkmalsmengen (im folgenden Merkmalsraum genannt) bei dieser Form der Merkmalsselektion entsteht dadurch, daß jedes Merkmal in einer Merkmalsmenge enthalten sein kann oder nicht. Damit hat der Merkmalsraum im n–dimensionalen Instanzenraum genau 2^n Elemente. Schon für kleine n ist es offensichtlich, daß nicht der gesamte Merkmalsraum bei der Selektion betrachtet werden kann. Zwischen den Elementen des Merkmalsraumes existiert eine partielle Ordnung. Es besteht genau dann eine Beziehung zwischen zwei Merkmalsmengen, wenn der Durchschnitt beider Mengen genau ein Merkmal enthält, d.h. in einer der beiden Merkmalsmengen ist genau ein Merkmal gegenüber der anderen Merkmalsmenge hinzugekommen. Desweiteren ist mit jedem Element des Merkmalsraumes eine Qualität verbunden, die sich durch die Schätzung des echten Fehlers auf der entsprechend merkmalsreduzierten Trainingsmenge ergibt. Das Problem der Merkmalsselektion wird somit zu einem Suchproblem in einem Graphen. In Kohavi (1996) wird die Suchstrategie in Tabelle 3.2 vorgeschlagen. Diese Suchstrategie kann in zwei Richtungen angewendet werden:

Vorwärtsgerichtete Suche. Bei der vorwärtsgerichteten Suche[7] beginnt man mit der leeren Merkmalsmenge in Schritt 1. Der Expansionsoperator im Schritt 2 fügt jeweils genau ein noch nicht vorhandenes Merkmal zur aktuellen Merkmalsmenge v hinzu. Im allgmeinen reduziert sich der Aufwand des Lernens in niedrigdimensionalen Instanzenräumen — mit diesen beginnt das Verfahren. Außerdem terminiert das Verfahren in einer sehr kleinen Merkmalsmenge (Kohavi 1996). Weiterhin wurde festgestellt, daß durch vorwärtsgerichtete Selektion die Varianz des Lernverfahrens sehr klein wird. Nachteil dieses Verfahrens ist die Tatsache, daß korrelierte Merkmale nicht immer gefunden werden. Bei korrelierten Merkmalen steigt die Qualität nicht, wenn nur eines der beiden korrelierten Merkmale benutzt wird. Die Suchstrategie stoppt eventuell zu früh.

Rückwärtsgerichtete Suche. Bei der rückwärtsgerichteten Suche[8] beginnt man mit allen Merkmalen im Startzustand. Hier wird bei einer Expansion jeweils immer genau ein schon vorhandenes Merkmal aus der aktuellen Merkmalsmenge v entfernt. Die rückwärtsgerichtete Suche hat den Nachteil sehr langer Laufzeiten, da dieses Verfahren

[7] Im folgenden mit FFSS (*forward feature subset selection*) abgekürzt.
[8] Im folgenden mit BFSS (*backward feature subset selection*) abgekürzt.

in hochdimensionalen Instanzenräumen startet. Dagegen ist die rückwärtsgerichtete Merkmalsselektion in der Lage, Korrelationen zwischen Merkmalen zu erkennen. Das Entfernen eines korrelierten Merkmals würde die Qualität der Merkmalsmenge deutlich senken.

Kapitel 4

Ein überwachter Segmentierungsalgorithmus

In diesem Kapitel wird ein überwachter Segmentierungsalgorithmus präsentiert, der sich mit Hilfe maschineller Lernverfahren bestmöglich an eine gegebene Problemverteilung anpaßt. Die Problemverteilung ist dem Verfahren über die angebotenen Beispielsegmentierungen gegeben. Ein Segmentierungsalgorithmus wird überwacht genannt, wenn er — ähnlich wie ein überwachtes Lernverfahren — Beispielsegmentierungen benötigt. Beim hier vorgestellten Algorithmus wird mit Hilfe der Beispiele das beste Modell selektiert, in dem dann die eine Hypothese h^{min} — die Segmentierungsprozedur — induziert wird. Der im folgenden vorgestellte Ansatz kombiniert die in Kapitel 2 vorgestellten Gaborfilter mit Regelbauminduktion und Merkmalsselektion aus Kapitel 3 zu einem neuen Segmentierungsalgorithmus.

4.1 Modell des Segmentierungsalgorithmus

In Abbildung 4.1 ist das allgemeine Modell des Segmentierungsalgorithmuses dargestellt. Als

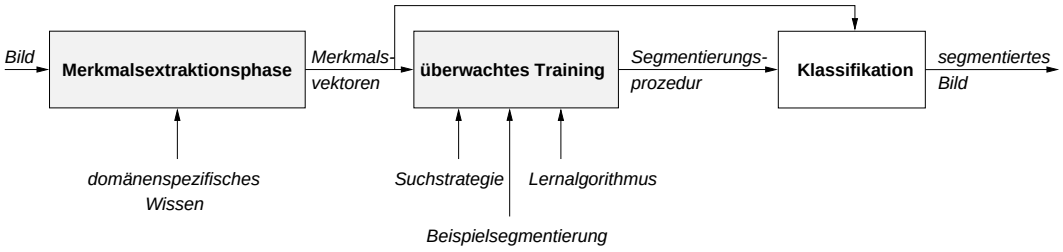


Abbildung 4.1: Schema des Segmentierungsalgorithmus

Eingangsdaten erhält der Algorithmus digitale Bilder. Aus jedem Bild wird — unter eventueller Zuhilfenahme von domänenspezifischem Wissen — für jedes Pixel ein n–dimensionaler Merkmalsvektor x extrahiert. Die aus der Beispielsegmentierung erhaltenen Merkmalsvektoren $\{x\}$ werden anschließend mit der Regionennummer, die durch diese Segmentierung gegeben ist, zur Trainingsmenge $S = \{x, y\}$ verbunden. In einem überwachten Trainingsschritt wird diese Menge S zum Lernen der besten Hypothese h^{min} benutzt. Es werden

Regelbäume zur Repräsentation der Hypothesen verwendet. Jede Regel der Hypothese kann als *if–then–else* Anweisung interpretiert werden. Der Regelkörper wird dabei zur Bedingung; die Regionennummer ist der Rückgabewert der *then* bzw. *else* Anweisung. Die Hypothese stellt eine Verkettung von *if–then–else* Anweisungen dar. Aus diesem Grunde wird die Hypothese hier als Segmentierungsprozedur bezeichnet. Anschließend wird die induzierte Segmentierungsprozedur benutzt, um nicht zum Training benutzte Bilder zu segmentieren. Die beiden wichtigsten Funktionen dieses allgemeinen Schemas, die Merkmalsextraktion und das überwachte Training, sollen im folgenden aufgeschlüsselt werden.

In Abbildung 4.2 ist der Funktionsblock der Merkmalsextraktion detailiert aufgelöst. Im er-

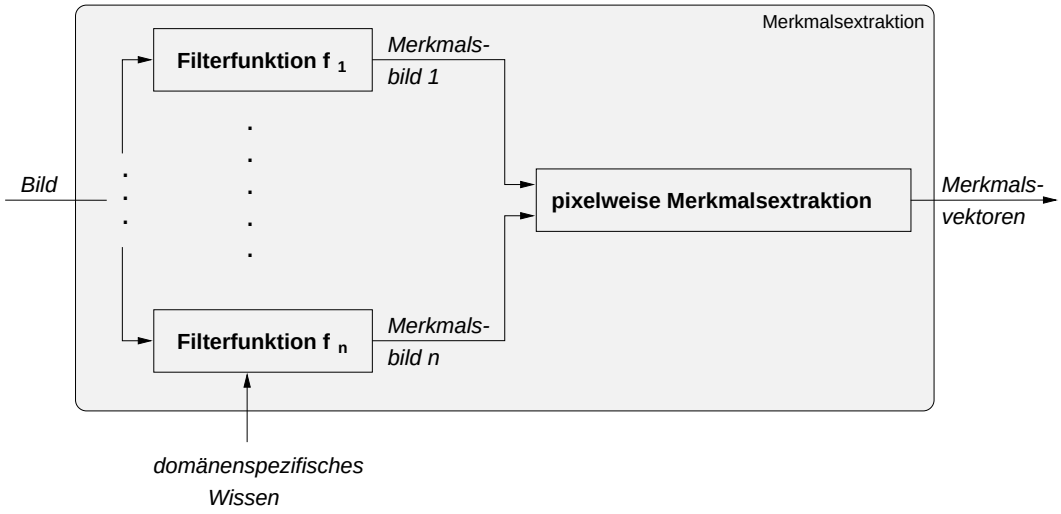


Abbildung 4.2: Merkmalsextraktion aus den Bilddaten

sten Schritt wird die Information des ursprünglichen Bildes mittels Faltung mit einer Filterbank vervielfacht. Als Filter werden die in Abschnitt 2.4 vorgestellten Gaborfilter verwendet. Diese Filter sind aufgrund ihrer Lokalität in Orts- und Frequenzraum sehr gut zur Texturseg mentierung geeignet. Der Frequenzraum wird systematisch abgetastet, da dabei keine Information über das ursprüngliche Bild verloren geht (siehe Abschnitt 2.4.4). Desweiteren wird die Ortsraumrepräsentation des Bildes (Grauwert) als ein Merkmalsbild benutzt. Damit erhält man für jedes Pixel des Bildes einen n–dimensionalen Merkmalsvektor. Die ersten $n-1$ Merkmale sind hierbei die Amplitudenwerte, die sich durch Faltung des Bildes mit den einzelnen Gaborfiltern ergeben. Sind bei einem Datensatz Ergebnisbilder der Anwendung domänenspezifischer Filter gegeben, werden diese ebenfalls pixelweise als Merkmal benutzt. Dies ist z.B. beim Röntgenbilderdatensatz in Kapitel 5 der Fall.

In Abbildung 4.3 ist der Funktionsblock des überwachten Trainings dargestellt. Die durch Merkmalsextraktion erhaltenen Merkmalsvektoren der Beispielsegmentierung in Verbindung mit den Regionennummern definieren eindeutig daß zu lernende Konzept $\mathcal{C}$. Allerdings hat man bei Bilddaten mit nur wenigen Beispielbildern schon sehr viele Trainingsdaten (>100000 Beispiele). Alle hier vorgestellten propositionalen Lernsysteme (der Instanzenraum $\mathcal{X}$ ist in diesem Fall der $\mathbb{R}^n$) sind weder vom Aufwand zur Induktion noch in Bezug auf die Qualität der induzierten Hypothese in der Lage, mit so großen Trainingsmengen umzugehen. Aus

diesem Grund wird durch zufälliges Ziehen aus der großen Trainingsmenge eine größenmäßig reduzierte Menge von Merkmalsvektoren gebildet. Dieser Prozeß wird *subsampling* genannt. Es wird mit Zurücklegen gezogen, um mit hoher Wahrscheinlichkeit in der kleineren Trainingsmenge die gleiche Klassenverteilung der Instanzen wie in der großen Trainingsmenge zu haben[1]. Die so erhaltene Trainingsmenge wird nun, zusammen mit einer gewählten Suchstrategie und einem Lernalgorithmus, benutzt, um die optimale Merkmalsmenge mit der *Wrapper* Methode aus Abschnitt 3.2.1 zu suchen. Anschließend wird die gefundene Merkmalsmenge zur Induktion der Segmentierungsprozedur benutzt. Die so gelernte Prozedur bildet extrahierte Merkmalsvektoren auf Regionennummern ab.

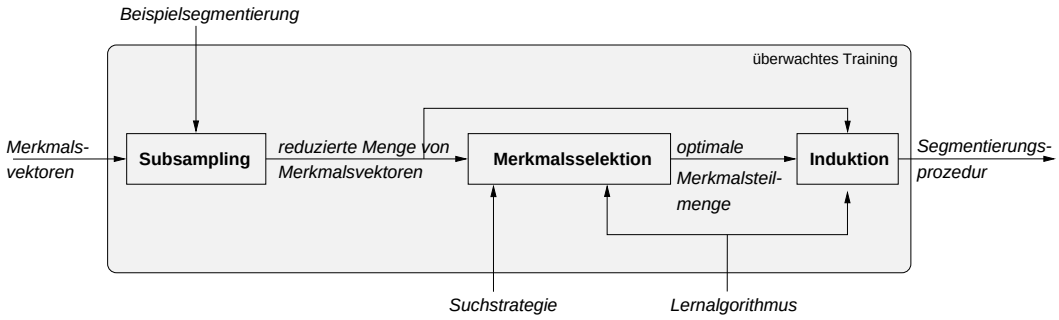


Abbildung 4.3: Überwachtes Training mit Merkmalsselektion

Als Lernverfahren wird das in Abschnitt 3.1.2 beschriebene Regelbaumlernsystem HERAKLES benutzt. Dies hat folgende Gründe:

1. Im Gegensatz zu dem in (Campbell et al. 1996) benutzten Lernverfahren SOFM ist die Regelbauminduktion ein überwachtes Lernverfahren. Damit paßt sich das Verfahren der Problemverteilung und nicht der Verteilung der Instanzen im Instanzenraum an.

2. Regelbäume sind die für Experten am verständlichsten und am leichtesten modifizierbaren Wissenstrukturen. Sie könnten gegenüber neuronalen Netzen sehr leicht in Programme transformiert werden.

3. Das verwendete System HERAKLES lernt im Gegensatz zu C4.5 und CAL5 multivariate Tests und kann somit Korrelationen zwischen einzelnen Gaborfiltern bei der Testgenerierung berücksichtigen. Dies ist notwendig, da der Frequenzraum bei der hier vorgestellten Merkmalsextraktion systematisch abgetastet wird (Guerin-Dugue und Palagi 1994).

4.2 Eigenschaften des Segmentierungsalgorithmus

Die induzierte Segmentierungsprozedur wird jeweils pixelweise angewendet. Aus diesem Grund bezeichnet man die gefundene Hypothese auch als Pixelklassifikator. Die Unterschiede

[1] Man hat im Mittel weniger unterschiedliche Beispiele gegenüber dem Ziehen ohne Zurücklegen aber die Klassenverteilung hat eine kleinere Varianz gegenüber dem Ziehen ohne Zurücklegen.

des hier vorgestellten Verfahren zu Michie et al. (1994), wo ein Pixelklassifikator für Satellitenbilder die Segmentierung in Lehmboden, Getreidefeld und Gemüsefelder anhand von Mehrkanalbildern lernen sollte, sind in der Merkmalsexpansion durch Gaborfiltern und Merkmalsselektion als Lernmechanismus zu sehen.

Pixelklassifikatoren auf Grundlage des Grauwertes haben üblicherweise das Problem, nicht–homogene Muster (Texturen) zu erkennen. Texturen sind dadurch gekennzeichnet, daß sie im Ortsraum sehr stark variieren (regelmäßig oder zufällig). Selbst wenn ein Pixelklassifikator die Nachbarschaft eines Pixels als zusätzliche Merkmale erhält — wie im Falle des Satellitenbilderdatensatzes in (Michie et al. 1994), ist er im allgemeinen nicht in der Lage, ein Bild mit Texturen zufriedenstellend zu segmentieren. Dies liegt daran, daß das Segmentierungsprädikat (Definition 2.3) unter Verwendung des Grauwertes nicht separierbar ist. Benutzt man dagegen Faltungsbilder von Gaborfiltern als zusätzliche Merkmale, so wird das Prädikat unter Umständen wieder separierbar. In diese Fall kann man das Segmentierungsproblem in linearer Zeit lösen! Im besten Fall kann das Amplitudenbild mit einem Schwellwertalgorithmus (Definition 2.4) segmentiert werden. Diese Annahme liegt den Arbeiten von Bovik et al. (1990), Weldon et al. (1996), Dunn et al. (1994) und Tan (1995) zu Grunde. Dadurch sind diese Verfahren allerdings sehr abhängig von der Wahl des Gaborfilters. Jede dieser Arbeiten präsentiert eine andere Methode, um ein sogenanntes optimales Gaborfilter zu finden. Mit dieser Methoden schränkt jeder dieser Ansätze die Klasse der segmentierbaren Bilder *a priori* ein. Das hier vorgestellte Verfahren kommt dagegen ohne eine *a priori* Einschränkung aus.

Außerdem wird durch die Verwendung von Merkmalsselektion als Lernmechanismus erreicht, daß je nach Problemverteilung in den Beispielsegmentierungen, diejenige Merkmalsteilmenge ausgewählt wird, die im Mittel zum kleinsten Fehler führt. Damit paßt sich dieses Verfahren bestmöglich an das gegebene Problem an. Ohne Merkmalsselektion würde eine Erhöhung der Merkmale durch Anwendung von Faltungen mit Gaborfiltern wegen des *Bias*-Varianz Dilemmas zu keiner Verbesserung führen (siehe Abbildung 3.3) — eventuell sogar zu einer Verschlechterung! Das Lernverfahren wäre zwar in der Lage, den Fehler auf den Trainingsdaten zu minimieren. Dies würde aber mit hoher Wahrscheinlichkeit zu einer Überanpassung (*overfitting*) führen. Mit Hilfe der *Wrapper* Methode kann dasjenige Modell selektiert werden, in dem sich der Effekt der Überanpassung minimiert, so daß der echte Fehler minimiert wird. Der Algorithmus benötigt dafür keine datensatzabhängigen Parameter. Damit ist dieser Segmentierungsalgorithmus parameterlos.

Zusätzlich wird durch die Benutzung von vorwärtsgerichteter Selektion die Varianz des Lernverfahrens minimiert (siehe Abschnitt 3.2.1). Dies führt dazu, daß die Qualität der induzierten Hypothese weniger von der ausgewählten Trainingsmenge und damit vom *subsampling* abhängt. Damit ist dieser Ansatz auch stabiler in der praktischen Anwendung. Allerdings ist es sehr wahrscheinlich, daß es bei grober Abtastung des Frequenzraumes eine starke Korrelation zwischen den Amplitudenbildern gibt. Diese können aber nur bei der rückwärtsgerichteten Selektion berücksichtigt werden.

Zusammenfassend läßt sich der Algorithmus folgerndermaßen beschreiben: Die eindimensionale Information eines Pixels im Eingabbild (Grauwert) wird durch lineare Transformationen (z.B. Faltung mit Gaborfiltern) in einen hochdimensionalen Raum expandiert. Die Hoffnung dabei ist, daß die Problemverteilung im hochdimensionalen Raum optimal approximiert werden kann. Durch die Benutzung von Merkmalsselektion wird beim anschließenden Lernen eine Überanpassung vermieden. Dazu wird die Information wieder derart in einen niedrigdimensionalen Raum projiziert, daß der Erwartungswert des echten Fehlers minimiert wird.

Kapitel 5

Experimente und Ergebnisse

In diesem Kapitel soll die Qualität des in Kapitel 4 vorgestellten Segmentierungsalgorithmus an zwei Datensätzen — Brodatztexturen und Röntgenbildern — untersucht werden. Die Brodatztexturen wurden als ein Standartdatensatz der Textursegmentierung ausgewählt. Es handelt es sich hierbei um eine nicht–praxisrelevante Segmentierungsaufgabe, auf der im allgemeinen sehr gute Ergebnisse mit Gaborfiltern erreicht werden. Am Röntgenbilderdatensatz soll die praktische Anwendbarkeit des Algorithmus getestet werden.

Mit dem dritten Datensatz (Zellbilderdatensatz) wird die Möglichkeit untersucht, unter Verwendung kontextueller Information — in Form räumlicher Relationen — und mit Hilfe bestehender Lernsysteme das Konzept „Gefäßwandzelle“ zu lernen. Leider war es nicht möglich, das in dieser Arbeit vorgestellte Verfahren der Segmentierung zu benutzen. Die Segmentierungen wurden von Dr. Hufnagl an der Charité Berlin mit dem System AMBA vorgenommen.

Folgende Fragestellungen werden im folgenden untersucht:

1. Ist der präsentierte Segmentierungsalgorithmus in der Lage eine Textursegmentierungsprozedur für die Texturen aus Brodatz (1966) zu lernen? Welche Unterschiede ergeben sich bei vorwärts-, rückwärtsgerichteter und keiner Merkmalsselektion?

2. Ist der präsentierte Segmentierungsalgorithmus in der Lage eine Segmentierungsprozedur für die Röntgenbilder zu lernen? Welche Merkmale spielen dabei eine besonders große Rolle? Welche Unterschiede ergeben sich bei vorwärts-, rückwärtsgerichteter und keiner Merkmalsselektion?

3. Ist ein struktureller Klassifikator geeignet, um — bei gegebener optimaler Segmentierung — das Konzept „Gefäßwandzelle“ auf Zellbildern zu lernen? Eignen sich bestimmte Lernverfahren besonders gut, um dieses Konzept zu lernen?

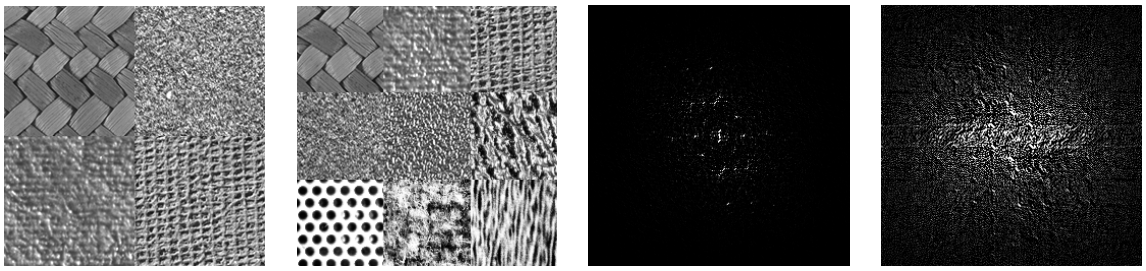
5.1 Experimentieranordnung

In den folgenden Experimenten soll jeweils die Genauigkeit der vorgestellten Verfahren getestet werden. Ist der Datensatz groß genug, liefert ein einmaliger Test auf einer nicht zum Training benutzten Testmenge (mehr als 1000 Datensätze) einen sehr genauen Schätzer für die echte Genauigkeit. Hat man allerdings nur wenige Daten — wie zum Beispiel beim Zellbilderdatensatz mit 398 Datensätzen — zur Verfügung, wird *cross validation* (siehe Abschnitt 3.2) zur Schätzung der Genauigkeit verwendet.

Zum Vergleich der Ergebnisse bei *cross validation* wird der t–Test benutzt. Beim einmaligen Trainieren und Testen wird der in Dietterich (1997) verwendete Test auf Unterschied zweier Verteilungen benutzt[1]. Es wird immer nur der p–Wert (das sogenannte beobachtbare Signifikanzniveau) angegeben. Zu beachten bleibt, daß die Varianz des wirklichen Fehlers bei *cross validation* überschätzt wird — dieser Effekt verstärkt sich mit steigender Anzahl von Durchläufen — und es dadurch zu überschätzten p–Werten beim t–Test kommt (Dietterich 1997). Deshalb wird anstelle von 10–facher *cross validation* 5x2cv durchgeführt. Hier wird mit 5 verschiedenen Reihenfolgen eine 2–fach *cross validation* durchgeführt. Man erhält ebenfalls einen Schätzer über 10 Durchläufe, der aber eine kleinere Varianz als der Schätzer bei 10–facher *cross validation* hat (Dietterich 1997). Bei allen Verfahren werden nur Parameteranpassungen durchgeführt, wenn dies zu keiner optimistischen Verzerrung der wirklichen Genauigkeiten führen kann (Scheffer und Herbrich 1997). Es wird jeweils ein einseitiger Test durchgeführt. Dies führt zu stärkeren Aussagen als der zweiseitige Test, der nur die Ungleichheit zweier Algorithmen berücksichtigt, nicht aber deren Vorzeichen.

5.2 Brodatztexturen

Bei diesem Datensatz werden mehrere Fotos aus Brodatz (1966) zu einem „Karomuster“ verbunden. Aufgabe ist es dann, dieses zusammengesetzte Bild wieder in das Karomuster zu segmentieren. Diese Segmentierungsaufgabe ist eine Standartaufgabe im Kontext der Textursegmentierung, da es sich bei den Bildern in (Brodatz 1966) um realistische Texturen handelt. In dieser Arbeit wurden die Bilder in Abbildung 5.1 ausgewählt.



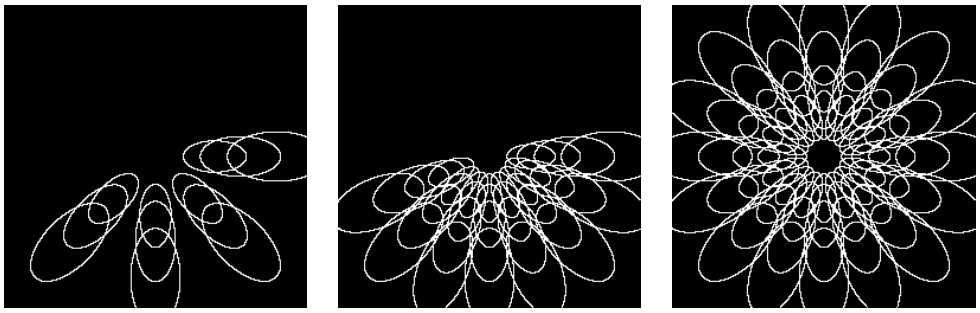
(a) (b) (c) (d)

Abbildung 5.1: **(a)** Karomuster aus 4 Texturen aus Brodatz (1966) **(b)** Karomuster aus 9 Texturen aus Brodatz (1966) **(c)** Frequenzspektrum von (a) **(d)** Frequenzspektrum von (b). Man erkennt sehr deutlich, daß sich das Frequenzspektrum in (c) durch viele sehr lokale *Peaks* auszeichnet, während das Frequenzspektrum in (d) eine sehr breite Streuung mit einigen wenigen *Peaks* aufweist.

Zur Segmentierung wurden jeweils 12, 32 bzw. 64 Gaborfilter mit dem Bild gefaltet und die Amplitudenbilder errechnet. Die Filterbänke wurden mit den *gabor wavelets* aus Lee (1996) berechnet. Dabei betrug die Bandbreite eines Filter immer 1 Oktave (siehe Anhang B). In Abbildung 5.2 ist für jede Filterbank deren Abdeckung im Frequenzraum dargestellt. Durch

[1] Alternativ könnte auch der McNemar Test verwendet werden. Bei Experimenten mit realen Daten zeigte sich aber, daß der vorgeschlagene Test ausreichend genau ist.

jede Ellipse ist die Fouriertransformation des entsprechenden Gaborfilters (Gaußglocke) zu sehen. Nach der Amplitudenberechnung wurden 1000 Pixel zufällig mit Zurücklegen gezogen und zur Regelbauminduktion benutzt. Für alle drei Filterbänke wurde jeweils eine vorwärts-, eine rückwärtsgerichtete und keine Merkmalsselektion durchgeführt. Dazu wurde die Genauigkeit bei der Merkmalsselektion mit 5x2cv (siehe Abschnitt 5.1) bestimmt. Nach erfolgter Selektion wurde dann ein Regelbaum mit HERAKLES gelernt. Anschließend wurde dieser Regelbaum zur Segmentierung des gesamten 256x256 Bildes benutzt (65536 Pixel). Es bleibt zu beachten, daß hier auch die 1000 zum Training benutzten Pixel klassifiziert wurden. Der Erwartungswert von unterschiedlichen Pixeln beim Ziehen mit Zurücklegen von 1000 Pixeln aus 65536 Pixeln beträgt[2] 992. Das bedeutet, daß durchschnittlich 1.51% des Bildes zum Training benutzt wurden. Da HERAKLES immer einen zu 95% korrekten Klassifikator lernte, müßte von der berechneten Genauigkeit exakt 1.43% subtrahiert werden. Diese Verzerrung ist aber konstant bei jeder Schätzung enthalten und kann aus diesem Grunde ignoriert werden.



(a) (b) (c)

Abbildung 5.2: Abtastung des Frequenzraumes mit **(a)** 12 **(b)** 32 und **(c)** 64 Filtern.

Die Segmentierungsergebnisse für das 4–Texturen Bild sind in Tabelle **??** dargestellt. In Tabelle **??** sind die Ergebnisse für das 9–Texturen Bild zu sehen. Desweiteren sind die gemessenen Genauigkeiten sowie die Anzahl ausgesuchter Merkmale in den Tabellen 5.1 und 5.2 protokolliert. Die Laufzeit variierte zwischen einigen Minuten bis zu mehreren Stunden. Die Segmentierungsergebnisse wurden nicht mit morphologischen Operatoren (Joo 1991) nachbearbeitet. Damit sind die Ergebnisse nicht direkt mit denen von Weldon und Higgins (1997) vergleichbar, der nach der Segmentierung einige speziell angepaßte morphologische Filter anwendet. Außerdem sind die Ergebnisse nicht direkt mit denen von Greenspan (1996) vergleichbar, wo eine große Trainingsmenge (300 Pixel je Klasse) aber eine sehr kleine Testmenge (4 bis 100 Beispiele je Klasse) verwendet wird. In (Scheffer und Herbrich 1997) ist gezeigt, daß so kleine Testmengen in Kombination mit einem mehrschichtiges neuronales Netz zu einer sehr großen Verzerrung der Genauigkeitsschätzung führen — gerade wenn, wie hier geschehen, Parameteranpassung im Hinblick auf Testfehlerminimierung durchgeführt wurde.

Am besten sind die hier vorgestellten Ergebnisse mit dem Ansatz in Campbell et al. (1996) und Campbell und Thomas (1997) vergleichbar, wo weder Parameteranpassung noch Testmengensubsampling oder morphologische Nachbearbeitungen durchgeführt wurden.

[2]Die Wahrscheinlichkeit, daß beim Ziehen von K aus N Elementen (mit Zurücklegen) irgendeines der N Elemente gezogen wird ist $1-(1-1/N)^K$. Daher beträgt der Erwartungswert $N(1-(1-1/N)^K)$.

Merkmalsselektion	Anzahl Gaborfilter					
	4–Texturen Bild			9–Texturen Bild		
	12	32	64	12	32	64
keine FSS	0.7707	0.8454	0.8681	0.5013	0.6028	0.5722
FFSS	0.8077	0.8906	0.8453	0.4841	0.5736	0.6486
BFSS	0.8085	0.8477	0.8183	0.4923	0.5887	0.5696

Tabelle 5.1: Genauigkeit der Segmentierung auf den Brodatztexturen

Merkmalsselektion	Anzahl Gaborfilter							
	4–Texturen Bild				9–Texturen Bild			
	12	32	64	durchschn.	12	32	64	durchschn.
keine FSS	13	33	65	37	13	33	65	37
FFSS	7	8	**5**	**6.6**	8	10	**14**	**10.6**
BFSS	13	32	64	36	12	32	64	36

Tabelle 5.2: Anzahl ausgewählter Merkmale bei Merkmalsselektion. Zusätzlich zu den Amplitudenbildern wurde der Grauwert als ein Merkmal benutzt.

Auswertung der Ergebnisse bei Textursegmentierung

Vergleicht man die erhaltenen Genauigkeiten mit dem in Dietterich (1997) vorgeschlagenen Test, so stellt man man fest, daß die p–Werte aller Tests fast 1 sind, d.h. es existieren in jedem Fall signifikante Unterschiede. Das hat zwei Ursachen:

- Der Test selbst ist sehr pessimistisch, d.h. ein Test auf einem Niveau von 95% liefert das Ergebnis eines Tests auf rund 98% (Dietterich 1997). Damit sind die berechneten p–Werte stark überschätzt.
- Die Anzahl der Testbeispiele ist enorm hoch (65536). Damit wird die Schätzung der Genauigkeit sehr gut, d.h. sie besitzt eine geringe Varianz.

Allerdings wurde eine Annahme dieses Tests verletzt. Es wurde zwar die gleiche Testmenge verwendet, aber nicht die gleiche Trainingsmenge (die Trainingsbeispiele wurden zufällig gezogen!). Stattdessen wurde nur eine gleich große Trainingsmenge benutzt! Damit bleibt bei dieser Schätzung die Varianz des Lernverfahrens selbst unberücksichtigt. In (Dietterich und Kong 1995) ist gezeigt, daß die Varianz eines Entscheidungsbaumlernverfahrens — und wegen des analogen Induktionschemas (siehe Tabelle 3.1) auch Regelbaumlernverfahrens — hoch ist und auch mit steigender Größe der Trainingsmenge nicht abnimmt! Daher sollte der vorgeschlagene Test in diesem Fall nicht angewendet werden.

Eine Methode, bei welcher der Varianz des Lernverfahrens Rechnung getragen wird, ist *cross validation* (siehe Abschnitt 3.2). Aus diesem Grund werden für einen Vergleich der Unterschiede bei Merkmalsselektion die Schätzungen genutzt, die sich während der vorwärts- und rückwärtsgerichteten Merkmalsselektion ergeben. In Tabelle 5.3 sind diese Schätzungen, die durch

Merkmalsselektion	Anzahl Gaborfilter			p–Wert
	4–Texturen Bild			
	12	32	64	
keine FSS	0.8284 ± 0.0071	0.8466 ± 0.0030	0.8620± 0.0114	1
FFSS	0.8260 ± 0.0125	**0.8740** ± 0.0110	0.8612 ± 0.0118	1
BFSS	0.8284 ± 0.0071	0.8596 ± 0.0157	0.8670 ± 0.0064	1
	9–Texturen Bild			
	12	32	64	
keine FSS	0.5060 ± 0.0135	0.6078 ± 0.0100	0.6096 ± 0.0308	1
FFSS	0.5330 ± 0.0141	0.6352 ± 0.0130	**0.6428** ± 0.0093	1
BFSS	0.5264 ± 0.0080	0.6342 ± 0.0266	0.6332 ± 0.0144	1

Tabelle 5.3: Geschätzte Genauigkeit der Segmentierung auf den Brodatztexturen. In der letzten Spalte ist der p–Wert für den Test auf Verbesserung durch Hinzunahmen von mehr Gaborfiltern angegeben.

5x2cv entstanden sind, angegeben. Hinter der Genauigkeit ist jeweils die Standartabweichung der Genauigkeit angegeben. Damit ergeben sich die p–Werte in Tabelle 5.4 beim Vergleich der Genauigkeit mit den verschiedenen Merkmalsselektionsmechanismen. Ausgewählt wurden hier die Ergebnisse bei Benutzung von 32 Gaborfilter für den Fall von 4 Texturen bzw. bei Benutzung von 64 Gaborfilter für den Fall von 9 Texturen, da hier die p–Werte maximal sind.

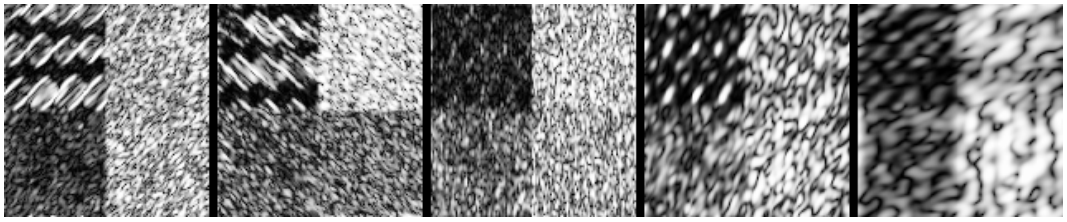
4–Texturen Bild				9–Texturen Bild			
	keine	FFSS	BFSS		keine	FFSS	BFSS
keine	—	0.0001	0.0150	keine	—	0.0043	0.0243
FFSS	**0.9999**	—	0.9848	FFSS	**0.9957**	—	0.9515
BFSS	0.9850	0.0152	—	BFSS	0.9757	0.0484	—

Tabelle 5.4: p–Werte bei Test auf höhere Genauigkeit zwischen zwei Selektionsmechanismen

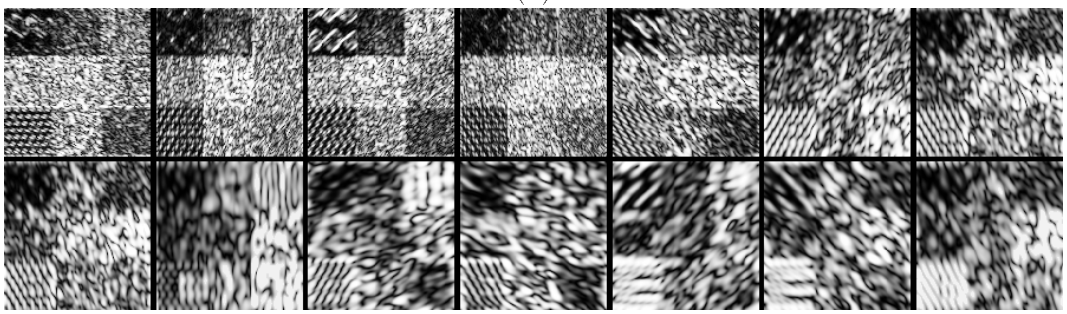
Folgende Aussagen können gemacht werden:

- Durch die vorwärtsgerichtete Merkmalsselektion verschlechtert sich die Qualität der Segmentierung nicht; unter Umständen verbessert sich die Qualität der Segmentierung auf einem Signifikanzniveau von über 99% (siehe Tabelle 5.4). Man erkennt die Verbesserung ebenfalls in den Tabellen **??** und **??**. Während beim 4–Texturen Bild alle drei Selektionsverfahren annähernd gleich gute Ergebnisse mit 32 und 64 Gaborfiltern erzielen, wird es erst durch die Benutzung von vorwärtsgerichteter Merkmalsselektion möglich, daß 9–Texturen Bild zu segmentieren. Dies wird durch die starke Dimensionsreduktion des Instanzenraumes bei vorwärtsgerichteter Selektion möglich. So werden aus durchschnittlich 37 Merkmalen im 4–Texturen Bild nur 6.6 Merkmale und im 9–Texturen Bild nur 10.6 Merkmale ausgewählt. Im oberen Teil von Abbildung 5.3 sind die 5 durch FFSS ausgewählten Merkmale bei der Segmentierung des 4–Texturen Bil-

des mit 64 Gaborfiltern dargestellt. Man sieht sehr deutlich, daß mittels der ersten drei



(a)



(b)

Abbildung 5.3: Bei FFSS ausgewählte Merkmalsbilder. **(a)** 4–Texturen **(b)** 9–Texturen

Merkmalsbilder schon sehr viel Information für die Segmentierung geliefert wird. Im unteren Teil dieser Abbildung sind die ausgewählten Merkmalsbilder für das 9–Texturen Problem dargestellt (14 Merkmale). Hier fällt auf, daß die Textur in der untersten Zeile (Mitte) durch kein Gaborfilter eindeutig von allen anderen Texturen unterschieden werden kann. Betrachtet man das Frequenzspektrum dieser Textur, so fällt eine breite Streuung ohne lokale *Peaks* auf (Abbildung 5.1, Teil (d) ist durch das Frequenzspektrum dieser einen Textur so breitgestreut!). Allerdings sind in den ausgewählten Merkmalsbildern auch noch irrelevante Merkmalsbilder enthalten. Die letzten beiden Merkmale im 4–Texturen Bild führten nur noch zu einer geschätzten Verbesserung von 4%.

- Die rückwärtsgerichtete Merkmalsselektion selektiert fast keine Merkmale aus. In beiden Bildern wurde durchschnittlich nur eine Merkmal verworfen. Es existieren viele Scheinkorrelationen zwischen den Amplitudenbildern der einzelnen Gaborfilteranwendungen. In einem Fall wurden sogar alle Merkmale benutzt (4–Texturen Bild und 12 Gaborfilter). Hier ist sehr schön zu sehen, daß Regelbaumlernverfahren eine große Varianz besitzen. Im Falle von 4 Texturen und 12 Gaborfiltern sind in Tabelle **??** zwei völlig verschiedene Segmentierungen ohne und mit rückwärtsgerichteter Merkmalsselektion dargestellt. In beiden Fällen wurde aber die gleiche Merkmalsmenge und eine gleich große Trainingsmenge benutzt. Während ohne Merkmalsselektion die Segmentierung schlecht ist, ist durch rückwärtsgerichtete Selektion ein Regelbaum gelernt worden, der die beiden linken Texturen gut trennen kann. Dieses Beispiel untermauert nochmals die Benutzung von durch *cross validation* geschätzten Genauigkeiten zum Vergleich der Verfahren!

- Tastest man den Frequenzraum zu grob ab (12 Gaborfilter), reicht die so erhaltene Information nicht aus, um eine gute Segmentierungsprozedur erlernen zu können. Durch Verwendung von 32 bzw. 64 gegenüber 12 Gaborfiltern wurde mit einem p–Wert von 1 eine wirkliche Verbesserung in der Segmentierungsqualität erreicht (Tabelle 5.3). Allerdings führt die immer feinere Abtastung des Frequenzraumes nicht zwangsläufig zu einer Verbesserung. So wurde im 9–Texturen Bild die Anzahl der Gaborfilter von 32 auf 64 Gaborfilter verdoppelt, führte aber ohne Benutzung von Merkmalsselektion zu keiner sichtbaren und signifikanten Verbesserung (p–Wert: 0.5680 bzw. 0.4591). Erst durch die Benutzung von vorwärtsgerichteter Merkmalsselektion konnte das Segmentierungsproblem besser gelöst werden (p–Wert: 0.9239).

- Die erhaltenen Ergebnisse sind vergleichbar mit den Resultaten von Campbell und Thomas (1997). Hier werden die Gaborfilterparameter bei gegebener Anzahl von Gaborfilter durch einen genetischen Algorithmus optimiert. Der Algorithmus muß aus einer fast unendlich großen Mengen die besten Gaborfilter auswählen. Es findet keine Modellselektion statt; die Anzahl der Filter ist fest. Demgegenüber muß sich der hier präsentierte Ansatz aus einer gegebenen Anzahl von Gaborfiltern die zum Lernen geeignetsten suchen. Hier ist die Eignung über die Fehlerschätzung durch m–fache *cross validation* definiert. Diese Methode ist — gegenüber der Methode von Campbell — nicht mit einer vorgegebenen Anzahl von Filtern parametrisiert, sondern selektiert automatisch das beste Modell! Campbell erreichte auf einem 6–Texturen Bild eine Qualität von 83.4% was mit den Ergebnissen auf dem 4–Texturen Bild (max. 86.81%) vergleichbar ist.

5.3 Röntgenbilder

Dieser Datensatz wurde von Dr. Lucas Parra bei Siemens Corporate Research erstellt. Er enthält die Röntgenbilder von 15 Personen. Der Datensatz ist in Tabelle **??** dargestellt. Desweiteren sind die Ergebnisse von 6 speziellen Filtern, die auf die Röntgenbilder angewandt wurden, gegeben. Die gegebenen Filterbilder sind in den Tabellen **??** bis **??** dargestellt. Die Aufgabe besteht im Erlernen der optimalen Segmentierung (in Tabelle **??** dargestellt). Mit Hilfe dieser Segmentierung kann dann entschieden werden, welche Region des menschlichen Körpers abgebildet ist. Davon hängen die Einstellungen der Röntgenkanone für eine dauerhafte Bestrahlung während einer Operation[3] ab. In Tabelle **??** ist dargestellt, mit welcher Stärke (Kilovolt) einzelne Körperregionen bestrahlt werden müssen. Man sieht sehr deutlich, daß es — im Hinblick auf Strahlungsbelastung und Bildqualität — sehr wichtig ist, die richtigen Einstellungen zu wählen. So wird z.B. der Fingerknochen nur mit rund 50kV bestrahlt, während eine Wirbelsäule mit 80kV bestrahlt werden muß. Eine gute Segmentierung ist ausschlaggebend für eine korrekte Erkennung der abgebildeten Körperregion.

Zur Segmentierung wurden jeweils 12 bzw. 32 Gaborfilter mit den Bildern gefaltet und die Amplitudenbilder errechnet. Dazu wurde der Datensatz folgendermaßen in 3 Gruppen unterteilt: 9 Bilder der Wirbelsäule (*spine-*), 2 Bilder der Füße (*foot-*) und 4 Bilder der Beine (*leg-*). Aufgrund dieser Unterteilung ist die Lernbarkeit der Segmentierungen deutlich vereinfacht. Es wurden die gleichen Filterbänke und Lernverfahren wie im Abschnitt 5.2 benutzt. Ein Unterschied war bei diesen Versuchen, daß Trainingsbeispiele aus allen Bildern

[3] Die Röntgenbilder werden preoperativ aufgenommen.

Merkmalsselektion	Anzahl Gaborfilter		p–Wert
	Wirbelsäule		
	12	32	
keine FSS	0.7388 ± 0.0150	0.7568 ± 0.0156	0.9913
FFSS	0.7860 ± 0.0188	$\mathbf{0.7902} \pm 0.0089$	0.7324
BFSS	0.7426 ± 0.0075	0.7782 ± 0.0101	0.9992
	Fuß		
	12	32	
keine FSS	0.8698 ± 0.0057	0.8774 ± 0.0051	0.9987
FFSS	$\mathbf{0.9090} \pm 0.0116$	0.8988 ± 0.0047	0.0129
BFSS	0.8752 ± 0.0157	0.8896 ± 0.0061	0.9897
	Beine		
	12	32	
keine FSS	0.7794 ± 0.0108	0.7986 ± 0.0114	0.9994
FFSS	0.8448 ± 0.0076	$\mathbf{0.8626} \pm 0.0116$	0.9995
BFSS	0.7972 ± 0.0062	0.8100 ± 0.0121	0.9947

Tabelle 5.5: Geschätzte Genauigkeit der Segmentierung auf Röntgenbildern. In der letzten Spalte ist der p–Wert für den Test auf Verbesserung durch Verwenden von 32 anstatt 12 Gaborfiltern angegeben.

Merkmalsselektion	Anzahl Gaborfilter								
	Wirbelsäule			Fuß			Beine		
	12	32	durchschn.	12	32	durchschn.	12	32	durchschn.
keine FSS	19	39	29	19	39	29	19	39	29
FFSS	6	6	**6**	5	4	**4.5**	3	3	**3**
BFSS	18	38	28	18	38	28	16	37	26.5

Tabelle 5.6: Anzahl ausgewählter Merkmale bei Merkmalsselektion. Zusätzlich zu den Amplitudenbildern wurde der Grauwert und die 6 gegebenen Filterbilder als Merkmale benutzt.

einer Gruppe gezogen wurden. Dadurch sind die Testmengen (Bilder einer Gruppe) so groß, daß die Verzerrung, die sich aufgrund des *Subsamplings* der Trainingsmenge aus der Testmenge ergibt, vernachlässigbar klein ist.

Die Ergebnisse bei Anwendung von keiner, vorwärts- oder rückwärtsgerichteter Merkmalsselektion und Faltung mit jeweils 12 oder 32 Gaborfiltern sind in den Tabellen **??** bis **??** dargestellt.

Auswertung der Ergebnisse bei Röntgenbildsegmentierung

In Tabelle 5.5 sind, analog zum Abschnitt 5.2, die geschätzten Genauigkeiten für die drei Gruppen von Bildern bei alle Merkmalsselektionsverfahren mit jeweils 12 und 32 Gaborfiltern angegeben. Hinter der Genauigkeit ist die Standartabweichung der Genauigkeit protokolliert. Es wurden die Schätzungen benutzt, die sich bei der Merkmalsselektion ergaben.

Um zu untersuchen, ob es durch Merkmalsselektion zu Unterschieden zwischen den Genauigkeiten kommt, wurde die Gruppe „Wirbelsäule“ bei Benutzung von 32 Gaborfiltern ausgewählt. Die p–Werte sind in Tabelle 5.7 dargestellt. In allen anderen Gruppen der Tabelle 5.5 verbessert die vorwärtsgerichtete Selektion die Segmentierung auf einem Signifikanzniveau von 100%! Am deutlichsten ist der Unterschied bei der Gruppe „Beine“ und Anwendung von 32 Gaborfiltern. Der Unterschied beträgt hier 6.4%. Das sind 31.7% relativer Verlust an Segmentierungsfehler! Außerdem sind in Tabelle 5.6 die Anzahl benutzter Merkmale protokolliert.

Wirbelsäule			
	keine	FFSS	BFSS
keine	—	0.0001	0.0012
FFSS	**0.9999**	—	0.9941
BFSS	0.9988	0.0059	—

Tabelle 5.7: p–Werte bei Test auf höhere Genauigkeit zwischen zwei Selektionsmechanismen

Folgende Schlußfolgerungen können gemacht werden:

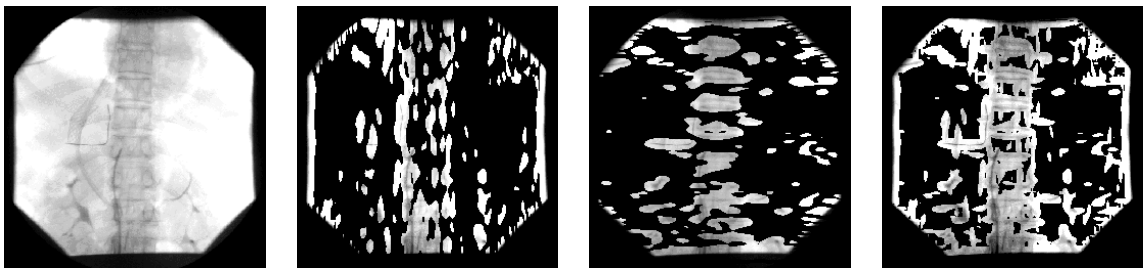
- Die vorwärtsgerichtete Merkmalsselektion verbessert die Segmentierung auch bei diesem Datensatz auf einem Signifikanzniveau von über 99% (Tabelle 5.7)! Dabei sind die Unterschiede beim Röntgenbilderdatensatz noch offensichtlicher als bei den Brodatztexturen (vergleicht man beispielsweise die Wirbelsäulenbilder in Tabelle **??** und **??**). Allerdings kann das Konzept „Fuß“ auch ohne Merkmalsselektion schon sehr gut gelernt werden (Fußbilder in Tabelle **??** und **??**). Dies liegt daran, daß dieses Konzept sehr einfach zu lernen ist. Benutzt man nur den Grauwert, erhält man schon eine Segmentierungsprozedur mit einer Genauigkeit von $81.36\% \pm 0.62\%$!

  Die signifikante Verbesserung durch vorwärtsgerichtete Selektion ist auch bei diesem Datensatz darauf zurückzuführen, daß durch Merkmalsselektion nur sehr wenige Merkmale zur letztendlichen Hypothesenindukti on ausgewählt werden. Interessant ist die Tatsache, daß auf den Wirbelsäulenbildern bei Benutzung von 12 und 32 Gaborfiltern immer genau 6 Merkmalsbilder zur Segmentierung benutzt wurden (Tabelle 5.6). Dabei wurden in beiden Fällen die gleichen Merkmalsbilder gewählt! Vergrößert man den Umfang der Trainingsmenge von 1000 auf 5000 Pixel — um die Verzerrung aufgrund des Trainingsdatenumfangs zu minimieren (siehe Abschnitt 3.2), bleibt die Qualität gleich und es werden wieder die gleichen 6 Merkmalsbilder ausgewählt. Ähnlich sieht es bei den „Beine“ Bildern aus. Auch hier werden mit 1000 wie mit 5000 Trainingsbeispielen genau 3 Merkmale ausgewählt. Zwei dieser drei Merkmale sind in jedem Fall der Grauwert und das Ergebnisbild des `BONE`–Filters.

- Die rückwärtsgerichtete Selektion verwirft auch bei diesem Datensatz im Durchschnitt nur ein Merkmal (Tabelle 5.6). Erneut sind die Amplitudenbilder der einzelnen Gaborfilterfaltungen sehr stark korreliert. Diese geringe Reduktion der Dimensionalität des Instanzenraumes führt zu keiner signifikanten Verbesserung gegenüber keiner Merkmalsselektion. Dies ist ebenfalls in den Segmentierungsergebnissen (Tabelle **??** und **??**) zu sehen.

- Erhöht man die Abtastung des Frequenzraumes von 12 auf 32 Filter, so kommt es zu einer signifikanten Verbesserung der Segmentierung (p–Werte > 0.99 aus Tabelle 5.5). Nur im Falle der Wirbelsäulenbildern und vorwärtsgerichteter Selektion ist kein signifikanter Unterschied festzustellen (p–Wert: 0.7324). Dies liegt daran, daß hier die gleichen Merkmale verwendet werden. Auch optisch kann eine kleine Verbesserung festgestellt werden. Vergleicht man beispielsweise *spine–3* bzw. *spine–8* bei den Segmentierungen ohne Merkmalsselektion (Tabelle **??** und **??**) oder *spine–1* bzw. *spine–5* bei den Segmentierungen mit rückwärtsgerichteter Selektion (Tabelle **??** und Tabelle **??**) , so fällt auf, daß die gefundenen Regionen kompakter sind, d.h. weniger Segmentierungsfehler an den Kanten zwischen zwei Regionen auftreten.

- Die Information, die in den Amplitudenbildern von Gaborfilterfaltungen enthalten ist, wird so gut wie möglich vom Lernverfahren benutzt. Dazu wurde exemplarisch das



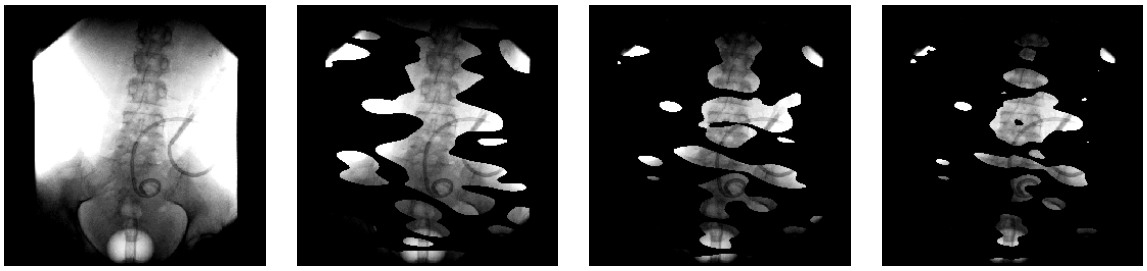
(a) (b) (c) (d)

Abbildung 5.4: **(a)** Wirbelsäulenbild *spine–8*. **(b)** Maskierung von (a) mit einem vertikal parametrisiertem Gaborfilter. **(c)** Maskierung von (a) mit einem horizontal parametrisierten Gaborfilter (niedrige Frequenz). **(d)** Überlagerung von (b) und (c).

vertikale und horizontale Gaborfilter mit der niedrigsten Frequenz aus der Filterbank (a) in Abbildung 5.2 ausgewählt. Die Amplitudenbilder dieser Filteranwendungen enthalten die meiste Information. Danach wurde ein Schwellwert für die Amplitudenbilder bestimmt, der die Wirbelsäule möglichst gut hervorhebt. In die aus den biniarisierten Amplitudenbildern erhaltenen maskierten Regionen wurde das ursprüngliche Bild eingeblendet (Bild (b) und (c) von Abbildung 5.4). Danach wurden die beiden so erhaltenen Maskierungen überlagert (Bild (d) von Abbildung 5.4). Man sieht sehr schön, daß — wie erwartet — nur die vertikalen und horizontalen Komponenten der Wirbelsäule ausmaskiert wurden. Vergleicht man dieses Muster mit dem Segmentierungsergebnis ohne Merkmalsselektion mit 12 Gaborfiltern (*spine–8* in Tabelle **??**), dann spiegeln sich Großteile des Überlagerungsbildes in der endgültigen Segmentierung wider.

- Bei vorwärtsgerichteter Selektion wurden **immer** sämtliche Amplitudenbilder der Gaborfilterfaltung verworfen. Betrachtet man dazu die Frequenzspektren aller Bilder dieses Datensatzes (Tabelle **??**) so fällt auf, daß es in keinem dieser Spektren signifikante, texturkennzeichnende *Peaks* gibt. Die Frequenzen streuen sehr stark um den Nullpunkt. Diese dichte Streuung deutet darauf hin, daß die Bilder aus sehr vielen zweidimensionalen Schwingungen mit niedriger Frequenz aufgebaut sind. Demzufolge müßten Gaborfilter, die auf diese niedrige Frequenzen parametrisiert sind, relevante Information

über die entsprechenden Amplitudenbilder liefern. Es ist schon in Abschnitt 2.4.4 hervorgehoben worden, daß Gaborfilter, die auf niedrige Frequenzen parametrisiert sind, im Ortsraum sehr breit streuen und damit schlecht zur Segmentierung geeignet sind. Dies ist am Beispiel des Bildes *spine–1* in Abbildung 5.5 dargestellt. Hier wurde eine

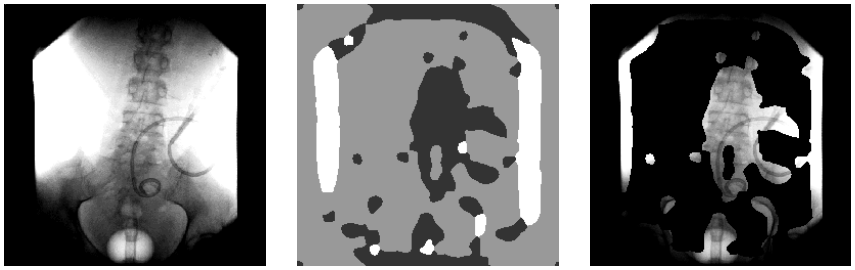


(a) (b) (c) (d)

Abbildung 5.5: **(a)** Wirbelsäulenbild *spine–1*. **(b)** Maskierung von (a) mit einem auf 10° orientierten Gaborfilter (17 Zyklen je Bild). **(c)** Maskierung von (a) mit einem auf 10° orientierten Gaborfilter (23 Zyklen je Bild). **(d)** Maskierung von (a) mit einem auf 10° orientierten Gaborfilter (30 Zyklen je Bild). Man sieht in (b) sehr schön, daß die Amplituden im Ortsraum sehr breit streuen, wenn das Gaborfilter auf eine niedrige Frequenz parametrisiert ist. Bei hohen Frequenzen kommt es zu vielen kleinen Regionen mit geringer Streuung.

Menge von Gaborfiltern gefunden, deren Orientierung genau der Orientierung der Wirbelsäule im Bild entsprach. Anschließend wurden manuell Schwellwerte gesucht, um aus den biniarisierten Amplitudenbildern möglichst gut die Wirbelsäule erkennen zu können. In die maskierten Regionen wurde dann das ursprüngliche Bild eingeblendet. Das Filter, das genau auf die Frequenz der Wirbelsäulenknochen parametrisiert ist (17 Knochen bzw. Zyklen pro Bild), streut sehr stark an den Rändern (Bild (b) in Abbildung 5.5). Erhöht man die Frequenz der Gaborfilter, verringert sich zwar die Streuung, aber die zu findende Region wird nun in kleinere Komponenten „zerlegt“. Das Filter in Bild (d) antwortete nur noch auf den etwas kleineren Mittelkörper der Wirbelsäule und zerlegt diese damit in viele kleine Regionen. Keine Kombination dieser Filter würde zufriedenstellende Segmentierungsergebnisse liefern. Diese Aussage wird auch von den Ergebnissen belegt, die mit dem System von Weldon und Higgins (1997) berechnet wurden (Abbildung 5.6). Dieses System schätzt nach dem *maximum–likelihood* Prinzip die Parameter von genau drei Gaborfiltern. Nachdem mehr als 2000 Pixel je Region vorgegeben wurden, ist keine Kombination der drei besten Gaborfilter in der Lage, das Bild zufriedenstellend zu segmentieren. Eine der drei berechneten Gaborfilter besitzt die gleichen Parameter wie das manuell bestimmte Gaborfilter in Abbildung 5.5 (Bild (b)).

- Im vorherigen Absatz wurde festgestellt, daß bei der vorwärtsgerichteten Merkmalsselektion nur die Ergebnisbilder der 6 Filter von Dr. Parra ausgewählt wurden. Dies führte in allen Fällen zu einer signifikanten Verbesserung. Aus diesem Grunde sollen diese Filter kurz beschrieben werden (Parra 1997). Das `TEX`–Filter besteht hauptsächlich aus einer speziell angepaßten Anwendungsreihenfolge und Parametrisierung von morphologischen Operatoren (Joo 1991). Bei dem `SPINE`–Filter wird zusätzlich nach den hori-



(a) (b) (c)

Abbildung 5.6: **(a)** Wirbelsäulenbild *spine–1*. **(b)** Segmentierungsergebnis mit dem System teXan (Weldon (1997)). **(c)** Überlagerung von (a) und (b).

zontalen Grenzen von Wirbelsäulenknochen im Bild gesucht und anhand des Abstandes entschieden, ob die gefundene Region zur einer Wirbelsäule gehört. Das `BONE`–Filter sucht zuerst nach Kanten im Bild und verbindet diese — mit massiv handangepaßten Algorithmen — zu einem Kurzenzug. Anschließend wird mit Hilfe einer festen Menge von Regeln entschieden, ob eine Region ein Knochen ist (Sind zwei Kurvenzüge parallel? Haben sie die gleiche Länge? usw.). Das `COLLIM`–Filter benutzt die Houghtransformation und adaptiert Schwellwerte, um Linien und Kreise im Bild zu finden. Hier wird mit einem festen Satz von Regeln entschieden, ob eine gefundene Primitive eine Kante einer Bleiplatte sein kann. Das `CONTR`–Filter schließlich ist die Anwendung eines lokalen Schwellwertalgorithmuses auf daß Bild. Die Lokalität sichert zu, daß scharfe Konturen (Darmbereich) nicht zerstört werden.

Man sieht sehr deutlich, daß diese Filter extrem angepaßt sind für die Segmentierung in Tabelle **??**. Solche massiv nichtlinearen Merkmale können nicht ohne domänenspezifisches Wissen gefunden werden. Es ist offensichtlich, daß solche Merkmale nicht mittels einer Kombination von Gaborfiltern modelliert werden können, da die Faltung mit einem Gaborfilter eine lineare Transformation ist (siehe Abschnitt 4.2) (Parra 1997).

5.4 Zellbilder

Mit diesem Datensatz soll die Qualität des in Kapitel 2 vorgestellte Verfahren der strukturellen Mustererkennung an einem praktischen Problem in der Gewebeschnittanalyse untersucht werden. Das Problem bei der Analyse von Gewebeschnitten ist die Bestimmung der Durchflußmenge eines Gefäßes. Dazu muß im ersten Schritt eine Segmentierung des Bildes in die einzelnen Zellen vorgenommen werden. Im nächsten Schritt wird versucht, anhand der Merkmale jeder Region (Zelle) zu erkennen, ob es sich um eine Gefäßwandzelle handelt. Je genauer diese Klassifikation durchgeführt werden kann, um so genauer kann die exakte Durchflußmenge des geschnittenen Gefäßes bestimmt werden. Der Datensatz wurde von Dr. Hufnagl an der Charité Berlin erstellt. Der Datensatz umfaßt die in Tabelle **??** dargestellten Gefäßbilder. Die Segmentierung wurde mit dem Bildverarbeitungssystem AMBA vorgenommen.

Beim vorliegenden Datensatz wurden aus der vorgegebenen Segmentierung die Merkmale in Tabelle 5.8 für jeden Knoten des RAG (jede Zelle) berechnet. Desweiteren stehen folgende

Knotenmerkmale	Wertebereich
Breite	$\mathbb{R}$
Höhe	$\mathbb{R}$
Länge der Kontur	$\mathbb{R}$
Geometrische Fläche	$\mathbb{R}$
Formfaktor	$\mathbb{R}$
Semikonvexitätsmaß	$\mathbb{R}$
Zelltyp (Endothel/Tumor)	*bool*

Tabelle 5.8: Zellmerkmale (Knotenmerkmale im RAG)

Algorithmus	Genauigkeit	Tests	Genauigkeit	Tests
	ohne Ellipsenprädikat		mit Ellipsenprädikat	
	propositionale Lernverfahren			
C4.5	**0.9250** $\pm$ 0.0277	34	0.9600 $\pm$ 0.0255	22
CAL5	0.9172 $\pm$ 0.0303	1	**0.9623** $\pm$ 0.0319	2
HERAKLES	0.8843 $\pm$ 0.0510	30	0.9422 $\pm$ 0.0483	17
	graphbasierte Lernverfahren			
TRITOP	0.9120 $\pm$ 0.0336	13	0.9583 $\pm$ 0.0316	5

Tabelle 5.9: Genauigkeiten auf dem Zellbilderdatensatz.

relationale Merkmale zur Verfügung:

- Ellipsenzugehörigkeit einer Zelle; die Ellipsenextraktion erfolgte mit dem Algorithmus (Roth und Levine 1993) auf den Regionenmittelpunkten (siehe dazu auch Abschnitt 2.3 und Anhang A)
- über die Voronoitriangulation berechnete Nachbarschaftsrelation (siehe Tabelle **??**, rechte Seite) einer Zelle; die Triangulation erfolgte mit dem Algorithmus in (Hufnagl et al. 1985)

Beim propositionalen Datensatz wird ein 7–dimensionaler Merkmalsvektor für jede Zelle verwendet, wobei der Zelltyp in $\{0; 1\}$ abgebildet wird. Darüber hinaus steht ein 8–dimensionaler Datensatz zur Verfügung, bei dem die Ellipsenzugehörigkeit jeder Zelle als achtes Merkmal in $0 \ldots n$ abgebildet wird. Jede Zelle wird der größten extrahierten Ellipse, auf der sie liegt, zugeordnet wobei n hier die Anzahl extrahierter Ellipsen bezeichnet. Beim relationalen Datensatz besteht darüber hinaus die Möglichkeit, eine Nachbarschaftsrelation über das Hintergrundwissen hinzuzufügen.

Es wurde jeweils mit den Lernsystemen C4.5, CAL5, HERAKLES und TRITOP mit und ohne Benutzung des Ellipsenprädikats versucht, daß Konzept „Gefäßwandzelle“ zu lernen. In Tabelle 5.9 sind die durch 10–fache *cross validation* entstehenden Ergebnisse zusammengefaßt. Hinter der Genauigkeit ist jeweils die Standartabweichung der Genauigkeit angegeben. Es ist

Algorithmus	p–Wert
C4.5	0.9954
CAL5	0.9976
HERAKLES	0.9908
TRITOP	0.9972

Tabelle 5.10: p–Wert bei Test auf Verbesserung durch Benutzung des Ellipsenprädikats

zusätzlich die durchschnittliche Anzahl von Tests protokolliert. Alle propositionalen Systeme brauchten zur Induktion unter einer Minute. Die graphbasierten Verfahren brauchten zwischen ein bis zwei Stunden.

Auswertung der Ergebnisse bei Gefäßwanderkennung

In Tabelle 5.10 ist der p–Wert für den Test auf Verbesserung der Genauigkeit durch Einführung des Ellipsenprädikats berechnet worden. Es ist sehr deutlich zu sehen, daß mit einer Wahrscheinlichkeit von über 99% die Genauigkeit bei Benutzung des Ellipsenprädikats höher ist als ohne dieses Prädikat. Dabei spielte es keine Rolle, in welcher Form dieses Prädikat repräsentiert wurde! Dieses Merkmal ist ein Komplexmerkmal, d.h. ein strukturelles Merkmal, was die Beziehung einer einzelnen Zelle zu einer räumlichen Struktur (Ellipse) widerspiegelt. Es läßt sich nur errechnen, wenn die Position aller anderen Zellen bekannt ist. Solche Komplexmerkmale werden in dem meisten Fällen zur Restriktion der Hypothesensprache benutzt (Geibel und Wysotzki 1996). Zum einen wird damit die Induktion deutlich beschleunigt. Zum anderen führen propositionale Beschreibungen durch Komplexmerkmale zu einem starken absoluten *Bias* gegenüber der graphischen Repräsentation. Damit sinkt eventuell der echte Fehler der induzierten Hypothese (Abbildung 3.3).

Ersetzt man das Ellipsenprädikat durch eine Nachbarschaftsrelation wie oben beschrieben, ist die graphbasierte Version von HERAKLES und TRITOP nicht mehr in der Lage, daß Konzept zu lernen (Genauigkeit der graphbasierten Version von HERAKLES: $90.20\% \pm 4.35\%$). Dazu müßte ein ähnliches Konzept wie „Ellipse“ ausschließlich durch eine logische Formel über Nachbarschaftsrelationen ausgedrückt werden. Obwohl dies durchaus möglich ist, ist diese Formel so komplex, daß sie aufgrund des relativen *Bias* nicht generiert wird! Desweiteren steigt mit der Anzahl von Relationen bei ILP–Systemen die Modellkomplexität an. Dies führt unter Umständen dazu, daß der echte Fehler steigt (Abbildung 3.3).

Weiterhin sollte untersucht werden, ob bestimmte Lernverfahren geeigneter als andere zum Erlernen des Konzepts „Gefäßwand“ sind. In Tabelle 5.11 sind die p–Werte für den Test auf Unterschiede zwischen allen benutzten Lernverfahren berechnet worden. Man sieht hier sehr deutlich, daß es keine signifikanten Unterschiede zwischen den Genauigkeiten der einzelnen Lernsystem gibt. Die höchste Wahrscheinlichkeit für einen Unterschied gibt es zwischen den Systemen CAL5 und HERAKLES mit 85.53%. Diese ist aber statistisch nicht signifikant. Zu beachten bleibt, daß alle vier Systeme eine *Top–Down* Induktion durchführen, d.h. nach einem ähnlichen Schema im Refinementgraph nach der besten Hypothese suchen. Betrachtet man allerdings die Anzahl der Tests, die vom jeweiligen System als Hypothese generiert wurden, so fällt folgendes auf:

	C4.5	CAL5	HERAKLES	TRITOP
C4.5	—	0.4304	0.8392	0.5805
CAL5	0.5696	—	**0.8553**	0.6488
HERAKLES	0.1608	0.1447	—	0.1601
TRITOP	0.4195	0.3512	0.8399	—

Tabelle 5.11: p–Werte bei Test auf höhere Genauigkeit zwischen zwei Lernsystemen; es wurden die Genauigkeiten bei Benutzung des Ellipsenprädikates betrachtet.

- CAL5 findet in beiden Fällen, d.h. mit und ohne Benutzung des Ellipsenprädikates, die kürzeste Hypothese. Die durch CAL5 gefundene Hypothese bei Benutzung des Ellipsenprädikates lautet schlicht und einfach: `if` *Endothelzelle* `and` *auf größter Ellipse* `then` *Gefäßwandzelle* `else` *Nicht–Gefäßwandzelle*. Diese Hypothese wurde in allen zehn Durchläufen von *cross validation* gefunden. Bei allen anderen Lernsystemen variiert die Anzahl der Tests bedeutend stärker.

- HERAKLES und C4.5 bilden durchschnittlich gleich viele Tests in den Hypothesen. Dies liegt daran, daß beide Systeme den *information gain* als Gütekriterium für Tests bei der Regelinduktion benutzen.

- TRITOP lernt kurze Entscheidungsbäume mit wenigen Tests. Wird das Ellipsenprädikat benutzt, weicht die Anzahl der Tests nur in 3 der 10 Durchläufe um einen Test vom Mittelwert 5 (Tabelle 5.9) ab. Dies liegt daran, daß TRITOP bei der Testgenerierung *top–down* vorgeht. HERAKLES erzeugt die Tests dagegen nach einem *bottom–up* Schema (siehe Abschnitt 3.1.2 in Schritt 2).

Abschließend wurde untersucht, ob es durch die Benutzung des Ellipsenprädikates sichtbar zu einer Verbesserung kommt. Dazu wurden jeweils zwei der drei Zellbilder zum Lernen des Klassifikator und das verbleibende Zellbild zum Validieren benutzt. In der Tabelle **??** sind die Ergebnisse dargestellt. Es wurde HERAKLES als Lernsystem benutzt. Es ist grafisch dargestellt, wenn es zu einer Fehlklassifikation kam. Alle weißen Flächen sind richtig klassifiziert. Alle hellgrauen Flächen sind zwar Gefäßwandzellen, wurden aber nicht als Gefäßwandzellen klassifiziert (falsch negativ beurteilt). Alle dunkelgrauen Flächen sind keine Gefäßwandzellen, wurden aber als Gefäßwandzellen klassifiziert (falsch positiv beurteilt). Folgende Schlußfolgerungen können gezogen werden:

- Zu sichtbaren Verbesserung kam es beim zweiten und dritten Zellbild. Die Anzahl der falsch negativen Zellen im zweiten und die Anzahl der falsch positiven Zellen im dritten Zellbild ist deutlich kleiner geworden. Im ersten Zellbild kam es zu keiner sichtbaren Veränderung.

- Die Problemverteilung ist in den einzelnen Bildern zu unterschiedlich um einen signifikanten Klassifikator zu lernen. Dies zeigt sich daran, daß die Genauigkeit mit diesem Testverfahren 91% gegenüber 96% bei zehnfacher *cross validation* beträgt. Um optisch deutlichere Aussagen zu erhalten, müßte die Anzahl der Bilder erhöht werden, was in diesem Fall aber nicht möglich war.

Kapitel 6

Zusammenfassung und Ausblicke

In dieser Arbeit wurde ein überwachter Segmentierungsalgorithmus vorgestellt, der mit sich Hilfe von Regelbauminduktion und Merkmalsselektion sehr gut an eine gegebene Problemverteilung anpaßt. Dazu wird die eindimensionale Grauwertinformation eines Pixels durch Faltung mit einer Filterbank von Gaborfiltern oder Anwendung domänenspezifischer Filter in einen hochdimensionalen Merkmalsraum transformiert. In diesem Merkmalsraum ergibt sich beim Lernen allerdings ein Modellselektionsproblem. Versucht man ohne Modellselektion einen Regelbaum zu induzieren, paßt man sich mit hoher Wahrscheinlichkeit ausschließlich an die Trainingsmenge an. Aus diesem Grund wird mit Hilfe von Merkmalsselektion die hochdimensionale Information wieder — abhängig von der Problemverteilung — in einen niedrigdimensionalen Raum projiziert. Die gefundene Merkmalsmenge beschreibt für das gegebene Problem das beste Modell. Dieses beste Modell muß nicht das optimale Modell sein, da aufgrund der Komplexität nur mit einer Gradientenstrategie nach dem besten Modell gesucht werden kann. Anschließend wird mit Hilfe der gefundenen Merkmalsmenge und eines Regelbaumlernverfahrens eine Segmentierungsprozedur induziert, deren erwarteter echter Fehler kleiner ist als ohne Merkmalsselektion oder unter ausschließlicher Benutzung des Grauwertes (rechte und linke Ende der Kurve „echter Fehler“ in Abbildung 3.3).

Diese Idee, die Dimensionalität des Instanzenraumes massiv zu erhöhen — mit der Hoffnung, die Problemverteilung besser approximieren zu können, liegt auch *Support–Vector–Machines* (SVM) (Cortes und Vapnik 1995) zugrunde. Allerdings wird bei SVMs anstelle eines Regelbaumes eine „optimale“ Trennebene gelernt. Der Lernalgorithmus für diese Trennebene trägt dem Modellselektionsproblem durch Addition eines modellkomplexitätsabhängigen Strafterms Rechnung. Außerdem ist die Dimensionalitätserhöhung bei SVMs eine nichtlineare Transformation.

Wie erwartet ist dieser Algorithmus in der Lage, das Standartproblem in der Textursegmentierung — die Segmentierung eines Karomusters aus unterschiedlichen Texturen aus Brodatz (1966) — zu lösen. Dabei erreicht der parameterlose Algorithmus zu den in (Campbell und Thomas 1997) vorgestellten Ergebnissen vergleichbare Resultate. Im System von Campbell müssen allerdings die Anzahl der Gaborfilter und die Parameter des Lernsystems SOFM vorgegeben werden. Ein vergleichbares Ergebnis wird auch in der Arbeit von Weldon und Higgins (1997) erzielt. Hier werden bei einer vorzugebenden Anzahl von Gaborfiltern die besten Gaborfilterparameter nach dem *maximum–likelihood* Prinzip geschätzt. Demgegenüber sind die Ergebnisse von Greenspan (1996) deutlich besser. Hier wurde anstelle des Regelbaumlernsystems HERAKLES das Lernsystem ITRULE (Goodman et al. 1992) zur Regelinduktion

benutzt. Das System ITRULE besitzt sehr viele Parameter. Es wurden massiv Parameteranpassungen vorgenommen, was mit hoher Wahrscheinlichkeit zu einer optimistischen Verzerrung des echten Fehlers führte. Desweiteren wurden sehr kleine Testmengen verwendet (4 Testpixel je Klasse!). In den Experimenten, in denen Testmengen mit bis zu 100 Beispielen je Klasse verwendet wurden, erhielt man die gleichen oder sogar schlechtere Ergebnisse als mit dem hier präsentierten Ansatz!

Darüberhinaus ist dieser Segmentierungsalgorithmus in der Lage, einen Röntgenbilderdatensatz zu segmentieren. Allerdings liefern die Faltungsbilder von Gaborfiltern keine Information, welche die Regelbauminduktion verbessern würde. Dies hat zwei Ursachen: Auf der einen Seite sind Gaborfilter nur zur Segmentierung geeignet, wenn sich die einzelnen Regionen durch lokale *Peaks* im Frequenzspektrum auszeichnen. Dies ist bei den Röntgenbildern nicht der Fall. Zum anderen ist die Faltung eine lineare Transformation. In diesem Datensatz sind aber nichtlineare Transformationen des Bildes notwendig, um relevante Merkmale zur Segmentierung zu erhalten. Nichtsdestotrotz ist der hier vorgestellte Algorithmus durch vorwärtsgerichteter Merkmalsselektion in der Lage, parameterlos die relevanten nichtlinearen Merkmale zu selektieren und verbessert dadurch signifikant und sichtbar die Segmentierungsergebnisse. Der in (Weldon und Higgins 1997) beschriebene Segmentierungsalgorithmus war nicht in der Lage, das Problem zu lösen.

Es stellte sich heraus, daß rückwärtsgerichtete Merkmalsselektion in keinem Fall zu einer Verbesserung führte. Es existieren zu viele Scheinkorrelationen zwischen den Amplitudenbildern von Gaborfilterfaltungen.

Die Bildsegmentierung ist nur ein Teilprozeß in der Musterkennung (siehe Abbildung 2.1). Eine Segmentierung des Bildes ist vor allem dann wichtig, wenn der Bildinhalt in eine graphische Repräsentation überführt werden soll. Diese Repräsentation ist notwendig, wenn eine strukturelle Musterbeschreibung induziert werden soll. Aus diesem Grunde wurde an einem gegebenem Problem untersucht, ob strukturelle Mustererkennung zu besseren Ergebnissen als statistische Mustererkennung führt. Es stellte sich heraus, daß das Problem mit Strukturinformation signifikant besser gelöst werden konnte. Allerdings war keines der benutzten graphbasierten Lernverfahren in der Lage, die notwendige Strukturinformation auf Grundlage einfacher Relationen über dem RAG zu induzieren. Bei graphbasierten Lernverfahren ergibt sich aufgrund des theoretisch unbeschränkt großen Hypothesenraumes ein Modellselektionsproblem, daß durch die hier verwendeten Algorithmen nicht gelöst werden konnte. Es mußte explizit ein komplexes relationales Merkmal berechnet werden. Mit Hilfe dieses Merkmals konnte die Genauigkeit der Mustererkennung verbessert werden. Dabei waren graphbasierte wie propositionale Lernverfahren gleichermaßen geeignet. Durch den kleineren Hypothesenraum waren die propositionalen Lernverfahren sogar in der Lage, kürzere Musterbeschreibungen zu finden.

So stellt sich abschließend die Frage: Was bleibt noch zu tun? Im folgenden sollen zwei mögliche Ansätze für die weitere Forschung vorgestellt werden.

Varianzminimierung. Der vorgestellte Algorithmus sucht nach dem Modell (Merkmalsmenge), in dem der erwartete echte Fehler einer in diesem Modell induzierten Hypothese am kleinsten ist. Allerdings wird im Anschluß nur genau eine Hypothese induziert. Im besten gefundenen Modell ist die Varianz des Lernverfahrens im allgemeinen aber nicht 0. Dadurch kann es bei der einmaligen Induktion dazu kommen, daß trotz optimalem Modell eine Hypothese induziert wird, die einen großen echten Fehler hat (siehe Tabelle **??** und **??**). In

der weiteren Forschung soll deshalb untersucht werden, inwieweit man mit der Kombination mehrerer Hypothesen diese Varianz minimieren kann. Die Idee hierbei ist, l unabhängige Hypothesen zu induzieren. Dazu müssen — nachdem das beste Modell gefunden wurde — l Trainingsmengen der Größe $\|S\|$ unabhängig aus den Bilddaten gezogen werden. Bei der großen Menge an Trainingsdaten ist dies sehr leicht möglich. Anschließend wird für jede der l Trainingsmengen eine Hypothese h^{min} induziert. Bei der Pixelklassifikation wird der extrahierte Merkmalsvektor durch alle l gelernten Hypothesen klassifiziert. Das Pixel wird dann mit der häufigsten Klasse über alle l Klassifikationen klassifiziert. In Tabelle **??** sind erste Ergebnisse auf dem Röntgenbilderdatensatz dargestellt. Es wurden $l = 10$ Regelbäume induziert. Vergleicht man diese Ergebnisse mit denen in Tabelle **??** und **??**, so ist eine sichtbare Verbesserung der Segmentierung festzustellen. Außerdem zeigte sich auch bei mehreren Durchläufen ein konstantes Segmentierungsergebnis! Allerdings muß noch untersucht werden, wie man diese Methode beschleunigen kann und ob es allgemeine Schranken für die Anzahl l der Hypothesen gibt.

Rückkopplung. Durch die Merkmalsexpansion mit Gaborfiltern werden in der induzierten Segmentierungsprozedur ausschließlich Regeln über die Amplituden der Faltungen mit einzelnen Gaborfilter — dem Verschaltungsmuster einfacher Zellen im V1 mit der Retina (siehe Abschnitt 2.4.2) — zur Segmentierung benutzt. Es ist bekannt, daß auch in der höheren Bildverarbeitung der menschlichen visuellen Wahrnehmung die Ausgaben einzelner Zellen des V1 durch ein Neuron kombiniert werden. Die dem hier präsentierten Algorithmus zugrundeliegende Annahme ist allerdings, daß es keine Verbindung zwischen den einzelnen Pixeln (V1–Zellen mit gleicher Orientierung) gibt. Allerdings ist bekannt, daß zwischen den einfachen V1–Zellen gleicher Orientierung Verbindungen bestehen[1]! Eine diese Tatsache berücksichtigende Erweiterung würde in einer Rückkopplung der gelernten Segmentierungsprozedur zur erneuten Merkmalsexpansion bestehen. Nachdem mit dem Verfahren in Abbildung 4.1 eine erste Segmentierungsprozedur gelernt wurde, werden die Beispielbilder mit dieser Prozedur segmentiert. Anschließend können dann pixelweise räumliche Relationen über die Regionen, zu denen die einzelnen Pixel gehören, berechnet werden. Dadurch erhält man neue, auf der ersten induzierten Segmentierungsprozedur basierende, Merkmale. Anschließend wird versucht, eine zweite Segmentierungsprozedur zu lernen. Dieser Prozedur würde die Relationen zwischen den Regionen, zu denen einzelne Pixel gehören, berücksichtigen. Allerdings hat sich in ersten Experimenten gezeigt, daß diese Merkmale keine Information enthalten, welche die Segmentierung der Röntgenbilder verbessert. Durch die Merkmalsselektion beim Lernen der zweiten Segmentierungsprozedur wurden alle relationalen Merkmale verworfen. Problematisch an diesen Merkmalen ist, daß ihre Benutzung zwar zu einer Reduktion der ursprünglich falsch klassifizierten Pixel führt, in gleichem oder stärkerem Maße aber schon ursprünglich richtig klassifizierte Pixel wieder falsch klassifiziert werden. Dies liegt daran, daß der verwendete Lernalgorithmus keine Schätzung des echten Fehlers einer Regel liefern kann. Dadurch kann nicht zwischen den gemachten Entscheidungen der beiden Segmentierungsprozeduren diejenige mit dem kleineren Fehler ausgewählt werden. In der Zukunft soll aus diesem Grunde ein Lernalgorithmus entwickelt werden, der Schätzungen für den echten Fehler einer einzelnen Regel liefert. Ein Pixel wird dann immer so klassifiziert, daß der geschätzte echte Fehler minimiert wird.

[1] Dadurch können in der menschlichen visuellen Wahrnehmung räumliche Relationen auf dem Bild bei der Objekterkennung berücksichtigt werden.

Anhang A

Mathematische Grundlagen

Ellipsenextraktionsverfahren

Das RMS Verfahren (Roth und Levine 1993) zur Extraktion von Primitiven aus einer Menge von Punkten beruht darauf, daß zu einer gegebenen minimalen Teilmenge von R Punkten die Parameter der entsprechenden Primitive berechnet werden können. Diese Parameter werden zur Abstandsberechnung benötigt. In Roth und Levine (1993) wird vorgeschlagen, daß man eine implizite Repräsentation der Primitive wählt, die linear in den gesuchten Parametern ist. So ist beispielsweise $f(\boldsymbol{a}, x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_3 y^2 = 0$ eine implizite Repräsentation eines beliebigen Kreises. Diese Gleichung ist linear in den gesuchten Parametern a_0, a_1, a_2 und a_3, wenn auch nicht in x und y. Die implizite Beschreibung der unbeschränkten Ellipse lautet:

$$f(\boldsymbol{a}, x, y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 = 0$$

Die Berechnung der a_i erfolgt jetzt über die Lösung des Gleichungssystems $\boldsymbol{X}\boldsymbol{a} = 0$. Die Lösung dieses Gleichungssystems liefert die gesuchten Parameter $\boldsymbol{a} = (a_0, \ldots, a_5)^T$. Im Fall der unbeschränkten Ellipse hat $\boldsymbol{X}$ folgende Gestalt:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x & y & xy & x^2 & y^2 \\ 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3 y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4 y_4 & x_4^2 & y_4^2 \\ 1 & x_5 & y_5 & x_5 y_5 & x_5^2 & y_5^2 \end{bmatrix}$$

Die symbolische Lösung für die a_i kann mit Hilfe des Mathematiksystems MAPLE oder MATHEMATICA bestimmt werden. Da diese Lösung über 5 Seiten füllt, wird auf die Angabe verzichtet. Hat man die Lösungen dieses Gleichungssystems bestimmt läßt sich bei gegebenem Vektor $\boldsymbol{a}$ und einem Punkt (x_p, y_p) der Abstand des Punktes (x_p, y_p) über die Gleichung $\frac{f(\boldsymbol{a}, x_p, y_p)}{\|\nabla f(\boldsymbol{a}, x_p, y_p)\|}$ sehr gut approximieren.

Anhang B

Implementierung

Das Segmentierungssystem GTS ist vollständig in C programmiert und bietet die Möglichkeit, eine Fülle von Algorithmen und Verfahren zu kombinieren, um Grauwertbilder mit Gaborfiltern zu segmentieren. Das System arbeitet kommmandozeilenorientiert. Die Parameter bezeichnen jeweils die Dateinamen eines Bildes und seiner Segmentierung. Diese Liste von Dateinamen kann auch wahlweise in einer Datei gespeichert werden, deren Dateiname dann den einzigen Parameter darstellt. Allerdings muß diesem Dateinamen ein `@` vorangestellt werden. Als letzter Parameter in der Kommandozeile muß der Name einer Steuerdatei angegeben werden. In dieser Steuerdatei finden sich alle Variablen, die während dieser Segmentierung benutzt werden. Das System arbeitet mit Bildern im BIFF–Format. Dieses Dateiformat wird im Bildverarbeitungspacket XITE benutzt.

Dateiformat

Eine GTS–Datei besteht aus Kommentar oder Variablenzuweisung. Ein Kommentar wird am Zeilenanfang mit einem `#` eingeleitet und erstreckt sich über den Rest dieser Zeile. Eine Variablenzuweisung hat immer das Format `<variable> '=' <wert>`, wobei für Wert jede Zeichenkette zugelassen ist. Leerzeilen werden überlesen. Es gibt drei Typen von Variablen. Werte numerischer Variablen können als Fließkommazahl oder in Exponentialschreibweise eingegeben werden. Die Werte von Zeichenkettenvariablen dürfen keine Leerzeichen enthalten und können nicht gequotet werden. Wahrheitswerte werden über `on` oder `off` dargestellt. Ein Beispielaufruf, welcher das Bild `example.img` und die Segmentierung `example.CLASS` mit der Steuerdatei `supervised.gts` verarbeitet, sieht also so aus

```
gts example.img example.CLASS supervised.gts
```

Im folgenden sollen die allgemeinen Systemvariablen erklärt werden. Dabei werden die Variablentypen über *num*, *string* und *bool* angezeigt.

`algorithm` <*string*> Diese Variable bestimmt den zur Segmentierung zu benutzenden Algorithmus. Möglich Werte sind `max` (Bovik et al. 1990), `em`, `som` (Campbell et al. 1996), und `supervised`. Standartwert ist `max`.

`bankselect` <*string*> Diese Variable bestimmt, welcher Algorithmus zum Finden der besten Filter benutzt werden soll. Die beiden möglichen Werte sind `user` und `sensor`. Standartwert ist `user`. Der Algorithmus bei der Einstellung `sensor` ist zur Zeit noch nicht implementiert.

`postfilter` <*num*> Diese Variable bestimmt, ob und mit welchem Verhältnis eine Postfilterung der Amplitudenbilder durchgeführt werden soll (Weldon et al. 1996). Die möglichen Werte reichen von 1.0 bis maximal 10. Standartwert ist `0`, d.h. keine Postfilterung.

`freqcoverage` <*string*> Diese Variable gibt den Dateinamen an, in dem der Frequenzraum und die Überlappung durch die gewählten Filter angezeigt werden soll. Ist standartmässig nicht gesetzt, d.h. es wird kein Bild erzeugt.

`amplitude` <*bool*> Dieses Flag zeigt an, ob die berechneten Amplitudenbilder ausgegeben werden sollen. Die Dateinamen bilden sich dann jeweils aus dem originalen Dateinamen mit einem vorangestellten `amp-`. Standartwert ist `off`. ACHTUNG: Diese Dateien können sehr groß werden!

`response` <*bool*> Dieses Flag zeigt an, ob die berechneten Faltungsbilder ausgegeben werden sollen. Die Dateinamen bilden sich dann jeweils aus dem originalen Dateinamen mit einem vorangestellten `response-`. Standartwert ist `off`. ACHTUNG: Diese Dateien können sehr groß werden!

`dumpgabor` <*bool*> Dieses Flag zeigt an, ob die berechneten Gaborfilterbilder in der Datei `gabor.img` ausgegeben werden sollen. Standartwert ist `off`. ACHTUNG: Diese Datei kann sehr groß werden!

`gabor` <*bool*> Dieses Flag zeigt an, ob Gabormerkmale zur Regelinduktion benutzt werden sollen. Standartwert ist `on`.

`reduce` <*num*> Mit dieser Variablen wird gesteuert, wieviel Pixel beim überwachten bzw. unüberwachten Segmentieren benutzt werden sollen[1]. Dabei kann man entweder die absolute Anzahl (Wert >1) oder die relative Anzahl (Wert <1) angeben.

`filtertype` <*string*> Diese Variable gibt an, welche *gabor wavelets* in der Gaborfilterbankberechnung benutzt werden sollen. Mögliche Werte sind `lee` (Lee 1996), `maclennan` (MacLennan 1991) und `bovik` (Bovik et al. 1990). Standartwert ist `lee`.

`ext[0---256]` <*string*> Diese Variablen geben Erweiterungen für den Basisnamen der zu segmentierenden Bilddateien an. Jede dieser Dateien wird mit eingelesen und als domänenspezifisches Merkmalsbild bei der Segmentierung benutzt. Standartmäßig sind diese Variablen nicht gesetzt, d.h. es werden keine Zusatzbilder benutzt.

Benutzerdefinierte Filter

Wenn die Variable `bankselect` den Wert `user` hat, so kann je nach Filtertyp eine Rosette im Frequenzraum parametrisiert werden. Dazu werden folgende Variablen benutzt.

[1] Ausnahme: wenn `algorithm` den Wert `max` hat, wird diese Variable ignoriert.

`freqstart` <*num*> Diese Variable gibt in Einheiten von Bildbreite an, mit welcher Frequenz beim Aufbau der Rosette begonnen werden soll. Der Wert 0.5 steht hier also für die Nyquistfrequenz, d.h. die höchste im Bild abtastbare Frequenz. Standartwert ist `0.5`.

`freqdepth` <*num*> Diese Variable gibt an, wieviel verschiedene Frequenzen je Orientierung zur Berechnung der Rosette benutzt werden sollen. Standartwert ist `4`.

`freqfactor` <*num*> Diese Variable bestimmt den Faktor, mit dem die Frequenzen der nächsthöheren Tiefe multipliziert wird. Dabei steht ein Wert von 0.5 beispielsweise für eine Halbierung der Frequenz (siehe auch `freqdepth`). Standartwert ist hier `0.5`.

`notheta` <*num*> Diese Variable bestimmt die Anzahl der Orientierungen, in denen ein Filter der Rosette generiert wird. Dieser Wert bezieht sich immer auf `maxtheta`. Standartwert ist hier `8`.

`maxtheta` <*num*> Diese Variable gibt die maximale Orientierung in *rad* an, die bei der Erzeugung von Orientierungen benutzt werden soll. Diese Variable braucht im allgemeinen nicht geändert zu werden und ist standartmäßig 2π.

Speziell bei Benutzung von `filtertype = lee` kann noch spezifiziert werden:

`bandwidht` <*num*> Diese Variable gibt die Bandbreite jedes Gaborfilters in Oktaven an. Sie kann nur zwischen 1.0 und 1.5 liegen. Standartwert ist `1.0`.

Speziell bei Benutzung von `filtertype = maclennan` kann noch spezifiziert werden:

`lambda` <*num*> Diese Variable gibt das Verhältnis zwischen Standartabweichung der Gaborfilter in X– und Y–Richtung im Ortsraum an. Standartwert ist `1.0`.

Unüberwachtes Segmentieren

Zum unüberwachten Segmentieren muß die Variable `algorithm` auf eine der Werte `max`, `em`, `som` oder `split` gesetzt sein.

Beim `max` Verfahren wird jedes Pixel in die Klasse eingeordnet, in welcher die entsprechende Gaborfilterantwort (Amplitudenbild) maximal war. Dieses Verfahren hat keine weiteren Parameter.

Beim `em` Verfahren wird eine reduzierte Auswahl der Pixel benutzt, um mit Hilfe des EM–Algorithmus eine Clusterung im Merkmalsraum durchzuführen. Mit Hilfe der Cluster wird dann segmentiert, d.h. ein Pixel wird in die Klasse seines Clusters eingeordnet. Parameter sind ausschließlich

`cluster` <*num*> Gibt die Anzahl der Cluster an, die benutzt werden, um eine Clusterung durchzuführen. Standartwert für die Diskrimination von zwei Texturen ist `2`.

Beim `som` Verfahren wird eine Clusterung mit einer SOFM durchgeführt . Parameter sind hier

`cluster` <*num*> Gibt die Anzahl der Cluster an, die benutzt werden, um eine Clusterung durchzuführen. Standartwert für die Diskrimination von zwei Texturen ist `2`.

`eta` <*num*> Gibt die Gewichtung der Nachbarschaftsrelation in der SOFM an. Standartwert ist `0.2`.

`learnrate` <*num*> Gibt die Lernrate an, d.h. mit welchem Faktor wird `eta` multipliziert, wenn ein Beispiel angeboten wurde. Standartwert ist `0.99`.

`somradius` <*num*> Gibt den Radius an, in dem ein Neuron des SOFM einwirkt, d.h. die maximale Anzahl aufeinanderfolgender Kanten, bis die Gewichtung der Nachbarschaft nur noch 0 beträgt. Standartwert ist `1`.

Überwachtes Segmentieren

Beim überwachten Segmentieren muß die Variable `algorithm` auf `supervised` gesetzt sein. Nach dem Erzeugen der Amplitudenbilder werden die durch `reduced` angegebene Anzahl von Pixeln zum Lernen eines Pixelklassifikators aus den Bildern gezogen. Der Ergebnisregelbaum wird dann in der Datei gespeichert, deren Dateinamen sich jeweils aus dem originalen Dateinamen mit einem vorangestellten `rdr-` bildet. Im folgenden wird mit Hilfe des gelernten Klassifikators eine pixelweise Klassifikation durchgeführt. Parameter sind hier:

`beta` <*num*> Gibt die Genauigkeit an, die der Regelbaum auf den Trainingsdaten (-pixeln) erreichen soll. Standartwert ist `1.0`.

`prune` <*num*> Gibt die minimale Anzahl von Beispielen der Trainingsmenge an einem Knoten des gelernten Baumes an. Über diesen Parameter wird das Pruning des Baumes gesteuert. Standartwert ist `10`, d.h. mindestens 10 Pixel müssen durch die Regel klassifiziert werden.

`fss` <*string*> Gibt die verwendete Merkmalsselektionsmethode an. Mögliche Werte sind `forward`, `backward` und `none`. Standartwert ist `forward`, d.h. es wird mit Hilfe eines *Hill–Climbing* Algorithmuses (Tabelle 3.2) versucht die beste Teilmenge von Merkmalen zu finden.

`useall` <*bool*> Gibt an, ob nur ein oder alle Beispielbilder zur Extraktion der Trainingsdaten benutzt werden soll. Standartwert ist `on`, d.h. es werden aus allen Bildern Pixel gezogen.

Anhang C

Notationen

Notation	definiert auf Seite	Beschreibung
$\mathbb{R}$	5	reele Zahlen
$\mathbb{N}$	5	natürliche Zahlen
$\boldsymbol{j}$	12	imaginäre Einheit mit $\boldsymbol{j}^2 = -1$
$\mathcal{P}$	8	Potenzmenge
$\boldsymbol{a}$	9	Vektor a
$\boldsymbol{x}$	9	Vektor über algebraische Verknüpfungen von x und y
$\boldsymbol{X}$	48	Matrix X
$\mathfrak{D}$	5	Definitionsbereich eines Bildes
$\mathfrak{R}$	5	Wertebereich eines Bildes
$\mathcal{N}$	8	Nachbarschaft eines Punktes $\mathcal{N} : \mathfrak{D} \rightarrow \mathcal{P}(\mathfrak{D})$
P	8	Segmentierungsprädikat $P : \mathcal{P}(\mathfrak{D}) \rightarrow \mathit{bool}$
$f \otimes g$	10	Faltung von f und g
$\mathfrak{F}_n^{-1}, \mathfrak{F}_n^1$	11	(inverse) n–dimensionale Fouriertransformation
$g_{(\sigma_x, \sigma_y)}$	12	zweidimensionale Gaußglocke
$\mathcal{X}$	17	Instanzenraum
Y	17	Menge der Klassensymbole
S	17	Trainingsmenge
x	17	Instanz $x \in \mathcal{X}$
X_i	17	i–tes Merkmal des Instanzenraumes
x_i	17	Ausprägung des i–ten Merkmals der Instanz x
y	17	Klasse $y \in Y$
h	17	Hypothese $h : \mathcal{X} \rightarrow Y$
$\mathcal{H}$	17	Hypothesenraum $\mathcal{H} = \{h\}$
$\mathcal{C}$	17	Konzpet, d.h. Klassenverteilung über dem Instanzenraum
t	20	Test über dem Instanzenraum $t \in \mathcal{H}$
T	20	Testerzeugungsfunktion $T : S \rightarrow \mathcal{P}(\mathcal{H})$
Q	20	Qualitätsfunktion eines Tests, $Q : \mathcal{H} \rightarrow \mathbb{R}$
h^{min}	21	Hypothese mit minimalen Trainingsfehler
m	22	Anzahl Durchläufe bei *cross validation*
n	26	Anzahl Merkmale über dem Instanzenraum $\mathcal{X} = \mathbb{R}^n$
B	18	Kopf einer Hornklausel $A \rightarrow B$ (Konklusio)
A	18	Körper einer Hornklausel $A \rightarrow B$

Literatur

Bigun, J. und J. D. Buf (1994). N-folded symmetries by complex moments in gabor space and their application to unsupervised texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*(01), 80–87.

Blumer, A., A. Ehrenfeucht, D. Hausler, und M. Warmuth (1986). Classifying learnable geometric concepts with vapnick-chervonenkis dimension. In *ACM Symposium on Theory of Computing*, pp. 273–282. Published as ACM Symposium on Theory of Computing, number 18.

Bovik, A. C., M. Clark, und W. S. Geisler (1990, January). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*(1), 55–73.

Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. Dover.

Brunelli, R. und T. Poggio (1993). Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*(10), 1042–1052.

Bunke, H. (1986). Hybrid approaches. *Computer and Systems Sciences: Proceedings of the NATO Advanced Research Workshop on Syntatctic and Strcutural Pattern Recognition 45*, 335–36.

Bunke, H. und B. Messmer (1995). Efficient attributed graph matching and its application to image analysis. *Lecture Notes in Computer Science 974*, 45–55.

Campbell, N. W. und B. T. Thomas (1997, July). Automatic selection of gabor filters for pixel classification. In *Sixth International Conference on Image Processing and its Applications*, pp. 761–765. IEE.

Campbell, N. W., B. T. Thomas, und T. Troscianko (1996). Segmentation of natural images using self-organising feature maps. In *British Machine Vision Conference*, pp. 223–232. British Machine Vision Association.

Cortes, C. und V. Vapnik (1995). Support-vector networks. *Machine Learning 20*, 273–297.

Daugman, J. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America 2*(7), 1160–1169.

Devijver, P. A. (1986). Segmentation of binary images using third-order markov image models. In *8th IEEE Intl. Conf. Pattern Recognition*, pp. 259–261.

Diaconis, P. und E. Efron (1983). Computer-intensive methods in statistics. *Scientific America 248*, 116–129.

Dietterich, T. G. (1997, June). Statistical tests for comparing supervised classification learning algorithms. submitted.

Dietterich, T. G. und E. B. Kong (1995). Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Department of Computer Science, Oregon State University Corvallis.

Dunn, D., W. Higgins, und J. Wakeley (1994). Texture segmentation using 2-d gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*(02), 130–149.

Eshera, M. A. und K. S. Fu (1986). An image understanding system using attributed symbolic representation and inexact graph-matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence 8*, 604–619.

Fiorio, C. und J. Gustedt (1994). A general method for segmentation using multiple criteria for decision. Technical report, TU Berlin, Germany.

Gabor, D. (1946). Theory of communication. *Journal of the Institution of Electrical Engineers 93*(3), 429–457.

Gaines, B. und P. Compton (1992). Induction of ripple down rules. In *Proc. 5th Australian Conference on Artifical Intelligence*, pp. 349–355.

Gaskill, J. (1978). *Linear Systems, Fourier Transforms, and Optics*. John Wiley and Sons.

Geibel, P. und F. Wysotzki (1996). Learning relational concepts with decision trees. In *Proceedings of the International Conference on Machine Learning*.

Geibel, P. und F. Wysotzki (1997). A logical framework for graph theoretical decision tree learning. In *Proceedings of the International Workshop on Inductive Logic Programming*, pp. 173–180.

Geman, S., E. Bienenstock, und R. Doursat (1992). Neural networks and the bias/variance dilemma. *Neural Computation 4*, 1–58.

Goodman, R. M., C. Higgins, J. Miller, und P. Smyth (1992). Rule–based networks for classification and probability estimation. *Neural Computation 4*, 781–804.

Greenspan, H. (1996). *Non-Parametric Texture Learning*, pp. 1–31. Oxford University Press.

Guerin-Dugue, A. und P. M. Palagi (1994). Texture segmentation using pyramidal gabor functions and self-organising feature maps. *Neural Processing Letters 1*(1), 9–25.

Haralick, R. und J. Lee (1988). Context dependent edge detection. In *9th IEEE International Conference on Pattern Recognition*, Volume 1, pp. 203–207.

Heidemann, G. und H. Ritter (1996). A neural 3-d object recognition architecture using optimized gabor filters. In *Proceedings of 13th International Conference on Pattern Recognition*, Vienna, Austria, pp. 70–74.

Hough, P. (1962, Dec.). Methods and means for recognizing complex patterns. US. Pat. 3069654. Washington.

Hubel, D. (1995). *Eye, brain, and vision*. Scientific American Library.

Hubel, D. und T. N. Wiesel (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology (London) 160*, 106–154.

Hufnagl, P., K. Voss, und A. Schlosser (1985). Ein algorithmus zur konstruktion von voronoidiagramm und delaunaygraph. *Bild und Ton 38*, 241–245.

Joo, H. (1991). *Automatic Morphology*. Ph. D. thesis, University of Washington.

Klette, R. und P. Zamperoni (1995). *Handbuch der Operatoren für die Bildbearbeitung*. Braunschweig: Vieweg Verlag Berlin.

Kohavi, R. (1996). *Wrappers for performance enhancement and oblivious decision graphs*. Ph. D. thesis, Stanford University, USA.

Kohonen, T. (1984). *Self-organisation and associative memory*. Springer Verlag Berlin.

Landraud, A. M. und S. Oh Yum (1995). Texture segmentation using local phase differences in gabor filtered images. *Lecture Notes in Computer Science 974*, 447–452.

Lavrač, N. und S. Dzeroski (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.

Lawrence, S., C. L. Giles, A. C. Tsoi, und A. D. Back (1997). Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks 8*(1), 98–113.

Lee, T. S. (1996). Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*(10), 959–971.

MacLennan, B. (1991). Gabor representations of spatiotemporal visual images. Technical Report UT-CS-91-144, Department of Computer Science, University of Tennessee. Mon, 6 Jan 97 17:00:24 GMT.

MacLennan, B. (1992). Field computation in the brain. Technical Report UT-CS-92-174, Department of Computer Science, University of Tennessee. Mon, 6 Jan 97 17:00:24 GMT.

Michie, D., D. H. Spiegelhalter, und C. C. Taylor (1994). *Machine Learning, Neural and Statistical Classification*. Series in Artificial Intelligence. Ellis Horwood.

Mingers, J. (1988). An empirical comparsion of selection measures for decision–tree induction. In *Machine Learning*, Volume 3, pp. 319–342.

Morik, K. (1995). *Einführung in die künstliche Intelligenz*, Chapter Maschinelles Lernen, pp. 243–297. Addison–Wesley.

Muggleton, S. und F. Cao (1990). Efficient induction of logic programs. In *Proceedings of the Workshop on Algorithmic Learning Theory*, Tokyo, pp. 368–381.

Parra, L. (1997, September). personal communication via email: lparra@sarnoff.com, David Sarnoff Research Center, Princeton.

Pavlidis, T. (1977). *Structural Pattern Recognition*. Springer Verlag Berlin.

Pichler, O., A. Teuner, und B. Hosticka (1996). A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition 29*(05), 733–742.

Pötzsch, M., T. Maurer, L. Wiskott, und C. Malsburg (1996). Reconstruction from graphs labeled with responses of gabor filters. In C. v.d. Malsburg, W. v. Seelen, J. C. Vorbrüggen, und B. Sendhoff (Eds.), *Proceedings of the ICANN 1996*, Springer Verlag, Berlin, Heidelberg, New York, Bochum, pp. 845–850.

Pratt, W. (1991). *Digital Image Processing* (2 ed.). Wiley.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*(1), 81–106.

Quinlan, J. R. (1990, August). Learning logical definitions from relations. *Machine Learning 5*(3), 239–266.

Rojas, R. (1993). *Theorie der neuronalen Netze*. Springer Verlag.

Roth, G. und M. D. Levine (1993). Extracting geometric primitives. *Computer Vision, Graphics, and Image Processing. Image Understanding 58*(1), 1–22.

Scheffer, T. (1996). Algebraic foundation and improved methods of induction of ripple down rules. In *Proccedings of the Pacific Rim Workshop on Knowledge Acquisition*, pp. 279–292.

Scheffer, T. und R. Herbrich (1997). Unbiased assessment of learning algorithms. In *Proccedings of the International Conference on AI*, pp. 798–803.

Scheffer, T., R. Herbrich, und F. Wysotzki (1997a). Efficient θ-subsumption based on graph algorithms. In S. Muggleton (Ed.), *Inductive Logic Programming: 6th International Workshop, ILP-96, Selected Papers*, LNAI 1314, pp. 212–228. Springer Verlag Berlin.

Scheffer, T., R. Herbrich, und F. Wysotzki (1997b). Neue methoden des maschinellen lernens (dfg-projekt), zwischenbericht.

Schyns, P. G. und H. H. Bulthoff (1993). Conditions for viewpoint dependent face recognition. Technical Report AI memo 1432, MIT AI Lab.

Seni, G., R. K. Srihari, und N. Nasrabadi (1996, July). Large vocabulary recognition of online handwritten cursive words. *Pattern Analysis and Machine Intelligence 18*(7), 757–762.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal 27*(7), 379–423.

Stockman, G. und A. Agrawala (1977). Equivalence of hough transform to template matching. *Communinications of the ACM 20*, 820–822.

Tan, T. (1995). Texture edge detection by modelling visual cortical channels. *Pattern Recognition 28*(09), 1283–1298.

Tönnies, K. und H. Lemke (1994). *3D – Computergrafische Darstellungen*. München Wien: Oldenbourg Verlag.

Unger, S. und F. Wysotzki (1981). *Lernfähige Klassifizierungssysteme*. Akademie Verlag Berlin.

Vistnes, R. (1988). Texture models and image measures for segmentation. In *Image Understanding Workshop (Cambridge MA, April 6-8, 1988)*, San Mateo, CA, pp. 1005–1015. Defense Advanced Research Projects Agency: Morgan Kaufmann.

Watanabe, L. und L. Rendell (1991). Learning structural decision trees from examples. In *Proccedings of the International Joint Conference on Artificial Intelligence*, pp. 770–776.

Weldon, T., W. Higgins, und D. Dunn (1996). Efficient gabor filter design for texture segmentation. *Pattern Recognition 29*(12), 2005–2016.

Weldon, T. P. und W. E. Higgins (1997). Algorithm for designing multiple gabor filters for segmenting multi-textured images. unpublished.

Wiskott, L. und C. von der Malsburg (1995). Face recognition by dynamic link matching. In *International Conference on Artificial Neural Networks*, Paris, pp. 347–352.