Inhaltsbasierter Zugriff auf große Bilddatenbanken

Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen

Prof. Dr.-Ing. H. Ney

vorgelegt von:

Cand. Inform. Thomas Theiner
Matrikelnummer 178 609

Gutachter:

Prof. Dr.-Ing. H. Ney
Prof. Dr. M. Jarke

Betreuer:

Dipl. Inform. J. Dahmen

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, den 09.02.2000

Thomas Theiner

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Danksagung

Eine solche Ausarbeitung ist nie ohne die tatkräftige Mithilfe vieler Personen möglich, erst recht nicht in einem so umfangreichen Projekt wie *IRMA*. Ich möchte mich also ganz besonders bei Prof. Hermann Ney und meinem Betreuer, Dipl.-Inform. Jörg Dahmen bedanken. Weiterhin gilt mein Dank Dipl.-Ing. Jörg Bredno. Er hat das Datenbanksystem RETRIEVAL, das als Basis für meine Anwendung IRMA dient, in kürzester Zeit fertiggestellt. Ein weiterer besonderer Dank gilt Daniel Keysers, der die Grundimplementierung des Tangentenverfahrens vorgenommen hat und bei Fragen jederzeit zur Verfügung stand. Schließlich gilt mein Dank einigen Mitarbeitern der Klinik für Radiologische Diagnostik und des Instituts für Medizinische Informatik an der RWTH Aachen, unter anderem Prof. Berthold Wein, Dr. Thomas Lehmann und Dipl.-Inform. Frank Vogelsang.

Kapitel 1

Einleitung

Die ständige Zunahme multimedialer Daten in den verschiedensten Bereichen, wie zum Beispiel in der Medizin und der Unterhaltungsbranche oder auch in lernunterstützender Software (z.B. multimediale Atlanten), stellt immer höhere Ansprüche an einen gezielten und effizienten Zugriff auf diese großen Datenbanken, in denen Bilder einen wesentlichen Bestandteil ausmachen. Auch in heutigen Systemen geschieht der Zugriff auf solch umfangreiche Datenbestände in der Regel via SQL. Dies setzt aber eine textuelle Indizierung der Bilder voraus, die weder eindeutig noch vollständig und darüber hinaus sehr zeitaufwendig ist.

Die vollständige Beschreibung eines Bildes scheitert dabei an seiner hohen Informationsdichte, zumal oft nur wenige Schlüsselwörter zur Indizierung herangezogen werden können. Dies führt nicht nur zu einem Informationsverlust, darüber hinaus beschreiben unterschiedliche Beobachter dasselbe Bild oft mit völlig anderen Schlüsselwörtern, abhängig vom Erfahrungshorizont und ihrer jeweiligen Intuition. So würde beispielsweise ein Radiologe zur Beschreibung eines Abdomen-Röntgenbildes sicher andere Schlüsselwörter heranziehen als ein Elektrotechniker oder ein Zeitungsreporter. Dieser Sachverhalt wird zusätzlich dadurch verschärft, dass eine nachträgliche Indizierung mit erweiterten Attributen oftmals gar nicht oder nur unzureichend möglich ist.

Das Problem der aufwendigen und nicht eindeutigen Indizierung wird in der Praxis häufig durch die manuelle Suche in sogenannten *Thumbnails* umgangen. Dabei wird dem Benutzer die Möglichkeit gegeben, verkleinerte Abbilder der Originalbilder zu durchsuchen und so die gesuchten Bilder ausfindig zu machen. Allerdings ist diese Methode für große Bilddatenbanken offensichtlich ungeeignet.

In der aktuellen Forschung werden daher zahlreiche Versuche unternommen, Verfahren zur Abfrage großer Bilddatenbanken zu entwickeln, die sich unter dem Oberbegriff *Content-Based Image Retrieval* (CBIR) oder kurz *Image Retrieval* (IR) zusammenfassen lassen (siehe Abbildung 1.1).

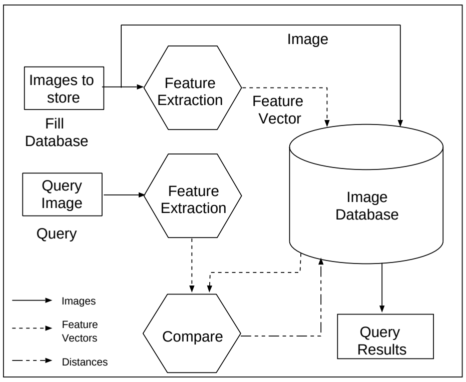

Abbildung 1.1: Aufbau eines IR-Systems

1.1 Bildzugriff durch Bildanalyse

IR-Systeme beruhen in der Regel auf dem folgenden Ansatz: Bei der Eingabe eines Bildes in das Datenbanksystem werden aus diesem bestimmte Merkmale (Features) extrahiert, die den Merkmalsvektor bilden. Da das Bild selber im allgemeinen sehr hochdimensional ist (z.B. 2000×2000 Pixel $= 2 \cdot 10^6$ Merkmale für eine gewöhnliche Thorax-Radiographie), ist das Ziel häufig die vorherige Merkmalsreduktion, d.h. das Extrahieren weniger, diskriminativer Features. Diese werden nun zusätzlich zum eigentlichen Bild in der Datenbank gespeichert. Anfragen an das Datenbanksystem geschehen fast immer über das 'Query by Visual Example'-Paradigma (QBE), nach dem der Benutzer dem System ein Beispielbild vorgibt. Dessen Aufgabe ist es dann, alle Bilder zurückzuliefern, die dem Anfragebild visuell 'ähnlich' sind. Zur Bestimmung der 'Ähnlichkeit' zweier Bilder werden dabei Merkmale wie Farbe, Textur oder Form herangezogen. Der entstehende Merkmalsvektor des Anfragebildes wird dann mit den Merkmalsvektoren der anderen Bilder in der Datenbank unter Verwendung geeigneter Distanzmaße verglichen. Unter *Query-Refinement* [Schä1997] versteht man dabei die sukzessive Verfeinerung einer QBE-Anfrage, in dem beispielsweise mit einem zurückgelieferten Bild eine weitere QBE-Anfrage gestartet wird. Mit *Relevance Feedback* bezeichnet man die Möglichkeit, zurückgelieferte Bilder (manuell) in erwünschte und unerwünschte zu klassifizieren und anhand dieser Information ein Query-Refinement vorzunehmen.

1.2 Gängige Features für den inhaltsbasierten Bildzugriff

Die Definition der Ähnlichkeit zweier Bilder geschieht in der Regel über eines oder mehrere der folgenden Features:

- Farbe
- Textur
- Form
- Objekte im Bild
- Räumliche Anordnung der Bildobjekte

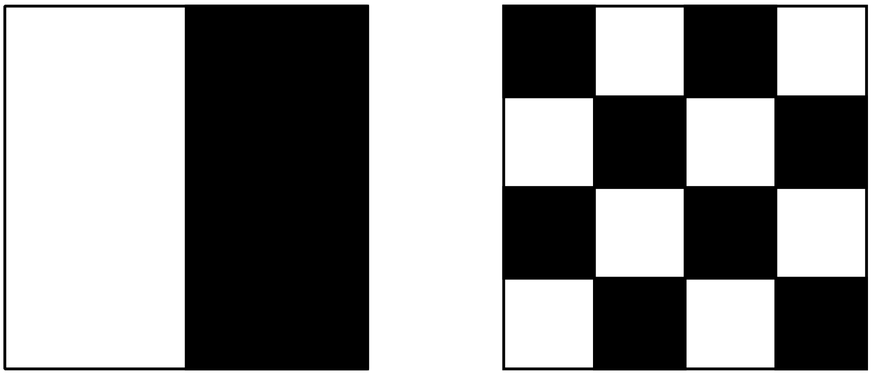

Abbildung 1.2: Schwäche des einfachen Histogramm-Vergleichs

1.2.1 Farbe

Beim Image Retrieval über Farbinformationen unterscheidet man zwischen globalen und lokalen Verfahren. Globale Verfahren treffen Aussagen über das gesamte Bild, während lokale Verfahren Farbinformationen bestimmter Regionen im Bild ermitteln, wobei unter Umständen auch deren räumliche Anordnung berücksichtigt wird. Einige Systeme verwenden das komplette Histogramm (siehe Abschnitt 3.4) als Merkmalsvektor, andere beschränken sich auf einige wenige dominante Farben oder gar auf die Durchschnittsfarbe. Diese Einschränkung wird dabei durch den hohen Aufwand von Histogrammvergleichen motiviert. Ein qualitativ hochwertiges Foto enthält heute nur selten weniger als 2^{24} Farben, was einen sehr hochdimensionalen Merkmalsvektor zur Folge hat. Distanzmaße zum Vergleich von Histogrammen sind etwa der euklidische Abstand oder die *Histogram Intersection* Technik (s. Kapitel 2 [Smi1996b, Swa1991]). Eine offensichtliche Schwäche des globalen Histogramms wird in Abbildung 1.2 gezeigt. Die beiden dort dargestellten einfachen Strukturen ergeben aufgrund ihrer identischen Farbverteilung eine exakte Übereinstimmung im Histogramm-Vergleich.

1.2.2 Textur

Auch Texturinformationen aus Bildinhalten können zur Generierung von Merkmalen herangezogen werden. Ausgangspunkt dieses Ansatzes ist dabei HARALICK's Grundlagenwerk [Har1973], in dem zahlreiche Texturdeskriptoren, aufbauend auf sogenannten Cooccurrence-Matrizen, vorgestellt werden. Generierte Merkmale sind dabei beispielsweise die Körnigkeit der Textur, die Hauptkantenrichtung oder die Wavelet-Filterung. Über diese Features ist dann erneut der Vergleich von Bildern oder Bildregionen möglich.

Da Texturen oftmals über statistische Verfahren beschrieben werden, kommt als Distanzmaß meist die Mahalanobis-Distanz [Dud1973] zum Einsatz. Diese berücksichtigt die Kovarianzmatrix einer Verteilung zur Distanzbestimmung. Entspricht die Kovarianzmatrix der Einheitsmatrix, so ist die Mahalanobis-Distanz gleich der euklidischen.

1.2.3 Form

Eine weitere Variante der Anfrage stellt das Skizzieren eines Umrisses dar. Das IR-System versucht dann solche Bilder zu finden, die ein Objekt der skizzierten Form enthalten. Die dabei verwendeten Verfahren sollten im Hinblick auf ein gutes Retrieval-Resultat invariant gegenüber affinen Transformationen (z.B. Translation, Rotation, Skalierung) sein. Diese Invarianz kann beispielsweise über die FOURIER-MELLIN-Transformation [Che1994] erzielt werden. Dabei sollte beachtet werden, dass in manchen Fällen eine komplette Invarianz durchaus unerwünscht sein kann. So ist zum Beispiel in der Ziffernerkennung die Rotationsinvarianz auf eine Abweichung von wenigen Grad einzuschränken, da es sonst zu vermehrten Fehlklassifikationen der Ziffern 6 und 9 kommen würde. Erwähnenswert ist an dieser Stelle noch, dass in den meisten aktuellen Systemen zur Segmentierung der Bilder in formrelevante Bereiche noch manuelle oder halbautomatische Verfahren eingesetzt werden.

1.2.4 Objekte im Bild

Teilweise hat der Benutzer die Möglichkeit, Bilder bei ihrer Eingabe in das Datenbanksystem manuell zu segmentieren, indem er beispielsweise mit der Maus den Umriss wichtiger Objekte nachzeichnet. Zu diesen Objekten werden dann gewisse Features wie Kreisförmigkeit, Schwerpunkt oder Hauptachsenrichtung bestimmt. Gibt der Benutzer dann als Query-Objekt ebenfalls eine Form vor, so werden aus dieser die erwähnten Features extrahiert und mit denen in der Datenbank verglichen. Zahlreiche objektbasierte Retrievalverfahren setzen allerdings die Segmentierung der Objekte als gegeben voraus. Gerade diese stellt aber – bis auf einige Spezialfälle – ein immer noch ungelöstes Problem dar, insbesondere in der medizinischen Bildverarbeitung. Eine mögliche Alternative stellen hier Verfahren zur automatischen Objekterkennung dar [Dah2000a]. Im weiteren Verlauf dieser Diplomarbeit wird zur Grobklassifikation von Röntgenbildern von nur einem Objekt pro Bild ausgegangen.

1.3 Evaluierungsmaße

In den folgenden Abschnitten werden Maße vorgestellt, die zur Evaluierung von (Image) Retrieval Systemen herangezogen werden können.

1.3.1 Precision und Recall

Die Effizienz von IR-Systemen wird üblicherweise über die zwei Größen *Precision P* und *Recall R* definiert [Car1997]:

$$P = \frac{rp}{rp + fp}, \qquad R = \frac{rp}{rp + fn} \tag{1.1}$$

mit $rp =$ 'richtig positiv', $fp =$ 'falsch positiv' und $fn =$ 'falsch negativ'. Als *positiv* werden dabei die durch das System auf eine Anfrage zurückgelieferten Bilder bezeichnet, als *negativ* entsprechend die nicht zurückgelieferten Bilder. Die Precision gibt an, wieviel Prozent der zurückgelieferten

Bilder tatsächlich korrekt sind. Im Gegensatz dazu zeigt der Recall-Wert an, wieviel Prozent der Bilder in der Datenbank, die die Anfrage erfüllen, auch zurückgeliefert wurden.

Beispiel 1.1 *Sucht man in einer Datenbank mit 32 Bildern, die 5 Sonnenuntergangsbilder enthält, nach eben diesen und werden dabei 16 Bilder zurückgeliefert, die 4 Sonnenuntergangsbilder enthalten, so entspricht dies einer Precision von 25% (4 von 16) und einem Recall von 80% (4 von 5).*

Für die Precision ist es dabei unbedeutend, ob diese 4 Bilder in der Bewertungsreihenfolge die Plätze 1-4 oder 13-16 belegen. Darüber hinaus fließt die Größe der Datenbank nicht in die Effizienz ein, obwohl es offensichtlich ein erheblicher Unterschied ist, ob ein IR-System aus einer Gesamtmenge von 32 oder 1000 Bildern eben diese 16 Bilder extrahiert.

1.3.2 Sensitivität und Spezifität

Aus diesem Grund bietet sich als Bewertungskriterium für die Effizienz eines IR-Systems das aus der Medizin bekannte statistische Bewertungskonzept von *Sensitivität* SE und *Spezifität* SP an, welches dort zum Vergleich eines neuen Diagnoseverfahrens mit einem bereits etablierten Verfahren (dem sogenannten *Goldenen Standard*) herangezogen wird:

$$SE = \frac{rp}{rp + fn}, \qquad SP = \frac{rn}{rn + fp} \tag{1.2}$$

Die *Sensitivität* ergibt sich als das Verhältnis der richtig positiven zu der Anzahl der kranken Patienten, die *Spezifität* als das Verhältnis von richtig negativen zu der Anzahl der gesunden Patienten. 'Richtig positiv' sind dabei Patienten, die durch das Verfahren als krank eingestuft wurden und auch tatsächlich krank sind, 'richtig negativ' sind gesunde Patienten, die durch das Diagnose-Verfahren auch als gesund eingestuft wurden. Analog verhält es sich mit falsch positiven / negativen Patienten. Dabei ergibt sich die Anzahl der kranken Patienten als Summe der 'richtig positiven' und 'falsch negativen', die Anzahl der gesunden Patienten als Summe von 'richtig negativen' und 'falsch positiven' [Dah1998a].

Bezogen auf IR definieren sich diese Größen in völliger Analogie zum medizinischen Fall, indem lediglich die Begriffe *krank* bzw. *positiv* durch 'erwünschte Bilder' und *gesund* bzw. *negativ* durch 'unerwünschte Bilder' ersetzt werden. Unser Beispiel 1.1 erreicht also bei einer Datenbankgröße von 32 Bildern eine Sensitivität von $4/(4+1) = 80\%$ bei einer Spezifität von lediglich $15/(15+12) = 55{,}6\%$. Ein identisches Ergebnis aus einer Menge von 1000 Bildern erreicht dahingegen eine Spezifität von $(1000-17)/(1000-5) = 98{,}8\%$ bei nach wie vor 80% Sensitivität. Man beachte, dass die Größen Recall und Sensitivität dabei identisch sind, die Spezifität aber bezüglich der Datenbankgröße deutlich aussagekräftiger als die Precision ist, die im medizinischen Bereich auch als *positiver Vorhersagewert* bekannt ist.

1.4 Zur vorliegenden Diplomarbeit

Im ersten Teil dieser Diplomarbeit wurde ein Image Retrieval System implementiert, welches auf dem QBE-Paradigma basiert und grundlegende Features wie Farbe und Textur zum Retrievalvorgang heranzieht. Darüber hinaus wurde ein erweitertes Nearest-Neighbour-basiertes Verfahren implementiert, das anhand unterschiedlicher Distanzmaße eine Grobklassifikation von Röntgenbildern vornimmt. An dieser Stelle sei auf die enge Verwandtschaft von Klassifikation und Retrieval hingewiesen. Während bei ersterer ein Objekt einer Klasse zugeordnet wird, so bestimmt beim letzteren das Anfrageobjekt eine solche und es werden alle Objekte gesucht, die dieser Klasse angehören. In diesem Sinne kann IR auch als Zwei-Klassen-Problem angesehen werden. Ein zurückgeliefertes Bild erfüllt entweder die Anfrage oder es erfüllt sie nicht.

In Kapitel 2 wird dazu zunächst auf den Stand der Technik im Bereich des inhaltsbasierten Bildzugriffs eingegangen. Dabei wird veranschaulicht, dass sich die dort vorgestellten Verfahren nur schlecht auf medizinischen Daten einsetzen lassen. Damit wird zugleich das im weiteren Verlauf dieser Diplomarbeit implementierte Nearest-Neighbour-basierte Verfahren motiviert, das insbesondere auch für medizinische Daten geeignet ist.

Kapitel 3 beschäftigt sich mit dem Projekt IRMA (Image Retrieval in Medical Applications) und der gleichnamigen Anwendung, die im Rahmen dieser Diplomarbeit implementiert wurde. Diese basiert auf dem Datenbanksystem RETRIEVAL, welches ebenfalls vorgestellt wird.

Da diese Arbeit die erste einer ganzen Reihe von Untersuchungen darstellen soll, wurde zunächst vor allem Wert auf die Zusammenstellung eines sinnvollen medizinischen Corpus gelegt, auf dem die erstellten Verfahren getestet werden sollen. In Kapitel 4 wird diese Datensammlung anhand von Beispielen vorgestellt. Dann wird kurz der von der Klinik für Radiologische Diagnostik am Klinikum Aachen eingeführte achtstellige Regionencode erläutert, der im vorliegenden Datenmaterial zur Indizierung medizinischer Bilddaten in die Bilddateinamen aufgenommen wurde.

Im Rahmen dieser Diplomarbeit, die eine Teilaufgabe des IRMA-Projektes behandelt, stand zunächst die Aufgabe an, dieses Datenmaterial automatisch in die 6 Grobklassen Abdomen, Extremitäten, Mamma, Schädel, Thorax und Wirbel zu klassifizieren. Dazu wurde das bereits angesprochene Nearest-Neighbour-basierte Verfahren implementiert, welches die Klassifikation unter Zuhilfenahme unterschiedlicher Distanzmaße vornimmt. So wurde zur Kompensierung globaler Bildtransformationen die in [Sim1993] vorgestellte *Tangentendistanz* verwendet. Lokale Invarianz wurde durch das im Rahmen dieser Diplomarbeit eingeführte *Image Distortion Modell* erreicht. Diese Verfahren werden in Kapitel 5 vorgestellt.

Kapitel 6 widmet sich ganz der Darstellung der erreichten Ergebnisse bezogen auf die unterschiedlichen Modelle, die zur Klassifikation herangezogen wurden. Dort wird nochmals verdeutlicht, wie die in Kapitel 5 bereits vorgestellten Verfahren motiviert wurden.

Ein anderes Verfahren, das im Rahmen des IRMA-Projektes vom Institut für medizinische Informatik am Klinikum Aachen implementiert wurde, verwendet die in [Aks1998] und [Har1973] vorgestellten Cooccurrence-Matrizen als Features und bestimmt dann die Ähnlichkeit zweier Bilder anhand eines sogenannten synergetischen Klassifikators [Hak1991]. In Kapitel 7 wird diese Vorgehensweise mit einem im Rahmen dieser Diplomarbeit implementierten Verfahren verglichen, das im wesentlichen den synergetischen Klassifikator durch einen Nearest-Neighbour-Klassifikator ersetzt und sich zumindest in der vorliegenden Anwendung als effektiver erweist.

Dieses optimierte Cooccurrence-Matrix-Verfahren erreichte dabei eine Fehlerrate von $29,4\%$. Ein einfacher Nearest-Neighbour-Vergleich führte zu einer Fehlerrate von $18,1\%$, die mit Hilfe der im weiteren Verlauf dieser Diplomarbeit vorgestellten Verfahren auf $8,6\%$ verbessert werden konnte.

Im Anhang werden die entwickelten Programme und Tools mit ihren jeweiligen Parametern sowie Input/Output-Formaten vorgestellt, um nachfolgenden Mitarbeitern das Fortsetzen dieser Arbeit zu erleichtern.

An dieser Stelle sei noch darauf hingewiesen, dass die Ergebnisse dieser Arbeit auch in [Dah2000d] veröffentlicht wurden.

Kapitel 2

Stand der Technik

Im folgenden werden beispielhaft einige IR-Systeme vorgestellt, die zum Teil bereits vor einigen Jahren, teilweise aber auch nahezu parallel zu dieser Diplomarbeit entwickelt und implementiert wurden. Es werden dabei Verfahren gezeigt, die die Attribute Farbe, Form und auch Textur berücksichtigen.

2.1 Das QBIC-System (IBM)

IBM's 'Query by Image Content' (QBIC) System [Fli1995] war die erste auch kommerziell verfügbare IR-Applikation. QBIC ist, wie zahlreiche andere Systeme auch, in der Lage, neben Bildern auch Videosequenzen zu indizieren. Darauf soll hier aber nicht weiter eingegangen werden, es sei nur erwähnt, dass dieses Problem in der Regel mittels der Bestimmung repräsentativer Einzelbilder weitestgehend auf ein Image Retrieval Problem zurückgeführt wird. Die in Videosequenzen meist enthaltenen Audiodaten bleiben unter QBIC unberücksichtigt.

QBIC erlaubt sowohl Anfragen über die Angabe von Schlüsselwörtern als auch über den QBE-Ansatz. Neben kompletten Bildern kann der Benutzer aber auch Farbverteilungen, Texturen oder Formen als Beispiel vorgeben. Für den Farbvergleich nutzt das System Histogramme, die über eine Farbähnlichkeitsmatrix miteinander verglichen werden. Diese Matrix wird über die Abstände zweier Farben im gewählten Farbraum bestimmt. Dabei können die Histogramme sowohl für das ganze Bild als auch für bestimmte Regionen ermittelt werden, wobei der Benutzer die *Region of Interest* (ROI) manuell (also mit der Maus) definieren muss. In neueren Implementierungen von QBIC wird er allerdings bereits von halbautomatischen Segmentierungsverfahren wie dem *Flood-Fill* Algorithmus bzw. *aktiven Konturmodellen* unterstützt [Ash1995].

Bei Angabe einer Textur extrahiert QBIC die Features *Coarseness*, *Contrast* und *Direction* (siehe auch [Har1973]). Diese Merkmale werden nur aus zuvor manuell segmentierten Regionen extrahiert. Nicht manuell vorsegmentierte Bilder werden also nicht berücksichtigt. Zusätzlich zu diesen Texturmerkmalen extrahiert QBIC noch andere Merkmale aus vorsegmentierten Objekten, nämlich die Kreisförmigkeit (Verhältnis Umfang/Fläche), die Hauptachsenrichtung und die Exzentrizität.

Durch die Verwendung normierter Zentralmomente wird Invarianz gegenüber affinen Transformationen erreicht. Neben diesen Methoden, die eine Vorsegmentierung voraussetzen, hat der Benutzer weiterhin die Möglichkeit, vollautomatisch über eine Skizze, die zunächst in ein Gradientenbild umgerechnet wird, auf die Datenbank zuzugreifen. Zur Reduktion der zum Retrieval benötigten Vergleiche wird als Datenstruktur der R^*-Baum [Bec1990] eingesetzt, wobei als Abstandsmetrik die euklidische Distanz zum Einsatz kommt.

Die für QBIC genutzten Algorithmen werden im wesentlichen auch in IBM's kommerziellen *Ultimedia Manager* verwendet. Weitere Informationen zu QBIC finden sich im Internet unter [URL1]. Der Ultimedia Manager wird kurz unter [URL2] vorgestellt.

2.2 VisualSEEk (Columbia University, New York)

Im Gegensatz zu QBIC, das die räumliche Anordnung von Features nur für zuvor manuell extrahierte Objekte unterstützt, detektiert *VisualSEEk* automatisch solche Regionen und ist damit am ehesten mit dem weiter unten vorgestellten Blobworld-Projekt der University of California at Berkeley zu vergleichen. Grundlegend für VisualSEEk sind binäre Histogramme, die sogenannten *Colorsets*. Nur solche Farben, die im Bild mindestens mit einer gewissen Wahrscheinlichkeit auftreten, werden im Colorset mit einer 1 codiert. Der Vergleich zweier Colorsets ist nun wesentlich effizienter möglich als der zweier vollständiger Histogramme [Smi1996a].

Die automatische Extraktion von ROIs basiert auf einer *Colorset Backprojection* genannten Methode [Smi1996b]. Bei diesem Verfahren werden im wesentlichen alle für ein gegebenes Bild möglichen Colorsets betrachtet, die eine ROI beschreiben könnten. Um die Anzahl der in Frage kommenden Regionen zu reduzieren (beispielsweise wären das bei nur 32 signifikanten Farben bereits 2^{32} = ca. 4 Milliarden mögliche Kombinationen), wird ein geeignetes *Pruning* vorgenommen, welches sich letztendlich aus gewissen Anforderungen an eine ROI ergibt, etwa deren Mindestgröße. Die Beschreibung der übrigbleibenden ROIs geschieht dann über *Minimum Bounded Rectangles* (MBR). Es sei darauf hingewiesen, dass diese automatische Regionenextraktion nicht mit einer Segmentierung des Bildes verwechselt werden darf. Während bei einer Segmentierung jedes Pixel genau einer Region zugeordnet wird, so kann ein Pixel bei der vorliegenden Backprojection-Methode durchaus mehreren Regionen zugeordnet werden oder sogar völlig unberücksichtigt bleiben.

VisualSEEk erlaubt es dem Benutzer dann, eine QBE-Anfrage zu stellen, die aus bis zu 3 ROIs und deren räumlicher Anordnung besteht. Ebenso wie QBIC nutzt auch VisualSEEk R^*-Bäume [Bec1990] zur Reduzierung der zum Retrieval benötigten Vergleiche.

Weitere und aktuellere Informationen zu VisualSEEk sind im Internet unter [URL3] zu finden.

2.3 Blobworld (University of California at Berkeley)

Eine weitere IR-Applikation stellt das Blobworld-Projekt der University of California at Berkeley dar [Car1997]. Dieses System zielt darauf ab, IR auf *natürlichen* Bildern, also beispielsweise Landschafts- und Tieraufnahmen, zu ermöglichen. Diese Art von Bildern erweist sich oftmals, von wenigen Ausnahmen wie zum Beispiel Sonnenuntergängen abgesehen, als schwierig zu handhaben, da sie nur sehr schwer segmentiert werden können. Blobworld arbeitet dabei lediglich mit den beiden Features Farbe und Textur, wobei auch und gerade räumliche Zusammenhänge berücksichtigt werden.

Ähnlich zu VisualSEEk geschieht auch in Blobworld die Extraktion der ROIs automatisch, und zwar unter Verwendung des *Expectation Maximization* (EM) Algorithmus. Dazu werden zunächst je drei Farb- und Texturmerkmale für jeden Bildpunkt extrahiert, die Punkte in einem 6-dimensionalen Merkmalsraum bilden. Diese werden nun mittels des EM-Algorithmus zu Clustern zusammengefasst, wobei in der Regel weniger als 10 Cluster betrachtet werden. Jede der so extrahierten ROIs wird nun durch eine äquivalente Ellipse, also eine Ellipse mit gleichen ersten und zweiten Momenten, ersetzt. Diese Ellipsen, die grafisch um die zuvor extrahierten Textur- und Farbmerkmale ergänzt werden, bilden die sogenannten Blobs. Blobs repräsentieren also Bildregionen, die bezüglich ihrer Farb- und Textureigenschaften homogen sind. In neueren Implementierungen des Blobworld-Systems werden die segmentierten Regionen nicht länger durch Ellipsen, sondern durch wesentlich genauere Annäherungen der Region-Konturen beschrieben. Dies ändert jedoch nichts an der grundlegenden Arbeitsweise des Systems.

Der Benutzer hat nun direkten Zugriff auf die Blobs bzw. die Darstellung eines Bildes mittels dieser Blobs. Durch die Offenlegung dieser Interna des Systems erhoffen sich die Entwickler ein besseres Verständnis des Benutzers für dessen Arbeitsweise. Werden beispielsweise Bilder zurückgeliefert, die dem Query-Objekt visuell völlig unähnlich sind, so versteht der Benutzer vielleicht über den Vergleich der zugehörigen Blobdarstellungen, warum das System diese Bilder als ähnlich eingestuft hat.

Der Benutzer kann als Anfrage an das in Java implementierte System entweder ein komplettes Bild oder Blobs (auch aus verschiedenen Bildern) vorgeben. Der Vergleich der Blobs erlaubt nun Ähnlichkeitsaussagen über verschiedene Bilder, wobei als Distanzmaß die Mahalanobis-Distanz [Dud1973] herangezogen wird. Nähere Informationen und eine Online-Implementierung von Blobworld sind im Internet zu finden, und zwar unter [URL4]

2.4 IRIS (IBM und Universität Bremen)

Das IRIS-Projekt (*Image Retrieval for Information Systems*) wurde 1994 als Kooperation von IBM und der Universität Bremen gestartet [Als1996]. Es wird heute unter dem Namen *ImageMiner* weiterentwickelt. Dieses System ermöglicht eine automatische textuelle Beschreibung der Bilder. Dazu wird das Bild in homogene Regionen aufgeteilt, zu denen gewisse Features extrahiert werden.

Über eine Datenbank, die Regeln für die Erkennung von Bildinhalten unterschiedlicher Domänen enthält (z.B. 40 Regeln für Landschaftaufnahmen, 105 für technische Zeichnungen), können nun Bildinhalte erkannt und textuell beschrieben werden. Der Benutzer kann dann theoretisch auf Bilder zugreifen, ohne diese vorher manuell indizieren zu müssen. Damit ist IRIS/ImageMiner das bislang einzige IR-System, welches Retrieval auf semantischer Ebene erlaubt [Eak1996]. Im Internet findet man unter [URL5] eine Ausarbeitung über ImageMiner, eine Kurzbeschreibung des Programms und der grafischen Oberfläche sowie eine Online-Demonstration der Retrieval-Eigenschaften.

2.5 FIDS (Universität Washington, Seattle)

Das FIDS-System (*Flexible Image Database System*), das in [Berm1999] vorgestellt wird, präsentiert eine Möglichkeit, unterschiedliche Distanzmaße für komplexe Bilddatenbankanfragen flexibel zu kombinieren. Dem Benutzer werden für diesen Fall alle bereits implementierten Distanzmaße vorgelegt, aus denen er beliebige auswählen kann. Intern werden diese dann zu einer Datenbankanfrage kombiniert. Dadurch werden nicht nur Linearkombinationen von Distanzmaßen möglich (wie es in QBIC zum Beispiel über eine Anforderung der Art 'Farbe mit 60% gewichten, Textur mit 40%' durchführbar ist) sondern auch beliebig verschachtelte Kombinationen mit UND- bzw. ODER-Verknüpfungen.

Beispiel 2.1 *Eine Anfrage der Form 'Ich möchte eine möglichst gute Übereinstimmung in der Farbe und entweder in der Form oder Textur' würde über folgende Kombination dargestellt:*
$d(x, y) = \max[d_{farbe}(x, y), \min(d_{form}(x, y), d_{textur}(x, y))]$

Um eine vereinfachte Abschätzung der in ihrer Anzahl dadurch extrem zunehmenden Distanzvergleiche zu erhalten, wird die Dreiecksungleichung herangezogen:

$$d(x, y) \leq d(x, z) + d(z, y) \tag{2.1}$$

Dabei gilt unter der Voraussetzung, dass als Distanzen nur Metriken verwendet werden, dass die Distanz zwischen zwei Bildern einer Datenbank niemals geringer sein kann als die Differenz ihrer Distanzen zu einem beliebigen anderen Bild, d.h. es gilt

$$d(x, y) \geq |d(x, z) - d(z, y)| \tag{2.2}$$

für beliebige Bilder x, y und z (siehe [Berm1999]). Der Indizierungs-Algorithmus in FIDS liefert darauf basierend für jedes Bild der Datenbank eine untere Distanzgrenze zu dem Anfragebild. Diese Wertemenge kann nun folgendermaßen zur Reduktion der tatsächlichen Distanzvergleiche herangezogen werden:

- Bilder, die aufgrund dieser Werte eine zu große Distanz zu haben scheinen, um weiterhin ein potenzielles Anfrageergebnis zu sein, werden im weiteren Verlauf nicht mehr betrachtet

- Die Bilder werden nach diesen Werten sortiert. Experimente im Rahmen des FIDS-Projekt haben gezeigt, dass die in dieser Sortierung im vorderen Bereich auftauchenden Bilder höchstwahrscheinlich auch die besten Übereinstimmungen mit dem Anfragebild beim tatsächlichen Distanzvergleich ergeben

Eine weiterführende Erläuterung der Implementierung und Anwendung dieses Verfahrens ist neben [Berm1999] auch im Internet verfügbar, und zwar unter [URL6]

2.6 Fazit und eigene Arbeiten

FALOUTSOS zeigt in [Fal1994], dass die besten Retrievalresultate mit dem Kriterium Farbe erzielt werden. Dieses Kriterium greift allerdings auf medizinischen Bilddaten im allgemeinen nicht, wenn man von bereits vorbearbeiteten und eingefärbten Aufnahmen absieht, da es seine diskriminativen Eigenschaften verliert. Als ebenso unzureichend können die Kriterien 'Form', 'Objekte im Bild' bzw. deren räumliche Anordnung angesehen werden, da wegen immer noch unbefriedigender automatischer Segmentierung meist noch eine manuelle Vorsegmentierung notwendig ist. Dies ist wegen des zeitlichen Aufwandes und des einzubringenden Expertenwissens in diesem Bereich nicht praktikabel. Auch zur Gewinnung von Texturmerkmalen ist eine Segmentierung unabdingbar, da global für ein Bild extrahierte Texturmerkmale wenig aussagekräftig sind, wie im Kapitel 7 noch gezeigt wird.

In der Zwischenzeit wurden weitere Verfahren entwickelt, beispielsweise in [Schm1997], [Schi1996] oder [Rav1998]. Untersuchungen zu deren Anwendbarkeit und Effizienz auf medizinischen Bilddaten stehen allerdings noch aus.

Im weiteren Verlauf dieser Diplomarbeit werden nun Verfahren vorgestellt, die das gesamte Bild als Feature nehmen und die Kriterien Farbe (bzw. Grauton), Form, Texturen und Objekte im Bild explizit berücksichtigen, indem jeder Pixel des Bildes zur Merkmalsextraktion herangezogen wird. Im Gegensatz zu den in diesem Kapitel vorgestellten Verfahren leistet diese Arbeit einen ersten wichtigen Schritt hin zu einem Image Retrieval System für medizinische Bilddaten.

Kapitel 3

Das IRMA-Projekt

Im Zuge der immer mehr anwachsenden Bereiche der medizinischen Bildverarbeitung und allgemeiner Bildarchivierungssysteme (PACS = *Picture Archiving Systems*) steigt auch die Notwendigkeit, digitale Image Retrieval Verfahren zu untersuchen. Bis zum heutigen Tag ist es zwingend erforderlich, bei der Suche nach Bildern eines Krankenhaus-Archivs auf textuell indizierte Einträge einer Datenbank zurückzugreifen (Name des Patienten, Geburtsdatum, Diagnose etc.), auch wenn die Daten bereits im DICOM-Format [URL8] vorliegen. Wie schon im letzten Kapitel gesagt, beschäftigen sich zur Zeit viele Forschungsprojekte mit dem Problem des Image Retrieval. Allerdings zielt die Mehrzahl der aktuellen Systeme darauf ab, heterogene Bilddatenbanken zu durchsuchen, die zum Beispiel aus dem World Wide Web zusammengestellt wurden. Üblicherweise beschreiben die dabei extrahierten Features das gesamte Bild, weniger konkrete Bildbereiche oder gar Objekte. FALOUTSOS hat in [Fal1994] dargestellt, dass das Merkmal *Farbe* das effektivste für solche Systeme ist. Es ist leicht einzusehen, dass farbbasierte Merkmale in den meisten medizinischen Anwendungen nicht sehr vielversprechend sein können, da sie ihre diskriminativen Eigenschaften verlieren (z.B. Röntgenbilder). Desweiteren beinhalten diagnostisch relevante Datenbankanfragen die Suche nach Organen, ihre relative Anordnung und andere unterschiedliche Merkmale. Aus diesen Gründen können aktuelle Retrieval Systeme keine vollständige Erfüllung von Anfragen im medizinischen Kontext garantieren. Dazu wurde das IRMA-Projekt (*Image Retrieval in Medical Applications*) ins Leben gerufen, ein gemeinschaftliches Projekt dreier Institute der RWTH Aachen, namentlich des Lehrstuhls für Informatik VI, des Instituts für Medizinische Informatik und der Klinik für Radiologische Diagnostik am Klinikum der RWTH. Weiterhin ist auch Philips Medical Systems beteiligt. Im Rahmen dieses Projektes soll ein System für den alltäglichen klinischen Gebrauch entwickelt werden.

Die vorliegende Diplomarbeit erklärt im folgenden den Aufbau des IRMA-Systems und beschäftigt sich dann mit der Erstellung einer Rahmenanwendung mit dem Namen IRMA, die auf dem vom Institut für Medizinische Informatik implementierten Datenbanksystem RETRIEVAL basiert und die Konzepte des IRMA-Systems berücksichtigt. In den weiteren Kapiteln wird der erste wichtige Schritt der Bilddatenverarbeitung behandelt, nämlich die Bildklassifikation.

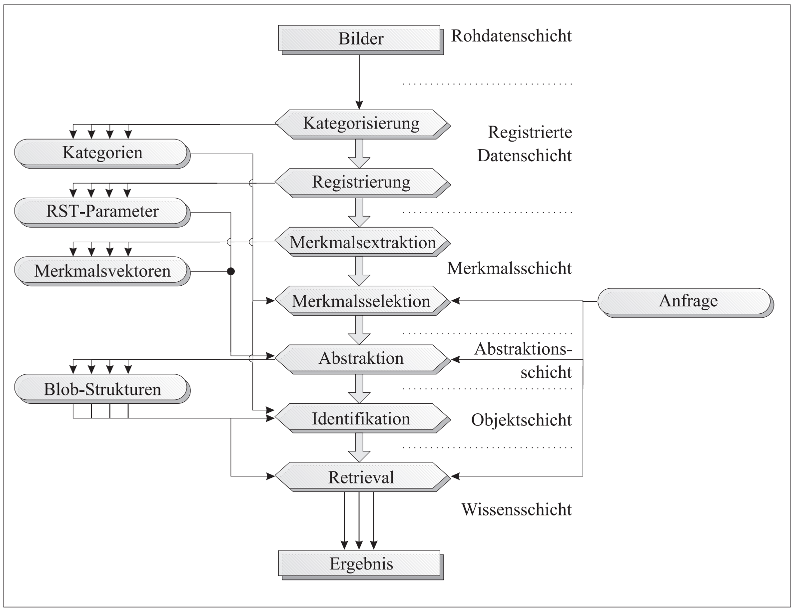

Abbildung 3.1: Aufbau des IRMA-Systems

3.1 Das IRMA-System

Das Konzept zu IRMA basiert auf der strikten Trennung und eindeutigen Formalisierung von sieben sequentiellen Schritten [Leh2000a] (siehe Abbildung 3.1):

- Kategorisierung (globale Bildbeschreibung)
- Registrierung (geometrische Ausrichtung, Kontrastabgleich)
- Merkmalsextraktion (lokale Bildbeschreibung)
- Merkmalsselektion (kontextabhängig)
- Abstraktion (hierarchische Multiskalenbeschreibung durch Blobs)
- Identifikation (Erkennung von Bildobjekten)
- Retrieval (auf abstrakter Blob-Ebene)

3.1.1 Kategorisierung und Grobklassen

Im Sinne einer intelligenten Verarbeitung der Bildinformation ist es hilfreich, wenn dem IRMA-System vor dem eigentlichen Retrieval bekannt ist, um welche Art von Bild es sich handelt. So ist z.B. ein Röntgenbild nach anderen Maßgaben zu verarbeiten als ein Ultraschall-Bild. Die Bilder werden dazu mit globalen Bildverarbeitungsalgorithmen oder durch die Auswertung von DICOM-Informationen in die drei Hauptklassen Modalität (physikalisch), Körperregion (anatomisch) und Aufnahmeorientierung (technisch) eingeteilt [Wen1994]. In einem ersten Schritt wurden sechs anatomische Regionen unterschieden: (1) Abdomen, (2) Extremitäten, (3) Mamma, (4) Schädel, (5) Thorax und (6) Wirbel, die im weiteren Verlauf dieser Diplomarbeit als *Grobklassen* referenziert werden. Sobald einmal die Grobklasse eines gegebenen Bildes bestimmt wurde, können nachfolgende Bearbeitungsschritte des IRMA-Systems diese Information nutzen, um problemspezifische Merkmale zu extrahieren, die für komplexere Anfragen benötigt werden. Sucht beispielsweise ein Arzt nach Aufnahmen eines Kniescheibenbruchs, so kommen natürlich nur solche Aufnahmen in Frage, die zur Region 'Extremitäten' gehören. Nähere Untersuchungen dieses Bildmaterials können danach anhand angepasster Algorithmen erfolgen, die schließlich die gewünschten Aufnahmen zurückliefern. Weil diese Klassifikation im allgemeinen nicht eindeutig sein muss (eine 'Abdomen'-Aufnahme könnte gleichzeitig auch mit 'Wirbel' bezeichnet sein), wird dieser Schritt innerhalb des IRMA-Systems 'Kategorisierung' genannt. Demnach kann jedes Bild mit mehreren verschiedenen Kategorien verknüpft sein. Eine Wahrscheinlichkeit für jede dieser Kategorien wird in der Datenbank gespeichert.

3.1.2 Registrierung

Innerhalb jeder Kategorie erfolgt die Berechnung der Registrierungsparameter bezüglich eines Musterbildes. Das Musterbild kann eine reale Aufnahme, aber auch synthetisch erzeugt worden sein. Die eigentliche Transformation des Bildes mit den berechneten Parametern findet allerdings erst auf einer späteren, abstrakten und damit informationsreduzierten Ebene statt, in der sie effizienter durchgeführt werden kann.

3.1.3 Merkmalsextraktion

Durch automatische Algorithmen der Bildverarbeitung werden Merkmalsvektoren berechnet und zusammen mit dem ursprünglichen Bild in der Datenbank abgelegt. Im Gegensatz zu den globalen Merkmalen, die zur Kategorisierung für jedes Bild zu einem einzigen Merkmalsvektor zusammengefasst werden, wird zur lokalen Bildbeschreibung für jedes Pixel ein individueller Merkmalsvektor berechnet.

3.1.4 Merkmalsselektion

Die Kategorisierung liefert Vorgaben für eine sinnvolle Parameterauswahl. So spielen z.B. farbdiskriminative Merkmalskomponenten beim Retrieval in Röntgenbildern keine Rolle und werden durch die Merkmalsselektion ausgeblendet. Weiterhin bestimmt der Kontext der jeweiligen Anfrage relevante Komponenten der Merkmalsvektoren.

3.1.5 Abstraktion

Auf Basis der selektierten Merkmalsvektoren werden Cluster gebildet und ähnlich dem bereits in Abschnitt 2.3 vorgestellten *Blobworld-System* zur abstrakten Beschreibung der Bilder verwendet. Der Detailreichtum, also die Anzahl und Größe der zu berücksichtigenden Blobs werden dazu in einem hierarchischen Multiskalenansatz organisiert. Dieses Modell basiert auf der Tatsache, dass ein großer Teil der Information in medizinischen Bildern durch die Lokalität der Strukturen bestimmt ist [Tag1997]. Das anschließende Retrieval erfolgt auf Ebene dieser Cluster (Blobs) sowie deren

Lage zueinander. Durch die vorangegangene Kategorisierung und Registrierung der Bilder ist ein erfolgreiches Retrieval möglich.

3.1.6 Identifikation

Für semantische Anfragen ist die Identifizierung von Objekten in Bildern sehr hilfreich. Im Gegensatz zur rein datenbasierten Cluster-Bildung im Abstraktionsschritt erfolgt diese inhaltliche Zuordnung von Blobs zu Organen oder anderen benennbaren Strukturen durch massiven Einsatz von a-priori Wissen. Zum jetzigen Zeitpunkt ist durch Kategorisierung und Registrierung bekannt, an welchen Positionen welche Objekte mit welchen Eigenschaften zu erwarten sind. Die Identifikation muss jedoch nicht vollständig erfolgen, um ein Retrieval durchführen zu können.

3.1.7 Retrieval

Die Datenbankanfragen erfolgen auf abstrakter Ebene, also durch Vergleich der aus den Merkmalsvektoren gebildeten Blobs. Zunächst muss dazu die Auflösungsebene der Blobs bestimmt werden, auf der die Anfrage ausgeführt werden soll. Die bereits erfolgte Registrierung der Bilder in eine Normallage erlaubt auch die Definition von ROIs (*Regions of Interest*), auf denen Anfragen zu bearbeiten sind. Über solche ROIs können zusätzlich unterschiedliche Blob-Ebenen innerhalb einer Anfrage modelliert werden, wenn Details bestimmter Objekte oder in bestimmten Bildbereichen von Interesse sind. Somit können semantische Anfragen sowohl für die diagnostische Routine als auch für die klinische Forschung behandelt werden.

Für nähere Informationen sei an dieser Stelle auf [Leh2000a] hingewiesen. Parallel zu dieser Diplomarbeit ausgeführte Arbeiten im Rahmen des IRMA-Projektes, die sich allesamt an dem soeben vorgestellten System orientieren, werden in [Bre2000a],[Bre2000b], [Koh2000a],[Koh2000b] und [Leh2000b] präsentiert.

3.2 Die Datenbank RETRIEVAL

Die Datenbank RETRIEVAL wurde im Rahmen des IRMA-Projektes gemeinsam mit dem Institut für Medizinische Informatik speziell für die Verwaltung medizinischer Bilddaten, darauf anzuwendender Algorithmen, den damit aus den Bildern extrahierten Features und weiteren Anfrageeigenschaften entwickelt und implementiert [Bre1999]. Grundlage bildete das im Internet frei erhältliche relationale Datenbank Management System *postgreSQL* [URL7]. Die Datenbank wurde so aufgebaut, dass Bilddaten und die Sourcen der in das System einzubindenden Algorithmen verteilt im lokalen Netz bzw. gar auf entfernten Rechnern (beispielsweise im Internet) liegen können.

Zu jedem Bild werden dabei in einer Tabelle die Eigenschaften *Name* und *Verzeichnis* und zusätzlich ein Verweis auf eine Tabelle mit weiteren Eigenschaften, wie z.B. Größe in Pixeln, Dimension und Farbtiefe abgelegt. Diese Zusatztabelle wird auch von den auf die Bilder anzuwendenden Algorithmen benötigt, da nicht zwangsläufig jeder Algorithmus mit jedem Bild umgehen kann. Desweiteren wird zu jedem Algorithmus ein Bezeichner, der Name des Executables, die Anzahl und Art der

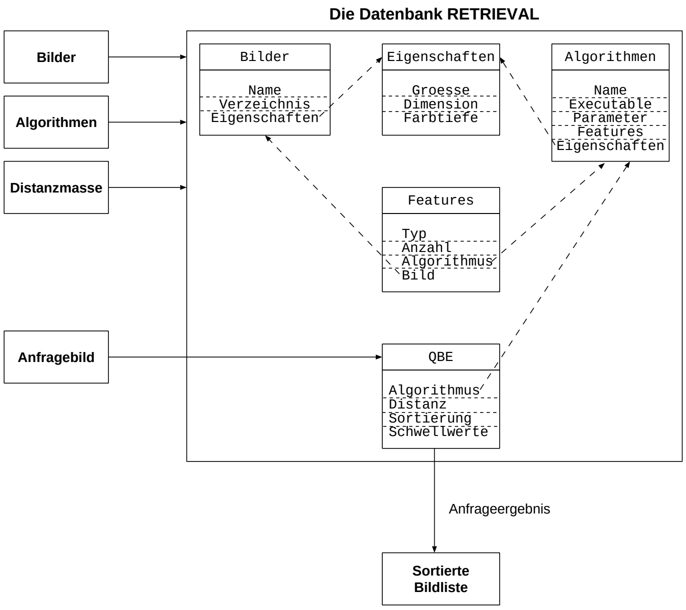

Abbildung 3.2: Aufbau der Datenbank RETRIEVAL

zu übergebenden Parameter sowie der extrahierten Features abgelegt. Nun können abhängig vom gewählten Algorithmus zu den dazu passenden Bildern Features vom Typ *int*, *float* oder *string* erzeugt und in der Datenbank abgelegt werden. Zum Anfragezeitpunkt werden gegebenenfalls vom Anfragebild (*Query by Visual Example*) die entsprechenden Features extrahiert und mittels eines anzugebenden Distanzmaßes verglichen. Das Ergebnis ist eine nach der Distanz wahlweise aufsteigend oder absteigend sortierte Bildliste. Abbildung 3.2 verdeutlicht grob schematisiert den Aufbau der Datenbank und den Ablauf einer Anfrage.

3.3 Die Anwendung IRMA

Die im Rahmen dieser Diplomarbeit implementierte Anwendung IRMA stellt nun ein grafisches Frontend zur Durchführung einer QBE-Anfrage an die Datenbank RETRIEVAL dar. Dazu ist es zunächst möglich, mittels eines einfachen Dateidialogs das Anfragebild auszuwählen, das in der aktuellen Implementierung im JPG-Format vorliegen muss (siehe Abbildung 3.3, links). Im folgenden wird dann das Bild auf dem Bildschirm dargestellt. Im unteren Bereich des Dialogs (siehe Abbildung 3.3, rechts) ist es dann möglich, einen der zu den Eigenschaften des Anfragebildes passenden Algorithmus auszuwählen und die QBE-Anfrage zu starten. Daraufhin werden gegebenenfalls aus dem Anfragebild die entsprechenden Features extrahiert und dann mittels des zu dem gewählten Algorithmus gehörenden Distanzmaßes mit allen anderen passenden Bildern verglichen. Die resultierende Bildliste wird dann wie in Abbildung 3.4 dargestellt ausgegeben. Die erste Spalte entspricht dabei der (internen) Datenbank-ID des Bildes, die zweite Spalte enthält die Distanz zum Anfragebild und in der dritten Spalte steht schließlich der vollständige Dateiname des Ergebnisbildes. Durch Anklicken einer dieser Ergebniszeilen und Aktivieren des Buttons *Zeigen* kann das jeweilige Ergebnisbild auch optisch mit dem Anfragebild verglichen werden.

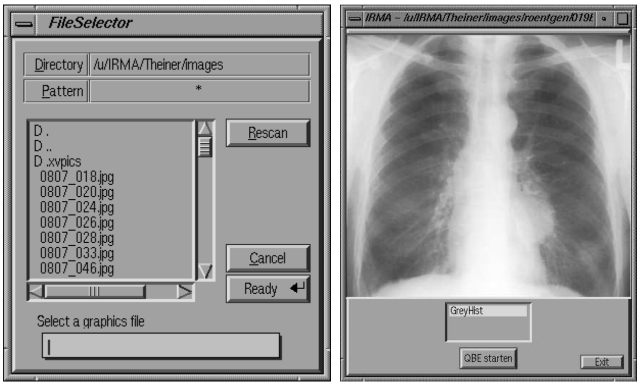

Abbildung 3.3: Auswahl des Anfragebildes

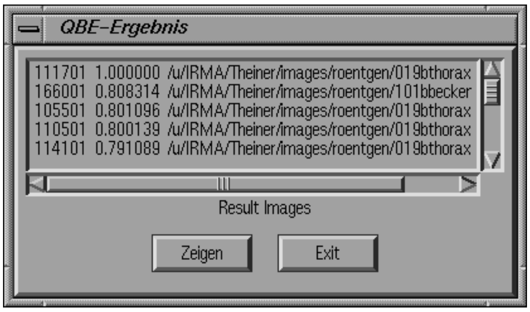

Abbildung 3.4: Anzeige der QBE-Ergebnisse

3.4 Ein Beispielalgorithmus: Histogram Intersection

Zum Testen der Anwendung IRMA und insbesondere der Schnittstellen zur Datenbank RETRIEVAL wurde ein einfaches Histogrammverfahren implementiert. Das Histogramm eines Bildes beschreibt dessen Farbverteilung. Jede Farbe im Bild korrespondiert mit einem Punkt im zugrundeliegenden Farbraum (der beispielsweise bei RGB-Bildern 3-dimensional ist). Jeder Pixel eines 24-Bit Bildes kann demnach eine von 2^{24} Farben annehmen. Eine solch hohe Auflösung wird im allgemeinen für die spezielle Aufgabe des Image Retrieval nicht benötigt. Typischerweise wird der Farbraum zur Reduktion der Farbanzahl in Blöcke quantisiert, die der Einfachheit halber eine Zweierpotenz als Kantenlänge besitzen. Ein Histogramm wird nun durch einen N-dimensionalen Vektor repräsentiert. N gibt dabei die Anzahl der Blöcke an. Ein Eintrag des Vektors ist dabei entweder die absolute Anzahl der Pixel in dem korrespondierenden Block, oder aber deren relative Häufigkeit.

Als Distanzmaß wurde die in [Swa1991] vorgestellte *Histogram Intersection* Technik HIT implementiert. Sie ist dort für zwei Bilder I und J bei N Histogrammfarben definiert als

$$d(I,J) = HIT(I,J) = \frac{\sum\limits_{n=1}^{N} \min(I_n, J_n)}{\sum\limits_{n=1}^{N} I_n} \tag{3.1}$$

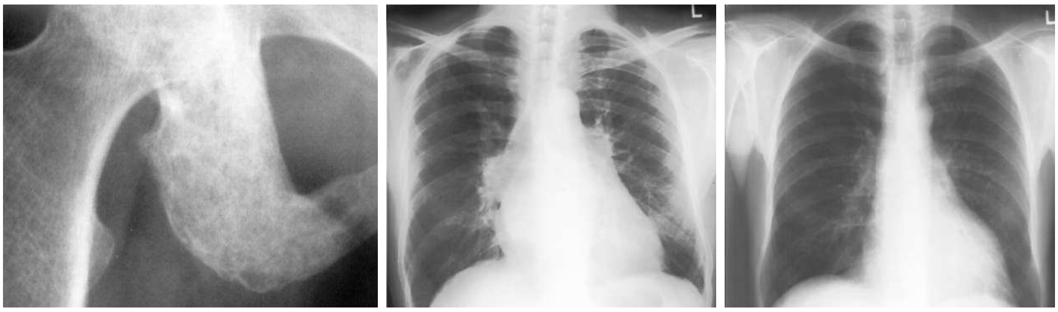

Abbildung 3.5: Die drei besten zurückgelieferten Bilder

I_n bzw. J_n geben dabei die Anzahl der Pixel in Bild I bzw. J mit der Farbe n an. Eine für Distanzmaße oft geforderte Eigenschaft ist die Symmetrie, d.h. es soll gelten

$$d(I,J) = d(J,I) \tag{3.2}$$

Dies ist jedoch für Gleichung (3.1) bei Bildern unterschiedlicher Dimension offenbar nicht gegeben. Um die Symmetrie-Eigenschaft einhalten zu können, muss sie demnach wie folgt umgeformt werden

$$d(I,J) = \sum_{n=1}^{N} \min[p(I_n), p(J_n)], \qquad 0 \le d(I,J) \le 1 \tag{3.3}$$

$p(I_n)$ bzw. $p(J_n)$ sind dabei die relativen Häufigkeiten von Pixeln in Bild I bzw. J mit der Farbe n. In beiden Fällen gibt die Distanz $d(I,J) = 1$ eine maximale Ähnlichkeit der verglichenen Histogramme an. Für die Darstellung einer offensichtlichen Schwäche jedes Histogramm-Vergleichs sei auf Abbildung 1.2 und die Erklärung dazu in Abschnitt 1.2.1 verwiesen.

Das Ergebnis einer Anfrage an die Datenbank mittels der in Gleichung (3.3) vorgestellten modifizierten Histogram Intersection ist in den Abbildungen 3.4 und 3.5 zu sehen. Die bei Abb. 3.4 in der ersten Zeile in der Distanzspalte auftauchende Distanz von $d(I,J) = 1$ ist dadurch zu erklären, dass das Anfragebild selber Teil der Datenbank ist und dementsprechend zu sich selber die maximale Ähnlichkeit hat. Das Ergebnis in diesem Beispiel ist durchaus als gutes Retrieval-Resultat anzusehen, da zwei der drei besten Bilder tatsächlich Thorax-Bilder darstellen (Precision $66,6\%$), in der Regel werden jedoch mit diesem einfachen Histogrammvergleich schlechte Retrieval-Ergebnisse erzielt. In Bezug auf eine 1-NN-Klassifikation des Anfragebildes ist das Ergebnis als Fehlklassifikation zu verstehen, da das beste zurückgelieferte Bild aus der Grobklasse 'Abdomen' stammt (siehe dazu auch die Kapitel 4 bis 6).

Kapitel 4

Datensammlung

Für die Experimente stand eine Datensammlung von 2080 Bildern aus dem 'medizinischen Alltag' zur Verfügung, die von der Klinik für Radiologische Diagnostik am Klinikum Aachen anonymisiert bereitgestellt wurden. Sie liegen sekundär digitalisiert vor, d.h. sie stammen nicht von einem digitalen Röntgengerät, sondern wurden nachträglich eingescannt. Zur Zeit nimmt ein solches digitales Röntgengerät am Klinikum der RWTH Aachen seinen Betrieb auf, das Bilder mit 12 Bit Graustufen (entspricht 4096 Grauwerten) liefern kann. Die vorliegenden Bilder wurden jedoch mit 256 Graustufen digitalisiert. Das ursprüngliche Filmmaterial beinhaltet natürlich grundsätzlich unendlich viele 'Graustufen', zum Zwecke der hier verfolgten Grobklassifikation reichen die vorliegenden 256 Graustufen allerdings aus. Die Bildgröße variiert stark, sie liegt zwischen etwa 200x200 und 2000x2000 Pixeln, je nach dargestellter Körperregion. Die Bilder reflektieren die reale Verteilung innerhalb der Klinik, wurden von einem Experten klassifiziert und mit einem entsprechenden Dateinamen versehen. Dieser enthält einen 8-stelligen Code, aus dem unter anderem auch die sechs Grobklassen abgeleitet werden können.

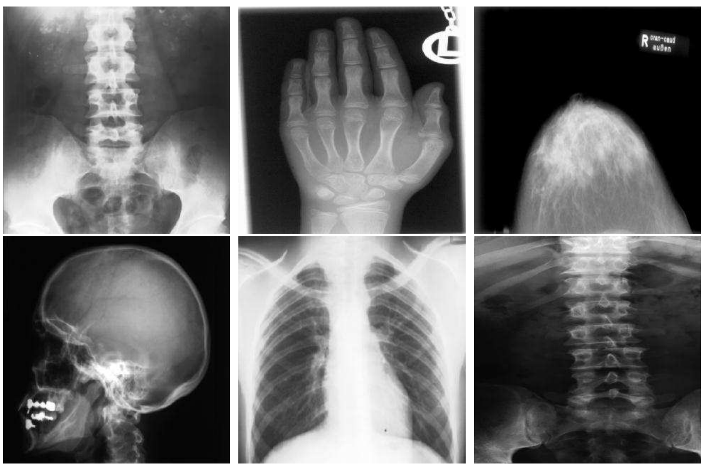

Abbildung 4.1: Röntgenbilder aus der verwendeten Datensammlung (bereits auf eine einheitliche Größe skaliert). Von o.l. nach u.r.: Abdomen, Extremitäten, Mamma, Schädel, Thorax und Wirbel.

4.1 Der 8-stellige Regionen- und Typcode

In [Wen1994] wurde ein 8-stelliger hierarchischer Code vorgestellt, der Angaben zur Region und zum Untersuchungstyp der jeweiligen Aufnahme macht und wie folgt aufgebaut ist:

- 3 Ziffern für die Region der Aufnahme, die hierarchisch von links nach rechts angeordnet sind
  - 1. Ziffer:
    - (0) Ganzer Körper
    - (1) Obere Extremitäten
    - (2) Untere Extremitäten
    - (3) Becken
    - (4) Wirbel
    - (5) Abdomen
    - (6) Thorax
    - (7) Schädel
    - (8) Mamma
    - Für die hier betrachteten sechs Grobklassen wurde die Region (0) ignoriert. Die Regionen (1) und (2) wurden zur Grobklasse 'Extremitäten' und die Regionen (3) und (5) zur Grobklasse 'Abdomen' zusammengefasst.
  - 2. und 3. Ziffer: Genauere Beschreibung der Region, beispielsweise beschreibt die Kombination (111) Finger und (112) die Mittelhandknochen
- Ein Buchstabe für die Aufnahmemodalität, unter anderem (C) für Computertomogramme oder (X) für Röntgenaufnahmen
- Ein Buchstabe für die Aufnahmetechnik, zum Beispiel (B) für Darmkontrastierung oder (E) für intraluminalen Kontrast
- Ein weiteres alphanumerisches Zeichen zur genaueren Spezifizierung der Aufnahmetechnik, beispielsweise (1) für stehend, (2) für liegend oder (Z) für hochauflösend
- Ein Buchstabe zur Beschreibung der Lateralität der Aufnahme: (B) beidseitig, (L) links, (R) rechts usw.
- Ein letzter Buchstabe für die Aufnahmeorientierung: (A) AP-Strahlengang, (B) PA-Strahlengang oder (D) rechts-vorne-Schrägstellung

Beispiel 4.1 *Der Dateiname* `thorax.00153.612IANLA` *beschreibt die 153. Thoraxaufnahme der Datensammlung, die ein Schlüsselbein darstellt (612), die Modalität 'Bildverstärker-Radiographie' (I) besitzt, mit intra-arterieller Kontrastmittelgabe (A) durchgeführt wurde, keine weitere Unterdifferenzierung der Aufnahmetechnik enthält (N), linksseitig ist (L) und einen AP-Strahlengang als Grundlage hat.*

Tabelle 4.1: Verteilung der Röntgenbilder in der vorliegenden Datensammlung

Klasse k	N_k	$\frac{N_k}{N}$
Abdomen	110	0,068
Extremitäten	706	0,437
Mamma	103	0,064
Schädel	110	0,068
Thorax	410	0,253
Wirbel	178	0,110
$N = \sum_{k=1}^{K} N_k$	1617	1,000

Obwohl jedes der vorliegenden Bilder mit einem solchen 8-stelligen Code versehen wurde, konzentriert sich diese Arbeit ausschließlich auf die angesprochenen sechs Grobklassen für Film-Folien-Radiographien (Röntgenbilder, Modalitäts-Code='X'), was einer dann übrigbleibenden Datenmenge von 1617 Bildern entsprach (die dazugehörigen Dateinamen sind dem File `/u/IRMA/Theiner/images/corpus/medical/scale32/X-Filenames.txt` zu entnehmen). Tabelle 4.1 gibt die Verteilung der vorliegenden Röntgenbilder wieder (N_k ist also die Anzahl der Bilder in Klasse k). Die Häufung in der Klasse 'Extremitäten' ist damit zu erklären, dass am Klinikum der RWTH Aachen vermehrt Hand-Aufnahmen zur *Maturitätsbestimmung* (Vergleich zwischen biologischem und physikalischem Alter) gemacht werden. Diese Verteilung ist also nicht ohne weiteres auf andere Kliniken übertragbar.

Die Grobklassifikation dieser Röntgenbilder stellt ein schwieriges Problem dar. Einerseits variieren die Bildqualitäten beträchtlich und es gibt eine erhebliche Varianz innerhalb der Kategorien (verursacht beispielsweise durch unterschiedliche Röntgen-Dosen, Objekt-Orientierung im Bild, Bilder mit und ohne pathologischen Befund usw.). Andererseits existiert beispielsweise eine starke visuelle Ähnlichkeit zwischen Bildern der Klasse 'Abdomen' und 'Wirbel'. Abbildung 4.1 zeigt jeweils ein Beispielbild für jede der sechs Grobklassen und verdeutlicht die gerade angesprochene Problematik. Abbildung 4.2 zeigt mögliche Variationen innerhalb der Grobklasse 'Thorax'. Im ersten Beispielbild ist dabei am unteren Bildrand ein kleines schwarzes Rechteck zu erkennen. Dies markiert die Position des sogenannten *Scribor* im Originalbild. Der Scribor beinhaltet Angaben zur Information der Person, von der die Aufnahme stammt. Er wurde zwecks Anonymisierung schwarz 'übermalt' und dadurch unkenntlich gemacht. Später wird noch gezeigt werden, dass diese 'ungewollten Zusatzinformationen' durchaus das Klassifikationsergebnis beeinflussen können und somit gesondert betrachtet werden müssen (siehe Abschnitt 6.2). Im Praxiseinsatz ist offenbar eine Scribor-Schwärzung nicht zu empfehlen, da die dort enthaltenen Daten klinikintern durch eine Mustererkennung ausgelesen und verwendet werden können, um Zusatzinformationen für ein erfolgreiches Retrieval zu erhalten. Dies wird eine weitere Teilaufgabe im IRMA-Projekt darstellen.

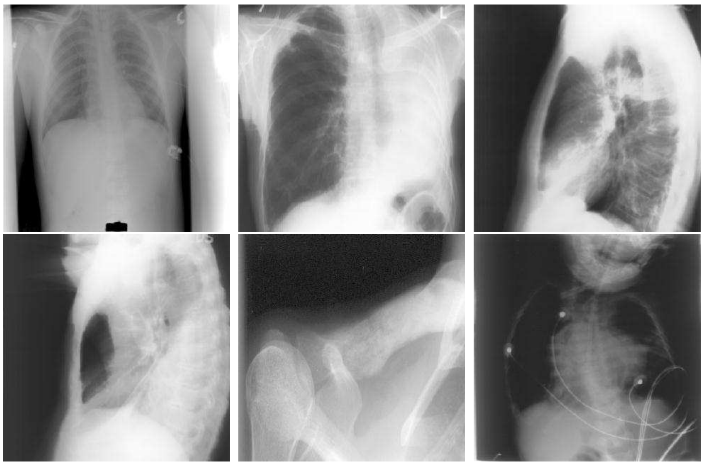

Abbildung 4.2: Variationen in der Klasse Thorax

4.2 Vorüberlegungen

In den Experimenten wurde die sogenannte *erscheinungsbasierte Mustererkennung* (*appearance based pattern recognition*) verwendet, d.h. jeder Pixel des Bildes wird als Merkmal interpretiert. Auf diese Weise wird jede im Bild enthaltene Information implizit zur Klassifikation herangezogen. Als Vorverarbeitungsschritt wurde lediglich eine Histogramm-Normalisierung durchgeführt, indem gegebenenfalls der Grauwert-Bereich jedes Bildes auf den zur Verfügung stehenden Bereich von 0 bis 255 gespreizt wurde (*MIN-MAX-Spreizung*). In einer ersten Testreihe unter Zuhilfenahme eines einfachen *Nearest-Neighbour-Klassifikators* wurde festgestellt, dass bei einer Skalierung der Bilder auf eine gemeinsame Größe von 128x128, 64x64, 32x32 und sogar 16x16 keine bedeutende Veränderung der Fehlerrate zu bemerken war. Der beste Kompromiss zwischen Datenmenge und Fehlerrate wurde bei einer Bildgröße von 32x32 erreicht, so dass diese Größe für alle weiteren Untersuchungen zugrundegelegt wurde (siehe Kapitel 6). Das dabei offensichtlich verlorengehende Seitenverhältnis der Bilder (die sogenannte *Aspect Ratio*) wurde zur späteren Verwendung separat in der Datenbank abgelegt.

Obwohl gerade die Eigenschaft der Invarianz von Röntgenbildern bezüglich kleinen Rotationen, Translationen oder Skalierungen offenbar eine wichtige Rolle spielt, wurden keine invarianten Merkmale extrahiert. Stattdessen wurden diese Invarianzen mit in den Klassifikationsalgorithmus selbst einbezogen. Dies wurde dadurch erreicht, dass Distanzmaße verwendet wurden, die beispielsweise gegenüber Rotation, Achsen-Deformation oder Skalierung invariant sind. Im weiteren Verlauf dieser Ausarbeitung werden Bilder der Dimension $I \times J$ folgendermassen beschrieben

$$x = \{x_{ij} \in \mathbb{R}^+\}, \quad i = 1, \ldots, I \text{ und } j = 1, \ldots, J \tag{4.1}$$

und es werden invariante Distanzmaße vorgestellt, die die euklidische Distanz $d(x, \mu) = ||x - \mu||^2$ zwischen dem gegebenen Bild x und einem Referenzbild μ ersetzen werden.

Kapitel 5

Klassifikation

Zur Klassifikation einer Beobachtung $x \in \mathbb{R}^D$ wird im folgenden die *Bayes'sche Entscheidungsregel* [Dud1973] herangezogen:

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \; \{p(k)p(x|k)\} \tag{5.1}$$

$p(k)$ ist dabei die a-priori Wahrscheinlichkeit der Klasse k, $p(x|k)$ die klassenbedingte Wahrscheinlichkeit für die Beobachtung x gegeben die Klasse k und $r(x)$ die Entscheidung des Klassifikators. Die Bayes'sche Entscheidungsregel ist optimal, falls $p(k)$ und $p(x|k)$ bekannt sind. In der Regel sind sie jedoch unbekannt. Daher werden Modelle für sie gewählt und deren Parameter mit Hilfe der Trainingsdaten geschätzt. $p(k)$ kann als relative Häufigkeit der Klasse k in den vorliegenden Trainingsdaten angenommen werden, wohingegen $p(x|k)$ häufig mit Hilfe *Gauß'scher Mischverteilungen* modelliert wird [Dah1998b, Dah1999, Mog1997].

5.1 Gauß'sche Mischverteilungen

Eine Gauß'sche Mischverteilung ist als Linearkombination Gauß'scher Einzelverteilungen $\mathcal{N}(x|\mu_{ki}, \Sigma_{ki})$ (im folgenden auch *Dichten* genannt) definiert, so dass für die klassenbedingten Wahrscheinlichkeiten folgt:

$$p(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) \tag{5.2}$$

I_k gibt dabei die Anzahl der Einzelverteilungen an, die zur Modellierung der Klasse k herangezogen wurden, c_{ki} ist Gewichtskoeffizient (mit $c_{ki} > 0$ und $\sum_i c_{ki} = 1$), μ_{ki} der Mittelwertsvektor und Σ_{ki} die Kovarianzmatrix der Dichte i der Klasse k.

Da in unserem Fall zur Schätzung der Verteilungsparameter $\{c_{ki}, \mu_{ki}, \Sigma_{ki}\}$ nur 1617 sehr hochdimensionale Daten zur Verfügung standen, wurde von diesem Ansatz Abstand genommen. Eine Anwendung von Gauß'schen Mischverteilungen auf medizinischen Daten wird beispielsweise in [Schr2000] und [Dah2000b] beschrieben. Stattdessen wurden die modellfreien Verfahren *Nearest Neighbour* (NN) bzw. ein *Kernel Density* basierter *m-NN* eingesetzt. Da auch dabei für eine (in

Bezug auf die Signifikanz der erzielten Fehlerraten) sinnvolle Aufteilung in eine Test- und eine Trainingsmenge zu wenig Daten verfügbar waren, wurden die Experimente mittels *leaving one out* durchgeführt, bei dem jeweils ein Datum als alleiniges Testdatum und alle anderen Daten als Trainingsdaten (in diesem Fall besser 'Referenzdaten') betrachtet werden. Bei diesem Ansatz sind die Trainings- und Testmenge also stets disjunkt.

5.2 Modellfreie Klassifikatoren

Betrachtet wird die Beobachtung $x \in \mathbb{R}^D$ und alle Referenzdaten x_{nk} ($n = 1, \ldots, N_k$, $x \neq x_{nk}$ im leaving-one-out Fall). Für $m = 1$ ist dann die (1-)Nearest-Neighbour-Entscheidungsregel (z.B. in [Ney1998]) definiert als

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmin}} \ \{ \min_{n=1,\ldots,N_k} d(x, x_{nk}) \} \tag{5.3}$$

Die Entscheidung fällt also mit geeignetem Distanzmaß $d(x, x_{nk})$ für die Beobachtung x auf die Klasse $r(x)$, der der 'nächstliegende Nachbar' aus den Referenzdaten angehört. Als Erweiterung dieser 1-NN-Entscheidungsregel wurden Kernel Densities (KD) herangezogen ([Dev1996], S. 147-153). Man erhält dabei für jede Beobachtung x folgende klassenbedingte Wahrscheinlichkeit (vergleiche Formel (5.1)):

$$p(x|k) = \frac{1}{N_k} \sum_{n=1}^{N_k} \frac{1}{A_k} \exp\left(-\frac{d(x, x_{nk})}{\sigma_k^2}\right), \tag{5.4}$$

Dabei ist σ_k^2 die klassenabhängige Varianz in den Referenzdaten und N_k die Anzahl der Referenzbilder, die der Klasse k angehören. Strenggenommen hängt der Normalisierungsfaktor A_k von k ab, diese Abhängigkeit ist allerdings so schwach, dass der Faktor in den Experimenten vernachlässigt wurde. Nach der Multiplikation von $p(x|k)$ mit der (über relative Häufigkeiten modellierten) a-priori Wahrscheinlichkeit $p(k) = \frac{N_k}{N}$ ergibt sich

$$\begin{aligned} p(k) \cdot p(x|k) &= \frac{N_k}{N} \frac{1}{N_k} \sum_{n=1}^{N_k} \exp\left(-\frac{d(x, x_{nk})}{\sigma_k^2}\right) && (5.5) \\ &= \frac{1}{N} \sum_{n=1}^{N_k} \exp\left(-\frac{d(x, x_{nk})}{\sigma_k^2}\right) && (5.6) \end{aligned}$$

Um jetzt die Tatsache zu kompensieren, dass die Varianzen σ_k^2 bei wenigen Referenzdaten im allgemeinen zu klein geschätzt werden, werden diese noch mit einem Faktor $\gamma > 1$ multipliziert. Der Vorfaktor $\frac{1}{N}$ ist darüberhinaus unabhängig von k und spielt daher bei einer Maximierung über k von (5.1) keine Rolle. Für die KD-Entscheidungsregel ergibt sich schließlich:

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \ \left\{ \sum_{n=1}^{N_k} \exp\left(-\frac{d(x, x_{nk})}{\gamma \cdot \sigma_k^2}\right) \right\} \tag{5.7}$$

Die in den Experimenten dieser Diplomarbeit verwendete KD-basierte m-NN-Entscheidungsregel läßt sich nun leicht herleiten. Ein m-Nearest-Neighbour berücksichtigt nicht mehr nur die minimale

Distanz der Beobachtung zu den Referenzdaten, sondern die m kleinsten Distanzen. Es werden also die m nächsten Nachbarn aus den Referenzdaten ausgewählt. Diese bilden dann die Menge $M \subset \{x_{nk} : n = 1, \ldots, N_k, \quad k = 1, \ldots, K\}$. Die Berechnung in der Formel (5.7) wird nun nicht mehr über alle Referenzdaten, sondern nur noch über die in der Menge M enthaltenen durchgeführt. Die KD-basierte m-NN-Entscheidung wird dann analog nach Gleichung (5.7) wie folgt getroffen:

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \left\{ \sum_{\substack{n=1 \\ x_{nk} \in M}}^{N_k} \exp\left(-\frac{d(x, x_{nk})}{\gamma \cdot \sigma_k^2} \right) \right\} \tag{5.8}$$

Diese Vereinfachung wird dadurch motiviert, dass 'weit entfernte' Referenzdaten in der Gleichung (5.7) wegen der im wesentlichen durchgeführten Berechnung von $e^{-d(x, x_{nk})}$ nur minimal eingehen und daher bei der Entscheidung keine Rolle spielen. Im Falle von $m = \sum_k N_k = N$ erhält man für M die Menge aller Referenzdaten, und es ergibt sich wieder der ursprüngliche KD-Klassifikator.

5.3 Invarianzen und Distanzmaße

Von zentraler Bedeutung für die zu treffende Entscheidung ist natürlich die Wahl eines geeigneten Distanzmaßes. In den durchgeführten Experimenten bildete diese Wahl und die experimentelle Evaluierung unter Beobachtung der Fehlerrate auch den größten Anteil.

In der *erscheinungsbasierten Mustererkennung* (*appearance based pattern recognition*) wird jeder Pixel als Merkmal interpretiert. Zunächst wird als Distanzmaß für die Gleichungen (5.3), (5.7) und (5.8) der einfache euklidische Vergleich gewählt. Dabei gilt für zwei Bilder x und μ der Dimension $I \times J$:

$$d(x, \mu) = \sum_{i=1}^{I} \sum_{j=1}^{J} \|x_{ij} - \mu_{ij}\|^2 \tag{5.9}$$

Es wird also jeder Pixel des Bildes x mit dem Pixel aus Bild μ verglichen, der sich exakt an derselben Position (i,j) befindet. Invarianzen bezüglich Translation, Rotation oder Skalierung werden dabei nicht berücksichtigt. Im folgenden werden Distanzmaße vorgestellt, in denen Invarianzen implizit kompensiert werden, ohne dass sie in den Referenzdaten selbst auftauchen müssen.

Eine andere Möglichkeit, Invarianzen zu modellieren, liegt in der Erzeugung sogenannter *virtueller Daten*. Dabei werden die Originaldaten zunächst geeignet transformiert, ohne dabei ihre Klassenzugehörigkeit zu ändern. Diese transformierten Daten werden dann der Trainings- oder auch Testmenge hinzugefügt [Dah2000a]. Dort können sie dann sehr effizient zur Verringerung der Klassifikationsfehlerrate beitragen, da eine robustere Parameterschätzung möglich ist und eine lokale Invarianz gegenüber den Transformationen vorliegt, die die 'virtuellen' Daten erzeugt haben (siehe dazu auch Kapitel 8).

5.3.1 Tangentendistanz nach Simard

Im Jahre 1993 stellte SIMARD ein invariantes Distanzmaß vor, genannt *Tangentendistanz*, das sich im Bereich der Schrifterkennung als sehr effektiv herausstellte [Sim1993]. Er bemerkte dabei zunächst, dass entsprechend kleine Transformationen bestimmter Objekte (wie zum Beispiel Schriftzeichen, oder wie in diesem Fall Objekte in Röntgenbildern) nicht deren Klassenzugehörigkeit verändern. Wie bereits im letzten Abschnitt gezeigt, reagiert die euklidische Distanz sehr empfindlich auf solche Transformationen.

Wird ein Muster x mittels einer Transformation $t(x, \alpha)$ transformiert (zum Beispiel skaliert, wobei α in diesem Fall den Skalierungsfaktor darstellt), so bildet die Menge aller transformierten Muster

$$M_x = \{t(x, \alpha) : \alpha \in \mathbb{R}\} \subset \mathbb{R}^{I \times J} \tag{5.10}$$

eine eindimensionale Kurve innerhalb des Merkmals-Raumes. Werden nun L solche Transformationen mit Parametern α_l betrachtet, so ist M_x eine Mannigfaltigkeit mit höchstens L Dimensionen. Die Distanz zwischen zwei Mustern kann nun als minimale Distanz zwischen ihren entsprechenden Mannigfaltigkeiten definiert werden, die bezüglich der Transformationen $t_1, \ldots, t_L$ vollständig invariant sind. Unglücklicherweise stellt die Berechnung dieser Distanz ein schwieriges Optimierungsproblem dar, und die benötigten Mannigfaltigkeiten besitzen darüber hinaus im allgemeinen keine analytische Darstellung. Kleine Transformationen eines Musters x können jedoch von einem Tangenten-Unterraum $\hat{M}_x$ der Mannigfaltigkeit M_x am Punkt x approximiert werden. Diese Transformationen kann man dadurch erhalten, dass man zu x eine Linearkombination der Vektoren $T_l(x)$ addiert, die den Tangenten-Unterraum aufspannen. Die Mannigfaltigkeit M_x kann also als Näherung erster Ordnung wie folgt dargestellt werden:

$$\hat{M}_x = \{x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) : \alpha_l \in \mathbb{R}\} \subset \mathbb{R}^{I \times J} \tag{5.11}$$

Die (einseitige) Tangentendistanz $d_T(x, x_{nk})$ ist dann definiert als:

$$d_T(x, x_{nk}) = \min_{\bar{\alpha}} \{||x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) - x_{nk}||^2\}, \tag{5.12}$$

wobei $\bar{\alpha} \in \mathbb{R}^L$ die Vektordarstellung der α_l ist. Die sogenannten Tangentenvektoren $T_l(x)$ können mittels einfacher Differenzen (Gradientenbildung) zwischen dem Originalbild x und einer angemessen kleinen Transformation von x berechnet werden [Sim1993]. In Abbildung 5.1 sind Beispielbilder dargestellt, die mit (5.11) berechnet wurden (es wurden Ziffern ausgewählt, da sie sich besonders zur Demonstration der bei der Tangentenberechnung auftretenden Effekte eignen). Eine zweiseitige Tangentendistanz ist ebenfalls berechenbar:

$$d_T(x, x_{nk}) = \min_{\bar{\alpha}, \bar{\beta}} \{||x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) - x_{nk} - \sum_{l=1}^{L} \beta_l \cdot T_l(x_{nk})||^2\} \tag{5.13}$$

Allerdings steigt dabei der Berechnungsaufwand dramatisch an, ohne zu einer bedeutenden Verbesserung der Klassifikation zu führen [Sim1993, Dah2000d]. In den im Rahmen dieser Arbeit durchgeführten Experimenten wurden die Tangentenvektoren für Translationen (in x und y-Richtung),

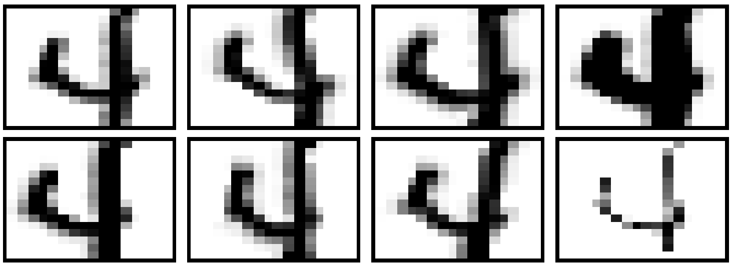

Abbildung 5.1: Auswirkung der Tangentenberechnung auf Ziffern (Originalbild links oben)

Rotation, Skalierung und Achsen-Deformationen (ebenfalls in x und y-Richtung) wie von SIMARD vorgeschlagen berechnet, allerdings wurde der Tangentenvektor für die *Strichdicke* durch einen bei Röntgenbildern vielversprechenderen Tangentenvektor für *Helligkeit* ersetzt. Alle Elemente dieses Vektors wurden auf einen konstanten Wert gesetzt, um Helligkeitsschwankungen im Bild zu modellieren, wie sie bei unterschiedlichen Röntgen-Dosen auftreten. Diese sieben Tangentenvektoren wurden dann unter Verwendung des KD-basierten Klassifikators aus Abschnitt 5.2 zur Röntgenbild-Klassifikation herangezogen.

5.3.2 Effiziente Berechnung der Tangentendistanz

Die einseitige Tangentendistanz wird konzeptionell in den folgenden Schritten berechnet:

1. Berechne die Tangentenvektoren für die Beobachtung x.
2. Berechne eine Orthonormalbasis des Tangenten-Unterraumes (mittels einer *Singular Value Decomposition* (SVD) [Pre1992], S. 59-70).
3. Berechne die Projektion $\hat{x}_{nk}(x)$ eines Referenzbildes x_{nk} in den Tangenten-Unterraum (mit Ursprung x).
4. Berechne $d_T(x, x_{nk}) = d(\hat{x}_{nk}(x), x_{nk})$.

Die Schritte 1. und 2. können vorab berechnet werden, nur die Schritte 3. und 4. müssen während der Klassifikation durchgeführt werden. Schritt 2. dient dabei lediglich der Vereinfachung der nachfolgenden Projektion, da diese dann unter Zuhilfenahme einfacher Skalarprodukte statt aufwändiger linearer Gleichungssysteme vorgenommen werden kann. Dieser Schritt kann im allgemeinen natürlich nur für die Trainingsdaten vorab berechnet werden. In dem hier vorliegenden Fall (*leaving one out*) macht das allerdings keinen Unterschied.

Unter der Voraussetzung, dass die Tangentenvektoren orthogonal sind (was durch eine SVD gewährleistet wird), können die Schritte 3. und 4. effizient in einem einzigen Schritt wie folgt durchgeführt werden:

$$d_T(x, x_{nk}) = \|x - x_{nk}\|^2 - \sum_{l=1}^{L} \frac{[(x - x_{nk})^t \cdot T_l(x)]^2}{\|T_l(x)\|^2} \tag{5.14}$$

5.3.3 Iterative Berechnung der Tangentendistanz

Wie leicht zu sehen ist, kann die oben angegebene schrittweise Berechnung der einseitigen Tangentendistanz um ein Iterationsverfahren erweitert werden, indem die folgenden Schritte hinzugefügt werden:

5. Setze $x := \hat{x}_{nk}(x)$.
6. Ist das Abbruchkriterium erfüllt? Falls nicht, dann weiter mit Schritt 1.

Ein 'Abbruchkriterium' in den Experimenten dieser Arbeit war beispielsweise, dass 30 Iterationen durchgelaufen sind. Denkbar wäre allerdings auch eine Untersuchung der Distanzabweichungen von Iteration zu Iteration. Liegt diese Abweichung unter einem festzulegenden Schwellwert, so bricht die Iteration ab (Konvergenzkriterium).

Diese Vorgehensweise hat allerdings einen gravierenden Nachteil. Die Schritte 1. und 2. können nun ganz offensichtlich nicht mehr aus dem Klassifikationsprozess herausgenommen werden. Sie müssen für jedes Bildpaar und für jede Iteration erneut berechnet werden, was den Zeitaufwand für eine Klassifikation deutlich ansteigen lässt. Auch wird man in Kapitel 6 erkennen, dass dieser Mehraufwand in Relation zu der damit erreichbaren Verbesserung bei der Grobklassifikation von Röntgenbildern nicht auf Anhieb zu rechtfertigen ist. Allerdings wurden vor allem im Bereich der Ziffernerkennung erkennbare Verbesserungen erzielt [Dah2000c, Key2000].

Die Tangentendistanz ist ein sehr effektives Mittel zur Kompensation kleiner globaler Transformationen eines Bildes. Im folgenden Abschnitt wird ein einfaches, aber dennoch sehr effektives *Image Distortion Modell* für *lokale* Transformationen vorgestellt. Im Kapitel 6 wird gezeigt, dass beide Verfahren für sich genommen sehr gut arbeiten, aber dass die besten Resultate bei der Kombination der beiden erzielt werden, sie also erfreulicherweise additiv sind.

5.3.4 Das Image Distortion Modell

Der vierte Schritt bei der (konzeptionellen) Berechnung der Tangentendistanz erfordert weiterhin die Berechnung der euklidischen Distanz zwischen einem Referenzbild x_{nk} und dessen Projektion $\hat{x}_{nk}(x)$ in den Tangenten-Unterraum mit der Beobachtung x als Ursprung. Obwohl kleine globale Transformationen durch diesen Projektionsschritt kompensiert werden, reagiert die Berechnung der Distanz weiterhin sehr empfindlich auf *lokale* Transformationen der Bilder, wie sie z.B. durch *Rauschen* oder unterschiedliche Scribor-Positionen (siehe Abschnitt 4.1) verursacht werden. Genau dies berücksichtigt aber das im folgenden vorgestellte *Image Distortion Modell* (IDM) [Dah2000d, Key2000].

Bei der Berechnung der Distanz zwischen einem Bild x und einem Referenzbild $\mu = x_{nk}$ der Dimension $I \times J$ werden dabei lokale Deformationen erlaubt. Es wird also nicht einfach der quadratische Fehler zwischen den Pixeln x_{ij} und μ_{ij} berechnet, sondern es wird zunächst in einer bestimmten Umgebung R_{ij} um den Pixel μ_{ij} des betrachteten Referenzbildes nach dem 'am besten passenden' Pixel gesucht (siehe Abbildung 5.2):

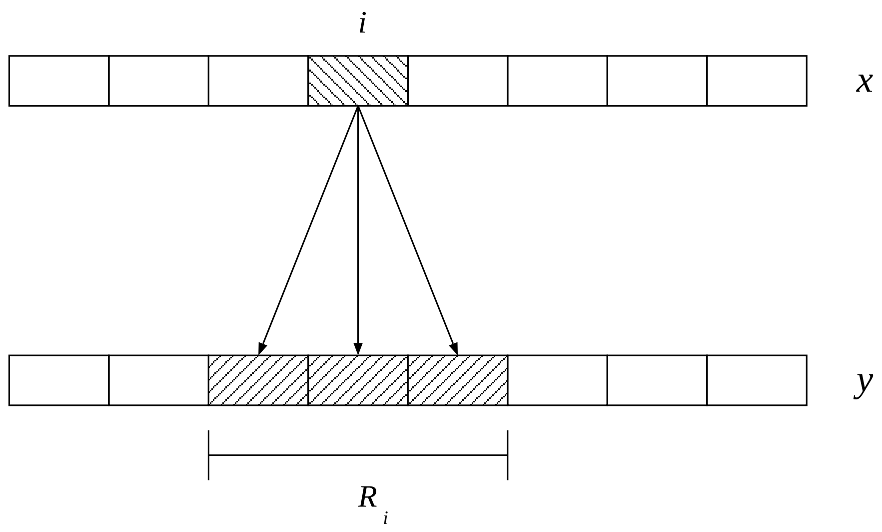

Abbildung 5.2: Eindimensionales Beispiel für das Image Distortion Modell mit $r = 1$

$$d_{\text{IDM}}(x, \mu) = \sum_{i=1}^{I} \sum_{j=1}^{J} \min_{(i',j') \in R_{ij}} \|x_{ij} - \mu_{i'j'}\|^2 \tag{5.15}$$

Typischerweise wird die Umgebung $R_{ij} = (2r + 1) \times (2r + 1)$ quadratisch um den zentralen Pixel μ_{ij} gewählt (siehe dazu Abbildung 6.3). Augenscheinlich können mit diesem uneingeschränkten *Deformationsmodell* sowohl gewollte als auch ungewollte (d.h. bedeutungslose) Transformationen modelliert werden. Nichtsdestotrotz führt eine geeignete Wahl von R_{ij} zu einer signifikanten Verbesserung bei der Klassifikation von Röntgenbildern (siehe Kapitel 6).

5.3.5 Ein erweitertes Distortion Modell

Bei Betrachtung der Gleichung (5.15) wird deutlich, dass mit wachsender Umgebung R_{ij} die daraus entstehende Transformation die Voraussetzung verletzt, dass die Klasse des Ursprungsbildes dieselbe wie die des transformierten Bildes sein soll. Tatsächlich kann $d_{\text{IDM}}(x, \mu)$ damit zwischen fast beliebig wählbaren Bildern x und μ nahe Null gebracht werden, was natürlich zu einem gravierenden Anstieg der Klassifikations-Fehlerrate führt. Um dies abzufangen, wird eine Kostenfunktion $C(i, i', j, j')$ eingeführt, die eine Strafe für die Deformation des Pixels x_{ij} der Beobachtung in das Pixel $\mu_{i'j'}$ des Referenzbildes angibt:

$$d_{\text{IDM}}(x, \mu) = \sum_{i=1}^{I} \sum_{j=1}^{J} \min_{(i',j') \in R_{ij}} \{\|x_{ij} - \mu_{i'j'}\|^2 + C(i, i', j, j')\} \tag{5.16}$$

Es gibt zumindest zwei Möglichkeiten zur Bestimmung der Kostenfunktion $C(i, i', j, j')$:

- Wähle $C(i, i', j, j')$ empirisch, zum Beispiel mittels einer gewichteten euklidischen Distanz zwischen den Pixeln (i, j) und (i', j'). Dadurch werden kleinere Transformationen gegenüber (höchstwahrscheinlich ungewollten) weitreichenderen Pixeltransformationen bevorzugt.

- Lerne $C(i, i', j, j')$ unter Betrachtung von Trainingsbildern und mittels eines *maximum-likelihood* Ansatzes. Das heißt, man führt im Training gewünschte (sprich 'bedeutsame') Transformationen durch und wählt $C(i, i', j, j')$ anhand der relativen Häufigkeiten dieser Transformationen. Je häufiger eine Transformation im Training durchgeführt wurde, desto geringer ihre Kosten.

Die in diesem Kapitel vorgestellten Verfahren wurden anhand ausführlicher Experimente untersucht und ihre Parameter unter Betrachtung der Fehlerrate optimiert. Die Ergebnisse dieser Experimente werden im folgenden Kapitel präsentiert.

Kapitel 6

Ergebnisse

Wie bereits in Kapitel 4 angekündigt, wurden die Bilder der Datensammlung für die im Rahmen dieser Diplomarbeit durchgeführten Experimente zur Grobklassifikation auf eine Standardgröße von 32×32 Pixeln skaliert. Dies zog keine merkliche Veränderung in der Klassifikationsfehlerrate nach sich, führte allerdings zu einem deutlichen Geschwindigkeitsgewinn. Die Berechnung eines einfachen 1-NN auf Röntgenbildern der Größe 320×320 ergab eine Fehlerrate von $18,0\%$, wobei zur Klassifikation eines einzelnen Bildes auf einem Rechner mit einer 500MHz Digital ALPHA CPU etwa 30 Sekunden benötigt wurden. Die Klassifikation desselben Bildmaterials, diesmal jedoch auf 32×32 herunterskaliert, ergab eine Fehlerrate von $18,1\%$, benötigte jedoch nur noch etwa $0,4$ CPU-Sekunden pro Bild.

6.1 Erste Experimente

Nach der Wahl einer einheitlichen Bildgröße wurden als Merkmale zunächst Histogramme $H \in \mathbb{R}^{256}$ extrahiert. Histogramme geben die relativen Häufigkeiten an, mit der die unterschiedlichen Grautöne im Bild x der Dimension $I \times J$ vorkommen:

$$H_x(g) = \frac{1}{I \cdot J} \sum_{i=1}^{I} \sum_{j=1}^{J} \delta(x_{ij}, g) \tag{6.1}$$

$g \in G$ ist dabei ein erlaubter Grauwert und δ das Kroneckersymbol:

$$\delta(a,b) = \begin{cases} 1 & \text{falls} \quad a = b \\ 0 & \text{sonst} \end{cases} \tag{6.2}$$

x_{ij} gibt den Grauwert im Bild x an der Stelle (i,j) an. Die räumliche Ordnung der Pixel geht dabei offensichtlich verloren, was einen erheblichen Informationsverlust zur Folge hat. Die euklidische Distanz zwischen zwei Histogrammen H_x und H_μ ist dann definiert als:

$$d(H_x, H_\mu) = \sum_{g \in G} \|H_x(g) - H_\mu(g)\|^2 \tag{6.3}$$

Tabelle 6.1: Fehlerraten (*leaving one out*) für unterschiedliche Distanzmaße ohne Clipping

Distanzmaß	Fehlerrate [%]		Ranking
	1-NN	KD	1-NN
Euklidische Distanz	18,1	16,4	1,30
Tangentendistanz	15,3	14,8	1,28
Distortion Modell $r = 1$	16,5	14,7	1,29
IDM + TD $r = 0,8$	14,2	12,5	1,25

Tabelle 6.2: Fehlerraten (*leaving one out*) für unterschiedliche Distanzmaße mit Clipping ($S = 3500$)

Distanzmaß	Fehlerrate [%]		Ranking
	1-NN	KD	1-NN
Euklidische Distanz	14,8	14,2	1,26
Tangentendistanz	13,7	12,9	1,23
Distortion Modell $r = 0,5$	14,5	13,2	1,25
IDM + TD $r = 0,8$	10,6	10,3	1,21

Eine 1-NN-Klassifikation mit diesem Histogrammvergleich ergab eine Fehlerrate von $42,8\%$ auf den sechs Grobklassen. Die in Abschnitt 3.4 betrachtete (erweiterte) *Histogram Intersection Technik* ergab sogar eine Fehlerrate von $45,5\%$. Nahezu jede zweite Beobachtung x wurde also falsch klassifiziert. Dadurch motiviert wurden im weiteren Verlauf der Experimente Verfahren getestet, die das ganze Bild als Merkmal interpretieren und dementsprechend neben den reinen Grauwerten jede weitere im Bild enthaltene Information berücksichtigen.

Die naheliegendste Idee ist dabei die in Kapitel 5 vorgestellte 1-NN-Klassifikation mit einem euklidischen Distanzmaß. Sie lieferte mit einer Fehlerrate von $18,1\%$ das Basisergebnis für alle weiteren Untersuchungen. Eine ebenfalls in Kapitel 5 eingeführte KD-basierte m-NN-Klassifikation (im folgenden kurz *KD-Klassifikation* genannt) ergab immerhin schon eine Verbesserung von $18,1\%$ auf $16,4\%$. Daraufhin wurde die dort vorgestellte einseitige Tangentendistanz zur Röntgenbild-Klassifikation herangezogen. Wie in Tabelle 6.1 zu sehen ist, wurde dadurch die KD-Fehlerrate von $16,4\%$ auf $14,8\%$ gesenkt. Danach wurden Experimente mit dem *Image Distortion Modell* durchgeführt, wobei die Kostenfunktion für alle i, j, i', j' zunächst auf $C(i, i', j, j') = 0$ gesetzt wurde. Erstaunlicherweise wurde dabei mit $14,7\%$ ein Ergebnis erreicht, das sogar leicht besser ist als das der Tangentendistanz.

Ein weiteres Experiment sollte nun klären, ob sich die 'Gewinne' der beiden Verfahren additiv verhalten. Tatsächlich konnte durch die Kombination der beiden Distanzmaße die Fehlerrate nochmals von $14,8\%$ auf $12,5\%$ verbessert werden, indem bei Schritt 4 der konzeptionellen Berechnung der Tangentendistanz statt der euklidischen Distanz d nunmehr die Distanz des Image Distortion

Modells d_{IDM} zwischen einem Referenzbild x_{nk} und dessen Projektion $\hat{x}_{nk}(x)$ berechnet wurde (vergleiche Abschnitt 5.3.2). Abbildung 6.1 zeigt die erreichten Fehlerraten in Abhängigkeit von der betrachteten quadratischen Umgebung $R_{ij} = (2r + 1) \times (2r + 1)$. Das beste Resultat wurde mit $12,5\%$ bei $r = 0,8$ erreicht. Dabei wurde diese 'gebrochene' (also nicht ganzzahlige) Umgebung mittels linearer Interpolation zwischen den entsprechenden Pixeln der Umgebung bei $r = 1$ erreicht, veranschaulicht in Abbildung 6.3 (rechts). Bei dieser Umgebung mit $r = 0,8$ geht also der zentrale Pixel an der Position (i,j) mit einem Faktor von $0,2$, der jeweils betrachtete Pixel der Umgebung bei $r = 1$ mit einem Faktor von $0,8$ in die Berechnung ein.

6.2 Beschränkung der lokalen Strafe (Clipping)

Der maximale Beitrag, den ein einzelnes Pixel an der Gesamtdistanz zweier betrachteter Bilder haben kann, ist $255 \cdot 255 = 65025$, da es sich um Graustufenbilder mit 256 Graustufen handelt. Demnach können also schon einige wenige Pixel einen so gravierenden Anteil zu der Gesamtdi-

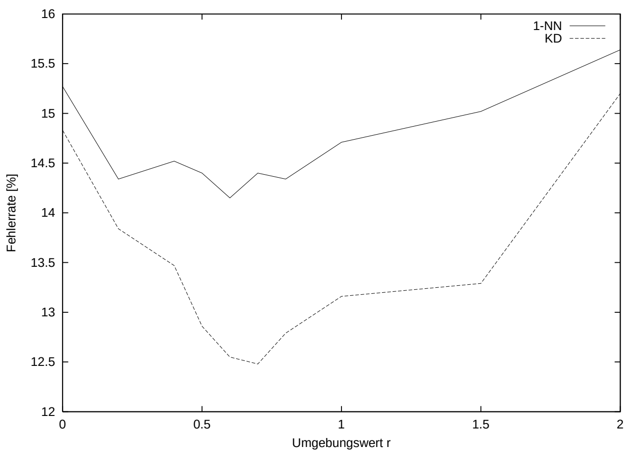

Abbildung 6.1: Fehlerraten (IDM+TD) in Abhängigkeit von dem Umgebungswert r, ohne Clipping

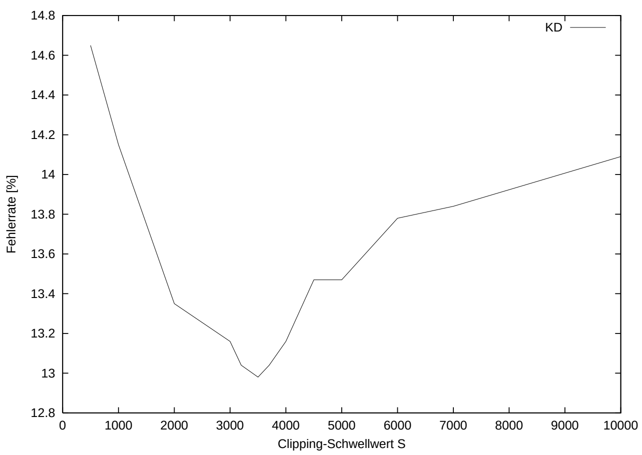

Abbildung 6.2: Fehlerraten (Tangentendistanz) bei unterschiedlichen Clipping-Schwellwerten

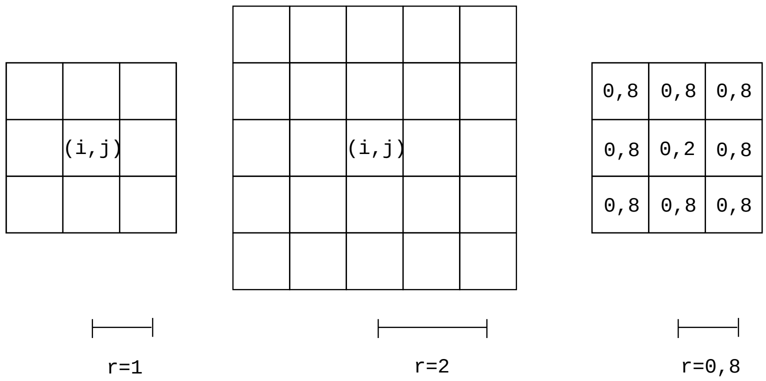

Abbildung 6.3: Beispiele für ganzzahlige und gebrochene Umgebungswerte r

stanz beitragen, dass es zu einer Fehlklassifikation kommt (beispielsweise durch Rauschen oder unterschiedliche Positionen des Scribor im Bild (siehe Abschnitt 4.1)). Man kann nun diesen Effekt kompensieren, indem man den maximal erreichbaren Beitrag eines einzelnen Pixels durch einen Schwellwert S nach oben beschränkt (das sogenannte *Clipping*). Auch diese einfache Methode hat eine deutliche Verbesserung der Fehlerrate zur Folge, wie in Tabelle 6.2 zu sehen ist. Das beste dadurch erzielte Resultat liegt bei $10,3\%$. Abbildung 6.2 zeigt den Verlauf der Fehlerrate für unterschiedliche Schwellwerte. Dabei verbesserte sich die Fehlerrate von $14,8\%$ ohne Clipping (sprich $S = 65025$) auf $12,9\%$ bei einem Schwellwert von $S = 3500$ (vergleiche in den Tabellen 6.1 und 6.2 die Einträge für KD-Fehlerraten der Tangentendistanz).

6.3 TOP-K-Evaluierung und Ranking

Als weiteres Kriterium zur Bestimmung der 'Güte' eines getesteten Verfahrens (bezogen auf seine Retrieval-Eigenschaften) wird hier ein sogenanntes *Ranking* verwendet. Dabei werden für jedes Testdatum x die nächsten Nachbarn jeder Klasse aus den Referenzdaten x_{nk} nach der Distanz sortiert. Man erhält zunächst also ein sogenanntes *TOP-K* der Klassen für jedes Testdatum. Das (mittlere) Ranking ρ ergibt sich schließlich als Mittelwert der Position der *korrekten* Klasse in diesen TOP-K über alle Testdaten. Offensichtlich gilt $1 \leq \rho \leq K$ für K betrachtete Klassen. Je näher ρ an 1 liegt, desto besser das Verfahren ($\rho = 1$ bedeutet, dass alle Bilder korrekt klassifiziert wurden). In den Tabellen 6.1 und 6.2 ist beispielhaft das jeweilige Ranking für die 1-NN-Klassifikation angegeben. Tabelle 6.3 zeigt die TOP-K-Klassifikationsraten für ein 1-NN-Experiment mit der Gesamtfehlerrate $9,46\%$. Dabei ist zu erkennen, dass bereits bei der Betrachtung der besten drei Klassen nur noch eine Restfehlerrate von rund $1,9\%$ übrigbleibt. Daher ist das Verfahren trotz der ansonsten noch recht hohen TOP-1-Fehlerrate bereits für den Einsatz innerhalb des IRMA-Systems (siehe Kapitel 3) geeignet.

Tabelle 6.3: TOP-K-Klassifikationsraten für eine 1-NN-Klassifikation

	Klassifikationsrate [%]
TOP-1	90,54
TOP-2	96,41
TOP-3	98,08
TOP-4	99,26
TOP-5	99,75
TOP-6	100,00

6.4 Berücksichtigung der Aspect Ratio

Eine Untersuchung der verbleibenden Klassifikationsfehler ergab, dass eine ganze Reihe der Fehlklassifikationen vermieden werden könnte, wenn das ursprünglich durch die Skalierung auf eine Einheitsgröße verlorengegangene Seitenverhältnis der Bilder (die sogenannte *Aspect Ratio*) berücksichtigt würde. Daraufhin wurde ein Bestrafungsterm für das Seitenverhältnis eingeführt, der auf dem Unterschied der Aspect Ratios der beiden betrachteten Bilder basiert.

Die Distanz d_{IDM} zwischen zwei Bildern für das Image Distortion Modell ergibt sich somit zu:

$$d_{\text{IDM}}(x,\mu) = \nu \cdot [A(x) - A(\mu)]^2 + \sum_{i=1}^{I} \sum_{j=1}^{J} \min_{(i',j') \in R_{ij}} \{\|x_{ij} - \mu_{i'j'}\|^2 + C(i,i',j,j')\} \tag{6.4}$$

Dabei gibt $A(x)$ das Seitenverhältnis (längere Seite/kürzere Seite) eines Bildes x und ν einen Wichtungsfaktor an. Analog kann der Strafterm $\nu \cdot [A(x) - A(\mu)]^2$ natürlich auch mit den anderen vorgestellten Distanzmaßen eingesetzt werden. Durch diesen zusätzlichen Bestrafungsterm ließ sich die (KD-) Klassifikationsfehlerrate mit $\nu = 1220$ von $10,3\%$ auf $8,9\%$ (Ranking 1,18) reduzieren. Danach wurde die Kostenfunktion $C(i,i',j,j')$ auf eine gewichtete euklidische Distanz gesetzt (vergleiche Abschnitt 5.3.5), was zu einer weiteren Verbesserung von $8,9\%$ auf $8,7\%$ führte, bei einem Ranking von 1,17.

Tabelle 6.4 zeigt beispielhaft eine sogenannte *Confusion-Matrix* für eine durchgeführte 1-NN-Klassifikation *ohne* Berücksichtigung der Aspect Ratio, wohingegen in Tabelle 6.5 dasselbe Experiment zu sehen ist, diesmal jedoch *mit* Berücksichtigung der Aspect Ratio.

In der Matrix stehen links die *korrekten* Klassen der Testdaten und oben die durch die Klassifikation zugeordneten Klassen. Die Hauptdiagonale der Matrix gibt also die Anzahl der korrekt klassifizierten Testdaten an. Dabei ist sofort zu erkennen, dass es durch die Verwendung der Aspect Ratios zu einem signifikanten Rückgang der Fehlklassifikationen auf die Klasse 'Extremitäten' (2. Spalte) gekommen ist, was durch das in dieser Klasse auftretende markante Seitenverhältnis von durchschnittlich etwa 2.5:1 zu erklären ist. Auch verbesserte sich die Fehlerrate für jede Klasse deutlich. Die vergleichsweise hohen Fehlerraten für die Klassen 'Abdomen' und 'Wirbel' sind auf

Tabelle 6.4: Beispiel einer 1-NN-Confusion-Matrix ohne Aspect Ratio (Fehlerrate $10,8\%$)

	Abd.	Ext.	Mam.	Sch.	Tho.	Wir.	$\Sigma = N_k$	Fehlerrate [%]
Abdomen	**73**	20	2	7	4	4	110	33,6
Extremitäten	11	**651**	11	11	10	12	706	7,8
Mamma	0	3	**98**	2	0	0	103	4,8
Schädel	2	7	0	**92**	2	7	110	16,4
Thorax	2	9	0	4	**394**	1	410	3,9
Wirbel	7	23	3	11	5	**129**	178	27,5

Tabelle 6.5: 1-NN-Confusion-Matrix mit Aspect Ratio (Fehlerrate $9,4\%$)

	Abd.	Ext.	Mam.	Sch.	Tho.	Wir.	$\Sigma = N_k$	Fehlerrate [%]
Abdomen	**78**	11	3	10	6	2	110	29,1
Extremitäten	11	**652**	10	14	9	10	706	7,6
Mamma	0	0	**102**	0	0	1	103	1,0
Schädel	2	3	0	**96**	2	7	110	12,7
Thorax	2	7	0	2	**399**	0	410	2,7
Wirbel	5	20	6	9	3	**135**	178	24,2

das dort besonders inhomogene Datenmaterial und teilweise auf mehrfach gelabelte Bilder zurückzuführen (siehe dazu auch Abbildung 6.4 und Abschnitt 4.1). Die sehr gute Klassifikation in der Klasse 'Mamma' ergibt sich ebenfalls durch das Datenmaterial. Mamma-Bilder weisen im Vergleich zu Bildern aus den anderen Grobklassen sehr große, nahezu schwarze Bereiche auf, die diskriminativ für ihre Klasse sind.

6.5 Iterationsverfahren

In Abschnitt 5.3.3 wurde ein Iterationsverfahren für die Berechnung der Tangentendistanz vorgestellt, das bei 30 Iterationen nochmals zu einer Verbesserung der Fehlerrate von $8,7\%$ auf $8,6\%$ führte. Diese minimale Verbesserung steht allerdings in keinem Verhältnis zum Zeitaufwand der Klassifikation, der dadurch etwa um den Faktor 80 anstieg (von ca. 20 auf ca. 1600 Minuten für die Klassifikation der 1617 Röntgenbilder). Die in Abschnitt 4.2 angesprochene Histogramm-Normalisierung (MIN-MAX-Spreizung) als Vorverarbeitungsschritt, die in einem parallel zu diesem Iterationsversuch ablaufenden Experiment durchgeführt wurde, brachte nämlich ohne signifikante Änderung des Zeitaufwandes für die Erkennung ebenfalls eine Verbesserung auf $8,6\%$. Das Ranking konnte dadurch auf $1,16$ verbessert werden.

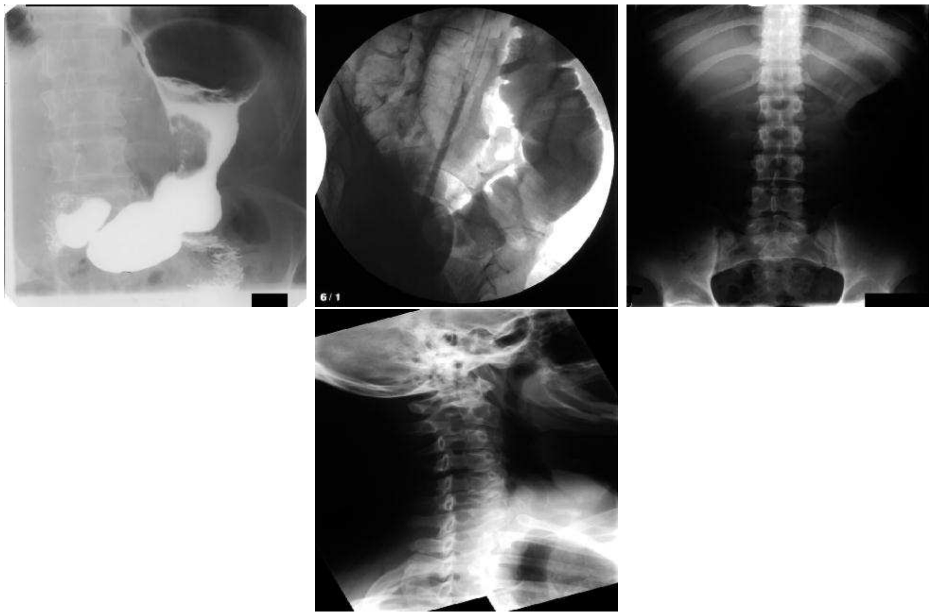

Abbildung 6.4: Obere Reihe: Falsch klassifizierte Bilder der Klasse 'Abdomen', v.l.n.r: Abdomen → Thorax, Abdomen → Schädel, Abdomen → Wirbel. Unten: Wechselseitig falsch klassifiziertes, mehrfach gelabeltes Bild (das gleiche Bild existiert sowohl in der Klasse 'Schädel' als auch 'Wirbel')

Tabelle 6.6: 4-NN-Confusion-Matrix mit Aspect Ratio (Fehlerrate 8, 6%)

	Abd.	Ext.	Mam.	Sch.	Tho.	Wir.	$\Sigma = N_k$	Fehlerrate [%]
Abdomen	**80**	18	0	4	5	3	110	27,3
Extremitäten	8	**678**	1	5	5	9	706	4,0
Mamma	1	8	**93**	0	0	1	103	9,7
Schädel	2	9	0	**91**	1	7	110	17,3
Thorax	2	10	0	2	**396**	0	410	3,4
Wirbel	2	28	1	6	1	**140**	178	21,3

In Tabelle 6.6 ist die zu Tabelle 6.5 passende 4-NN-Klassifikation als Confusion-Matrix dargestellt. Sie repräsentiert gleichzeitig das beste im Rahmen dieser Diplomarbeit erreichte Ergebnis. Dabei ist der Einfluss der in Gleichung (5.7) eingehenden a-priori-Wahrscheinlichkeit als relative Häufigkeit der auftretenden Klassen deutlich zu erkennen. Die zweite Spalte ist stärker besetzt, da die a-priori-Wahrscheinlichkeit der Klasse 'Extremitäten' bei $p(k_{\mathrm{Ex}}) = \frac{706}{1617} = 0,437$ liegt (siehe Tabelle 4.1). Dies reduziert zwar die Klassifikationsfehlerrate dieser Klasse von $7,6\%$ auf $4,0\%$, führt aber beispielsweise bei der Klasse 'Mamma' zu einer vermehrten Fehlklassifikation auf die Klasse 'Extremitäten', da diese Klasse aufgrund der höchsten a-priori Wahrscheinlichkeit generell präferiert wird.

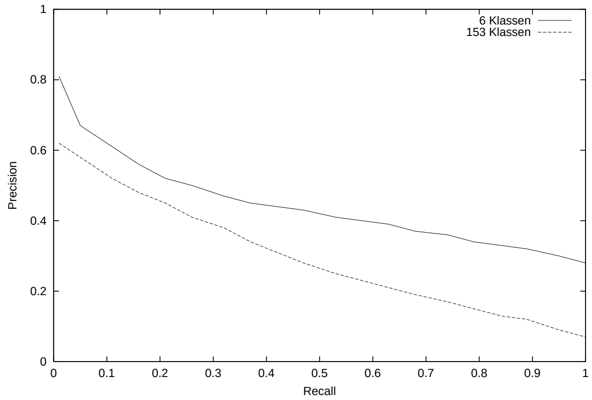

Abbildung 6.5: Precision over Recall

6.6 Precision/Recall-Evaluierung

Auf die enge Verwandtschaft zwischen Klassifikation und Retrieval wurde bereits in Kapitel 1 hingewiesen. Die gängigen Evaluierungsmaße für Retrieval-Systeme wie *Precision* und *Recall* können jedoch aus einem solchen Klassifikationsexperiment nicht gleich ermittelt werden. In [Schä1997] wird jedoch ein Verfahren vorgestellt, das aus einer *leaving-one-out* Testreihe anhand von während der Klassifikation erzeugten Informationen einen sogenannten *Precision over Recall* Graphen erstellen kann (vergleiche auch Abschnitt 1.3). Dazu wird für jedes Testbild eine Liste erzeugt, die (aufsteigend nach der Distanz sortiert) zu dem entsprechenden Referenzbild angibt, ob es zur selben Klasse gehört wie das Testbild, sprich ob es für eine Anfrage an eine Datenbank mit dem Testbild als Anfragebild relevant ist oder nicht. Das Verfahren errechnet daraus (gemittelte) Precision/Recall-Paare. Abbildung 6.5 zeigt zwei solcher Precision-over-Recall Graphen. Im zweiten dargestellten Beispiel wurde die Relevanz bezüglich aller 153 Klassen bestimmt, die sich nach dem in Abschnitt 4.1 vorgestellten 8-stelligen Code unterscheiden und mehr als einen Repräsentanten in der Datensammlung haben. Wünschenswert bei einer Datenbankanfrage ist natürlich grundsätzlich eine möglichst gleichbleibend hohe Precision bei steigendem Recall. Das würde in der dargestellten Grafik bedeuten, dass die Kurve in der rechten oberen Hälfte verliefe. Die dargestellten Grafiken brechen allerdings bei einer zu Beginn ohnehin nicht sehr hohen Precision schon früh ein. Die vorgestellten Klassifikationsverfahren sind also zum Zwecke des *Image Retrieval* nur unter der Bedingung geeignet, dass sie in dem Schritt 'Kategorisierung' innerhalb des IRMA-Systems (vergleiche Abschnitt 3.1) verwendet werden. Dort folgen dann vor dem eigentlichen Retrieval-Schritt noch wichtige Zwischenschritte, die das Anfrage-Resultat verbessern können. Es ist allerdings anzumerken, dass auch andere Retrieval-Verfahren bislang keine optimalen Ergebnisse erzielen konnten. Der Verlauf der Precision-over-Recall Kurve ist typisch für viele Retrieval-Systeme.

6.7 Reject

Bei Klassifikationssystemen wird häufig mit dem sogenannten *Reject* gearbeitet. Dabei werden Bilder, die aufgrund ihrer Distanzen nicht 'deutlich' einer Klasse zugeordnet werden können, vom System zurückgewiesen. Die Fehlerrate lässt sich dann im allgemeinen senken, allerdings sollte die Rejectrate nicht zu groß werden. Diese Rejectrate ist direkt abhängig von einem vorher festzulegenden Reject-Kriterium, etwa einem einfachen Schwellwert, der angibt, wie sehr sich die Distanzen eines Bildes zu den nächstliegenden zwei Klassen (TOP-2) unterscheiden müssen, um sich eindeutig für die TOP-1-Klasse entscheiden zu können. Im Rahmen dieser Diplomarbeit wurden Versuche mit Reject durchgeführt, wobei sich der Schwellwert prozentual auf die jeweils kleinere betrachtete Distanz einer 1-NN-Klassifikation (also auf die zur TOP-1-Klasse) bezog. Abbildung 6.6 zeigt den Verlauf der Fehlerrate und der Rejectrate abhängig von der gewählten Schwelle. Bei einer Schwelle von 6% ergibt sich beispielsweise eine Restfehlerrate von $6,4\%$ bei einer Rejectrate von $5,6\%$. Natürlich sollten möglichst nur solche Bilder vom System zurückgewiesen werden, die ansonsten falsch klassifiziert worden wären. Man will also die Anzahl der Bilder, die vom System zurückgewiesen werden, obwohl sie ursprünglich richtig klassifiziert worden wären, möglichst gering halten. Abbildung 6.7 zeigt, dass diese Anzahl mit zunehmender Schwelle ansteigt. In Abbildung 6.8 ist zu erkennen, dass bei einer Schwelle von 6% nur noch $38,9\%$ der zurückgewiesenen Bilder ursprünglich korrekt klassifiziert worden wären. Strenggenommen könnte man an dieser Stelle von *training on the testing data* sprechen, da der Schwellwert anhand der Testdaten optimiert wurde. Das zeigt allerdings nur die Notwendigkeit einer neuen Stichprobe für einen *blind test* an, mit dem die hier gewonnenen Informationen verfiziert werden könnten. Darauf, und auf die Implementierung von Reject bei der KD-Klassifikation, wird allerdings erst in nachfolgenden Diplomarbeiten eingegangen werden.

Versuche mit Reject sind grundsätzlich kritisch zu bewerten, bieten sie doch den Anschein, die Fehlerrate mit geringem Aufwand zu senken (werden alle Bilder zurückgewiesen, so ist die Restfehlerrate sogar gleich Null). Es muss also streng darauf geachtet werden, dass bei Verwendung

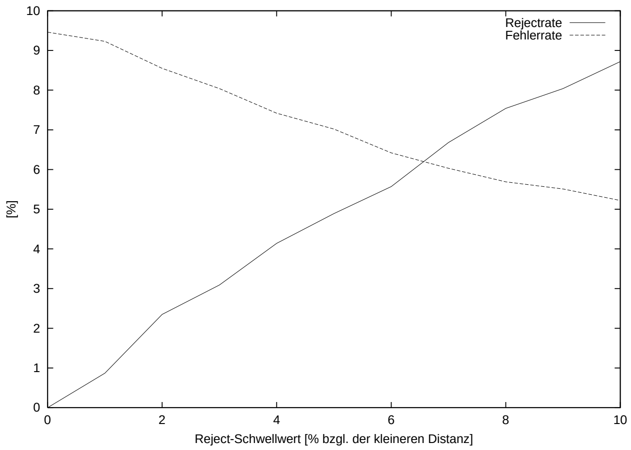

Abbildung 6.6: Verlauf der Fehlerrate bei Reject, abhängig vom Reject-Schwellwert (1-NN-Basisfehlerrate: $9,4\%$)

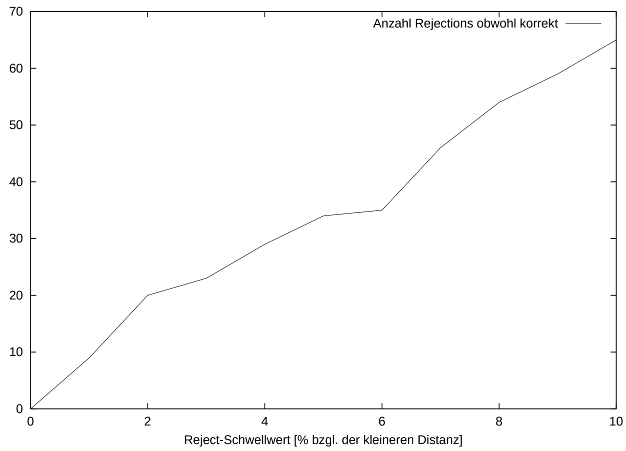

Abbildung 6.7: Anzahl der Bilder, die zurückgewiesen werden, obwohl korrekt klassifiziert

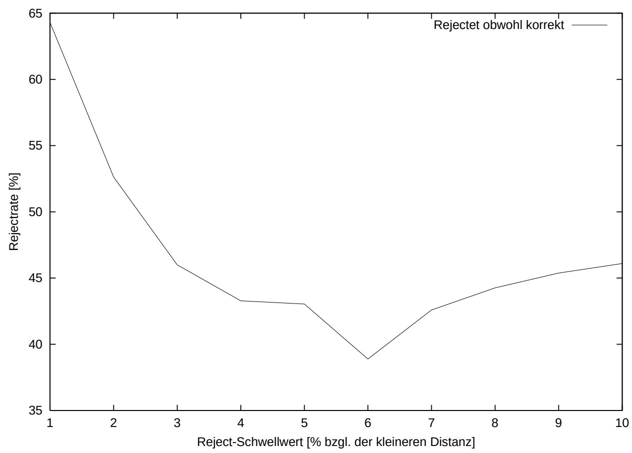

Abbildung 6.8: Anteil der Bilder, die zurückgewiesen werden, obwohl korrekt klassifiziert, an allen zurückgewiesenen Bildern

von Reject neben der Fehlerrate auch die Rejectrate angegeben wird, um das Ergebnis nicht zu verfälschen. Wenn im medizinischen Bereich mit Reject gearbeitet werden soll, so ist es unumgänglich, vom System zurückgewiesene Bilder nicht einfach 'unter den Tisch fallen zu lassen', sondern sie dem Arzt vorzulegen, der dann die Entscheidung treffen kann.

6.8 Weitere Untersuchungen

Zur Berechnung der Tangenten werden zunächst Gradienten des betrachteten Bildes berechnet [Vap1998]. Jedoch stößt man bei der Betrachtung der Pixel am Rand des Bildes auf das Problem, dass nicht alle zur Berechnung benötigten Pixel vorhanden sind. Diese Randpixel müssen also gesondert behandelt werden. Dazu werden im allgemeinen unterschiedliche Ansätze verfolgt (beispielsweise ein *wrap around*). Im Rahmen dieser Arbeit wurde zunächst versucht, für alle nicht vorhandenen Pixel einen Grauwert als Mittelwert vorhandener, angrenzender Pixel, zu verwenden. Auch wurde dieser Grauwert auf einen konstanten Wert gesetzt, dessen 'Optimum' empirisch unter Betrachtung der Fehlerrate ermittelt wurde. Abbildung 6.9 zeigt, dass dieser Wert bei dem verwendeten Bildmaterial etwa bei $g_{\text{opt}} = 142$ liegt. Das beste Ergebnis wurde allerdings erzielt, wenn die Randpixel bei der Berechnung der Tangenten komplett ignoriert wurden. Dabei gehen bei 32×32 Pixel großen Bildern natürlich 124 Pixel verloren. Das entspricht $12,1\%$ der Pixel. Bei einer weiteren Berechnung, die sogar die beiden äußeren Randpixel-Reihen berücksichtigt, sind es mit 244 Pixeln sogar $23,8\%$. Da dort auch relevante Informationen liegen können, ist das nicht wünschenswert. Deshalb wird in einer weiterführenden Diplomarbeit (siehe Kapitel 8) ein anderes Verfahren für die Behandlung der Randpixel vorgestellt.

SIMARD erwähnt in [Sim1993], dass die Berechnung der Tangenten auf hochdimensionalen Bildern dadurch verbessert und ihre Reichweite vergrößert werden könnte, wenn auf diese Bilder vorher ein Gaußfilter angewendet wird. Bei Verwendung eines Gaußfilters erhält jeder Pixel des Bildes einen neuen Grauwert, der sich aus einer bestimmten Kombination der Grauwerte der umgebenden Pixel ergibt. Wieviele Pixel der Umgebung dabei einbezogen werden, ist abhängig von der Dimension der verwendeten Gaußmatrix (3×3, 5×5, 7×7 usw.). Auf den im Rahmen dieser Diplomarbeit benutzten Bildern der Dimension 32×32 ließ sich damit jedoch keine signifikante Verbesserung der Klassifikationsfehlerrate erzielen. Lediglich bei 64×64 zeigte sich bei einer 5×5-Gaußmatrix der Einfluß der Reichweitenvergrößerung, indem mit $11,8\%$ eine Fehlerrate erreicht wurde, die deutlich besser ist als ohne Gaußfilterung ($13,5\%$), und die mit der bei 32×32 ohne Gaußfilter erreichten Fehlerrate vergleichbar ist ($11,7\%$ ohne Parameteroptimierung, eine solche Optimierung lieferte dann die Fehlerraten aus Tabelle 6.2, also eine Reduzierung auf $10,3\%$).

In weiteren Experimenten wurde versucht, die Fehlerrate mittels der Berechnung alternativer Tangenten zu senken (siehe Abschnitt 5.3.1). Dazu wurden neben der bereits verwendeten Helligkeitstangente noch eine Tangente für den Kontrast bzw. zwei Tangenten für die Strichdicke in x- und y-Richtung eingesetzt, die die bisher betrachtete (richtungsunabhängige) Strichdickentangente ersetzten. Der erhoffte Erfolg blieb allerdings jeweils aus. Die Fehlerrate ließ sich zwar nicht senken, stieg allerdings auch nicht deutlich an.

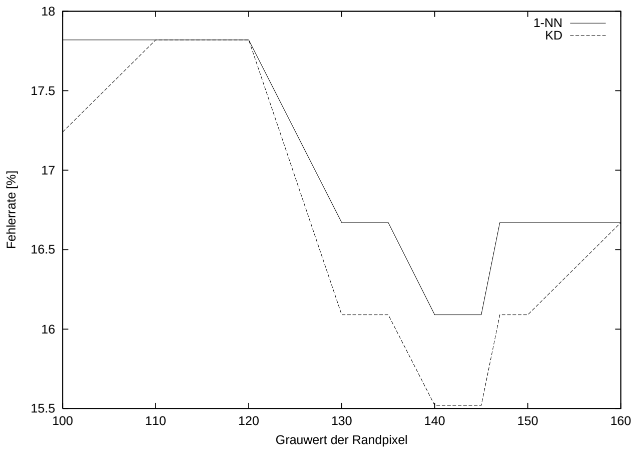

Abbildung 6.9: Fehlerraten in Abhängigkeit vom betrachteten Randgrauwert

Das in Abschnitt 5.3.5 angesprochene Trainieren der Deformations-Kostenfunktion wurde im Rahmen dieser Diplomarbeit nicht durchgeführt. Dazu (wie auch für weitere Experimente, beispielsweise die Einbeziehung von Gradientenbildern) sei auf nachfolgende Diplomarbeiten verwiesen (siehe Kapitel 8).

Kapitel 7

Cooccurrence-Matrizen

Um eine möglichst umfassende Information über die stochastische Struktur der Texturen eines vorliegenden Graustufen-Bildes zu erhalten, wird in [Har1973] eine Form von Verbundverteilungen betrachtet, die Aussagen über die Wahrscheinlichkeit liefert, mit der zwei Punkte mit definiertem Abstand bestimmte Grauwerte annehmen. Dabei wird die *Cooccurrence-Matrix* als Beschreibung der Textur gewählt.

Man betrachtet einen Punkt x_{ij} im zugrundeliegenden Bild x. Dieser habe den Grauwert $g \in G$. Nun werden alle Punkte betrachtet, die von x_{ij} den gleichen Abstand besitzen (z.B. seine direkten Nachbarn) und deren Grauwerte bestimmt. Besitzt nun beispielsweise einer dieser 'Nachbarpunkte' den Grauwert $h \in G$, so wird in einer Matrix der Dimension $|G| \times |G|$ der Eintrag an der Stelle (g,h) erhöht. Damit liefert diese Matrix nach Betrachtung aller Bildpunkte und anschließender Normierung die Wahrscheinlichkeiten, mit der bestimmte Grauwerte an 'benachbarten' Positionen im Bild auftreten. Formal kann man diesen Sachverhalt wie folgt beschreiben:

Das zweidimensionale Histogramm $H_{\bar{d}} \in \mathbb{R}^{|G|\times|G|}$ mit Abstandsvektor $\bar{d} = (\Delta x, \Delta y)$ ist definiert als

$$H_{\bar{d}}(g,h) = \sum_{i=1}^{I} \sum_{j=1}^{J} \delta(x_{ij}, g) \cdot \delta(x_{i+\Delta x, j+\Delta y}, h), \tag{7.1}$$

wobei $g, h \in G$ zwei Grauwerte angeben und δ das Kroneckersymbol ist. Seien nun (Δ,α) die Polarkoordinaten des Abstandsvektors $\bar{d}$, so ist die Cooccurrence-Matrix definiert durch:

$$C_\Delta(g,h) = \frac{1}{n_\Delta} \sum_{\alpha} H_{\Delta,\alpha}(g,h) \tag{7.2}$$

Dabei bezeichnet n_Δ die Anzahl der einbezogenen Punkte. Der festgelegte Abstand $\Delta = |\bar{d}|$ wird auch als *Displacement* bezeichnet. Man muss bei dieser Berechnung beachten, dass in einem diskreten Bild die betrachteten Punkte für Winkelgrößen $\alpha \notin \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ keinen Kreis mit Radius Δ beschreiben. Daher ist in diesen Fällen der entsprechende Kreis pixelweise anzunähern. Dies ist beispielsweise mit dem Algorithmus von Bresenham möglich [Doh1991].

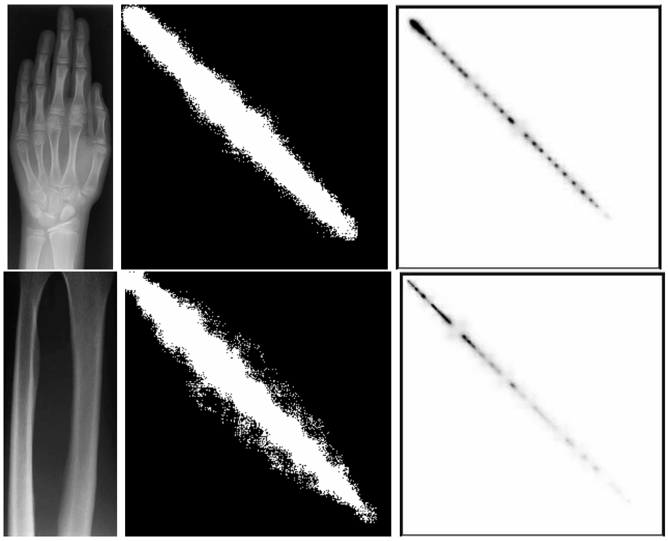

Abbildung 7.1: Beispiele für Cooccurrence-Matrizen in unterschiedlicher Darstellung

In einer im Rahmen des IRMA-Projektes durchgeführten Studienarbeit ([Bon1999]) findet die wohl häufigste Form der Cooccurrence-Matrix Verwendung, die mit einem Displacement der Länge 1 arbeitet. Dementsprechend werden die Histogramme H_{rechts} (0°), $H_{\text{rechts-oben}}$ (45°), H_{oben} (90°) und $H_{\text{links-oben}}$ (135°) berechnet, summiert und gemittelt (das ergibt die Abstandsvektoren (1,0), (1,-1), (0,-1) und (-1,-1)). Die anderen Werte ergeben sich entsprechend der offensichtlichen Symmetrie-Eigenschaft $H_{\bar{d}} = H_{-\bar{d}}$. Somit erhält man folgende Beschreibung:

$$C_1(g,h) = \frac{1}{4}(H_{\text{rechts}} + H_{\text{rechts-oben}} + H_{\text{oben}} + H_{\text{links-oben}}) \tag{7.3}$$

Abbildung 7.1 stellt beispielhaft C_1-Cooccurrence-Matrizen dar. In der ersten Spalte sind die Originalbilder zu sehen, wobei das untere Bild eine Radiographie von Elle und Speiche darstellt. Die zweite Spalte zeigt die Cooccurrence-Matrizen als binarisierte Darstellungen, d.h. alle Matrixeinträge ungleich Null wurden mit einem weißen Pixel, Werte gleich Null mit einem schwarzen Pixel dargestellt. In der dritten Spalte wurden schließlich den unterschiedlichen Häufigkeiten Grauwerte zwischen 0 und 255 zugeordnet (zur besseren Darstellung hier invertiert).

Es fällt auf, dass die Hauptdiagonale besonders stark besetzt ist. Dies resultiert daraus, dass benachbarte Punkte bei Displacement 1 in ihren Grauwerten meist nur geringe Unterschiede aufweisen. Die eigentliche Texturbeschaffenheit des Bildes wird also durch die Einträge in der Cooccurrence-Matrix charakterisiert, die abseits der Hauptdiagonalen liegen. Natürlich beinhaltet auch die Hauptdiagonale relevante Bildinformationen, denn auch in stark texturierten Strukturen wie Knochen und Weichgewebe existieren benachbarte Bildpunkte mit annähernd gleichen Grauwerten.

In [Bon1999] wurde ein auf [Hak1991] basierender synergetischer Klassifikator herangezogen. Dabei werden aus den Trainingsdaten ein oder mehrere Prototypen pro Klasse manuell ausgewählt und deren Cooccurrence-Matrizen extrahiert. Diese bilden nun durch 'Aneinanderreihen' der Zeilen jeweils einen Prototypvektor. Zu einer Klasse gehörende Prototypvektoren werden dann unter Berücksichtigung von Nebenbedingungen (z.B. Orthogonalität, Mittelwertfreiheit) zu einem einzigen Prototypvektor zusammengefasst (aus dem sich wiederum die sogenannte *Adjungierte Cooccurrence-Matrix* bilden lässt). Schließlich werden die Cooccurrence-Matrizen aller Testbilder (exklusive der gewählten Prototypen) mit den adjungierten Prototypvektoren der unterschiedlichen Klassen mittels des Skalarproduktes oder auch des euklidischen Abstandes verglichen. Das Skalarprodukt beschreibt aufgrund der vorhergehenden Normierungen den Winkel zwischen zwei Prototypvektoren. Es gibt somit ein Maß dafür an, inwieweit die Vektoren in die gleiche Richtung zeigen. Auf den vorgestellten sechs Grobklassen Abdomen, Extremitäten, Mamma, Schädel, Thorax und Wirbel ergab sich bei Verwendung der euklidischen Distanz dabei eine Klassifikationsfehlerrate von $71,7\%$ auf 272 sekundär digitalisierten Röntgenbildern [Bon1999]. In der hier vorliegenden Diplomarbeit wurde testweise eine Klassifikation basierend auf einem 1-NN-Klassifikator mit *leaving one out* durchgeführt, indem die Cooccurrence-Matrix des jeweiligen Testbildes mit den Cooccurrence-Matrizen der Referenzbilder verglichen wurde. Der 'Umweg' über adjungierte Prototypen wurde demnach vermieden. Alle Bilder wurden vorher auf eine einheitliche Größe skaliert. Dabei ergab sich bei einer Skalierung auf 320×320 mit $29,4\%$ auf den 2080 Bildern der Datensammlung (s. Kapitel 4) eine Klassifikationsfehlerrate, die die des synergetischen Klassifikators deutlich verbesserte. Das kann bedeuten, dass beispielsweise die Prototypen ungeschickt gewählt wurden, oder dass dieser Klassifikator für diese Problemstellung grundsätzlich ungeeignet ist. Eine genauere Untersuchung steht allerdings noch aus (siehe Kapitel 8).

Diese Fehlerraten lassen im Vergleich zu den in Kapitel 6 erreichten vermuten, dass sich Cooccurrence-Matrizen nicht sinnvoll in der Klassifikation von Röntgenbildern einsetzen lassen. Allerdings ist zu beachten, dass bisher nur globale Cooccurrence-Matrizen und damit globale Texturmerkmale extrahiert wurden. Daher könnte es sich für die nachfolgenden Schritte des IRMA-Systems als sinnvoll erweisen, wenn stattdessen die Cooccurrence-Matrizen nur für kleinere Bildbereiche mit einheitlichen und diskriminativen Texturmerkmalen berechnet werden, etwa zur Tumordetektion (siehe dazu auch [Vog1997]). Auch kann es durch die vorhergehende Skalierung der Bilder zu Stauchungen im Bildmaterial kommen. Daher erscheint es sinnvoll, neben den C_1-Matrizen auch Matrizen mit größeren Displacements zu verwenden, um dies zu kompensieren.

Bei der Verwendung von Cooccurrence-Matrizen ergeben sich mit $256 \cdot 256 = 65536$ Einträgen sehr hochdimensionale Merkmalsvektoren. Daher schlägt Haralick in [Har1973] vor, daraus zunächst 14 Texturmerkmale (wie Kontrast, Korrelation, Varianz oder Entropie) zu extrahieren und nur diese zur Klassifikation heranzuziehen. Darauf wurde allerdings im Rahmen des IRMA-Projektes bislang nicht eingegangen. Es macht allerdings deutlich, dass neben den Cooccurrence-Matrizen noch weitere Texturdeskriptoren existieren.

Kapitel 8

Zusammenfassung und Ausblick

In dieser Diplomarbeit wurden Verfahren vorgestellt, die zur Röntgenbild-Klassifikation innerhalb des IRMA-Systems verwendet werden und einen ersten wichtigen Schritt für die Realisierung eines Images Retrieval Systems für medizinische Bilddaten darstellen. Es wurde das *Image Distortion Modell* eingeführt, dessen Effizienz in Experimenten auf den vorliegenden 1617 Röntgenbildern aus dem 'medizinischen Alltag' bewiesen wurde. Obwohl schon dieses Modell alleine recht effektiv einsetzbar ist, wurde gezeigt, dass es in dieser Anwendung zur Verbesserung von Simards *Tangentendistanz* herangezogen werden kann. Die beste (KD-) Klassifikationsfehlerrate von $8,6\%$ wurde durch eben diese Kombination erzielt. Man beachte, dass es sich dabei um eine relative Verbesserung von 52% bezüglich der Verwendung der euklidischen Distanz bei einer 1-NN-Klassifikation (Fehlerrate $18,1\%$) handelt.

Zur Zeit wird in nachfolgenden Diplomarbeiten an Verbesserungen und Erweiterungen des Image Distortion Modells gearbeitet, indem beispielsweise die Kostenfunktion $C(i,i',j,j')$ durch Lernen aus den Trainingsdaten sinnvoller gewählt wird. Weiterhin erscheint es noch vielversprechend, zusätzlich Gradientenbilder der betrachteten Bilder zu berechnen, um Kantenverläufe besser beschreiben zu können. Das in Kapitel 6 angesprochene Problem bei der Betrachtung von Randpixeln in der Tangentenberechnung wird ebenfalls noch zu behandeln sein.

Interessant ist sicherlich auch, die Auswirkungen des Image Distortion Modells anhand von Beispielen zu visualisieren. So könnte man erkennen, wie ein Referenzbild auf ein Testbild 'reagiert', d.h. welche Deformationen ausgewählt werden, um die Distanz letztendlich zu minimieren. Möglicherweise ist dabei ein Muster erkennbar, das man dann beispielsweise zur Datenvervielfältigung einsetzen kann.

Bei der Verwendung des Image Distortion Modells ist es denkbar, dass ein Pixel $\mu_{i'j'}$ des aktuellen Referenzbildes verhältnismäßig oft ausgewählt wird (bei Umgebung $r = 1$ bis zu 9mal, bei Umgebung $r = 2$ schon bis zu 25mal). Dies könnte neben der bereits vorgestellten Kostenfunktion $C(i,i',j,j')$ beispielsweise dadurch verringert werden, dass ein Zähler für jeden Pixel mitgeführt wird, der eine bestimmte Anzahl nicht überschreiten darf, bzw. der die Kostenfunktion erweitert.

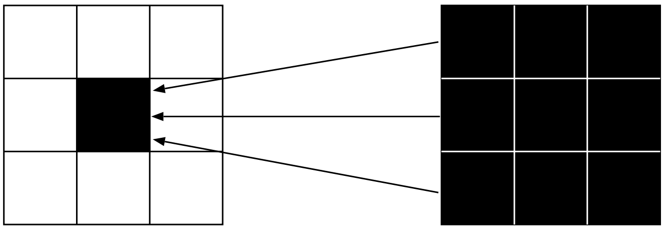

Abbildung 8.1: Ein Extremfall für das Image Distortion Modell

Je größer der Zähler, desto größer die Bestrafung. Dadurch wird der Extremfall vermieden, der in Abbildung 8.1 zu sehen ist. Alle Pixel des rechten Bereichs werden auf den zentralen Pixel im linken Bereich abgebildet. Bei Einführung der Zähler-Bestrafung würde dieser zentrale Pixel nicht mehr unbegrenzt herangezogen werden können.

Bereits bei der Vorstellung des Datenmaterials in Kapitel 4 wurde auf den sogenannten *Scribor* eingegangen, der unter anderem Patienten-Informationen enthält und zum Zwecke der Anonymisierung für die Experimente dieser Diplomarbeit durch schwarze Balken entfernt wurde. Für den Praxis-Einsatz der vorgestellten Verfahren ist es allerdings denkbar, diese Informationen zu nutzen, indem sie beispielsweise in einem zusätzlichen Verarbeitungsschritt (*Optical Character Recognition*, OCR) ausgelesen werden.

Die in Kapitel 6 angesprochenen Versuche mit Reject wurden bislang nur für eine 1-NN-Klassifikation durchgefürt. Die Implementierung und Anpassung für KD-Klassifikationen steht noch aus. Auch wurde das Verfahren bisher nur anhand der vorliegenden Daten optimiert. Daher werden weitere Stichproben für einen *blind test* benötigt.

Der statistische Standard-Ansatz, Modelle für die in der Bayes'schen Entscheidungsregel auftretenden klassenbedingten Wahrscheinlichkeiten bzw. a-priori Wahrscheinlichkeiten zu wählen und deren Parameter dann anhand der Trainingsdaten zu schätzen, scheiterte bisher an der Tatsache, dass mit 1617 Bildern, die dazu noch sehr hochdimensional sind, zu wenig Datenmaterial zur Verfügung stand. Dies kann zum Beispiel dadurch umgangen werden, dass Invarianzen bereits im Bildmaterial berücksichtigt werden. Dazu werden bestimmte Transformationen, die die Klassenzugehörigkeit nicht beeinflussen dürfen, auf alle Trainingsdaten (und evtl. auch Testdaten) angewendet, und die dabei entstehenden Bilder mit in den Datenbestand aufgenommen. Diese Erzeugung *virtueller Daten* hat in anderen Bereichen, wie zum Beispiel der Ziffernerkennung, schon zu deutlichen Verbesserungen in der Klassifikation geführt. Die Tangentendistanz deckt zwar kleinere globale Transformationen ab, kann aber offenbar durch die Erzeugung virtueller Daten noch erweitert werden.

Das Problem der Hochdimensionalität der Merkmalsvektoren kann durch eine Dimensionsreduktion umgangen werden. Zur Zeit laufen in einer nachfolgenden Diplomarbeit Versuche, eine Dimensionsreduktion zusammen mit der hier bereits vorgestellten Tangentendistanz einzusetzen.

Ein weiterreichendes Ziel wird sicherlich sein, die bisher durchgeführte Grobklassifikation zu verfeinern. Dazu müssen angepasste Algorithmen entwickelt werden, die auf den hier vorgestellten basieren. Dazu kann das in Kapitel 3 vorgestellte IRMA-System als Grundlage dienen, das eine Hierarchie im Zusammenhang der unterschiedlichen Bearbeitungsstufen (von Kategorisierung bis Retrieval) aufzeigt. So ist beispielsweise das hier vorgestellte Verfahren, das Cooccurrence-Matrizen verwendet, in der Stufe 'Kategorisierung' nur eingeschränkt verwendungsfähig. Es ist allerdings durchaus denkbar, dass durch die Verfeinerungen in den nachfolgenden IRMA-Stufen (beispielsweise eine automatische Segmentierung in *Regions of Interest* (ROI)) dieses Verfahren an Bedeutung gewinnt und zum erfolgreicheren Retrieval eingesetzt werden kann.

Kapitel 9

Anhang
Programme und Parameter

9.1 Die Anwendung IRMA

Die Anwendung IRMA, die auf dem Datenbanksystem RETRIEVAL basiert, wurde in Kapitel 3 vorgestellt. Sie dient zur Darstellung des Anfragebildes, zur Auswahl des passenden Extraktionsalgorithmus' und schließlich zum Starten der Merkmalsextraktion. Nach der Ausführung des (externen) Programms zur Merkmalsextraktion werden die besten Ergebnisse aufgelistet. Falls das Anfragebild noch nicht in der Datenbank ist, wird zunächst dessen Merkmalsvektor bestimmt und in der Datenbank abgelegt. Danach wird es mit allen passenden Bildern der Datenbank anhand eines in der Datenbank eingetragenen Distanzmaßes miteinander verglichen.

Die Datenbank selbst muss dafür zunächst natürlich aktiv sein. Dazu muss der sogenannte *postmaster* auf einem Rechner gestartet werden (Aufruf: `nohup postmaster -i &`). Danach muss die Datenbank RETRIEVAL entsprechend vorbereitet werden, indem alle benötigten Bilder, Algorithmen und Distanzmaße eingetragen werden. Dazu existieren die passenden Tools unter

`/u/IRMA/Theiner/retrieval/Tools/`

9.1.1 InsertImages, InsertExtraction und die Tabelle 'qbe'

`InsertImages` erwartet als Parameter die Bilder, die in die Datenbank eingefügt werden sollen. Ein Beispielaufruf lautet also wie folgt (bei Verwendung von *Wildcards* ist unbedingt auf die Anführungszeichen zu achten!):

```
InsertImages "/u/IRMA/Theiner/images/corpus/medical/scale32/*.jpg"
```

Alle notwendigen Daten werden daraufhin in die Datenbank eingetragen. Dazu gehören der Pfad zu den jeweiligen Dateien, der Dateiname selbst und die Eigenschaften der einzelnen Bilder (Dimension, Farbtiefe usw.). Ein entsprechendes Tool existiert für die Einbindung von Extraktionsalgorithmen in Form von ausfürbaren Dateien. Dazu muss zunächst unter

`/u/IRMA/Theiner/retrieval/Extraction`

ein neues Verzeichnis mit dem Namen des Executables angelegt werden. In diesem liegt dann das Executable mit dem gleichen Namen. Der Eintrag in die Datenbank geschieht folgendermaßen über das Tool `InsertExtraction`:

```
InsertExtraction I:Extraktionsname I:Executable [I:-p PARAMETER]
[I:-ep OPTPARAMETER] [I:-bpv BytesPerValue] [I:-vpp ValuesPerPixel]
[I:-dim Bilddimension] [I:-cf #] [I:-ci #] [I:-cs #]

Beispielaufruf:
InsertExtraction GreyHIT GreyHIT -bpv 1 -vpp 1 -cf 256
```

Der 'Extraktionsname' gibt an, wie der Eintrag in der Datenbank bezeichnet werden soll. 'Executable' ist der *genaue* Name des Executables und damit auch des Verzeichnisnamens, in dem dieses Executable liegt. Mit '-p' und '-ep' können verlangte bzw. optionale Parameter des Executables angegeben werden. Dort verwendete Unterstriche '`_`' werden beim späteren Aufruf des Executables durch Bindestriche '`-`' ersetzt. Wichtig sind die Angaben, mit welchem Bildmaterial der Algorithmus umgehen kann. Diese werden durch die optionalen Parameter '-bpv', '-vpp' und '-dim' angegeben. Die Anzahl und Art der Merkmalseinträge, die der Algorithmus als Ausgabe erzeugt, werden mit den letzten drei Parametern '-cf' (für *Float*), '-ci' (*Integer*) und '-cs' (*String*) festgelegt. Informationen zum inneren Aufbau der Executables können den Beispielen unter

`/u/IRMA/Theiner/retrieval/Extraction/`

oder der Programmdokumentation von *Jörg Bredno* entnommen werden:

`/u/IRMA/Theiner/retrieval/RetrDoc.ps`

Der letzte Vorbereitungsschritt, nämlich das Festlegen des zu verwendenden Distanzmaßes geschieht direkt in der Datenbank über die *postgres*-Console `psql` (Aufruf: `psql retrieval`) auf dem Rechner, auf dem der *postmaster* läuft. Dies ist ein Ausschnitt aus einer solchen Consolen-Sitzung:

```
palladium:retrieval # psql retrieval
Welcome to the POSTGRESQL interactive sql monitor:
  Please read the file COPYRIGHT for copyright terms of POSTGRESQL

   type \? for help on slash commands
   type \q to quit
   type \g or terminate with semicolon to execute query
 You are currently connected to the database: retrieval
```

```
retrieval=> select * from qbe;
extractor|distance|sort|threshbelow|threshabove
---------+--------+----+-----------+-----------
     1701|       1|   2|          1|       0.49
   167301|       1|   3|         20|          0
    16101|       1|   3|         20|          0

retrieval=> select * from extraction where (id=167301);
name   |executable|param|extraparam|vector      |    id
-------+----------+-----+----------+------------+------
GreyHIT|GreyHIT   |     |          |f:256,i:0,s:0|167301
```

`psql` akzeptiert direkt SQL-Befehle. Die erste Tabelle 'qbe' gibt an, zu welchem Extraktionsalgorithmus welches Distanzmaß verwendet wird. Der Extraktionsalgorithmus wird dabei über seine datenbankinterne ID angesprochen. Die untere Tabelle 'extraction' liefert dann die näheren Angaben zu diesem Algorithmus. Bislang ist als Distanzmaß die 'Histogram Intersection' (siehe Abschnitt 3.4) mit dem Eintrag '1' in der Spalte 'distance' realisiert. Die Spalte 'sort' gibt an, in welcher Reihenfolge die Daten bei der Ausgabe sortiert werden sollen. Dabei entspricht der Eintrag '1' einer aufsteigenden und '2' einer absteigenden Sortierung nach der Distanz. Die Ober- und Untergrenze der zu berücksichtigenden Distanzen sind in den Spalten 'threshbelow' und 'threshabove' angegeben. Der Eintrag '3' in der Spalte 'sort' gibt ebenfalls eine absteigende Sortierung nach der Distanz vor, allerdings werden diesmal keine Ober- und Untergrenzen festgelegt, sondern eine prozentuale Schwelle unter 'threshbelow'. In diesem Fall werden nur diejenigen Bilder berücksichtigt, die nicht mehr als 20% von der bestplatzierten Distanz abweichen.

Um jetzt eine Merkmalsextraktion durchzuführen, muss das passende Executable mit den entsprechenden, vorher bei *InsertExtraction* festgelegten Parametern aufgerufen werden (siehe Tabelle 'extraction' im obigen Beispiel). Dabei spielt die Tabelle 'worklist' in der Datenbank noch eine entscheidende Rolle. Sie gibt an, welche Bilder der Datenbank behandelt werden sollen:

```
retrieval=> select * from worklist;
extractor| image
---------+------
   167301|     0
```

Ein Eintrag '0' in der Spalte 'image' zeigt an, dass von *allen* passenden Bildern Merkmale extrahiert werden sollen, wenn sie nicht schon in der Datenbank existieren. Die Tabelle 'feature' in der Datenbank gibt dazu an, zu welchem Algorithmus/Bild-Paar bereits Merkmale extrahiert wurden. Ist nun der entsprechende Eintrag in der Tabelle 'worklist' vorgenommen worden, so kann die Extraktion gestartet werden:

```
Executable I:-id EXTRACTOR-ID I:PARAMETER [I:OPTPARAMETER]
```

Beispielsweise startet der Aufruf `GreyHIT -id 167301` mit obiger Worklist die Extraktion von Merkmalen aus allen passenden Bildern (in diesem Fall Bilder mit 256 Graustufen).

Die Anwendung IRMA setzt nun auf einer nach den oben genannten Schritten vorbereiteten Datenbank auf. Sie wird als

`/u/IRMA/Theiner/retrieval/Tools/IRMA`

aufgerufen und präsentiert sich wie unter Abschnitt 3.3 beschrieben. Die Kernroutine ist dabei das Öffnen und Darstellen von JPG-Dateien, das unter Zuhilfenahme der Libraries *libjpeg.so* und *Xlib.so* implementiert wurde. Da auf den verwendeten Rechnersystemen nur 256 Farben für die Darstellung zur Verfügung standen, wurde ein Clusterverfahren in

`/u/IRMA/Theiner/retrieval/Tools/Create_Pseudoclasses_jf_tt.c`

implementiert, das die Farben des Bildes auf maximal 256 Pseudoklassen clustert, die jeweils einen Farbton im Bild repräsentieren. Der Aufruf dieses Teilprogrammes geschieht bei Bedarf transparent direkt aus der Anwendung IRMA heraus.

9.2 Tangenten, IDM und Kombinationen

Die zentrale im Rahmen dieser Diplomarbeit implementierte Anwendung ist in

`/u/IRMA/Theiner/retrieval/Extraction/NearestNeighbor/nn_tangent.c`

realisiert und beinhaltet alle getesteten Verfahren. Der Aufruf muss mit den folgenden Parametern durchgeführt werden:

```
Usage: nn_tangent <I:l1o-data-file> <I:Varianzfaktor> <I:Distortion> <I:Tangenten>
<I:Helligkeit> <I:Strafe> <O:ID-File-Name> <I:UseGauss> <I:GaussLevel>
<I:Distortionfaktor> <I:AspectFaktor> <I:m-NN> <I:Reject> <I:Rejectschwelle%>
<I:m-Rejectschwelle%>
```

```
Varianzfaktor    : DOUBLE (nur wichtig bei Tangenten=1)
Distortion       : DOUBLE
Tangenten        : 0 NEIN
                   1 JA
Helligkeit       : 0 NEIN
                 : 1 JA
Max. Strafe      : DOUBLE
ID-File-Name     : CHAR*
UseGauss         : 0 NEIN
                 : 1 Nur Tangenten auf Gauss berechnen
                 : 2 Tangenten auf gaussgefaltetes Originalbild addieren
GaussLevel       : INT (ungerade)  Verwende (GaussLevel x GaussLevel)-Matrix
```

```
Distortionfaktor : DOUBLE
AspectFaktor     : DOUBLE
m-NN             : INT Wievielter m-NN soll besonders betrachtet werden?
Reject           : 0 NEIN
                 : 1 JA
Rejectschwelle%  : DOUBLE Schwellwert, bezogen auf die jew. kleinere Distanz
m-Rejectschwelle%: DOUBLE Dasselbe fuer m-NN
```

9.2.1 Beschreibung der Parameter

- l1o-data-file: Datei mit den Bildern im 'i6-ASCII-Format' (siehe Abschnitt 9.5), dabei sind nur Klassen von 0 bis 5 erlaubt! Die Dateinamen der verwendeten Daten stehen in `/u/IRMA/Theiner/images/corpus/medical/scale32/X-Filenames.txt`
- Varianzfaktor: Beschreibt den Faktor γ aus Gleichung (5.7)
- Distortion: Gibt den Wert r an, der die Größe der Umgebung $R_{ij} = (2r+1) \times (2r+1)$ (siehe Abschnitt 5.3.4 und Kapitel 6) bestimmt
- Tangenten: Sollen Tangenten berechnet werden?
- Helligkeit: Soll statt der Strichdicken-Tangente die Tangente für Helligkeit verwendet werden?
- Max. Strafe: Entspricht dem Clipping-Schwellwert S aus Abschnitt 6.2
- ID-File-Name: Filename für spätere Precision/Recall-Untersuchungen (Output!)
- UseGauss: Soll ein Gaußfilter verwendet werden? Wenn ja, auf welche Weise?
- GaussLevel: *Beispiel:* GaussLevel=3 bedeutet, dass eine 3×3-Gaußmatrix verwendet werden soll
- Distortionfaktor: Wichtungsfaktor für die Kostenfunktion $C(i, i', j, j')$ (siehe Abschnitt 5.3.5)
- AspectFaktor: Wichtungsfaktor ν für die Bestrafung der Unterschiede in der *Aspect Ratio* (siehe Abschnitt 6.4)
- m-NN: Für welches m soll nach der KD-Klassifikation eine Confusion-Matrix (siehe Abschnitt 6.4) ausgegeben werden?
- Reject: Soll mit *Reject* gearbeitet werden? (Abschnitt 6.7)
- Rejectschwelle%: Gibt prozentual an, wie weit die zweitplatzierte Distanz von der erstplatzierten mindestens abweichen muss, damit das Bild nicht vom System zurückgewiesen wird
- m-Rejectschwelle%: Dieselbe Angabe für den betrachteten m-NN in der KD-Klassifikation

Zu beachten ist, dass die Ausgabe in das File 'ID-File-Name' nur dann Sinn macht, wenn im Quelltext vorher die Konstante `NN` mittels `#define` auf die Gesamtanzahl aller Referenzbilder gesetzt wurde (= Anzahl aller Bilder -1), weil nur dann anhand der Positionsnummer eines Referenzbildes dessen Dateiname ermittelt werden kann. Das beste Ergebnis von 8,6% in der KD-Klassifikation mit $m = 4$ wurde mit dem folgenden Aufruf erzielt:

```
/u/IRMA/Theiner/images/corpus/medical/scale32 # nn_tangent X-Files-neu.joerg 1.0
0.8 1 1 3700.0 test.tmp 0 0 50.0 1220.0 4 0 0.0 0.0
```

9.2.2 Precision/Recall-Evaluierung

Für die in Abschnitt 6.6 angesprochene Precision/Recall-Evaluierung der durchgeführten Klassifikationen wird auf das File ’ID-File-Name’ zurückgegriffen. Die folgenden beiden Programmaufrufe erzeugen auf STDOUT ein Datenfile mit Recall/Precision-Wertpaaren, das direkt von *gnuplot* weiterverwendet werden kann:

```
programm_b <I:ID-File-Name> <O:precision/recall-File>
eval_precrec <I:precision/recall-File> <I:AnzahlTestdaten>
```

Das zwischendurch erzeugte Precision/Recall-File beinhaltet zu jedem Testbild eine Liste von ’0’ oder ’1’ sortiert nach Distanzen zu dem jeweiligen Referenzbild. Dabei gibt eine ’1’ an, dass das Referenzbild zur selben Klasse wie das Testbild gehört, demnach also für eine Retrieval-Anfrage relevant wäre, und eine ’0’, dass es nicht relevant wäre. Beide Programme befinden sich unter

```
/u/IRMA/Theiner/retrieval/Extraction/NearestNeighbor
```

9.3 Weitere Tools

`ConvertToJoerg` erzeugt aus einer Liste von JPG-Bildnamen in dem File ’ConvertToJoerg.tmp’, die der von der Klinik für Radiologische Diagnostik eingeführten Namenskonvention entsprechen müssen, ein File im ’i6-ASCII-Format’ (siehe Abschnitt 9.5) mit den Klassen 0 bis 5:

```
/u/IRMA/Theiner/images/corpus/medical/hand/ConvertToJoerg O:JOERGFILE I:BREITE
I:PFAD/
```

’JOERGFILE’ ist dabei der zu erzeugende Dateiname, die Endung ’.joerg’ wird automatisch angehängt. ’BREITE’ ist die einheitliche Breite und Höhe des Bildmaterials, und ’PFAD/’ ist das Verzeichnis, in das das resultierende File ausgegeben werden soll. Dabei ist unbedingt auf das abschliessende ’/’ zu achten!

Die Bilder werden meist im ’PGM-Format’ zur Verfügung gestellt. Mittels eines bash-Scripts, das den folgenden Code enthält, können daraus JPG-Bilder generiert werden:

```
for f in *.pgm ; do
  echo $f > ~/hilf.txt
  hilf=‘sed s/.pgm//g < ~/hilf.txt‘
  echo "Converting $hilf.pgm to $hilf.jpg..."
  convert -colorspace GRAY -quality 85 -geometry 32x32! $hilf.pgm $hilf.jpg
done
```

Das Tool `Get_AspectRatio` erzeugt aus einer Liste von JPG-Dateien im File 'AspectRatio.tmp' eine Datei 'AspectRatios.txt', die neben den Dateinamen noch das Seitenverhältnis der Bilder enthält. Es befindet sich unter

```
/u/IRMA/Theiner/retrieval/Extraction/NearestNeighbor
```

Dieses Programm macht nur Sinn, wenn es in einem Verzeichnis ausgeführt wird, in dem die Bilder in Original-Größe vorliegen. Die resultierende Datei 'AspectRatios.txt' ist zwingend zur Ausführung des Programms `nn_tangent` erforderlich und muss daher vor dessen Aufruf in das entsprechende Verzeichnis kopiert worden sein!

Das Programm `nn_tangent` liefert eine LOG-Ausgabe auf STDOUT. Die falsch klassifizierten Bilder der dort über den Parameter 'm-NN' besonders betrachteten KD-Klassifikation können wie folgt visualisiert werden:

```
/u/IRMA/Theiner/images/corpus/medical/scale32/fehlerbilder <I:nn_tangent_logfile>
>makeindexhtml.tmp
```

```
/u/IRMA/Theiner/images/corpus/medical/scale32/makeindexhtml
```

Danach existiert im aktuellen Verzeichnis eine Datei 'index.html', die beispielsweise mit *Netscape* betrachtet und ausgedruckt werden kann.

Das Tool `Covariance_for_class` berechnet zu einer Liste von JPG-Dateien im File 'Covarianceforclass.tmp' das Mittelwertbild und die (als diagonal angenommene) Kovarianzmatrix:

```
/u/IRMA/Theiner/retrieval/Extraction/NearestNeighbor/Covariance_for_class I:BREITE
O:dateiname
dateiname
```

Das Mittelwertbild wird als PGM-Datei '*dateiname*' ausgegeben, die Kovarianzmatrix wird in eine einfache Textdatei mit dem Namen '*dateiname*.covariance' geschrieben, in der die Diagonaleinträge zeilenweise hintereinander eingetragen werden. Das Bildmaterial muss einheitlich in den Ausmaßen 'BREITE $\times$ BREITE' vorliegen.

Schließlich erzeugt das Tool `CoocView` eine Visualisierungsmöglichkeit für C_1-Cooccurrence-Matrizen, wie sie etwa in Abbildung 7.1 jeweils rechts zu sehen ist:

```
/u/IRMA/Theiner/retrieval/Extraction/NearestNeighbor/CoocView I:dateiname.jpg
O:SCHWELLE
```

Der Float-Wert 'SCHWELLE' dient dabei zur Einschränkung des Wertebereichs und damit der Kontrasterhöhung im resultierenden Bild '*dateiname*.jpg.cooc'. Eine Visualisierung ist einfach über das *KHOROS*-Tool `editimage -i` möglich.

9.4 Beispiel

Die Vorgehensweise bei der Durchführung einer Klassifikation von neu eingegangenen Dateien wird im folgenden anhand eines Beispiels verdeutlicht, das von der Konvertierung der Originaldaten bis zur Visualisierung der Fehlerbilder reicht. Zusätzlich wird noch erläutert, wie Mittelwertbilder, Kovarianz- und Cooccurrence-Matrizen erzeugt werden können. Dieses Beispiel ist nachvollziehbar unter:

`/u/IRMA/Theiner/images/corpus/medical/scale32/beispiel`

Es seien die folgenden vier neueingetroffenen Dateien zu klassifizieren:

```
-rw-r--r--    1 theiner  user        343 Feb  3 11:41 abdomen.00998.532XPBLA.pgm
-rw-r--r--    1 theiner  user        362 Feb  3 11:42 abdomen.00999.500XPBNA.pgm
-rw-r--r--    1 theiner  user        357 Feb  3 11:42 thorax.00999.600XP2NA.pgm
-rw-r--r--    1 theiner  user        356 Feb  3 11:43 wirbel.00999.430XPBNC.pgm
```

Dazu sind sie zunächt mit dem oben vorgestellten Script in JPG-Dateien umzuwandeln. Dabei ist die Originalgröße beizubehalten, um die Berechnung der Aspect Ratios zu ermöglichen (Parameter `-geometry 32x32!` entfernen):

```
for f in *.pgm ; do
  echo $f > ~/hilf.txt
  hilf=`sed s/.pgm//g < ~/hilf.txt`
  echo "Converting $hilf.pgm to $hilf.jpg..."
  convert -colorspace GRAY -quality 85 $hilf.pgm $hilf.jpg
done
```

Nun werden die Aspect Ratios wie folgt berechnet:

```
# ls -1 --color=no *.jpg >AspectRatio.tmp
# Get_AspectRatio
Aspect Ratios werden geschrieben....
4 Muster gelesen
```

In einem weiteren Scriptdurchlauf werden nun JPG-Bilder mit einer Skalierung auf 32×32 erzeugt. Für die Klassifikation mittels `nn_tangent` wird ein Eingabefile im 'i6-ASCII-Format' mit allen zu klassifizierenden Bildern benötigt. Diese Datei 'beispiel.joerg' wird in den folgenden 2 Schritten angelegt:

```
# ls -1 --color=no *.jpg >ConvertToJoerg.tmp
# ConvertToJoerg beispiel 32 ./
Writing JOERG file....
Opening abdomen.00998.532XPBLA.jpg....
Opening abdomen.00999.500XPBNA.jpg....
Opening thorax.00999.600XP2NA.jpg....
Opening wirbel.00999.430XPBNC.jpg....
```

Damit wird also aus der Liste aller verfügbaren JPG-Dateien (die in der Datei 'ConvertToJoerg.tmp' vorliegt) eine einzelne Datei im 'i6-ASCII-Format' erzeugt, die alle Bilder und die dazugehörigen Klassen enthält. Die Klassen wurden dabei den Dateinamen entnommen (0 =abdomen,becken/ 1 =arm, bein/ 2 =mamma/ 3 =schaedel/ 4 =thorax/ 5 =wirbel). Nun kann die Klassifikation der Daten gestartet werden:

```
# nn_tangent beispiel.joerg 1.0 0.8 1 1 3700.0 beispiel_id.txt 0 0 50.0 1220.0 4
0 0.0 0.0 >beispiel.log
```

Es ist zu beachten, dass dieser Aufruf offensichtlich (wegen *leaving one out*) für nur 4 Bilder keinen Sinn ergibt. Er dient damit nur der Veranschaulichung! Die Fehlerbilder in dem erzeugten Logfile 'beispiel.log' können schließlich folgendermaßen visualisiert werden:

```
# fehlerbilder beispiel.log >makeindexhtml.tmp
# makeindexhtml
```

Die entstandene Datei 'index.html' kann nun bearbeitet, mit einem Browser betrachtet und ausgedruckt werden.

9.4.1 Erzeugung von Kovarianz- und Cooccurrence-Matrizen

Manchmal ist es nötig, klassenabhängige oder globale Kovarianzmatrizen mit den dazugehörenden Mittelwertbildern zu erzeugen. Für die Klasse Abdomen in den vorliegenden Beispielbildern kann das wie folgt geschehen:

```
# ls -1 --color=no abdomen*.jpg >Covarianceforclass.tmp
# Covariance_for_class 32 abdomen
Mittelwertberechnung beginnt....
Mittelwertbild berechnet und ausgegeben
Dateiname: abdomen.00998.532XPBLA.jpg
Dateiname: abdomen.00999.500XPBNA.jpg
2 Bilder waren es!

Varianz :   433.383757

Mittelwert der Standardabweichung :    20.817871
```

Es wurden dabei die beiden Dateien 'abdomen' und 'abdomen.covariance' erzeugt. Die erstgenannte Datei enthält das Mittelwertbild und liegt im PGM-Format vor. Die zweite Datei ist eine Textdatei, die zeilenweise die Diagonaleinträge der Kovarianzmatrix beinhaltet.

Die Erzeugung von C_1-Cooccurrence-Matrizen ist für ein einzelnes Bild beispielsweise wie folgt durchführbar:

```
# CoocView abdomen.00998.532XPBLA.jpg 0.25
```

In diesem Fall wurde eine Datei 'abdomen.00998.532XPBLA.jpg.cooc' erzeugt, die eine binarisierte Darstellung einer C_1-Cooccurrence-Matrix darstellt (maximaler Kontrast, siehe auch die mittlere Spalte in Abbildung 7.1). Eine Schwelle von '0.0' würde die exakte Verteilung der Einträge widergeben (keine Kontrastveränderung).

9.5 Das 'i6-ASCII-Format'

In den meisten durchgeführten Experimenten lagen die betrachteten Bilder gemeinsam in einem File mit der folgenden Struktur:

- 1. Zeile: Die Anzahl der in diesem File vorhandenen Klassen und die Dimension jedes Bildes. Innerhalb eines Files sind also nur Bilder gleicher Dimension erlaubt.
- Zeilen 2 bis $(n-1)$: Zunächst die Klasse k $(0 \leq k \leq (K-1))$, danach die Pixel des Bildes (zeilenweise hintereinander)
- Zeile n: -1 als Ende-Markierung

Dieser Aufbau von Dateien wurde in vorhergehenden Abschnitten als 'i6-ASCII-Format' bezeichnet. Im folgenden ein Beispiel für ein File im 'i6-ASCII-Format', das zwei Bilder unterschiedlicher Klasse der Dimension 3×3 (also mit je 9 Pixeln) enthält:

```
2 9
0 127 138 67 18 7 45 39 250 201
1 53 74 106 178 190 255 0 98 123
-1
```

Literaturverzeichnis

[Aks1998] S. Aksoy, R. M. Haralick:
Content-Based Image Database Retrieval Using Variances of Gray Level Spatial Dependencies, Multimedia Information Analysis and Retrieval, IAPR International Workshop, 1998, S. 3-19

[Als1996] P. Alshuth, T. Hermes, C. Klauck, J. Kreyß, M. Röper:
IRIS - A System for Image and Video Retrieval, Proc. of CASCON '96, 12-14 November, Toronto, 1996.

[Ash1995] J. Ashley et al.:
Automatic and Semiautomatic Methods for Image Annotation and Retrieval in QBIC, Storage and Retrieval for Image and Video Databases III, SPIE, Vol. 2420, S. 24-35, 1995.

[Bec1990] N. Beckmann, H. P. Kriegel, R. Schneider, B. Seeger:
The R-Tree: An Efficient and Robust Access Method for Points and Rectangles,* Proc. ACM SIGMOD, Int. Conf. on Management of Data, S. 322-331, 1990.

[Berm1999] A. P. Berman, L. G. Shapiro:
A Flexible Image Database System for Content-Based Retrieval, Computer Vision & Image Understanding, Vol. 75, S. 175-195, August 1999.

[Bon1999] D. Bongartz, J. Bredno:
Texturanalyse zur Grobklassifikation sekundär digitalisierter Röntgenaufnahmen in medizinischen Image-Retrieval-Anwendungen, Studienarbeit am Institut für Medizinische Informatik der RWTH Aachen, Mai 1999.

[Bre1999] J. Bredno, F. Vogelsang, J. Dahmen, T. Lehmann et al.:
Eine Entwicklungsumgebung für die interdisziplinäre Zusammenarbeit bei der Entwicklung des Image-Retrieval-Systems IRMA, Bildverarbeitung für die Medizin, Springer Verlag, Berlin, 1999.

[Bre2000a] J. Bredno, M. Kohnen, J. Dahmen, F. Vogelsang, B. Wein, T. Lehmann:
Synergetic Impact obtained by a Distributed Developing Platform for Image Retrieval in Medical Applications (IRMA), Proceedings SPIE 3972(33), 2000, in press.

[Bre2000b] J. Bredno, S. Brandt, J. Dahmen, B. Wein, T. Lehmann:
Kategorisierung von Röntgenbildern mit aktiven Konturmodellen, Bildverarbeitung für die Medizin 2000, München, März 2000, in press.

[Car1997] C. Carson, S. Belongie, H. Greenspan, J. Malik:
Color- and Texture-Based Image Segmentation Using EM and Its Application to Image Querying and Classification, to appear in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997.

[Che1994] Q. Chen, M. Defrise, F. Deconinck:
Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, Nr. 12, S. 1156-1168, 1994.

[Dah1998a] J. Dahmen, T. M. Lehmann, K. Spitzer, H. Ney:
Image Retrieval für klinische Bilddatenbanken, in T. M. Lehmann, V. Metzler, K. Spitzer, T. Tolxdorff (Hrsg.): *Bildverarbeitung für die Medizin,* Springer Verlag, Berlin, S. 442-446, 1998.

[Dah1998b] J. Dahmen, K. Beulen, H. Ney:
Objektklassifikation mit Mischverteilungen, 20. DAGM Symposium Mustererkennung, S. 167-174, Stuttgart, September 1998.

[Dah1999] J. Dahmen, R. Schlüter, H. Ney:
Discriminative Training of Gaussian Mixtures for Image Object Recognition, 21. DAGM Symposium Mustererkennung, S. 205-212, Bonn, September 1999.

[Dah2000a] J. Dahmen, K. Beulen, M. O. Güld, H. Ney:
A Mixture Density Based Approach to Object Recognition for Image Retrieval, 6th International Conference on Content-Based Multimedia Information Access (RIAO), Paris, April 2000, in press.

[Dah2000b] J. Dahmen, J. Hektor, R. Perrey, H. Ney:
Automatic Classification of Red Blood Cells using Gaussian Mixture Densities, Bildverarbeitung in der Medizin, München, März 2000, in press.

[Dah2000c] J. Dahmen, D. Keysers, M. O. Güld, H. Ney:
Invariant Image Object Recognition using Mixture Densities, eingereicht für *International Conference on Pattern Recognition* ICPR 2000, Barcelona, Spanien, 2000.

[Dah2000d] J. Dahmen, T. Theiner, D. Keysers, H. Ney, T. Lehmann, B. Wein:
Classification of Radiographs in the 'Image Retrieval in Medical Applications' System (IRMA), 6th International Conference on Content-Based Multimedia Information Access (RIAO), Paris, April 2000, in press.

[Dev1996] L. Devroye, L. Györfi, G. Lugosi:
A Probabilistic Theory of Pattern Recognition, Springer, New York, 1996.

[Doh1991] M. Dohmen, W. Oberschelp:
Mathematische Methoden für Bildverarbeitung und Computergraphik - Spezielle algorithmische Probleme der Computergraphik, Schriften zur Informatik und angewandten Mathematik.

Bericht Nr. 149, Juli 1991, Interner Bericht am Lehrstuhl für Angewandte Mathematik insbesondere Informatik der RWTH Aachen.

[Dud1973] R. O. Duda, P. E. Hart:
Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.

[Eak1996] J. P. Eakins:
Automatic Image Content Retrieval - Are we getting anywhere?, in Third International Conference on Electronic Library and Visual Information Research (ELVIRA3), pp. 123-135, De Montford University, Milton Keynes, 1996.

[Fal1994] C. Faloutsos, R. Barber, M. Flickner, J. Hafner:
Efficient and Effective Querying by Image Content, Journal of Intelligent Information Systems, Vol. 3, S.231-262, 1994.

[Fli1995] M. Flickner, H. Sawhney, W. Niblack et al.:
Query by Image and Video Content: The QBIC System, IBM Research Report RJ 9949, 1995.

[Hak1991] H. Haken:
Synergetic Computers and Cognition, Springer Verlag, New York, S. 36-50, 1991.

[Har1973] R. M. Haralick, K. Shanmugam, I. Dinstein:
Textural Features for Image Classification, IEEE Transactions on Systems, Man and Cybernetics SMC-3(6), S. 610-621, 1973.

[Key2000] D. Keysers, J. Dahmen, T. Theiner, H. Ney:
Experiments with an Extended Tangent Distance, eingereicht für *International Conference on Pattern Recognition* ICPR 2000, Barcelona, Spanien, 2000.

[Koh2000a] M. Kohnen, F. Vogelsang, B. Wein, M. Kilbinger, R. Günther, F. Weiler, J. Bredno, J. Dahmen:
Kategorisierung von digitalen Röntgenbildern mit parametrisierbaren Formmodellen, Bildverarbeitung für die Medizin 2000, München, März 2000, in press.

[Koh2000b] M. Kohnen, F. Vogelsang, B. Wein, M. Kilbinger, R. Günther, F. Weiler, J. Bredno, J. Dahmen:
Knowledge Based Automated Feature Extraction to Categorize Secondary Digitized Pictures, Proceedings SPIE 3979-70, 2000, in press.

[Leh2000a] T. Lehmann, B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, M. Kohnen:
Content-Based Image Retrieval in Medical Applications: A Novel Multi-Step Approach, Procs. Int. Society for Optical Engineering (SPIE), Vol. 3972(32), 2000, in press.

[Leh2000b] T. Lehmann, B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, M. Kohnen:
Ein strukturiertes Konzept zum inhaltsbasierten Zugriff auf medizinische Bildarchive, Bildverarbeitung für die Medizin 2000, München, März 2000, in press.

[Mog1997] B. Moghaddam, A. Pentland:
Probabilistic Visual Learning for Object Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, S. 696-710, Juli 1997.

[Ney1998] H. Ney:
Script zur Vorlesung Mustererkennung und Neuronale Netze, RWTH Aachen, 1998.

[Pre1992] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery:
Numerical Recipes in C - The Art of Scientific Computing, Cambridge University Press, USA 1992.

[Rav1998] S. Ravela, R. Manmatha:
On Computing Global Similarity in Images, Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98), S. 82-87, 1998.

[Schä1997] P. Schäuble:
Multimedia Information Retrieval, Kluwer Academic Publishers, Norwell MA, S. 20-29, 1997.

[Schi1996] B. Schiele, J. L. Crowley:
Object Recognition Using Multidimensional Receptive Field Histograms, Proceedings 4th European Conference Computer Vision, Cambridge, U.K., 1996.

[Schm1997] C. Schmid, R. Mohr:
Local Greyvalue Invariants for Image Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI(19), No. 5, S. 530-535, Mai 1997.

[Schr2000] M. Schreckenberg, J. Dahmen, G. Schummers, D. Meyer-Ebrecht, H. Ney:
Automatische Endokarderkennung in 3D mit approximierenden Thin-Plate-Spline Modellen unter Einsatz von Gauß'schen Mischverteilungs-Modellen für die lokale Klassifikation, Bildverarbeitung für die Medizin 2000, München, März 2000, in press.

[Sim1993] P. Simard, Y. Le Cun, J. Denker:
Efficient Pattern Recognition Using a New Transformation Distance, Advances in Neural Information Processing Systems 5, Morgan Kaufmann, San Mateo CA, S. 50-58, 1993.

[Smi1996a] J. R. Smith, S. F. Chang:
VisualSEEk: A Fully Automated Content-Based Image Query System, ACM Multimedia 96, Boston MA, November 1996.

[Smi1996b] J. R. Smith, S. F. Chang:
Tools and Techniques for Color Image Retrieval, Storage and Retrieval for Still Image and Video Databases IV, SPIE, Vol 2670, S.426-437, 1996.

[Swa1991] M. J. Swain, D. H. Ballard:
Color Indexing, International Journal of Computer Vision, Vol. 7, Nr. 1, 1991.

[Tag1997] H. D. Tagare, C. C. Jaffe, J. Dungan:
Medical Image Databases: A Content-based Retrieval Approach, Journal of the American Medical Informatics Association, 4:184-198, 1997.

[Vap1998] V. Vapnik:
Statistical Learning Theory, John Wiley & Sons, New York, 1998.

[Vog1997] F. Vogelsang, F. Weiler, B. Wein, M. Kilbinger, R. W. Günther:
Image Content Analysis using Textural Information and Synergetic Classifiers, Proceedings European Congress of Radiology ECR 97, S. 291ff., Wien, 1997.

[Wen1994] T. Wendler, B. Wein, R. van den Brock:
Deliverable USINT in the EurIPACS-Project A1009, Technical Report, AIM, Ref. XIII, European Commission, Brussels, Februar 1994.

[URL1] QBIC: *http://www.qbic.almaden.com/*

[URL2] Ultimedia Manager: *http://www-4.ibm.com/software/data/umm/umm.html*

[URL3] VisualSEEk: *http://disney.ctr.columbia.edu/VisualSEEk/*

[URL4] Blobworld: *http://www.cs.berkeley.edu/~ carson/blobworld/*

[URL5] IRIS: *http://www.informatik.uni-bremen.de/grp/ag-ki/projects/iris/iris-start.html*

[URL6] FIDS: *http://www.cs.washington.edu/research/imagedatabase/dissertation/*

[URL7] postgreSQL: *http://www.postgresql.org/*

[URL8] DICOM: *http://www.desacc.com/*