Affin - invariante Merkmale für die 2D-Bilderkennung

Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen

Prof. Dr.-Ing. H. Ney

vorgelegt von:

Cand. Inform. Ralf Perrey
Matrikelnummer 180385

Gutachter:

Prof. Dr. Ney
Prof. Hromkovič

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 Graphiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, den 24.02.2000

Ralf Perrey

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Danksagung

Bedanken möchte ich mich bei Herrn Prof. Dr. Hermann Ney, der mir die Gelegenheit gegeben hat, diese Diplomarbeit am Lehrstuhl für Informatik VI anzufertigen. Mein besonderer Dank gilt Herr Dipl.-Inform. Jörg Dahmen für die gute und umfangreiche Betreuung der Arbeit und dafür, daß er mir jederzeit mit seinem Rat zur Seite stand.

Ein weitere Dank gilt Jens Hektor, der die in dieser Arbeit verwendeten Erythrozyten-Bilder aufbereitet und zur Verfügung gestellt hat. Darüberhinaus möchte ich mich bedanken bei Mark Oliver Güld, Alexander Crämer und Daniel Keysers, mit deren Unterstützung einige der verwendeten Programme entstanden sind.

Kapitel 1

Einführung

Die grundsätzliche Aufgabe der Mustererkennung besteht darin, ein gegebenes Objekt einer Klasse $k = 1, ..., K$ zuzuweisen. Dazu ist es notwendig, die zu klassifizierenden Objekte numerisch in Form von Merkmalsvektoren zu beschreiben. Faßt man beispielsweise den Grauwert eines jeden Bildpixels als Merkmal auf, so ergibt sich ein (in der Regel hochdimensionaler) Merkmalsvektor $x \in \mathbb{R}^{I \times J}$, wobei das Bild von der Dimension $I \times J$ ist. Wird die Klassifikation direkt auf diesen Merkmalen durchgeführt, spricht man auch von *appearance based pattern recognition.* Abbildung 1.1 zeigt einen

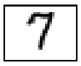 $\longrightarrow 7$

Abbildung 1.1: Beispiel für eine Klassifikationsaufgabe

Merkmalsvektor, der aus $16 \times 16 = 256$ Merkmalen besteht (als Bild interpretiert). Dieser soll nun der Klasse 7 zugeordnet werden. Gesucht ist eine geeignete Funktion $r : x \in \mathbb{R}^{256} \longrightarrow \{0, ..., 9\}$. Die Klassifikation einer Beobachtung erfolgt dann mittels

$$x \in \mathbb{R}^{256} \longrightarrow r(x) = \underset{k}{\operatorname{argmax}}\{g(x,k)\}, \qquad (1.1)$$

wobei $g(x, k)$ die sogenannte Diskriminantenfunktion ist.

Dieses Problem ist nicht einfach, da es offenbar zu Verwechslungen kommen kann, so kann eine 7 etwa mit einer 1 verwechselt werden. Aus diesem Grunde ist eine Entscheidungsfunktion zu finden, die bei einer gegebenen Menge von zu klassifizierenden Objekten die geringste Fehlerrate liefert. Die Fehlerrate bezeichnet dabei das Verhältnis der fehlerhaften Zuordnungen zur Gesamtanzahl der betrachteten Objekte.

Eine typische Erkennungsaufgabe gliedert sich dabei in die in Abbildung 1.2 gezeigten Arbeitsschritte. Eine besondere Bedeutung kommt dabei den Arbeitsschritten der Merkmalsanalyse und der Bildvorverarbeitung zu. Ziel dieser Schritte ist die Extraktion von diskriminativen Merkmalen, also Merkmalen, die eine zuverlässige Klassentrennung ermöglichen. Oftmals ist man auch an einer Dimensionsreduktion der Merkmalsvektoren interessiert, um die Anzahl der freien Modellparameter zu reduzieren. Im Bereich der Bilderkennung kommt dabei den sogenannten invarianten Merkmalen eine besondere Bedeutung zu, da gewisse Bildtransformationen wie etwa Ro-

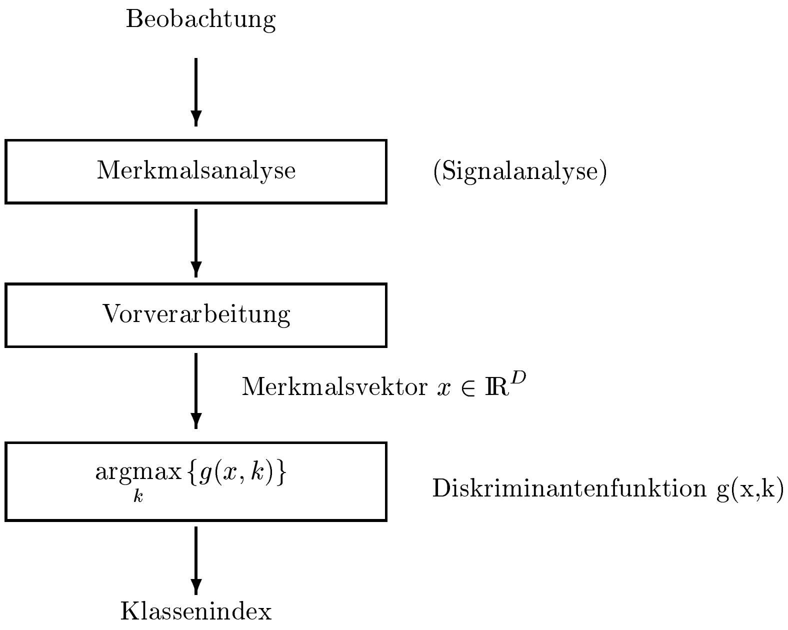

Abbildung 1.2: Struktur eines Klassifikators

tationen, Skalierungen, Verschiebungen oder ähnliche die Klassenzugehörigkeit eines Bildes meist nicht verändern. Aus diesem Grunde sind wir an einer Merkmalsanalyse interessiert, die für Bilder, die sich nur durch solche Transformation unterscheiden, identische Merkmale (auch 'Features' genannt) berechnet. Zu beachten ist, daß der oben erwähnte Ansatz der 'appearance-based' Mustererkennung in Verbindung mit einem Distanznaß wie der Euklidischen Distanz zunächst einmal keine Invarianzeigenschaften aufweist.

Um die erstrebte Invarianz des Klassifikators zu erreichen, existieren verschiedene Möglichkeiten. In dieser Arbeit wird insbesondere auf Merkmale eingegangen, die sich unter den betrachteten Transformationen nicht verändern. Desweiteren können Distanzmaße betrachten, die mögliche erlaubte Veränderungen berücksichtigen. Ein weiterer Ansatz zur Invarianz, der hier nicht betrachtet wird, ist, die Daten im Schritt der Vorverarbeitung so zu normieren, daß die Variabilität verkleinert wird.

Bevor im folgenden die Methoden der invarianten Merkmalsanalyse ausführlich beschrieben werden, werden in Kapitel 2 dieser Arbeit die in den Experimenten verwendeten Datensammlungen vorgestellt. Um eine Bewertung der erzielten Ergebnisse zu gewährleisten, wird in Kapitel 3 auf den Stand der Technik im Bereich der Mustererkennung eingegangen. Kapitel 4 beschäftigt sich mit der in den Experimenten verwendeten Bayes'schen Entscheidungsregel, wobei die Modellierung der benötigten klassenbedingten Wahrscheinlichkeiten unter Verwendung von Gauß'schen Mischverteilungen erfolgt. Kapitel 5 beschäftigt sich mit Methoden der invarianten Merkmalsgenerierung und bildet den Hauptteil dieser Arbeit. Die Leistungsfähigkeit dieser Merkmale wird in Kapitel 6 anhand der erzielten Ergebnisse beurteilt. Kapitel 7 beendet die Arbeit mit einer Zusammenfassung der erzielten Ergebnisse und mit einem Ausblick auf weitere mögliche Arbeiten in diesem Gebiet. Im Anhang wird schließlich ein Teil der in dieser Arbeit erstellten bzw. verwendeten Software vorgestellt.

Kapitel 2

Verwendete Datensammlungen

Diese Diplomarbeit findet in mehreren Projekten, in denen Probleme der Bilderkennung behandelt werden, ihre Anwendung. Die für diese Arbeit besonders relevanten Aufgabenstellungen sind die Klassifikation von Erythrozyten (rote Blutzellen), die sich in ihrer Form unterscheiden, sowie die Erkennung von handgeschriebenen Ziffern.

2.1 Erythrozyten Daten

Die Klassifikation der roten Blutzellen wurde in Zusammenarbeit mit dem Institut für Physiologie der RWTH Aachen bearbeitet. Die roten Blutkörperchen gewährleisten im menschlichen Blut den Transport und die Versorgung des Körpers mit Sauerstoff. Erythrozyten sind runde, scheibenförmige, kernlose Zellen, die eine zentrale Aufhellung ('Delle') aufweisen. Sie haben einen Durchmesser von etwa 8.4 μm, sind am Rand etwa 2.4 μm und in der Mitte etwa 1 μm dick. [Pschyrembel 256, Seite 483]. Rote Blutkörperchen haben normalerweise eine mittlere Lebensdauer von 120 Tagen. Die Form der Erythrozyten kann sich durch Alterung oder durch die Veränderung des osmotischen Drucks verändern. In einem hypertonischen Medium (hohe Ionenkonzentration) verlieren die Zellen Wasser und verformen sich zu einer Stechapfelform (Echinozyt). Die Stärke der Veränderung der Form bei gleicher Lösung läßt auf den Zustand des Erythrozyten schließen. Die verformten Erythrozyten sind weniger flexibel, so daß sie in den Kapillaren einen höheren Widerstand bilden, die weniger flexiblen Formen der Erythrozyten bleiben vermehrt im Maschenwerk der Milz hängen, wo sie abgebaut werden.

Im wesentlichen können drei Formen von Erythrozyten unterschieden werden (bei der folgenden Klassifikation von roten Blutzellen handelt es sich also um ein 3-Klassen-Problem):

- **Stomatozyten**: Die Erythrozyten sind eingewölbt, haben die Form einer 'Schüssel'.
- **Diskozyten**: Die häufigste Form: rund, 'platt' mit einer 'Delle' im Zentrum.
- **Echinozyten**: Eine kugelähnliche Form mit 'Ausstülpungen' (Stechapfelform).

Dabei ist der Übergang zwischen den verschiedenen Formen oft fließend (vergleiche Abb. 2.1), was deren Klassifikation offensichtlich erschwert. In vielen Fällen kann auch ein Fachmann keine

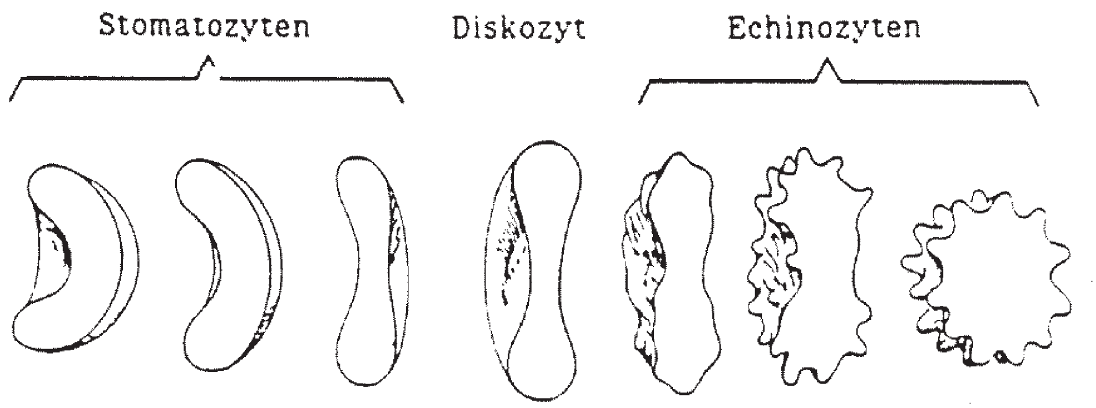

Abbildung 2.1: Erythrozytenklassen

eindeutige Klassifikation vornehmen, insbesondere dann nicht, wenn nur ein Einzelbild vorliegt (anstelle einer Bildsequenz, die die Zelle aus unterschiedlichen Ansichten zeigt).

Da Erythrozyten einen einfachen Aufbau haben sind sie besonders geeignet, um an ihnen die Eigenschaften von Zellmembranen zu erforschen. Diese Grundlagenforschung ist ein Ziel, das am Institut für Physiologie im Universitätsklinikum der RWTH Aachen verfolgt wird. Dabei wird die Formveränderung von Erythrozyten unter Hinzugabe von bestimmten Medikamenten gemessen und mit dem Verhalten 'gesunder' Zellen verglichen [Deuticke et al. 1990].

Die Stärke der Verformung der roten Blutkörperchen in gegebener Lösung hängt von weiteren Einflüssen ab, zum Beispiel haben die Erythrozyten von Vegetariern und Diabetikern eine erhöhte Neigung, sich in Echinozyten zu formen. Aus dem Verhältnis zwischen Stomatozyten, Diskozyten und Echinozyten kann so auf den Zustand der Erythrozyten geschlossen werden. Wenn es ein einfaches Zähl- und Klassifikationsverfahren gibt, kann so ein diagnostisches Verfahren entwickelt werden (bislang erfolgt die Klassifikation der Zellen manuell durch mikroskopieren).

Am Institut für Physiologie der RWTH Aachen wurde ein Verfahren zur photographischen Erfassung der Blutzellen entwickelt. Wenn die Blutzellen auf einem Objektträger aufgenommen würden, so hätte dieses Verfahren offenbar (einen unerwünschten) Einfluß auf die Form der Zellen. Die Erythrozyten würden durch ihr Gewicht verformt und so kein gutes, d.h. der Realität entsprechendes, Meßergebnis erzielt.

Im Gegensatz zu dieser 'konventionellen' Lösung werden die Blutzellen im Institut für Physiologie frei sedimentierend mikroskopiert. Dazu befindet sich in einer Kapillaren eine Lösung, in der Erythrozyten schwimmen, wobei sie durch ihr Gewicht langsam nach unten sedimentieren. Der Lichtstrahlenverlauf im Mikroskop verläuft waagerecht (siehe Abb. 2.2 und 2.3). Durch das langsame Absinken können die Erythrozyten einzelnd mikroskopiert/ photographiert werden. Da die Erythrozyten frei schwimmen wird ihre Form nicht verändert.

Die vom Sedimentier-Mikroskop gelieferten Bilder werden automatisch weiterverarbeitet. Im geglätteten Bild werden Regionen gleichen Grauwertes markiert. Da die Größe der Erythrozyten und die

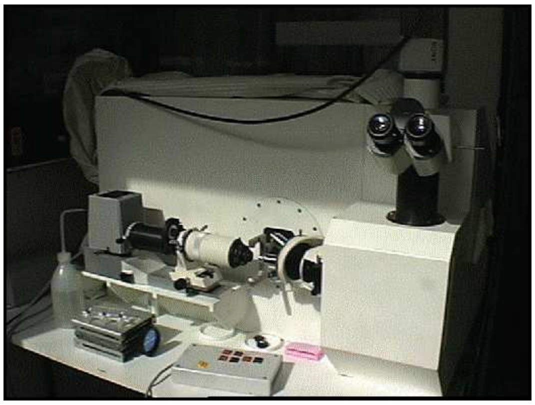

Abbildung 2.2: Sedimentier-Mikroskop

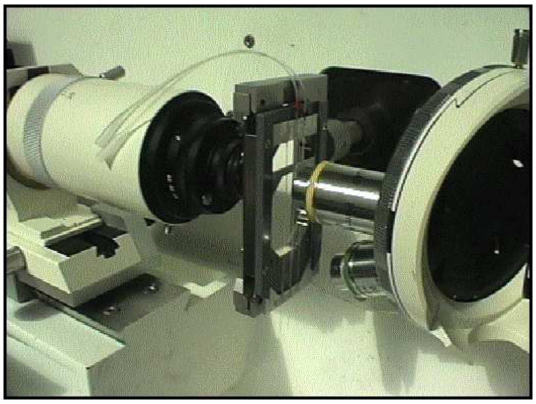

Abbildung 2.3: Sedimentier-Mikroskop mit Kapillare

Vergrößerung des Mikroskop bekannt ist, können Regionen (zwischen 400 und 10000 Pixel) der passenden Größe als Erythrozyten segmentiert werden. Es werden Grauwertbilder der Größe 128×128 Pixel bei einer Farbtiefe von 256 Grauwerten abgespeichert, die jeweils einen isolierten Erythrozyten zeigen [Weber 1996].

Die Aufgabe, die sich nun stellt, ist, die Bilder (wie sie in Abbildung 2.6 beispielhaft dargestellt sind) den Klassen Stomatozyt, Diskozyt bzw. Echinozyt zuzuordnen. Besonders erschwert wird diese Aufgabe dadurch, daß die Erythrozyten in der Lösung frei schwimmen und prinzipiell aus jeder beliebigen Ansicht aufgenommen werden können. Insbesondere Stomatozyten und Diskozyten sind in Abhängigkeit vom Blickwinkel des Betrachters sehr ähnlich (etwa wenn sie von oben betrachtet werden). Sie können dann nur durch die unterschiedliche Lichtbrechung unterschieden werden. In der Seitenansicht können sie allerdings recht zuverlässig unterschieden werden. Die Ansichten der Echinozyten unterscheiden sich dagegen nicht sehr stark. Eine weitere Schwierigkeit ist es, daß die Erythrozyten in jeder Ansicht noch beliebig rotiert sein können und sich in ihrer Größe unterscheiden können. Dies macht - neben der offenbar notwendigen Translationsinvarianz - auch

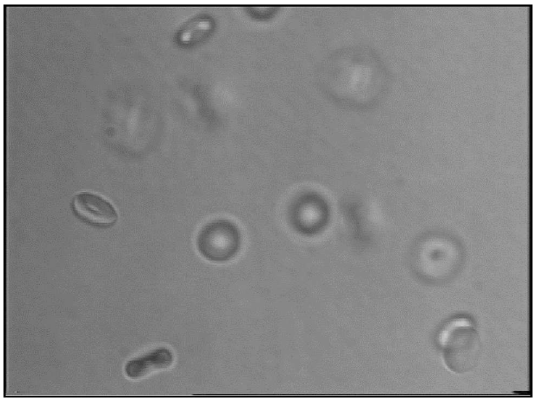

Abbildung 2.4: Beispiel für eine Mikroskopaufnahme (Rohdaten)

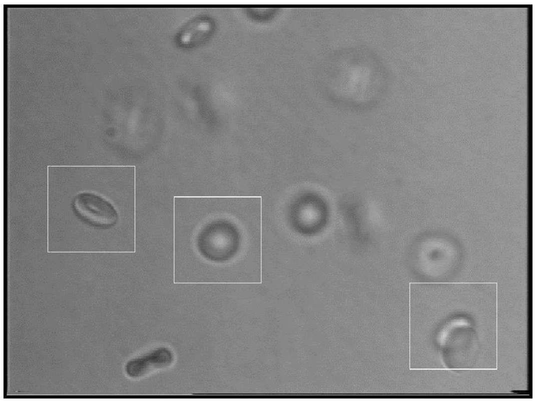

Abbildung 2.5: Eine Mikroskopaufnahme mit markierten Erythrozyten

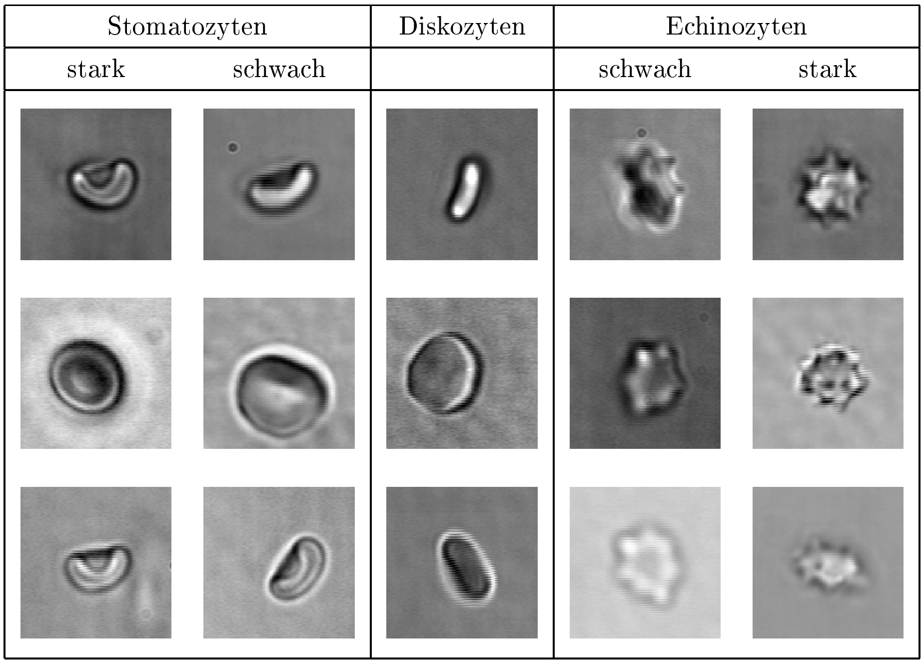

Abbildung 2.6: Beispielbilder Erythrozyten

eine bezüglich Rotation und Skalierung invariante Erkennung erforderlich.

Die verwendete Datensammlung besteht aus 5162 Bildern von roten Blutzellen, die von einem Experten in die Klassen

(1) starker Stomatozyt (2159 Aufnahmen),

(2) schwacher Stomatozyt (1157 Aufnahmen),

(3) Diskozyt (928 Aufnahmen),

(4) schwacher Echinozyt (504 Aufnahmen)

(5) starker Echinozyt (414 Aufnahmen)

eingeteilt wurden (siehe Abbildung 2.6).

Da die Übergänge fließend sind, ist nicht in allen Fällen eine sichere Zuordnung möglich. Die in dieser Arbeit verwendeten Daten wurden durch mehrfaches Anschauen und Korrigieren in ihre Klassen eingeteilt. Blutzellen, die während des Sedimentiervorganges häufiger aufgenommen wurden, können dabei mit einer höheren Sicherheit klassifiziert werden, da der selbe Erythrozyt aus unterschiedlichen Positionen betrachtet werden kann. Man beachte, daß dem automatischen Klassifikationsalgorithmus diese Information nicht zur Verfügung steht, da nur Einzelbilder, keine Bildsequenzen betrachtet werden. Trotz allem liegt die menschliche Fehlerrate für diese Aufgabe bei $> 20\%$ [Fischer 1999]. Außerdem ist die manuelle Klassifikation nicht nur ermüdend, sondern auch zeit- und kostenintensiv, so daß eine automatische Auswertung erstrebenswert ist. Wie in Kapitel 6 dargestellt wird, liegt die Fehlerrate des realisierten statistischen Klassifikators deutlich unter der menschlichen Fehlerrate, obwohl dieser nur Einzelbildinformation verwendete.

2.2 Die US-Postal-Service Datensammlung

Beim US-Postal-Service Task (USPS) handelt es sich um eine standardisierte Datensammlung (**ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/**) von handgeschriebenen Ziffern, die sich besonders gut zur Evaluierung von Klassifikationsalgorithmen eignet, da zahlreiche Vergleichsergebnisse anderer Arbeitsgruppen existieren. Mit diesen ist ein fairer Vergleich der eingesetzten Verfahren möglich.

Die Datensammlung besteht aus 7291 Trainingsdaten und 2007 Testdaten, bei denen es sich um vorsegmentierte, handgeschriebene Ziffern handelt. Jedes Bild ist 16×16 Pixel groß und besteht aus 256 Grauwerte. Es handelt sich um Daten aus der Praxis, genauer gesagt um amerikanische Postleitzahlen. Die Aufgabe gilt mit einer menschlichen Fehlerrate von 2.5% [Simard et al. 1993] als sehr schwer. Ein Grund hierfür liegt in der automatischen Segmentierung der Ziffern, bei der es teilweise zu Fehlern kam (insbesondere dann, wenn die Ziffern zu dicht aneinander geschrieben wurden). Diese Daten wurden aber bewußt in den Testdaten belassen, da sie für den Einsatz von OCR-Systemen ('optical character recognition') in der Praxis typisch sind.

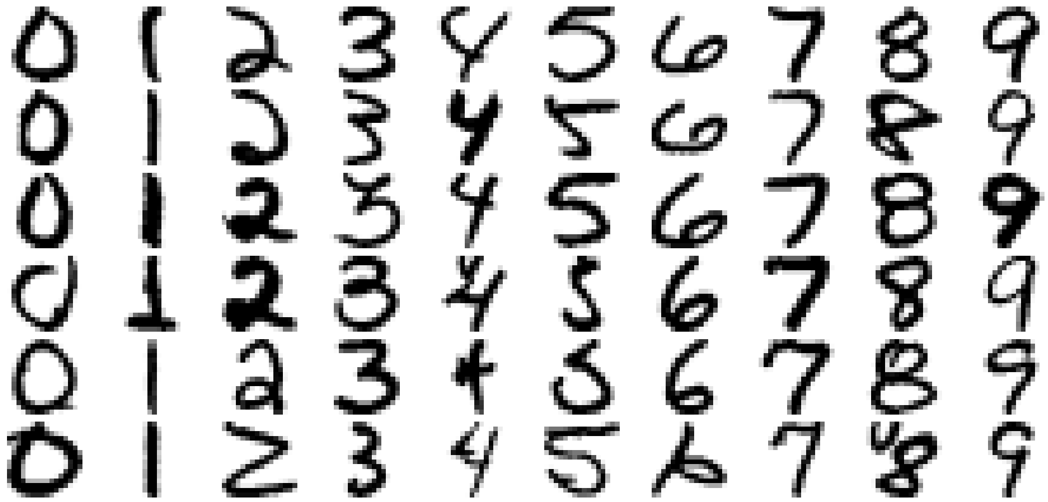

Abbildung 2.7: Einige USPS Testdaten

In Abbildung 2.7 sind einige Beispiele von USPS-Ziffern zu sehen. Es sind auch typische Störungen (Fehlsegmentierungen) wie bei der ¨8¨ in der 6. Zeile zu erkennen.

Im folgenden Kapitel gehen wir kurz auf den Stand der Technik im Bereich der automatischen Objekterkennung ein, da eine Evaluierung der in dieser Arbeit vorgestellten Algorithmen nur im Vergleich zu internationalen state-of-the-art Ergebnissen sinnvoll ist.

Kapitel 3

Stand der Technik

In diesem Kapitel soll kurz auf den Stand der Technik eingegangen werden, also auf die von anderen Gruppen erzielten Ergebnisse auf den in dieser Arbeit verwendeten Daten. Dies ist im Sinne einer objektiven Evaluierung unserer Ergebnisse notwendig.

3.1 Stand der Technik: Erythrozyten

Für die Blutzellen sind keine vergleichbaren Ergebnisse verfügbar, da diese Daten aus einer speziellen Anwendung stammen, die am Institut für Physiologie des Universitätsklinikums der RWTH Aachen verfolgt wird (dazu später mehr). Die menschliche Fehlerrate liegt bei $> 20\%$ [Fischer 1999]. Die Klassifikation der verwendeten Daten erfolgte durch mehrfaches Durchschauen unterschiedlicher Personen und dadurch, daß von ein und derselben Blutzelle in der Regel mehrere Aufnahmen aus ihrem Sedimentierprozeß existieren, die unterschiedlich gut für die Klassifikation geeignet sind.
Im Institut für Physiologie wurde versucht die Außenkante der Erythrozyten im Bild zu erkennen, und über die Formmerkmale, wie etwa der Glattheit der Kurve, eine Aussage über die Klassenzugehörigkeit zu machen. Die Kantenerkennung wies jedoch nicht die nötige Präzision auf, so daß der Weg nicht zu Ende gegangen wurde. Die erzielbare Fehlerrate lag in jedem Fall deutlich über der menschlichen.
Aus diesem Grunde wurden die Experimente auf den roten Blutzellen in der vorliegenden Diplomarbeit recht bald eingestellt. Stattdessen wurde der in Kapitel 2 beschriebene USPS Korpus verwendet, für den zahlreiche Vergleichsergebnisse vorhanden sind, auf die im folgenden genauer eingegangen wird.

3.2 Stand der Technik: USPS

Auf Grund der leichten Verfügbarkeit der Daten und der Existenz zahlreicher Vergleichsergebnisse veröffentlichten im Laufe der Zeit viele internationale Arbeitsgruppen Erkennungsergebnisse auf USPS. Ein weiterer Grund für diese Tatsache liegt darin begründet, daß die USPS Aufgabe als deutlich schwieriger gilt als etwa der vergleichbare MNIST Korpus, für den die besten Fehlerraten bei unter 1% liegen. Im Gegensatz dazu gibt Simard die menschliche Fehlerrate auf den USPS-Testdaten mit 2.5% an [Simard et al. 1993]. Vapnik [Vapnik 1995] berichtet von einer Fehlerrate von 16.2% unter Verwendung eines C4.5 Entscheidungsbaumes [Breiman et al. 1993]. Auch wenn diese extrem schlechte Fehlerrate angezweifelt werden darf, so zeigt sie doch die Schwere der Aufgabe. Ebenso berichtet Vapnik eine Fehlerrate von 5.9% auf dem bislang besten bekannten 2-lagigen neuronalen Netz (realisiert durch B. Schölkopf). Simard et al. erzielten 1998 eine Fehlerrate von 4.2% unter Verwendung eines 5-lagigen neuronalen Netzes. Bei diesem Netz handelte es sich um eine verbesserte Version des LeNet1, welches unter Berücksichtigung von Simard's Tangentendistanz trainiert wurde [Simard et al. 1998]. Simard hatte bereits 1993 die sogenannte Tangentendistanz eingeführt. Dieses Distanzmaß [Simard et al. 1993] ist invariant gegenüber vom Benutzer zu wählenden Transformationen (in der Originalarbeit affine Transformationen und Strichdicken). In Verbindung mit einem 1-NN Klassifikator erzielte Simard eine Fehlerrate von 2.6% auf den USPS Testdaten. Allerdings fügte er den Trainingsdaten etwa 2500 maschinengedruckte Ziffern zu, so daß dieses Ergebnis kaum mit den anderen hier genannten vergleichbar ist. 1997 berichtete Schölkopf [Schölkopf 1997] eine Fehlerrate von 4.0% unter Verwendung der von Vapnik (bereits Jahre vorher) wiederentdeckten Support Vektor Maschine [Schölkopf 1997]. Diese gewann darauf enorm an Popularität, da prinzipiell jeder Algorithmus, der nur auf der Berechnung von Skalarprodukten basiert, teilweise deutlich verbessert werden konnte (so entstanden etwa die Kernel-PCA (1997, [Schölkopf et al. 1997]) und etwas später die Kernel-LDA (1999, [Mika et al. 1999]). Schölkopf selbst erweiterte 1998 seinen Support Vektor basierten Klassifikator, indem er die Simard's Tangentendistanz berücksichtigte. Diese sogenannte 'invariante Support Vektor Maschine' erzielte auf den Original USPS Trainings- und Testdaten eine Fehlerrate von 3.0%.

Tabelle 3.1: Ergebnisse USPS

Methode	Fehlerrate [%]
Menschliche Fehlerrate [Simard et al. 1993]	2.5
C4.5 Decision Tree [Vapnik 1995]	16.2
Neuronales Netz (2 Layer) [Vapnik 1995]	5.9
LeNet1 (5 Layer) [Vapnik 1995]	5.1
Neuronales Netz (5 Layer) [Simard et al. 1998]	4.2
Support Vektor Maschine [Schölkopf 1997]	4.0
Invariante Support Vektoren [Schölkopf et al. 1998]	3.0
GMV [Dahmen et al. 1999]	3.4
invariante GMV [Dahmen et al. 2000a]	2.7
invariante Kernel Densities [Keysers et al. 2000]	2.2

Diese Fehlerrate war lange Zeit die beste auf USPS publizierte. Sie wurde erst 1999 unter Verwendung statistischer Klassifikatoren unterboten. Mit 'invarianten Gauß'schen Mischverteilungen' [Dahmen et al. 2000a] wurde am Lehrstuhl i6 eine Fehlerrate von 2.7% erzielt, invariante Kernel-densities lieferten sogar eine Fehlerrate von 2.2% [Keysers et al. 2000].

Diese Ergebnisse wurden parallel zu dieser Arbeit mit der invarianten Tangentendistanz erzielt. Da das Ziel dieser Arbeit darin besteht, die Evaluierung von verschiedenen Methoden zur Generierung invarianter Merkmale durchzuführen, werden wir als Vergleichsgrundlage das beste bislang erzielte Ergebnis unter Verwendung von Gaußschen Mischverteilungen ohne Simard's Tangentendistanz heranziehen. Dieses liegt unter Verwendung der in Kapitel 4.4.3 beschriebenen Subspace Methode (zur Dimensionsreduktion) bei einer Fehlerrate von 3.4% [Dahmen et al. 1999].

Eine Übersicht einiger bislang erzielter USPS Ergebnisse zeigt Tabelle 3.1.

Kapitel 4

Gauß'sche Mischverteilungen im Bayes-Kontext

Nachdem in den vorherigen Kapiteln die in dieser Arbeit verwendeten Datensammlungen und die auf diesen erzielten *state-of-the-art* Ergebnisse anderer Arbeitsgruppen vorgestellt wurden, beschäftigen wir uns in diesem Kapitel mit dem von uns verwendeten, statistischen Klassifikationsansatz. Wie in Kapitel 1 bereits angedeutet, ist die Wahl einer geeigneten Entscheidungsregel von großer Bedeutung. In dieser Arbeit wurde die sogenannte Bayes'sche Entscheidungsregel gewählt, auf die nun genauer eingegangen wird.

4.1 Bayes'sche Entscheidungsregel

Die Klassifikation einer Beobachtung $x \in \mathbb{R}^D$ erfolgt im folgenden unter Verwendung der Bayes'schen Entscheidungsregel

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

Dabei bezeichnet $p(k)$ die sogenannte a-priori Wahrscheinlichkeit einer Klasse $k = 1, \ldots, K$ und $p(x|k)$ die $k$lassenbedingte Wahrscheinlichkeit für eine Beobachtung x gegeben die Klasse k. Wir werden im folgenden zeigen, daß die Bayes'sche Entscheidungsregel für den Fall bekannter Wahrscheinlichkeiten $p(k)$ und $p(x|k)$ optimal ist, also die geringstmögliche Fehlerrate erzielt (siehe auch [Duda & Hart 1973, S. 10-43]). In der Regel sind diese Größen jedoch unbekannt, so daß Modelle zu ihrer Beschreibung gewählt und deren Parameter aus den Trainingsdaten gelernt werden müssen (wir verfolgen den Ansatz des 'überwachten Lernens'). Bevor wir uns im folgenden genauer mit dieser Problematik beschäftigen, soll zunächst die Bayes'sche Entscheidungsregel formal hergeleitet werden, wobei wir als Zielfunktion eine minimale Fehlerrate wählen.

Dazu definieren wir zunächst eine (lokale) Kostenfunktion $L[c, k]$ für einen gegebenen Merkmalsvektor $x \in \mathbb{R}^D$ aus der Klasse c wie folgt, wobei $r(x) = k$ gelte:

$$L[c, k] := \begin{cases} 0 & c = k \quad richtig \\ 1 & c \neq k \quad falsch \end{cases} \tag{4.1}$$

(es werden also - anschaulich gesprochen - die Erkennungsfehler gezählt bzw. im folgenden minimiert). Die Gesamtkosten R ergeben sich dann über eine Integration über den gesamten Raum, es folgt also:

$$\begin{aligned} R &= \int_x \sum_{c=1}^{K} p(x,c)\, L[c, r(x)]\, dx \\ &= \int_x p(x) \sum_{c=1}^{K} p(c|x)\, L[c, r(x)]\, dx \end{aligned} \tag{4.2}$$

Für einen gegebenen Merkmalsvektor x werden die Kosten minimal, indem die folgende Entscheidungsregel gewählt wird (Bayes'sche Entscheidungsregel in der Mustererkennung):

$$x \longmapsto r(x) \quad := \quad \underset{k=1,\dots K}{\operatorname{argmax}} \left\{ \sum_{c=1}^{K} p(c|x) \cdot L[c,k] \right\} \tag{4.3}$$

Für die Entscheidungsregel ergibt sich dann:

$$r(x) \quad := \quad \underset{k=1,\dots K}{\operatorname{argmin}} \left\{ \sum_{c=1}^{K} \Big[p(c|x) \cdot L[c,k] \Big] \right\} \tag{4.4}$$

$$\begin{aligned} &= \underset{k=1,\dots K}{\operatorname{argmin}} \left\{ \sum_{c=1}^{K} \Big[p(c|x) \cdot (1 - \delta_{c,k}) \Big] \right\} \\ &= \underset{k=1,\dots K}{\operatorname{argmin}} \left\{ \sum_{c=1\, c\neq k}^{K} \Big[p(c|x) \Big] \right\} \\ &= \underset{k=1,\dots K}{\operatorname{argmin}} \left\{ \sum_{c=1}^{K} \Big[p(c|x) \Big] - p(k|x) \right\} \\ &= \underset{k=1,\dots K}{\operatorname{argmin}} \Big\{ 1 - p(k|x) \Big\} \\ &= \underset{k=1,\dots K}{\operatorname{argmax}} \Big\{ p(k|x) \Big\} \\ &= \underset{k=1,\dots K}{\operatorname{argmax}} \left\{ \frac{p(x,k)}{p(x)} \right\} \\ r(x) &= \underset{k=1,\dots K}{\operatorname{argmax}} \Big\{ p(x,k) \Big\} \end{aligned}$$

$$r(x) \quad = \quad \underset{k=1,\dots K}{\operatorname{argmax}} \Big\{ p(k)p(x|k) \Big\} \tag{4.5}$$

Zur Verwendung dieser Entscheidungsregel in realen Klassifikationsaufgaben müssen nun noch Modelle für $p(k)$ und $p(x|k)$ gewählt und deren Parameter aus den Trainingsdaten geschätzt werden. Für die Modellierung der a-priori-Wahrscheinlichkeit $p(k)$ werden in der Regel relative Häufigkeiten geschätzt. In einigen Erkennungsaufgaben wird auch einfach $p(k) = \frac{1}{K}$ gesetzt, also eine Gleichverteilung angenommen. So ist zum Beispiel bei der Ziffernerkennung nicht einzusehen, warum die Klasse '4' eine höhere a-priori Wahrscheinlichkeit haben soll als etwa die Klasse '7'. Anders ist dies bei der Buchstabenerkennung, da zum Beispiel bekannt ist, daß im Deutschen der Buchstabe 'e' der häufigste ist.

Für die Modellierung der klassenbedingten Wahrscheinlichkeit ist solch ein einfaches Modell unzureichend. Die klassenbedingte Wahrscheinlichkeit $p(x|k)$ wird daher über Gauß'sche Mischverteilungen modelliert, deren Parameter aus den Trainingsdaten geschätzt werden müssen (dies geschieht in der Regel unter Verwendung des Expectation-Maximization Algorithmus [Dahmen et al. 1998a]). In diesem Zusammenhang soll zunächst auf die Definition multivariater, multimodaler Gauß-Verteilungen eingegangen werden.

4.2 Multivariate Gaußverteilungen

Es wird angenommen, daß die Daten im wesentlichen gaußverteilt sind, das heißt, jede einzelne Komponente eines Merkmalsvektors schwankt um einen Mittelwert μ mit einer Varianz σ^2. Die Annahme der Gaußverteilung ist keine echte Einschränkung, da durch die im nächsten Abschnitt vorgestellte Gauß'sche Mischverteilung jede beliebige Verteilungsfunktion beliebig genau angenähert werden kann.

$$p(x_d|k) = \frac{1}{\sqrt{2\pi\sigma_{k,d}^2}} \exp\left[-\frac{1}{2}\left(\frac{x_d - \mu_{k,d}}{\sigma_{k,d}}\right)^2\right]$$

x_d	:	d-te Komponente des Merkmalsvektors
k	:	Klassenindex
$\sigma_{k,d}^2$	:	Varianz der d-ten Komponente der Klasse k
$\mu_{k,d}$	:	Mittelwert der d-ten Komponente der Klasse k

Setzt man voraus, daß die einzelnen Komponenten des Merkmalsvektors statistisch unabhängig sind, ergibt sich die Wahrscheinlichkeit des Merkmalsvektors als das Produkt der Einzelwahrscheinlichkeiten:

$$\begin{aligned} p(x|k) &= p(x_1, ..., x_d, ...x_D) = \prod_{d=1}^{D} p(x_d|k) && (4.6) \\ &= \frac{1}{\prod_{d=1}^{D}\sqrt{2\pi\sigma_{k,d}^2}} \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d - \mu_{k,d}}{\sigma_{k,d}}\right)^2\right] && (4.7) \end{aligned}$$

Allerdings sind die Pixel keine voneinander unabhängigen Merkmale, in der Regel sind insbesondere benachbarte Pixel korreliert. Diese Tatsache wird durch die Einführung einer Kovarianzmatrix berücksichtigt, so daß folgt:

$$p(x|k) = \frac{1}{\sqrt{(2\pi)^2 \det \Sigma_k}} \exp\left[-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)\right] \tag{4.8}$$

mit (hier klassenabhängiger) Kovarianzmatrix Σ_k

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N_k} (x_{nk}-\mu_k)(x_{nk}-\mu_k)^t \tag{4.9}$$

und N_k vielen Trainingsdaten x_{nk} aus der Klasse K. Gleichung 4.8 wird häufig auch 'Normalverteilung' genannt und daher im folgenden mit $\mathcal{N}(x|\mu_k, \Sigma_k)$ bezeichnet.

Da in der Realität auftretende Daten nur in den seltensten Fällen exakt gaußverteilt sind (also unter anderem unimodal), gehen wir im folgenden auf Gauß'sche Mischverteilungen ein. Diese bieten den Vorteil der multimodalen Modellierung, sind also zur Modellierung realer Daten in der Regel deutlich besser geeignet (vgl. etwa die auf USPS erzielten Ergebnisse für Einzelverteilungen mit denen für Mischverteilungen in Kapitel 6).

4.3 Gauß'sche Mischverteilungen

Wie bereits erwähnt, ist im Normalfall die Verteilung der Daten nicht so einfach, daß die Werte eines Merkmals genau um einen typischen Mittelwert streuen. So unterscheidet sich etwa die Draufsicht eines Diskozyten erheblich von dessen Seitenansicht, obwohl beide Bilder offensichtlich der Klasse 'Diskozyt' zuzuordnen sind. Um solche unterschiedlichen Häufungsbereiche von Merkmalen zu modellieren, werden Gauß'sche Mischverteilungen verwendet. Eine Gauß'sche Mischverteilung ist dabei definiert als Linearkombination von Gauß'schen Einzelverteilungen, so daß für die klassenbedingten Wahrscheinlichkeiten folgt:

$$p(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) \tag{4.10}$$

Dabei bezeichnet I_k die zur Modellierung der Klasse k verwendete Anzahl an Einzelverteilungen (die im folgenden auch kurz als 'Dichten' bezeichnet werden). Um zu gewährleisten, daß $p(x|k)$ wieder eine Verteilung ist, müssen die Gewichte c_{ki} die Bedingungen

$$c_{ki} > 0 \tag{4.11}$$

und

$$\sum_{i=1}^{I_k} c_{ki} = 1 \tag{4.12}$$

erfüllen. Weiterhin bezeichnen μ_{ki} bzw. Σ_{ki} den Mittelwert respektive die empirische Kovarianzmatrix der i-ten Dichte der Klasse k.

Da für Merkmalsvektoren $x \in \mathbb{R}^D$ zur Bestimmung einer einzigen Kovarianzmatrix $D \times D$ viele Parameter geschätzt werden müssen, kann nur in den seltensten Fällen mit einer vollen Kovarianzmatrix gearbeitet werden (da nur eine begrenzte Anzahl Trainingsdaten zur Verfügung stehen). Interpretiert man im Falle der roten Blutzellen jedes Pixel eines Bildes als Merkmal (*appearance* *based pattern recognition*), so müßten pro Kovarianzmatrix 128^4 Parameter geschätzt werden (da ein Bild aus 128×128 Pixeln besteht), wobei für jede Klasse nur etwa 1500 Bilder zur Verfügung stehen.

Da man im Falle von Mischverteilung prinzipiell für jede Dichte i einer Klasse k eine eigene Kovarianzmatrix Σ_{ki} schätzen müßte, ist es oft sinnvoll, die Anzahl der zu schätzenden Kovarianzmatrixen durch sogenanntes Varianzpooling zu reduzieren:

a) *klassenspezifisches Pooling:* Bestimme für jede Klasse k nur ein Σ_k,
also: $\Sigma_{ki} = \Sigma_k \quad \forall\, i \in I_k$.

b) *globales Pooling:* Bestimme nur eine einzige Kovarianzmatrix Σ
also: $\Sigma_{ki} = \Sigma \quad \forall\, k$ und $\forall\, i \in I_k$.

Darüberhinaus wird in unseren Experimenten nur mit diagonalen Kovarianzmatrizen (also mit Varianzvektoren) gearbeitet. Es ist zu beachten, daß auch mit Gauß'schen Mischverteilungen dieser Form noch jede Verteilungsfunktion beliebig genau approximiert werden kann.

4.4 Dimensionsreduktion

Eine weitere offensichtliche Möglichkeit der Parameter-Reduktion bieten Verfahren zur Dimensionsreduktion. Deren Aufgabe ist es, hochdimensionale Merkmalsvektoren in einer für die Mustererkennung geeigneten Art und Weise in einen niedrigdimensionalen Merkmalsraum zu projizieren. Im folgenden werden 2 populäre Verfahren zur Dimensionsreduktion, nämlich die Karhunen-Loève-Transformation und die lineare Diskriminanzanalyse, vorgestellt. Diese unterscheiden sich im wesentlichen durch die bei der Dimensionsreduktion verfolgten Ziele. Weiterhin werden wir eine numerisch robustere Variante der LDA vorstellen, die im folgenden als 'Subspace-Methode' bezeichnet werden wird.

4.4.1 Karhunen-Loève-Transformation

Wie schon im vorherigen Abschnitt angesprochen, ist die Dimensionsreduktion der Daten von großer Bedeutung, da in der Regel nicht genügend Trainingsdaten zur Verfügung stehen. Das Ziel der Karhunen-Loève-Transformation (KLT) [Schukat-Talamazzini, 1995] ist es, den Rekonstruktionsfehler zwischen dem Originalvektor und dem aus dem dimensionsreduzierten Vektor zurückgewonnenen Vektor zu minimieren. Im Gegensatz zur linearen Diskriminanzanalyse wird die anschliessende Klassifikation der Daten nicht berücksichtigt. Trotz dieser Tatsache wird die KLT häufig als Vorverarbeitung für Klassifikationszwecke eingesetzt, da sie leicht zu berechnen ist und darüberhinaus eine Dekorrelation der Merkmale vornimmt. Die Kovarianzmatrix der KLT-transformierten Daten ist eine Diagonalmatrix.

Ausgehend von der Matrix Φ, deren Spaltenvektoren Φ_i eine Orthonormal-Basis des $\mathbb{R}^N$ bilden, das heißt

$$\Phi_i^T \, \Phi_j \quad = \quad \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \tag{4.13}$$

ergibt sich eine Basistransformation mit dieser Matrix Φ und die Rücktransformation der folgenden Form:

$$\dot{x} = \Phi^T x \tag{4.14}$$
$$x = \Phi \dot{x} \tag{4.15}$$

Hierbei ist x der Ausgangsvektor und $\dot{x}$ der transformierte Vektor. Wird nun die Matrix Φ durch Weglassen von Spaltenvektoren verkleinert, (in Φ^T entfallen Zeilen) so ergibt sich die Reduktionsmatrix Ψ^T mit $\Psi \subset \Phi$, $\Psi \in \mathbb{R}^{M \times N}$,$\Phi \in \mathbb{R}^{N \times N}$ und $M < N$. Die Dimensionsreduktion und die Rücktransformation ergeben sich nun durch 4.16 und 4.17.

$$\hat{x} = \Psi^T x \tag{4.16}$$
$$\check{x} = \Psi \hat{x} \tag{4.17}$$

Wird ein höherdimensionaler Vektor $x \in \mathbb{R}^N$ in einen niederdimensionalen Vektor $\hat{x} \in \mathbb{R}^M$ transformiert, so ergibt sich bei der Rücktransformation ein vom Originalvektor x abweichender Vektor $\check{x} \in \mathbb{R}^N$. Der Fehler, der sich daraus ergibt, läßt sich berechnen aus

$$e = \sum_{n=1}^{N} (x_n - \check{x}_n)^2 \tag{4.18}$$

Es soll nun eine optimale Transformation gesucht werden, so daß dieser Fehler minimal wird. Um diese optimale Transformation zu finden, muß zuerst die empirische Kovarianzmatrix Σ der Stichprobe geschätzt werden. Die Minimierung des Rekonstruktionsfehlers e bezüglich der Transformationsmatrix Ψ führt auf die Lösung des Eigenwertproblems $\mu\Sigma\Phi_i = \lambda\Phi_i$. Dieses läßt sich am besten mit iterativen Verfahren lösen. Die positiv-definite Matrix besitzt dann die Zerlegung :

$$\Sigma = \Phi^T \Lambda \Phi, \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \vdots & \\ 0 & & \lambda_D \end{pmatrix}$$

mit durchgehend positiven Eigenwerten. Der zu erwartende Fehler der unvollständigen Entwicklung beträgt aufgrund der Orthonormalitätseigenschaften des Basissystems

$$e_M = \sum_{i=M+1}^{N} \Psi_i^T \Sigma \Psi_i = \sum_{i=M+1}^{N} \Psi_i^T \ \Psi_i = \sum_{i=M+1}^{N} e_i^T \Lambda e_i = \sum_{i=M+1}^{N} \lambda_i$$

Der erwartete Fehler ergibt sich aus der Summe derjenigen Eigenwerte, deren Eigenvektor nicht in die Reduktionsmatrix Ψ eingeht. Werden nun die Spalten der Matrix Φ mit den Eigenvektoren der Kovarianzmatrix absteigend nach der Größe der Eigenwerte sortiert, so ergibt sich für e_i der minimale Fehler.

Zur Veranschaulichung der Wirkungsweise der Karhunen-Loève-Transformation soll das folgende Beispiel dienen (vergleiche Abbildung 4.1):

Die Hauptachsen bezüglich der Varianz der Punktmenge werden auf die Koordinatenachsen gedreht. Auf die x_1-Achse wird die größte Varianz gedreht, sie ist für den Repräsentationsfehler die

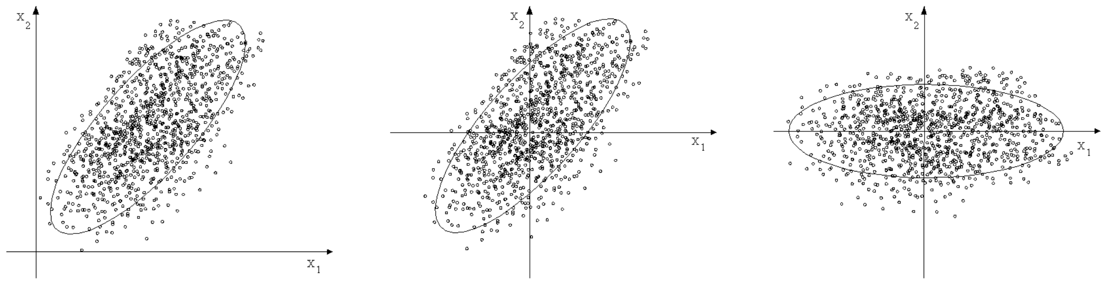

Abbildung 4.1: Beispiel: Translation und Rotation einer 2D Punktwolke mittels KLT

wichtigste Richtung. Die andere Dimension wird in die weniger wichtigen Varianzrichtungen gedreht. Die Dimensionen werden nach ihrer Wichtigkeit für den Repräsentationsfehler absteigend sortiert. Die Komponenten der Merkmale sind im transformierten Koordinatensystem voneinander unabhängig, so daß ihre Kovarianzmatrix diagonal ist. Die Klasseninformation der Merkmalsvektoren geht nicht in die Transformation ein.

Zusammenhang zur Whitening Transformation

Aufbauend auf der KLT läßt sich die sogenannte Whitening-Transformation definieren. Diese erreicht durch eine Skalierung der Eigenvektoren, daß die Kovarianzmatrix der reduzierten Merkmale nicht nur diagonal, sondern sogar die Einheitsmatrix ist. Man kann zeigen, daß dies durch die Division jeder Eigenvektorkomponente durch die Wurzel aus dem zugehörigen Eigenwert erreicht wird. Die Whitening Transformation wird im folgenden für uns interessant werden, da die im nächsten Abschnitt vorgestellte lineare Diskriminanzanalyse für 'weiße' Daten recht einfach zu berechnen ist. Diese Überlegung führt uns dann zur am i6 entwickelten Subspace-Methode.

4.4.2 Die lineare Diskriminanzanalyse

Die Karhunen-Loève-Transformation liefert gute Resultate, wenn es um Minimierung des Rekonstruktionsfehlers geht, ist aber nur bedingt geeignet, wenn es um Klassentrennbarkeit geht. In diesem Fall erweist sich die lineare Diskriminanzanalyse (LDA, [Schukat-Talamazzini, 1995]) der KLT oft als überlegen. Ein einfaches Beispiel ist das *Adidas*-Problem, dargestellt in Abbildung 4.2. Die drei Klassen sind durch Rotation der Punktwolke um 50^0 gut trennbar (also Rotation der Daten in eine Lage parallel zur x_2 Achse, dann eine Projektion auf die x_1-Achse). Wird die Punktwolke allerdings mit der KLT transformiert, so wird die Richtung der größten Varianz auf die x_1-Achse gedreht: eine Klassentrennung der transformierten Merkmalsvektoren ist dann nicht mehr möglich.

Um eine geeignete Transformation zu erhalten, muß die Klasseninformation der Lernstichprobe mit eingebracht werden. So können Merkmale extrahiert werden, die für die Klassifikation geeignet sind. Es wird eine Abbildung angestrebt, welche die Abstände der Merkmale innerhalb einer Klasse gleich läßt, die Mittelwerte der Klassen aber möglichst weit auseinanderzieht. Diese Abbildung - die

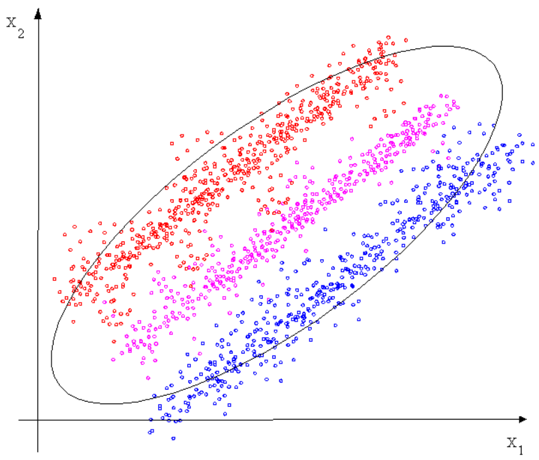

Abbildung 4.2: Das Adidas-Problem

im folgenden lineare Diskriminanzanalyse (LDA) genannt wird - basiert auf den folgenden Größen:

$$S_w = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N_k} (x_n - \mu_k)(x_n - \mu_k)^T \tag{4.19}$$

$$S_b = \frac{1}{N} \sum_{k=1}^{K} N_k \cdot (\mu_k - \mu)(\mu_k - \mu)^T \tag{4.20}$$

Hierbei ist S_w die Streuungsmatrix innerhalb der Klassen (Within-Class-Scatter Matrix) und S_b die Streuungsmatrix zwischen den Klassen (Between-Class-Scatter Matrix). Die für die Berechnung benötigten Werte sind die Anzahl der Klassen K, die Anzahl der Beobachtungen für jede Klasse N_k, der Gesamtmittelwertsvektor μ und die Mittelwertsvektoren der einzelnen Klassen μ_k und natürlich die Beobachtungsvektoren x_n.

Ähnlich wie im Fall der KLT läßt sich nun zeigen, daß das Bestimmen einer bezüglich der Klassentrennbarkeit optimalen linearen Projektion erneut auf das Lösen eines Eigenwertproblems hinausläuft. Im Fall der LDA müssen jedoch die Eigenwerte der Matrix $S_w^{-1} S_b$ berechnet werden (analog kann auch das verallgemeinerte Eigenwertproblem

$$B \cdot v_d = \lambda_d \cdot W \cdot v_d \tag{4.21}$$

gelöst werden, mit Eigenwerten $\lambda_d \in \mathbb{R}, d = 1, ..., D$ und Eigenvektoren $v_d \in \mathbb{R}^D$).

Analog zur KLT wählt man nun die dominierenden Eigenvektoren zum Erstellen der gesuchten Transformationsmatrix. Man beachte, daß - gegeben K Klassen - die maximale Zieldimension der reduzierten Daten $K - 1$ ist, da der Rang von B maximal $K - 1$ ist (der globale Mittelwert μ ist linear abhängig von den klassenspezifischen Mittelwerten μ_k).

Wie die Invertierung der Within-Class-Scatter Matrix bereits vermuten läßt, ist die Berechnung der LDA - insbesondere für den Fall einer kleinen Anzahl von Trainingsdaten - oft numerisch problematisch. Im folgenden Kapitel gehen wir daher auf eine numerisch robustere Variante der LDA ein. Wie im Ergebniskapitel gezeigt wird, erzielte diese LDA-Variante auf den nicht vervielfachten USPS Daten (also unter Verwendung von 7291 Trainingsdaten) eine bessere Fehlerrate als die klassische LDA (berechnet unter Verwendung von LAPACK Routinen).

4.4.3 Die Subspace Methode

Die Subspace Methode ist ein am i6 vorgeschlagenes Verfahren, das sich als numerisch stabiler erwiesen hat als die 'konventionelle' LDA. Für den Spezialfall, daß die Within-Class Scatter Matrix S_W die Einheitsmatrix I ist, reduziert sich das Eigenwertproblem 4.21 auf das Bestimmen einer Orthogonalbasis des Vektorraums, der von den Differenzen zwischen den klassenspezifischen Mittelwerten μ_k und dem Gesamtmittelwert μ aufgespannt wird [Ney 1998]. Es werden Prototypvektoren der Form

$$(\mu_k - \mu); k = 1, ..., K \tag{4.22}$$

betrachtet (bzw. der von diesen aufgespannte Unterraum).

Für die Berechnung der LDA ergibt sich daher die folgende, alternative Vorgehensweise die im weiteren Verlauf als 'Subspace-Methode' bezeichnet wird:

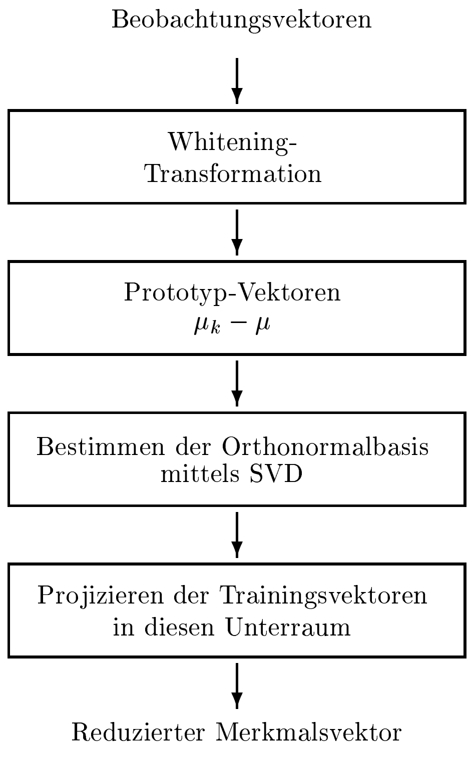

Abbildung 4.3: Struktur der Dimensionsreduktion

Im ersten Schritt wird unter Verwendung der Whitening-Tranformation die Within-Class-Scatter Matrix S_W auf 'Normalform' gebracht (also $S_W = I$). Dies wird erreicht, indem zunächst die Within-Class-Scatter Matrix der Daten berechnet wird. Anschließend werden deren Eigenwerte und die Eigenvektoren berechnet. Die Eigenvektoren werden nach der Größe der Eigenwerte sortiert, welches der oben beschriebenen KLT entspricht. Zusätzlich werden nun allerdings noch die Eigenvektoren durch die Wurzel ihrer zugehörigen Eigenwerte dividiert. Die mit der so entstehenden Matrix transformierten Trainingsdaten haben nun eine Einheitsmatrix als Within-Class-Scatter Matrix. Der oben beschriebene 'Spezialfall' der LDA wird also 'erzwungen'.

Da in einigen Fällen einige der Eigenwerte gleich 0 sind, können diese im folgenden nicht betrachtet werden, so daß sich schon in diesem Schritt eine geringfügige Dimensionsreduktion ergeben kann.

Im Folgeschritt werden nun Prototypvektoren der oben beschriebenen Form gebildet (man beachte die Verwandtschaft zur Between-Class-Scatter Matrix). Für den von diesen Vektoren aufgespannten Vektorraum wird nun eine Orthonormalbasis bestimmt. Dies geschieht unter Verwendung der sogenannten Singular Value Decomposition (SVD) [Press et al. 1992], da klassische Verfahren wie etwa die Gram-Schmidt-Orthogonalisierung auf dem mehrfachen Lösen von hochdimensionalen Gleichungssystemen beruhen und aus diesem Grund numerisch äußerst instabil sind.

Die Projektion der Daten in den so aufgespannten Raum liefert dann die gewünschte Dimensionsreduktion, wobei die singulären Werte der SVD nun die Rolle der Eigenwerte übernehmen. Wie bei der LDA erhalten wir auch bei dieser Methode maximal $K - 1$ Merkmale.

Die Dimensionsreduktion auf $K - 1$ ist oft zu stark, insbesondere wenn - wie im Fall der roten Blutkörperchen - nur sehr wenige Klassen gegeben sind (hier 5 bzw. 3). Ein Vektorraum dieser Dimension kann die ursprüngliche Verteilung der Daten allerdings nicht mehr hinreichend genau beschreiben (eine Dimensionsreduktion hat immer einen Informationsverlust zur Folge, siehe dazu [Ney 1998]). Um nun eine höhere Dimensionalität der Vektoren zu erreichen, werden die Klassen künstlich in Unterklassen eingeteilt, die sogenannten *Pseudoklassen*. Die Trainingsdaten werden mit Hilfe des EM-Algorithmus [Dempster et al. 1977] geclustert und jeder dieser Cluster als eigene Klasse für die Subspace-Methode interpretiert. Im Falle der Erythrozyten wurde jede Klasse in 8 Pseudoklassen geteilt und so eine Reduktion der Daten von einem 128^2 dimensionalen Raum in einen 39-dimensionalen Merkmalsraum erreicht.

Aufbauend auf diesen Merkmalen wurden dann die Mischverteilungsparameter trainiert. Auf das Training der Parameter soll hier nicht weiter eingegangen werden. Interessenten finden genauere Informationen zu diesem Schritt in [Dahmen et al. 2000a] oder in [Ney 1998]. Allerdings wurde die Dimensionsreduktion nicht in jedem Fall auf die Originaldaten angewendet, da das Thema dieser Diplomarbeit die Evaluierung von invarianten Merkmalen ist. Im folgenden Kapitel werden nun die untersuchten Methoden zur Erzeugung invarianter Merkmale vorgestellt. Die Dimensionsreduktion wird dann in der Regel - soweit nicht ausdrücklich anders erwähnt - auf die extrahierten, invarianten Merkmale angewendet.

Kapitel 5

Invariante Merkmale

In diesem Kapitel widmen wir uns der Extraktion von invarianten Features. Auf die Bedeutung einer invarianten Erkennung von Bildobjekten muß wohl nicht näher eingegangen werden, da offensichtlich nur in den seltensten Fällen von normierten Bilddaten ausgegangen werden kann. Im folgenden Abschnitt beschäftigen wir uns zunächst mit der Fourier-Transformation basierten Extraktion von Merkmalen, die rotations-, skalierungs- und translations-invariant (RST-invariant) sind. Die beschriebene Methode wird in der Literatur auch FOURIER-MELLIN Transformation genannt [Reddy & Chatterji 1996].

5.1 Die Fourier-Mellin-Transformation

Da die zu erkennenden Objekte in der Regel nicht zentriert sind, ist es sinnvoll, Merkmale zu benutzen, die translationsinvariant sind. Das heißt, daß ein Bild das um ein Stück verschoben ist, auf die gleichen Merkmale abgebildet wird wie das Originalbild. Um solche Merkmale zu extrahieren wird die Translationsinvarianz des Betragsspektrums der Fourier-Transformation ausgenutzt [Reddy & Chatterji 1996]. Im folgenden werden wir der Einfachheit halber die eindimensionale, kontinuierliche Fourier-Transformation betrachten. Die erzielten Erkenntnisse lassen sich in ähnlicher Weise allerdings auch für die zweidimensionale diskrete Fourier-Transformation (2D-DFT) herleiten.
Die Fourier-Transformation $H(\omega)$ für ein kontinuierliches Signal $h(t)$ ist definiert durch

$$\mathcal{F}\{h(t)\} = H(\omega) := \int_{-\infty}^{\infty} h(t)e^{-i\omega t}dt \tag{5.1}$$

so daß sich für die Fourier-Transformation der um den Wert t_0 verschobenen Funktion, die folgende Fourier-Transformierte ergibt:

$$\mathcal{F}\{h(t-t_0)\} \quad = \quad \int_{-\infty}^{\infty} h(t-t_0)e^{-i\omega t}dt \tag{5.2}$$

$$= e^{-i\omega t_0} \int_{-\infty}^{\infty} h(t-t_0) e^{-i\omega(t-t_0)} dt$$

mit $\tau = t - t_0$

$$= e^{-i\omega t_0} \int_{-\infty}^{\infty} h(\tau) e^{-i\omega\tau} d\tau$$

$$\mathcal{F}\{h(t-t_0)\} = e^{-i\omega t_0} H(\omega) \tag{5.3}$$

mit:

$$|e^{-i\omega t_0}| = \sqrt{(\cos(-\omega t_0))^2 + (\sin(-\omega t_0))^2}$$
$$= \sqrt{(\cos(\omega t_0))^2 + (-\sin(\omega t_0))^2} = 1$$

folgt:

$$|\mathcal{F}\{h(t-t_0)\}| = |\mathcal{F}\{h(t)\}| \tag{5.4}$$

mit komplexwertigem $i = \sqrt{-1}$.

Durch die obigen Umformungen ergibt sich, daß die Fourier-Transformierte der verschobenen Funktion sich nur um einen imaginären Faktor (Phasenfaktor) von der Fourier-Transformierten der nicht verschobenen unterscheidet (Formel: 5.3). Da der Betrag dieses komplexwertigen Faktors gleich 1 ist, ergibt sich in 5.4, daß der Betrag der Fourier-Transformierten (das sogenannte Amplitudenspektrum) translationsinvariant ist.

Die Verwendung des Amplitudenspektrums anstelle des Originalbildes erlaubt bereits translationsinvariante Mustererkennung. Da das Amplitudenspektrum eines reellwertigen Eingangssignals symmetrisch ist, reicht es für den Fall reellwertiger Eingabebilder (wie etwa die von uns betrachteten USPS bzw. Blutzellen Daten) aus, nur eine Hälfte des Amplitudenspektrums zu verwenden.

Neben der Translationsinvarianz ist die Skalierungs- und Rotationsinvarianz noch von besonderer Bedeutung. Besonders letztere ist für die Erythrozyten wichtig, da diese frei in der Lösung sedimentieren und daher in den entsprechenden Bilddaten beliebig orientiert sein können. Um rotationsinvariante Merkmale zu erhalten, werden die Rotation und Skalierung des Originalbildes auf Translationen (des bereits verschiebungs-invarianten Amplitudenspektrums) zurückgeführt. Dies geschieht durch Umrechnen der kartesischen in Polar-log-Koordinaten.

$$(\varphi, r) := R(x,y) = (\arctan(\frac{y}{x}), \frac{1}{2} ln(x^2+y^2)) \tag{5.5}$$

Die Position jedes Pixels wird vom kartesischen ins Polarkoordinatensystem umgerechnet, dabei werden alle Punkte die den gleichen Abstand zum Mittelpunkt haben (also auf einem Kreis liegen), auf eine Gerade im Polarkoordinatensystem abgebildet. Eine Rotation des Ausgangsbildes wird so in eine Translation in der φ-Achse des Polarkoordinatensystems überführt (vgl. Abbildung 5.1).

Der Abstand zum Mittelpunkt repräsentiert dabei die Größe des Objektes. Durch logarithmieren der Radius-Achse kann eine Multiplikation in eine Addition umgewandelt werden (Rechenschieber-Prinzip, Logarithmengesetz),

$$ln(x * y) = ln(x) + ln(y) \tag{5.6}$$

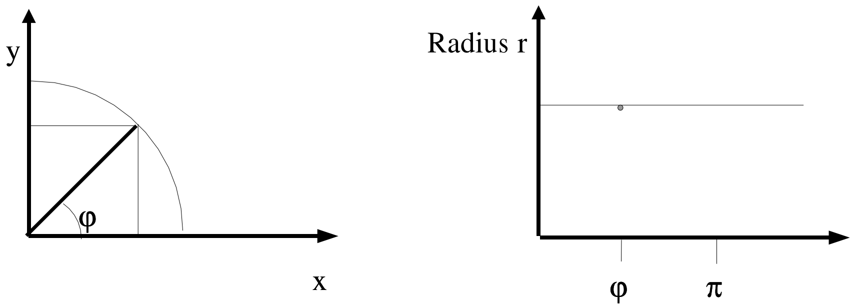

Abbildung 5.1: Kartesische vs. Polarkoordinaten

was dazu führt, daß auch eine Skalierung des Bildes auf eine Translation zurückgeführt wird, diesmal entlang der $r - Achse$. Deutlich wird das Vorgehen und die Rotationsinvarianz im folgenden (nun 2-dimensionalen) Beispiel, in dem ein Bild eines Diskozyten in einen RST-invarianten Merkmalsvektor transformiert wird. Da die Pixel, der während dieses Prozesses entstehenden Bilder, in recht unterschiedlichen Wertebereichen liegen, sind für jedes einzelne Bild Farbtafeln angegeben, die eine Zuordnung von Grauwerten zu einem bestimmten Wertebereich erlauben.

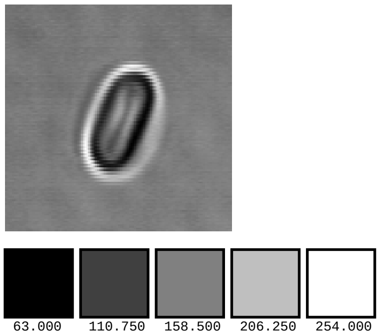

Abbildung 5.2: Beispiel: Ausgangsbild aus Mikroskop

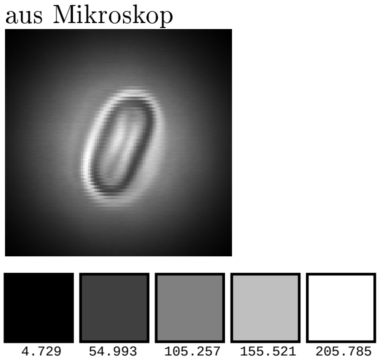

Abbildung 5.3: Mit Gaußglocke gefenstertes Bild

In Abbildung 5.2 ist ein Beispiel für ein Bild zu sehen, wie es durch das Mikroskop aufgenommen wurde.
Das Bild 5.2 wird mit einem Gaußglocke multipliziert (gefenstert), um an den Bildrändern einen homogenen Übergang zu erhalten (da für die Fourier-Transformation das Bild als unendlich fortgesetzt interpretiert wird). Die eigentliche Bildinformation wird in der Mittte des Bildes erwartet. Es ergibt sich Abbildung 5.3.
Das Bild wird im nächsten Schritt der 2-DFT unterworfen, d.h.:

$$\mathcal{F}(x,y) := \frac{1}{MN} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n,m) e^{-i\frac{2\pi}{N}xn} e^{-i\frac{2\pi}{M}ym} \tag{5.7}$$

mit $x = 0, 1, 2 ... N-1$ und $y = 0, 1, 2 ... M-1$
Der helle Bereich im Bildzentrum steht für die niedrigen Frequenzen im Originalbild. Die stärkere Ausprägung des Bildes 5.3 in senkrechter Richtung macht sich durch die hellen Bereiche oben und unten in Bild 5.4 bemerkbar.

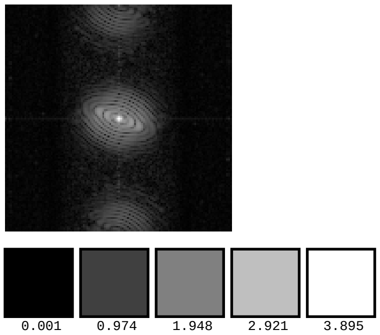

Abbildung 5.4: Betrag der Fourier-Transformierten des Bildes

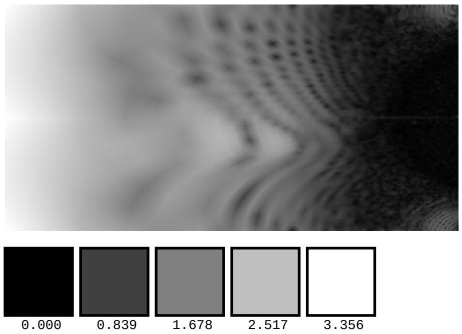

Abbildung 5.5: Polar-log Transformierte des Amplitudenspektrums aus 5.4

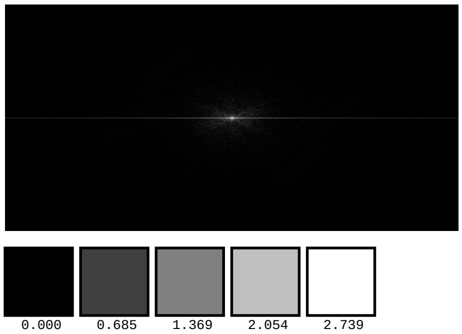

Abbildung 5.6: Betrag der Fourier-Transformierten des Polar-log-Bildes

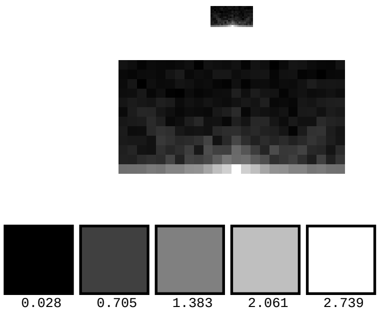

Abbildung 5.7: Ausschnitt des Zentrums

Da das Ausgangsbild 5.3 reellwertig ist, ist die Fouriertransformierte punktsymmetrisch [Ney 1998]. Die Fourier-Transformation liefert komplexe Werte. In deren Polardarstellung enthält die Amplitude die Information über die vorkommenen Frequenzen, nur sie wird für die weitere Merkmalsgewinnung verwendet, da nur die Amplitute translationsinvariant ist. Die Winkel der komplexen Werte entsprechen den Phasenverschiebungen der Frequenzen, diese Werte sind für die Rücktransformation des Bildes von großer Bedeutung, werden aber - da sie nicht translationsinvariant sind - im folgenden nicht weiter verwendet.

Abbildung 5.4 wird durch Polar-logtransformation in 5.5 überführt. Im linken Bereich der hellen Werte von Bild 5.5 sind die niedrigen Frequenzen des Ursprungbildes, die linke Spalte entspricht dem Gleichanteil (also der Durchschnittshelligkeit des Bildes). Durch das Logarithmieren werden die niedrigeren Frequenzen stärker betont, da sie einen größeren Anteil am Bild ausmachen. Die hohen Frequenzen sind weniger fein abgetastet, was sinnvoll ist, da in vielen Fällen nur die niedrigen Frequenzen die wesentliche Information tragen. In den hochfrequenten Bildanteilen sind oftmals unerwünschte Rauschanteile und weitere Störungen enthalten.

Da die Fourier-Transformierte punktsymmetrisch ist, wurde nur eine Hälfte des Ausgangsbildes in das Polarkoordinaten System umgerechnet.

Dieses Bild ist für die weitere statistische Mustererkennung noch zu hochdimensional. Glücklicherweise ist das Polar-log-Bild reellwertig, so daß dessen Fourier-Transformation erneut punktsymmetrisch ist und es somit ausreicht, nur eine Hälfte des Bildes als Merkmale zu nutzen. Da die hohen Frequenzen darüberhinaus auch hier nur von untergeordneter Bedeutung sind, werden sie vernachlässigt. Dies geschieht durch das Ausschneiden des Zentrums der Fourier-Transformation des Polar-log-Bildes (in unseren Experimenten wurde das beste Ergebnis mit einer 'Fenstergröße' von 24×12 Pixeln erzielt). Das Ergebnis dieses Ausschneidens wird in Abbildung 5.7 dargestellt. Oben ist der Ausschnitt in der vergleichbaren Größe mit den anderen Bildern dargestellt, unten vergrößert.

Um die Rotationsinvarianz zu veranschaulichen wurde das Bild 5.2 um 90^0 rotiert (siehe Abb. 5.8), und die RST-Merkmale berechnet. Die Zwischenschritte sind in Abb. 5.9 bis Abb. 5.11 zu verfolgen. Im Vergleich der beiden Bildreihen (Abb. 5.2 - 5.7) und (Abb. 5.8 - 5.12) ist gut zu sehen wie die Rotationsinvarianz erreicht wird. Werden die Abbildungen 5.4 und Abb. 5.9 verglichen, so sieht man, daß auch die Fourier-Transformation der Bilder um 90^0 gedreht ist (allgemein bewirkt eine Rotation des Originalbildes um einen Winkel α eine Rotation der Fourier-Transformierten um $-\alpha$).
Durch die Transformation der Bilder in ein Polar-log-Koordinatensystem wird diese Drehung in eine Verschiebung umgewandelt. In Abbildung 5.5 und 5.10 ist folgerichtig eine zyklische Verschiebung zu erkennen. Die Ausbuchtung die in Abb.5.5 mittig ist, liegt in Abb. 5.10 am oberen Rand. Beide Bilder sind also durch einen vertikalen Verschiebung (mit wraparound) ineinander überführbar.
Die Merkmalsvektoren (Abb. 5.6 und 5.11) unterscheiden sich nur sehr gering (was durch Rundungsfehlern zu erklären ist) , in Abb. 5.13 ist die Differenz zwischen den Abbildungen 5.6 und 5.11 dargestellt. Besonders zu beachten ist der sehr kleine Wertebereich dieser Differenzen.

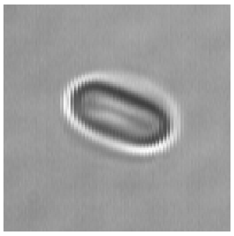

Abbildung 5.8: Um 90^0 gedrehtes Bild

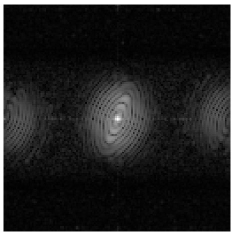

Abbildung 5.9: Amplitudenspektrum zu Abb. 5.8

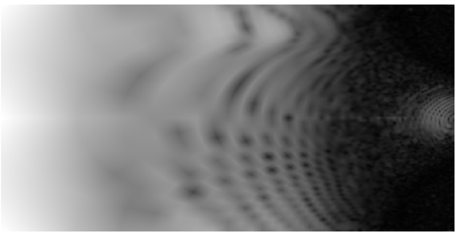

Abbildung 5.10: Polar-log-Bild zu Abbildung 5.9

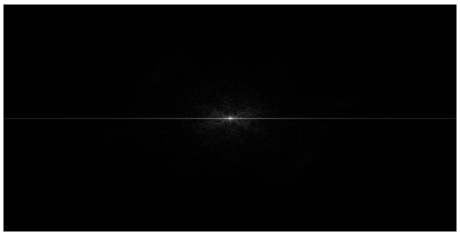

Abbildung 5.11: Amplitudenspektrum des Polar-log-Bildes aus Abbildung 5.10

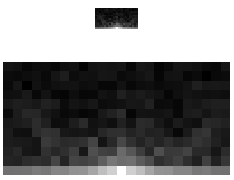

Abbildung 5.12: Ausschnitt aus dem Zentrum

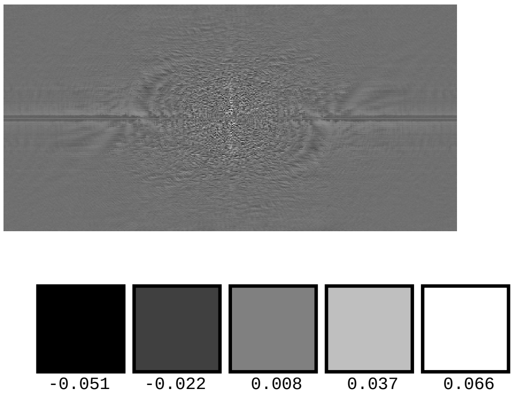

Abbildung 5.13: Differenz der Bilder Abb. 5.6 und Abb. 5.11

5.2 Invariante Momente

In der Bildverarbeitung werden oft als invariante Merkmale sogenannte Momente verwendet. Dieses Verfahren wurde von Hu [Hu 1962] vorgeschlagen. Die Momente sind aus der Physik entnommen und entsprechen dort beispielsweise der Fläche oder dem Schwerpunkt eines Objektes oder dem Trägheitsmoment. In der Bildverarbeitung werden diese Momente oft als *geometrische* Momente bezeichnet. Die Momente sind für ein kontinuierliches Bild definiert durch:

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p \, y^q \, f(x,y) \, dx \, dy \tag{5.8}$$

Wobei x und y die Bildachsen sind und $f(x,y)$ der Grauwert des Bildpixels an der Position (x,y). Für diskrete Bilder der Größe $I \times J$ Pixel ergibt sich statt des Doppelintegrals eine Doppelsumme mit:

$$m_{p,q} = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} i^p \, j^q \, f(i,j) \tag{5.9}$$

Diese Momente $m_{p,q}$ können anschaulich interpretiert werden als:

$m_{0,0}$	Fläche des Objektes
$m_{1,0}/m_{0,0}$	X-Koordinate des Schwerpunktes
$m_{0,1}/m_{0,0}$	Y-Koordinate des Schwerpunktes
$m_{2,0}, m_{0,2}$	Trägheitsmoment des Objektes bezüglich der Koordinatenachsen
$m_{2,0} + m_{0,2}$	Polares Trägheitsmoment des Objektes bezüglich des Koordinatenursprungs

Für höhere Werte von p und q ist eine anschauliche geometrische Interpretation nicht länger möglich. Es ist im allgemeinen sinnvoll, die Momente bis zur Ordnung 4 zu berecnhen $(p,q \leq 3)$. Damit die Momente translationsinvariant werden, muß der Schwerpunkt des Bildes zur Momentenberechnung auf den Koordinatenursprung verschoben werden, so daß folgt:

$$\mu_{p,q} = \sum_{i,j} \sum_{\in \mathbb{R}} \left(i - \frac{m_{1,0}}{m_{0,0}}\right)^p \left(j - \frac{m_{0,1}}{m_{0,0}}\right)^q f(i,j) \tag{5.10}$$

Um Größenunterschiede herauszurechnen, zur Extraktion skalierungsinvarianter Merkmale, werden diese Momente durch die Objektgröße dividiert. Es ergeben sich dann die translations- und skalierungsinvarianten Momente

$$\eta_{p,q} = \mu_{p,q} / \mu_{0,0}^{\frac{p+q}{2}+1} \tag{5.11}$$
$$p + q = 2, 3 ....$$

Von Hu [Hu 1962] wurden jedoch folgende, zusätzlich noch rotationsinvariante, Merkmale vorgeschlagen:

$$\phi_1 = \eta_{2,0} + \eta_{0,2} \tag{5.12}$$

$$\phi_2 = (\eta_{2,0} + \eta_{0,2})^2 + 4\eta_{1,1}^2 \tag{5.13}$$

$$\phi_3 = (\eta_{3,0} + 3\eta_{1,2})^2 + (3\eta_{2,1} - \eta_{0,3})^2 \tag{5.14}$$

$$\phi_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2 \tag{5.15}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{3,0} + 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})\,[\,(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} - \eta_{0,3})^2\,] \\ & +(3\eta_{2,1} - 3\eta_{0,3})(\eta_{2,1} + \eta_{0,3})\,[\,3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} - \eta_{0,3})^2\,]\end{aligned} \tag{5.16}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{2,0} - \eta_{0,2})\,[\,(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2\,] \\ & +4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} - \eta_{0,3})\end{aligned} \tag{5.17}$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})\,[\,(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1}) + \eta_{0,3})^2\,] \\ & -(\eta_{3,0} - 3\eta_{1,2})(\eta_{2,1} + \eta_{0,3})\,[\,3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1}) + \eta_{0,3})^2\,]\end{aligned} \tag{5.18}$$

Diese Momente nach Hu wurden auf ihre Güte bezüglich der Klassifikationseigenschaften getestet. Die erzielten Ergebnisse befinden sich im Kapitel 6.

Bevor wir uns der Diskussion der Ergebnisse zuwenden, soll noch ein letztes Verfahren vorgestellt werden, daß im Zuge dieser Arbeit verwendet wurde, nämlich die Tangentendistanz nach Simard.

5.3 Ein Tangenten-Distanz basiertes Verfahren

Im Jahre 1993 stellte Simard ein invariantes Distanzmaß vor, genannt *Tangentendistanz*, das sich im Bereich der Schrifterkennung als sehr effektiv herausstellte [Simard et al. 1998]. Er bemerkte dabei zunächst, daß entsprechend kleine Transformationen bestimmter Objekte (wie zum Beispiel Schriftzeichen) 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

$$M_x = \{t(x, \alpha) : \alpha \in \mathbb{R}\} \subset \mathbb{R}^{I \times J} \tag{5.19}$$

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:

$$\hat{M}_x = \{x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) : \alpha_l \in \mathbb{R}\} \subset \mathbb{R}^{I \times J} \tag{5.20}$$

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

$$d_T(x, x_{nk}) = \min_{\bar{\alpha}} \{\|x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) - x_{nk}\|^2\}, \tag{5.21}$$

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 [Simard et al. 1998]. In Abbildung 5.14 sind Beispielbilder dargestellt, die mit (5.20) 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:

$$d_T(x, x_{nk}) = \min_{\bar{\alpha}, \bar{\beta}} \{\|x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) - x_{nk} - \sum_{l=1}^{L} \beta_l \cdot T_l(x_{nk})\|^2\} \tag{5.22}$$

Allerdings steigt dabei der Berechnungsaufwand dramatisch an, ohne zu einer bedeutenden Verbesserung der Klassifikation zu führen [Simard et al. 1998]. In den im Rahmen dieser Arbeit durchgeführten Experimenten wurden die Tangentenvektoren für Translationen (in x und y-Richtung), Rotation, Skalierung und Achsen-Deformationen (ebenfalls in x und y-Richtung) wie von Simard vorgeschlagen berechnet.

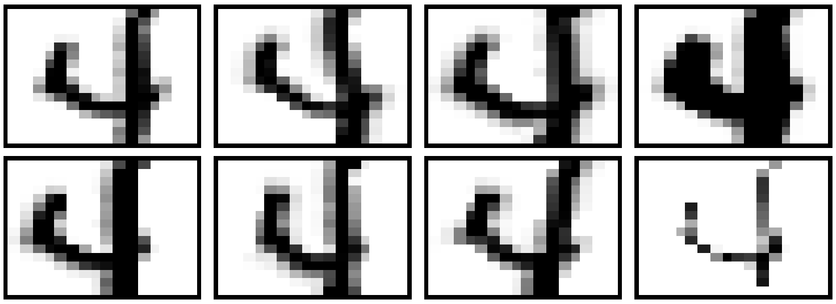

Abbildung 5.14: Durch Linearkombination der Tangentenvektoren erzeugte Modfifikationen des Originalbildes (oben links)

In dieser Arbeit wurden nicht die Abstände zwischen den Mannigfaltigkeiten berechnet, sondern der Ansatz verfolgt, die Tangentenvektoren, die diese Mannigfaltigkeiten aufspannen, als zusätzliche Merkmale zu verwenden.

Diese Tangentenvektoren lassen sich einfach - wie etwa in [Vapnik 1998] beschrieben, über Differenzenquotienten (im diskreten Fall) bzw. über Richtungsableitungen berechnen (vergleiche Abbildung 5.15). Ohne Beweis seien hier die Berechnungsvorschriften für die betrachteten Tangentenvektoren angegeben, ausgehend von der Darstellung einer allgemeinen affinen Transformation mit den Parametern $a, \ldots f$:

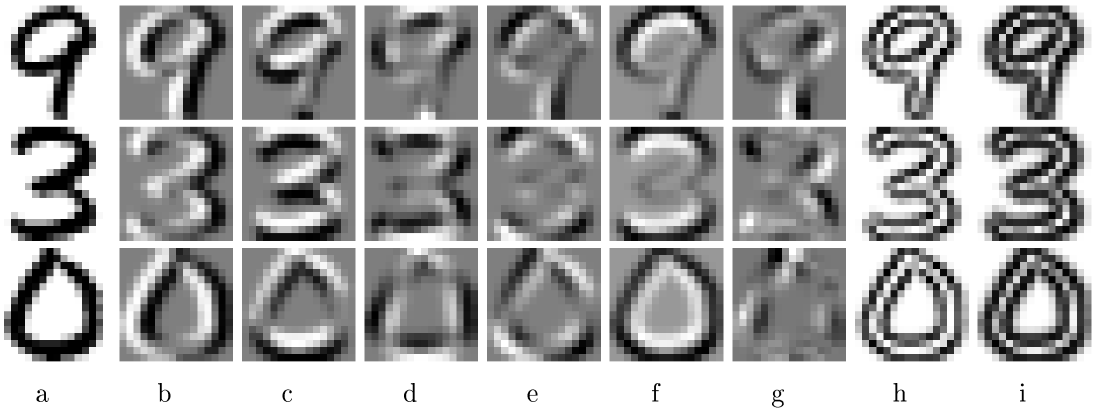

Abbildung 5.15: Visualisierung der Tangentenvektoren zu USPS-Beispielbildern
a: Originalbild, b: Verschiebung x-Achse, c: Verschiebung y-Achse, d: Rotation, e: Skalierung, f: Achsendeformation g: diagonale Achsendeformation, h: Strichstärke, i: Wurzel aus der Strichdicke

$$\begin{pmatrix} \dot{t} \\ \dot{s} \end{pmatrix} = \begin{pmatrix} 1+a & b \\ c & 1+d \end{pmatrix} \begin{pmatrix} t \\ s \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (5.23)$$

Horizontale Translation:

$a=b=c=d=f=0 \qquad \dot{t}=t+e \qquad \dot{s}=s$

$$x_1(t,s) := \lim_{e\to 0} \frac{x(t+e,s)-(t,s)}{e} \quad (5.24)$$

Vertikale Translation:

$a=b=c=d=e=0 \qquad \dot{t}=t \qquad \dot{s}=s+f$

$$x_2(t,s) := \lim_{f\to 0} \frac{x(t,s+f)-(t,s)}{f} \quad (5.25)$$

Rotation:

$a=d=e=f=0 \qquad b=-c \qquad \dot{t}=t+bs \qquad \dot{s}=s-bt$

$$x_3(t,s) := sx_1(t,s)-tx_2(t,s) \quad (5.26)$$

Skalierung:

$c=b=e=f=0 \qquad a=d \qquad \dot{t}=t+at \qquad \dot{s}=s+as$

$$x_4(t,s) := tx_1(t,s)+sx_2(t,s) \quad (5.27)$$

Achsendeformation:

$a=d=e=f=0 \qquad b=c \qquad \dot{t}=t+cs \qquad \dot{s}=s+ct$

$$x_5(t,s) := s\,x_1(t,s)+t\,x_2(t,s) \quad (5.28)$$

Diagonale Achsendeformation:

$b=c=e=f=0 \qquad a=-d \qquad \dot{t}=t+dt \qquad \dot{s}=s-ds$

$$x_6(t,s) := t\,x_1(t,s)-s\,x_2(t,s) \quad (5.29)$$

Strichstärken-Variation:

$$x_7(t,s) := (x_1(t,s))^2+(x_2(t,s))^2 \quad (5.30)$$

Wurzel der Strichstärke:

$$x_8(t,s) := \sqrt{(x_1(t,s))^2+(x_2(t,s))^2} \quad (5.31)$$

Insbesondere die Tangentenvektoren für die Strichdicke lassen die Arbeitsweise des Verfahrens erkennen. Addiert man etwa den zur '9' gehörigen Tangentenvektor für die Strickdicke auf das Originalbild, so wird die Zahl 'dicker'. Subtrahiert man ihn jedoch, so wird eine geringere Strichdicke erreicht. Ähnlich verhält es sich mit den Tangentenvektoren zu den anderen Transformationen. An dieser Stelle sei nochmal auf Abbildung 5.14 verwiesen, die einige so entstehende Modifikationen einer Ziffer zeigt.

Die so berechneten Vektoren wurden in unseren Experimenten als zusätzliche Merkmale betrachtet, das heißt, der Merkmalsvektor besteht aus einer Konkatenation aus Originalbild und einem Tangentenvektor. Dieser Vektor wird dann der Subspace-Methode unterworfen, in der Hoffnung, daß die so realisierte LDA die entsprechende Transformation 'lernt'. Auf Grund der hohen Dimensionalität der Originalbilder bzw. der Tangentenvektoren wurde in unseren Experimenten nur jeweils ein Tangentenvektor als Zusatz-Feature verwendet (auch die Subspace Methode benötigt eine Kovarianzmatrix im Original-Merkmalsraum, deren Berechnung wird mit wachsender Dimension jedoch immer unpräziser, da mit $x \in \mathbb{R}^D$ D^2 Parameter bestimmt werden müssen).

5.4 Erzeugung virtueller Trainings- und Testdaten

Eine weitere Methode, Invarianzen zu modellieren besteht darin, die vorhandenen Daten künstlich zu vervielfachen. Die Trainingsdaten werden auf jeder Seite um eine Pixelbreite vergrößert und die neuen Pixel (in diesem Fall) mit der Hintergrundfarbe (hier Grauwert 0) aufgefüllt. Nun können die Bilder beispielsweise in alle 8 Richtungen (links, rechts, oben, unten und die 4 Diagonalrichtungen) verschoben werden, so daß zusammen mit dem Originalbild eine Verneunfachung der Trainingsdaten erreicht wird.

Die Translationsinvarianz ist dann explizit in den Trainingsdaten enthalten (da zu jedem Bild auch translatierte Versionen in den Trainingsdaten enthalten sind). Ein zweiter Vorteil ergibt sich aus diesem Vorgehen, da durch die Vermehrung der Trainingsdaten die Parameter der Mischverteilungen (insbesondere die Varianz) stabiler geschätzt werden können (vgl. Kapitel 4), da die Diskrepanz zwischen der hohen Dimensionalität des Raums und der oftmals zu geringen Anzahl an Trainingsdaten verringert wird.

Natürlich läßt sich dieser Ansatz auf alle Transformationen übertragen, die die Klassenzugehörigkeit eines Objektes nicht verändern. Im Falle der USPS Daten erwies sich jedoch die Translation der Daten als besonders effektiv [Güld 2000].

Ein ähnlicher Ansatz läßt sich mit den Testdaten durchführen. Die Testdaten können auf gleiche Weise durch Verschiebung der Bilder vervielfacht werden. Die einzelnen Bilder werden dann separat klassifiziert und die Ergebnissse der einzelnen Klassifikationen geeignet zu einer Entscheidung für das originale Testdatum kombiniert (für diesen Schritt stehen zahlreiche Verfahren zur Verfügung, siehe etwa [Kittler 1998].

Genauere Informationen zu diesem Thema finden sich etwa in [Güld 2000] oder [Dahmen et al. 2000a].

Kapitel 6

Ergebnisse

Begonnen wurde die Diplomarbeit mit der Klassifikation von roten Blutkörperchen. Ausgehend von sehr einfachen Ansätzen wie dem Verwenden des unveränderten Bildes als Merkmalsvektor (d.h. jeder Bildpixel wurde als Merkmal interpretiert, ohne jede Vorverarbeitung der Daten) wurden dann die in den vorherigen Kapiteln beschriebenen Verfahren und Algorithmen auf ihre Leistungsfähigkeit hin untersucht.

6.1 Erythrozytenklassifikation

Eine vorgegebene Aufteilung der Daten in Trainings- und Testdaten bestand nicht. Die Daten teilten sich vielmehr in 72 Messungen auf, die sich durch die Hinzugabe von verschiedenen Chemikalien unterscheiden, die eine Formänderung der Erythrozyten hervorrufen. In jedem Experiment sind circa 70 Aufnahmen aufgenommen worden, die in die 5 Klassen

(1) starker Stomatozyt,

(2) schwacher Stomatozyt,

(3) Diskozyt,

(4) schwacher Echinozyt und

(5) starker Echinozyt

eingeteilt sind (vergleiche Kapitel 2). Darüberhinaus existieren pro Messung circa 50 Bilder in 2 nicht zu klassifizierenden Klassen ((6) ‘unscharfe Aufnahmen’, (7) ‘Klassifikation nicht möglich’, d.h. der menschliche Experte konnte keine eindeutige Klassifikation vornehmen). Diese Klassen wurden in unseren Experimenten aber nicht weiter berücksichtigt. Nachdem diese aussortiert wurden, blieben noch 5162 Aufnahmen übrig. Allerdings standen nicht alle Daten von Beginn der Experimente an zur Verfügung, sie wurden in mehreren Blöcken hinzugenommen. Das hat den großen Nachteil, daß nicht alle im Laufe dieser Arbeit erzielten Ergebnisse vollständig vergleichbar sind, da es zu einem recht deutlichen Streuen der Ergebnisse innerhalb der (jeweils vorhandenen)

Daten gekommen ist. Erwartungsgemäß nahm die Fehlerrate mit wachsender Anzahl an Trainingsdaten ab. Im folgenden werden jedoch nur Ergebnisse präsentiert, die sich auf den vollständigen Datensatz von 5162 Blutzellenbildern beziehen.

Die insgesamt 72 Messungen unterscheiden sich einerseits durch die unterschiedlichen Medikamente, die der Lösung zugesetzt wurden, um eine Verformung der Erythrozyten zu erreichen, zum anderen unterscheiden sich die Messungen durch unterschiedliche technische Einstellungen der Meßgeräte (vergleiche Kapitel 2). Die unterschiedliche Medikamentenkonzentration hat zur Folge, daß die Klassen in den unterschiedlichen Messungen nicht gleichverteilt sind. Die verschiedenen technischen Einstellungen führen zu unterschiedlichen Bildern. So gibt es Messungen die in der Regel 'hellere' Bilder lieferten und solche mit dunkleren Aufnahmen. Um solche Schwankungen möglichst gut zu eliminieren wurden die gesamten Daten zusammengefaßt und dann in 10 zufällig erstellte Partitionen aufgeteilt. Ein Teil der Daten wurde als Testdaten verwendet, und die anderen 9 Teile zum Training. Da sich die Ergebnisse der verschiedenen Partitionen recht stark unterscheiden, wurde die Gesamtfehlerrate als Durchschnitt der 10 ermittelten Fehlerraten bestimmt.

Begonnen wurde mit einem sehr einfachen Test, in dem jedes Bildpixel als Merkmal interpretiert, also auf eine Vorverarbeitung der Daten verzichtet wurde. Die Bilder wurden mit der Subspace Methode in einen niederdimensionalen Vektorraum abgebildet (der Dimension 39, unter Verwendung von 8 Pseudoklassen für jede der 5 Originalklassen), in dem die Klassen einen möglichst großen Abstand voneinander haben, beziehungsweise möglichst gut trennbar sind. Auf diesen Merkmalen wurden dann - wie in Kapitel 4 beschrieben - die Parameter der Mischverteilungen geschätzt und die eigentliche Klassifikation der Daten vorgenommen.

Dieses einfache Vorgehen liefert aber - was nicht überrascht - völlig inakzeptable Fehlerraten, die für das 5-Klassen-Problem nicht unter 50% lagen. In einem nächsten Schritt wurden nun - wie in Kapitel 5 beschrieben - mittels der Fourier-Mellin-Transformation RST-invariante Merkmale extrahiert. Die erzielten Fehlerraten für das 5-Klassen-Problem in Abhängigkeit der gewählten Fenstergröße zeigt Tabelle 6.1. Dabei bedeutet eine Fenstergröße von $m \times n$ das $m \cdot n$ viele Features aus dem RST-invarianten Amplitudenspektrum (siehe Kapitel 5.1) extrahiert wurden. Als optimale Wahl erwies sich eine Fenstergröße von 24×12 Pixeln. Man beachte, daß diese Merkmale dann noch mittels der erwähnten Subspace Methode dimensionsreduziert wurden und zwar erneut unter Verwendung von 8 Pseudoklassen auf $K - 1 = 39$ Merkmale.

Tabelle 6.1: 5-Klassen-Fehlerraten für Mischverteilungen in Abhängigkeit der gewählten Fenstergröße, globales Varianzpooling

Fenstergröße	Fehlerrate [%]
48×24	43.2
32×16	41.7
24×12	36.5
16×8	43.7

Die optimale Fehlerrate von nunmehr 36.5% liegt zwar bereits deutlich unter der vergleichbaren Fehlerrate ohne RST-Merkmalsanalyse, ist jedoch immer noch sehr hoch. Eine eingehende Analyse der noch vorhandenen Fehler ergab, daß sich zahlreiche Fehlerkennungen durch Verwechseln der Klassen 'starker Stomatozyt' mit 'schwacher Stomatozyt' bzw. 'starker Echinozyt' mit 'schwacher Echinozyt' ergeben. Diese 'Fehler' sind medizinisch beziehungsweise für die Analyse von Blutbildern jedoch nicht relevant, da das Problem der Echinozytenklassifikation letztlich ein 3-Klassen-Problem darstellt. Verwechslungen innerhalb einer Klasse können also ignoriert werden. In einem weiteren Experiment wurden daher Verwechslungen innerhalb einer Klasse erlaubt, jedoch weiterhin mit 5 Klassen trainiert. Dieser Schritt ergab eine beträchtliche Senkung der Mischverteilungs-Fehlerrate von 36.5% auf 16.1% (erneut mit globalem Varianzpooling). Zum Beitrag der Dimensionsreduktion sei erwähnt, daß diese Fehlerrate ohne Subspace-Methode von 16.1% auf 18.8% ansteigt.

An dieser Stelle könnte man berechtigterweise einwenden, daß auch das Trainieren der Verteilungsparameter, also der Modelle in unserem Bayes-Klassifikator, ebenfalls mit nur 3 Klassen erfolgen sollte. Interessanterweise ergab sich für diesen Ansatz jedoch eine schlechtere Fehlerrate, nämlich 19.3%. Interpretiert man die feinere Unterscheidung in stark bzw. schwach ausgeprägte Blutzellen als 'Experten-Mischverteilungskomponenten', so gelangt man zu dem Schluß, daß die menschliche Clusteranalyse der des Computers - zumindest in diesem Fall - offensichtlich überlegen ist, und daher, wenn sie zusätzlich in das Training eingeht, die Klassifikation verbessert.

Ein weiteres Problem der Erythrozyten-Daten besteht darin, daß zahlreiche Daten ungünstig zugeordnet waren, da die Laboranten sich auch im Zweifelsfall häufig für eine Klasse entschieden (um dem statistischen Klassifikator möglichst viele Daten zur Verfügung zu stellen). Dies führte zu der Vermutung, daß zahlreiche schlecht gelabelte Daten die Klassifikationsleistung verminderten. Aus diesem Grunde wurden einige offensichtlich besonders schlecht gelabelte Meßreihen in Anwesenheit von Medizinern einer visuellen Prüfung unterzogen. Dabei wurden insgesamt 100 Aufnahmen aus der Datensammlung entfernt und circa 100 weitere Daten anderen Klassen zugeordnet. Dadurch konnte die Fehlerrate weiter auf 14.4% gesenkt werden. Die Abhängigkeit der Fehlerrate von der Anzahl der verwendeten Dichtenzahl pro Klasse für dieses Experiment zeigt Abbildung 6.1. Dabei

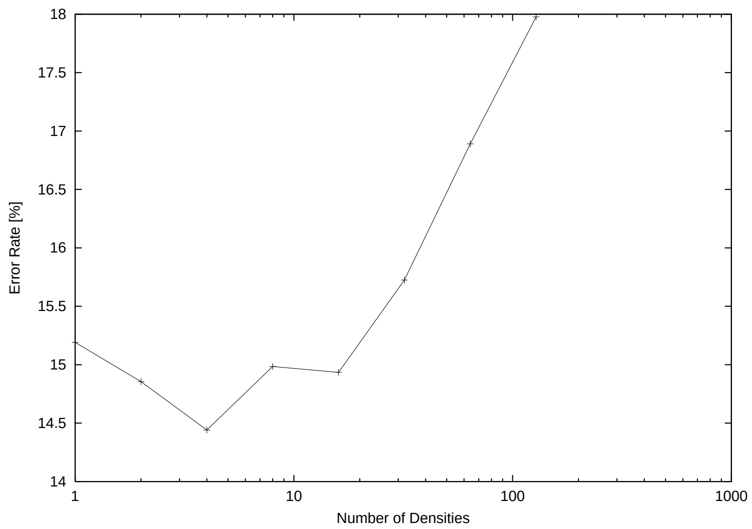

Abbildung 6.1: Abhängigkeit der Fehlerrate von der Anzahl der verwendeten Dichtenzahl pro Klasse, globales Varianzpooling

ist zu beachten, daß die Fehlerraten erneut über alle 10 Partitionen gemittelt wurden. Außerdem muß berücksichtigt werden, daß zum Training der Verteilungsparameter für jede der 5 Klassen nur etwa 800 stark streuende Trainingsdaten zur Verfügung standen. Darüberhinaus war die Anzahl der Trainingsdaten nicht gleichverteilt über die Klassen. So waren etwa 3000 Stomatozyten vorhanden, aber nur jeweils etwa 1000 Disko- und Echinozyten. Dies mag eine Erklärung des etwas untypischen Kurvenverlaufs sein (die optimale Fehlerrate wird bereits mit 4 Dichten pro Klasse erreicht).

In einem letzten Experiment wurde der Klassifikator zusätzlich auf Wunsch der Physiologie um eine Reject-Option erweitert. Als Reject-Funktion wurde ein einfacher Schwellwert verwendet, d.h. die Wahrscheinlichkeit für die beste erkannte Klasse mußte mindestens 20% besser sein als die der zweitbesten. Diese einfache Regel bewirkte eine weitere Senkung der Klassifikationsfehlerrate auf 13.6%, allerdings bei einer Rejectrate von 2.4%. Tabelle 6.2 zeigt exemplarisch die Fehler- und Rejectraten der einzelnen Partitionen (bei einer unsicheren Entscheidung zwischen stark bzw. schwach ausgeprägten Versionen einer Klasse findet natürlich kein Reject statt).

Tabelle 6.2: Fehlerraten der verschiedenen Testdaten

Partition	Fehlerrate [%]	Rejectrate [%]
0	13.2	2.4
1	16.2	2.2
2	14.4	2.5
3	10.7	2.1
4	12.0	3.2
5	14.3	3.2
6	16.8	2.6
7	13.0	2.6
8	14.7	1.8
9	10.8	1.8
Durchschnitt	13.6	2.4

Insgesamt bleibt festzuhalten, daß die beste erzielte Fehlerrate bei 14.4% (ohne Reject) bzw. 13.6% (mit Reject) liegt. Dies mag für ein 3-Klassen-Problem hoch erscheinen, es ist allerdings zu bemerken, daß die menschliche Fehlerrate für diese Aufgabe bei über 20% liegt [Fischer 1999]. An dieser Zahl läßt sich wohl die Schwere der Aufgabe ablesen. In jedem Fall erwies sich der entwickelte Klassifikator als leistungsfähig genug, die manuelle Klassifikation von roten Blutzellen durch eine automatische zu ersetzen, zumindest für die uns vorliegenden Daten. Für eine Anwendung in der Praxis sollte jedoch das Trainingsmaterial erweitert werden, um eine Invarianz gegenüber variierenden Versuchsaufbauten zu erreichen, die in unseren Experimenten nicht berücksichtigt wurden (anderes Mikroskop, variierende Beleuchtungsverhältnisse etc.). Wie bereits erwähnt, gibt es für die Blutzellen-Daten (abgesehen von der menschlichen Fehlerrate) keine Vergleichsergebnisse. Aus diesem Grunde wenden wir uns nun dem USPS-Korpus zu.

6.2 USPS-Ergebnisse

Es soll zunächst auf einige ‘Baseline-Ergebnisse’ eingegangen werden, die im folgenden zum Vergleich der (mit invarianten Merkmalen) erzielten Fehlerraten herangezogen werden (vergleiche [Güld 2000]).

6.2.1 Ausgangs-Ergebnisse USPS

Ausgangspunkt für unsere Experimente auf dem USPS-Korpus stellt die Anwendung des statistischen ‘Standard-Ansatzes’ dar. Das heißt, die Daten werden unter Verwendung der in Kapitel 4.4.3 beschriebenen Subspace-Methode in einen niederdimensionalen Merkmalsraum transformiert. Als optimal erwies sich dabei die Verwendung von 4 Pseudoklassen, so daß die reduzierten Merkmalsvektoren 39-dimensional sind. Mit diesen Features wurde dann ein auf Gauß’schen Mischverteilungen basierender Klassifikator realisiert (vergleiche Kapitel 4.3). Die unter Verwendung von - gegebenenfalls vervielfachten Trainings- und Testdaten erzielten Ergebnisse, die im folgenden als ‘Baseline’-Ergebnisse dienen werden, zeigt Tabelle 6.3. Dabei bedeutet die Schreibweise $a - b$, daß die Trainingsdaten mit einem Faktor a, die Testdaten mit einem Faktor b vervielfacht wurden (vgl. Kapitel 5.4). Die dargestellten Fehlerraten entstanden in der parallel laufenden Diplomarbeit [Güld 2000].

Tabelle 6.4 zeigt einen Vergleich der Subspace Methode für die Datenvervielfachungen 1-1 und 9-1. Wie man sieht, gelingt die LDA Schätzung für nur 7291 Trainingsdaten nicht - die Subspace Methode ist der LDA deutlich überlegen. Erst bei Verneunfachung der Trainingsdaten kann die LDA robust geschätzt werden und erzielt zur Subspace Methode identische Ergebnisse (diese Ergebnisse stammen erneut aus [Güld 2000]).

Tabelle 6.3: Baseline Ergebnisse auf dem USPS Corpus, mit Dimensionsreduktion (Subspace) und globalem Varianzpooling

Methode:	Fehlerrate [%]			
	1-1	1-9	9-1	9-9
Single Densities	12.8	12.4	13.1	11.7
Mixture Densities	7.1	6.4	4.5	3.4

Tabelle 6.4: Baseline Ergebnisse: Vergleich von LDA und Subspace-Methode, globales Varianzpooling

Methode:	Fehlerrate [%]	
	1-1	9-1
Mixture Densities, Subspace	7.1	4.5
Mixture Densities, LDA	10.9	4.5

Die im folgenden erzielten Ergebnisse werden also mit den in Tabelle 6.3 gezeigten verglichen werden, um eine sinnvolle Beurteilung der Leistungsfähigkeit der verwendeten invarianten Merkmale geben zu können.

6.2.2 Translationsinvarianz durch Fourier-Transformation

Ein erster Ansatz zur Verbesserung des Baseline-Ergebnisses ist es, die Translationsinvarianz des Amplitudenspektrums der Fourier-Transformation auszunutzen und so Merkmale zu extrahieren, die unter Umständen eine bessere Klassifikation ermöglichen.

Zu jedem Bild wurde zu diesem Zweck das Amplitudenspektrum der Fourier-Transformierten berechnet und dieses tiefpaßgefiltert, indem im Zentrum des Spektrums eine bestimmte Anzahl an Pixeln herausgeschnitten wurde. Wie in Kapitel 5 gezeigt, genügt es, nur eine Hälfte des Spektrums zu verwenden, da das Ausgangsbild reellwertig und dessen Fourier-Transformierte somit punktsymmetrisch (vergleiche Abb. 5.4) ist.

Es wurden unterschiedliche Größen für das ausgeschnittene Fenster evaluiert, wobei zu beachten ist, daß die so entstehenden Merkmalsvektoren jeweils mittels der Subspace Methode auf 39 Merkmale dimensionsreduziert wurden. Die nur mit diesen Merkmalen erzielten (jeweils besten) Mischverteilungs-Ergebnisse zeigt Tabelle 6.5. Man beachte, daß eine Vervielfachung der Daten über Verschiebungen in diesem Experiment sinnlos ist, da die extrahierten Merkmale für die virtuell erzeugten Daten gleich denen des Originalbildes sind. Die dargestellten Fehlerraten sind also mit den 1-1 Baseline-Ergebnissen zu vergleichen.

Tabelle 6.5: Fenstergröße gegen Fehlerrate (globales Varianzpooling), nur FFT-Features

Fenstergröße	1-1 Fehlerrate [%]
10×5	9.1
12×6	9.0
14×7	8.1
16×8	7.7
18×9	8.0
20×10	8.2
22×11	8.1

Man erkennt, daß die auf der Fourier-Transformation basierenden Features alleine keine guten Erkennungsergebnisse liefern (das beste Ergebnis von 7.7% ist signifikant schlechter als das beste Baseline-Ergebnis mit 7.1%). Dies läßt sich unter Umständen dadurch erklären, daß zur Extraktion der verwendeten Features die Phaseninformation vernachlässigt wurde. Gerade die Phase enthält aber wichtige Bildinformationen (siehe [Jähne 1991]). Aus diesem Grunde wurde in einem weiteren Experiment das jeweilige Originalbild an die extrahierten, translationsinvarianten Merkmale konkateniert und der entstehende Vektor erst dann der Subspace Methode unterworfen. Dieses Vorgehen bewirkte eine deutliche Verbesserung der Klassifikationsfehlerrate auf den originalen USPS Daten (ohne Vervielfachung der Daten). Die so erzielten Ergebnisse in Abhängigkeit von der gewählten Fenstergröße zeigt Tabelle 6.6.

Tabelle 6.6: Fenstergröße gegen Fehlerrate, FFT-Features plus Originalbild

Fenstergröße	1-1 Fehlerrate [%]
10×5	6.0
12×6	5.6
14×7	5.3
16×8	5.4
18×9	5.0
20×10	4.5
22×11	5.1

Da die entstehenden Merkmalsvektoren nun insgesamt nicht mehr translationsinvariant sind, macht nun auch eine Vervielfachung der Trainings- und Testdaten Sinn. Allerdings konnten diese Verfahren in diesem speziellen Fall die beste erzielte Fehlerrate von 4.5% nicht mehr verbessern (vergleiche Tabelle 6.7).

Man beachte dabei jedoch, daß durch Verwendung der Fourier-Features die 1-1 Fehlerrate von 7.1% auf nunmehr 4.5% gesenkt werden konnte, was einer relativen Verbesserung von 36.6% entspricht. Es konnte nicht endgültig geklärt werden, warum die Datenvervielfachung in diesem Fall keine Verbesserung ergab - gegebenenfalls sollte dieser Sachverhalt in weiteren Untersuchungen geklärt werden.

Abbildung 6.2 zeigt die erzielten 1-1 Fehlerraten für unterschiedlich gewählte Merkmalskombinationen bei variierender Fenstergröße in Abhängigkeit von der Gesamtanzahl der verwendeten Dichtenzahl, erneut für globales Varianzpooling. Interessant ist insbesondere die Tatsache, daß die Merkmalskombination ‘FFT-basierte Features + Originalbild’ die Single-Density Fehlerrate von 12.8% (nur Originalbild) auf 7.3% senkt (das entspricht einer relativen Verbesserung um 43.0%). Im Optimum wird eine Verbesserung von 7.1% auf 4.5% erzielt, was, wie bereits erwähnt - immer noch einer relativen Verbesserung von 36.6% entspricht. Tabelle 6.7 zeigt eine Übersicht der besten erzielten Ergebnisse mit translationsinvarianten, FT-basierten Merkmalen im Vergleich zu den Baseline Ergebnissen. Die beste Mischverteilungs-Baseline-Fehlerrate von 3.4% konnte trotz der sehr guten 1-1 Ergebnisse leider nicht verbessert werden.

Tabelle 6.7: Ergebnisse auf dem USPS Corpus, FFT vs. Baseline, 39 Subspace Features, globales Varianzpooling

Methode:	Fehlerrate [%]			
	1-1	1-9	9-1	9-9
Baseline: SD	12.8	12.4	13.1	11.7
Baseline: MD	7.1	6.4	4.5	3.4
FFT, SD	7.1	7.2	7.2	7.3
FFT, MD	4.5	4.9	4.6	4.6

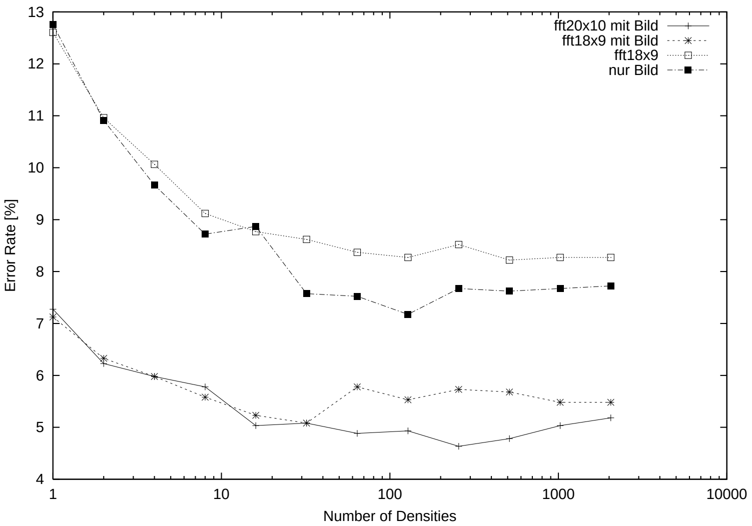

Abbildung 6.2: Fehlerrate in Abhängigkigkeit von der Gesamt-Dichtenzahl für FFT-Merkmale, einfache Trainings- und Testdaten, 39 Subspace-Merkmale, globales Varianzpooling

Als Fazit könnte folgender Schluß gezogen werden: Wenn die Möglichkeit der Erzeugung virtueller Daten besteht (was nicht immer so einfach ist wie im Falle vorsegmentierter Ziffern), scheint es sinnvoller zu sein, die Translationsinvarianz schon bei der Wahl der Trainingsdaten zu berücksichtigen, statt translationsinvariante Merkmale zu erzeugen. Ist das Erzeugen virtueller Daten jedoch nicht oder nur schwer möglich (etwa im Falle medizinischer Bilddaten), erweist sich das Extrahieren invarianter Features als sehr gute Wahl.

Abschließend sei bemerkt, daß es unter Umständen auch Sinn machen könnte, die Phaseninformation beizubehalten. Die Features (Ausschnitt aus dem Fourier-Bild incl. Phase) wären dann natürlich nicht mehr translations-invariant, allerdings erwies sich in unseren Experimenten auch das Hinzunehmen des Originalbildes als sinnvoll (was ebenfalls die Translationsinvarianz zerstört). Aus Zeitgründen konnte dieses Experiment nicht mehr durchgeführt werden, sollte jedoch in weiteren Arbeiten untersucht werden.

6.2.3 Fourier-Mellin basierte Merkmale für USPS-Daten

Als weiteres Verfahren wurde nun die RST-invariante Merkmalsextraktion mittels der Fourier-Mellin-Transformation getestet. Nach der Fourier-Transformation wird eine Polar-log-Transformation des Amplitudenspektrums durchgeführt und diese wieder Fourier-transformiert (siehe Kapitel 5). Wenn die RST-Merkmale (nach der Subspace Methode) alleine verwendet wurden, lag die beste erzielte Fehlerrate bei 12.6% Dies ist auch nicht verwunderlich, da hier insbesondere die Vertauschung der Ziffern *6* und *9* vorhersehbar ist, von den 254 Fehlern (entspricht 12.6%) gehen alleine 39 (1.9% der Daten bzw. 15.4% der Fehler) auf Vertauschungen von *6* und *9* zurück. Werden die RST-Werte als zusätzliche Komponenten zu den Bildwerten hinzugenommen, erhält man eine Fehlerrate von immerhin 5.8%. Da dieses Ergebnis bereits deutlich schlechter ist als das mit translationsinvarianten Merkmalen erzielte Ergebnis (4.5%), wurde dieses Verfahren nicht weiter verfolgt. Auch hier gelingt jedoch eine deutliche Verbesserung der 1-1 Fehlerrate, so daß die getesten Verfahren von Interesse sein können, wenn das Augenmerk auf die *schnelle* Klassifikation der Daten gelegt wird.

Es ist allerdings anzumerken, daß RST-invariante Merkmale auf anderem Datenmaterial durchaus Sinn machen können, etwa im IRMA Projekt [Dahmen et al. 2000d], einem Image Retrieval System für medizinische Bilddaten. Gerade im Fall der USPS Daten spielen aber Rotationen und Skalierungen offenbar keine große Rolle, lediglich die Translationsinvarianz scheint von Bedeutung zu sein.

6.2.4 Tangentenbilder als Merkmal für USPS-Daten

In diesen Experimenten wurden die Tangentenvektoren (wie in Kapitel 5.3 beschrieben) nicht zur Berechnung der eigentlichen Tangentendistanz herangezogen. Stattdessen wurde getestet, inwieweit die Tangentenvektoren für die Merkmalsextraktion geeignet sind. Es wurde also nicht wie von Simard vorgeschlagen der Abstand der Mannigfaltigkeiten berechnet, sondern die Tangentenvektoren als zusätzliche Merkmale verwendet. Ähnliche Verfahren - wie etwa das Berücksichtigen von richtungsabhängigen Gradientenbildern - erwiesen sich in der Bildererkennung oftmals als nützlich [Blanz et al. 1996].

Diese Tangentenvektoren (vergleiche Abbildung 5.15 wurden sowohl einzeln als auch in Kombination mit dem Originalbild getestet (d.h. im letzteren Fall werden die Tangentenvektoren und das Originalbild zu einem Vektor konkateniert und der so entstehende Vektor der Subspace Methode unterworfen).

Die Wurzel der Strichstärke wurde hinzugenommen, da die Strichdicke ein quadratischer Wert ist und somit deutlich größere Werte liefert, die - wenn sie mit den Bildwerten und den anderen Tangenten kombiniert wird - zu numerischen Instabilitäten führen kann (in der Tat werden mit der so 'normierten' Strichdickentangente bessere Ergebnisse erzielt).

Tabelle 6.8: Mischverteilungs-Fehlerraten mit Tangentenbildern für globales Varianzpooling und 39 Subspace-Merkmale

Tangentenvektor	Einfache Daten		Test- und Trainingsdaten 9-fach	
	nur Tangentenbild	Tangentenbild + Originalbild	nur Tangentenbild	Tangentenbild + Originalbild
Verschiebung x	7.0%	7.1%	3.9%	3.9%
Verschiebung y	7.1%	6.6%	4.4%	4.2%
Rotation	6.7%	6.8%	4.0%	4.3%
Skalierung	7.1%	6.7%	4.1%	3.8%
Achsendeformation	6.3%	7.3%	4.5%	4.2%
diag. Achsendeformation	8.8%	7.1%	4.8%	3.8%
Strichstärke	9.9%	8.0%	5.6%	4.1%
Wurzel der Strichstärke	8.8%	8.0%	5.1%	4.0%

Ergebnisse, die mit unterschiedlichen Merkmalskombinationen erzielt wurden, zeigt Tabelle 6.8. Es wird ersichtlich, daß die Tangentenvektoren ohne Datenvervielfachung eine geringe Verbesserung der Baseline-Fehlerrate von 7.1% auf 6.3% bewirken (unter Verwendung des Tangentenvektors für die Achsendeformation). Eine leichte Verbesserung erzielt auch der Tangentenvektor für Rotationen, und zwar auf 6.7%. Eine Verneunfachung von Training und Testdaten verbessert das Ergebnis signifikant auf 3.8%, allerdings liegt auch diese Fehlerrate unter der besten Baseline-Fehlerrate von 3.4%.

Insgesamt bleibt festzuhalten, daß sich die Tangentenvektoren offenbar besser zur tatsächlichen Berechnung der Tangentendistanz eignen. An dieser Stelle sei auf die parallel verlaufende Diplomarbeit [Keysers 2000] verwiesen.

6.2.5 7 Momente nach Hu

Die Momente nach Hu werden häufig als gute Wahl für invariante Merkmale angegeben, deshalb wurden sie im Rahmen dieser Arbeit überprüft. Sie wurden wie die anderen Merkmale sowohl alleine als auch in Kombination mit den Ursprungsbildern getestet. Werden die in Kapitel 5.2 beschriebenen 7 Momente isoliert als Merkmale betrachtet, so ergibt sich eine 'optimale' Fehlerrate von 46% auf USPS, also eine völlig indiskutable Fehlerate. Werden sie als Zusatz zu den Originalbildern hinzugenommen, so verschlechtern sie das Baseline-Ergebnis immer noch von 7.1% auf 7.8%, für einfache Daten und von 3.4% auf 4.0% für verneunfachte Daten. Weitere Tests wurden nicht durchgeführt, da keine signifikant besseren Ergebnisse zu erwarten sind. Offenbar eignen sich die Momente nur für die Klassifikation von Daten, deren Form nicht variabel ist, also etwa für die Erkennung maschinengedruckter Zeichen.

Dieses experimentelle Ergebnis deckt sich mit Aussage von [Süße 1999]. Um die letzte These weiter zu bestätigen, wurde folgender Test durchgeführt, um die Qualität der Moment-basierten Features zu messen. Zu allen 7291 Trainingsdaten wurden die Momente extrahiert und als 7-dimensionaler Merkmalsvektor gespeichert. Anschließend wurde die Klasseninformation der Daten vernachlässigt und eine Mischverteilung mit 128 Dichten trainiert (in der Maximum-Approximation, siehe [Dahmen et al. 2000a]). Die durchschnittliche 'Reinheit' einer Dichte wurde dann wie folgt ermittelt:

Zu jeder Dichte wurde die Klasse ermittelt, welche die meisten Beobachtungen beitrug. Diese Anzahl wurde dividiert durch die Gesamtanzahl der Beobachtungen, die der entsprechenden Dichte zugeordnet wurden. Je besser die extrahierten Merkmale sind, umso deutlicher sollte die durchschnittliche Reinheit aller Dichten gegen 1 bzw. 100% konvergieren. Für die 7 Momente ergab sich dabei ein Durchschnittswert von 49.9%, wohingegen die Interpretation eines jeden Pixels als Merkmale (also 256 Merkmale) bei gleicher Dichtenzahl eine durchschnittliche Reinheit von 86.2% ergab. Dies erklärt zumindest zum Teil die schlechte Performance der momentbasierten Erkennung.

6.2.6 Übersicht über die erzielten Resultate

Zusammenfassend läßt sich feststellen, daß die Kombination unterschiedlicher Verfahren zur Erzeugung von invarianten Merkmalen keine Verbesserung des besten Baseline-Ergebnisses (3.4%) liefern. Auf dem Amplitudenspektrum der Fourier-Transformation basierende, translationsinvariante Merkmale ermöglichten allerdings in Verbindung mit dem Originalbild eine deutliche Senkung der 1-1 Fehlerrate von 7.1% auf 4.5%. Insbesondere ist bei diesen Merkmalen das 1-1 Ergebnis bereits optimal und kann durch Datenvervielfachung nicht weiter verbessert werden. Die Verneunfachung der Daten wiederum ist insbesondere bei den Tangenten- und den Moment-basierten Merkmalen effektiv. Die beste Baseline-Fehlerrate konnte aber auch in diesem Fall nicht verbessert werden.

Es wurden außerdem Experimente mit Boosting [Freund & Schapire 1996] durchgeführt. Diese konnten in der Tat die 9-1 Baseline Fehlerrate von 4.5% auf 4.2% senken (unter Verwendung von 10 geboosteten Klassifikatoren). Da dieses Ergebnis jedoch signifikant schlechter war als das mit der Erzeugung virtueller Testdaten erzielten (3.4%), wurden diese Aktivitäten nicht weiter verfolgt.

Einen abschließenden Vergleich aller erzielten USPS-Fehlerraten zeigt Tabelle 6.9.

Tabelle 6.9: Übersicht der Mischverteilungs-Ergebnisse auf USPS

	1-1	1-9	9-1	9-9
Baseline	7.1	6.4	4.5	3.4
Bild + T-Invarianz	4.5	4.9	4.6	4.6
Bild + RST-Invarianz	5.8	-	5.7	-
Bild + Skalierung	6.7	-	4.5	3.8
Bild + diag. Achsendeformation	7.1	-	4.6	3.8
Boosting	-	-	4.2	-

Im Gegensatz zu den USPS-Daten hat die Verwendung der Fourier-Mellin-Transformation auf den Blutzellen-Daten allerdings eine signifikante Senkung der Fehlerrate zur Folge. Die Wahl der richtigen Merkmale für eine gegebene Erkennungsaufgabe scheint also erwartungsgemäß problemspezifisch zu sein.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Diplomarbeit wurden verschiedene Verfahren zur Erzeugung von affin-invarianten Merkmalen für die 2D Bilderkennung getestet. Diese Tests wurden im wesentlichen auf zwei recht unterschiedlichen Datensätzen durchgeführt. Zum einen wurde eine Erkennung unterschiedlicher Formen von roten Blutkörperchen gemacht, zum anderen wurde die Erkennung handgeschriebener Ziffern am Beispiel der US Postal Service Datensammlung implementiert.
Die unterschiedlichen Ansätze, die getestet wurden, sind im einzelnen:

- ein Ansatz für translationsinvariante Objekterkennung, der auf dem Amplitudenspektrum der Fourier-Transformation basiert,
- ein RST-invarianter Ansatz, der auf der Fourier-Mellin-Transformation beruht,
- ein weiterer RST-Ansatz, der auf den von Hu vorgeschlagenen 7 Momenten basiert, sowie
- ein Ansatz, der mit den von Simard vorgeschlagenen Tangenten-Vektoren arbeitet.

Auf den Erythrozytenbildern wurde mit der Fourier-Mellin-Transformation und anschließender Merkmalsreduktion mittels der vorgestellten Subspace-Methode, sowie der Klassifikation mit Mischverteilungen eine Fehlerrate von 13.6% (bei einer Zurückweisungsrate von 2.4%) bzw. 14.4% (ohne Zurückweisung) erzielt.
Dieses Ergebnis ist mit der menschlichen Fehlerrate von mehr als 20% vergleichbar [Fischer 1999]. Für diese Aufgabe erscheint das angestrebte Ziel, nämlich das Ersetzen der manuellen Klassifikation durch eine automatische, erreichbar. Da im Verhältnis zur Komplexität der Aufgabe jedoch nur relativ wenige Daten (5162 Bilder) zur Verfügung standen, sind die Untersuchungen mit diesen Daten unter Umständen noch nicht aussagekräftig genug. Vor dem Praxiseinsatz sollte auf jeden Fall eine Evaluation des Klassifikators auf neuen Aufnahmen durchgeführt werden. Die Aufgabe ist somit noch nicht abgeschlossen, die in der vorliegenden Arbeit erzielten Ergebnisse geben jedoch Grund zur Annahme, daß sie prinzipiell lösbar ist. Die Ergebnisse dieses Teils der Diplomarbeit wurden in [Dahmen et al. 2000b] publiziert.

Die auf den handgeschriebenen Ziffern der US Postal Service Datensammlung durchgeführten Tests verliefen insgesamt weniger erfolgreich. Hier ist zu unterscheiden zwischen den Experimenten mit der einfachen Translationsinvarianz mit einer Fourier-Transformation (bzw. deren Amplitudenspektrum) und der RST-Invarianz, die über zwei Fourier-Transformationen erreicht werden kann. Die Translationsinvarianz erbrachte für nicht vervielfachte Daten einen sehr deutlichen Gewinn, nämlich eine Reduktion der Fehlerrate von 7.1% auf 4.5% (was einer relativen Verbesserung um 36.6% entspricht). Wurde die Translationsinvarianz aber durch virtuelle Vervielfachung der Trainings- und/ oder der Testdaten berücksichtigt, so war eine Senkung der Fehlerrate (ohne Extraktion invarianter Features) auf 3.4% möglich (vorliegendes Baseline-Ergebnis). Im Gegensatz dazu konnte die Fehlerrate von 4.5% für die FFT-Features durch Datenvervielfachung nicht weiter gesenkt werden. Man beachte, daß das Originalbild zusätzlich zu den extrahierten, invarianten Features als Merkmal interpretiert wurde. Die Vervielfachung der Daten macht also prinzipiell auch in diesem Fall Sinn.

Weiterhin bleibt festzuhalten, daß die RST-Merkmale schlechter waren als die nur translationsinvarianten Merkmale. Dies mag durch Rundungsfehler bei der doppelten Fourier-Transformation zu erklären sein, oder aber dadurch, daß im Laufe der Fourier-Mellin Transformation zweimal die Phaseninformation vernachlässigt wird, so daß streng genommen nur ein kleiner Teil der Bildinformation verwendet wird (etwa 25%). Darüberhinaus tritt hier offenbar eine prinzipielle Schwierigkeit auf, da echt RST-invariante Merkmale im Regelfall keine Unterscheidung zwischen der Klasse '6' und '9' leisten können. Auf anderen Datensammlungen kann das Ergebnis daher anders ausfallen. So erwiesen sich die RST-invarianten Merkmale auf den Blutzellen-Daten als die beste untersuchte Methode.

Die Momente nach Hu, die oft als gute Wahl bezeichnet werden, wurden ebenfalls getestet. Sie lieferten allerdings signifikant schlechtere Ergebnisse als alle anderen getesteten Verfahren. Die Hinzunahme dieser Features zu den Originalbildpixeln verschlechterte das Baseline-Ergebnis (also das Ergebnis ohne Berücksichtigung von Invarianzen) erheblich. Es kann also mit großer Sicherheit davon ausgegangen werden, daß diese Merkmale zur Klassifikation von Daten, die eine nicht unerhebliche Streuung innerhalb der jeweiligen Klassen aufweisen, nicht geeignet sind. Sie eignen sich wohl nur zur Klassifikation von Daten mit geringer Streuung, etwa maschinengedruckte Buchstaben.

Die von Simard vorgeschlagenen Tangentenvektoren wurden in modifizierter Form angewendet. Es wurde nicht mit den entsprechenden Mannigfaltigkeiten bzw. der Distanz zwischen diesen gearbeitet. Stattdessen wurden die Tangentenvektoren (zusätzlich zum Originalbild) in den Merkmalsvektor aufgenommen. Der so entstehende Vektor wurde dann der Subspace Methode unterworfen und zur Klassifikation verwendet. Die hier verwendeten 7 Transformationen führen zu unterschiedlichen Ergebnissen. Die Richtungsvektoren der Rotation und die Achsendeformation konnten das Baseline Ergbnis von 7.1% leicht verbessern, ebenso die Verschiebung in y-Richtung und die Skalierung. Allerdings fiel der Gewinn mit einer Fehlerrate von 6.6% recht gering aus. Alle anderen Tangentenvektoren führten gar zu einer Verschlechterung des Ergebnisses. Die virtuelle Vervielfachung der Daten verbesserte das Ergebnis auf 3.8%, insgesamt aber konnte auch dieses Verfahren keine Verbesserung der besten Baseline-Fehlerrate von 3.4% erreichen.

Im Hinblick auf die begonnenen Arbeiten lassen sich weitere Anwendungen und Ideen verfolgen. Eine Idee wäre zum Beispiel, die Phaseninformation des Amplitudenspektrums nicht zu vernachlässigen, sondern weiter zu verwenden. Die so entstehenden Features sind natürlich nicht länger translationsinvariant, allerdings gilt dies auch für die Kombination des Bildes mit den invarianten Merkmalen. Nichtsdestotrotz lieferte diese Kombination in unseren Experimenten die besten Ergebnisse (besser als das (invariante) Amplitudenspektrum alleine). Auch könnten die auf den US Postal Service Daten gewonnenen Erkenntnisse (etwa bezogen auf die Tangentenvektoren) auch noch auf die Daten der Erythrozytenbilder angewendet werden. Darüberhinaus sollte in weiteren Arbeiten getestet werden, ob die Erzeugung virtueller Daten in Kombination mit invarianten Merkmalen stets erfolglos ist, oder ob dies eine Eigenschaft der USPS-Daten ist.

Abschliessen bleibt festzuhalten, daß insbesondere die translationsinvarianten Features - abhängig von der gerade betrachteten Anwendung - sicherlich von Interesse sein können, nämlich dann, wenn eine *schnelle* Erkennung gewünscht ist. Es sein daran erinnert, daß der Rechenaufwand für vervielfachte Trainings- und Testdaten deutlich über dem für nicht vervielfachte Daten liegt. Die verwendete Testdatenvervielfachung etwa führt zu einer ungefähr 9-fachen Laufzeit und senkt die Fehlerrate in Kombination mit ebenfalls vervielfachten Trainingsdaten (verglichen zu nicht vervielfachten, translationsinvarianten Features) von 4.5% auf 3.4%. Wann dieser Aufwand den dadurch erzielten Gewinn berechtigt erscheinen läßt, hängt stark von der zu lösenden Aufgabe ab und muß im Einzelfall entschieden werden.

Anhang A

Programme und Datenformate

In dieser Diplomarbeit wurde mit verschiedenen Programmen und Datenformaten gearbeitet. Um nachfolgende Arbeiten zu erleichtern, wird im folgenden Abschnitt auf diese eingegangen. Die beschriebenen Programme finden sich im Verzeichnis /u/perrey/PROGRAMME/.

A.1 Das verwendete ASCII-Format (*.jf-Format)

Das *.jf-Format ist ein einfaches ASCII-Format, das zum Austausch der Daten zwischen den einzelnen Programmen verwendet wird. Die *.jf Dateien haben folgenden Aufbau:

Zeile	Typ	Bedeutung
1	Integer	Zahl der Klassen n
	Integer	Länge der Merkmalsvektoren m
2 bis	Integer	Klasse des Merkmalsvektors
vorletzte		es dürfen nur Werte $0 \leq x \leq n-1$ auftretren
	m Float	Merkmalsvektor mit m Werten
letzte	Integer	-1 Ende Kenzeichen

Dieses nicht binäre Datenformat ist sicherlich nicht sonderlich effizient, bietet aber den Vorteil, auf unterschiedlichen Rechnern (Indy/Alpha etc.) gelesen und geschrieben werden zu können. Darüberhinaus ist es sehr leicht les- und schreibbar.

A.2 Script `tang_ablauf.csh`

Dies ist ein übergeordneter Job, der die gesamte Berechnung der Tangenten bzw. der Tangentenbasierten Merkmale durchführt (der Ablauf der Berechnung der Tangentenvektoren erfolgt dabei über ein Shellskript). Die Ausgabedateien enthalten dann unmittelbar die jeweils erzielten Fehlerraten.

Aufgerufene Programme: `tang_prog`, `concat_alpha`
Aufgerufene Jobs: `subspace_9.csh`

Eingabe :	ocr16-train.dat.mult.jf (9-fache Trainingsdaten)
	ocr16-test.dat.mult.jf (9-fache Testdaten)
Ausgabe :	out9_nur_tang1,out9_nur_tang2,out9_nur_tang3,
	out9_nur_tang4,out9_nur_tang2,out9_nur_tang6,
	out9_nur_tang7,out9_nur_tang8
	out9_tang1,out9_tang2,out9_tang3,
	out9_tang4,out9_tang2,out9_tang6,
	out9_tang7,out9_tang8

$1 \leq$ Nummer ≤ 8
Berechnung der Tangenten Testdaten (9fach) `tang_prog`
Aneinander hängen Testdaten (9fach)und Tangenten `concat_alpha`
Berechnung der Tangenten Traindaten (9fach) `tang_prog`
Aneinanderhängen Traindaten (9fach) und Tangenten `concat_alpha`
Merkmalstransformation mit `subspace_9.csh`
Löschen der Hilfsdaten

A.3 Script `subspace_9.csh`

Dieses Script realisiert die Subspace-Methode zur Merkmalsextraktion.

Aufgerufene Programme: `create_pseudo_classes.jf`, `whitening.jf`, `transform.jf`, `train_means_single_global.jf`, `svd.jf`, `reduz.jf`
Aufgerufene Jobs: `train_and_classify9_.csh`

Eingabe :	Name der Trainingsdatendatei
	Name der Testdatendatei
	Name der Ausgabedatei
Ausgabe :	Ergebnisse werden an die
	Ausgabedatei angehängt (falls diese nicht leer ist)

Erzeugung der Pseudoklassen. `create_pseudo_classes.jf`
Berechnung der Whitening-Matrix `whitening`
Transformation der Trainingsdaten `transform.jf`
Bestimmen der Prototypvektoren $(\mu_k - \mu)$ `train_means_single_global.jf`
Berechen der Orthonormalbasis der Prototypvektoren `svd.jf`
Projektion der Trainingsdaten in den 'Prototypen-Raum' `reduz.jf`
Transformation der Testdaten `transform.jf`
Projektion der Testdaten in den 'Prototypen-Raum' `reduz.jf`
Training und anschliessende Erkennung `train_and_classify9_.csh`

Am Ende erfolgt dann mittels `train_and_classify9_.csh` das Trainieren der Modelle und die anschliessende Klassifikation der Testdaten.

A.4 Script train_and_classify9_.csh

Aufgerufene Programme:	train_mix_em.jf, classify_mix_em_multiplied_testdata.2.jf
Aufgerufene Jobs:	keine

Eingabe : Name der Trainingsdaten
Name der Testdaten
Name der Ausgabedatei
Ausgabe : Ergebnisse werden an die
Ausgabedatei angehängt

Training einer Single Density train_mix_em.jf

Klassifikation classify_mix_em_multiplied_testdata.2.jf

setze Dichtenzahl = 2,4,8,16,32,64,128,256,512,1024,2048

- Auf letztem Training aufsetzen (Densities splitten)train_mix_em.jf
- Klassifikation classify_mix_em_multiplied_testdata.2.jf

A.5 Programm tang_prog

Dieses Programm berechnet die Tangentenvektoren zu einer Menge von Bildern.

Benutzung: tang_prog <infile> <nr> <outfile>

Eingabe- und Ausgabe-Datei müssen beide im *.jf-Format sein, die Nummer nr gibt an, welcher Tangentenvektor berechnet werden soll:

nr	Tangente
1	Horizontale Translation
2	Vertikale Translation
3	Rotation
4	Skalierung
5	Achsendeformation
6	Diagonale Achsendeformation
7	Strichstärke
8	Wurzel der Strichstärke

A.6 Programm `concat_alpha`

Das Programm hängt 2 Dateien im *.jf-Format aneinander, die Ausgabe Datei ist wieder im *.jf-Format. Die Merkmalsvektoren werden hintereinander gehängt, die neue Merkmalsvektorlänge Verktorlänge ergibt sich aus der Addition der Eingangsvektorlängen. Die Vektoren, die aneinander gehängt werden, müssen aus der gleichen Klasse sein.

Mit diesem Programm wird also etwa die Konkatenation von Originalbild und dazugehörigen Tangentvektoren oder RST-invarianten Features zu einem Merkmalsvektor erreicht. Benutzung:

```
concat_alpha <datei1> <datei2> <ausgabe-datei>
```

A.7 Programm `create_pseudo_classes.jf`

Das Programm `create_pseudo_classes.jf` erzeugt zu einer gegebenen Menge von Trainingsdaten die erwähnten Pseudoklassen. Die Ausgabedatei ist erneut im *.jf-Format, die Klassenzahl wächst aber auf 'Original-Klassenzahl * Anzahl Pseudoklassen pro Klasse'.

Benutzung:

```
create_pseudo_classes.jf   <eingabe>
                           Iterations per density
                           Max-Densities
                           Min-Observation/density)
                           Mode
```

Eingabe:	`<eingabe>`	Name der Eingabedatei mit den Trainingdaten Datei muß im *.jf-Format sein
	`Iterations per density`	Anzahl der Iterationen die zwischen 2 Splits ausgeführt werden
	`Max-Densitie`	Maximale Anzahl der Unterklassen in die jede Klasse geteilt wird muß $\in 2^n$ mit $n \in \mathbb{N}$ sind
	`Min-Observation/density`	Minimale Anzahl der Beobachtungen, die in einer Dichte sein müssen
	`Mode`	Pooling der Varianzen: 0 kein Pooling 1 pooling über inerhalb der Klassen 2 Pooling global
Ausgabe:	`<eingabe>.pseudo_classes`	Die Ausgabe erfolgt in ein File, dessen Name sich aus dem Namen der Eingabedatei und dem Anhang '`.pseudo_classes`' zusammensetzt Datei ist im *.jf-Format

A.8 Programm whitening.jf

Dieses Programm erwartet ein Eingabefile im *.jf-Format und berechnet dazu die Whitening-Transformationsmatrix.

Benutzung:

whitening.jf -i<infile> -m<1|2> -t<temp_file> -x<hoehe> -y<breite>

Eingabe:	-i	<infile>	Name der Eingabedatei im *.jf-Format
	-m	1,2	Art der Mittelwertbildung: 1 Klassenmittelwert (‘echtes’ Whitening) 2 Globaler Mittelwert default = 0 default : global
	-x	hoehe	Höhe der 2D-Merkmalsvektoren in <infile>
	-y	breite	Breite der 2D-Merkmalsvektoren in <infile>
Ausgabe:		<infile>.whitened	Ausgabe Erfolgt in einen File dessen Name sich aus dem Namen der Eingabedatei und dem Anhang ’.whitened’ zusammensetzt

Dateiformat der Ausgabe:

Zeile	Typ	Bedeutung
1	Integer	n Zahl der Eigenwerte sowie Zahl und Länge der Eigenvektoren
2	n float	n Eigenwerte
3 bis n+2	n float	n Komponenten je Eigenvektor

A.9 Programm transform.jf

Transformiert die Merkmalsvektoren unter Verwendung der Whitening-Transformationsmatrix (oder auch LDA-Matrix) Die Eigenwerte werden nach der Größe der zugehörigen Eigenwerte sortiert, nur die dominierenden n werden verwendet.

Aufruf: **transform.jf <n> <transmat> <infile> <outfile>**

Eingabe:	<n>	Maximalzahl der verwendeten Eigenvektoren Entspricht der maximalen Dimension der Ausgabevektoren
	<transmat>	Eigenwerte und Eigenvektoren im obigen Dateiformat
	<infile>	Name der Eingabedatei
	<outfile>	Name der Ausgabedatei
Ausgabe:	<outfile>	Die transformierten Merkmale werden in die Datei <outfile> geschrieben Ausgabe erfolgt im *.jf-Format

A.10 Programm train_means_single_global.jf

Berechnung der Prototypvektoren der Form $(\mu - \mu_k)$

Aufruf: `train_means_single_global.jf <Trainingsset>`

Eingabe:	`<Trainingsset>`	Name der Trainingsdaten im *.jf-Format
Ausgabe:	`<Trainingsset>`.Means	Prototypvektoren

Dateiformat der Ausgabe:

Zeile	Typ	Bedeutung
1 bis m	n float	Prototyp-Vektoren n= Länge der Vektoren für jede der m Klassen eine Zeile Beachte: Eingabedatei ist in der Regel in Pseudoklassen eingeteilt

A.11 Programm svd.jf

Berechnung der Orthogonalbasis des Prototypen-Raumes

Aufruf: `svd.jf <infile> [n]`

Eingabe:	`<infile>`	Name der Eingabedatei mit den Prototypvektoren Dateiformat siehe oben
	`n`	Wenn n angegeben, so werden nur n der Basisvektoren verwendet (die n größten Singulärwerte) Merkmalsvektoren also n-dimensional
Ausgabe	`<infile>.basis`	Die Ausgabe erfolgt in eine Datei, deren Name sich aus dem Namen der Eingabedatei und Anhang '`.basis`' zusammensetzt

Dateiformat der Ausgabe:

Zeile	Typ	Bedeutung
1	Integer	Zahl der Vektoren n
	Integer	Länge der Vektoren m
2 bis n+1	m float	Basisvektoren

A.12 Programm `reduz.jf`

Programm zum Projizieren der Merkmalsvektoren in den von den Prototypvektoren aufgespannten Raum (der orthonormalisiert sein muß!).

Aufruf: `reduz.jf <input> <basis> <output>`

Eingabe:	`<input>`	Name der Eingabedatei mit den zu projizierenden Vektoren Datei ist im *.jf-Format
	`<basis>`	Name der Datei mit der Orthogonalbasis Dateiformat siehe oben
	`<output>`	Name der Ausgabedatei in diese Datei werden die projizierten Vektoren geschrieben
Ausgabe	`<output>`	Die Ausgabe erfolgt in eine Datei `<output>` Ausgabe erfolgt in *.jf-Format

A.13 Programm `train_mix_em.jf`

Trainings-Programm, das den EM-Algorithmus realisiert.

`train_mix_em.jf`	`<Trainingsset>`	`<Iterations>`	`<Max-Densities>`
	`<Min-Observation>`	`<Pooling>`	`<mem-mode>`
	`<norm_flag>`	`<goodsplit>`	`[trainfile]`

Eingabe:	`<Trainingsset>`	Name der Trainingsdatei
	`<Iterations >`	Anzahl Iterationen zwischen den Splits
	`<Max-Densities>`	max. Anzahl Dichten/Klasse
	`<Min-Observation>`	(min. Anzahl von Trainingsvektoren in einer Dichte)
	`<Pooling>`	Pooling der Varianz: 0 kein Pooling 1 klassenspezifisches Pooling 2 globales Pooling
	`<mem-mode>`	0 Trainingsvektoren immer aus Datei lesen 1 Trainingsvektoren im Speicher halten
	`<norm_flag>`	0 Vektoren nicht normieren 1 Vektoren normieren
	`<goodsplit>`	0 altes Splitkriterium verwenden 1 neues Splitkriterium verwenden
	`[trainfile]`	Das Programm kann auf einem altem Training aufsetzen und die Dichten weiter teilen. Diese Verfahren spart Rechenzeit, wenn verschiedene Dichtenanzahlen getestet werden sollen
Ausgabe	`<infile> .trained`	Die Ausgabe erfolgt in eine Datei, deren Name sich aus dem Namen der Eingabedatei und dem Anhang '`.trained`' zusammensetzt

A.14 Programm classify_mix_em_multiplied_testdata.jf

Das Programm führt die Klassifikation der Testdaten durch, die Ergebnisse werden in Form von Tabellen ausgegeben.

```
classify_mix_em_multiplied_testdata.jf  <Testset>
                                        <Trainfile>
                                        <AP_SCALE>
                                        <norm_flag>
                                        <factor>
                                        <n>
```

Eingabe:	<Testset>	zu klassifizierende Testdaten im *.jf-Format
	<Trainfile>	trainierte Verteilungen
	<AP_SCALE>	Gewichtungsfaktor für a-priori-Wahrscheinlichkeit
	<norm_flag>	Merkmalsvektoren normieren?
	<factor>	Vervielfachungsfaktor der Testdaten
	<n>	betrachte von den <factor> vielen Testvektoren eines Blocks nur die mit den <n> besten scores
Ausgabe	Standard Out	Tabelle mit dem Klassifkationsergebnis

A.15 Programm jf2khoros4

Die translationsinvarianten Merkmale basieren auf der FFT. Um diese (und andere Bildbearbeitungsalgorithmen) nicht neu schreiben zu müßen, wurde auf die Khoros-Umgebung zurückgegriffen. Dieses Programm ermöglicht die Konvertierung von Dateien im *.jf-Format in das Khoros-Format *.viff (und ruft gegebenenfalls weitere Routinen per Systemcall auf, im Default das Shellskript job.csh). Da die FFT ein Bild erwartet, dessen Kantenlänge eine Zweierpotenz ist, verfügt das Programm darüberhinaus um eine Zero-Padding Funktion.

[t] Eingabe :	Name der Eingabedatei im *.jf-Format
	Name der Ausgabedatei
	In diese Datei wird vom Programm die Klasse des Vektors und vom aufgerufenen Skript der über die Khorosroutinen berechnete Vektor geschrieben
	Grösse der Bilder in x-Richtung
	Grösse der Bilder in y-Richtung
	Grösse des Bildes, das vom Programm ausgeben werden soll in x-Richtung
	Grösse des Bildes, das vom Programm ausgeben werden soll in y-Richtung
Ausgabe :	Die transformierten Merkmale werden in die Ausgabedatei geschrieben

Lese Eingabevektor

Daten in der Eingabedatei

- Ausgabe der Klasse in Ausgabedatei
- Ausgabe des Vektors im Khoros-Format in eine temporäre Datei
- Aufruf des Scripts zur Vektortransformation job.csh
- Lese Eingabevektor

Der Aufruf des Programmes ist z.B. für die verneunfachten Trainigsdater:

```
jf2khoros4 ocr16-test.dat.mult.jf ocr16-test.fft.milt.jf 18 18 32 32
```

Es wird ein Zero-Padding der Daten von 18×18 Pixel auf 32×32 Pixel durchgeführt und anschliessend die FFT-Merkmale zu diesen Bildern berechnet. Die Ausgabedatei enthält die transformierten Daten im 'quasi' *.j-Format (es fehlen die erste und letzte Zeile)

A.16 Script `job.csh`

Eingabe : Name der Ausgabedatei `<name>` Datei mit Namen `<name>.temp_file` Ausgabe : Ergebnisse werden an die Ausgabedatei `<name>` angehängt
Fourier-Transformation `kfft`
Berechnen des Amplitudenspektrums `kcmplx2real`
Ausscheiden des Zentrums `CutCenter`
Transformieren des Datenformates `kformats k2a`
Anhängen der transformierten Daten an Ausgabedatei

Format, es fehlen die erste Zeile (Zahl der Klassen und Vektorlänge) so wie die letzte Zeile mit -1 (End-of-File Markierung) die Datei kan mit dem Programm "`in_jf` "in das *.jf-Format gebracht werden.

Anmerkung:

Es gibt ebenfalls ein Shellskript `rst.csh`, welches die beschriebene Fourier-Mellin-Transformation durchführt.

Im Skript sind verschiedene Parameter wie z.B. die Grösse des ausgeschnittenen Fensters fest eingebaut, eine Variation dieser Parameter ist jedoch durch Änderung des Scriptes jederzeit möglich. Nach Ablauf der Programm-Skript-Kombination hat die Ausgabe-Datei noch nicht ganz *.jf-

Literaturverzeichnis

[Blanz et al. 1996] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, T. Vetter, *Comparison of View–Based Object Recognition Algorithms using Realistic 3D Models*, C. von der Malsburg, W. von Seelen, J. Vorbrüggen, B. Sendhoff (eds.): Artificial Neural Networks - ICANN'96, Springer Lecture Notes in Computer Science, Vol. 1112, Seiten 251-256, Berlin, 1996.

[Breiman et al. 1993] l.Breiman, J. Friedman, R. Olson, C. Stone: *Classification And Regression Trees* Chapman & Hall New York, 1993.

[Dahmen et al. 1998a] J. Dahmen, K. Beulen, H. Ney: *Objektklassifikation mit Mischverteilungen*, 20. DAGM Symposium Mustererkennung 1998, Seiten 167-174, Stuttgart, 1998.

[Dahmen et al. 1998b] 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, Seiten 442-446, 1998.

[Dahmen et al. 1999] J. Dahmen, R. Schlüter, H. Ney: *Discriminiative Training of Gaussian Mixtures for Image Object Recognition* 21. DAGM Symposium Mustererkennung 1999 Seiten 205-212 Bonn, 1999.

[Dahmen et al. 2000a] J. Dahmen, K. Beulen, M. O. Güld, H. Ney: *A Mixture Density Based Approach to Object Recognition for Image Retrieval*, 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, 2000.

[Dahmen et al. 2000b] 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.

[Dahmen et al. 2000c] J. Dahmen, D. Keysers, M. O. Güld, H. Ney: *Invariant Image Object Recognition using Mixture Densities*, eingereicht für ICPR 2000, Barcelona, Spanien, 2000.

[Dahmen et al. 2000d] 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 RIAO Conference on Content-Based Multimedia Information Access, Paris, 2000.

[Dempster et al. 1977] A. Dempster, N. Laird, D. Rubin, “Maximum Likelihood from *Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society, 39(B), Seiten 1-38, 1977.

[Deuticke et al. 1990] B. Denticke, R. Grebe, C. Haest: *Action of Drugs on the Erythrocyte Membrane*, in Blood Cell Biochemistry, Vol. 1; Erythroid Cells J.R. Harris, editor Plenum Press, New York, Seiten 475-529, 1990.

[Duda & Hart 1973] R. Duda, P. Hart: *Pattern Classification and Scene Analysis*, Wiley & Sons 1973.

[Hu 1962] M. K. Hu: *Visual pattern recognition by moment invariants.* IEEE Transactions on Information Theory 8, S. 179-187, 1962.

[Güld 2000] Marc Oliver Güld: *Inhaltsbasierter Bildzugriff mittels statistischer Objekterkennung*, Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen, in Arbeit (Abgabe Juli 2000).

[Fischer 1999] T.Fischer: Department of Physiology, RWTH Aachen University of Technology, persñliche Mitteilung 1999.

[Freund & Schapire 1996] Y. Freund, R. Schapire: *Experiments With a New Boosting Algorithm*, Procs. 13th Int. Conf. on Machine Learning, Bari, Italy, July 1996.

[Fukunaga 1990] K. Fukunaga: *Introduction to Statistical Pattern Recognition*, Computer Science and Scientific Computing Academic Press Inc., San Diego, CA, 2nd Edition, 1990.

[Jähne 1991] B. Jähne: *Digitale Bildverarbeitung*, Springer-Verlag, Berlin Heidelberg 1991.

[Keysers 2000] D. Keysers, *Approaches to invariant Image Object Recognition*, Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen, in Arbeit (Abgabe Juni 2000).

[Keysers et al. 2000] 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.

[Kittler 1998] J. Kittler, *On Combining Classifiers*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, Seiten 226-239, March 1998.

[Ney 1996] H. Ney: *Skript zur Vorlesung Suchverfahren der Spracherkennung*, RWTH Aachen, 1996.

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

[Ney 1999] H. Ney: *Skript zur Vorlesung Digitale Signalverarbeitung für Sprache und Bilder* RWTH Aachen, 1999.

[Mika et al. 1999] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K. Müller, *Fisher Discriminant Analysis with Kernels*, Proc. IEEE Neural Networks for Signal Processing Workshop, 1999.

[Press et al. 1992] W. Press, S. Teukolsky, W. Vetterling, B. Flannery: *Numerical Recipes in C*, University Press Cambridge, 1992.

[Pschyrembel 256] Pschyrembel: *Klinisches Wörterbuch*, 256. Auflage Walter de Gruyten 1990.

[Reddy & Chatterji 1996] B. Shrinivasa Reddy, B. Chatterji: *An FFT-based Technique for Translation, Rotation and Scale Invariant Image Registration*, IEEE Transaction on Image Processing, Vol. 5, August 1996.

[Schmidt & Thews 1997] Schmidt, Thews: *Physiologie des Menschen* 27. Auflage, Springer-Verlag, Berlin, Heidelberg, New York 1997.

[Schölkopf et al. 1997] B. Schölkopf, A. Smola, A. Müller, *Kernel Principal Component Analysis*, Proceedings ICANN'97, S. 583-588, Springer Lecture Notes in Computer Science, 1992.

[Schölkopf 1997] B. Schölkopf: *Support Vector Learning*, Oldenbourg Verlag München, 1997.

[Schölkopf et al. 1998] B. Schölkopf, P. Simard, A. Smola and V. Vapnik: *Prior knowledge in support vector kernels*, M. Jordan, M. Kearns and S. Solla, editors, Advances in Neural Information Processing Systems 10, MIT Press, Seiten 640-646, 1998.

[Schönfeld & Grebe 1989] M. Schönfeld, R. Grebe: *Automatic Shape Quantification of Freely Suspended Red Blood Cells by Isodensity Contour Tracing and Tangent Counting*, Computer Methods and Programs in Biomedicine Vol. 28, Elsevier, Seiten 217-224, 1989.

[Schukat-Talamazzini, 1995] E. Schukat-Talamazzini, *Automatische Spracherkennung*, Vieweg & Sohn, Braunschweig, 1995.

[Simard et al. 1993] P. Simard, Y. Le Cun and J. Denker: *Efficient Pattern Recognition using a New Transformation Distance*, in S. Hanson, J. Cowan and C. Giles, editors, Advances in Neural Information Processing Systems 5, Morgan Kaufmann, San Mateo CA, Seiten 50-58, 1993.

[Simard et al. 1998] P. Simard, Y. Le Cun and J. Denker, B. Victorri: *Transformation Invariance in Pattern Recognition – Tangent Distance and Tangent Propagation*, Lecture Notes in Computer Science, Vol 1524, Springer, Seiten 239-274, 1998.

[Süße 1999] H. Süße, Friedrich-Schiller-Universität Jena, persüliche Mitteilung im Rahmen der 21. DAGM, Bonn, 1999.

[Vapnik 1995] V. Vapnik: *The Nature of Statistical Learning Theory*, Springer-Verlag , New York, Seiten 142-143, 1995.

[Vapnik 1998] V. Vapnik: *Statistical Learning Theory*, Springer-Verlag New York 1998.

[Weber 1996] Dirk Weber: *Entwicklung eines Verfahrens zur Detektion, Verfolgung und Fokussierung frei sedimentierender Blutzellen im Mikroskop*, Diplomarbeit Biomedizinische Systemanalytik Institut für Physiologie, RWTH Aachen, Mai 1996.