

# Nichtlineare Verformungsmodelle für die Bilderkennung

Diplomarbeit im Fach Informatik

Lehrstuhl für Informatik VI  
Prof. Dr.-Ing. Hermann Ney  
Rheinisch-Westfälische Technische Hochschule Aachen

vorgelegt von:

*Christian Gollan*  
Matrikelnummer 209 972

Gutachter:

*Prof. Dr.-Ing. Hermann Ney*  
*Prof. Dr. Thomas Seidl*

Betreuer:

*Dipl.-Inform. Daniel Keysers*



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

Aachen, im September 2003

Christian Gollan



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>13</b>
<b>2</b>	<b>Klassifikationssystem</b>	<b>15</b>
<b>3</b>	<b>Verformungsmodelle für Distanzfunktionen</b>	<b>19</b>
3.1	Motivation für Verformungsmodelle in der Bilderkenung . . . . .	19
3.2	Distanzfunktion basierend auf Verformungsmodellen . . . . .	21
3.3	Image Distortion Modell . . . . .	22
3.4	2D Hidden Markov Modell . . . . .	23
3.4.1	Warped Wake 2DHM-Verfahren . . . . .	25
3.4.2	Simulated Annealing 2DHM-Verfahren . . . . .	28
3.5	Pseudo-2DHM-Modell . . . . .	29
3.6	Pseudo-2DHM Distortion Modell . . . . .	30
<b>4</b>	<b>Erweiterungen für Distanzfunktionen</b>	<b>31</b>
4.1	Bildvorverarbeitung . . . . .	31
4.1.1	Segmentierung und Padding . . . . .	31
4.1.2	Rotation und Spiegelung . . . . .	32
4.1.3	Skalierung . . . . .	32
4.1.4	Filteroperationen . . . . .	33
4.1.5	Gradientenbilder . . . . .	33
4.2	Erweiterung der Pixeldistanzfunktion . . . . .	34
4.3	Anpassung der Verformungsmodelle . . . . .	35
<b>5</b>	<b>Reduzierung der Rechenzeit</b>	<b>37</b>
5.1	Motivation . . . . .	37
5.2	Laufzeit . . . . .	37
5.3	Implementierung . . . . .	39
5.4	Rechenzeit . . . . .	39
5.5	Begrenzung der Referenzdaten . . . . .	40
5.5.1	Vorauswahl der Referenzbilder . . . . .	40
5.5.2	Reduzierung der Referenzdaten durch Verwendung von Prototypen . . . . .	41

<b>6</b>	<b>Datensammlungen und Experimente</b>	<b>43</b>
6.1	US Postal Service Datensammlung . . . . .	43
6.1.1	Stand der Technik . . . . .	44
6.1.2	Experimente und Ergebnisse . . . . .	44
6.2	UCI-Datensammlung . . . . .	53
6.2.1	Stand der Technik . . . . .	53
6.2.2	Experimente und Ergebnisse . . . . .	53
6.3	CEDAR-Datensammlung . . . . .	56
6.3.1	Stand der Technik . . . . .	56
6.3.2	Experimente und Ergebnisse . . . . .	56
6.4	MNIST-Datensammlung . . . . .	58
6.4.1	Stand der Technik . . . . .	58
6.4.2	Experimente und Ergebnisse . . . . .	58
6.5	Siemens-Datensammlung . . . . .	60
6.5.1	Stand der Technik . . . . .	60
6.5.2	Experimente und Ergebnisse . . . . .	60
6.6	ETL6A-Datensammlung . . . . .	62
6.6.1	Stand der Technik . . . . .	63
6.6.2	Experimente und Ergebnisse . . . . .	63
6.7	IRMA-Datensammlung . . . . .	64
6.7.1	Stand der Technik . . . . .	64
6.7.2	Experimente und Ergebnisse . . . . .	65
6.8	WANG-Datensammlung . . . . .	67
6.8.1	Stand der Technik . . . . .	67
6.8.2	Experimente und Ergebnisse . . . . .	68
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>69</b>
	<b>Literaturverzeichnis</b>	<b>71</b>
<b>A</b>	<b>Verwendete Software</b>	<b>75</b>
<b>B</b>	<b>Erstellte Software</b>	<b>77</b>
B.1	Datenformate . . . . .	77
B.2	Programme . . . . .	78

# Tabellenverzeichnis

5.1	Laufzeiten der Verformungsverfahren . . . . .	38
6.1	Ergebnisse für die USPS-Datensammlung . . . . .	44
6.2	USPS-Ergebnisse der Verformungsverfahren . . . . .	47
6.3	Beamsearch-Ergebnisse des WW2DHM-Verfahrens für die USPS-Datensammlung . . . . .	49
6.5	USPS-Ergebnisse unter Verwendung von Prototypen . . . . .	52
6.7	Ergebnisse für die UCI-Datensammlung . . . . .	55
6.8	Ergebnisse der MCEDAR-Datensammlung . . . . .	56
6.9	MCEDAR-Ergebnisse der Verformungsverfahren . . . . .	57
6.10	Ergebnisse für die MNIST-Datensammlung . . . . .	59
6.11	MNIST-Ergebnisse der Verformungsverfahren . . . . .	59
6.12	SiOI- und SiFF-Ergebnisse der Verformungsverfahren . . . . .	61
6.14	Ergebnisse für die ETL6A-Datensammlung . . . . .	63
6.15	Ergebnisse für die IRMA-Datensammlung . . . . .	65
6.16	IRMA-Ergebnisse der Verformungsverfahren . . . . .	65
6.17	Ergebnisse für die WANG-Datensammlung . . . . .	68
7.1	Niedrigste veröffentlichte Fehlerraten vorgestellter Datensammlungen . . . . .	70



# Abbildungsverzeichnis

1.1	Beispiel einer möglichen Bildbeschreibung . . . . .	14
2.1	Darstellung eines Klassifikationssystems . . . . .	16
3.1	Beispiele für die mangelhafte Invarianz der euklidischen Distanz . . . . .	20
3.2	Verschiebungsrichtungen der Shift $3 \times 3$ Methode . . . . .	20
3.3	Beispiele für Pixelabbildungsmöglichkeiten nach dem IDM . . . . .	23
3.4	Beispiel für eine Signalanpassung nach dem HMM . . . . .	23
3.5	Beispiel für ein nach dem 2DHMM angepasstes Signal . . . . .	24
3.6	Pixelabbildungsstrafe für ein 2DHMM . . . . .	25
3.7	Wakes und zugehörige Warped Wakes . . . . .	26
3.8	Darstellung der Abbildungsbedingungen . . . . .	27
3.9	Darstellung des WW2DHM-Verfahrens . . . . .	27
3.10	Darstellung des SA2DHM-Verfahrens . . . . .	28
3.11	Darstellung von Spaltenabbildungen nach dem P2DHMM . . . . .	29
3.12	Darstellung von P2DHM-Pixelabbildungen . . . . .	29
3.13	Darstellung möglicher P2DHMD-Pixelabbildungen . . . . .	30
4.1	Umgehung der ursprünglichen Randbedingung . . . . .	32
4.2	Rotation um $90^\circ$ gegenüber Abbildung 4.1 . . . . .	33
4.3	Beispielbild mit verschiedenen Filtern . . . . .	33
4.4	Beispiel für Gradientenbilder durch Splinekoeffizienten . . . . .	34
5.1	Verformungsbeispiele verschiedener Distanzverfahren . . . . .	40
6.1	Je ein Beispielbild für jede Klasse der USPS-Datensammlung . . . . .	43
6.2	Ergebnisse mit unterschiedlichen Strafkostenfunktionen für die USPS-Datensammlung und das P2DHM-Verfahren . . . . .	45
6.3	Beispiel einer unerwünschten Verformung durch das P2DHM-Verfahren . . . . .	45
6.4	Fehlerraten der USPS-Datensammlung durch das P2DHM-Verfahren abhängig von der Gewichtung der Gradienten- gegenüber den Grauwerten . . . . .	48
6.5	Beispiel-Verformungen durch WW2DHMM . . . . .	50
6.6	Durchschnittsbilder der USPS-Trainingsdaten . . . . .	51
6.7	Beispiel für nach dem IDM bestimmte USPS-Prototypen . . . . .	51

6.8	Beispiel für nach dem IDM bestimmte USPS-Prototypen von Gradientenbildern . . . . .	51
6.9	Bildresultate des EM-Cluster-Algorithmus durch Verwendung des IDM . . . . .	51
6.10	Je ein Beispielbild für jede Klasse der UCI-Datensammlung . . . . .	53
6.11	UCI-Ergebnisse mit unterschiedlichen Auflösungen für das P2DHM-Verfahren . . . . .	54
6.12	Beispiel für Ergebnisse von Interpolationsverfahren . . . . .	54
6.13	Je ein Beispielbild für jede Klasse der MCEDAR-Datensammlung . . . . .	56
6.14	Je ein Beispielbild für jede Klasse der MNIST-Datensammlung . . . . .	58
6.15	Je ein Beispielbild für jede Klasse der SiOI-Datensammlung . . . . .	60
6.16	Je ein Beispielbild für jede Klasse der SiFF-Datensammlung . . . . .	60
6.17	Je ein Beispielbild für jede Klasse der ETL6-Datensammlung . . . . .	62
6.18	Je ein Beispielbild für jede Klasse der ETL6A-Datensammlung . . . . .	62
6.19	Beispielbild für jede Klasse der IRMA-Datensammlung . . . . .	64
6.20	IRMA-Beispielbilder der Klasse „Thorax“ . . . . .	64
6.21	Fehlerraten der IRMA-Datensammlung durch nichtlineare Verformungsverfahren abhängig von dem Threshold-Parameter . . . . .	66
6.22	Je ein Beispielbild für jede Klasse der WANG-Datensammlung . . . . .	67

# Danksagung

Ich danke meiner Familie, besonders meinen Eltern, auf deren Unterstützung ich mich stets verlassen kann. Darüberhinaus gilt mein Dank Herrn Prof. Dr.-Ing. Hermann Ney und Prof. Dr. Thomas Seidl, den Gutachtern meiner Arbeit. Ich danke Daniel Keyzers für seine sehr gute Betreuung. Desweiteren danke ich Christina, Karin, Thomas, Tobias und Carsten für die zuverlässige Durchsicht dieser Arbeit.



# Kapitel 1

## Einleitung

Wird von Bilderkennung im Zusammenhang mit Computern gesprochen, wird ein System beschrieben, welches digitalisierte Bilder vorgegebenen Klassen zuordnet. Es gibt zahlreiche Aufgabengebiete für solche Klassifikationssysteme. Ein Aufgabengebiet ist beispielsweise die Schrifterkennung. Diese kann zur Sortierung von Briefen anhand der Adressen oder zur maschinellen Erfassung ausgefüllter Vordrucke, wie etwa Banküberweisungen, genutzt werden. Ein weiteres Gebiet ist die Gesichtserkennung, z.B. zur Zugangskontrolle. Ein zusätzliches Beispiel ist die automatische Erkennung des Kennzeichens von Kraftfahrzeugen, beispielsweise zur Kontrolle von Mautgebühren. Diese Klassifikationssysteme werden speziell für ihre Aufgaben konzipiert: indem ein solches System für die Klassifikation bestimmter Bilder optimiert und ausschließlich eine gegebene Menge von möglichen Klassen berücksichtigt wird.

Eine allgemeine Bildklassifikation könnte eine Bildbeschreibung liefern, z.B. nach dem MPEG-7 Standard (MPEG=Moving Picture Experts Group). Abbildung 1.1 stellt ein Bild mit einer möglichen Beschreibung dar. Ein Bilderkennungssystem, welches Bilder durch solche Beschreibungen zufrieden stellend klassifiziert, wird es aufgrund der Komplexität des Problems voraussichtlich in nächster Zukunft nicht geben. Diese Diplomarbeit beschreibt ein spezialisiertes Klassifikationssystem, das einem zu erkennenden Bild eine Klasse aus einer endlichen Menge zuweist, z.B. aus einer Menge von 10 Klassen, die jeweils für eine Ziffer stehen.

Bei Klassifikationssystemen sind zwei Ziele zu unterscheiden: zum einen die Minimierung der Fehlerrate und zum anderen die Minimierung des Rechenaufwands. Die Fehlerrate gibt die Anzahl der falsch klassifizierten Bilder, abhängig von den insgesamt klassifizierten Bildern, in Prozent an. Die Motivation zur Minimierung der Fehlerrate liegt in der Aufgabe, die Bilder sinnvoll zu klassifizieren. Die Reduzierung des Rechenaufwands zur Einhaltung einer maximalen Rechenzeit ist für Echtzeitanwendungen erforderlich. Diese Anwendungen müssen die zugehörige Klasse in einer vorgegebenen Zeitspanne bestimmen. Ein Anwendungsbeispiel ist die Klassifikation von Verkehrsschildern zur Unterstützung von Autofahrern. Ausschlaggebend für die Verwendung eines Klassifikationssystems ist oft die Fehlerrate, da ein schnelles aber fehleranfälliges System unerwünscht ist.

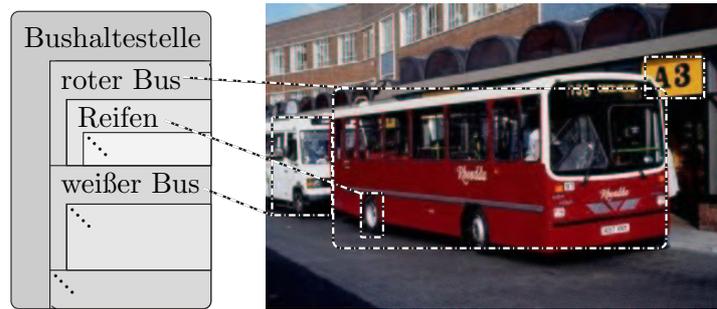


Abbildung 1.1: Beispiel einer möglichen Bildbeschreibung

Das Thema dieser Diplomarbeit ist die Verwendung von nichtlinearen Verformungsverfahren zur Bewertung eines Bildvergleichs. Die Bewertung eines Bildvergleichs basiert auf einer Distanzfunktion, die den Unterschied zwischen zwei gegebenen Bildern bewertet. Solche Distanzfunktionen können für die Bildererkennung in Klassifikationssystemen verwendet werden.

Im folgenden Kapitel wird ein Klassifikationssystem, das auf einer Distanzfunktion basiert, beschrieben. Die in Kapitel 3 vorgestellte Distanzfunktion wird durch ein nichtlineares Verformungsmodell gestaltet. In diesem Kapitel werden zusätzlich zur Distanzfunktion auch verschiedene Verformungsmodelle erläutert. Das 4. Kapitel beschreibt Methoden zur Vorverarbeitung der Bilddaten. Das 5. Kapitel nennt Möglichkeiten zur Reduzierung der Rechenzeit des hier vorgestellten Klassifikationssystems. Im 6. Kapitel werden verschiedene Bilddatensammlungen und Ergebnisse durch Experimente mit der in dieser Arbeit vorgestellten Distanzfunktion beschrieben. Das letzte Kapitel fasst die erreichten Ziele zusammen und beschreibt mögliche Erweiterungen des präsentierten Klassifikationssystems.

## Kapitel 2

# Klassifikationssystem

Ein Klassifikationssystem hat die Aufgabe, eine Beobachtung durch die Verwendung von Klassen zu beschreiben. Das im Folgenden erläuterte Klassifikationssystem reduziert die Beschreibung der Beobachtung auf eine Klasse  $k$  aus einer endlichen Menge von Klassen  $\{1, \dots, K\}$ . In der Bilderkennung ist der Ausgangspunkt in einem solchen System ein digitalisiertes Bild der Beobachtung. Die Beobachtung wird z.B. mit einer Videokamera als elektrisches Signal erfasst und durch eine Signalverarbeitung diskretisiert [Ney 03]. Ein digitales Einzelbild der Aufnahme wird zu einem Merkmalsvektor  $x \in \mathbb{R}^D$  weiterverarbeitet. Das eigentliche Klassifikationssystem weist dem Merkmalsvektor  $x$  eine Klasse  $k \in \{1, \dots, K\}$  zu, indem es das Ergebnis einer Entscheidungsfunktion  $r(x)$  liefert.

$$r : x \mapsto r(x) = k \quad (2.1)$$

Ein Merkmalsvektor kann das unveränderte digitale Bild einer Beobachtung sein. Ist dies der Fall, wird von einem erscheinungsbasierten Klassifikator gesprochen. Die Menge  $X = \{x \in \mathbb{N}^{I \times J} \mid 0 \leq x_{ij} \leq 255\}$  von Merkmalsvektoren beinhaltet alle möglichen Bilder mit einer Breite von  $I$  und einer Höhe von  $J$  Pixeln. Diese Menge wird Merkmalsraum genannt. Die Pixel eines Bildes aus dieser Menge  $X$  können 256 verschiedene Werte annehmen, die z.B. die Grauwerte repräsentieren. Jedem dieser Bilder der Menge  $X$  könnte eine Klasse  $k \in \{1, \dots, K\}$  zugewiesen werden, um die Klassifikation eines unbekanntes Bildes aus der Menge  $X$  als Suchprozess zu gestalten. Dies ist in der Praxis schwierig zu realisieren, da schon eine aus  $16 \times 16$  Pixel und 256 Graustufen bestehende Bildmenge  $256^{16 \cdot 16}$  mögliche Bilder enthält. Stattdessen wird eine Teilmenge  $\{\mu_1, \dots, \mu_N\} \subset X$  des Merkmalsraumes klassifiziert. Diese Teilmenge wird Trainingsbilder oder Referenzdaten genannt. Diese Daten können nach ihrer zugehörigen Klasse getrennt werden  $\{\mu_{11}, \dots, \mu_{1N_1}\} \cup \dots \cup \{\mu_{K1}, \dots, \mu_{KN_K}\} = \{\mu_1, \dots, \mu_N\}$ . Oft wird die Diskretheit der Grauwerte in der Modellierung vernachlässigt und der Merkmalsraum auf den Vektorraum  $\mathbb{R}^D$ ,  $D = I \cdot J$  erweitert.

Anhand der Referenzdaten wird die Entscheidungsfunktion  $r(x)$  gestaltet, welche sich durch eine Diskriminantenfunktion  $g(x, k)$  beschreiben lässt.

$$r(x) = \underset{k}{\operatorname{argmax}}\{g(x, k)\} \quad (2.2)$$

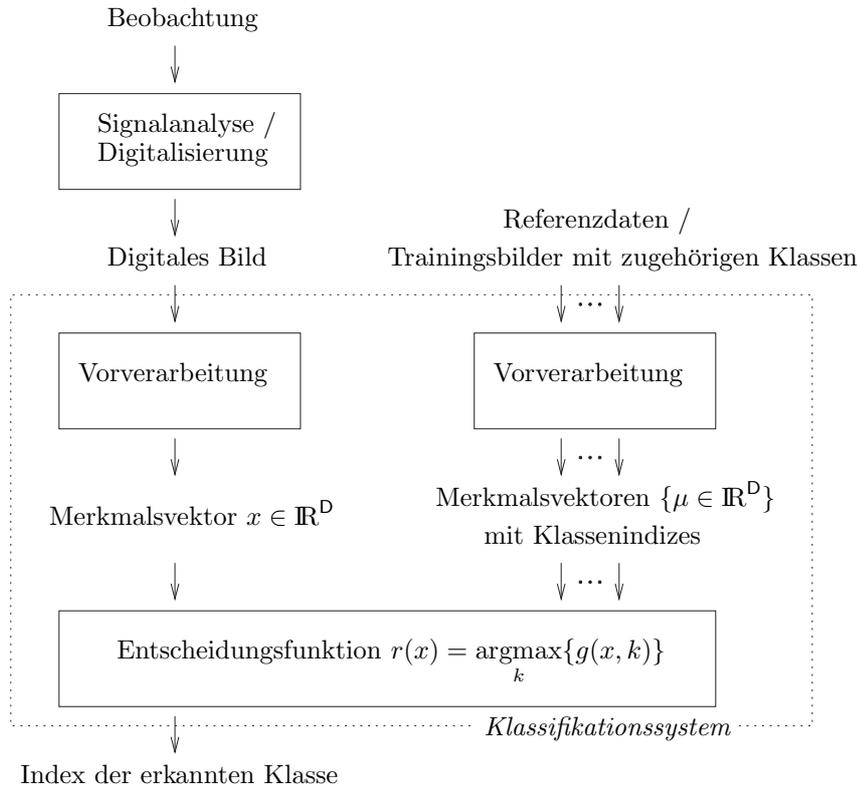


Abbildung 2.1: Darstellung eines Klassifikationssystems

Abbildung 2.1 stellt ein solches Klassifikationssystem graphisch dar. Dieses erhält ein digitales Bild einer Beobachtung als Eingabe und kann auf die klassifizierten Trainingsbilder zugreifen. Als Ausgabe liefert das System die erkannte Klasse, die das Ergebnis einer Entscheidungsfunktion ist. Die Diskriminantenfunktion  $g(x, k)$  kann auf einer Entscheidungsregel und einer Distanzfunktion basieren. Eine populäre Entscheidungsregel ist die so genannte Nächste-Nachbar-Regel (NN-Regel). Eine der bekanntesten Distanzfunktionen ist die quadrierte euklidische Distanz

$$\begin{aligned} D_{\text{euk}}^2(x, \mu) &= \sum_{d=1}^D (x_d - \mu_d)^2 \\ &= \|x - \mu\|_2^2. \end{aligned} \quad (2.3)$$

Die NN-Regel entscheidet sich für die Klasse  $k$ , die das Trainingsbild  $\mu \in \{\mu_1, \dots, \mu_N\}$  mit der geringsten Distanz  $D(x, \mu)$  zum zu klassifizierenden Bild  $x$  enthält.

$$g_{\text{NN}}(x, k) = \begin{cases} 1, & \text{für } k = \underset{k}{\operatorname{argmin}} \left\{ \min_{n=1, \dots, N_k} \{D(x, \mu_{kn})\} \right\} \\ 0, & \text{sonst} \end{cases} \quad (2.4)$$

Die Menge  $\{\mu_1, \dots, \mu_N\}$  kann Bilder enthalten, die falsch zu einer der vorgegebenen Klassen zugeordnet sind oder nicht eindeutig zu einer Klasse gehören. Dies wird in

der dargestellten Diskriminantenfunktion (2.4) nicht berücksichtigt. Vorteilhafter kann es sein, die Diskriminantenfunktion durch eine Wahrscheinlichkeitsverteilung über den Klassen zu gestalten:

$$g_{\text{Stat}}(x, k) = p(k|x). \quad (2.5)$$

Die a-posteriori-Wahrscheinlichkeit  $p(k|x)$  beschreibt die Wahrscheinlichkeit einer Klasse  $k$ , für ein gegebenes Testbild  $x$ . Die daraus resultierende Entscheidungsregel – die so genannte Bayes'sche Entscheidungsregel – ist optimal bezüglich der Fehlerrate, falls die Wahrscheinlichkeitsverteilung bekannt ist [Ney 02]. Nach Bayes gilt:

$$\begin{aligned} p(k|x) &= \frac{p(x, k)}{p(x)} \\ &= \frac{p(x|k)p(k)}{p(x)} \\ &= \frac{p(x|k)p(k)}{\sum_{k'=1}^K p(x|k')p(k')}. \end{aligned} \quad (2.6)$$

Die a-posteriori-Wahrscheinlichkeit  $p(k|x)$  wird durch die klassenbedingte Wahrscheinlichkeit  $p(x|k)$  und die a-priori-Wahrscheinlichkeit  $p(k)$  dargestellt. Durch die a-priori-Wahrscheinlichkeit  $p(k)$  wird die Wahrscheinlichkeit für das Auftreten der Klasse  $k$  beschrieben. Diese Wahrscheinlichkeit kann z.B. durch die relative Häufigkeit der Klassen definiert werden:  $p(k) = \frac{N_k}{N}$ . Eine weitere Möglichkeit ist die Annahme, dass jede Klasse gleich häufig auftritt:  $p(k) = \frac{1}{K}$ . Nach der Bayes'schen Entscheidungsregel ergibt sich die folgende Entscheidungsfunktion:

$$\begin{aligned} r_{\text{Stat}}(x) &= \operatorname{argmax}_k \{p(k|x)\} \\ &= \operatorname{argmax}_k \left\{ \frac{p(x, k)}{p(x)} \right\} \\ &= \operatorname{argmax}_k \{p(x, k)\} \\ &= \operatorname{argmax}_k \{p(k)p(x|k)\}. \end{aligned} \quad (2.7)$$

Die Qualität dieser Entscheidungsfunktion ist abhängig von den verwendeten Wahrscheinlichkeitsverteilungen. Die a-posteriori-Wahrscheinlichkeit  $p(x|k)$  kann z.B. durch Kernel Densities (KD) und einer Distanzfunktion modelliert werden:

$$\begin{aligned} p_{\text{KD}}(x|k) &= \frac{1}{N_k} \sum_{n=1}^{N_k} \mathcal{N}(x|\mu_{kn}, \Sigma_{\mu_{kn}}), & \Sigma_{\mu_{kn}} &= \sigma_k^2 I \\ &= \frac{1}{N_k} \sum_{n=1}^{N_k} \left[ \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp \left[ -\frac{1}{2} \left( \frac{D(x, \mu_{kn})}{\sigma_k} \right)^2 \right] \right]. \end{aligned} \quad (2.8)$$

Dies führt zur folgenden Entscheidungsfunktion:

$$\begin{aligned}
 r_{\text{KD}}(x) &= \operatorname{argmax}_k \{p(k)p_{\text{KD}}(x|k)\}, & p(k) &= \frac{N_k}{N} \\
 &= \operatorname{argmax}_k \left\{ \frac{1}{\sigma_k} \sum_{n=1}^{N_k} \exp \left[ -\frac{1}{2} \left( \frac{D(x, \mu_{kn})}{\sigma_k} \right)^2 \right] \right\}. & & (2.9)
 \end{aligned}$$

Es wird deutlich, dass die KD-Wahrscheinlichkeitsverteilung und die Distanzfunktion voneinander abhängig sind. Diese Abhängigkeit liefert die Motivation für eine invariante Distanzfunktion gegenüber klassenerhaltenden Bildtransformationen, da für die Klassifikation die ideale Verteilung ebenfalls invariant bezüglich solcher Transformationen ist.

Ein Praxisbeispiel für die Anwendung des beschriebenen Klassifikationssystems wäre eine Maschine, die Äpfel und Birnen auf einem Fließband sortiert. Dabei wird eine digitale Videoaufnahme des Fließbands erstellt. Die so gewonnenen Einzelbilder werden verwendet, um die Früchte zu klassifizieren. Die Einzelbilder werden so in der Bildvorverarbeitung segmentiert, dass von der zu erkennenden Frucht ein ausgefülltes Bild entsteht. Dieses zu klassifizierende Bild wird mit Referenzbildern, von denen bekannt ist, ob sie einen Apfel oder eine Birne darstellen, verglichen. Hierzu dient die Distanzfunktion, die die Ähnlichkeit dieser Bilder durch einen Distanzwert bewertet. Das System geht nach der NN-Regel davon aus, dass es sich bei den Bildern mit der geringsten Distanz zueinander um dieselbe Klasse handelt. Somit liefert das Klassifikationssystem eine Klasse als Ausgabe, die die Obstsorte beschreibt. Die Maschine kann nun einen von der erkannten Frucht abhängigen Arbeitsablauf starten.

Die auftretenden Klassifikationsfehler sind abhängig von der Signalverarbeitung der Beobachtungen, der Vorverarbeitung der Bilddaten und der Gestaltung der Diskriminantenfunktion. Damit man verschiedene Klassifikationssysteme (etwa unterschiedlicher Forschungseinrichtungen) vergleichen kann, gibt es Bildsammlungen mit zugehörigen Klassifikationsaufgaben. Anhand der Fehlerraten werden die Systeme bewertet. In Kapitel 6 werden verschiedene Datensammlungen und die Ergebnisse unterschiedlicher Forschungseinrichtungen vorgestellt. Einige Datensammlungen unterscheiden die Bilder in Referenz- und Testdaten. Die Testdaten sind die unter Verwendung der Referenzdaten zu klassifizierenden Bilder. Andere Datensammlungen haben keine solche Unterscheidung der Daten. Bei diesen Bildsammlungen sollen alle Bilder klassifiziert werden. Dazu dienen alle Bilder, ausgenommen das zu erklärende Bild, als Referenzbilder. Diese Methode wird „Leaving One Out“-Verfahren (engl. „to leave out“: auslassen) genannt.

In diesem Kapitel wurde die Motivation für eine invariante Distanzfunktion bezüglich klassenerhaltenden Transformationen erläutert. Das folgende Kapitel stellt Verformungsmodelle für Distanzfunktionen vor. Das Ziel ist die Gestaltung eines Verformungsmodells, das die klassenerhaltenden Transformationen beschreibt. Ein solches Verformungsmodell kann zur Modellierung einer Distanzfunktionen verwendet werden.

## Kapitel 3

# Verformungsmodelle für Distanzfunktionen

### 3.1 Motivation für Verformungsmodelle in der Bilderkennung

Die euklidische Distanzfunktion ist eine der bekanntesten Distanzfunktionen. Sie wird unter anderem in der Bilderkennung zur Bestimmung der Distanz zwischen zwei Bildern verwendet. Die Implementierung dieses Verfahrens ist unkompliziert und die Laufzeit linear abhängig von der Pixelanzahl. Unter Verwendung einer großen Menge von Referenzbildern und der NN-Regel erzielt man bei vielen Klassifikationsaufgaben bereits mit dieser einfachen Distanz gute Ergebnisse. Dies hat dazu beigetragen, dass sich viele Bildklassifikationssysteme an diesen Resultaten messen.

Ein großer Nachteil der euklidischen Distanz ist jedoch die fehlende Invarianz gegenüber Bildverformungen, die die Klasse des Bildes erhalten. In Abbildung 3.1 werden Bilder handgeschriebener Ziffern mit den Distanzwerten der euklidischen Distanz dargestellt. Diese Bilder stammen aus der USPS-Datensammlung, die im Kapitel 6.1 beschrieben wird. Jede Zeile von Bildern ist ein Beispiel für einen Klassifikationsfehler durch die NN-Regel aufgrund der euklidischen Distanz. Das jeweils erste Bild von links in einer Zeile ist ein Testbild. Die folgenden fünf Bilder sind die dem Testbild aufgrund der niedrigsten Distanzwerte zugeordneten Referenzbilder. Diese sind die fünf nächsten Nachbarn des Testbilds. Diese Klassifikationsfehler treten aufgrund der relativ hohen Distanzen zwischen Bildern derselben Klasse auf.

Um die Anzahl der Klassifikationsfehler zu verringern, sollte die Distanzfunktion so verändert werden, dass sie klassenerhaltende Transformationen für die Bestimmung der Distanz berücksichtigt. Eine Möglichkeit ist, bestimmte Verformungen der Referenzbilder zu erlauben, um dadurch die minimale Distanz zu bestimmen. Es werden beispielsweise affine Verformungen wie Translation, Rotation und Skalierung verwendet, um die euklidische Distanzfunktion zu erweitern [Perrey 00]. Eine Möglichkeit ist die Gestaltung einer Distanzfunktion, die die minimale Distanz von zusätzlich betrach-

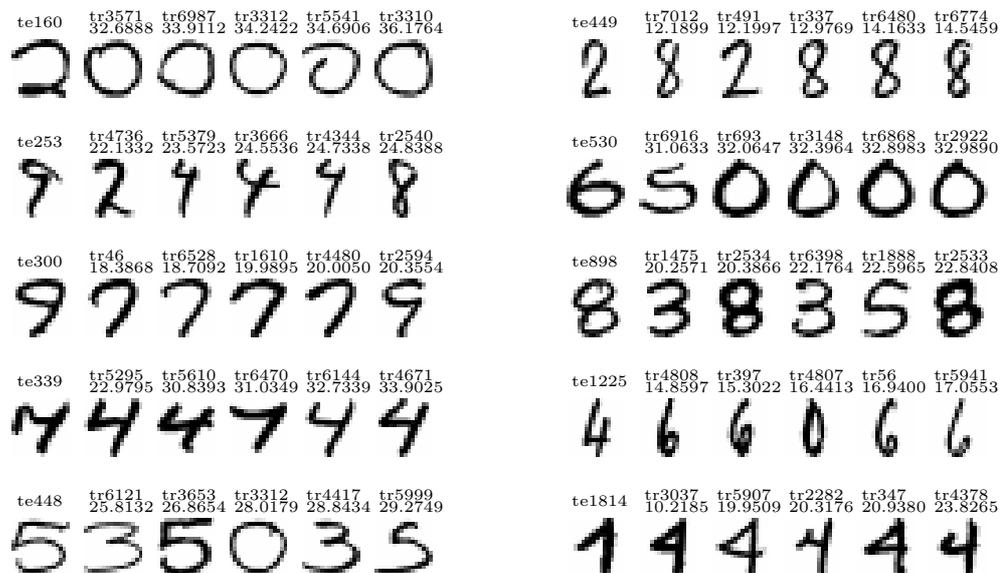


Abbildung 3.1: Beispiele für die mangelhafte Invarianz der euklidischen Distanz

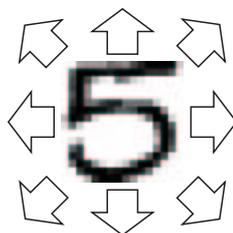


Abbildung 3.2: Verschiebungsrichtungen der Shift 3×3 Methode

teten Referenzbildern verwendet. Diese können aus den ursprünglichen Referenzbildern, durch eine Verschiebung um eine Pixelposition in alle 8 möglichen Richtungen, erzeugt werden. Abbildung 3.2 stellt die Shift 3×3 Methode dar (engl. „to shift“: verschieben). Durch diese Erweiterung der euklidischen Distanz wird die Fehlerrate der USPS-Datensammlung von 5.6% auf 4.8% gesenkt. Eine weitere Distanzfunktion, die sehr gute Resultate erzielt und auf linearen Verformungen beruht, ist die Tangentendistanz [Keysers & Dahmen<sup>+</sup> 00]. In der Spracherkennung wird erfolgreich ein nicht-lineares Verformungsmodell, das Hidden Markov Modell (HMM), zur Gestaltung der Distanzfunktion verwendet [Rabiner & Juang 93]. Dies gab den Ausschlag für Experimente mit einer Distanzfunktion, die auf nichtlinearen Verformungsmodellen beruht, zur Bewertung eines Bildvergleichs.

### 3.2 Distanzfunktion basierend auf Verformungsmodellen

Einer Distanzfunktion werden zwei zu vergleichende Bilder  $A$  und  $B$  übergeben:

$$\begin{aligned} A &= \{a_{ij}\}, \quad 1 \leq i \leq I, \quad 1 \leq j \leq J, \quad a_{ij} \in \mathbb{R}^D \\ B &= \{b_{xy}\}, \quad 1 \leq x \leq X, \quad 1 \leq y \leq Y, \quad b_{xy} \in \mathbb{R}^D \end{aligned} \quad (3.1)$$

Das Bild  $A$  hat die Breite von  $I$  und die Höhe von  $J$  Pixel. Das Bild  $B$  hat die Breite von  $X$  und die Höhe von  $Y$  Pixel. Die Pixelwerte  $a_{ij}$  und  $b_{xy}$  können z.B. die RGB-Farbwerte oder die Grauwerte eines Bildes repräsentieren. Jeder Pixelwert ist dabei im Allgemeinen durch einen Vektor von Pixelmerkmalen gekennzeichnet. Für ein Klassifikationssystem ist das Bild  $A$  ein unbekanntes, zu klassifizierendes Testbild und das Bild  $B$  eines der Referenzbilder. Das Bild  $A$  soll vollständig erklärt werden, indem jeder Pixelwert des Bildes  $A$  mit einem Pixelwert des Bildes  $B$  verglichen wird. Somit darf nur das Bild  $B$  verformt werden. Diese nichtlineare Bildverformung wird durch eine so genannte Warpingabbildung beschrieben (engl. „to warp“: verformen, verzerren). Dies geschieht, indem jeder Pixelkoordinate  $(i, j)$  des Bildes  $A$  eine Pixelkoordinate  $(x, y)$  des Bildes  $B$  zugewiesen wird. Die Warpingabbildung

$$(x_{11}^{IJ}, y_{11}^{IJ}), \quad (i, j) \rightarrow (x, y) = (x_{ij}, y_{ij}) \quad (3.2)$$

gibt an, welche Pixelwerte miteinander verglichen werden und beschreibt somit das verformte Bild

$$B_{(x_{11}^{IJ}, y_{11}^{IJ})} = \{b_{x_{ij}y_{ij}}\}. \quad (3.3)$$

Die Warpingabbildung wird durch die Wahl eines Verformungsmodells, das bestimmte Verformungen ausschließt, eingeschränkt.

Die Distanzfunktion wird durch eine Bildkostenfunktion und eine Strafkostenfunktion gestaltet. Die Bildkostenfunktion

$$C(A, B, (x_{11}^{IJ}, y_{11}^{IJ})) = \sum_{i=1}^I \sum_{j=1}^J d(a_{ij}, b_{x_{ij}y_{ij}}) \quad (3.4)$$

beschreibt die Kosten, die durch die aufsummierten Pixeldistanzen der Pixelwerte des Bildes  $A$  mit den durch die Warpingabbildung  $(x_{11}^{IJ}, y_{11}^{IJ})$  zugehörigen Pixeln des Bildes  $B$  entstehen. Für die Pixeldistanz kann z.B. die quadrierte euklidische Distanz

$$d_{\text{euk}}^2(a, b) = \sum_{d=1}^D (a_d - b_d)^2 \quad (3.5)$$

gewählt werden.

Durch die Berechnung von zusätzlichen Kosten durch eine Strafkostenfunktion  $R((x_{11}^{IJ}, y_{11}^{IJ}))$  ist es möglich, die erlaubten Bildverformungen weiter einzuschränken. Beispielsweise verringert die Strafkostenfunktion

$$R_{\text{len}}((x_{11}^{IJ}, y_{11}^{IJ})) = \sum_{i=1}^I \sum_{j=1}^J \sqrt{(x_{ij} - i)^2 + (y_{ij} - j)^2} \quad (3.6)$$

die Wahrscheinlichkeit für starke Bildverformungen, da die Strafkosten umso höher steigen, je weiter ein Pixel von seinem Ursprung verschoben wird. Bei  $R_{\text{len}}$  handelt es sich um eine absolute Strafkostenfunktion, weil die abgebildete Position des Pixel mit der ursprünglichen verglichen wird. Demgegenüber sind auch relative Strafkostenfunktionen sinnvoll, die den Abstand der abgebildeten Positionen zwischen benachbarten Pixeln betrachten, wie z.B. die später vorgestellte Strafkostenfunktion  $R_{\text{pen}}$  (3.11).

Die Gesamtkosten ergeben sich aus den Bildkosten und den gewichteten Strafkosten. Das Ziel ist die Minimierung der Kosten durch eine Bildverformung. Diese minimalen Kosten entsprechen dem Distanzwert, der zur Bewertung des Bildunterschiedes dient.

$$D(A, B) = \min_{(x_{11}^{IJ}, y_{11}^{IJ})} C(A, B, (x_{11}^{IJ}, y_{11}^{IJ})) + \alpha R((x_{11}^{IJ}, y_{11}^{IJ})) \quad (3.7)$$

Der Aufwand einer solchen Distanzbestimmung ist abhängig von der Wahl des Verformungsmodells, welches die Einschränkungen der Warpingabbildung festlegt. Erst durch Abbildungseinschränkungen wird die Distanzfunktion für Klassifikationssysteme interessant, da sonst jedem Pixelwert  $a_{ij}$  der Pixelwert  $b_{xy}$  mit der geringsten Pixeldistanz zugewiesen werden würde. Die Abbildungseinschränkungen für die Warpingabbildung dienen zur Erhaltung der für die Klasse typischen Eigenschaften des verformten Bildes. Die zulässigen Bildverformungen können zusätzlich abhängig von der Klasse oder dem Referenzbild modelliert werden.

Im Folgenden werden Verformungsmodelle, die Bedingungen für die Warpingabbildung einer solchen Distanzfunktion vorgeben, vorgestellt.

### 3.3 Image Distortion Modell

Das Image Distortion Modell (IDM) zur Gestaltung einer Distanzfunktion unter Berücksichtigung von Transformationen wurde am Lehrstuhl für Informatik VI der RWTH bereits früher untersucht [Keysers & Dahmen<sup>+</sup> 03]. In dieser Arbeit wurde festgestellt, dass die Ergebnisse, einer auf diesem Modell basierenden Distanzfunktion, durch eine Erweiterung der Pixeldistanzfunktion deutlich verbessert werden. Das Verfahren ist einfach zu implementieren, da das IDM auf einer einzigen Abbildungseinschränkung basiert, der Warprangebedingung

$$\begin{aligned} & |x_{ij} - i| \leq W \\ \wedge & |y_{ij} - j| \leq W. \end{aligned} \quad (3.8)$$

Diese Bedingung schränkt das Pixel, das zur Bestimmung der Pixeldistanz zum Testpixel zugeordnet wird, auf einen quadratischen Bildbereich ein und kann durch eine absolute Strafkostenfunktion, die für nicht zulässige Pixelabbildungen eine unendlich hohe Strafe angibt, beschrieben werden. Abbildung 3.3 illustriert Abbildungsmöglichkeiten eines Pixel nach diesen Einschränkungen. Diese Abbildungseinschränkungen lassen alle Bildverformungen zu, bei denen ein Pixelwert maximal um den Warprange  $W \in \mathbb{N}$  von seinen Ursprungs koordinaten verschoben wird. Die Warprangebedingung lässt sich

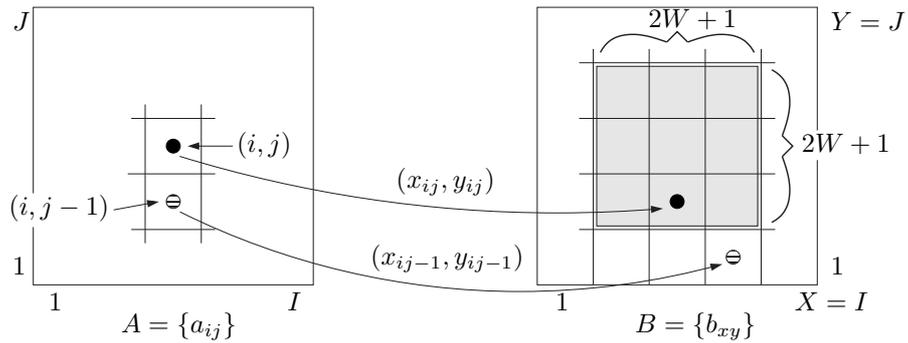


Abbildung 3.3: Beispiele für Pixelabbildungsmöglichkeiten nach dem IDM

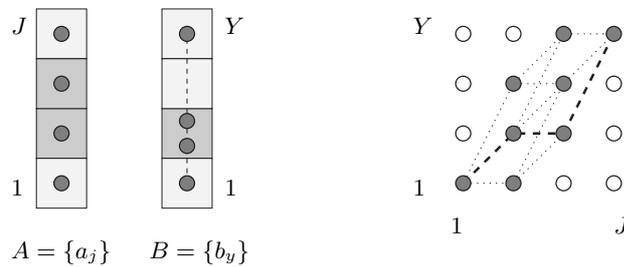


Abbildung 3.4: Beispiel für eine Signalanpassung nach dem HMM

ebenfalls durch eine absolute Strafkostenfunktion beschreiben, indem die Strafe unendlich hoch ausfällt, für absolute Pixelabbildungen um mehr als den Warprange  $W$ . Es gibt bei diesem Modell keine Bedingungen für die Abbildung eines Pixel, welche von den Abbildungen der Nachbarpixel abhängen. Deshalb wird dieses Modell auch „Zero Order“-Modell genannt.

### 3.4 2D Hidden Markov Modell

In der Mustererkennung wird zum Vergleich von zwei beobachteten Signalen die zeitliche Anpassung eines der Signale vorgenommen. Das Hidden Markov Modell (HMM) wird in der Spracherkennung zur Anpassung eines diskreten Signals verwendet [Ney 01].

Abbildung 3.4 zeigt zwei Signale und die möglichen Anpassungen nach dem HMM. Die Abbildungsbedingungen des HMM können in folgender Weise beschrieben werden:

Ein Signal darf durch die Werte des anderen Signals so erklärt werden, dass

- ein Wert beliebig oft wiederholt wird,
- maximal ein Wert zwischen zwei Werten ausgelassen wird
- und die zeitliche Reihenfolge der Werte eingehalten wird.

Als weitere Bedingung kann die Erhaltung des Startwertes und des Endwertes des angepassten Signals gefordert werden.

Das HMM zur Anpassung eines 1D-Signals kann auf das 2DHMM zur Anpassung eines 2D-Signals erweitert werden. Abbildung 3.5 stellt ein nach dem HMM angepasstes 1D-Signal sowie ein angepasstes 2D-Signal dar. Bei der hier vorgestellten Distanzfunktion kann das 2DHMM für die Modellierung der Warpingabbildung verwendet werden, wodurch die zulässigen Bildverformungen festgelegt werden. Das 2DHMM lässt sich durch die Monotonie- und Stetigkeitsbedingungen

$$\begin{aligned}
& 0 \leq (x_{ij} - x_{i-1j}) \leq 2 \\
& \wedge |x_{ij} - x_{ij-1}| \leq 1 \\
& \wedge 0 \leq (y_{ij} - y_{ij-1}) \leq 2 \\
& \wedge |y_{ij} - y_{i-1j}| \leq 1
\end{aligned} \tag{3.9}$$

für die Warpingabbildung beschreiben. Die Monotoniebedingungen verhindern Verformungen, bei denen Bildbereiche gespiegelt werden würden. Die Stetigkeitsbedingungen verhindern, dass größere Bildblöcke bei der Pixelzuordnung ausgelassen werden. Zusätzlich können folgende Randbedingungen festgelegt werden:

$$\begin{aligned}
& x_{1j} = 1 \\
& \wedge x_{Ij} = X \\
& \wedge y_{i1} = 1 \\
& \wedge x_{iJ} = Y
\end{aligned} \tag{3.10}$$

Diese lassen nur Bildverformungen zu, bei denen die Randpixel auf die zugehörigen Ränder abgebildet werden. Dieses Verformungsmodell orientiert sich an einer Arbeit von Uchida und Sakoe [Uchida & Sakoe 98].

Die Auswahl einer distanzminimierenden Verformung kann durch eine Strafkostenfunktion beeinflusst werden. Für Verformungen, die auf dem 2DHMM basieren, können die Abhängigkeiten der Nachbarpixel zur Gestaltung einer relativen Strafkostenfunktion verwendet werden. Die Strafkostenfunktion

$$\begin{aligned}
R_{\text{pen}^2}((x_{11}^{IJ}, y_{11}^{IJ})) &= \sum_{i=1}^I \sum_{j=1}^J (|x_{i-1j} + 1 - x_{ij}| + |y_{i-1j} - y_{ij}| \\
&\quad + |x_{ij-1} - x_{ij}| + |y_{ij-1} + 1 - y_{ij}|)
\end{aligned} \tag{3.11}$$

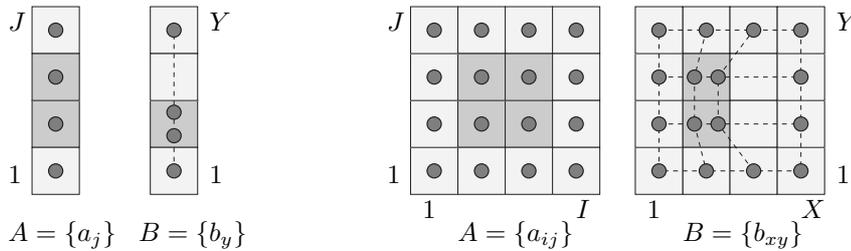


Abbildung 3.5: Beispiel für ein nach dem 2DHMM angepasstes Signal

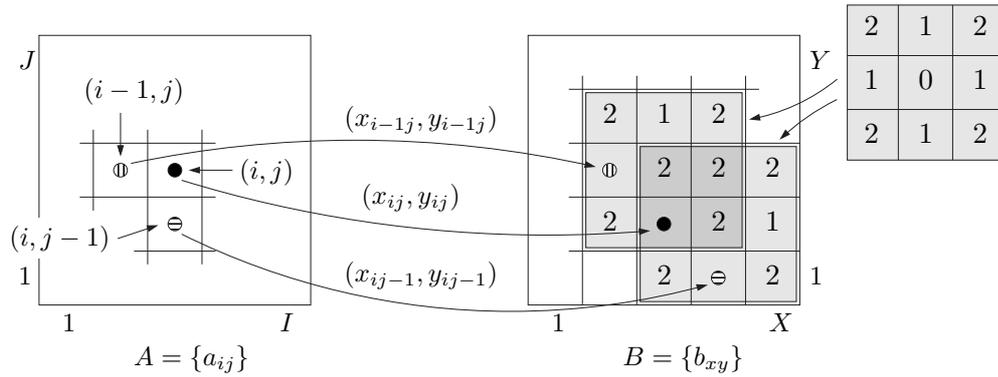


Abbildung 3.6: Pixelabbildungsstrafe für ein 2DHMM

betrachtet, im Gegensatz zu der vorgestellten Strafkostenfunktion  $R_{\text{len}}$ , nicht die Ursprungs koordinaten der Pixel, sondern ausschließlich die Warpingabbildung. Die Bestimmung der Strafkosten wird in Abbildung 3.6 veranschaulicht. Die Strafkosten ergeben sich durch Aufsummieren der Kosten der Pixelpositionen. Die Kosten einer Pixelposition sind abhängig von seiner Pixelabbildung und von der Pixelabbildung des linken und des unteren Nachbarn. Die von den Nachbarn abhängigen Kosten werden in der Abbildung durch die  $3 \times 3$  Pixel großen Quadrate dargestellt. Eine Pixelabbildung darf nach den Monotonie- und Stetigkeitsbedingungen nur auf die Schnittfläche der  $3 \times 3$  Pixel großen Quadrate erfolgen. Die Summe der Kosten, die durch die Quadrate vorgegeben ist, entspricht den Kosten der Pixelposition. Somit verursacht ein unverformtes Bild durch die identische Warpingabbildung keine Strafkosten.

### 3.4.1 Warped Wake 2DHM-Verfahren

Das Warped Wake 2DHM (WW2DHM)-Verfahren ist eine Implementierungsmöglichkeit für eine Distanzfunktion, die als Verformungsmodell das 2DHM-Modell verwendet. Dieses auf dynamischer Programmierung basierende Verfahren wurde von Uchida und Sakoe vorgestellt [Uchida & Sakoe 98]. Das WW2DHM-Verfahren findet wie ein HM-Verfahren die Verformung, die die Kosten minimiert. Die Vorgehensweise basiert auf den Monotonie- und Stetigkeitsbedingungen (3.10). Diese schränken eine Pixelabbildung aufgrund der Abbildungen der Nachbarpixel ein.

Das Verfahren arbeitet das Bild  $A$  in Pixelschritten durch, indem jedes Pixel einer Spalte von unten nach oben und die Spalten von links nach rechts bearbeitet werden. Es werden alle möglichen Pixelabbildungen, die abhängig von den Abbildungen des unteren und des linken Pixel sind (3.10), betrachtet. Damit die benötigten Informationen der Nachbarpixel effizient bearbeitet werden, werden so genannte Wakes und Warped Wakes verwendet. Ein  $\text{wake}(i, j) = ((i, j), (i, j - 1), (i, j - 2), \dots)$  beschreibt genau eine der Bildhöhe  $J$  entsprechende Anzahl von Pixelkoordinaten des Bildes  $A$ . Der  $\text{wake}(i, j)$  besteht aus der ersten Koordinate  $(i, j)$ , die restlichen  $J - 1$  Pixelpositionen des  $\text{wake}(i, j)$  sind die vorangegangenen Positionen der im Verfah-

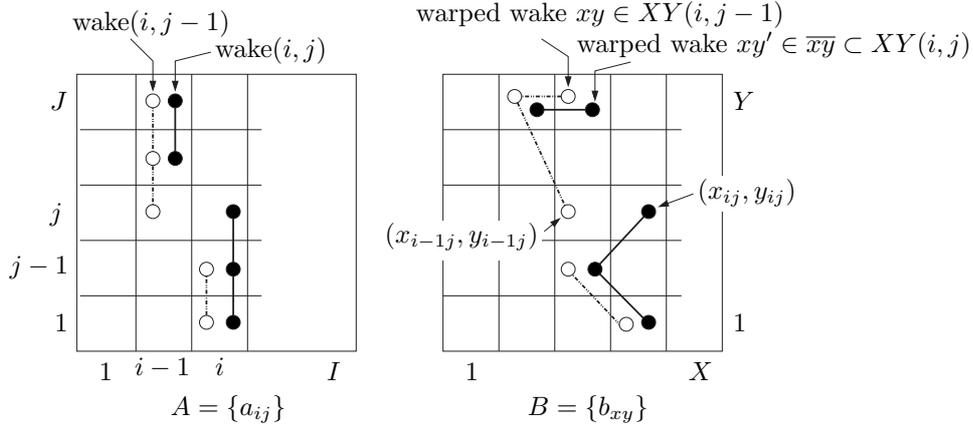


Abbildung 3.7: Wakes und zugehörige Warped Wakes

ren bearbeiteten Pixel. Somit ist ein  $\text{wake}(i, j)$  durch den Index  $(i, j)$ , der für die erste Pixelposition steht, eindeutig beschrieben. Der erste Wake des Verfahrens ist der  $\text{wake}(1, J)$ . Der zweite Wake,  $\text{wake}(2, 1) = ((2, 1), (1, J), (1, J - 1), \dots, (1, 2))$ , ist durch den nächsten Pixelschritt vorgegeben. Insgesamt gibt es  $(I - 1)J + 1$  Wakes, die durch das Verfahren bearbeitet werden. Die nach dem 2DHMM zulässigen Verformungen eines Wakes werden durch Warped Wakes beschrieben. Ein solcher Warped Wake  $xy(i, j) = ((x_{ij}, y_{ij}), (x_{ij-1}, y_{ij-1}), (x_{ij-2}, y_{ij-2}), \dots)$  ist eine mögliche Verformung des  $\text{wake}(i, j)$  und beschreibt die zu den Koordinaten des Wake zugehörigen Pixelpositionen des Referenzbildes. Die Menge  $XY(i, j)$  beinhaltet alle Warped Wakes  $xy(i, j)$  und beschreibt somit alle möglichen Verformungen des Wakes  $\text{wake}(i, j)$ .

$$XY(i', j') = \left\{ \{x_{ij}, y_{ij}\} : (i = i' \quad \wedge \quad j \in \{1, \dots, j'\}) \vee (i = i' - 1 \quad \wedge \quad j \in \{j' + 1, \dots, J\}) \right\} \quad (3.12)$$

Abbildung 3.7 stellt zwei aufeinander folgende Wakes und dazugehörige Warped Wakes dar. Zu jeder Menge  $XY(i, j)$  von Warped Wakes kann die nachfolgende Menge  $XY(i, j + 1)$  generiert werden, indem zu jedem Warped Wake  $xy \in XY(i, j)$  seine möglichen nachfolgenden Warped Wakes bestimmt werden. Die Menge  $\overline{xy} \subset XY(i, j + 1)$  beinhaltet die Nachfolger des Warped Wakes  $xy$ . Ein nachfolgender Warped Wake  $xy' \in \overline{xy}$  wird aus dem zugehörigen Vorgänger  $xy$  durch die Abbildungsbedingungen (3.10) eingeschränkt generiert. Dazu muss der Pixelposition  $(i, j + 1)$  eine mögliche Pixelposition des Referenzbildes zugewiesen werden. Die Abbildungsmöglichkeiten des Pixel an der Position  $(i, j + 1)$  wird durch das erste und das letzte Koordinatenpaar des Warped Wake  $xy$  eingeschränkt. Diese Koordinatenpaare beschreiben die Abbildungen des linken und unteren Nachbarn des Pixel an der Position  $(i, j)$ .

Durch die Abbildungsbedingungen (3.10) ist eine Menge  $\overline{xy}$  auf maximal 9 Warped Wakes begrenzt. Anders betrachtet hat jeder Warped Wake  $xy \in XY(i, j + 1)$  eine Menge  $\underline{xy} \subset XY(i, j)$  von maximal 9 Vorgängern, aus denen dieser entstanden sein

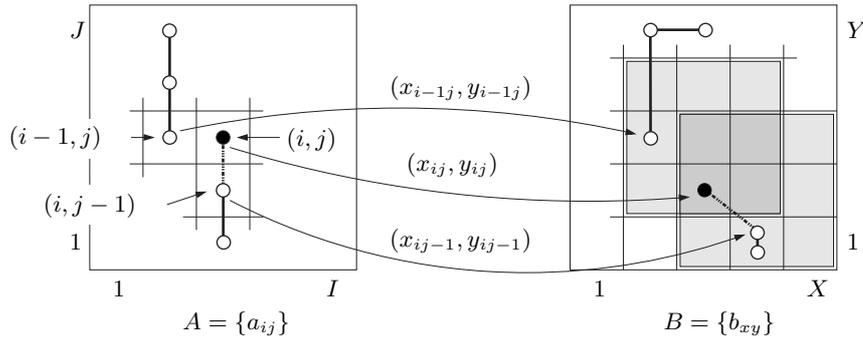


Abbildung 3.8: Darstellung der Abbildungsbedingungen

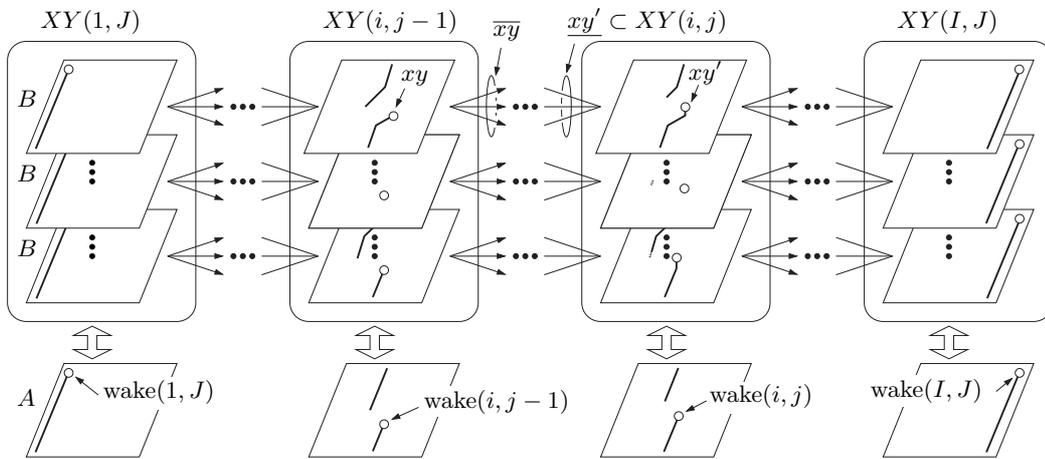


Abbildung 3.9: Darstellung des WW2DHM-Verfahrens

könnte. Abbildung 3.8 stellt die Abbildungsbedingungen (3.10) graphisch dar. Die Abbildung des linken und des unteren Pixel schränken jeweils die gesuchte Pixelabbildung auf einen  $3 \times 3$  Pixel großen Bildbereich ein. Die Schnittfläche dieser Bildbereiche stellt die möglichen Pixelabbildungen dar. Somit kann es für einen Warped Wake  $xy$  maximal 9 Nachfolger  $\overline{xy}$  und maximal 9 Vorgänger  $\underline{xy}$  geben.

Das WW2DHM-Verfahren sucht die kostenminimierende Verformung des Referenzbildes. Dafür wird zu jedem Wake  $wake(i, j)$  die Menge  $XY(i, j)$ , die die zugehörigen Warped Wakes beschreibt, betrachtet. Abbildung 3.9 stellt diesen Ablauf, von links nach rechts betrachtet, graphisch dar. Ein Warped Wake  $xy \in XY(i, j)$  beschreibt durch seine Vorgänger eine bis zur Position  $(i, j)$  mögliche Abbildung des Testbildes auf das Referenzbild. Jeder Warped Wake beinhaltet die Kosten, die aus den nach seiner Abbildung zugehörigen Pixeldistanzen berechnet werden. Wird in dem Pixelschritt  $(i, j)$  die nächste Menge  $XY(i, j)$  aus der Menge  $XY(i, j - 1)$  erzeugt, kann ein erzeugter Warped Wake  $xy \in XY(i, j)$  maximal 9 Vorgänger  $\underline{xy} \subset XY(i, j - 1)$  haben. Aus der Menge  $\underline{xy}$  werden alle Warped Wakes bis auf den mit den niedrigsten

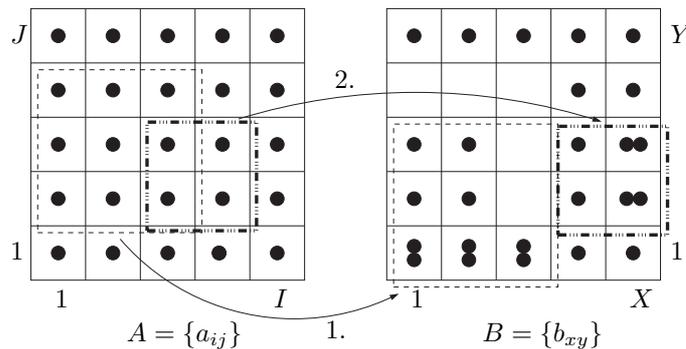


Abbildung 3.10: Darstellung des SA2DHM-Verfahrens

Kosten verworfen, da dieser die kostenminimale Abbildung bis zur Position  $(i, j)$  beschreibt, mit der Einschränkung, dass die letzten  $J$  Pixel, wie durch den Warped Wake beschrieben, abgebildet werden. Somit beschreibt der Warped Wake  $xy(I, J)$  mit den geringsten Kosten zu den restlichen Warped Wakes der Menge  $XY(I, J)$  eine kostenminimierende Warpingabbildung nach dem 2DHMM, indem durch Traceback (engl. „to traceback“: zurückverfolgen) jeweils der vorhergegangene Warped Wake mit seinen Pixelabbildungen betrachtet wird. Die Kosten dieses Warped Wake  $xy(I, J)$  entsprechen der gesuchten Distanz zwischen dem Testbild  $A$  und dem nach dieser Warpingabbildung verformten Referenzbild  $B$ .

### 3.4.2 Simulated Annealing 2DHM-Verfahren

Das 2DHM-Distanzverfahren ist in der hier beschriebenen Form ein  $NP$ -Vollständiges Problem [Keysers & Unger 03]. Solche Probleme werden oft durch heuristische Verfahren wie etwa Simulated Annealing (SA) Verfahren näherungsweise gelöst. Eine Implementierung der 2DHM-Distanzfunktion, die auf diesem Ansatz beruht, wird in dieser Diplomarbeit Simulated Annealing 2DHM (SA2DHM)-Verfahren genannt. Dieses Verfahren ist in der Laufzeit durch die  $T$  Iterationsschritte begrenzt. In jedem Iterationsschritt wird zufällig ein Bildblock des Referenzbildes bestimmt und dieser wird zufällig in eine von 8 möglichen Richtungen um eine Pixelposition verschoben. Abbildung 3.10 stellt die ersten zwei möglichen Verschiebungen von Bildblöcken dar. Die durch die Verschiebung resultierenden Pixelabbildungen müssen die Abbildungsbedingungen erfüllen, sonst wird die entstandene Verformung verworfen. Desweiteren werden Verformungen verworfen, wenn ein von dem Iterationsschritt abhängiger Schwellwert für die Bildkosten überschritten wird. Dadurch kann dieses Verfahren so gestaltet werden, dass bis zur ersten Hälfte der gesamten Iterationsschritte  $T$  eine Verschlechterung der Bildkosten erlaubt wird und danach nur noch Bildkosten verringernde Verformungen beachtet werden. Ein Nachteil dieses Verfahrens kann die schwankende Distanz für denselben Bildvergleich sein, da eine optimale Verformung zufällig bestimmt wird. Dies könnte behoben werden, indem abhängig von dem Referenzbild eine Abfolge von Bildblockverschiebungen vorgegeben werden.

Einen detaillierten Überblick über SA-Verfahren gibt z.B. [Hromkovic 03].

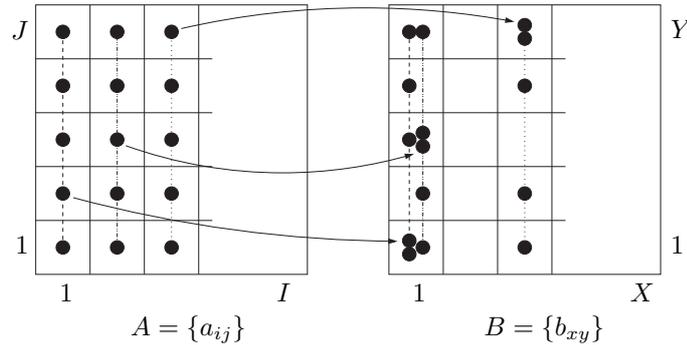


Abbildung 3.11: Darstellung von Spaltenabbildungen nach dem P2DHMM

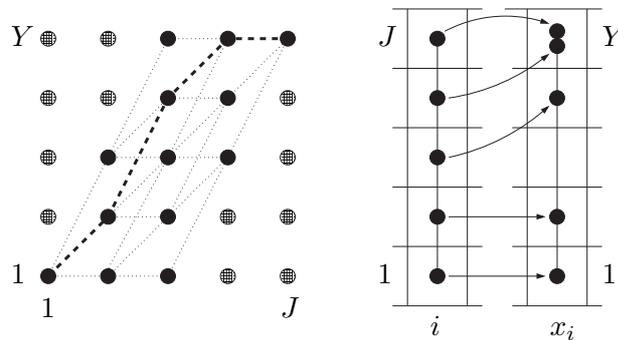


Abbildung 3.12: Darstellung von P2DHM-Pixelabbildungen

### 3.5 Pseudo-2DHM-Modell

Das 2DHM-Modell wurde unter anderem von Agazzi und Kuo [Agazzi & Kuo 93] vereinfacht. Das resultierende Pseudo-2DHM-Modell (P2DHM-Modell) erfüllt nicht alle Abbildungseinschränkungen und beachtet nicht alle möglichen Verformungen des vollständigen 2DHMM. Die Vorteile sind ein geringerer Implementierungsaufwand und eine geringere Laufzeit gegenüber anderen 2DHM-Verfahren. Die Abbildungsbedingungen

$$\begin{aligned}
 & 0 \leq (x_{ij} - x_{i-1j}) \leq 2 \\
 \wedge & \quad |x_{ij} - x_{ij-1}| = 0 \\
 \wedge & \quad 0 \leq (y_{ij} - y_{ij-1}) \leq 2
 \end{aligned} \tag{3.13}$$

schränken eine Abbildung so ein, dass Bildspalten des Testbildes nur auf Bildspalten des Referenzbildes abgebildet werden. Diese Spaltenabbildungen werden nach dem HM-Modell modelliert. Abbildung 3.11 stellt eine solche Spaltenabbildung dar. Alle Pixelabbildungen einer Spalte sind durch die Spaltenabbildungen auf eine Spalte des Referenzbildes festgelegt. Auf welche Bildzeile ein Pixel abgebildet wird, wird ebenfalls nach dem HM-Modell modelliert.

Abbildung 3.12 zeigt mögliche Pixelabbildungen für eine Spaltenabbildung nach dem P2DHM-Modell.

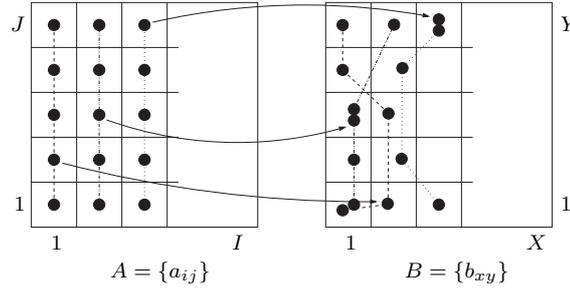


Abbildung 3.13: Darstellung möglicher P2DHMD-Pixelabbildungen

### 3.6 Pseudo-2DHM Distortion Modell

Das Pseudo-2DHM Distortion Modell (P2DHMD-Modell) ist ein Vorschlag für eine Erweiterung des P2DHM-Modells. Es stellte sich in dieser Arbeit heraus, dass eine auf nichtlinearen Verformungen basierende Distanzfunktion, die nach diesem erweiterten Modell modelliert wird und eine später vorgestellte Pixeldistanzfunktion verwendet, bessere Ergebnisse erzielt.

Die Abbildungseinschränkungen des P2DHMM werden nach dem Vorbild des IDM angepasst, indem Spaltenabbildungen nach dem P2DHMM als Ausgangspunkte dienen und Pixelabbildungen nicht nur auf diese Spalten, sondern auch auf die unmittelbaren Nachbarspalten erlaubt werden. Dieses Modell lässt sich durch folgende Abbildungsbedingungen beschreiben:

$$\begin{aligned}
 & 0 \leq (\tilde{x}_i - \tilde{x}_{i-1}) \leq 2 \\
 \wedge & \quad |x_{ij} - \tilde{x}_i| \leq 1 \\
 \wedge & \quad 0 \leq (y_{ij} - y_{ij-1}) \leq 2
 \end{aligned} \tag{3.14}$$

Durch die, nach dem HMM modellierte, Abbildung  $\tilde{x}_1^I$  werden die möglichen Pixelabbildungen auf einen Spaltenbereich festgelegt.

Die Einschränkung des P2DHMM, ganze Spalten aufeinander abzubilden, erwiesen sich für die Modellierung einer Distanzfunktion als hinderlich oder unnötig, da die Verwendung des IDM mit einer erweiterten Pixeldistanzfunktion zu vergleichbaren oder besseren Ergebnissen führt. Dies motivierte die Einführung des P2DHMD-Modells, welches sich bei den getesteten Klassifikationsaufgaben bewährt. Abbildung 3.13 stellt eine mögliche Waringabbildung nach dem P2DHMD-Modell (P2DHMDM) dar. In dem Modell werden kreuzende Pixelabbildungen geduldet, da eine Anpassung des Modells, die dies verhindert, die Bestimmung einer optimalen Waringabbildung erschwert.

## Kapitel 4

# Erweiterungen für Distanzfunktionen

Im Verlauf dieser Arbeit stellt sich heraus, dass die Abbildungseinschränkungen der vorgestellten Verformungsmodelle für untersuchte Klassifikationsaufgaben nicht ausreichen, um gute Ergebnisse zu erzielen. Durch Anpassung und Erweiterung eines Verformungsmodells wird eine auf ihm basierende Distanzfunktion gestaltet. Das Ziel ist die Gestaltung einer Distanzfunktion, die Vergleiche mit Bildern derselben Klasse durch kleine Distanzen und Bilder verschiedener Klassen durch große Distanzen bewertet. In diesem idealen Fall könnte die Distanzfunktion als Diskriminantenfunktion in einem Klassifikationssystem verwendet werden und es würde ein Referenzbild pro Klasse genügen.

In diesem Kapitel werden Anpassungen und Erweiterungen für auf Verformungsmodellen basierende Distanzfunktionen vorgestellt.

### 4.1 Bildvorverarbeitung

Eine Vorverarbeitung der Bilddaten beeinflusst die Ergebnisse einer Distanzfunktion. Das Ziel einer solchen Bildvorverarbeitung sind Distanzwerte, die Bildvergleiche zuverlässiger bewerten, um die Fehlerrate einer Klassifikation zu senken. Eine Vorverarbeitung, die Bilddaten reduziert, kann zur Verringerung der Rechenzeit eines Klassifikationssystems verwendet werden.

In den folgenden Abschnitten werden verschiedene Bildvorbehandlungsverfahren vorgestellt.

#### 4.1.1 Segmentierung und Padding

Segmentierung bezeichnet die Extraktion eines Bildausschnitts. Dieser Bildausschnitt kann durch das Auslassen von Spalten und Zeilen des Bildrandes gewonnen werden. Für die Bestimmung dieses Ausschnitts gibt es verschiedene Methoden, z.B. die Verwendung der äußersten Kante des zugehörigen Kantenbildes [Jähne 02, S. 449–462].

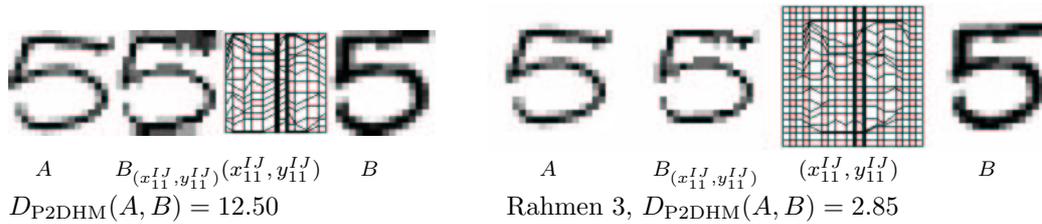


Abbildung 4.1: Umgehung der ursprünglichen Randbedingung

Im Gegensatz zur Segmentierung bezeichnet Padding das Hinzufügen von Spalten und Zeilen zum Bildrand. Die Veränderung der Bilddaten durch Segmentierung oder Padding beeinflusst die möglichen Pixelabbildungen der vorgestellten Verformungsmodelle. Durch Padding zweier zu vergleichender Bilder mit jeweils einem Pixelrahmen, wird die Einschränkung der Randbedingung für die ursprünglichen Bildränder umgangen. Abbildung 4.1 stellt eine gewonnene Verformung von ursprünglichen Bildern und die resultierende Verformung durch das Hinzufügen von Rahmen mit der Breite von 3 Pixel dar. Die Pixelwerte der Rahmen wurden auf den Hintergrundwert der Bilder gesetzt. In diesem Beispiel ergibt sich durch die auf diese Weise erweiterte Distanzfunktion, die zusätzlich mögliche Bildverformungen berücksichtigt, eine geringere Distanz.

#### 4.1.2 Rotation und Spiegelung

Das WW2DHM-Verfahren unter Anwendung von Beamsearch und das P2DHM-Verfahren können abhängig von der Implementierung zu unterschiedlichen Ergebnissen führen. Das P2DHM-Verfahren kann anstelle von Spaltenabbildungen die Zeilenabbildungen zuerst berücksichtigen. Die Ergebnisse des WW2DHM-Verfahrens unter Verwendung von Beamsearch sind abhängig von der Startposition und der Richtung der Pixelschritte, die in der Implementierung festgelegt werden. Werden die Warped Wake Mengen vollständig betrachtet, wird unabhängig von der Implementierung eine distanzminimierende Verformung berechnet. Anstatt die verschiedenen Implementierungsmöglichkeiten umzusetzen, können dieselben Ergebnisse durch Rotation in  $90^\circ$  Schritten und durch vertikale oder horizontale Spiegelung der Bilder erzielt werden. Abbildung 4.2 zeigt die von der Implementierung abhängigen Ergebnisse des P2DHM-Verfahrens. Die in diesem Beispiel dargestellte Verformung, die auf Zeilenabbildungen beruht, wird durch eine Rotation der Bilder um  $90^\circ$  und dem P2DHM-Verfahren, das auf Spaltenabbildungen beruht, berechnet.

#### 4.1.3 Skalierung

Durch Skalierung eines Bildes wird die Pixelauflösung und somit die Bildgröße des ursprünglichen Bildes verändert. Die Pixelwerte des skalierten Bildes werden dazu aus den ursprünglichen Bilddaten interpoliert. Es gibt verschiedene Verfahren für eine solche Interpolation, z.B. die lineare Interpolation oder die Interpolation durch Verwendung von

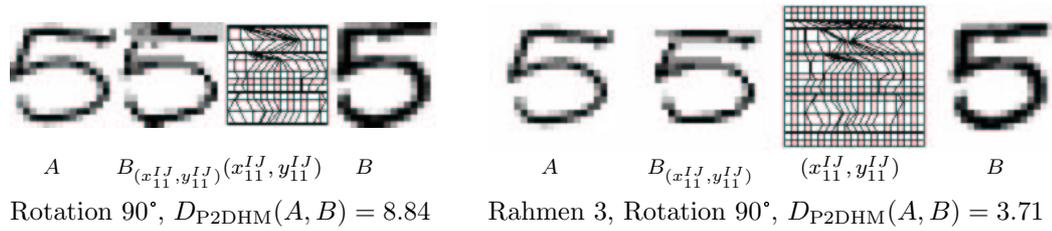


Abbildung 4.2: Rotation um 90° gegenüber Abbildung 4.1

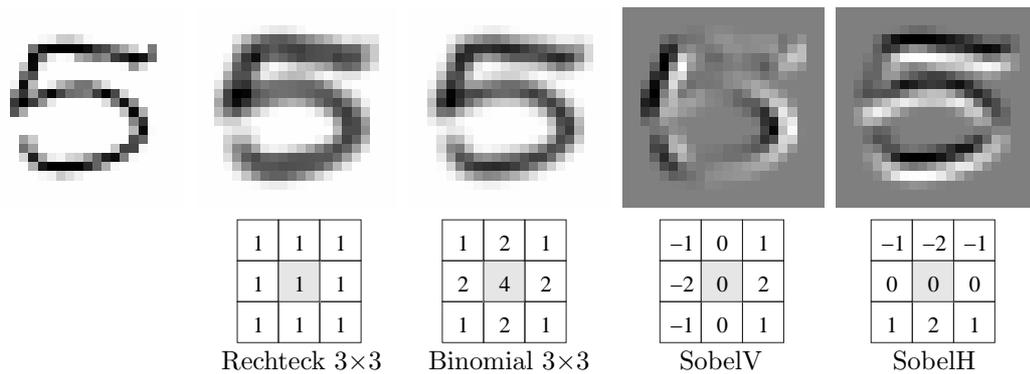


Abbildung 4.3: Beispielbild mit verschiedenen Filtern

Splines [Jähne 02, S. 282–295].

Die Bildauflösung kann die durch ein HMM berechnete Distanz beeinflussen. Beinhaltet z.B. ein 1D-Signal einen Wert, der nach dem HMM nicht zugeordnet werden kann, kann dieser Wert ausgelassen werden. Wird hingegen dasselbe Signal in der doppelten Auflösung betrachtet, muss in der Anpassung durch das HMM einer der zwei Werte berücksichtigt werden.

#### 4.1.4 Filteroperationen

Die Pixelwerte eines Bildes können durch die Anwendung eines Filters transformiert werden. Diese Transformation ist abhängig von dem verwendeten Filter und seiner Größe [Jähne 02, S. 103–130]. Der Sobelfilter wird z.B. zur Berechnung des vertikalen und des horizontalen Gradientenbildes verwendet [Jähne 02, S. 350–351]. Der Rechteckfilter sowie der Binomialfilter können zur Glättung eines Bildes verwendet werden [Jähne 02, S. 302–311]. Abbildung 4.3 stellt die in den Experimenten verwendeten Filter und Beispiele für derart transformierte Bilder dar.

#### 4.1.5 Gradientenbilder

Im Abschnitt 4.1.4 wird durch den Sobelfilter eine Möglichkeit zur Bestimmung des horizontalen und vertikalen Gradienten eines Bildes angegeben. Die durch diese Filte-

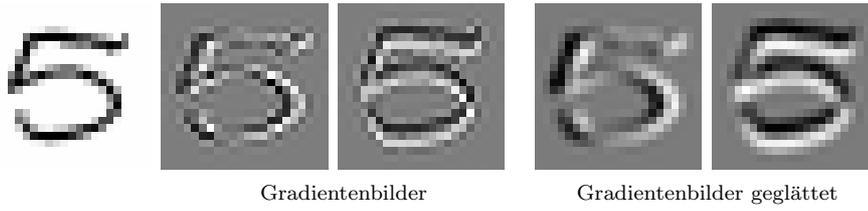


Abbildung 4.4: Beispiel für Gradientenbilder durch Splinekoeffizienten

operation gewonnenen Gradientenbilder sind relativ unscharf [Jähne 02, S. 350–351]. Eine Methode, die auf der Verwendung von Splinekoeffizienten beruht, erzeugt weniger fehlerhafte (visuell schärfere) Gradientenbilder [Jähne 02, S. 344]. In Abbildung 4.4 werden Gradientenbilder, die auf Splinekoeffizienten beruhen, und zusätzlich die durch den vorgestellten Binomialfilter geglätteten Gradientenbilder dargestellt.

Die Verwendung der durch den Sobelfilter bestimmten Gradientenbilder führt in den durchgeführten Experimenten zu besseren Ergebnissen, als die durch Splinekoeffizienten bestimmten Gradientenbilder. Werden diese durch den  $3 \times 3$  Binomialfilter geglättet, passen sich die Ergebnisse an.

## 4.2 Erweiterung der Pixeldistanzfunktion

Durch die Bildvorverarbeitung kann eine distanzminimierende Verformung beeinflusst werden. Die Verwendung der Gradientenbilder zur Bestimmung der Distanz mit den vorgestellten Verfahren verbessert die Klassifikationsergebnisse deutlich. Dies motiviert die Untersuchung von Pixeldistanzfunktionen, die Kontextinformationen berücksichtigen, da die Gradientenbilder diese Informationen zum Teil in den Pixelwerten enthalten.

Die Pixeldistanzfunktion wurde zur Bestimmung der Distanz zwischen zwei Bildausschnitten erweitert. Das Zentrum dieser Bildausschnitte sind die zu vergleichenden Pixelwerte. Die Pixeldistanzfunktion  $d_{\text{rec}5 \times 5}(a_{ij}, b_{xy})$  berechnet die quadrierte euklidische Distanz zwischen den zwei  $5 \times 5$  Pixel großen Bildausschnitten, deren Positionen durch die Koordinaten der Pixel  $(i, j)$  und  $(x, y)$  vorgegeben sind. Allgemein kann eine solche Pixeldistanzfunktion, die die Distanz anhand quadratischer Bildausschnitte bestimmt, wie folgt angegeben werden:

$$d_{V \times V}(a_{ij}, b_{xy}) = \sum_{v=1}^V \sum_{w=1}^V z_{vw} \cdot (a_{i+v-M, j+w-M} - b_{x+v-M, y+w-M})^2 \quad (4.1)$$

mit  $z \in R^{V \times V}$ ,  $M = \frac{V-1}{2}$ ,  $V = 1, 3, 5, 7 \dots$

Die Gewichtung der in dieser Funktion berechneten Distanzen erfolgt über den Gewichtsparameter  $z$ . Dieser kann in einem Trainingsprozess durch die Berücksichtigung der beobachteten Varianzen bestimmt werden. Ein Beispiel für eine in dieser Arbeit

verwendete Pixeldistanzfunktion ist:

$$d_{\text{bin}3 \times 3}(a_{ij}, b_{xy}) = \sum_{v=1}^3 \sum_{w=1}^3 z_{vw} \cdot (a_{i+v-1, j+w-1} - b_{x+v-1, y+w-1})^2 \quad (4.2)$$

mit  $z = \frac{1}{4}(1, 2, 1) \cdot \frac{1}{4}(1, 2, 1)^T$ .

Durch Verwendung der  $d_{\text{rec}3 \times 3}$  Pixeldistanzfunktion konnten mit dem einfachen ID-Verfahren gute Ergebnisse erzielt werden.

Die Verwendung dieser Pixeldistanzfunktion ist vergleichbar mit einer Klassifikationsmethode, die lokale Merkmale verwendet [Paredes & Pérez<sup>+</sup> 01], indem die Abbildungseinschränkungen aufgehoben werden und die kostenminimierenden Pixelabbildungen eines Testbildes zusätzlich auf verschiedene Referenzbilder erfolgen darf.

Die Pixeldistanzfunktion kann durch einen Threshold-Parameter erweitert werden, der den maximalen Distanzwert einer Pixeldistanzfunktion beschränkt (engl. „threshold“: Schwellwert).

### 4.3 Anpassung der Verformungsmodelle

Die vorgestellten Erweiterungen beeinflussen die berechnete ideale Verformung nicht ausreichend genug, um eine Anpassung an ein Bild einer anderen Klasse zu verhindern. Durch die Verwendung von erweiterten Strafkostenfunktionen können die Verformungsmodelle zusätzlich gestaltet werden. Es können verschiedene Strafkostenfunktionen miteinander kombiniert und abhängig von der Klasse des Bildes modelliert werden. Diese Modellierung kann anhand der Trainingsdaten erfolgen. In einem Trainingsprozess kann eine Strafkostenfunktion abhängig von der Klasse, dem Referenzbild oder sogar von jedem Pixel modelliert werden.



## Kapitel 5

# Reduzierung der Rechenzeit

### 5.1 Motivation

Neben dem Ziel, ein Klassifikationssystem für eine minimale Fehlerrate zu optimieren, gibt es das Ziel, die Rechenzeit auf ein Maximum zu begrenzen. Die Texteingabe durch einen Stift gehört beispielsweise bei einem Personal Digital Assistant (PDA) mittlerweile zum Standard. Die ersten PDAs hatten für rechenintensive und speicherintensive Verfahren nicht genügend Leistung. Diese PDAs verwendeten so genannte „Graffiti“-Schriftzeichen, um die Fehlerrate bei einer für den Anwender vertretbaren Rechenzeit gering zu halten. Die Schreibweise dieser Schriftzeichen erleichtert die Klassifikationsaufgabe für einfache Verfahren. Durch die wachsende Leistung der PDAs werden aufwändigere Verfahren, die geringere Fehlerraten bei gewöhnlichen Schriftzeichen erzielen, möglich. Wenn ein Verfahren die durch die begrenzte Leistung der Hardware maximale Rechenzeit oder den maximalen Speicherverbrauch nicht einhält, muss dieses Verfahren zur Einhaltung der Leistungsgrenzen modifiziert werden oder es kann erst bei leistungsstärkeren Systemen verwendet werden. In dem folgenden Unterkapitel werden die Laufzeiten, der Speicherbedarf und die Rechenzeiten der hier vorgestellten Verfahren beschrieben. Die weiteren Abschnitte dieses Kapitels beschreiben verschiedene Ansätze zur Reduzierung der Rechenzeit und des Speicherbedarfes des hier vorgestellten Klassifikationssystems.

### 5.2 Laufzeit

Tabelle 5.1 stellt die Laufzeiten und den Speicherbedarf verschiedener Verfahren zur Bestimmung der Distanz zweier Bilder abhängig von ihrer Bildgröße dar. Die Angaben beziehen sich auf quadratische Bilder gleicher Größe, die  $I \times I$  vielen Pixeln entspricht.

Der Speicherbedarf wird als der zusätzlich zu den Bilddaten nötige Bedarf von Speicher verstanden, der zur Berechnung der Distanz und der zugehörigen Verformung verwendet wird. Der Speicherbedarf für die Bilddaten ist bei jedem Verfahren gleich und ist linear abhängig von der Bildgröße  $O(I^2)$ .

Der benötigte Speicherbedarf für die euklidische Distanz sowie für das ID-Verfahren

Tabelle 5.1: Laufzeiten der Verformungsverfahren

	eukl. Distanz	ID	P2DHM	WW2DHM	SA2DHM	
Speicherbedarf*	$O(1)$	$O(1)$	$O(I^2)$	$O(I^3 9^I)$	$O(I^2 B)$	$O(I^2)$
Laufzeit	$O(I^2)$	$O(I^2)$	$O(I^4)$	$O(I^3 9^I)$	$O(I^2 B)$	$O(I^2 T)$

$$I = J = X = Y, \quad \text{Beamsize } B, \quad \text{Iterationen } T$$

\*zusätzlicher Bedarf zu den Bilddaten

ist konstant. Die Laufzeit beider Verfahren ist linear abhängig von der Bildgröße des Testbildes, die der Pixelanzahl entspricht, da jedes Pixel des Testbildes in einer konstanten Zeitspanne abgearbeitet wird. Da von quadratischen Bildern ausgegangen wird, können die Laufzeiten dieser Verfahren auch quadratisch abhängig von der Bildhöhe oder der Bildbreite angegeben werden.

Die Laufzeit des P2DHM-Verfahrens ist quadratisch abhängig von der Bildgröße, da die Anzahl der nach dem HMM zu vergleichenden Spalten quadratisch abhängig von der Bildbreite wächst. Die Pixelvergleiche, die ebenfalls nach dem HMM modelliert werden und für jeden Spaltenvergleich nötig sind, wachsen ebenfalls quadratisch und sind abhängig von der Bildhöhe. Aus Abbildung 3.4 ist diese quadratische Abhängigkeit von den Längen der nach dem HMM anzupassenden Signale ersichtlich. Der Speicherbedarf des P2DHM-Verfahrens ist quadratisch abhängig von der Bildhöhe oder der Bildbreite, da der HM-Graph der Spaltenverformung vollständig benötigt wird, um die minimierende Abbildung zu bestimmen. Anstatt alle Pixelabbildungen einer Spaltenabbildung zu speichern, können diese Pixelabbildungen für die minimale Spaltenabbildung ein weiteres Mal berechnet werden.

Die Laufzeit und der Speicherbedarf des WW2DHM-Verfahrens sind abhängig von der Anzahl der betrachteten Warped Wakes in einem Pixelschritt, da dieses Verfahren in jedem Pixelschritt die Menge der Warped Wakes abarbeitet. Die Anzahl der Warped Wakes einer solchen Menge ist abhängig von der Bildhöhe und der Bildbreite. Da zu jeder Position eines Warped Wakes maximal 9 Vorgänger existieren, dieser eine Länge von  $I$  Pixel hat und sich Warped Wakes durch unterschiedlich weit auseinanderliegende Brüche an der Bildkante unterscheiden, gibt es maximal  $I 9^I$  viele Warped Wakes pro Pixelposition [Uchida & Sakoe 98]. Die Laufzeit und der Speicherbedarf des WW2DHM-Verfahrens sind somit exponentiell abhängig von der Bildhöhe. Durch die Begrenzung der beachteten Warped Wakes in einem Pixelschritt auf die Beamsize  $B$  Besten, die Warped Wakes, die bisher die niedrigsten Bildkosten verursachen, ist die Laufzeit scheinbar linear abhängig von der Bildgröße. Der Beamsize  $B$  muss jedoch für größere Bilder ebenso vergrößert werden, um vergleichbar gute Verformungen zu berechnen.

Die Laufzeit des SA2DHM-Verfahrens ist linear abhängig von der Bildgröße und der Anzahl der Iterationsschritte  $T$ . Die Zahl der Iterationsschritte  $T$  muss vergleichbar mit dem Beamsize  $B$  des WW2DHM-Verfahrens an die Bildgröße angepasst werden. Der

Speicherbedarf ist linear abhängig von der Bildgröße, da in jedem Iterationsschritt die derzeitige beste Warpingabbildung betrachtet wird.

### 5.3 Implementierung

Die Rechenzeit dieser Verfahren ist abhängig von ihrer Implementierung. Der Implementierung des WW2DHM-Verfahrens wurde besondere Aufmerksamkeit geschenkt, da dieses Verfahren im Verhältnis zu den anderen hier vorgestellten Verfahren sehr aufwändig ist. Dieses Verfahren wurde mittels dynamischer Programmierung implementiert, indem die erste Warped Wake Menge  $XY(1, J)$  initialisiert wird und jede weitere Menge aus der vorherigen Menge generiert wird. Die Menge der Warped Wakes eines Pixelschrittes werden in einem Array gespeichert. Auf dieses Array wird über eine Hashfunktion, die von den Pixelpositionen eines Warped Wakes abhängig ist, zugegriffen, um effizient den Warped Wake von maximal 9 Gleichen zu finden, der den geringsten Kosten entspricht. Ein Warped Wake wird in diesem Array durch seine Pixelpositionen, den Abstand des Bruches am Bildrand und die bisherigen Bildkosten beschrieben. Dabei wird die erste Pixelposition eines Warped Wakes gespeichert und die restlichen Pixelpositionen werden durch die Differenzen der Positionen zu dem vorherigen Pixel beschrieben. Diese Darstellung wurde für eine effiziente Speichernutzung gewählt. Die Initialisierung der ersten Menge von Warped Wakes gleicht durch die Randeinschränkung einem HM Modell. Der  $A^*$ -Algorithmus wird hier verwendet, um diese Menge auf die Beamsize  $B$  besten Warped Wakes zu begrenzen. Die folgenden Mengen von Warped Wakes werden durch die Histogramm Pruning Methode auf Beamsize  $B$  viele begrenzt. Ebenso gibt es die Möglichkeit, die Warped Wake Mengen von mehreren Referenzbildern gleichzeitig zu betrachten und einen Schwellwert (engl. Threshold), der abhängig von dem vorher besten Warped Wake der betrachteten Referenzbilder ist, für die Begrenzung der Anzahl der Warped Wakes zu verwenden.

Diese Beamsearch-Methode wird ebenso in der Spracherkennung zur Reduzierung des Rechenaufwandes angewandt [Ney 01]. Implementiert wurden beide Beamsearch-Methoden: die Begrenzung der beachteten Warped Wakes in einem Pixelschritt auf einen fixen Beamsize  $B$  und die Variabilität durch einen Threshold zum vorherigen minimalen Warped Wake unter Beachtung mehrerer Referenzbilder.

### 5.4 Rechenzeit

Die Rechenzeit und die Ergebnisse der Verfahren werden an einem Beispiel dargestellt. Abbildung 5.1 zeigt die resultierende Verformung, den erzielten Distanzwert und die benötigte Rechenzeit der Verfahren für einen Bildvergleich. Die Rechenzeit wurde so normiert, dass der Bildvergleich durch die euklidische Distanz einer Zeiteinheit entspricht.

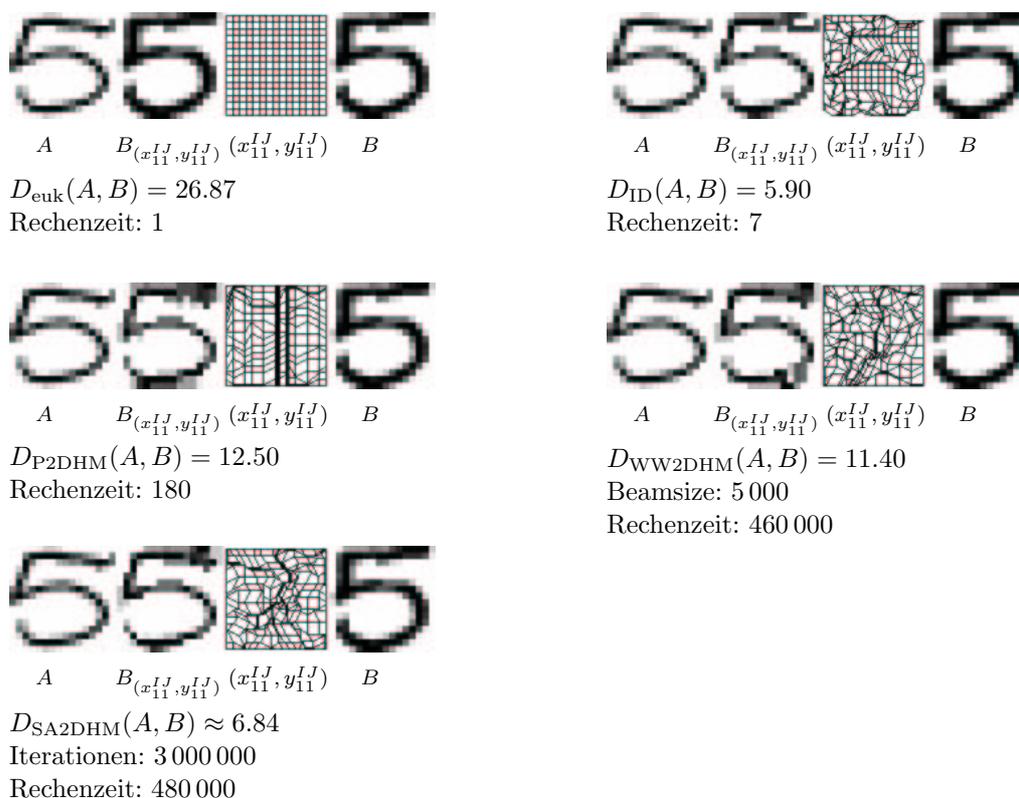


Abbildung 5.1: Verformungsbeispiele verschiedener Distanzverfahren

## 5.5 Begrenzung der Referenzdaten

Die implementierten Verformungsverfahren benötigen für die Bestimmung der Distanz zwischen zwei Bildern eine relativ hohe Rechenzeit. Die Klassifikation durch die Berechnung dieser Distanzen für jedes Testbild mit allen Referenzbildern einer Datensammlung kann äußerst zeitintensiv sein. Eine Möglichkeit, die Rechenzeit eines solchen Klassifikationsverfahrens zu verkürzen, ist die Reduzierung der zur Klassifikation verwendeten Referenzbilder. Durch diese einfache Methode können Experimente an einer Datensammlung auf einen Bruchteil der eigentlichen Rechenzeit verkürzt werden. Vergleicht man die so erzielten Ergebnisse, muss die verwendete Methode zur Begrenzung der Menge der Referenzbilder berücksichtigt werden.

### 5.5.1 Vorauswahl der Referenzbilder

Durch Verwendung einer relativ schnellen Distanzfunktion kann zu jedem Testbild eine Vorauswahl der Referenzbilder erfolgen. Anhand dieser kann die eigentliche Klassifikation durch die Anwendung eines rechenintensiveren Verfahrens erfolgen. Wird dabei die NN-Regel verwendet, können die durchzuführenden Vergleiche unter Umständen weiter reduziert werden, indem die Vergleiche der bis dahin besten Klasse zurückgestellt

werden. Denkbar ist auch, die Referenzdaten in einer Baumstruktur zu verwalten und abhängig von den Distanzergebnissen der Bilder eines Vaterknotens die zugehörigen Bilder seiner Söhne für Vergleiche zu verwenden oder auszulassen [Seidl 02].

### 5.5.2 Reduzierung der Referenzdaten durch Verwendung von Prototypen

Die nötigen Distanzberechnungen für die Klassifikation eines Bildes können ebenso durch die Verwendung von Prototypen reduziert werden, indem die Trainingsdaten auf einen oder wenige Prototypen pro Klasse reduziert werden. Eine einfache Methode zur Bestimmung eines Prototyps pro Klasse ist die Berechnung des Durchschnittsbildes anhand der Referenzbilder dieser Klasse. Diese Prototypen sind visuell äußerst unscharf, da sie keine Bildanpassungen berücksichtigen und aus der Überlagerung vieler Referenzbilder entstehen. Durch die Anwendung der Verformungsmodelle zur Bestimmung der Durchschnittsbilder werden visuell deutlich schärfere Prototypen erzeugt. Die Methode zur Bestimmung der Prototypen erfolgt nach dem in der Spracherkennung angewandten Verfahren [Ney 01].

Durch den Expectation Maximization (EM)-Algorithmus im Cluster-Verfahren und die Verwendung dieser schärferen Durchschnittsbilder als Mittelwerte, können mehrere Prototypen für eine Klasse bestimmt werden, indem die Bilder einer Klasse unterschiedlichen Clustern zugeordnet werden und als Ergebnis jeweils ein Prototyp pro Cluster ausgegeben wird [Ney 01].



## Kapitel 6

# Datensammlungen und Experimente

### 6.1 US Postal Service Datensammlung

Die US Postal Service Datensammlung (USPS-Datensammlung) besteht aus Bildern handgeschriebener arabischer Ziffern. Dieser Korpus ist aus binären Bildern der CEDAR-Datensammlung (CEDAR=Center of Excellence for Document Analysis and Recognition) entstanden [Wang & Srihari 88]. Die USPS-Daten wurden aus diesen binären Bildern durch eine lineare Transformation der Größe erzeugt [LeCun & Boser<sup>+</sup> 89]. Jedes Bild stellt eine von 10 Ziffern dar und ist einer von 10 Klassen zugeordnet. Die 9 298 Bilder der Datensammlung sind in 7 291 Trainingsbilder und 2 007 Testbilder unterteilt. Jedes Bild ist segmentiert und hat eine Größe von 16×16 Pixel. Die Pixelwerte repräsentieren Grauwerte, die nach der Vorverarbeitung der binären Bilder verwendet werden.

Die Daten entstanden durch Digitalisierung handgeschriebener Postleitzahlen von amerikanischen Briefumschlägen. Fehler, die bei der Segmentierung der Ziffern gemacht wurden, wurden übernommen, da solche Fehler ebenso in der Praxis auftreten. Die Aufgabe ist die Klassifikation der Testbilder unter Verwendung der Trainingsbilder. Die USPS-Datensammlung ist frei erhältlich<sup>1</sup> und wird von vielen Forschungsgruppen zum Vergleich ihrer Klassifikationssysteme verwendet.

---

<sup>1</sup><http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/data.html>  
<ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data>  
<http://www.kernel-machines.org/data.html>



Abbildung 6.1: Je ein Beispielbild für jede Klasse der USPS-Datensammlung

Tabelle 6.1: Ergebnisse für die USPS-Datensammlung

Verfahren		Fehler[%]
menschliche Fehlerrate	[Simard & LeCun <sup>+</sup> 93]	2.5
	[Dong & Krzyzak <sup>+</sup> 01]	1.5
euklidische Distanz, NN-Regel		5.6
Relevanz Vektoren	[Tipping 00]	5.1
Neuronales Netz (LeNet1)	[LeCun & Boser <sup>+</sup> 90]	*3.4
invariante Support Vektoren	[Schölkopf & Simard <sup>+</sup> 98]	3.0
(1) lokale Merkmale, $k$ -NN-Regel	[Keysers & Paredes <sup>+</sup> 02]	3.0
Neuronales Netz, Boosting	[Drucker & Schapire <sup>+</sup> 93]	*2.6
Tangentendistanz	[Simard & LeCun <sup>+</sup> 94]	*2.5
(2) erweiterte Tangentendistanz	[Keysers & Dahmen <sup>+</sup> 00]	2.4
erweiterte Support Vektoren	[Dong & Krzyzak <sup>+</sup> 02b]	2.2
Kombination von (1) und (2)	[Keysers & Paredes <sup>+</sup> 02]	2.0
nichtlineare Verformungsmodelle	diese Arbeit	2.1

\*Trainingsdaten erweitert mit maschinell erstellten Zahlen

### 6.1.1 Stand der Technik

Die leichte Verfügbarkeit der Datensammlung und die im Verhältnis zu vergleichbaren Datensammlungen, z.B. NIST-Korpus (NIST=National Institute of Standards and Technology), schwierige Aufgabe, welche sich durch die höheren Fehlerraten widerspiegelt, hat vermutlich zur Popularität der USPS-Datensammlung geführt. Unter Verwendung der euklidischen Distanz und der NN-Regel wird eine Fehlerrate von 5.6% erzielt. Bessere Ergebnisse werden durch die unterschiedlichsten Ansätze für Klassifikationsverfahren erreicht. Das erweiterte Tangentendistanz Verfahren erzielt eines der besten Ergebnisse mit einer Fehlerrate von 2.4%. Die Ergebnisse werden oft mit der von Simard [Simard & LeCun<sup>+</sup> 93] angegebenen menschlichen Fehlerrate von 2.5% verglichen. Diese ist jedoch subjektiv vom Betrachter abhängig und kann daher stark variieren. In dem Bericht [Dong & Krzyzak<sup>+</sup> 01] wird eine menschliche Fehlerrate von 2.5% angezweifelt und in Tests wurde eine durchschnittliche menschliche Fehlerrate von 1.5% festgestellt. In Tabelle 6.1 werden weitere Ergebnisse, die durch unterschiedliche Klassifikationssysteme erzielt werden, aufgeführt. Das in dieser Diplomarbeit vorgestellte Klassifikationssystem erzielt eine Fehlerrate von 2.1%.

### 6.1.2 Experimente und Ergebnisse

Der Graph der Abbildung 6.2 stellt die Fehlerrate der USPS-Datensammlung abhängig von der Gewichtung einer Strafkostenfunktion dar. Diese Fehlerraten resultieren aus der Klassifikation anhand der NN-Regel, der P2DHM-Distanz, der quadrierten eukli-

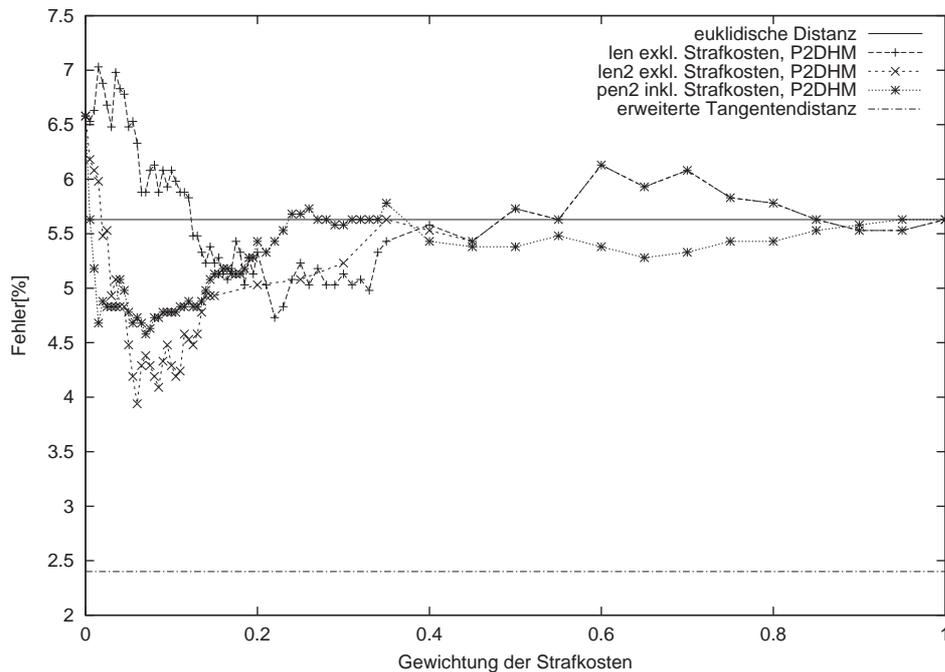


Abbildung 6.2: Ergebnisse mit unterschiedlichen Strafkostenfunktionen für die USPS-Datensammlung und das P2DHM-Verfahren

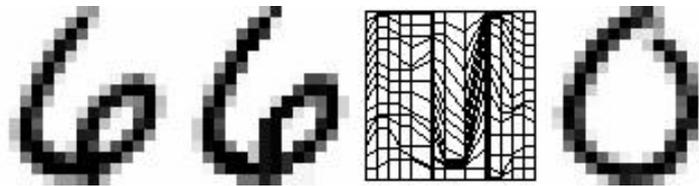


Abbildung 6.3: Beispiel einer unerwünschten Verformung durch das P2DHM-Verfahren

sehen Pixeldistanz und den auf den Wertebereich von  $[0, 1]$  normierten Bildwerten. Die Referenzbilder wurden durch die Shift  $3 \times 3$  Methode vervielfacht. Es werden durch die euklidische Distanz  $D_{\text{euk}}$  pro Testbild 100 Referenzbilder nach geringsten Distanzwerten ausgewählt. Die P2DHM-Distanzwerte werden durch Verwendung der  $R_{\text{len}}$ ,  $R_{\text{len}^2}$  und  $R_{\text{pen}^2}$  Strafkostenfunktionen, inklusive sowie exklusive der Strafkosten berechnet. Die niedrigste Fehlerrate, die bei diesen Versuchen erzielt wurde, beträgt 3.9%. Die relativ hohe Fehlerrate von 6.6%, die ohne zusätzliche Verformungseinschränkungen einer Strafkostenfunktion erzielt wird, lässt sich durch zu extreme, ungewollte Bildverformungen erklären.

Abbildung 6.3 stellt einen Klassifikationsfehler aufgrund einer unerwünschten Verformung dar. Unerwünscht sind Verformungen die den Distanzwert zwischen Bildern unterschiedlicher Klassen verringern.

Für die USPS-Klassifikationsaufgabe sind die Verformungseinschränkungen der vor-

gestellten Modelle alleine unzureichend, da bessere Ergebnisse erzielt werden, wenn keine Verformungen ermöglicht werden, was der euklidischen Distanz entspricht.

Durch Strafkostenfunktionen werden die möglichen Verformungen zusätzlich zu den Einschränkungen des Verformungsmodells beschränkt. Unter Verwendung verschiedener Strafkostenfunktionen konnten die Klassifikationsergebnisse verbessert werden. Bei diesen Versuchen wurde durch die absolute Strafkostenfunktion  $R_{\text{len}^2}$  das beste Ergebnis erzielt. Die Fehlerrate von 3.9% wird durch die Gewichtung  $\alpha = 0.6$  dieser Strafkostenfunktion  $R_{\text{len}^2}$  erzielt. Mit diesen Einstellungen wächst die Strafe für die Verschiebung eines Pixel quadratisch abhängig von der Weite der Pixelverschiebungen. Abhängig von der Gewichtung der Strafkostenfunktion wird die Anzahl der möglichen Verformungen reduziert.

Die Funktionen  $R_{\text{len}}$  und  $R_{\text{len}^2}$  schränken die Weite, die ein Pixel von seinem Ursprung verschoben wird, ein. Die relative Strafkostenfunktion  $R_{\text{pen}^2}$  berücksichtigt ausschließlich die Pixelnachbarn und ignoriert die absolute Weite einer Pixelverschiebung. Im Gegensatz zu den absoluten Strafkostenfunktionen wird die Anzahl der lokalen Verformungen abhängig von den Abbildungen der Pixelnachbarn bestraft.

Es wurde ebenfalls eine Kombination aus den Strafkostenfunktionen  $R_{\text{len}^2}$  und  $R_{\text{pen}^2}$  untersucht. Hierfür wurden die Strafkosten der Funktionen, nachdem sie einzeln gewichtet worden waren, aufsummiert und die Distanzwerte inklusive der Strafkosten berechnet. Durch diese Kombination der Strafkostenfunktionen konnte keine weitere Reduzierung der Fehlerrate erreicht werden.

Unerwünschte Verformungen können abhängig von der Klasse des verformten Bildes sein. Um dies zu berücksichtigen, kann eine Strafkostenfunktion abhängig von der Klasse oder dem Referenzbild gestaltet werden. Uchida und Sakoe arbeiten mit diesem Ansatz [Uchida & Sakoe 03b].

Eine andere Möglichkeit, unerwünschte Verformungen zu verhindern, sind erweiterte Pixelmerkmale. Um solche zu erhalten, wurde das horizontale und das vertikale Gradientenbild verwendet und ein Pixelwert als Tupel dieser Bildwerte aufgefasst. Diese Pixelwerte sind differenzierter als die Grauwerte, da Gradientenpixelwerte, die Werte an benachbarten Positionen im Bild berücksichtigen. Im Gegensatz zu den Grauwerten unterscheiden sich z.B. die linken und rechten Randpixel einer dargestellten Ziffer durch die Verwendung der Gradientenbilder sehr deutlich.

Tabelle 6.2: USPS-Ergebnisse der Verformungsverfahren

Verfahren	Fehler[%]
euklidische Distanz, NN-Regel	
+ Grauwerte	5.6
+ Rahmen 3, Shift 3×3	4.8
+ Rahmen 3, SobelV, SobelH	6.3
+ Shift 3×3	5.0
WW2DHM, SobelV, SobelH, $R_{\text{len}}$ , Rahmen 3, NN-Regel	
+ $B=200$ , Beste 50 $D_{\text{euk}}$	3.8
+ $d_{\text{rec}3\times3}$	3.2
+ $B=1000$	3.0
+ Beste 100 $D_{\text{euk}}$	2.7
P2DHM, Beste 100 $D_{\text{euk}}$ , Rahmen 3, NN-Regel	
+ Shift 3×3, Grauwerte	6.6
+ $R_{\text{len}^2}$	3.9
+ SobelV, SobelH	2.5
+ Grauwerte	2.9
+ Rotation 90°	3.1
+ SplineGradientV, SplineGradientH	3.4
+ $d_{\text{rec}3\times3}$	4.0
+ Binomialfilter 3×3	2.7
+ Grauwerte, $d_{\text{rec}3\times3}$	2.9
+ Grauwerte, $d_{\text{bin}3\times3}$	2.9
+ Grauwerte, $d_{\text{rec}5\times5}$	2.8
+ SobelV, SobelH, $d_{\text{rec}3\times3}$	2.7
P2DHM, Rahmen 3, SobelV, SobelH, NN-Regel	2.4
ID, Beste 100 $D_{\text{euk}}$ , NN-Regel	
+ Grauwerte, $W=1$	9.0
+ SobelV, SobelH, $W=1$	3.3
+ $d_{\text{rec}3\times3}$ , $W=2$	2.7
+ Grauwerte, $d_{\text{rec}3\times3}$ , $W=2$	3.6
+ Grauwerte, $d_{\text{bin}3\times3}$ , $W=2$	3.7
+ Grauwerte, $d_{\text{rec}5\times5}$ , $W=2$	3.4
ID, SobelV, SobelH, $d_{\text{rec}3\times3}$ , NN-Regel, $W=1$	2.4
P2DHMD, Beste 200 $D_{\text{euk}}$ , Rahmen 3	
+ SobelV, SobelH, $d_{\text{rec}3\times3}$ , NN-Regel	2.3
+ Beste 500 $D_{\text{euk}}$	2.2
+ 3-NN-Regel	2.1

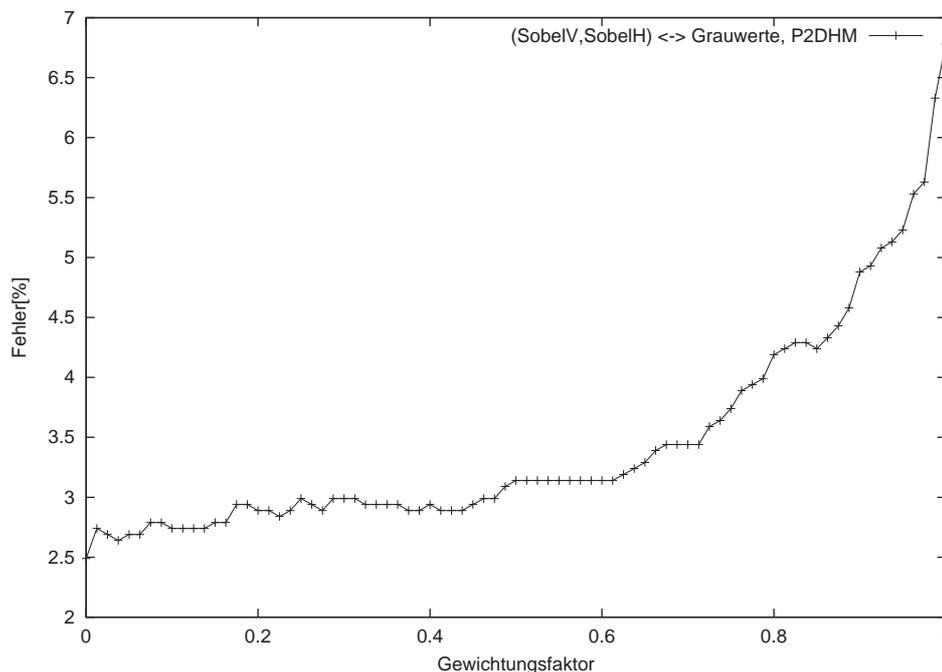


Abbildung 6.4: Fehlerraten der USPS-Datensammlung durch das P2DHM-Verfahren abhängig von der Gewichtung der Gradienten- gegenüber den Grauwerten

Die Fehlerrate wird durch Verwendung der Gradientenbilder deutlich verringert, ohne dass die Verformungsmodelle durch Strafkostenfunktionen angepasst werden. Durch die Verwendung von Gradientenbildern, die durch den Sobelfilter erzeugt wurden, und des P2DHM-Verfahrens wird eine Fehlerrate von 2.4% erzielt. Es wurde untersucht, ob die Hinzunahme der Grauwerte zu diesen Gradientenbildern eine weitere Verbesserung der Fehlerrate mit sich bringt. Hierfür wurden die Grauwerte und die Gradientenwerte unterschiedlich gewichtet. Abbildung 6.4 stellt die Ergebnisse dieses Versuches graphisch dar. Der Gewichtungsfaktor von 0 entspricht der alleinigen Verwendung der Gradientenbilder und der Faktor 1 entspricht dem anderen Extrem, der alleinigen Verwendung der Grauwerte. Es konnte keine Verbesserung durch die Hinzunahme der Grauwerte erzielt werden. Durch die Gradientenbilder und die dadurch differenzierten Pixeldistanzen ist die Distanz zwischen Bildern, die nach einem Verformungsmodell angepasst wurden, aussagekräftiger.

Die durch den Sobelfilter erzeugten Gradientenbilder sind durch die Filteroperation visuell unscharf. Genauere Gradientenbilder werden durch die Verwendung von Splinekoeffizienten erzeugt [Jähne 02, S. 344]. Diese erzielen unerwartet schlechte Fehlerraten, können aber durch Filteroperationen geglättet werden. Werden diese geglätteten Gradientenbilder verwendet, sind die Ergebnisse vergleichbar mit denen, die auf den durch den Sobelfilter erzeugten Gradientenbildern beruhen. Tabelle 6.2 stellt die erzielten Fehlerraten unter Verwendung der Gradientenbilder dar.

Die Ergebnisse, die unter Verwendung der Gradientenbilder erzielt wurden, moti-

Tabelle 6.3: Beamsearch-Ergebnisse des WW2DHM-Verfahrens für die USPS-Datensammlung

Threshold	Durchschnitt der aktiven		Fehler[%]
	Warped Wakes	Referenzen	
0.1	119	2	7.3
0.2	159	4	5.1
0.5	247	12	3.9
0.8	303	20	3.7

Beste 50  $D_{\text{euk}}$ , Rahmen 3, SobelV, SobelH,  $R_{\text{pen}}$ , NN-Regel

vieren Experimente mit Bildausschnitten der Graubilder als Pixelwerten zur Bestimmung der Pixeldistanz, da diese die lokale Information beinhalten, die zum Teil in den Pixelwerten der verwendeten Gradientenbilder enthalten sind. Die Pixelwerte der Bildausschnitte werden einzeln gewichtet. Beispielsweise kann dies unter Berücksichtigung der in einer Trainingsphase beobachteten Varianzen geschehen.

Durch die Anwendung ungewichteter Pixeldistanzen der Grauwerte können mit dem relativ einfachen ID-Verfahren gute Ergebnisse erzielt werden. Durch die zusätzliche Verwendung der Gradientenbilder und der NN-Regel wird mit dem ID-Distanzverfahren die Fehlerrate von 2.4% erzielt.

Das WW2DHM-Verfahren hat im Vergleich zu anderen Distanzverfahren eine relativ hohe Rechenzeit. Diese wird durch die Verwendung von Beamsearch mit den Parametern Beamsize und Threshold begrenzt. Die Berechnung der distanzminimierenden Verformung ist dadurch nicht mehr garantiert.

Tabelle 6.3 stellt die resultierenden Fehlerraten für verschiedene Einstellungen der Beamsearch-Methode dar. Durch die Begrenzung der betrachteten Warped Wakes pro Pixelschritt werden die lokal schlechten Warped Wakes verworfen. Dabei können absolut beste Warped Wakes verloren gehen.

Die Anzahl der Menge von Warped Wakes, die in einem Pixelschritt bei voller Suche betrachtet werden müssen, ist exponentiell abhängig von der Bildhöhe.

Abbildung 6.5 stellt Verformungen des WW2DHM-Verfahrens bei Anwendung eines generierten Test- und Trainingsbildes dar. Bei diesen Bildern müssen die gekreuzten weißen Linien aufeinander abgebildet werden, damit die Distanz minimiert wird. Die Bilder haben eine Größe von  $16 \times 16$  Pixel. Die gekreuzten Linien liegen um 2 Pixelpositionen nach rechts verschoben und das gesamte Testbild ist gegenüber dem Referenzbild um eine Pixelposition nach unten verschoben. Die obere der beiden dargestellten Verformungen wird durch einen Beamsize von  $B=5\,000$  gewonnen und das untere Verformungsergebnis wird mit einem Beamsize von  $B=10\,000$  erzielt. Zwischen diesen beiden Einstellungen liegt der Schwellwert des Verfahrens, der eine optimale Verformung dieser Bilder ermöglicht. Bei Bildern der gleiche Größe ist dieser Schwellwert von der Kom-

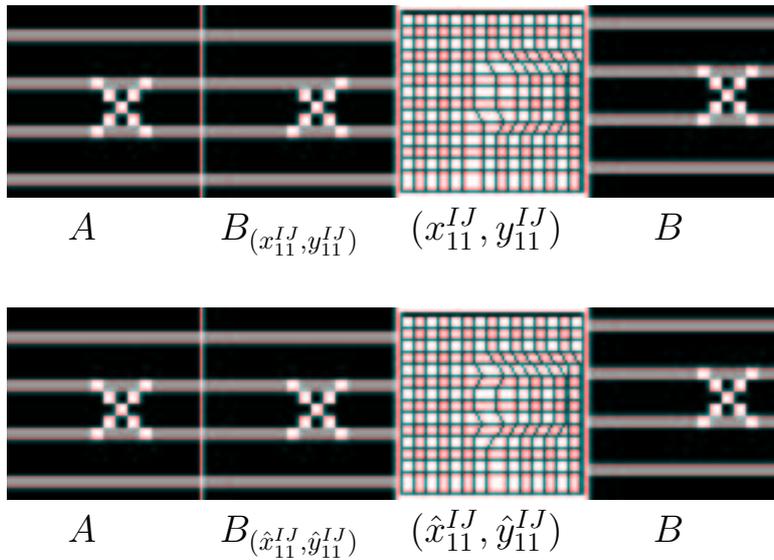


Abbildung 6.5: Beispiel-Verformungen durch WW2DHMM

plexität der gesuchten distanzreduzierenden Verformung abhängig. Interessant könnten Versuche mit Distanzverfahren sein, die auf anderen Näherungsverfahren des 2DHM-Modells basieren, wie dem Turbo-2DHM-Verfahren [Perronnin & Dugelay<sup>+</sup> 03].

Mit dem SA2DHM-Verfahren wurden wenige Experimente durchgeführt. Dieses Verfahren hat den Nachteil, dass ein Distanzwert zweier Bilder nicht eindeutig von den gewählten Iterationsschritten  $T$  abhängig ist, sondern aufgrund der zufällig bestimmten Verformung um einen Wert schwankt. Somit können diesem Verfahren keine eindeutigen Fehlerraten angegeben werden. Die schlechteste Fehlerrate, die in Experimenten mit dem SA2DHM-Verfahren und  $T=10\,000$  Iterationsschritten erzielt wurde, beträgt 3.0%. Die durch dieses Verfahren gefundene Bildverformung ist bei gleicher Rechenzeit (siehe Abbildung 5.1) meistens kostenminimierender, als die durch das WW2DHM-Verfahren bestimmte Verformung. Interessant könnte eine Erweiterung dieses Verfahrens sein, das die Verschiebung von Bildblöcken statistisch abhängig von den letzten erfolgreichen Verformungen durchführt und somit konstant einen Distanzwert für denselben Bildvergleich berechnet.

In dieser Diplomarbeit wurden verschiedene Experimente mit erlernten Prototypen durchgeführt. Abbildung 6.6 stellt jeweils ein Durchschnittsbild einer Klasse der USPS-Trainingsbilder dar. Abbildung 6.7 stellt die nach dem IDM erlernten Prototypen dar. Diese wurden durch die Verwendung der Durchschnittsbilder erzeugt, indem ein Durchschnittsbild zu den der Klasse nach zugehörigen Trainingsbildern angepasst wurde. Dann wurde die Summe, der nach den Warpingabbildungen zugehörigen Pixelwerte der Trainingsbilder, durch die Anzahl der im Durchschnittsbild verwendeten Positionen gemittelt. Dieser Ablauf kann mit den erstellten Prototypen wiederholt werden, bis diese sich nicht mehr verändern oder eine maximale Anzahl von Durchläufen erreicht wird. In Abbildung 6.8 werden die nach dem IDM erlernten Prototypen der Gradientenbilder


 A row of ten grayscale images showing the average appearance of handwritten digits 1 through 0 from USPS training data. The digits are somewhat blurry and have a soft, averaged appearance.

Abbildung 6.6: Durchschnittsbilder der USPS-Trainingsdaten


 A row of ten black and white binary images representing prototypes for digits 1 through 0, determined by the IDM method. The digits are sharp and high-contrast.

Abbildung 6.7: Beispiel für nach dem IDM bestimmte USPS-Prototypen

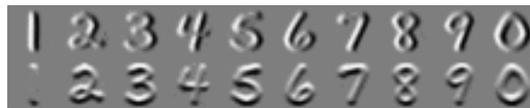

 A 2x10 grid of grayscale images. The top row shows the average images of digits 1-0, and the bottom row shows the corresponding prototypes determined by IDM from gradient images.

Abbildung 6.8: Beispiel für nach dem IDM bestimmte USPS-Prototypen von Gradientenbildern

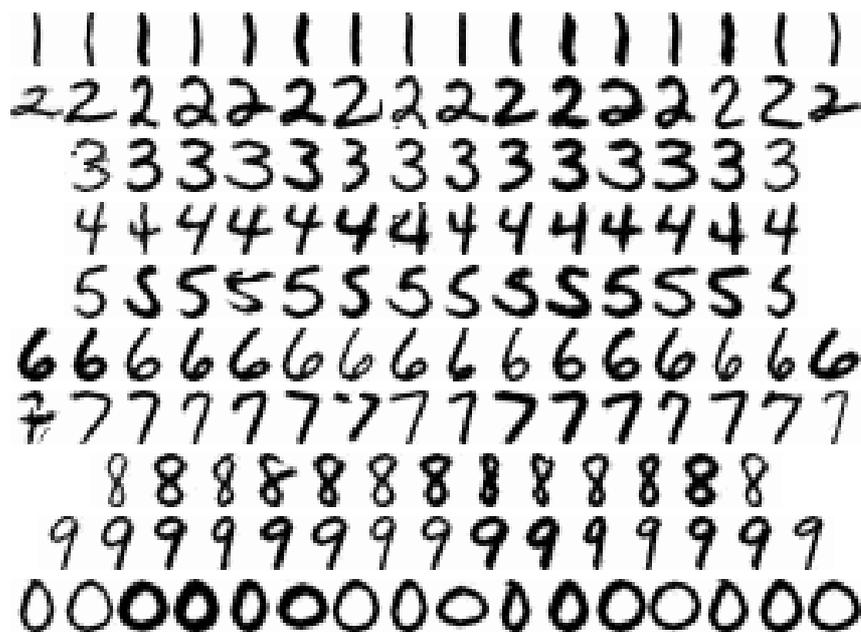

 A 10x10 grid of images. Each row contains ten examples of a single digit (1 through 0). The digits are arranged in a grid, showing the results of the EM-Cluster algorithm. The digits are sharp and high-contrast, representing prototypes.

Abbildung 6.9: Bildresultate des EM-Cluster-Algorithmus durch Verwendung des IDM

dargestellt. Die nach diesem Verfahren erstellten Prototypen wurden als Mittelwerte für das Cluster-Verfahren durch Mischverteilungen verwendet [Güld 00]. Abbildung 6.9 stellt eine erzielte Menge von Prototypen dar. Diese verschiedenen Prototypen wurden für Experimente verwendet, deren Ergebnisse in Tabelle 6.5 dargestellt sind.

Tabelle 6.5: USPS-Ergebnisse unter Verwendung von Prototypen

Verfahren	Fehler[%]			
	Grauwerte			Gradienten
	Durchschnitt	1-Prototyp	EM-Prototypen	1-Prototyp
eukl. Distanz, NN-Regel	18.6	26.1	12.8	30.6
ID, NN-Regel	25.8	7.6	4.5	7.3
HM, NN-Regel	27.0	6.8	4.6	7.2
HM, $d_{\text{rec}3 \times 3}$ , NN-Regel	21.9	7.6	4.8	7.6
HMD, NN-Regel	37.4	7.7	3.9	6.8
HMD, $d_{\text{rec}3 \times 3}$ , NN-Regel	25.3	4.9	3.1	4.4

## 6.2 UCI-Datensammlung

Die University of California, Irvine Datensammlung (UCI-Datensammlung) handgeschriebener arabischer Ziffern beinhaltet Bilder, die jeweils eine von 10 Ziffern darstellen und einer von 10 Klassen zugeordnet sind. Die Bilder haben eine Größe von  $8 \times 8$  Pixel und ein Pixelwert steht für einen von 16 Grauwerten. Der Korpus, der frei erhältlich ist <sup>2</sup>, wird in 1 797 Test- und 3 823 Trainingsbilder unterteilt.

### 6.2.1 Stand der Technik

Ein Klassifikationssystem, das auf der euklidischen Distanz und der NN-Regel beruht, erzielt eine Fehlerrate von 2.0%. In einer aktuellen Veröffentlichung wird eine Fehlerrate von 1.5% durch die Anwendung eines PCA-Mixture Modells angegeben [Kim & Kim<sup>+</sup> 02]. Die niedrigste Fehlerrate, die in Experimenten mit den vorgestellten nichtlinearen Verformungsmodellen erzielt wurde, beträgt 0.8%.

### 6.2.2 Experimente und Ergebnisse

Die Experimente mit der UCI-Datensammlung handgeschriebener Ziffern orientierten sich an den Erfahrungen, die mit der USPS-Datensammlung gemacht wurden. Es zeigte sich, dass die Ergebnisse von der Auflösung der Bilder abhängig sind. Der Graph in Abbildung 6.12 stellt die erzielten Fehlerraten des P2DHM-Verfahrens abhängig von der Auflösung der Bilder dar. Die starke Abhängigkeit der Fehlerrate von der Bildauflösung verdeutlicht, dass die Parameter der beschriebenen Verfahren an diese angepasst werden müssen.

Für die Interpolation der Pixelzwischenwerte zur Erhöhung der Auflösung wurde das lineare Interpolationsverfahren und die Interpolation durch Splines verwendet. Diese Berechnung von Zwischenwerten führte zu einer deutlichen Verbesserung der Ergebnisse, wobei sich die Interpolation durch Splines, im Vergleich zur linearen Interpolation, besonders bewährte. Abbildung 6.11 stellt ein Beispielbild in unterschiedlichen Auflösungen, die anhand dieser beiden Verfahren berechnet wurden, dar. Tabelle 6.7 beschreibt die erzielten Fehlerraten mit den verwendeten Einstellungen des Klassifikationssystems.

Die Fehlerrate von 0.8% wird durch das ID-Verfahren erzielt, wenn die Auflösung, der UCI-Bilder nach Anwendung einer Interpolation durch Splines, verdoppelt wird.

---

<sup>2</sup><ftp://ftp.ics.uci.edu/pub/machine-learning-databases/optdigits>

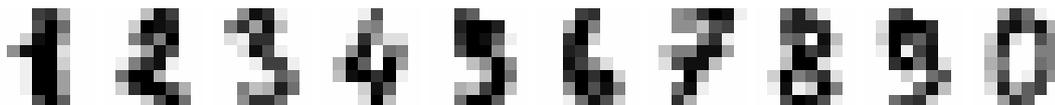


Abbildung 6.10: Je ein Beispielbild für jede Klasse der UCI-Datensammlung

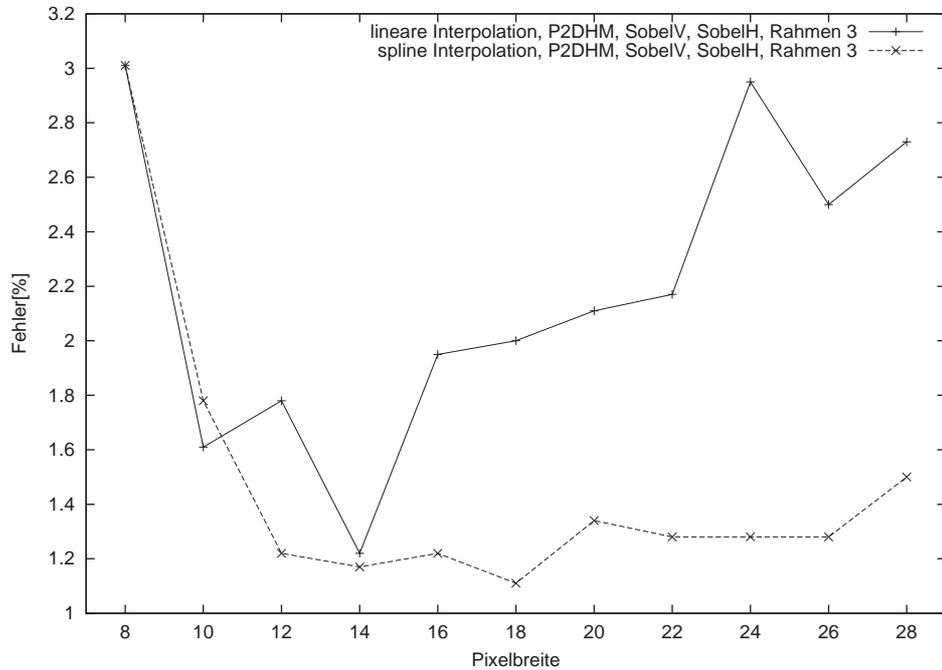


Abbildung 6.11: UCI-Ergebnisse mit unterschiedlichen Auflösungen für das P2DHM-Verfahren

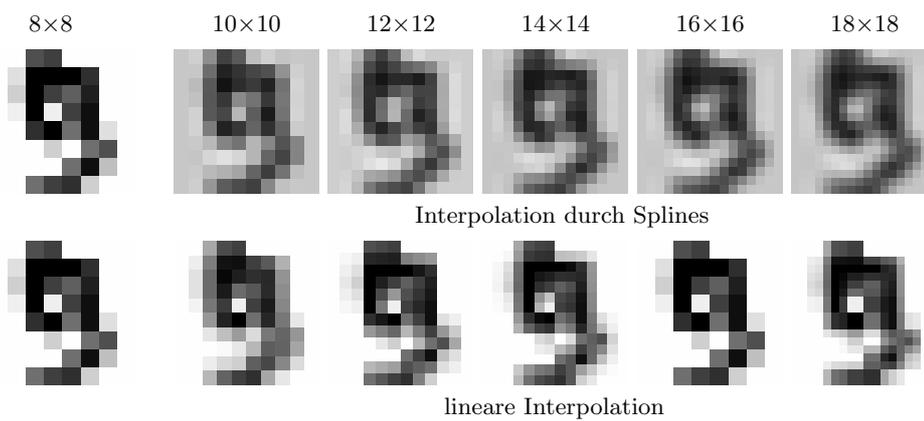


Abbildung 6.12: Beispiel für Ergebnisse von Interpolationsverfahren

Tabelle 6.7: Ergebnisse für die UCI-Datensammlung

Verfahren	Fehler[%]
PCA-Mixture Modells [Kim & Kim <sup>+</sup> 02]	1.5
euklidische Distanz, NN-Regel diese Arbeit	
+ Grauwerte	2.0
+ Rahmen 3, Shift 3×3	1.8
+ SobelV, SobelH, Rahmen 3	1.8
+ Shift 3×3	1.4
P2DHM, Beste 100 $D_{\text{euk}}$ , NN-Regel	
+ Rahmen 3, Grauwerte	8.0
+ $d_{\text{rec}3\times3}$	1.6
+ $d_{\text{bin}3\times3}$	2.0
+ Rahmen 3, SobelV, SobelH	3.0
+ $d_{\text{rec}3\times3}$	1.7
+ $d_{\text{bin}3\times3}$	1.6
ID, Beste 100 $D_{\text{euk}}$ , NN-Regel	
+ Rahmen 3, Grauwerte, $W=1$	13.2
+ $d_{\text{rec}3\times3}$ , $W=1$	1.5
+ $d_{\text{bin}3\times3}$ , $W=1$	1.5
+ Rahmen 3, SobelV, SobelH, $W=1$	2.1
+ $d_{\text{rec}3\times3}$ , $W=1$	1.3
+ $d_{\text{bin}3\times3}$ , $W=1$	1.5
Auflösung 16×16 durch Splines interpoliert	
+ Rahmen 3, SobelV, SobelH, NN-Regel	
+ Beste 100 $D_{\text{euk}}$	
+ P2DHM	1.2
+ $d_{\text{rec}3\times3}$	1.1
+ P2DHMD	1.1
+ ID, $d_{\text{rec}3\times3}$ , $W=2$	1.0
+ ID, $d_{\text{rec}3\times3}$ , $W=2$	0.8
+ P2DHMD, Beste 100 $D_{\text{euk}}$ , $d_{\text{rec}3\times3}$	0.8

### 6.3 CEDAR-Datensammlung

Die Center of Excellence for Document Analysis and Recognition Datensammlung (CEDAR-Datensammlung) handgeschriebener arabischer Ziffern beinhaltet Bilder, die jeweils eine von 10 Ziffern darstellen und einer von 10 Klassen zugeordnet sind. Der CEDAR-Korpus wurde modifiziert [Hinton & Dayan<sup>+</sup> 97]. Diese Modified CEDAR-Datensammlung (MCEDAR-Datensammlung) beinhaltet 13 711 Bilder mit der Größe von  $8 \times 8$  Pixel. Diese Bilder sind in 11 000 Trainingsbilder und 2 711 Testbilder unterteilt. Abbildung 6.13 stellt jeweils ein Bild pro Klasse des MCEDAR-Korpus dar.

#### 6.3.1 Stand der Technik

Tabelle 6.8 stellt die Fehlerraten unterschiedlicher Klassifikationssysteme unter Verwendung der MCEDAR-Datensammlung dar. Die niedrigste Fehlerrate, die von einer externen Forschungseinrichtung bekannt ist, beträgt 4.6%.

#### 6.3.2 Experimente und Ergebnisse

Nach den Erfahrungen durch Experimente mit unterschiedlicher Bildauflösungen am UCI-Korpus wurde die Auflösung der Bilder dieses Korpus verdoppelt. Die durchgeführten Versuche und die resultierenden Ergebnisse sind in Tabelle 6.9 aufgeführt. Die Ergebnisse bestätigen die bei der UCI-Datensammlung gemachten Erfahrungen, dass die Fehlerrate und damit die Güte der vorgestellten Distanzverfahren, abhängig von der Detailauflösung eines Bildes im Verhältnis zur Größe des für die Pixeldistanzberechnung verwendeten Bildausschnitts ist.

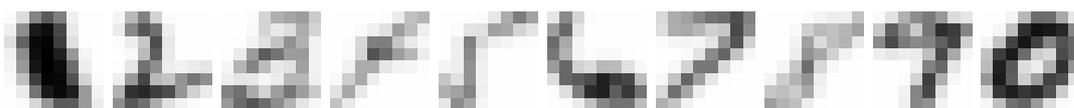


Abbildung 6.13: Je ein Beispielbild für jede Klasse der MCEDAR-Datensammlung

Tabelle 6.8: Ergebnisse der MCEDAR-Datensammlung

Verfahren		Fehler[%]
PCA	[Hinton & Dayan <sup>+</sup> 97]	4.9
Factor analysis	[Hinton & Dayan <sup>+</sup> 97]	4.7
Bayes'sches PCA Modell	[Bishop & Winn 00]	4.8
Probabilistic PCA	[Tipping & Bishop 99]	4.6
lokale Merkmale	RWTH i6	4.3
nichtlineare Verformungsmodelle	diese Arbeit	3.6

Tabelle 6.9: MCEDAR-Ergebnisse der Verformungsverfahren

Verfahren	Fehler[%]
euklidische Distanz, NN-Regel	
+ Grauwerte	5.7
+ Rahmen 3, Shift $3 \times 3$	5.4
+ Rahmen 3, SobelV, SobelH	5.9
+ Shift $3 \times 3$	5.5
Auflösung $16 \times 16$ durch Splines interpoliert	
+ Rahmen 3, SobelV, SobelH, NN-Regel	
+ Beste 100 $D_{\text{euk}}$	
+ P2DHM	4.0
+ ID, $d_{\text{rec}3 \times 3}$ , $W = 2$	3.9
+ P2DHMD	3.8
+ ID, $d_{\text{rec}3 \times 3}$ , $W = 2$	3.6

## 6.4 MNIST-Datensammlung

Die National Institute of Standards and Technology Datensammlung (NIST-Datensammlung) handgeschriebener arabischer Ziffern ist der Ursprung für die Modified NIST-Datensammlung (MNIST-Datensammlung). Ein Auszug der Test- und Trainingsbilder wurde neu in Test- und Trainingsdaten unterteilt. Die MNIST-Datensammlung beinhaltet Bilder, die eine von 10 Ziffern darstellen und die einer von 10 Klassen zugeordnet sind. Die Bilder sind auf eine Größe von  $20 \times 20$  Pixel segmentiert. Durch das gewählte Verfahren zur Normalisierung der Bildgröße nehmen die ursprünglich binären Pixelwerte einen Grauwert an. Die Bilder wurden durch einen Rahmen auf  $28 \times 28$  Pixel vergrößert und jedes Bild wurde so verschoben, dass der Massepunkt der Pixel in der Mitte des Bildes liegt [LeCun & Bottou<sup>+</sup> 98]. Die MNIST-Datensammlung besteht aus 60 000 Trainingsbildern und 10 000 Testbildern.

Die Trainingsdaten für NIST wurden von ca. 500 Büroangestellten gewonnen, wohingegen die Testbilder von Schülern stammen. Die Test- und Trainingsdaten sind unterschiedlich schwierig zu klassifizierende Daten. Dadurch wird das Training für ein Klassifikationssystem erschwert. Die MNIST-Datensammlung wurde konstruiert, um dieses Problem zu beheben und ist frei erhältlich<sup>3</sup>.

### 6.4.1 Stand der Technik

Ein Klassifikationssystem, das auf der euklidischen Distanz und der NN-Regel beruht, erzielt auf den weiterverarbeiteten  $28 \times 28$  Pixel großen Bildern eine Fehlerrate von 3.1%. Eines der besten Ergebnisse ist die Fehlerrate von 0.6%, die durch ein Verfahren, welches die Umrisse der Ziffern verwendet, erzielt wird. Dieses Verfahren verwendet die  $k$ -NN-Regel mit den  $k=3$  nächsten Nachbarn und klassifiziert so 63 Bilder falsch [Belongie & Malik<sup>+</sup> 02]. Das in dieser Arbeit präsentierte Klassifikationssystem macht unter Verwendung der 3-NN-Regel 58 Fehler. In der Tabelle sind weitere Ergebnisse unterschiedlicher Klassifikationsverfahren dargestellt.

### 6.4.2 Experimente und Ergebnisse

Tabelle 6.11 stellt die Fehlerraten, die in den durchgeführten Versuchen erzielt wurden, dar. Obwohl die Größen der MNIST-Bilder sich von den Bildgrößen der vorher verwendeten Datensammlungen unterscheiden, konnten mit denselben Parametern, für die verwendeten Distanzverfahren, diese guten Ergebnisse erzielt werden.

<sup>3</sup><http://yann.lecun.com/exdb/mnist>



Abbildung 6.14: Je ein Beispielbild für jede Klasse der MNIST-Datensammlung

Tabelle 6.10: Ergebnisse für die MNIST-Datensammlung

Verfahren		Fehler[%]
menschliche Fehlerrate	[Simard & LeCun <sup>+</sup> 93]	0.2
Vorverarbeitung, eukl. Distanz, $k$ -NN-Regel	[LeCun & Bottou <sup>+</sup> 98]	2.4
Neuronales Netz (LeNet1) $16 \times 16$	[LeCun & Bottou <sup>+</sup> 98]	1.7
mod. quadr. Diskriminantenfunktion	[Dong & Krzyzak <sup>+</sup> 99]	1.5
Tangentendistanz	[Simard & LeCun <sup>+</sup> 93]	1.1
erweiterte Tangentendistanz	[Keysers & Dahmen <sup>+</sup> 00]	1.0
invariante Support-Vektoren	[Schölkopf & Simard <sup>+</sup> 98]	0.8
Boosting, Neuronales Netz (LeNet4)	[LeCun & Bottou <sup>+</sup> 98]	0.7
Shape Matching, 3-NN-Regel	[Belongie & Malik <sup>+</sup> 02]	0.6
nichtlineare Verformungsmodelle	diese Arbeit	0.6

Tabelle 6.11: MNIST-Ergebnisse der Verformungsverfahren

Verfahren	Fehler[%]	
	$20 \times 20$	$28 \times 28$
euklidische Distanz, NN-Regel	3.5	3.1
Beste 100 $D_{\text{euk}}$ , NN-Regel		
+ ID, $d_{\text{rec}3 \times 3}$ , $W = 2/3$	1.5	1.2
+ SobelV, SobelH, $W=2$	0.9	0.9
ID, Beste 500 $D_{\text{euk}}$ , SobelV, SobelH, $d_{\text{rec}3 \times 3}$ , $W=2$	0.8	0.7
+ 3-NN-Regel	0.7	0.6
ID, Beste 1000 $D_{\text{euk}}$ , SobelV, SobelH, $d_{\text{rec}3 \times 3}$ , $W=2$	–	0.7
+ 3-NN-Regel	–	0.6

## 6.5 Siemens-Datensammlung

Die Siemens-Daten wurde dem Lehrstuhl für Informatik VI der RWTH von der Siemens Dematic AG zur Verfügung gestellt. Es handelt sich um zwei Datensammlungen. Die SiOI-Datensammlung besteht aus Bildern, die jeweils eine von zehn arabischen handgeschriebenen Ziffern darstellen. Die Bilder werden in 13 Klassen unterteilt, da Bilder, die die Ziffern „1“, „7“ und „0“ darstellen, jeweils in zwei Klassen unterteilt werden, die für unterschiedliche Schreibformen der Ziffern stehen. Die Bilder sind nicht quadratisch und unterschiedlich groß. Das Bild mit der geringsten Dimension aus den Daten hat eine Größe von  $3 \times 11$  Pixel. Die höchste Bilddimension in diesem Korpus besitzt ein Bild der Größe von  $110 \times 127$  Pixel. Die Pixelwerte der Bilder sind binär. Die Daten werden in 26 000 Trainings- und 6 500 Testbilder unterteilt. Die SiFF-Datensammlung ist aus der SiOI-Datensammlung entstanden. Die Bilder wurden weiterverarbeitet, segmentiert und normalisiert. Diese Bilder sind quadratisch und haben eine Bildgröße von  $16 \times 16$  Pixel. Es werden 256 Pixelwerte, die Grauwerte repräsentieren, unterschieden. Die Abbildungen 6.15 und 6.16 stellen zugehörige Bilder aus der SiOI- und der SiFF-Datensammlung dar.

### 6.5.1 Stand der Technik

Für diese Datensammlungen gibt es keine veröffentlichten Fehlerraten. Tabelle 6.12 stellt neben den durch das vorgestellte Klassifikationsystem erzielten Ergebnisse auch Resultate durch die Verwendung der Tangentendistanz dar. Die niedrigste Fehlerrate, die beim 13-Klassen-Problem erzielt werden konnte, beträgt 1.2% und die zugehörige Fehlerrate bei einer Reduzierung auf 10 Klassen, eine Klasse pro Ziffer, beträgt 1.0%.

### 6.5.2 Experimente und Ergebnisse

Tabelle 6.12 stellt die Ergebnisse der durchgeführten Experimente dar. Die angegebenen Fehlerraten sind für das 13-Klassen-Problem berechnet worden.



Abbildung 6.15: Je ein Beispielfeld für jede Klasse der SiOI-Datensammlung



Abbildung 6.16: Je ein Beispielfeld für jede Klasse der SiFF-Datensammlung

Tabelle 6.12: SiOI- und SiFF-Ergebnisse der Verformungsverfahren

Verfahren	Fehler[%]	
	SiOI	SiFF
euklidische Distanz, NN-Regel	11.2	2.5
Tangentendistanz RWTH i6	6.6	1.8
P2DHM, Rahmen 3, Beste 100 $D_{\text{euk}}$ , NN-Regel	7.7	4.1
+ $R_{\text{pen}^2}$	4.8	2.3
+ SobelV, SobelH	2.7	1.5
+ $d_{\text{rec}3 \times 3}$	–	1.6
ID, Rahmen 3, Beste 100 $D_{\text{euk}}$ , NN-Regel, $W=1$	–	4.5
+ $d_{\text{rec}3 \times 3}$ , $W=1$	–	1.4
+ SobelV, SobelH, $W=2$	–	1.7
+ $d_{\text{rec}3 \times 3}$ , $W=2$	–	1.3
ID, Rahmen 3, Beste 500 $D_{\text{euk}}$ , NN-Regel, $W=2$	–	1.2

## 6.6 ETL6A-Datensammlung

Die Electrotechnical Laboratory Datensammlung (ETL-Datensammlung) von Zeichen wurde vom National Institute of Advanced Industrial Science and Technology, Japan erstellt. Die Datensammlung beinhaltet ca. 1.2 Millionen Bilder von handgeschriebenen und maschinell erstellten Zeichen. Es sind japanische, chinesische, lateinischen Zeichen und arabische Ziffern. Die ETL-Datensammlung, die zu Forschungszwecken frei erhältlich ist<sup>4</sup>, ist in weitere Datensammlungen unterteilt. Eine dieser Datensammlungen ist der ETL6-Korpus.

Die erfassten Zeichen dieses Korpus werden in Abbildung 6.17 dargestellt. Der ETL6A-Korpus ist eine weitere Teilmenge und beinhaltet ausschließlich die lateinischen Zeichen. Dieser Korpus besteht aus 35 958 Bildern, die eines von 26 groß geschriebenen lateinischen Zeichen darstellen und einer von 26 Klassen angehören. Diese Bilder haben eine Größe von 64×63 Pixel und die 16 Pixelwerte repräsentieren Grauwerte. Nach den in [Uchida & Sakoe 03b] gemachten Angaben werden die ersten 600 Bilder einer Klasse des Korpus als Trainingsdaten verwendet und die weiteren 500 Bilder einer Klasse als Testdaten. Somit besteht die für die Experimente verwendete ETL6A-Datensammlung aus 13 000 Test- und 15 600 Trainingsbildern.

<sup>4</sup><http://www.is.aist.go.jp/etl6db>



Abbildung 6.17: Je ein Beispielbild für jede Klasse der ETL6-Datensammlung

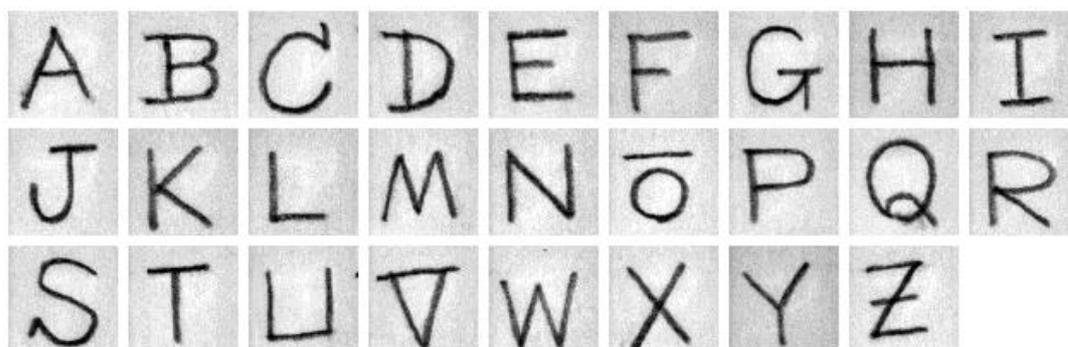


Abbildung 6.18: Je ein Beispielbild für jede Klasse der ETL6A-Datensammlung

Tabelle 6.14: Ergebnisse für die ETL6A-Datensammlung

Verfahren	Fehler[%]
Vorverarbeitung, eukl. Distanz, NN-Regel [Uchida & Sakoe 03b]	1.9
abschnittsweise lineares WW2DHM	0.9
Eigen-Deformationen	0.8
ID, $W=2$ [Uchida & Sakoe 02]	0.9
ID, $W=5$ , Eigen-Deformationen	0.6
Eigen-Deformationen [Uchida & Sakoe 03a]	0.5
eukl. Distanz, NN-Regel diese Arbeit	4.5
Beste 100 $D_{\text{euk}}$ , SobelV, SobelH, NN-Regel	
+ P2DHM	1.2
+ P2DHMD, $d_{\text{rec}3 \times 3}$	0.8
+ ID, $d_{\text{rec}3 \times 3}$ , $W=2$	0.8
+ Beste 1000 $D_{\text{euk}}$ , $W=2$	0.6
+ 3-NN-Regel	0.5

Bilder linear auf  $16 \times 16$  Pixel skaliert

### 6.6.1 Stand der Technik

Für diese ETL6A-Datensammlung sind externe Vergleichsergebnisse anderer Distanzverformungsverfahren bekannt [Uchida & Sakoe 02, Uchida & Sakoe 03b]. Diese Ergebnisse motivierten die Versuche mit dieser Datensammlung, um die unterschiedlichen Klassifikationssysteme, die auf Verformungsverfahren beruhen, zu vergleichen. Das externe Klassifikationssystem verwendet die Trainingsdaten zur Erzeugung der für die Klassifikation verwendeten Referenzdaten mit gelernten Verformungseinschränkungen und erzielt so die Fehlerrate von 0.5%. Diese Fehlerrate konnte mit dem in dieser Arbeit präsentierten Klassifikationssystem ebenfalls erzielt werden.

### 6.6.2 Experimente und Ergebnisse

Die Bilder des ETL6A-Korpus wurden mit dem linearen Interpolationsverfahren auf  $16 \times 16$  Pixel große Bilder skaliert. Das Ergebnis nach [Uchida & Sakoe 02] wurde ebenso mit kleiner skalierten und zusätzlich weiter verarbeiteten Bildern erzielt. In [Uchida & Sakoe 03b] wird auf [Mori & Yamamoto 84] referenziert, für die Beschreibung zur Gewinnung der verwendeten 4-dimensionalen Pixelmerkmale, die eine Richtung angeben.

Es wurde in den Versuchen durch die euklidische Distanz eine Vorauswahl der Trainingsbilder pro Testbild getroffen und die so bestimmten Referenzbilder für die eigentlichen Distanzberechnungen verwendet. Tabelle 6.14 stellt die erzielten Ergebnisse unterschiedlicher Verformungsverfahren dar.

## 6.7 IRMA-Datensammlung

Die Image Retrieval in Medical Applications Datensammlung (IRMA-Datensammlung) besteht aus digitalisierten Röntgenbildern. Die Bilder stammen aus einem Projekt der RWTH Aachen. Das Projektziel ist eine inhaltsbasierte Suche für medizinische Daten. Die IRMA-Datensammlung wird in die sechs Klassen Abdomen, Gliedmaßen, weibliche Brust, Schädel, Thorax und Wirbelsäule unterteilt. Abbildung 6.19 zeigt Beispielbilder des Korpus. In Abbildung 6.20 sind 6 Beispielbilder der Klasse „Thorax“ dargestellt, die die Schwierigkeit der Klassifikationsaufgabe verdeutlichen sollen. Die Daten entstanden aus Röntgenbildern, die nachträglich digitalisiert wurden. Das Bild mit der geringsten Dimension aus den Daten hat eine Größe von  $142 \times 233$  Pixel. Die höchste Bilddimension in diesem Korpus hat ein Bild der Größe von  $4928 \times 4008$  Pixel. Die 256 Pixelwerte der Bilder repräsentieren die Grauwerte. Die Datensammlung besteht aus 1617 Bildern, welche nicht in Test- und Trainingsbilder unterschieden werden. Die Ergebnisse werden durch das „Leaving One Out“-Verfahren erzielt.

### 6.7.1 Stand der Technik

Die Klassifikationsaufgabe der IRMA-Datensammlung ist zur Zeit ein schwieriges Problem. Dies verdeutlichen die in Tabelle 6.15 dargestellten Klassifikationsergebnisse. Das beste Ergebnis, die Fehlerrate von 8.0%, wird am Lehrstuhl für Informatik VI der RWTH erzielt. Durch die Verwendung des P2DHMD-Modells zur Gestaltung der vorgestellten Distanzfunktion wird die Fehlerrate von 5.9% erzielt.



Abbildung 6.19: Beispielbild für jede Klasse der IRMA-Datensammlung

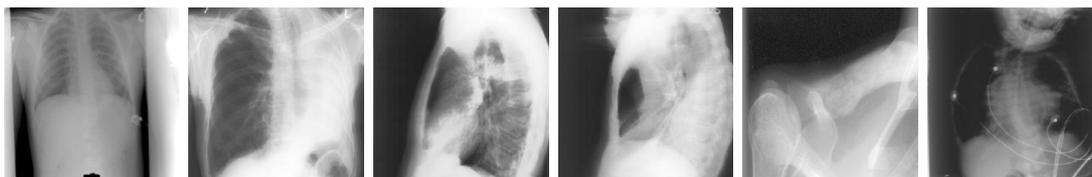


Abbildung 6.20: IRMA-Beispielbilder der Klasse „Thorax“

### 6.7.2 Experimente und Ergebnisse

Die Bilder der IRMA-Datensammlung wurden alle auf eine Bildhöhe von 32 Pixel skaliert, wobei das Seitenverhältnis der Bilder erhalten wurde. Desweiteren wurde jedes Bild auf den maximalen Grauwertbereich normiert. Tabelle 6.16 stellt die erzielten Fehlerraten der Verformungsverfahren dar. Durch die Anwendung des Threshold-Parameters für die Pixeldistanzen konnten die Ergebnisse deutlich verbessert werden [Keysers 00]. Abbildung 6.21 stellt die erzielten Fehlerraten abhängig von dem Threshold-Parameter dar.

Tabelle 6.15: Ergebnisse für die IRMA-Datensammlung

Verfahren	Fehler[%]
Aktive Konturmodelle [Bredno & Brandt <sup>+</sup> 00]	51.1
Cooccurrence Matrizen [Theiner 00]	29.0
Lokale Merkmale, Thresholding [Paredes & Keysers <sup>+</sup> 02]	9.7
RWTH i6: eukl. Distanz, NN-Regel, nicht quadratisch	15.8
Kernel Densities	14.0
+ Thresholding	11.2
+ IDM	9.0
+ Tangentenvektoren	8.0
nichtlineare Verformungsmodelle diese Arbeit	5.9

Tabelle 6.16: IRMA-Ergebnisse der Verformungsverfahren

Verfahren	Fehler[%]
Bildgröße auf $32 \times 32$ Pixel skaliert	
+ euklidische Distanz, NN-Regel	18.3
+ P2DHM, $R_{\text{pen}^2}$ , NN-Regel	13.8
+ SobelV, SobelH	12.0
Bildgröße auf $x \times 32$ Pixel skaliert	
+ P2DHM, $R_{\text{pen}^2}$ , NN Regel	9.2
+ P2DHMD, Beste 100 $D_{\text{euk}}$ , SobelV, SobelH, NN-Regel, Thresholding	5.9

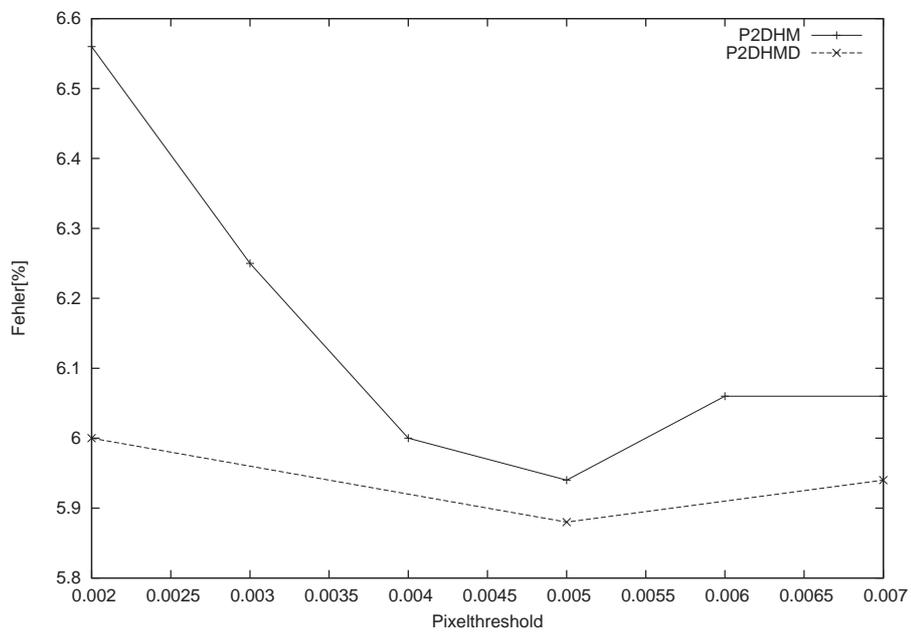


Abbildung 6.21: Fehlerraten der IRMA-Datensammlung durch nichtlineare Verformungsverfahren abhängig von dem Threshold-Parameter

## 6.8 WANG-Datensammlung

Die WANG-Datensammlung ist nach ihrem Ersteller benannt, dieser forscht auf dem Gebiet der automatischen Bildsuche [Li & Wang 03]. Der WANG-Korpus ist frei erhältlich<sup>5</sup> und ist eine Teilmenge der COREL-Datensammlung. Die WANG-Datensammlung besteht aus Fotografien verschiedener Szenen, welche in 10 Klassen unterteilt werden. Abbildung 6.22 stellt jeweils ein Beispielbild einer Klasse dar. Die 1 000 Bilder der Datensammlung haben eine Größe von  $256 \times 384$  oder  $384 \times 256$  Pixel. Die Bilder sind nicht in Trainings- und Testdaten unterteilt.

### 6.8.1 Stand der Technik

In dem Bericht [Wang & Li<sup>+</sup> 01] wird die Datensammlung als Eingabe eines Bildsuchsystems verwendet, welches zu jeweils einem Bild 100 Bilder aus der COREL-Datensammlung herausucht. Die Bewertung der erzielten Resultate unterscheidet sich von einer Bewertung durch die Fehlerrate. Aufgrund der Größe der COREL-Datensammlung (200 000 Bilder) und dem damit verbundenen Zeitaufwand zur Bestimmung vergleichbarer Ergebnisse wurden keine vergleichbaren Experimente durchgeführt. Stattdessen wurden für einige der vorgestellten Distanzverfahren die Fehlerrate

<sup>5</sup><http://wang.ist.psu.edu/>

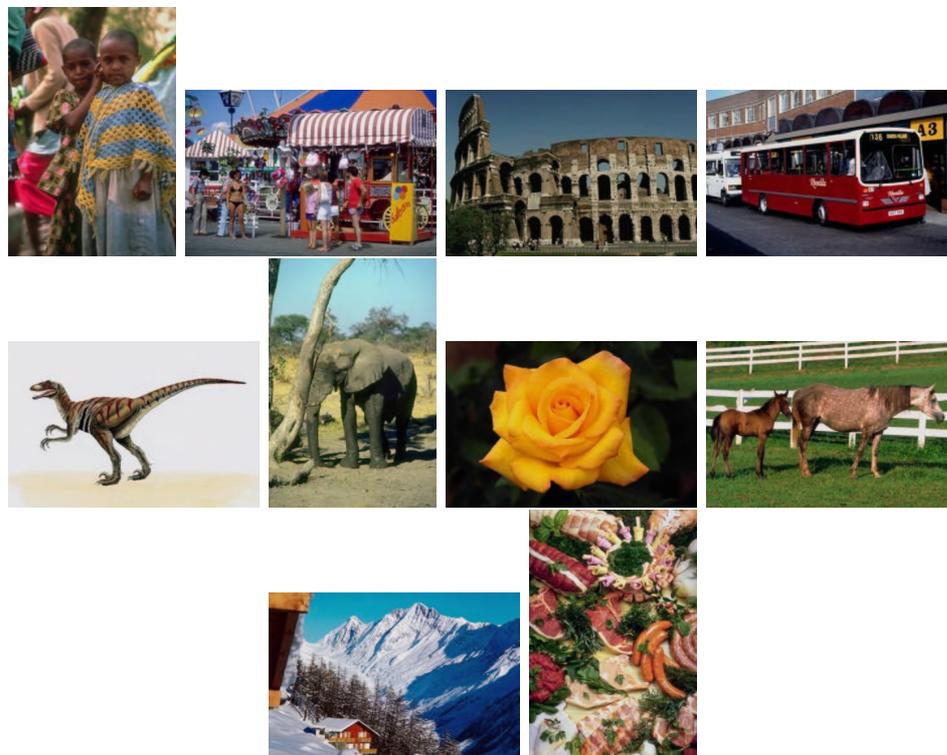


Abbildung 6.22: Je ein Beispielbild für jede Klasse der WANG-Datensammlung

Tabelle 6.17: Ergebnisse für die WANG-Datensammlung

Verfahren	Fehler[%]
verschiedene Bildmerkmale RWTH i6, [Deselaers 03]	21.8
linear auf $64 \times 64$ Pixel skaliert + P2DHM, NN-Regel	26.8
linear auf $32 \times 32$ Pixel skaliert + euklidische Distanz, NN-Regel	50.0
+ ID, $d_{\text{rec}5 \times 5}$ , $W=4$	29.9
+ ID, $d_{\text{rec}3 \times 3}$ , $W=4$	26.2
+ Pixelthreshold	22.1

ten nach der „Leaving One Out“-Methode bestimmt. Werden die RGB-Werte, der auf  $64 \times 64$  Pixel skalierten Bilder, als Merkmale verwendet, erzielt die euklidische Distanz nach der NN-Regel eine Fehlerrate von 50%. Diese Fehlerrate konnte durch Anwendung einer P2DHM-Distanzfunktion im ersten Versuch auf 26.8% gesenkt werden.

### 6.8.2 Experimente und Ergebnisse

Einige Versuche wurden mit  $64 \times 64$  Pixel großen Graubildern anstatt der Farbbilder durchgeführt, die so erzielten Ergebnisse fielen jedoch schlechter aus. Danach wurden ausschließlich die RGB-Werte für weitere Experimente verwendet. In diesen Versuchen wurden Bilder mit einer Größe von  $32 \times 32$  Pixel verwendet. Tabelle 6.17 stellt die gewonnenen Ergebnisse dar.

## Kapitel 7

# Zusammenfassung und Ausblick

In dieser Arbeit wurde eine auf nichtlinearen Verformungen basierende Distanzfunktion vorgestellt und untersucht. Das Ziel war die Modellierung einer Distanzfunktion, die invariant gegenüber klassenerhaltenden Transformationen eines Bildes ist und somit die Gestaltung einer geeigneten Entscheidungsfunktion für ein Klassifikationssystem ermöglicht. Es wurde eine allgemeine Definition für eine auf nichtlinearen Verformungen basierende Distanzfunktion gegeben. Diese ist abhängig von der Modellierung der zulässigen Verformungen. Es wurden verschiedene nichtlineare Verformungsmodelle mit ihren Verformungseinschränkungen vorgestellt und zur Durchführung von Experimenten implementiert. Bis auf das IDM beruhen die präsentierten Modelle auf dem HMM. Dieses nichtlineare Verformungsmodell wird unter anderem erfolgreich in der Spracherkennung zur Anpassung eines Signals verwendet.

Die gewonnenen Fehlerraten auf den zur Klassifikation benutzten Datensammlungen rückten durch die Verwendung der durch den Sobelfilter erzeugten vertikalen und horizontalen Gradientenbildern in den Bereich der besten, von anderen Forschungsgruppen erzielten Ergebnisse. Hierdurch wurden Experimente mit weniger fehlerhaften Gradientenbildern durch Splinekoeffizienten motiviert. Die durchgeführten Versuche zeigten, dass diese Gradientenbilder erst durch eine Glättung zu vergleichbar guten Ergebnissen, wie die durch den Sobelfilter erzeugten Merkmale, führen. Die Verwendung der Gradientenbilder lässt sich als Berücksichtigung der lokalen Kontextinformationen eines Pixels zur Pixeldistanzbestimmung verstehen. Dies motivierte Experimente mit Pixeldistanzen, die auf Bildausschnitten, deren Zentrum die zu vergleichenden Pixel sind, beruhen. Solche Pixeldistanzen schränken eine kostenminimierende Bildverformung zusätzlich ein und es wurden daher Versuche mit dem IDM gemacht, um zu testen, ob diese Pixeldistanzen sich positiv auf die Resultate dieses einfachen Verfahrens auswirken. Auf diesem Weg wurden mittels des ID-Verfahrens mit sehr gute Ergebnisse auf den vorgestellten Datensammlungen erzielt. Es wird ein erweitertes P2DHMM eingeführt, das Pseudo-2DHM Distortion Modell (P2DHMD-Modell), mit dem sich die Ergebnisse, im Vergleich zum IDM oder dem P2DHMM, weiter verbessern liessen.

Die Ergebnisse des vorgestellten Klassifikationssystems, das auf der präsentierten Distanzfunktion beruht, gehören zu den niedrigsten Fehlerraten, die auf den vorge-

Tabelle 7.1: Niedrigste veröffentlichte Fehlerraten vorgestellter Datensammlungen

Korpus	USPS	UCI	MCEDAR	MNIST	SiOI	SiFF	ETL6A	IRMA	WANG
Andere	2.0	1.5	4.3	0.6	6.6	1.8	0.5	8.0	21.8
diese Arbeit	2.1	0.8	3.6	0.6	2.7	1.2	0.5	5.9	22.1

stellten Datensammlungen, erzielt wurden. Tabelle 7.1 stellt die besten veröffentlichten Ergebnisse den in dieser Arbeit erzielten Fehlerraten gegenüber. Genaue Angaben zu den verwendeten Verfahren finden sich in den entsprechenden Abschnitten des Kapitels 6.

Es stehen weitere Experimente mit Verformungseinschränkungen aus, die abhängig von der Klasse oder dem Referenzdatum modelliert werden. Desweiteren wäre eine Erweiterung der Distanzfunktion interessant, die zur Bestimmung der idealen Pixellabbildungen, welche durch die Waringabbildung beschrieben werden, mehrere Referenzdaten derselben Klasse verwendet. Dadurch könnte die Mannigfaltigkeit handgeschriebener Zeichen möglicherweise besser modelliert werden. Ein möglicher Wechsel zu einem weiteren Referenzdatum zur Erklärung eines Testbildes könnte in einem solchen Verfahren abhängig von lokalen Kontextinformationen der Referenzbilder gestaltet werden. Diese Waringabbildung, die für die Erklärung eines Testbildes mehrere Referenzbilder berücksichtigt, könnte das Klassifikationssystem für die Erkennung geschriebener Wörter erweitern, indem ein Referenzdatum ein Wort und die zugehörigen Buchstabengrenzen dieses Wortes beschreibt. An den Buchstabengrenzen würden Abbildungswechselkosten für Referenzbildwechsel geringer ausfallen und das Klassifikationsergebnis wäre die kostenminimale Waringabbildung, die das Testbild durch Buchstabenabschnitte der Referenzbilder erklärt. Diese wortbeschreibende Abbildung könnte zusätzlich durch eine linguistische Wortwahrscheinlichkeit, die eine weitere Kostenfunktion modelliert, eingeschränkt werden.

Die nach der Bildvorverarbeitung verwendeten erweiterten Pixelmerkmale und die resultierende erweiterte Pixeldistanz sind abhängig von der Auflösung der betrachteten Bilder. Dies motiviert Experimente mit Pixeldistanzfunktionen, die sich an die Auflösung der Bilder anpassen, anstatt die Auflösung der Bilder anzupassen, da durch eine Reduzierung der Bildauflösung möglicherweise wichtige Merkmalsinformationen nicht berücksichtigt werden.

Diese Arbeit zeigt auf, dass das vorgestellte Bildererkennungssystem die vorgestellten Klassifikationsprobleme im Vergleich zu anderen Verfahren erfolgreich löst. Mit den untersuchten Verfahren konnte auf allen betrachteten Datensammlungen eine zu anderen Verfahren konkurrenzfähige, auf einigen sogar die beste bekannte Fehlerrate erzielt werden.

# Literaturverzeichnis

- [Agazzi & Kuo 93] O. Agazzi, S. Kuo: Pseudo two-dimensional hidden markov models for document recognition. In *AT&T Technical Journal*, pp. 60–72, September 1993.
- [Belongie & Malik<sup>+</sup> 02] S. Belongie, J. Malik, J. Puzicha: Shape matching and object recognition using shape contexts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24(4), pp. 509–522, April 2002.
- [Bishop & Winn 00] C.M. Bishop, J. Winn: Non-linear bayesian image modelling. In *6th European Conference on Computer Vision*, Vol. 1, pp. 3–17, Dublin, Irland, Juni 2000.
- [Bredno & Brandt<sup>+</sup> 00] J. Bredno, S. Brandt, J. Dahmen, B. Wein, T. Lehmann: Kategorisierung von Röntgenbildern mit aktiven Konturmodellen. In *Bildverarbeitung für die Medizin 2000*, pp. 356–360, März 2000.
- [Deselaers 03] T. Deselaers: Features for Image Retrieval. Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen. Dezember 2003.
- [Dong & Krzyzak<sup>+</sup> 99] J.X. Dong, A. Krzyzak, C.Y. Suen: Comparison of algorithms for handwritten numeral recognition. In *CENPARMI, Technical Report*. Concordia University, November 1999.
- [Dong & Krzyzak<sup>+</sup> 01] J.X. Dong, A. Krzyzak, C.Y. Suen: Statistical results of human performance on USPS database. In *CENPARMI, Technical Report*. Concordia University, Oktober 2001.
- [Dong & Krzyzak<sup>+</sup> 02a] J.X. Dong, A. Krzyzak, C.Y. Suen: Local learning framework for handwritten character recognition. In *Engineering Applications of Artificial Intelligence*, Vol. 15, pp. 151–159, April 2002.
- [Dong & Krzyzak<sup>+</sup> 02b] J.X. Dong, A. Krzyzak, C.Y. Suen: A practical SMO algorithm. In *ICPR 2002, 16th International Conference on Pattern Recognition*, Quebec City, Kanada, August 2002.
- [Drucker & Schapire<sup>+</sup> 93] H. Drucker, R. Schapire, P. Simard: Boosting performance in neural networks. In *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7(4), pp. 705–719, 1993.
- [Güld 00] M.O. Güld: Inhaltsbasierter Bildzugriff mittels Statistischer Objekterkennung. Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen. Juli 2000.
- [Hinton & Dayan<sup>+</sup> 97] G.E. Hinton, P. Dayan, M. Revow: Modeling the manifolds of images of handwritten digits. In *IEEE Transactions on Neural Networks*, Vol. 8(1), pp. 65–74, Januar 1997.
- [Hromkovic 03] J. Hromkovic: Algorithmics for hard problems. 2. Auflage. Springer-Verlag, 2003.
- [Jähne 02] B. Jähne: Digitale Bildverarbeitung. 5. Auflage. Springer-Verlag, 2002.

- [Keysers & Dahmen<sup>+</sup> 00] D. Keysers, J. Dahmen, T. Theiner, H. Ney: Experiments with an extended tangent distance. In *ICPR 2000, 15th International Conference on Pattern Recognition*, Vol. 2, pp. 38–42, Barcelona, Spanien, September 2000.
- [Keysers & Dahmen<sup>+</sup> 03] D. Keysers, J. Dahmen, H. Ney, B. Wein, T. Lehmann: Statistical framework for model-based image retrieval in medical applications. In *Journal of Electronic Imaging*, Vol. 12(1), pp. 59–68, Januar 2003.
- [Keysers & Paredes<sup>+</sup> 02] D. Keysers, R. Paredes, H. Ney, E. Vidal: Combination of tangent vectors and local representations for handwritten digit recognition. In *SPR 2002, International Workshop on Statistical Pattern Recognition*, LNCS 2396, pp. 538–547, Windsor, Kanada, August 2002.
- [Keysers & Unger 03] D. Keysers, W. Unger: Elastic image matching is NP-complete. In *Pattern Recognition Letters*, Vol. 24(1-3), pp. 445–453, Januar 2003.
- [Keysers 00] D. Keysers: Approaches to invariant image object recognition. Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen. Juni 2000.
- [Kim & Kim<sup>+</sup> 02] H.J. Kim, D. Kim, S.Y. Bang: A numeral character recognition using the PCA mixture model. In *Pattern Recognition Letters*, Vol. 23(1-3), pp. 103–111, Januar 2002.
- [LeCun & Boser<sup>+</sup> 89] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel: Backpropagation applied to handwritten zip code recognition. In *Neural Computation*, Vol. 1(4), pp. 541–551, 1989.
- [LeCun & Boser<sup>+</sup> 90] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel: Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2*, Denver, CO, 1990. Morgan Kaufman.
- [LeCun & Bottou<sup>+</sup> 98] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner: Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, Vol. 86(11), pp. 2278–2324, November 1998.
- [Levin & Pieraccini 92] E. Levin, R. Pieraccini: Dynamic Planar Warping for Optical Character Recognition. In *ICASSP 1992, International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 149–152, März 2003.
- [Li & Wang 03] J. Li, J.Z. Wang: Automatic linguistic indexing of pictures by a statistical modeling approach. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25(9), pp. 1075–1088, September 2003.
- [Mori & Yamamoto 84] S. Mori, K. Yamamoto: Research on machine recognition of handprinted characters. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6(4), pp. 386–405, April 1984.
- [Ney 01] H. Ney: Speech recognition. Skript zur Vorlesung. RWTH Aachen, Wintersemester 2001.
- [Ney 02] H. Ney: Mustererkennung und Neuronale Netze. Skript zur Vorlesung. RWTH Aachen, Sommersemester 2002.
- [Ney 03] H. Ney: Digital processing of speech and image signals. Skript zur Vorlesung. RWTH Aachen, Sommersemester 2003.
- [Paredes & Keysers<sup>+</sup> 02] R. Paredes, D. Keysers, T.M. Lehmann, B.B. Wein, H. Ney, E. Vidal: Classification of medical images using local representations. In *Bildverarbeitung für die Medizin 2002*, pp. 171–174, März 2002.

- [Paredes & Pérez<sup>+</sup> 01] R. Paredes, J.C. Pérez, A. Juan, E. Vidal: Local representations and a direct voting scheme for face recognition. In *Workshop on Pattern Recognition in Information Systems*, Setúbal, Portugal, Juli 2001.
- [Perrey 00] R. Perrey: Affin-invariante Merkmale für die 2D-Bildererkennung. Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen. Februar 2000.
- [Perronnin & Dugelay<sup>+</sup> 03] F. Perronnin, J.L. Dugelay, K. Rose: Iterative decoding of two-dimensional hidden markov models. In *ICASSP 2003, International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 329–332, April 2003.
- [Rabiner & Juang 93] N. Rabiner, B.H. Juang: Fundamentals of speech recognition. Englewood Cliffs, NJ, 1993. Prentice Hall.
- [Schölkopf & Simard<sup>+</sup> 98] B. Schölkopf, P. Simard, A. Smola, V. Vapnik: Prior knowledge in support vector kernels. In *Advances in Neural Information Processing Systems*, Vol. 10, pp. 640–646. MIT Press, 1998.
- [Seidl 02] T. Seidl: Modelle der Datenexploration. Skript zur Vorlesung. RWTH Aachen, Wintersemester 2002.
- [Simard & LeCun<sup>+</sup> 93] P. Simard, Y. LeCun, J. Denker: Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, 1993.
- [Simard & LeCun<sup>+</sup> 94] P. Simard, Y. LeCun, J. Denker: Memory based character recognition using a transformation invariant metric. In *ICPR 1994, 12th International Conference on Pattern Recognition*, Vol. 2, pp. 262–267, Jerusalem, Israel, Oktober 1994.
- [Theiner 00] T. Theiner: Inhaltsbasierter Zugriff auf große Bilddatenbanken. Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen. Februar 2000.
- [Tipping & Bishop 99] M.E. Tipping, C.M. Bishop: Mixtures of probabilistic principal component analyzers. In *Neural Computation*, Vol. 11 (2), pp. 443–482, 1999.
- [Tipping 00] M.E. Tipping: The relevance vector machine. In *Advances in Neural Information Processing Systems 12*, pp. 652–658. MIT Press, 2000.
- [Uchida & Sakoe 98] S. Uchida, H. Sakoe: A monotonic and continuous two-dimensional warping based on dynamic programming. In *ICPR 1998, 14th International Conference on Pattern Recognition*, pp. 521–524, Brisbane, Australia, September 1998.
- [Uchida & Sakoe 99] S. Uchida, H. Sakoe: Handwritten character recognition using monotonic and continuous two-dimensional warping. In *ICDAR 1999, 5th International Conference on Document Analysis and Recognition*, pp. 499–502, Bangalore, India, September 1999.
- [Uchida & Sakoe 00] S. Uchida, H. Sakoe: Piecewise linear two-dimensional warping. In *ICPR 2000, 15th International Conference on Pattern Recognition*, Vol. 3, pp. 538–541, Barcelona, Spanien, September 2000.
- [Uchida & Sakoe 02] S. Uchida, H. Sakoe: A handwritten character recognition method based on unconstrained elastic matching and eigen-deformations. In *8th International Workshop on Frontiers of Handwriting Recognition*, pp. 72–77, Niagara-on-the-Lake, Ontario, Canada, September 2002.
- [Uchida & Sakoe 03a] S. Uchida, H. Sakoe: Eigen-deformations for elastic matching based handwritten character recognition. In *Pattern Recognition*, Vol. 36(9), pp. 2031–2040, September 2003.

- [Uchida & Sakoe 03b] S. Uchida, H. Sakoe: Handwritten character recognition using elastic matching based on a class-dependent deformation model. In *ICDAR 2003, 7th International Conference on Document Analysis and Recognition*, Vol. 1, pp. 163–167, Edinburgh, Schottland, August 2003.
- [Wang & Li<sup>+</sup> 01] J.Z. Wang, J. Li, G. Wiederhold: Simplicity: Semantics-sensitive integrated matching for picture libraries. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23(9), pp. 947–963, September 2001.
- [Wang & Srihari 88] G.H. Wang, S.N. Srihari: A framework for object recognition and its application to locating address blocks on mail pieces. In *7th National Conference on Artificial Intelligence 1988*, St. Paul, MN, August 1988.

# Anhang A

## Verwendete Software

Die Software, die für diese Arbeit verwendet wurde, beruht auf dem freien Betriebssystem Linux<sup>1</sup>. Der Quelltext und die Diplomarbeit wurden mit dem GNU Emacs<sup>2</sup> Editor erstellt und bearbeitet. Diese Diplomarbeit ist mit L<sup>A</sup>T<sub>E</sub>X<sup>3</sup> gesetzt worden und die dargestellten Abbildungen wurden mit Xfig<sup>4</sup> erzeugt. Die Bearbeitung der Beispielbilder erfolgte durch die Programme XV<sup>5</sup> und GIMP<sup>6</sup>. Die Graphen wurden mit Gnuplot<sup>7</sup> erstellt. Für kleinere Textänderungen sowie die Erstellung von Skripten wurde der Vim<sup>8</sup> Editor verwendet. Es wurden BASH<sup>9</sup>, Perl<sup>10</sup>, awk<sup>11</sup> und sed<sup>12</sup> Skripte für sich oft wiederholende Aufgaben erstellt. Die Versionskontrolle der Quelltexte erfolgte mittels CVS<sup>13</sup>.

Für die Implementierung der für die Experimente nötigen Programme wurden die GNU Entwicklungstools<sup>14</sup> verwendet. Die erstellte Software ist in C geschrieben und verwendet neben den notwendigen Standardbibliotheken folgende aufgeführte Bibliotheken für optionale Features der implementierten Programme:

- zlib<sup>15</sup>, Leseroutinen gzip-komprimierter Dateien
- GD library<sup>16</sup>, Lese- und Schreibroutinen für Bilder
- pnglib<sup>17</sup>, Unterstützung für Bilder im PNG-Format
- jpeg library<sup>18</sup>, Unterstützung für Bilder im JPEG-Format
- spline code<sup>19</sup>, Spline Interpolation von Bildwerten
- X11 library<sup>20</sup>, Darstellung der Bilddaten

<sup>1</sup><http://www.kernel.org/>

<sup>2</sup><http://www.gnu.org/software/emacs/>

<sup>3</sup><http://www.latex-project.org/>

<sup>4</sup><http://www.xfig.org/>

<sup>5</sup><http://www.trilon.com/xv/>

<sup>6</sup><http://www.gimp.org/>

<sup>7</sup><http://www.gnuplot.info/>

<sup>8</sup><http://www.vim.org/>

<sup>9</sup><http://www.gnu.org/software/bash/>

<sup>10</sup><http://www.perl.org/>

<sup>11</sup><http://www.gnu.org/software/gawk/>

<sup>12</sup><http://www.gnu.org/software/sed/>

<sup>13</sup><http://www.cvshome.org/>

<sup>14</sup><http://gcc.gnu.org/>

<sup>14</sup><http://www.gnu.org/software/gdb/>

<sup>14</sup><http://www.gnu.org/software/autoconf/>

<sup>14</sup><http://www.gnu.org/software/automake/>

<sup>15</sup><http://www.gzip.org/zlib/>

<sup>16</sup><http://www.boutell.com/gd/>

<sup>17</sup><http://www.libpng.org/pub/png/>

<sup>18</sup><http://www.ijg.org/>

<sup>19</sup><http://bigwww.epfl.ch/thevenaz/interpolation/>

<sup>20</sup><http://www.xfree86.org/>



## Anhang B

# Erstellte Software

Durch die Verwendung der GNU Autotools wurde von der erstellten Software ein auf UNIX Systemen kompilierbares Softwarepaket erzeugt. Es ist geplant, dieses Softwarepaket für nicht kommerzielle Zwecke öffentlich zur Verfügung zu stellen. Es beinhaltet Programme, die für die in dieser Diplomarbeit beschriebenen Experimente implementiert wurden. Im Folgenden werden die Verwendungsmöglichkeiten dieser Programme und die verwendeten Datenformate beschrieben.

### B.1 Datenformate

Die Bilddaten werden im i6-ASCII-File-Format gespeichert, welches hier erläutert wird. Dieses Format wird am Lehrstuhl VI verwendet um mehrere Bilder und ihre zugehörige Klasse zu beschreiben. Eine Datei dieses Formats kann als Textdatei bearbeitet werden und besteht im Textformat aus ganzen sowie reellen Zahlen. Der Header beinhaltet die Klassenanzahl und die Dimension der Bilder. Jede weitere Zeile dieser Textdatei beschreibt ein Bild durch die Angabe der Klasse und der reellen Bildwerte.

```
<Klassenanzahl> <Bilddimension D>  
<Klasse> <Bildwert 1> ... <Bildwert D>  
...
```

Eine Erweiterung dieses Dateiformats ist das Extended-i6-ASCII-File-Format. Dieses Format berücksichtigt Bilder unterschiedlicher Größen, indem die Bildbreite und die Bildhöhe für jedes Bild einzeln angegeben wird.

```
<Klassenanzahl>  
<Klasse> <Bildbreite> <Bildhöhe> <Bildwert 1> ... <Bildwert D>  
...
```

Indem mehrere solcher Bildlisten im Extended-i6-ASCII-File-Format hintereinander geschrieben werden, kann eine auf diese Weise erstellte Datei Bilder mit Pixelvektoren

repräsentieren.

Ein weiteres Dateiformat, welches zur Gruppe der Textdateien gezählt werden kann, beinhaltet Distanzwerte von Bildern. Eine solche Distanzdatei ermöglicht die Trennung der Distanzberechnung von der Entscheidungsregel. Dadurch können ohne wiederholende Distanzberechnungen verschiedene Entscheidungsregeln mit unterschiedlichen Einstellungen erprobt werden. Die Datei beschreibt die Bildnummern und Klassen der verglichenen Bilder gefolgt von mehreren Distanzwerten. Diese Distanzwerte stehen z.B. für eine berechnete Distanz inklusive der Strafkosten oder exklusive dieser Kosten. Dabei darf die Anzahl der Distanzwerte pro Bildvergleich und somit pro Zeile der Distanzdatei nicht variieren.

```
# Dies ist ein Kommentar
$ Dies ist ebenso ein Kommentar
<Test Nr.> <Ref. Nr.> <Test Kl.> <Ref. Kl.> <Distanz 1> ...
...
```

Diese Dateiformate ermöglichen einfache Lesemethoden der Daten unabhängig von der Rechnerarchitektur und der damit verbundenen Byte-Ordnung. Der Nachteil dieses Formats ist der ineffiziente Verbrauch von Speicher. Dieser kann durch die Verwendung des gzip Programms zur Komprimierung der Daten umgangen werden. Mit den Lesefunktionen der zlib Bibliothek können die derart komprimierten Daten von den im folgenden Abschnitt vorgestellten Programmen gelesen werden.

## B.2 Programme

Das `images2jf` Programm kann verwendet werden um aus Bildern im PNG- oder JPEG-Format eine Liste von Bildern im i6-ASCII-File-Format zu erzeugen, indem dem Programm eine Textdatei mit den Dateinamen der zu verwendenden Bilder übergeben wird. Diese werden so benannt, dass der Dateiname mit der Klasse des Bildes beginnt.

```
Usage: images2jf [OPTION...] [FILE1] [FILE2]
images2jf -- The program reads FILE1 which contains the imagefilenames,
and writes the FILE2 in jf-format.
```

```
-B, --blue           Use blue image values
-E, --grey          Use grey image values
-G, --green         Use green image values
-R, --red           Use red image values
```

Das `idx2jf` Programm wird verwendet, um aus Bildsammlungen, die im `idx`-Format (MNIST-Datensammlung) vorliegen, Bildsammlungen im `i6-ASCII-File-Format` zu erzeugen.

```
Usage: idx2jf [OPTION...] [FILE1] [FILE2] [FILE3]
idx2jf -- A program to convert MNIST idx files.
FILE1 is the idx3 data file, FILE2 is the idx1 label file and FILE3
will be created in jf-format.
```

Das `etlm2jf` Programm wird verwendet, um aus Bildsammlungen, die im `ETLM`-Format vorliegen, Bildsammlungen im `i6-ASCII-File-Format` zu erzeugen.

```
Usage: etlm2jf [OPTION...] [FILE1] [FILE2]
-d, --downscale          Downscale Images to
                          16x16 Pixel
-v, --verbose            Produce verbose output
etlm2jf -- A program to convert ETLM files.
FILE1 is the ETLM data file and FILE2 will be created in jf-format.
```

Das `jf2jf` Programm wird verwendet, um Bildsammlungen, die im `i6-ASCII-File-Format` vorliegen, in eine neue Bildsammlung im `i6-ASCII-File-Format` umzuwandeln. Eine weitere Anwendungsmöglichkeit des Programms ist die Anzeige der Bilder der Datensammlung.

```
Usage: jf2jf [OPTION...] [OPTION...] [FILE1] <[FILE2]>
jf2jf -- The program reads the FILE1 in jf-format and if given creates
a new FILE2 in jf-format.
-C, --class=N            Class to separate
-I, --imglist=[FILE]     Textfile with image numbers
-K, --classlist=[FILE]  Textfile with class numbers
-p, --pause              Wait after each image
-v, --verbose            Verbose output
-x, --show               Show images on X11
```

Das `jfpad` Programm kann verwendet werden, um Padding an Bildern durchzuführen. Dadurch können z.B. nicht quadratische Bilder zu quadratischen Bildern umgeformt werden.

Usage: `jfpad [OPTION...] [FILE1] [FILE2]`

`jfpad --` The program reads `FILE1`, and pads the images to write `FILE2`.

<code>-b, --border=N</code>	Add border to images
<code>-p, --value=R</code>	Paddingvalue for added pixels
<code>-q, --quadratic</code>	Pad to quadratic images
<code>-s, --samesize</code>	Pad each image to same size

Für die Skalierung von Bildern kann das Programm `jfscale` verwendet werden. Zur Bestimmung der neuen Bildwerte kann zwischen der Interpolation durch Splines oder der linearen Interpolation gewählt werden.

Usage: `jfscale [OPTION...] [FILE1] [FILE2]`

`jfscale --` The program reads `FILE1`, scales the images and writes `FILE2`.

<code>-d, --degree=N</code>	Splinedegree 2-5
<code>-h, --height=N</code>	Destination image height
<code>-M, --mode=bresenham spline</code>	Choose the scale algorithm
<code>-w, --width=N</code>	Destination image width

Das Programm `jfnorm` kann verwendet werden, um Bildwerte von Bildern einer Datensammlung zu normieren.

Usage: `jfnorm [OPTION...] [FILE1] [FILE2]`

`jfnorm --` The program reads `FILE1`, and norms the images to write `FILE2`.

<code>-a, --all</code>	Norm all images (include imagedepth)
<code>-e, --end=R</code>	New endgreyvalue
<code>-g, --gamma=R</code>	Value for gammatransformation after greyvaluenorm
<code>-s, --start=R</code>	New startgreyvalue

Das Programm `jffilter` kann verwendet werden, um Filteroperationen an Bildern einer Datensammlung auszuführen.

```
Usage: jffilter [OPTION...] [FILE1] [FILE2]
jffilter -- The program reads FILE1, filters the images and writes FILE2.

  -F, --filter=<type>          Choose the filter
  -I, --filterfile=[FILE]     Load filterfile
  -v, --verbose                Verbose output
  -w, --warparound            Warparound
```

The default filter-file is 'filterlist.fildat'

Example:

```
filterA 2x2
+0 -1
-1 +0

filterB 3x1
0 -1.0 0
```

Die Bestimmung der Splinegradienten einer Bildsammlung kann durch das Programm `jfgradient` erfolgen.

```
Usage: jfgradient [OPTION...] [FILE1] [FILE2]
jfgradient -- The program reads FILE1, and writes gradient FILE2.

  -d, --degree=N              Splinedegree 2-5
```

Die Extraktion einzelner Bilder einer Datensammlung im i6-ASCII-File-Format kann durch das Programm `jf2png` erfolgen.

```
Usage: jf2png [OPTION...] [FILE]
jf2png -- The program reads a jf FILE, and writes png images.

  -C, --class=N              Class to seperate
  -E, --total=N             Total number of images to save
  -i, --inverse              Inverse image data output
  -O, --outname=[FILE...]   Name PNG files start with
  -S, --start=N             Image number to start output
  -v, --verbose              Verbose output
```

Die Distanzberechnung durch unterschiedliche Verformungsverfahren kann durch das Programm W2D erfolgen.

Usage: W2D [OPTION...] [FILE1] [FILE2]

W2D -- The program reads train FILE1 and test FILE2, the output are distances.

```

-A, --trainstart=N           First train image
-B, --beamsize=N            Beamsize for distance algorithm
-C, --clustermode=N         Clustermode
-D, --data=(file/image/    Image data type, mode and scale
  nulling/sobelv/sobelh/   factor
  penalty):(abs/euc2/
  euc2rec3x3/euc2rec5x5/
  euc2bin3x3/euc2bin5x5/
  warpsum:0.0/pen/pen2/
  len/len2/penborder/
  penwarp/
  pixelthreshold):R
-E, --testtotal=N          Number of test images to load
-F, --trainsave=[FILE]     Save modified train data
-G, --testclass=N         Class for test images
-h, --hflip                Flip all images horizontal
-H, --trainclass=N        Class for train images
-K, --threshold=R         Threshold for distance algorithm
-L, --disdat=[FILE]       Read disdat-file
-M, --modell=(WW2DHM/P2DHM/ Warping modell
  P2DHMD/SA2DHM/IDM/NO)
-N, --nearest=N           Nearest neighbor list size
-O, --addborder=N         Add border to the images
-p, --pause                Wait after each image
-Q, --normmode=N          Normmode(0=all images / 1=seperate
  train and test / 2=no)
-r, --rotate               Rotate all images 90° right
-R, --traintotal=N        Number of train images to load
-s, --shift                Shift 3x3 train data
-S, --teststart=N         First test image
-v, --vflip                Flip all images vertical
-V, --nearestmode=N       Nearest neighbor mode
-w, --write                Write PNG files from images
-W, --warprange=N         Set warprange
-x, --show                 Show images on X11
-y, --symmdis              Symmetric distance

```

Example:

```
W2D train.ejf.gz test.jf -MIDM -W2 -Dimage:euc2:1,penalty:len:0.5
```

Die Klassifizierung anhand von Distanzwerten erfolgt durch das Programm `classifier`.

Usage: `classifier [OPTION...] [FILE]`

`classifier --` The program read a disdat file and print the errorrate for each column of distances.

<code>-a, --alpha=R:R:R:...</code>	Scalefactor for variancefactor
<code>-c, --calc</code>	Calculate average and variance for each class
<code>-f, --disfield=N</code>	Select the N'th distance column, counted from 0
<code>-k N</code>	Select the N best distances for kNN, default 1
<code>-m, --matrix</code>	Classified matrix
<code>-M, --mode=kNN KD</code>	Choose the classifier
<code>-v, --verbose</code>	Verbose output
<code>-V, --variance=R</code>	Variancefactor

When FILE is -, read standard input.

Example:

```
W2D train.jf test.jf | classifier - -f1
```