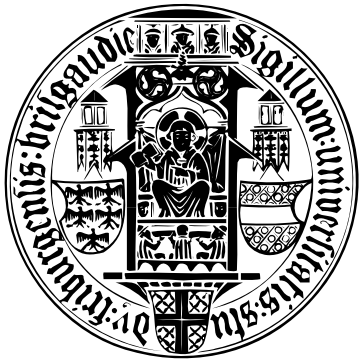
ALBERT-LUDWIGS-UNIVERSITÄT
FREIBURG
INSTITUT FÜR INFORMATIK

Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr. Hans Burkhardt



Erkennung von handgeschriebenen Wörtern mit CSDTW

Diplomarbeit

Kai Simon

November 2002 – Mai 2003

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit von mir selbständig und nur unter Verwendung der aufgeführten Hilfsmittel erstellt wurde.

Freiburg, den 19.05.2003

Inhaltsverzeichnis

KAPITEL 1

Einleitung

Im Laufe der Entwicklung des Computers wurden verschiedene Eingabegeräte für den Anwender/die Anwenderin[1] entworfen. In den 50er- und 60er-Jahren zählten zu den typischen Eingabegeräte einer Rechenanlage Lochstreifen- und Lochkartenleser, in den 70er und 80er-Jahren kam die Tastatur hinzu und im technischen Bereich der Lichtgriffel[2]. Heutzutage bilden Tastatur, Maus, Scanner, Videokamera und Spracheingabegeräte viele Möglichkeiten für die Dateneingabe (siehe Claus and Schwill [13]). Sie ermöglichen den Anwendern einen benutzerfreundlichen Umgang mit dem Computer. Während sich die Maus als Navigationshilfe (Zeigeinstrument) bei der Datenbearbeitung am Bildschirm etablieren konnte, dominiert die Tastatur weiterhin bei der Texteingabe. Der Wunsch nach 'natürlicheren' Alternativen zur Texteingabe führte dazu, dass bereits Anfang der 70er Jahre Sprach- und Handschrifterkennungssysteme entwickelt wurden. In den Anfangsjahren lag das wirtschaftliche Interesse und somit die bessere finanzielle Förderung der Forschungsarbeiten eher auf Seiten der Spracherkennungssysteme. Dies hatte zur Folge, dass im Bereich der Spracherkennung große Forschungserfolge erzielt werden konnten, währenddessen der Fortschritt in der Handschrifterkennung eher zögerlich von statten ging. Durch die Fortschritte in der Miniaturisierung und dem damit verbundenen Mobile-Computing erlangte auch die Handschrifterkennung ein neues Einsatzgebiet, was das Interesse der Industrie wiedererweckte. Viele Institute, die sich damals und heute noch mit der Spracherkennung beschäftigen, modifizierten daraufhin ihre bereits entwickelten Systeme. Die in den 70er Jahren entwickelten theoretischen Modelle bilden heute die Grundlage der meisten Erkennungssysteme.

Doch erst die Rechenleistung heutiger Rechner hat diesen Eingabe-Alternativen den Einstieg in die Anwenderwelt ermöglicht. Denn während bei der Tastatureingabe eine eindeutige Zuord-

[1] Aus Gründen der Lesbarkeit wird im weiteren Verlauf der Arbeit nur die männliche Form verwendet.

[2] Lichtgriffel (engl. light pen): Stift, in dessen Spitze anstelle einer Mine eine lichtempfindliche Zelle (Photozelle) eingebaut ist. Durch Berühren der Bildschirmfläche mit der Griffelspitze können Punkte und Linien gezeichnet werden.

nung zwischen Tastendruck (digitalem Signal) und ASCII Code (Semantik) erfolgt, ist dies bei der Sprach- und Schrifterkennung nicht möglich. Hier muss mittels Erkennungsalgorithmen aus dem digitalisierten Eingabesignal anhand von gelernten digitalen Referenzsignalen ein Bezug zur Semantik hergestellt werden.

Die vorliegende Arbeit beschäftigt sich mit der Handschrifterkennung. Dieses Gebiet läßt sich bereits anhand des verwendeten Verfahrens zur Datenerfassung in zwei Teilgebiete aufspalten.

1.1 Offline- vs. Online-Verfahren

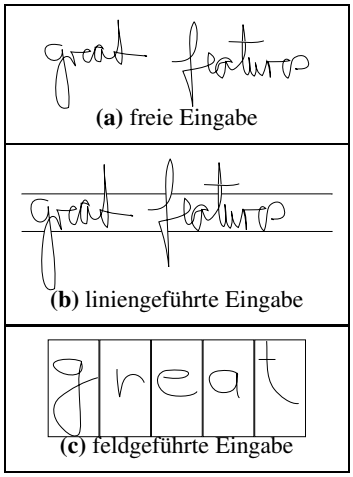
Man unterscheidet je nach Art des Erkennungszeitpunktes zwischen zwei Kategorien: *Online*- und *Offline*-Verfahren.

Beim Offline-Verfahren wird bereits handgeschriebener Text indirekt durch einen Scanner erfasst und vom Computer ausgewertet. Dieses Verfahren wird zum Beispiel bei der Briefsortierung eingesetz. Die Schwierigkeit besteht einerseits in der Extraktion der Handschrift aus der digitalisierten Bildmatrix, andererseits in der anschließenden Wort- beziehungsweise Buchstabenklassifikation.

Beim Online-Verfahren wird der Text direkt während des Schreibvorgangs eingelesen. Die Stiftbewegungen werden als eine Sequenz von Orts- und Zeitkoordinaten erfasst. Zusätzlich besteht die Möglichkeit, den Stiftwinkel und Information über Stiftkontakt mit der Schreiboberfläche in jedem Abtastpunkt digital einzulesen. Das ursprüngliche Verfahren zur Online-Schrifteingabe entstand aus dem Entwurf technischer Zeichengeräte. Die Eingabe erfolgt über ein Grafiktablett, auf dem mit einem Griffel geschrieben wird. Durch die Entwicklung so genannter PDAs [3] und Tablet PCs [4] hat sich ein weiteres Verfahren für die Online-Eingabe etabliert. Die PDAs sind hierzu mit drucksensitiven Bildschirmen (Displays) ausgestattet, auf denen mit einem gewöhnlichen Stift (ohne Mine) geschrieben werden kann. Im Falle der größeren Tablet PCs verwenden die meisten Geräte elektromagnetische Resonanztechnik, sodass der Stift rein passiv arbeitet und keine Batterie benötigt. Der Stift besteht aus einer Spule, die vom Tablet induzierte Resonanzschwingungen zurücksendet. Aus der Stärke und Reflexion bezieht das Tablet sowohl die Stiftkoordinaten als auch die Druckstärke. Die Eingabemöglichkeit mittels Stift ist besonders effizient und platzsparend und gewinnt durch den vermehrten Einsatz der Geräte an zunehmender Bedeutung.

[3] Abkürzung für 'Personal Digital Assistent' auch Organizer genannt. Handlicher Miniatur-Computer, bei dem Eingaben nicht über eine Tastatur, sondern auf dem Display mit Hilfe eines speziellen Stiftes erfolgen. Das PDA wird hauptsächlich für die Verwaltung von Adressen, Terminen und Kurznotizen benutzt.

[4] Computer der nur aus einem LCD Bildschirm besteht. Es ist keine physikalische Tastatur und Maus vorhanden, nur mittels eines Stiftes können Eingaben getätigt werden. Will man eine Eingabe per Tastatur durchführen, so muss man sich mit einer virtuellen Tastatur zurechtfinden. Die Handschrift bildet eine der wichtigsten Eingabemethoden.



(a) freie Eingabe

(b) liniengeführte Eingabe

(c) feldgeführte Eingabe

Abbildung 1.1: Vorgaben bei der Texteingabe

1.2 Handschriftsegmentierung

Ein wichtiger Schritt bei der Handschrifterkennung stellt die Segmentierung der Eingabe auf Wort beziehungsweise Zeichenebene dar. Beim Offline-Verfahren müssen hierfür die Daten zuerst auf der Pixel Ebene vom Hintergrund extrahiert werden, bevor eine Zerlegung stattfinden kann. Beim Online-Verfahren gestaltet sich dieser Prozess wesentlich einfacher, wobei er bei weitem noch nicht als trivial bezeichnet werden kann. Hier registriert ein Signal ob der Stift sich auf der Schreiboberfläche befindet oder nicht. Mit dessen Hilfe lassen sich Wortgrenzen wesentlich einfacher auffinden als dies beim Offline-Verfahren der Fall ist. Auf Zeichenebene stellen jedoch *kursiv* geschriebene Zeichen und verzögert gezeichnete Striche (engl. *delayed strokes*) weiterhin ein Problem dar. So werden beim kursiven Schreibstil mehrere Zeichen ohne absetzten des Stiftes zusammengeschrieben, wodurch eine Zeichensegmentierung anhand des Signals zur Fehlinterpretation führt. Während bei delayed strokes ein Absetzen des Stiftes innerhalb eines Zeichens stattfindet, wie es zum Beispiel bei den Buchstaben 'x','i','j' und 't' zumeist der Fall ist. Den schwierigsten Fall stellen Wörter dar, die aus mehreren Schreibstilen zusammengesetzt sind (engl. *mixed style*). Je schwieriger die Zeichensegmentierung ist, desto größer ist die Wahrscheinlichkeit einer Fehlklassifikation eines Wortes. Umso wichtiger werden für das Erkennungssystem Informationen auf linguistischer Ebene, anhand derer es möglich ist die Segmentierungshypothesen einzuschränken.

Desweiteren kann der Schreiber aufgefordert werden, seine Eingabe bezüglich vorgegebener Richtlinien zu erstellen. Somit können bei der Texteingabe weitere Unterscheidungen vorgenommen werden (siehe Abbildung 1.1):

- Freie Eingabe: Es existieren keine Vorgaben, der Text kann beliebig auf die Schreiboberfläche geschrieben werden.
- Liniengeführte Eingabe: Der Text wird anhand vorgegebener Führungslinien ausgerichtet.

- Feldgeführte Eingabe: Der Text wird Buchstabe für Buchstabe in vorgegebene Textfelder eingegeben.

- "Graffiti style[5]" Eingabe: Vom Schreiber muss ein neues Schriftbild erlernt werden, anhand dessen die Einzelzeichenerkennung erfolgt.

Je nach Vorgaben ergeben sich verbesserte Erkennungsvoraussetzungen, wobei der Schreibfluss jedoch teilweise stark beeinträchtigt wird. So entfällt beispielsweise im feldgeführten Fall eine Segmentierung der Schrift vollständig, wobei von einer flüssigen Eingabemöglichkeit nicht mehr die Rede sein kann. "Graffiti style" setzt sogar das Erlernen eines neuen Schriftbildes voraus, welches speziell für die Eingabe auf PDAs entwickelt wurde und dort auch zum Einsatz kommt.

1.3 Schreiberabhängige und schreiberunabhängige Verfahren

Für den Erkennungsprozess müssen dem Computer Einzelzeichen bekannt sein, anhand derer eine Klassifikation durchgeführt werden kann. Zu diesem Zweck werden Merkmale aus den Schriftdaten berechnet, die eine möglichst robuste Wiedererkennung ermöglichen. Je nach Art der zugrundegelegten Daten in der Trainingsphase unterscheidet man zwischen schreiberabhängigen und schreiberunabhängigen Systemen.

Benutzt nur ein Anwender das System so kann ein Training der persönlichen Handschrift zu einer enormen Verbesserung der Erkennungsleistung führen. Anhand dieser ersten Schreibprobe werden vom Rechner Zeichenmodelle für das persönliche Schreibprofil erstellt, die wenig komplex und dem eigenen Schreibstil optimal angepasst sind.

Beim schreiberunabhängigen System werden die Zeichenmodelle aus einer Vielzahl unterschiedlicher Schreibstile und Zeichen erstellt. Ziel dieses Ansatzes ist es, ein möglichst großes Spektrum an Schreibvariationen eines Einzelzeichens abzudecken. Dies führt dazu, dass eine Zeichenklasse zumeist aus mehreren verschiedenen Schreibvariationen *(Allographen* oder auch *Lexeme* genannt) besteht. Sind genügend Allographen für eine Zeichenklasse vorhanden, so ist die Wahrscheinlichkeit groß, dass der passende Allograph vorhanden ist und eine optimale Zuordnung getroffen werden kann. Jedoch verschwimmen die Grenzen zwischen den Zeichenklassen umso stärker, je größer die Anzahl der Allographen wird. Anwendungsgebiete solcher Systeme bilden vor allem Bereiche wo ein Erkennungssystem, ohne Training der persönlichen Schrift, unterschiedliche Handschriften erkennen muss, wie es beispielsweise bei der Briefsortierung im Offline-Verfahren der Fall ist.

[5] Bei der Entwicklung des Palm-PDA fand die Idee, dem Benutzer die Schrift des Computers beizubringen, ihre Umsetzung. Es lernten Millionen Käufer bereitwillig 'Graffiti', das Palm-Alphabet, in dem beispielsweise das 'K' eher einem griechischen ALPHA ähnelt. Für diese Lernarbeit wird der Benutzer belohnt — die Software erkennt die Kringel praktisch fehlerfrei. Dabei bezieht die Erkennungssoftware nicht nur die Form des fertigen Buchstaben in ihre Analyse ein, sondern auch seine Entstehung wird berücksichtigt: So "malt" man ein 'v' wie ein 'u', einziger Unterschied man beginnt rechts statt links.

1.4 Stand der Forschungsarbeiten zur Handschrifterkennung am LMB

Am Institut werden seit mehreren Jahren Forschungsarbeiten im Bereich der Handschrifterkennung unter der Leitung von Claus Bahlmann durchgeführt. Es konnte dabei ein System zur Online-Zeichenerkennung entwickelt werden, welches im internationalen Vergleich eine hervorragende Fehlerratenquote erreicht. Hierbei konnte zum Beispiel auf isoliert geschriebenen Kleinbuchstaben eine Fehlerrate von 9.3% erreicht werden. Ein entwickeltes System (CSDTW) gründet auf einer clusterbasierten, generativen, statistischen Modellierung der Schriftzeichen mithilfe des DTW-Verfahrens. Vor dem eigentlichen Schätzen der statistischen Modelle wird ein Clustering der Trainingsdaten durchgeführt, wodurch eine robuste Parameterschätzung möglich ist. Für eine effiziente Klassifikation werden Viterbi- und Strahlsuchverfahren, wie sie im Zusammenhang mit “Hidden Markov Modellen” bekannte sind, angewandt. Diesbezüglich kommt eine maximum-a-posteriori (MAP) beziehungsweise maximum-likelihood (ML) Klassifikation zum Einsatz, in der die Produktionswahrscheinlichkeit für die in Frage kommenden statistischen Modelle maximiert wird.

1.5 Ziele der Arbeit

Ziel dieser Arbeit ist die Erweiterung der CSDTW-Zeichenerkennung zur Erkennung von Zeichensequenzen beziehungsweise Wörtern. Hierzu müssen neue Vorverarbeitungstechniken entwickelt werden, mit deren Hilfe es möglich ist, robuste Merkmale für die Worterkennung zu generieren. Anschließend müssen entsprechend neue Zeichenmodelle, mit dem bereits bestehenden CSDTW Verfahren, geschätzt werden. Zur Klassifikation von Zeichensequenzen muss das CSDTW Verfahren an den Übergängen von einem Zeichenmodell ins nachfolgende Zeichenmodell erweitert werden. Um die Zahl der Erweiterungshypothesen an den Nahtstellen einschränken zu können ist es notwendig, effiziente Suchalgorithmen zu entwickeln. Eines der wichtigsten Instrumente zur Beschneidung der Worthypothesen stellt das linguistisches Kontextwissen der zu modellierenden Sprache dar. Hier hat sich das Lexikon als ein besonders effizientes Sprachmodell herausgestellt und soll auch in dieser Arbeit zugrunde gelegt werden. Weiterhin soll die Leistungsfähigkeit des Systems anhand von Benchmarks auf dem UNIPEN Datensatz dokumentiert werden. Den Abschluss bildet die Integration in ein Eingabemodul für einen Linux-PDA (Compaq iPAQ) .

1.6 State of the Art in der Worterkennung

Ein direkter Vergleich unterschiedlicher Handschrifterkennungssysteme leidet vor allem an den unterschiedlichen Datensätzen, die bei der Erkennung zugrunde gelegt wurden. So mussten viele Systeme an eigenen, intern erzeugten Datensätzen getestet werden, da keine öffentliche Datenbasis existierte. Das UNIPEN [18] Projekt, welches Ende 1999 Online-Daten öffentlich zugänglich wurde, sollte diesem Sachverhalt Abhilfe verschaffen.

Autor	Methode	Erkennungsrate	Bemerkungen
Nathan et al. [33]	HMM	81.1% ohne Vorgaben an den Schreibstil. (Anzahl der Wörter wurde nicht angegeben)	25 Schreiber schreiberunabhängig Vokabulargröße 21k
Manke et al. [29]	Multi-state TDNN	91.4% auf 2,500 Wörtern	schreiberunabhängig nur kleingeschriebene Worte 40 Schreiber Vokabulargröße 20k
Rigoll et al. [37][26]	HMM/NN hybrid	95% auf 200 Wörtern	schreiberabhängig 3 Schreiber Vokabulargröße 30k
Hu et al. [22]	HMM	91% auf 2,000 Wörtern	16 Schreiber schreiberunabhängig nur kleingeschriebene Worte Vokabulargröße 1,905
Seni et al. [46]	TDNN	62.4% schreiberunabhängig 91.6% schreiberabhängig	9 Schreiber 466 kursiv geschriebene Wörter Vokabulargröße 21k
Hu et al. [20]	Stroke-level HMM	94.5% auf 3,823 Wörtern ohne Vorgaben an den Schreibstil	schreiberunabhängig 18 Schreiber Vokabulargröße 32

Tabelle 1.1: Diese Übersicht, entnommen aus der Arbeit von Connell [14], zeigt eine Zusammenstellung der bis ins Jahre 2000 besten Resultate zum Worterkennungsproblem. Ein direkter Vergleich ist kaum möglich, da jeweils unterschiedliche Testdatensätze zur Auswahl standen.

Einen ausgewählten Überblick von Worterkennungssystemen, der im Jahre 2000 zusammengestellt wurde, findet sich in der Arbeit von Connell [14]. Eine ins deutsche übersetzte Abschrift zeigt Tabelle 1.1. Alle hier vorgestellten Systeme führten Tests anhand von eigenen Datensätzen durch. Im Bericht von Hu et al. [21], der im Jahre 2000 veröffentlicht wurde, wurden nachträglich Tests auf dem UNIPEN Datensatz durchgeführt. Tabelle 1.2 zeigt die Klassifikationsergebnissen dieser Arbeit. Bei den hier erzielten Resultaten stammen alle Wörter von Autoren, deren Schrift nicht beim Training gelernt wurde. Die Erkennungsleistungen sprechen für sich selbst und stellen sicherlich eine der besten Veröffentlichungen im Bereich der Worterkennung auf dem UNIPEN-Datensatz dar.

Vokabulargröße	Anzahl Testbeispiele	Erkennungsrate
500	1685	91.8%
1000	2447	90.5%
5000	2447	83.2%
10000	2447	79.8%
20000	2447	76.3%

Tabelle 1.2: Erkennungsresultate entnommen aus der Arbeit von Hu et al. [21], welche auf dem öffentlich zugänglichen UNIPEN Datensatz devtest_r01_v02 durchgeführt wurden. Die Testbeispiele wurden von 100 Schreibern zusammengestellt, deren Schrift nicht trainiert wurde.

1.7 Kapitelübersicht

Zuerst erfolgt eine genauere Spezifikation des UNIPEN Datensatzes, der sowohl für das Training als auch für die Klassifikation benutzt wird. Anschließend wird in Kapitel 3 die Vorverarbeitung der Online-Daten beschrieben. In Kapitel 4 erfolgt ein Einschub zu Verfahren, die es ermöglichen, die Wortsegmentierung vor der Erkennungsphase zu umgehen. Danach werden die Merkmale, welche zur Wiedererkennung eines Zeichens berechnet werden, vorgestellt. In Kapitel 6 erfolgt eine Übersicht über das DTW-Verfahren. Es wird in die Grundlagen des Verfahrens eingeführt, um schließlich das CSDTW System erläutern zu können. Ist dieses System bekannt, wird auf die Erweiterung im Hinblick auf die Worterkennung in Kapitel 7 eingegangen. Es wird der Aufbau des Lexikons beschrieben und weitere Suchstrategien werden vorgestellt. Im Kapitel 8 findet eine Untersuchung zur Leistungsfähigkeit des Systems anhand des UNIPEN Datensatzes train_r01_v07 statt. Kapitel 9 beschreibt kurz die Portierung auf einen iPQA PDA. Zum Abschluss erfolgt in Kapitel10 eine Zusammenfassung der Arbeit und ein Ausblick auf weiterführende Forschungsarbeiten.

KAPITEL 2

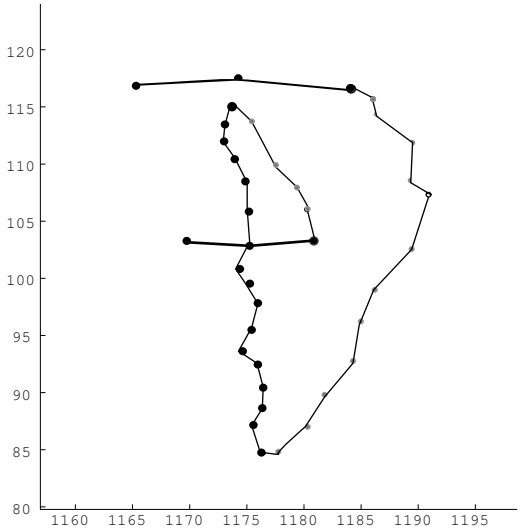
Trainings- und Testdatensatz

Um auf genügend große Trainings- und Testdatensätze von Online-Handschriften zugreifen zu können, wurden die UNIPEN [18] Benchmark Daten train_r01_v07 verwendet. Der UNIPEN Datensatz wurde speziell für die Erforschung und Entwicklung von Online-Handschriftanwendungen entworfen. Die gesammelten Online-Handschriften in der UNIPEN Datenbank wurden von verschiedenen Personen mit unterschiedlicher Hardware erfasst und in einem vorgegebenen Format abgespeichert. Er bietet außerdem die Möglichkeit verschiedene Online-Handschrifterkennungssysteme untereinander zu vergleichen, was bei der Verwendung von selbst erstellten Handschriftdaten nicht möglich ist.

Im UNIPEN-Format bezeichnet eine Hierarchie die Trennung zwischen Zeichen-, Wort- und Textebene. Zu den einzelnen Ebenen werden zusammengefasst:

1. SCHRIFTZEICHEN: Jedes einzelne Zeichen des ASCII-Alphabets kann hier auftreten.
   Beispiel: 'A', 'b', 'O', '2', ' ', '@'
2. WÖRTER: Bezeichnet ein Sequenz aus einem oder mehreren Schriftzeichen (ohne Leerzeichen) aus dem ASCII-Alphabet.
   Beispiel: 'A', 'b', '0', 'The', 'AC/DC', '{a\|b}'
3. TEXT: Kein oder mehrere Schriftzeichen aus dem ASCII-Alphabet (ohne führendes/endendes Leerzeichen), kein Auftreten mehrerer aufeinander folgender Leerzeichen
   Beispiel: ' ', 'A', 'that nothing more happened'

Ein Element innerhalb einer Hierarchieebene wird als *Segment* bezeichnet. Beim Aufzeichnen der Stiftbewegung werden zwei verschiedenen Zustände der Stiftposition in Bezug zur Schreiboberfläche unterschieden:



COORD_COMP	[3x2]	[12x2]	[17x2]	[4x2]	[2x2]
PEN_COMP	[.PEN_DOWN]	[.PEN_UP]	[.PEN_DOWN]	[.PEN_UP]	[.PEN_DOWN]

Abbildung 2.1: Aufgezeichnetes Datensegment mit [.PEN_DOWN]- (dicke Abtastpunkte) und [.PEN_UP]-Komponenten (dünne Abtastpunkte). Zur leichteren Erkennung des Buchstabens wurden die Abtastpunkte mittels Geraden verbunden.

- '.PEN_DOWN': Berührt der Stift die Schreiboberfläche, werden die diskreten Ortskoordinaten als '.PEN_DOWN'-Einheiten abgespeichert.

- '.PEN_UP': Verlässt der Stift die Schreiboberfläche, werden die diskreten Ortskoordinaten als '.PEN_UP'-Einheiten abgespeichert. Nicht bei allen Eingabegeräten ist es auch im '.PEN_UP'-Zustand möglich diskrete Werte abzutasten. In diesen Fällen wird eine leere Einheit erzeugt.

Eine solche Einheit von '.PEN_UP'- und '.PEN_DOWN'-Daten, wird in der UNIPEN Terminologie als *Komponente* bezeichnet. Abbildung 2.1 zeigt ein aufgezeichnetes Datensegment aus dem UNIPEN Datensatz.

Die Benchmark Datensätze sind in 11 Kategorien unterteilt:

Kategorie	Anzahl Segmente	Beschreibung
1a	15953	Zahlen
1b	28069	Großbuchstaben
1c	61351	Kleinbuchstaben
1d	17286	Symbole (Punkt, Strichpunkt, Semikolon usw)
2	122628	Groß- und Kleinbuchstaben
3	67352	Buchstaben im Kontext von Wörtern oder Text
4	0	Wörter ohne Zahlen und Symbole
5	0	Wörter inklusive aller Zeichensätze
6	75529	kursiv oder gemischt geschriebene Wörter ohne Zahlen und Symbole
7	85213	Wörter jeder Stilrichtung mit Zahlen und Symbolen
8	14544	Text: freier Text (mindestens zwei Worte) alle Zeichen

Für diese Arbeit wurden für das Training der Buchstabenmodelle die Kategorie 1a, 1b ,1c, 2 und 3 verwendet. Für die Klassifikation von Wörtern kam die Kategorie 6 zum Einsatz. Ein Problem des UNIPEN Datensatzes ist die zum Teil fehlerhafte Beschriftung der Segmente, die für die automatische Verifikation der Klassifikationsergebnisse benötigt werden. Es wurden jedoch weder für die Trainings- noch für die Testdaten Korrekturen vorgenommen.

KAPITEL 3

Vorverarbeitung

Die Vorverarbeitung bezeichnet den ersten Verarbeitungsschritt nach der Datenerfassung. Sie soll Störeinflüsse entfernen, schreiberabhängige Freiheitsgrade reduzieren und erste Informationen aus dem Datensatz herausfiltern. Fehlinterpretationen auf dieser Ebene führen bereits frühzeitig zu einer Reduktion der Erkennungsleistung. Dies hängt damit zusammen, dass einige Merkmale, wie sie in Kapitel 5 diskutiert werden, direkt von der Güte der Vorverarbeitung abhängig sind. Somit nimmt die Vorverarbeitung einen ebenso wichtigen wie auch diffizilen Teil am Gesamtsystem ein. Zu den hier betrachteten Vorverarbeitungsschritten zählen:

1. Eliminierung von Abtastduplikaten (engl. *duplicate elimination*)
2. Glättung und Neuabtastung (engl. *smoothing and resampling*)
3. Korrektur der Zeilenneigung (engl. *skew correction*)
4. Korrektur der Schriftneigung (engl. *slant correction*)
5. Referenzlinienschätzung (engl. *boundary lines estimation*)
6. Skalierung der Schriftgröße (engl. *scaling*) und äquidistante Neuabtastung

Nach Durchführung der Vorverarbeitung ist die Grundlage für ein schreiberunabhängiges Erkennungssystem geschaffen. Es erlaubt die Klassifikation mittels eines weniger komplexen Referenzdatensatzes, was implizit zu einer Reduktion der Erkennungszeit führt.

Sei im Folgenden $P = \{p_1, ..., p_N\}$ die Konkatenation der '.PEN_DOWN'-Komponenten eines Segments. Hierbei bezeichne $p_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ für $i \in [1, ..., N]$ die Ortskoordinate des i-ten Abtastpunktes. Diese Ortskoordinaten bilden den Ausgangspunkt für die Vorverarbeitung.

Jeder Verarbeitungsschritt führt zu einer Modifikation des Eingabesegments, wodurch sich eine Verarbeitungskette ergibt, in der Schritt für Schritt die korrigierten Daten weiterverarbeitet werden. Hierbei stellen die Ausgabedaten den Ausgangspunkt für den jeweils nachfolgenden Arbeitsschritt dar. Um diesem Sachverhalt in der Nomenklatur Ausdruck zu verleihen, erfolgt eine systematische Bezeichnungsvergabe des Übergabesegments. So erhalten die Eingabedaten die Erweiterung 'in' beziehungsweise die Ausgabedaten die Erweiterung 'out'. Diese Nomenklatur hat den Vorteil, dass nicht für jeden Bearbeitungsschritt eine neue Variablenbezeichnung eingeführt werden muss.

3.1 Eliminierung von Abtastduplikaten

Das Eingabegerät erfasst mit einer konstanten Abtastfrequenz die Schreibbewegungen. Verharrt der Schreiber innerhalb eines Abtastintervalls an ein und derselben Stelle entsteht ein Duplikat des vorherigen Abtastpunktes. Längere statische Schreibphasen führen somit zu einer Folge von Abtastkoordinaten, die sich außer in der zeitlichen Dimension nicht voneinander unterscheiden.

$$p_i^{\text{in}} = p_{i+1}^{\text{in}} = ... = p_{i+d}^{\text{in}} \text{ für } i, d \in [1, ..., N^{\text{in}}] \text{ und } d > i$$

Bei der Berechnung von Merkmalen ohne Neuabtastung, wie sie in Kapitel 5 noch ausführlich beschrieben werden, führen Duplikate zu Singularitäten in den Ableitungen. Aus diesem Grund werden die Duplikate $p_{i+1}^{\text{in}} = ... = p_{i+d}^{\text{in}}$ aus dem Datensatz entfernt. Hierdurch ergebt sich das Ausgangssegment durch:

$$P^{\text{out}} = \left\{ p_1^{\text{in}}, ..., p_{N^{\text{out}}}^{\text{in}} \right\} \quad \text{mit } N^{\text{out}} \leq N^{\text{in}}$$

Würde man den Schreiber dazu anhalten, nach Abschluss eines jeden geschriebenen Buchstabens eine kurze Zeit mit dem Stift zu verharren, so könnte man Abtastduplikate durchaus sinnvoll zur Auffindung von Buchstabengrenzen im geschriebenen Wort einsetzen (Wortsegmentierung). Dieser Ansatz wurde in frühen Spracherkennungssystemen durchaus verfolgt. Hierbei wurde der Sprecher angewiesen nach jedem Wort eine Sprechpause einzulegen, wodurch die Erkennung von Wortgrenzen erleichtert werden konnte. Die Akzeptanz im täglichen Umgang war jedoch eher gering. Ein Schreiber/Sprecher unterwirft sich nur ungern solchen Restriktionen. Die Entwicklung führte hin zu kontinuierlichen Spracherkennungssystemen, welche eine natürliche Sprechweise unterstützen.

3.2 Glättung und Neuabtastung

Betrachtet man die abgetastete Handschrift beim Online-Verfahren, so fällt auf, dass die digitalisierte Schrift, je nach Eingabegerät, ein mehr oder minder starkes Rauschen aufweist. Typische Ursachen hierfür sind:

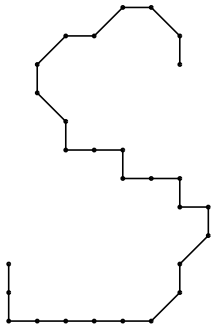


Abbildung 3.1: Treppenmuster bedingt durch zu geringe Rasterauflösung des Eingabegerätes

- Unzureichende Rasterauflösung:
  Verbindet man die Abtastpunkte miteinander, so ergibt sich eine typisches Treppenmuster beim Zeichnen einer Diagonalen. Abbildung 3.1 zeigt diesen Sachverhalt am Buchstaben 'S'.
- Geringe Abtastfrequenzen
- Mangelnde Fähigkeit des Benutzers mit dem Eingabegerät umzugehen

Durch eine konstante Abtastfrequenz enthält der Datensatz zusätzliche Informationen über die Schreibgeschwindigkeit. So ist der Datensatz charakterisiert durch einen großen Abstand benachbarter Abtastpunkte bei schneller Stiftbewegung und nah beieinanderliegender Abtastpunkte bei langsamer Stiftführung. Diese Eigenschaft ist in vielen Fällen rein schreiberabhängiger Natur und kann nur bedingt zur Informationsextration in einem schreiberunabhängigen Ansatz genutzt werden. In [43] wird sie zum Bespiel als zusätzliche Informationsquelle zur Auffindung von Segmentierungsgrenzen genutzt. Folgende Idee wird zugrunde gelegt: In Phasen hoher Schreibgeschwindigkeit steigt die Wahrscheinlichkeit potenzieller Segmentierungsgrenzen eines Buchstabens. Um diese Information in das Erkennungssystem zu integrieren, wurde ein Merkmal (engl. *speed coordinate*) erzeugt, welches die momentane Schreibgeschwindigkeit berechnet.

Vorwegnehmend kann jedoch gesagt werden, dass diese Technik in der vorliegenden Arbeit nicht zum Einsatz kommt, da zur Bearbeitungszeit keine passenden Trainingsdaten zur Verfügung standen, an denen dieses Merkmal hätte trainiert werden können. Um Invarianz gegenüber solchen Einflussfaktoren zu erreichen, wird der Datensatz einer Glättung mit anschließender Neuabtastung unterzogen. Bei der Glättung werden die Abtastpunkte durch *Splines*[1] approximiert. Die Neuabtastung des Schriftzuges zu diesem Bearbeitungszeitpunkt wird hauptsächlich im Hinblick auf die nachfolgenden Verarbeitungsschritte durchgeführt. Sie arbeiten anhand von

[1] Der Begriff Spline bezeichnete ursprünglich flexible Metallstreifen mit deren Hilfe technische Zeichner die Flächen von Flugzeugen, Autos und Schiffen konstruierten. Am Ende der Metallstreifen wurden Gewichte befestigt, um einen bestimmten Krümmungsverlauf zu formen. Solange sie nicht extrem belastet wurden, waren die Metall-Splines von zweiter Ordnung stetig.

Histogrammanalysen, die zur robusten Interpretation genügend Abtastdaten benötigen. Unterschiedliche Schreibgeschwindigkeiten würden bei diesen Verfahren sogar zu Fehlinterpretationen führen. Um die Daten einheitlich mit einer festen Länge abtasten zu können, erfolgt als abschließende Vorverarbeitung eine zweite Neuabtastung nach der Schriftgrößennormierung (siehe 3.6) . Sie stellt den eigentlichen Ausgangspunkt für die Merkmalsextraktion dar.

Zur Glättung werden *kubische B-Splines* verwendet, die approximativ nach Bogenlänge umparametrisiert und anschließend äquidistant neuabgetastet werden. Genauer beschrieben ist dieses Verfahren in der Studienarbeit [47] zur Vorverarbeitung von Einzelbuchstaben. Rein optisch können somit gute bis sehr gute Qualitätsverbesserungen erzielt werden. Jedoch täuschte die optische Verbesserung über eine Steigerung der Klassifikationsleistung hinweg, wie es in dieser Arbeit gezeigt werden konnte. So führt eine Glättung der Abtastdaten unweigerlich zu Informationsverlusten, welche die Erkennungsleistung beeinträchtigen indem vorherige Diskriminierungseigenschaften durch Angleichen ähnlicher Buchstabenpaare verloren gehen.

Im Kontext der Worterkennung kann jedoch auf diesen Vorverarbeitungsschritt nicht verzichtet werden. Aus diesem Grund wird nur eine leichte Glättung bei der Vorverarbeitung durchgeführt. Damit anschließend eine äquidistante Neuabtastung der Splinekurve erfolgen kann.

Zusammenfassend führt die Durchführung dieses Verarbeitungsschrittes hin zu einem Datensatz, in dem folgende gerätespezifische und schreibspezifische Varianzen eliminiert werden konnten:

Gerätespezifisch:	Schreibspezifisch:
Rasterauflösung	Schreibgeschwindigkeit
Abtastfrequenz	

3.3 Korrektur der Zeilenneigung

Unter der Korrektur der Zeilenneigung versteht man das horizontale Ausrichten des Schriftzuges. Findet die Texteingabe anhand vorgegebener Führungslinien statt, ist eine Zeilenneigungskorrektur meist nicht notwendig. Bei der Erstellung des UNIPEN Datensatzes wurden diesbezüglich keine allgemeinen Vorgaben festgelegt. Weshalb bei der Erkennung vom allgemeinsten Fall, der freien Eingabe ausgegangen werden kann und infolgedessen eine Korrektur der Zeilenneigung durchgeführt wird.

Für diesen Verarbeitungsschritt finden sich in der Literatur verschiedene Ansätze. Zum einen der Ansatz mittels linearer Regressionsgeraden, wie er beispielsweise von Caesar et al. [11], Seni et al. [46] beschreiben ist: hierbei approximieren die berechneten Regressionsgeraden die oberen und unteren Wendepunkte der Trajektorie. Die Steigungen der Geraden geben das Maß für den Winkel der Zeilenneigung wieder. Zum anderen der Ansatz, wie er zum Beispiel von Kosmala [25] beschreiben ist, indem eine Korrektur mittels so genannter Richtungshistogramme durchgeführt wird. Hierbei wird der abgetastete Schriftzug in kleinen Drehwinkeln um einen Fixpunkt rotiert, während in jeder Drehstellung das zugehörige y-Histogramm berechnet wird.

In dieser Arbeit wurde eine Korrektur anhand des y-Histogramms durchgeführt. Der Ansatz wurde favorisiert, weil er die Möglichkeit bietet, durch geringe Modifikationen eine Schriftneigungswinkelkorrektur und Referenzlinienschätzung durchzuführen. Zur Auswertung der Histogramme sind wiederum unterschiedliche Methoden denkbar. Ein einfacher und nahe liegender Ansatz ist die Berechnung der Entropie wie er in der Arbeit von Kosmala [25] beschrieben ist. In Kavallieratou et al. [24] erfolgt die Auswertung anhand der aufgezeichneten Maxima der *Wigner Ville Verteilung* (engl. *Wigner Ville Distribution, WVD*) je Drehwinkel. In beiden Fällen bestimmt das globale Maxima der über den Drehwinkeln aufgezeichneten Werte den Korrekturwinkel θ_0, um welchen der Datensatz korrigiert werden muss. Es wurden beide Methoden zur Auswertung der Histogramme implementiert und einander gegenübergestellt (siehe hierzu Abschnitt 3.3.4) .

3.3.1 Verfahren

Zur Berechnung des horizontalen Verteilungshistogramms für On-Line Daten, auch *Projektionsprofil* genannt, wird die *Projektionsebene* in eine feste Anzahl horizontaler Behälter (*Bins*) gleicher Breite unterteilt. Die Größe der Projektionsebene wird bestimmt durch die Extrema $\hat{y}_{\max}$, $\hat{y}_{\min}$, $\hat{x}_{\max}$ und $\hat{x}_{\min}$, die bei der Rotation des Schriftzuges im gesamten Drehbereich $\mathbb{D}_\theta$ entstehen können. Sei $\hat{p}_i^{\text{in}} = p_i^{\text{in}} - p_1^{\text{in}}$ die Verschiebung der Abtastkoordinaten in den Ursprung bezüglich der ersten Abtastkoordinate. So wird das Maxima wie folgt definiert

$$\hat{x}_{\max} = \max_{\theta \in \mathbb{D}_\theta} \left(\max_{1 \le i \le N^{\text{in}}} (\cos(\theta) \cdot \hat{x}_i - \sin(\theta) \cdot \hat{y}_i) \right)$$

$$\hat{y}_{\max} = \max_{\theta \in \mathbb{D}_\theta} \left(\max_{1 \le i \le N^{\text{in}}} (\sin(\theta) \cdot \hat{x}_i + \cos(\theta) \cdot \hat{y}_i) \right)$$

Entsprechendes gilt für das Minima, wobei hier die minimalen Extremwerte betrachtet werden. Das Profil ergibt sich aus der Anzahl an Abtastpunkten, die jedes einzelne Bin enthält. Genauer gilt:

Sei $\hat{P}$ das verschobene Segment der Abtastkoordinaten, $B = \{b_1, ..., b_L\}$ eine Menge von Bins mit gleicher Länge $L = \hat{x}_{\max} - \hat{x}_{\min}$ und Höhe $W = \frac{\hat{y}_{\max} - \hat{y}_{\min}}{|B|}$, so berechnet sich das Projektionsprofil in y-Richtung durch:

$$\mathsf{P}^y(l) = \left| \left\{ i \in \left\{ 1, .., N^{\text{in}} \right\}, \hat{y}_{\min} + (l-1) \cdot W \le \hat{y}_i < \hat{y}_{\min} + l \cdot W \right\} \right| \qquad \text{für } 1 \le l \le L$$

Betrachtet man das Profil bei unterschiedlichen Drehwinkeln θ, so lässt sich folgende Interpretation formulieren:

Bei horizontaler Ausrichtung zeigt das Profil eine kleine Streuung mit deutlicher Maxima-Bildung im Zentrum der Verteilung, bei zunehmender Schräglage steigt die Streuung und die Verteilung wird flacher, als Folge entstehen verschmierte Maxima.

Diese Interpretation ist natürlich nur dann korrekt, wenn genügend Abtastpunkte vorliegen und das Wort über eine hinreichende Wortlänge verfügt. Es sollte klar sein, dass ein einzelner Buchstabe wie zum Beispiel das 'i' bei dieser Interpretation um 90° gedreht werden würde. Eine Rückweisung wird somit durch zwei Faktoren bestimmt:

1. die Länge der Abtastsequenz und
2. die Eindeutigkeit zur Bestimmung des Verteilungsmaximas, über den Verlauf des gesamten Drehbereichs betrachtet.

Unter der Annahme, dass die UNIPEN Daten keine zu große Zeilenneigung aufweisen, wurde der Rotationsradius $\theta = \pm 45^\circ$ beschränkt. Dies hat zur Folge, dass weniger Rotationshistogramme bei einer gewählten Intervallbreite von 1° berechnet werden müssen. Als Drehpunkt (Fixpunkt) dient der Koordinatenursprung. Zur Auffindung des Verteilungsmaximas wurden nachfolgende Verfahren implementiert und einander gegenübergestellt:

3.3.2 Auswertung des Projektionsprofils mittels Entropie

Eine einfache Methode, das Profil zu bewerten, ist die Entropie. Sei die relative Häufigkeit der einzelnen Bin-Inhalte $\overline{\mathsf{P}^y}(l) = \frac{\mathsf{P}^y(l)}{N}$ für $1 \leq l \leq L$ so ergibt sich die Entropie des y-Profils aus:

$$E^y = -\sum_{l=1}^{|B|} \overline{\mathsf{P}^y}(l) \cdot \log\left(\overline{\mathsf{P}^y}(l)\right)$$

Um im Weiteren eine einheitliche Nomenklatur verwenden zu können, wurde die Entropie negiert $-E^y$. Somit wird aus der Minima-Bestimmung eine Maxima-Bestimmung. Zeichnet man die Entropie über den Rotationsverlauf hinweg auf, so bestimmt der dem globalen Maxima zugehörige Drehwinkel θ_0 den zu korrigierenden Neigungswinkel. In der Implementierung wird vor der Berechnung des Maximas ein Glättung des Entropieverlaufs durchgeführt, um eventuelles Rauschen zu entfernen.

3.3.3 Auswertung des Projektionsprofils mittels Wigner Ville Verteilung

Kavallieratou et al. [24] führen eine Bewertung des Projektionsprofils mittels der Wigner Ville Verteilung durch. Die Wigner Ville Verteilung berechnet zu jedem Zeitpunkt das zeitabhängige Energiedichtespektrum des Signals $s(t)$. Die Verteilung ist definiert durch:

$$W_s(t,f) = \int_{-\infty}^{+\infty} s\left(t + \frac{\tau}{2}\right) s^*\left(t - \frac{\tau}{2}\right) \cdot \exp^{-j2\pi f\tau}\, d\tau$$

Zur Auswertung des Projektionsprofils P^y wird das zeitdiskrete komplexe Signal $s(t)$ durch eine Hilbert-Transformation aus den Projektionswerten der n-ten Bin wie folgt berechnet:

$$s(l) = \mathsf{P}^y(l) + j \cdot H[\mathsf{P}^y(l)]$$

mit

$$H[\mathsf{P}^y(l)] = \sum_{m=-\infty}^{+\infty} \frac{2 \cdot \mathsf{P}^y(l+m)}{m \cdot \pi} \qquad \text{mit } m \in 2 \cdot \mathbb{R} + 1$$

Der wesentliche Unterschied zur Entropie liegt in der zusätzlichen (lokalen) zeitlichen Gewichtung. Bezogen auf das Projektionsprofil werden Ausreißer, die außerhalb der Hauptverteilungsdichte liegen, schwächer bewertet. Zeichnet man den maximalen Funktionswert über den Verlauf der Drehung hinweg auf, so bestimmt wiederum der dem globalen Maxima zugeordnete Drehwinkel θ_0 den gesuchten Korrekturwinkel. Auch hier wurde eine Glättung des Kurvenverlaufs vorgenommen, um ein vorhandenes Rauschen herauszufiltern. Abbildung 3.2 veranschaulicht diesen Sachverhalt am Beispiel des Wortes 'Freiburg'. Der Schriftzug befindet sich in einer Ausgangsstellung von etwa 36° zur Horizontalen. Relativ zu dieser Ausgangsstellung wurden innerhalb des Rotationsbereichs von $\pm 75^\circ$, in Rotationsintervallen von 1°, die entsprechenden Projektionsprofile berechnet. Es ist eine deutliche Maxima-Ausprägung des Profils im Bereich um -36° zu erkennen (zweite Zeile der Projektionsabbildungen), während in den umliegenden Bereichen eine Abschwächung des Profils zu beobachten ist. Die Berwertungsfunktionen, welche über den gesamten Rotationsbereich aufgezeichnet wurden, sind in Abbildung 3.3 zu sehen. Beide Funktionen zeigen eine deutliche Maximabildung im relevanten Bereich.

3.3.4 Bewertung

Um einen Vergleich der Bewertungsfunktionen durchführen zu können, wurden folgende zwei Datensätze zugrunde gelegt:

Datensatz	#Worte	mittlere Wortlänge	min Wortlänge	max Wortlänge
A	300	5.5	3	12
B	300	8.22	3	16

Dabei wurde Datensatz A zufällig aus einer Teilmenge von 700 Worten, die von sieben verschiedenen Schreibern erzeugt wurden, zusammengestellt. Dieser Datensatz wurde durch Mitarbeiter des Instituts erstellt. Datensatz B stammt aus einer zufälligen Auswahl von 628 Städtenamen, welche von vier unterschiedlichen Schreibern erzeugt wurden und dem UNIPEN-Datensatz entnommen worden sind. Beide Zusammenstellungen wurden jeweils bei der Erstellung an Führungslinien ausgerichtet. Trotz Führungslinien waren Korrekturen im Bereich von ± 5 Grad, bezüglich dem Original, durchaus akzeptabel und wurden bei der Bewertung als korrekt anerkannt. Dies lässt sich damit begründen, dass eine eindeutige Korrektur bei der Neigungswinkelkorrektur nicht existiert und somit eine absolute Genauigkeit nicht erwartet werden kann. Jede

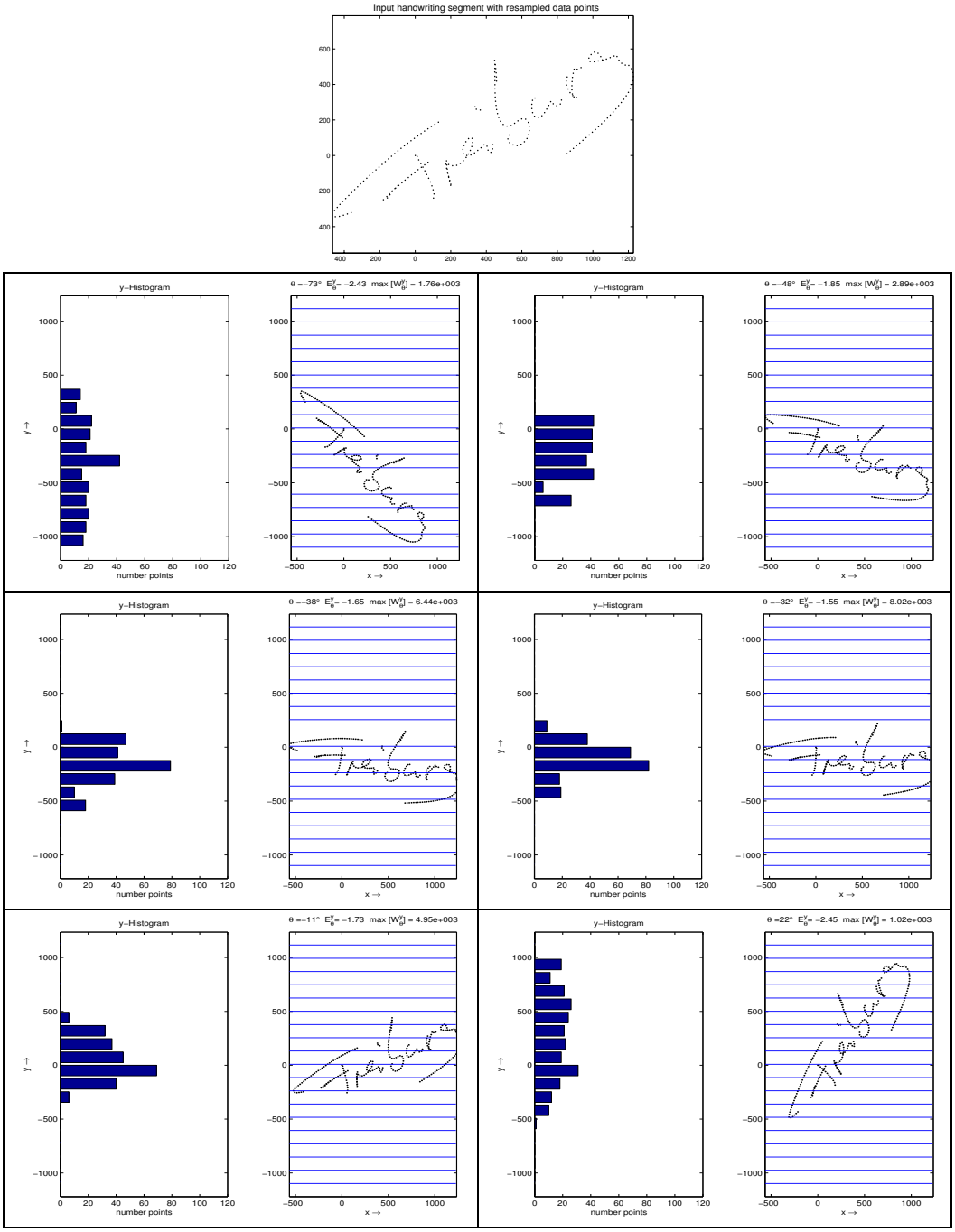


Abbildung 3.2: Das Original (oberste Abbildung) müsste um etwa -36° zur x-Achse gedreht werden, damit es horizontal ausgerichtet wäre. Von links oben nach rechts unten sind 6 Momentaufnahmen bei einem Rotationbereich von $\pm 75°$ zu sehen. Als Drehpunkt wurde der erste Abtastpunkt genommen.

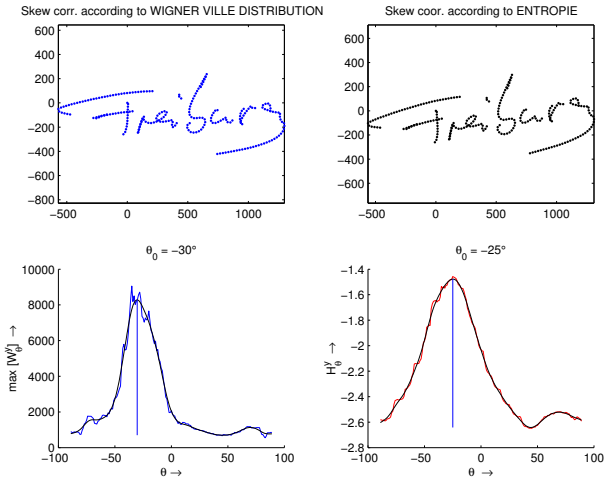


Abbildung 3.3: Ergebnisse des Wigner Ville und Entropie Auswerteverfahrens zur Schriftneigungskorrektur. Beide Verfahren zeigen eine deutliche Maximabildung im relevanten Bereich.

Datensatz A						
Mindestanzahl Abtastpunkte		ohne Neuabtastung			Neuabtastung	
		WVD	ENTROPIE		WVD	ENTROPIE
$c_{\min}$	#Rück	$E_{\text{Rück}}$	$E_{\text{Rück}}$	#Rück	$E_{\text{Rück}}$	$E_{\text{Rück}}$
1	0	29.67	25.00	0	36.00	40.00
50	6	28.91	24.83	0	36.00	40.00
75	101	18.59	18.09	53	19.43	32.39
100	176	6.45	8.87	114	9.14	26.88
125	212	3.41	6.82	180	2.50	26.67

Datensatz B						
Mindestanzahl Abtastpunkte		ohne Neuabtastung			Neuabtastung	
		WVD	ENTROPIE		WVD	ENTROPIE
$c_{\min}$	#Rück	$E_{\text{Rück}}$	$E_{\text{Rück}}$	#Rück	$E_{\text{Rück}}$	$E_{\text{Rück}}$
1	0	6.00	5.33	0	4.67	12.33
150	14	4.89	4.55	0	4.67	12.33
250	68	3.88	5.17	17	4.59	12.01
300	84	3.70	5.56	41	3.47	11.58

Tabelle 3.2: Vergleich der beiden Verfahren zur Zeilenneigungswinkelkorrektur.

Korrektur, die größer dieser Toleranz vorgenommen wurde, wurde jedoch als Fehlkorrektur interpretiert. Da diese Verfahren auf genügend Abtastpunkte (Wortlänge) angewiesen sind, wurde eine Mindestwortlänge von drei Buchstaben gefordert. Zusätzlich wurden Bewertungen anhand unterschiedlicher Rückweisungsschwellwerte $c_{\min}$ durchgeführt, was sich gerade bei Datensatz A als wichtig herausstellte. Hierduch war es unter anderem möglich einen Schwellwert für das Gesamtsystem zu definieren. Der Gesamtrotationsradius wurde im Bereich von $\pm 45^\circ$ gewählt. Zur Bewertung der Ergebnisse wurde folgende Fehlerrate $E_{\text{Rück}}$ berechnet:

$$E_{\text{Rück}} = \frac{\text{\#Fehlkorrekturen}}{\text{\#Worte} - \text{\#Rückweisungen}} \cdot 100$$

Tabelle 3.2 enthält die Auswertungsergebnisse unter zusätzlicher Angabe der Anzahl zurückgewiesener Worte (#Rück). Folgende Erkenntnisse ergaben sich:

Während die Entropie mit kleinen Wortlängen besser zurecht kommt und ein Neuabtasten der Schriftkurve sich eher negativ auswirkt, kommen dem WVD Verfahren viele Abtastpunkte und somit auch die Neuabtastung entgegen. Darüber hinaus spielt die Wortlänge bei beiden Verfahren eine entscheidende Rolle. So konnten im Datensatz B weitaus bessere Werte erzielt werden, als dies in Datensatz A der Fall war. Ein Grund für das schlechte Abschneiden der Neuabtastung bei kurzen Wörtern ist die erhöhte Abtastdichte in Bewegungsrichtung. Wurden zuvor zeichenverbindende Striche oder auch t-Striche zumeist mit einer schnellen Stiftführung und somit nur wenigen Abtastpunkten erfasst, so kommt es durch die äquidistante Neuabtastung zu einer höheren Gewichtung dieser Zeichensegmente was sich bei kurzen Wörtern negativ bemerkbar macht. Die WVD kann mit diesem Sachverhalt besser umgehen als dies bei der Entropie der Fall ist. Dies liegt daran, dass beim Bestimmen des Maximas die Nachbarschaftsregionen mitberücksichtigt werden. Desweiteren konnten natürlich durch die Neuabtastung die Anzahl der zurückgewiesenen Worte weiter verringert werden. Somit kommt bei der Schriftwinkelkorrektur das Wigner Ville Vefahren zum Einsatz.

3.3.5 Berechnung des neuen Datensegments

Die korregierten Daten berechnen sich mithilfe des Korrekturwinkels θ_0 wie folgt:

$$p_i^{\text{out}} = \begin{pmatrix} \cos(\theta_0) & -\sin(\theta_0) \\ \sin(\theta_0) & \cos(\theta_0) \end{pmatrix} \cdot p_i^{\text{in}} \quad \text{für } i \in [1..N^{\text{in}}] \text{ und } \theta_0 \in \mathbb{D}_\theta$$

3.4 Korrektur der Schriftneigung

Das Schriftbild eines Schreibers ist zudem geprägt durch die Schriftneigung. Unter der Schriftneigung versteht man das Schrägstellen einzelner oder aller Zeichen innerhalb eines Wortes. Die Neigungsrichtung lässt sich hauptsächlich an Zeichen mit *Ober-* beziehungsweise *Unterlängen* feststellen, diese Zeichen besitzen ausgeprägte vertikale Stiftbewegungen. In den meisten Fällen ist eine Hauptneigungsrichtung im Schriftbild zu erkennen, jedoch gibt es auch Ausnahmen. So

ist zu bemerken, dass das Schriftbild besonders häufig bei Linkshändern mehrere Neigungsrichtungen aufweist. Zudem ist bei zu kurzen Datensequenzen keine eindeutige Korrektur möglich.

Betrachtet man die Generierung des Referenzdatensatzes genauer, so ist die Information der Schriftneigung implizit in den geschätzten Modellen vorhanden. Geschätzte Varianzen in den Modellparametern können durchaus kleine Schriftneigungen ausgleichen und bedürfen keiner Korrekturen. Für größere Neigungswinkel sind sie jedoch ungeeignet, da Modelle mit zu großen Varianzen in den Modellparametern schnell zu allgemeingültigen Klassengebieten werden. Die Klassenzugehörigkeit einzelner Modelle verschwimmt und sie eignen sich nicht mehr zur Klassifikation. Ziel ist es, bei der Korrektur möglichst große Neigungen zu erkennen und diese zuverlässig zu beseitigen. Hierzu kann wiederum das Projektionsverfahren, wie es bereits zur Zeilenneigungskorrektur eingesetzt werden konnte, genutzt werden.

3.4.1 Verfahren

Zuerst wird der Datensatz auf die x-Achse verschoben, womit diese als Scherachse benutzt werden kann.

$$\hat{y}_i = y_i^{\text{in}} - y_{\min}^{\text{in}} \quad \text{mit } i \in [1..N^{\text{in}}]$$

Danach werden die Koordinatenpunkte mittels unterschiedlicher Scherungswinkel geschert, wobei in jeder neuen Scherstellung das zugehörige Projektionsprofil entlang der Vertikalen auf die Horizontale berechnet wird. Folgende Interpretation des Profils wird zugrunde gelegt:

Bei vertikaler Ausrichtung bilden Schriftzeichen mit Oberlänge/Unterläge deutliche Maxima im Projektionsprofil (*Peaks*). Bei zunehmender Schriftneigung überdecken diese Schriftzeichen teilweise oder ganz benachbarte Buchstabenregionen, was zu einem flacheren Projektionsprofil mit weniger stark ausgeprägten Maxima führt.

In Kavallieratou et al. [24] erfolgt die Auswertung der Projektionsprofile wiederum durch die Wigner Ville Verteilung diesmal als Funktion des Scherungswinkels. Auch hier bestimmt der maximale Wert der Verteilungsfunktion den Scherungswinkel ϕ_0.

Bei der Projektionsanalyse wird angenommen, dass eine Hauptneigungsrichtung vorliegt. Liegen mehrere Neigungswinkelrichtungen vor werden diese durch die einheitliche Scherrichtungskorrektur nicht berücksichtigt. In der Praxis zeigt es sich, dass das Projektionsprofil häufig keine eindeutigen Maxima besitzt. So existieren nicht nur für einen Scherwinkel typische Maxima Peaks, sondern für mehrere, meist weit auseinander liegende Scherwinkel. Dies sind genau so viele, wie unterschiedliche Neigungsrichtungen im Wort vorhanden sind.

Ist der Scherungswinkel ϕ_0 bekannt, kann eine Korrektur der horizontalen Komponenten durchgeführt werden, während die vertikalen Komponenten invariant bleiben.

$$p_i^{\text{out}} = \begin{pmatrix} x_i^{\text{in}} - \hat{y}_i \cdot \tan(\phi_0) \\ \hat{y}_i \end{pmatrix} \quad \text{für } i \in [1..N^{\text{in}}] \text{ und } \theta_0 \in \mathbb{D}_\theta$$

Abbildung 3.4 zeigt deutlich die zwei Peaks, im ersten Projektionsprofil in der dritten Zeile, welche durch die Oberlänge des Buchstabens 'H' und 'l' zustande kommen. Abbildung 3.5 zeigt die Resultate anhand des Wigner Ville und Entropie Verfahrens zur Schriftneigungskorrektur.

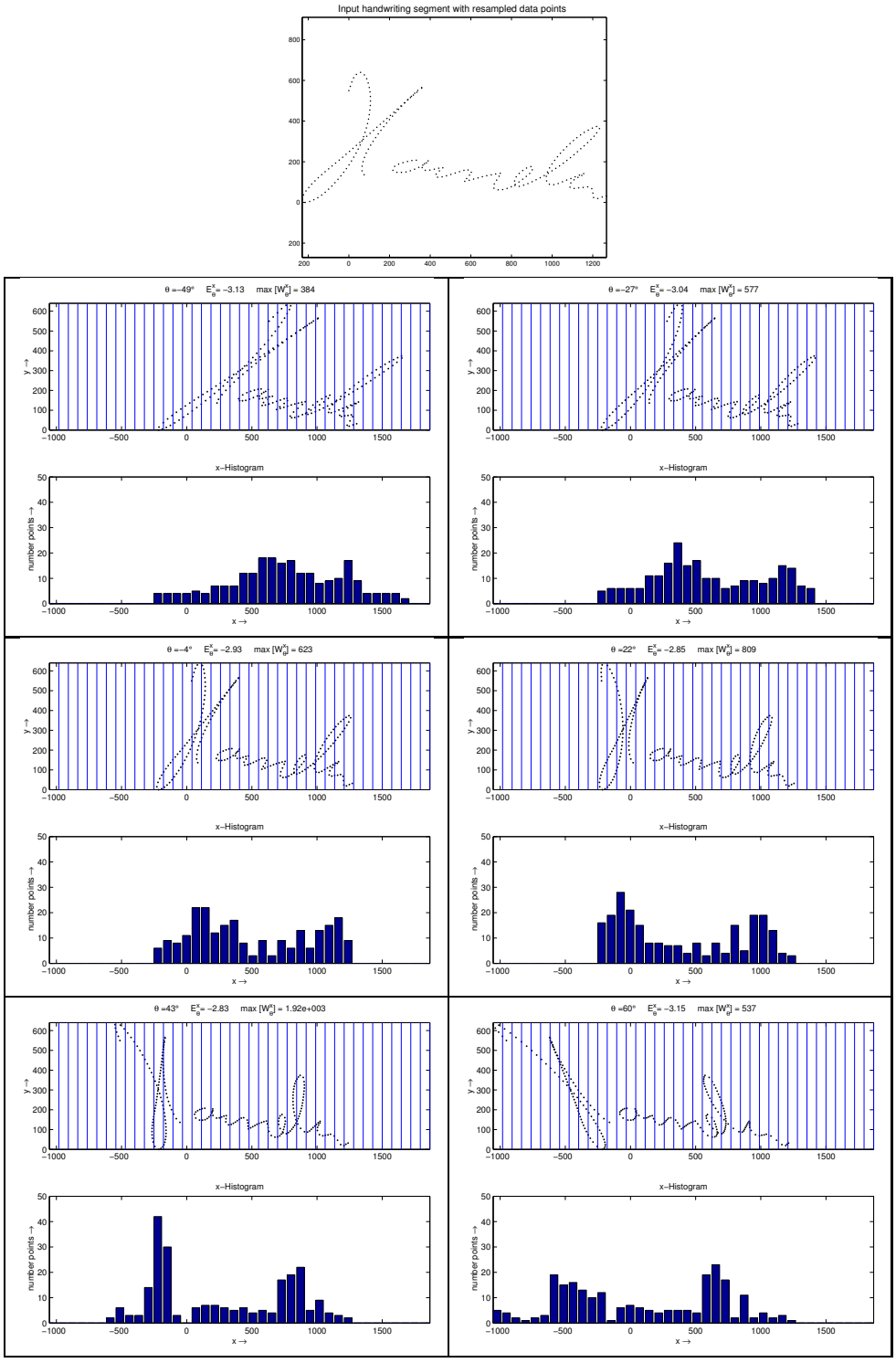


Abbildung 3.4: Korrektur der Schriftneigung anhand des neuabgetasteten Schriftzuges 'Hameln'. Im Bild in der ersten Spalte und dritten Zeile sind deutlich zwei Peaks durch die Oberlänge des Buchstabens 'H' und 'l' zu erkennen.

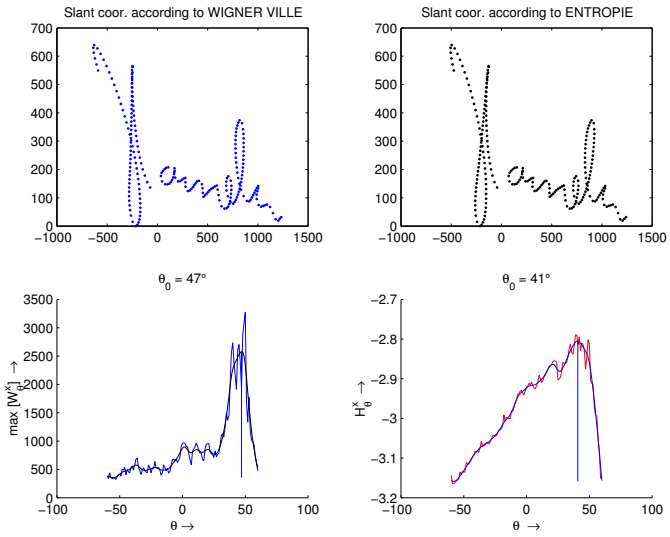


Abbildung 3.5: Ergebnisse des Wigner Ville und Entropie Auswerteverfahrens zur Schriftneigungskorrektur. Beide Verfahren zeigen eine deutliche Maximabildung im relevanten Bereich.

3.5 Referenzlinienschätzung

Die Erkennung von Schrift im Wortkontext bietet im Vergleich zu der Erkennung im Buchstabenkontext den Vorteil, dass relative Größenbeziehungen bei der Erkennung verwendet werden können. Liegt ohne Kontextwissen beispielsweise eine Verwechslung des kursiv geschriebenen Buchstabens 'e' gegenüber dem Buchstaben 'l' vor, kann nun durch die relative Größe eine Diskriminierung stattfinden. Die Unterscheidung erfolgt anhand horizontaler Linien (*Referenzlinien)*, die sich in jedem Wort erkennen lassen. Außer bei Wörtern die nur aus Großbuchstaben bestehen, lassen sich immer genau zwei parallel zueinander ausgerichtete Referenzlinien angeben. Die *Grundlinie* (engl. *base line*) und *Kernlinie* (engl. *core line*) legen durch ihren vertikalen Abstand die *Kernhöhe* fest. Eine Schätzung beider Linien und eine anschließende Normierung anhand der Kernhöhe ist daher sinnvoll und wird im Folgenden durchgeführt. Zwei weitere Linien, die sich jedoch nicht innerhalb jedes Wortes schätzen lassen, sind die *Ober- (*engl. *ascender line)* beziehungsweise *Unterlängenlinie* (engl. *descender line*). Sie werden durch Buchstaben bestimmt, die über die Kern- beziehungsweise unter die Grundlinie reichen und somit so genannte *Ober-* beziehnungweise *Unterlängen* besitzen. Diese Buchstaben spielten bereits bei der Schriftneigungskorrektur eine wichtige Rolle.

Es werden in der Literatur mehrere Verfahren zur Referenzlinienschätzung beschrieben. Zwei der wohl bekanntesten Verfahren zur Auffindung der Grund- und Kernlinie sind die Extrempunktmethode und die y-Projektionsmethode. Beim ersten Ansatz werden die lokalen Minima und Maxima innerhalb eines Schriftzuges berechnet. Anschließend werden mittels unterschiedlicher Ansätze Geraden aus den Extrema extrahiert. Im einfachsten Fall wird je eine Regressionsgerade durch die Maxima beziehungsweise Minima getrennt berechnet. In Rosenthal et al.

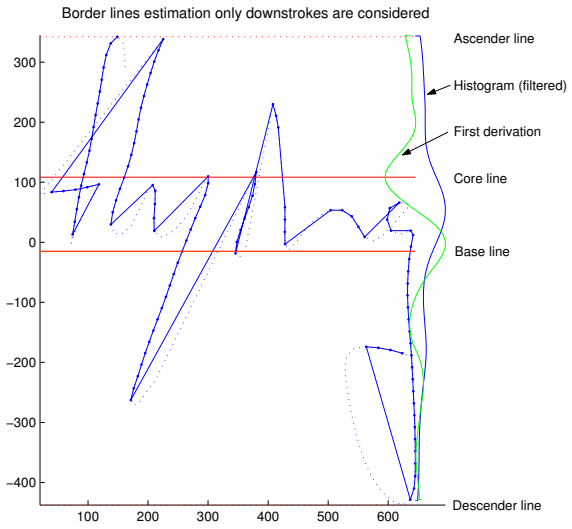


Abbildung 3.6: Referenzlinienschätzung durch die Histogrammanalyse. Zur besseren Auswertung des Histogramms wurden nur abwärts gerichtete Segmente betrachtet.

[38] wird eine modifizierte Hough-Transformation zur Schätzung der Geraden vorgeschlagen. In dieser Arbeit wurde die y-Projektionsmethode angewandt, die sich an den Arbeiten von Bozinovic [7] und Bunke [10] orientiert. Zudem erfolgt eine Restriktion auf abwärtsgerichtete Stiftbewegungen, wie es zum Beispiel in Seni et al. [46] vorgeschlagen wird. Hierdurch konnten Fehlinterpretationen bei der Auswertung des Histogramms weiter reduziert werden.

3.5.1 Verfahren

Unter der Annahme, dass das Wort durch die vorherigen Korrekturschritte horizontal zur x-Achse ausgerichtet ist, können die Referenzlinien parallel dazu geschätzt werden. Wie bereits bei der Zeilenneigunskorrektur festgestellt, bildet das y-Projektionsprofil P^y bei einer horizontalen Ausrichtung eine typische Verteilungsdichte mit ausgeprägtem Maxima (Hauptmaxima). Dies liegt daran, dass alle Buchstaben den Bereich zwischen der Grundlinie und Kernlinie (*Kernzone*) durchlaufen, jedoch nur einige davon Ober- beziehungsweise Unterlängen besitzen.

Um nun die Basis- und Kernlinie bestimmen zu können, werden die Extrema der ersten Ableitung $\mathsf{P}^{y'}$ bezüglich des Verteilungshistogramms bestimmt. Die Mitten der korrespondierenden Bins definieren die Grund- beziehungsweise Kernlinie. Die Ober- und Unterlängenlinie werden durch die y-Extrema definiert. Da diese Linien nicht weiter herangezogen werden, führen Fehlinterpretationen zu keinen Auswirkungen bei der Normierung. Für die Referenzlinien eines Wortes gilt somit:

$$y_{\text{Oberlängenlinie}} := \max_i \left(y_i^{\text{in}} \right)$$

$$
\begin{aligned}
y_{\text{Kernlinie}} &:= \left(\arg\min_l \left(\mathsf{P}^{y'}(l)\right) - 0.5\right) \cdot W + y^{\text{in}}_{\min} \\
y_{\text{Grundlinie}} &:= \left(\arg\max_l \left(\mathsf{P}^{y'}(l)\right) - 0.5\right) \cdot W + y^{\text{in}}_{\min} \\
y_{\text{Unterlängenlinie}} &:= \min_i \left(y^{\text{in}}_i\right)
\end{aligned}
$$

Vor Berechnung der ersten Ableitung wird eine Glättung des Projektionsprofils mithilfe eines Mittelwertfilters durchgeführt.

3.5.2 Verbesserung des Verfahrens

3.5.2.1 Beschränkung auf abwärtsgerichtete Stiftbewegungen

Eine Modifikation, die sich in der Praxis durchaus bewährt hat, stellt die Restriktion der in Betracht gezogenen Stiftbewegungen dar. In Brocklehurst and Kenward [9] und später auch in Seni et al. [46] wurden Versuche angestellt, ein Wort auf seine Grundinformationen einzuschränken. Hierzu wurde die Stiftbewegung in abwärtsgerichtete und aufwärtsgerichtete Bewegungen eingeteilt. Es konnte gezeigt werden, dass abwärtsgerichtete Stiftbewegungen bereits alle Informationen zur Referenzlinienschätzung beinhalten. Die Menge aller abwärtsgerichteten Stiftbewegungen berechnet sich hierbei wie folgt:

$$M_{y\downarrow} = \left\{ y^{\text{in}}_k \,\middle|\, \left(y^{\text{in}}_{k+1} - y^{\text{in}}_k\right) < 0 \quad \text{für } k \in [1..N^{\text{in}} - 1] \right\}$$

Diese Überlegung legt nahe, das y-Projektionsprofil nur anhand abwärtsgerichteter Stiftbewegungen zu berechnen. Die Folge ist, dass es zu weniger Irritationen bei der Referenzlinienschätzung kommt. Abbildung 3.6 zeigt das Verfahren am Beispiel des Wortes 'flying'.

3.5.2.2 Bestimmung der lokalen Extrema

Zur weiteren Verbesserung des Verfahrens wurden die Minima und Maxima der vertikalen Komponente eines jeden abwärtsgerichteten Stiftsegments berechnet. Bezeichne $M_{y\downarrow} = N_{1,y\downarrow} \cup ... \cup N_{H,y\downarrow}$ die Zerlegung der abwärtsgerichteten Stiftbewegungen in ihre zusammenhängenden Teilsegmente, für die gelte:

$$N_{i,y\downarrow} \cap N_{j,y\downarrow} = \emptyset \quad \text{für } i \neq j \text{ und } i, j \in [1, .., H].$$

Sei weiter

$$N_{0,y\downarrow} := \emptyset$$

$$M_{h-1,y\downarrow} := M_{y\downarrow} \setminus \{N_{1,y\downarrow} \cup ... \cup N_{h-1,y\downarrow}\} \quad \text{für } h \in [1...H]$$

$$k_{h-1,y\downarrow} := |M_{y\downarrow}| - |M_{h-1,y\downarrow}| + 1$$

$$L_{h-1,y\downarrow} = \begin{cases} \min\left\{ j \,\middle|\, \left(y^{\text{in}}_{j+1} - y^{\text{in}}_{j}\right) > 0 \,\text{mit}\, j \in [k_{h-1,y\downarrow} ... |M_{y\downarrow}|] \right\} \text{ für } 1 \leq h \leq H-1 \\ |M_{y\downarrow}| \text{ für } h = H \end{cases}$$

so ergibt sich ein Teilsegment durch

$$N_{h,y\downarrow} \quad = \quad \left\{ y^{\text{in}}_{l} \in M_{y\downarrow} \,|l \in [k_{h-1,y\downarrow} ... L_{h-1,y\downarrow}] \right\}$$

Anhand jedes Teilsegments können somit die Minima

$$Y_{\min} = \bigcup_{h \in [1...H]} \min N_{h,y\downarrow}$$

und entsprechend die Maxima

$$Y_{\max} = \bigcup_{h \in [1...H]} \max N_{h,y\downarrow}$$

zu Mengen (Cluster) lokaler Extrema zusammengefasst werden. Durch Anwendung eines iterativen Algorithmus werden nun solche Mengenelemente entfernt, die den jeweils weitesten Abstand zum Medianzentrum besitzen. Das Verfahren wird solange fortgesetzt bis sich das Clusterzentrum nur noch innerhalb einer vorgegebenen ϵ-Umgebung verschiebt. Anschließend wird eine stärkere Ausprägung des y-Histogramm, innerhalb des durch die beiden Clusterrepräsentanten aufgespannten Intervalls, durchgeführt. Zu jedem der betreffenden Bininhalte wird der global maximale Bininhalt hinzuaddiert. Es könnten zwar die Clusterrepräsentanten für sich selbst zur Referenzlinienbestimmung genutzt werden, jedoch führen diese zumeist zu einem zu engen setzen der Linien. Das Histogrammverfahren mit Erweiterung der Extrema-Methode hingegen betrachtet zusätzlich die Randregionen, wodurch die Linien moderater gesetzt werden. Insgesamt kommt es hierdurch zu weniger Schnitten zwischen den Referenzlinien und dem Schriftzug. Abbildung 3.7 veranschaulicht diesen Sachverhalt am Beispiel des Worte 'subject'. Die Extrema-Methode (gestichelte Linien) führt hier zu einem engen setzen der Referenzlinien, während das erweiterte Histogrammverfahren (durchgezogene Linien) die Referenzlinien toleranter setzt .

3.6 Schriftgrößenskalierung und äquidistante Neuabtastung

Ein weiterer Freiheitsgrad, der eingeschränkt werden muss, ist die Schriftgröße. Ziel einer Normierung sollte sein, dass ein Buchstabe, unabhängig von seiner Ausgangsgröße, auf dieselbe Größe normiert wird. Wie bei der Referenzlinienschätzung bereits erklärt wurde, kann in jedem Wort eine Kernhöhe geschätzt werden. Eine Normierung bezüglich der Kernhöhe ist somit eine gute Wahl und findet auch in dieser Arbeit statt. Seien P^{in} die aus den vorherigen Arbeitsschritten korrigierten Koordinatenpunkte. Es erfolgt eine Skalierung bezüglich dem Faktor

$$r = \left| y_{\text{Kernlinie}} - y_{\text{Grundlinie}} \right|$$

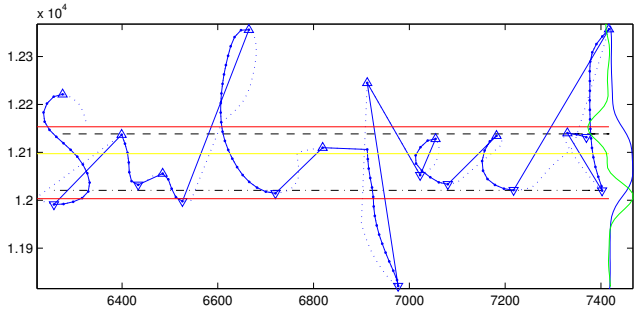


Abbildung 3.7: Erweiterte Histogrammanalyse unter Hinzunahme der lokalen Extrema. Die gestrichelten Linien kennzeichnen die geschätzten Referenzlinien anhand den beiden Clusterrepräsentanten der lokalen Minima beziehungsweise Maxima. Durch die Sättigung des Histogramms innerhalb diese Intervalls kann die Histogrammanalyse weiter verbessert werden. Insgesamt führt dies zu einer moderateren Bestimmung der Linien (durchgezogene Linien), wodurch auch der Anfangsbuchstabe 's', wie in diesem Fall, eher innerhalb der Kernzone zum Liegen kommt. Hierdurch kommt es zu weniger Schnitten zwischen den Referenzlinien und dem Schriftzug.

und eine Translation der Koordinatenpunkte in den Ursprung, indem die Mittenparallele der Basis- und Kernlinie mit der x-Achse zur Deckung kommt.

$$p_i^{\text{out}} = \frac{1}{r} \cdot \begin{pmatrix} x_i^{\text{in}} - x_{\min}^{\text{in}} \\ y_i^{\text{in}} - (y_{\text{Grundlinie}} + \frac{r}{2}) \end{pmatrix} \quad \text{für } i \in [1..N^{\text{in}}]$$

Besitzt der Schriftzug eine einheitliche Größe kann eine erneute äquidistante Neuabtastung der Daten erfolgen. Der Abstand zwischen den neuabgetasteten Koordinatenpunkten hat mehrere Auswirkungen auf das Gesamtsystem: eine Unterabtastung bedeutet auf der einen Seite Informationsverlust einhergehend mit unweigerlichen Einbußen in der Erkennungsleistung, auf der anderen Seite eine schnellere Klassifikation, da weniger Abtastpunkte (bzw. deren Merkmale) betrachtet werden müssen. Bei der Überabtastung verhält es sich genau umgekehrt: hier besteht der Datensatz zu einem Großteil aus redundanten Informationen, die die Klassifikationszeit unnötigt erhöhen. Ziel ist es, die Daten so informativ wie möglich zu halten, um bestmögliche Klassifikationsleistungen zu erreichen. Ein schwieriger Weg, der nur durch Testdurchläufe ermittelt werden kann.

3.7 Zusammenfassung

Dieses Kapitel beschrieb die Reduktion von Störgrößen, wie sie bei online erfassten Handschriftdaten vorzufinden sind. Am Ende der Vorverarbeitung stand die Normierung anhand von geschätzten Referenzlinien. Folgende Schritte wurden durchgeführt:

Durch eine Glättung mit anschließender Neuabtastung war es möglich, die variable Schreibgeschwindigkeit zu entfernen. Anschließend wurden Verfahren vorgestellt, mithilfe derer es mög-

lich ist den Zeilen- und Schriftneigungswinkel zu korrigieren. In beiden Verarbeitungsschritten kam das Histogrammverfahren zum Einsatz, wobei eine Auswertung durch die Entropie und die Wigner Ville Verteilung vorgestellt wurde. Im Falle der Zeilenneigungskorrektur wurde ein Vergleich beider Auswerteverfahren durchgeführt. Hierbei zeigte sich, dass die Wigner Ville Verteilung bessere Resultate nach einer Neuabtastung lieferte. Nach der Korrektur der Daten war es möglich Referenzlinien, parallel zur Horizontalen zu berechnen. Die Linien konnten ebenfalls mithilfe des Histogramms geschätzt werden, wobei nur abwärtsgerichtete Teilsegmente für die Berechnung herangezogen wurden. Zusätzlich konnte durch die Bestimmung der lokalen Extrema ein weiteres Entscheidungsmerkmal hinzugezogen werden. Zuletzt fand die Normierung der Abtastdaten anhand der geschätzten Kernhöhe statt.

Die Vorverarbeitungsschritte werden im Folgenden für die Erkennung eines Wortes durchgeführt. Da der Einfluss der Neuabtastung auf die Erkennungsrate zu diesem Zeitpunkt noch nicht eingeschätzt werden kann, findet eine Vorverarbeitung mit und ohne Neuabtastung statt. Hierdurch ist es möglich die Merkmale, wie sie im Kapitel 5 noch vorgestellt werden, auf unterschiedliche Arten zu berechnen.

KAPITEL 4

Segmentierung

Ein bis heute nur unzureichend gelöstes Problem stellt die Zerlegung eines Wortes in seine Einzelbuchstaben (Wortsegmentierung) vor dem Erkennungsschritt dar. So ist es notwendig, einen Buchstaben erst erkannt zu haben, bevor dessen Grenzen bestimmt werden können (Beispiel: 'u' 'w'). Dieses Beobachtung formulierte bereits Sayre [42] 1973: "To recognize a letter, one must know where it starts and where it ends, to isolate a letter, one must recognize it first". Zudem beeinflusst eine im Vorverarbeitungsschritt durchgeführte Fehlsegmentierung die Klassifikation so nachhaltig, dass eine korrekte Erkennung nicht mehr möglich ist. Aus diesem Grund wurden im Laufe der Jahre mehrere Ansätze entwickelt, die von einer Vorsegmentiertung absehen. Zwei der bekanntesten Ansätze sind einerseits die ganzheitliche *holistische* und andererseits die *analytische* Variante. Ein weniger bekannter, aber durchaus viel versprechender Ansatz ist der *wahrnehmungsorientierte* Ansatz.

4.1 Holistische Variante

Bei diesem Verfahren wird versucht anhand eines Referenzdatensatzes, bestehend aus ganzen Wörtern, ein Wort zu erkennen. Während des Erkennungsprozesses kann somit auf die Segmentierungsinformation verzichtet werden. Der entscheidende Nachteil liegt in der Größe des Referenzdatensatzes. Er muss von jedem zu erkennenden Wort mehrere Schreibvariationen beinhalten, wodurch die Größe überproportional mit dem Wortschatz ansteigt. Beispiele für Forschungsarbeiten auf kleinen Lexika finden sich in der Literatur bei Bercu and Lorette [5] und Farag [17].

4.2 Analytische Variante

Bei der analytischen Methode werden die Worte nicht als Ganzes betrachtet, sondern man baut sie aus kleineren Einheiten wie zum Beispiel Buchstaben (*Grapheme*) und Buchstabenverbindungen (*Ligaturen*) oder Strichen (engl. *strokes*) zusammen. Hierdurch ist es möglich unabhängig von einem fixen Referenzdatensatz die Erkennung modular voranzutreiben. Das analytische Verfahren lässt sich desweiteren in einen *expliziten* und einen *impliziten* Segmentierungsansatz unterteilen.

Explizit bedeutet, dass vor der Erkennung eine grobe Segmentierung des Wortes in kleinere Einheiten erfolgt, während bei der impliziten Methode die Segmentierungsinformation als Nebenprodukt während des Erkennungsprozesses entsteht. In beiden Fällen sollte mittels eines Lexikons der Suchraum sukzessive eingeschränkt werden. Das Lexikon kann entweder aus endlich vielen Wörtern, die zumeist in einem Lexikonbaum (engl. *trie*) organisiert werden, oder einer Statistik über die gemeinsame Auftretenswahrscheinlichkeit n-vieler Buchstaben (*n-Gramme*) bestehen. Typische n-Gramme sind bi-Gramme, welche Information über die Auftretenswahrscheinlichkeit zweier Buchstaben innerhalb einer Sprache enthalten. So besitzt zum Beispiel das bi-Gram 'au' im deutschen eine sehr viel höhere Auftretenswahrscheinlichkeit als das bi-Gram 'jb'. Der Vorteil dieser Variante liegt in der Unabhängigkeit des Lexikon zum Referenzdatensatz. Jedoch spielt die Lexikongröße einen nicht unwesentlichen Einfluss auf die Erkennungsleistung wie in vielen Arbeiten gezeigt werden konnte.

Die besten Forschungsergebnisse auf großen Lexika konnten im Bereich der analytischen Erkennung mit impliziter Segmentierung erreicht werden. Eine der wichtigsten Arbeiten mit hervorragenden Erkennungsleistungen stellen Hu et al. [21] dar. Die Einheiten dieses Systems bilden so genannte *nebulous strokes*, wobei ein Buchstabe aus mehreren solcher Strichen aufgebaut ist. In Schomaker and Teulings [44] werden die Vor- und Nachteile diese Verfahrens gegenüber dem buchstabenbasierten Ansatz bei der Erkennung von Worten diskutiert. Desweitern konnten sehr gute Erkennungsleistungen in Manke et al. [29] vorgestellt werden. Es sei bemerkt, dass ein allgemeiner Vergleich der Erkennungsysteme kaum möglich ist, da die meisten Resultate auf eigenen Datensätzen erzielt wurden.

4.3 Wahrnehmungsorientierte Variante

Betrachtet man die Herangehensweise eines Menschen beim Entziffern einer unleserlichen Schrift, so versucht dieser durch die Identifizierung einzelner ihm bekannt erscheinender Zeichen das Wort nach und nach zu rekonstruieren. Hierbei geht er zumeist nicht sequenziell vor, sondern orientiert sich an leicht zu identifizierenden Zeichen. Der wahrnehmungsorientierte Ansatz imitiert eben diese Vorgehensweise. Konnten einzelne Teile eines Wortes identifiziert werden, so versucht man die Zwischenräume mit Hilfe von kontextuellem Wissen zu ergänzen, um auf das ganze Wort schließen zu können. Diese Variante arbeitet unabhängig von vorangegangenen Beobachtungen und benutzt lokale oder globale Merkmale eines Wortbildes, wodurch eine Segmentierung nicht erforderlich ist. Dieses Paradigma, welches den Erkennungsprozess weg

von zu erkennenden Mustern und hin zu wissensbasierten Systemen führt, wurde unter anderem von Lorette [27] und Anquitil and Lorette [2] vorgeschlagen. Forschungsarbeiten, die auf kleinen Lexika beruhen, finden sich in der Literatur bei Cheriet and Suen [12] und Edelman et al. [15] wieder.

4.4 Zusammenfassung

In der vorliegenden Arbeit wird die implizite analytische Variante verfolgt, bei sich eine Segmentierung als Nebenprodukt während des Erkennungsprozesses ergibt.

KAPITEL 5

Merkmalsberechnung

5.1 Überblick

Zur Klassifikation eines Schriftzuges ist es notwendig, ihn auf seine Hauptmerkmale zu reduzieren. Im Idealfall könnte man ihn durch selbige eindeutig beschreiben und somit von allen anderen unterscheiden. Die Realität zeigt jedoch, dass solche eindeutigen Unterscheidungsmerkmale in der Handschrifterkennung nicht existieren. Dies hat zur Folge, dass in der Literatur eine Unmenge von Merkmalen speziell für die Worterkennung aufgeführt werden. Durch Kombination mehrerer Merkmale versucht man die Erkennungsleistung weiter in die Höhe zu treiben. Für die Handschrifterkennung sind vor allem Merkmale relevant, welche invariant gegenüber *Ähnlichkeitstransformationen* sind. Hierbei handelt es sich um eine Kombination von Verschiebung, Rotation und Skalierung. Sei $\mathbf{S}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ die parametrische Darstellung einer Kurve in der euklidischen Ebene mit $t \in \mathbb{R}$, so läßt sich die Ähnlichkeitstransformation $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ darstellen durch

$$\tilde{\mathbf{S}}(\tilde{t}) = k \cdot \mathbf{A} \cdot \mathbf{S}(t) + \mathbf{b}$$

mit dem Skalierungsfaktor $k \in \mathbb{R}^+$, der Drehmatrix $\mathbf{A} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$ mit dem Drehwinkel $\alpha \in [0, 2\pi[$ und $\mathbf{b} = \begin{pmatrix} b_x \\ b_y \end{pmatrix}$ als Verschiebungsvektor. Zwei Kurven heißen *äquivalent*, wenn sie sich mittels einer Ähnlichkeitstransformation ineinander überführen lassen. Ähnlichkeitsinvariante Merkmale besitzen somit die Eigenschaft, dass sie dieselben Werte an korrespondierenden Punkten innerhalb äquivalenter Kurven erzeugen.

Merkmale, die bezüglich einer Schriftkurve extrahiert werden können, lassen sich grob in eine lokale (engl. *local*) und globale (engl. *long-range*) Kategorie unterteilen. Zur lokalen Klasse gehören Merkmale, die nur einen begrenzten Ausschnitt (engl. *frame*) eines Schriftzuges betrachten. Es handelt sich zum Beispiel um normierte Abtastkoordinaten, Tangentensteigung und Krümmung in den Kurvenpunkten (Abtastpunkten). Im globalen Fall kommen Schleifen, sich schneidende Kurvensegmente und Spitzen in den Kurven zum Tragen. Wegen großer Formvarianzen in der natürlichen Handschrift, haben globale Merkmale die Eigenschaft aussagekräftiger, aber weniger robust zu sein; während sich lokale Merkmale weniger informativ aber umso zuverlässiger zeigen. Dies hat zur Folge, dass sich die Suche auf lokale Merkmale konzentriert und globale Merkmale meist nur in der Nachbearbeitungsphase zur Abdeckung von zweideutigen Klassifikationsergebnissen herangezogen werden.

In dieser Arbeit werden nur lokale Merkmale zur Klassifikation verwendet, wodurch jeder Abtastpunkt der Schriftkurve durch einen Vektor von lokalen Merkmalen repräsentiert wird, im Weiteren auch mit *Merkmalsvektor* bezeichnet. Genauer gilt:

Sei $p_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ der i-te Abtastpunkt des Kurve, so besteht sein Repräsentant $\mathbf{v}_i = (v_1, v_2, v_3, .., v_F)^T \in \mathbb{R}^F$ aus F reellwertigen Merkmalen. Sie spannen den F-dimensionalen *Merkmalsraum* auf und müssen den zu klassifizierenden Datensatz hinreichend genau beschreiben. Für N Abtastpunkte ergibt sich somit eine *Merkmalsvektorfolge* von zeitlich geordneten Merkmalsvektoren $[\mathbf{v}_1, \ldots, \mathbf{v}_N]$, welche dem Klassifikator übergeben werden.

Ansätze, globale Merkmale in ein solches System zu integrieren, finden sich in Manke et al. [28] und Hu et al. [22]. Manke benutzt lokale Bitmap-Merkmale, die das Schriftbild in einer begrenzten Umgebung erfassen. Hierdurch ist es möglich, alle zuvor zeitlich voneinander getrennten geometrischen Nachbarschaftsbeziehungen zu erfassen. Ein typisches Beispiel sind so genannte *delayed strokes*, welche nicht direkt beim Schreiben des jeweiligen Zeichens gesetzt werden, sondern erst nachträglich ins Wort eingefügt werden. Jedoch können keine Schleifen erfasst werden, die durch ihre zeitliche wie auch räumliche Ausdehnung charakterisiert sind. Einen anderen Ansatz, ohne diese fehlende Eigenschaft, zeigt Hu et al. [22] auf. Hierbei werden die drei globalen Merkmale Spitzen, sich schneidende Kurvenstücke und Schleifen in einem Vorverarbeitungsschritt erfasst und mittels einer 'Lokalisierungsmethode' auf die lokalen Nachbarpunkte verteilt, während weit entfernte Punkte mit einem konstanten Wert belegt werden.

5.2 Merkmalsdefinitionen

In diesem Abschnitt werden unterschiedliche Merkmale für die Worterkennung vorgestellt und es wird auf deren Berechnung näher eingegangen. Dass die Komplexität der Merkmale nicht entscheidend ist, konnte schon anhand der Buchstabenerkennung Bahlmann and Burkhardt [3] gezeigt werden. So ergaben hier die normierten x/y-Koordinaten kombiniert mit dem Tangentensteigungswinkel die besten Klassifikationsergebnisse. Ein einfacherer Merkmalssatz ist kaum auszumalen. Bedenkt man zudem, dass bei dem Vergleich eines Test- mit einem Referenzmuster die Berechnungskomplexität mit der Größe des Merkmalsatzes ansteigt, so sollte ein minimaler

Satz an Merkmalen bei maximaler Klassifikationsleistung das Ziel eines jeden Erkennungssystems im Bereich von Realtime-Anwendungen sein.

Ausgehend von den Erläuterungen aus Kapitel 3 liegt der Schriftzug nun sowohl als neuabgetastete Koordinatensequenz als auch als Splinekurve vor. Dies ermöglicht es, die Merkmale auch anhand der Splinekurve zu berechnen.

Wurde keine Neuabtastung nach dem Normierungsprozess durchgeführt, so werden diese Ortskoordinaten als *original abgetastete* beziehungsweise bei Neuabtastung als *neu abgetastete* Koordinaten bezeichnet. Zur Unterscheidung wird folgende Variablenbezeichnung eingeführt:

1. original abgetastete Koordinaten:

$$p_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \qquad \text{für } 0 \leq i \leq N$$

2. neu abgetastete Koordinaten :

$$\tilde{p}_j = \begin{pmatrix} \tilde{x}_j \\ \tilde{y}_j \end{pmatrix} \qquad \text{für } 0 \leq j \leq \tilde{N}$$

5.2.1 Normierte Ortskoordinaten $\mathbf{P}^{(\mathrm{norm})}$ beziehungsweise $\tilde{\mathbf{P}}^{(\mathrm{norm})}$

Nach Durchführung der Vorverarbeitung mit anschließender Normierung ist es möglich, die y-Koordinate als Merkmal direkt zu übernehmen. Bei der x-Koordinate ist dies nicht der Fall. Sie ist lageabhängig was bedeutet, dass ein und derselbe Buchstabe an unterschiedlichen Positionen im Wort unterschiedliche x-Merkmalswerte erzeugt. Erst einer Translation in x-Richtung, welche die beiden Buchstaben zur Deckung bringt, könnte ein Vergleich anhand der x-Merkmale ermöglichen. Da die Erkennung anhand von isolierten Buchstaben erfolgt, müsste während des Erkennungsprozesses jedes neu angelegte Referenzmodell um den bereits erkannten Wortteil verschoben werden. Der Betrag der Verschiebung hängt von den erkannten Segmentierungsgrenzen innerhalb des Wortes ab. Das Auffinden der richtigen Verschiebungsvektoren kann daher mit dem Segmentierungsproblem gleichgesetzt werden. Eine Möglichkeit diese Problem zu umgehen, ist die indirekte Rekonstruktion der x-Koordinate anhand der y-Koordinate und dem zugehörigen Tangentensteigungswinkels θ unter der Annahme, dass eine äquidistante Neuabtastung durchgeführt wurde. Diese Idee würde aber bedeuten, dass die Zwischensequenzen, in denen der Stift abgesetzt wurde, ebenfalls neuabgetastet werden müsste. Schießlich muss zu einer Rekonstruktion eine stetige Schriftkurve vorliegen.

$$p_i^{(\mathrm{norm})} := p_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

gleiches gilt für die neu abgetasteten Koordinaten

$$\tilde{p}_j^{(\mathrm{norm})} := \tilde{p}_j = \begin{pmatrix} \tilde{x}_j \\ \tilde{y}_j \end{pmatrix}$$

5.2.2 Tangentensteigungswinkel θ

Das Merkmal berechnet den Winkel θ zwischen der Horizontalen und dem Tangentenvektor in den Abtastpunkten. Genauer gilt für die Berechnung über Bildung des Differenzenquotienten folgendes Schema:

$$\theta_i = \arctan\left(\frac{\triangle x_i}{\triangle y_i}\right) \quad \text{für } 2 \leq i \leq N-1$$

wobei für $2 \leq i \leq N-1$

$$\triangle x_i = x_{i+1} - x_{i-1}$$
$$\triangle y_i = y_{i+1} - y_{i-1}$$

und in den Randbereichen

$$\triangle x_1 = x_2 - x_1 \quad \triangle x_i = x_N - x_{N-1}$$
$$\triangle y_1 = y_2 - y_1 \quad \triangle y_i = y_N - y_{N-1}$$

gilt.

Des Weiteren kann eine Berechnung des Merkmals über die Splinedarstellung

$$\mathbf{S}(t_j) = \begin{pmatrix} S_x(t_j) \\ S_y(t_j) \end{pmatrix}$$

bezüglich den neu abgetastete Ortskoordinaten stattfinden. Sei $i := \sqrt{-1}$ und angle die komplexe Winkelfunktion so gilt:

$$\mathbf{S}'(t_j) = \begin{pmatrix} S'_x(t_j) \\ S'_y(t_j) \end{pmatrix}$$
$$\tilde{\theta}(t_j) = \text{angle}(S'_x(t_j) - i \cdot S'_y(t_j))$$

5.3 Zusammenfassung

Zusammenfassend liegt eine Schriftprobe als Merkmalsrepräsentation $\mathrm{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathrm{t}}]$ mit $\mathbf{t}_i \in \mathbb{R}^F$ vor. Fand bei der Normierung eine Neuabtastung statt, so konnte zusätzlich die Splinekurve zur Berechnung herangezogen werden.

KAPITEL 6

Klassifikation von Einzelzeichen

6.1 Überblick und Einleitung

In Abschnitt 6.2 erfolgt eine Einführung in das so genannte Dynamic Time Warping (DTW) Verfahren, das sich bereits in der Spracherkennung durchsetzen konnte. Es werden im Weiteren die allgemeinen Randbedingungen, seine algorithmische Umsetzung und die Strahlsuche beschrieben. Bei diesen Erläuterungen reicht es aus, die Referenzmuster als einfache Merkmalsvektorfolge zu betrachten. In Abschnitt 6.3 wird eine Erweiterung des Referenzmusters anhand statistischer Größen vorgestellt. Vor der Schätzung der Statistiken ist es notwendig, die Trainingsdaten in Cluster ähnlicher Datensätze einzuteilen. Hierzu kann das beschriebene DTW Verfahren eingesetzt werden. Somit ergibt sich ein ganzheitlicher (*holistischer)* Ansatz, bei dem sowohl im Training als auch bei der Klassifikation die gleichen Verfahren angewandt werden können. In Abschnitt 6.4 wird die Erweiterung des DTW Verfahrens aufgezeigt, das als '*cluster-generative statistical DTW*' (CSDTW) bezeichnet wird. Zum Abschluss findet sich in Abschnitt 6.5 eine kurze Zusammenfassung.

In den vorherigen Kapiteln wurde beschrieben wie aus einem abgetasteten Schriftzug eine temporal geordnete Merkmalsvektorfolge $\mathrm{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathrm{t}}]$ entsteht. Zur Bestimmung der Semantik müssen dem Rechner so genannte *Referenzmuster* zur Verfügung gestellt werden, anhand derer eine Gegenüberstellung stattfinden kann. Wird ein Vergleich zwischen einem Testmuster und einem Referenzmuster durchgeführt, so bezeichnet man das Referenzmuster als *(aktive) Hypothese.* Aus der Menge der Referenzmuster wird diejenige Hypothese gesucht, welche gegenüber dem unbekannten Testmuster am "ähnlichsten" ist. Dabei muss zunächst ein Maß für die Ähnlichkeit zweier Muster gefunden werden. Hierzu hat sich in der Sprach- und Handschrifterkennung das Dynamic Time Warping Verfahren bewährt.

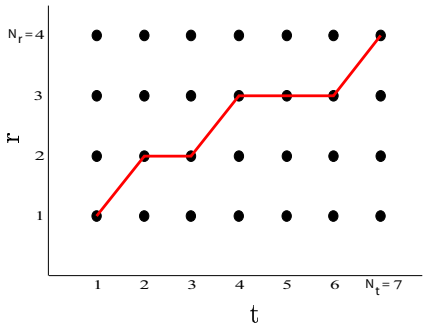


Abbildung 6.1: Beispiel für die zeitliche Angleichung zweier Merkmalsvektorfolgen entlang des Warpingpfades ϕ

Im ganzen Kapitel wird zunächst davon ausgegangen, dass es sich beim Testmuster, wie auch bei den Referenzmustern um Einzelzeichen handelt. Diese Einschränkung soll einerseits die Komplexität beim Erläutern reduzieren, andererseits trägt diese Sichtweise dem Verständnis des Trainings der Referenzmodelle bei. Erst im nachfolgenden Kapitel soll das Testmuster im Hinblick auf die Worterkennung auch ein ganzes Wort beschreiben dürfen.

6.2 Einführung DTW

Gegeben sei ein Testmuster, repräsentiert durch seine temporal geordnete Merkmalsvektorfolge $\mathrm{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathrm{t}}]$ und ein Referenzmuster $\mathrm{r} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathrm{r}}]$ mit $\mathbf{t}_i, \mathbf{r}_j \in \mathbb{R}^F$ für $1 \leq i \leq N_\mathrm{t}$ $1 \leq j \leq N_\mathrm{r}$. Eine typische Eigenschaft temporal gewonnener Merkmale ist die große Variabilität in der zeitlichen Struktur. Dies bedeutet unter anderem, dass die Merkmalsvektorfolgen zumeist unterschiedliche Längen besitzen $N_t \neq N_r$. Ein Charakteristika, welches sich auch in der Handschrift findet, wo jeder Buchstabe ein Unikat bezüglich der Länge, Schreibgeschwindigkeit und Form darstellt. Der Vergleich beider Muster muss der zeitlichen Strukturunterschiede Rechnung tragen, indem eine zeitliche Angleichung korrespondierender Merkmale stattfindet. In der Fachsprache nennt man einen solchen Vorgang *warping*. Sei $\Phi = (\phi(1), ...\phi(N))$ mit $\boldsymbol{\phi} = (\phi_\mathrm{t}, \phi_\mathrm{r}) : \{1, \ldots, N\} \rightarrow \{1, \ldots, N_\mathrm{t}\} \times \{1, \ldots, N_\mathrm{r}\}$ eine *Warpingfunktion*, welche aus N Merkmalvektorpaaren besteht. Trägt man die Muster in der Ebene gegeneinander auf, so spricht man auch von einem *Warpingpfad*, der durch die Indizes der aufgespannten Gitterpunkte führt (siehe Abbildung 6.1).

Abhängig vom gewählten Pfad Φ lässt sich eine globale *Unähnlichkeit* (*Dissimilarity*) $D_{\boldsymbol{\Phi}}[d]\,(\mathrm{t}, \mathrm{r})$ wie folgt definieren:

$$D_{\boldsymbol{\Phi}}[d]\,(\mathrm{t}, \mathrm{r}) = \sum_{n=1}^{N} d\left(\mathbf{t}_{\phi_\mathrm{t}(n)}, \mathbf{r}_{\phi_\mathrm{r}(n)}\right)$$

Als lokale Distanzfunktion $d : \mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}$, findet zumeist das Quadrat des euklidischen

Abstandes

$$d\left(\mathbf{t}_i, \mathbf{r}_j\right) = \left\|\mathbf{t}_i - \mathbf{r}_j\right\|^2 \tag{6.1}$$

Anwendung. Eine naheliegende Methode die Ähnlichkeit zweier Muster zu beschreiben, ist die Bestimmung der kleinsten akkumulierten Distanz über alle möglichen Warpingpfade. Diese minimale Distanz bezeichnet man als *Viterbidistanz* $D^*[d]\left(\mathrm{t},\mathrm{r}\right)$ und den zugehörigen optimale Pfad Φ^*, *Viterbipfad*. Genauer gilt für die Viterbidistanz:

$$D^*[d]\left(\mathrm{t},\mathrm{r}\right) = D_{\mathbf{\Phi}^*}[d]\left(\mathrm{t},\mathrm{r}\right) = \min_{\mathbf{\Phi}}\left\{D_{\mathbf{\Phi}}[d]\left(\mathrm{t},\mathrm{r}\right)\right\} \tag{6.2}$$

Zur Auffindung der Viterbidistanz wird auf die Rekombinationstechnik der *dynamischen Programmierung* (DP) [4] zurückgegriffen. Hierzu wird das zu lösende Problem rekursiv in Teilprobleme zerlegt, wobei die schrittweise Kombination der Teillösungen das Gesamtproblem löst. Teillösungen werden in Tabellen zwischengespeichert, um Mehrfachberechnungen zu vermeiden. Ist zusätzlich das Optimalitätsprinzip, wie im zugrunde liegenden Fall, gegeben (ein optimaler Pfad besteht aus optimalen Teilpfaden), so garantiert die DP die Bestimmung einer optimalen Lösung.

Der gesamte Prozess wird *Dynamic Time Warping (DTW)* genannt (auch bekannt unter dem Namen *Time Alignment* [36] oder *Elastic Matching* [48]), ein Verfahren, das seine Ursprünge in der Spracherkennung hat und 1971 in der Veröffentlichung von Sakoe and Chiba [40] erstmalig genauer beschrieben wurde. Eine zusätzliche statistische Modellierung führte schließlich zu den heute vielfach verwendeten *Hidden Markov Modellen (HMM)*. Das DTW Verfahren bildet nicht nur die Grundlage des entwickelten Klassifikators, sondern wird ebenfalls zur Generierung der Referenzmodelle eingesetzt. Im Folgenden werden Bedingungen an den Warpingpfad gestellt, um den Suchraum weiter einschränken zu können.

6.2.1 Einschränkungen des Warping-Pfades

Der Anwendungsdomäne entsprechend können Einschränkungen und Bedingungen an den Warpingpfad gestellt werden. So ist die Einhaltung der zeitlichen Reihenfolge wichtig, um die Ordnung der Muster nicht zu zerstören. Durch die Neuabtastung der Schrift kann zudem die Forderung gestellt werden, dass kein Merkmalsvektor bei einem Vergleich unbeobachtet bleiben darf. In der Spracherkennung hingegen, wo Sprechgeschwindigkeiten nicht ohne weiteres entfernt werden können, werden Auslassungen von Merkmalsvektoren zugelassen, wodurch eine bessere temporale Angleichung gewährleisten werden kann. Zudem ist man daran interessiert, dass der Pfad beide Muster möglichst komplett durchquert. Um diesen Anforderungen zu entsprechen, werden die nachfolgenden Einschränkungen gefordert.

6.2.1.1 Anfangs- und Endpunktbedingungen

Es wird gefordert, dass der erste Merkmalsvektor des Testmusters mit dem ersten Merkmalsvektor des Referenzmusters verglichen wird. Entsprechende Forderungen werden auch an die Merkmalsvektoren am Ende der Muster gestellt:

Anfangsbedingung	$\phi(1) = (1, 1)$
Endbedingung	$\phi(N) = (N_\mathrm{t}, N_\mathrm{r})$

6.2.1.2 Monotonie

Um die zeitliche Reihenfolge der Merkmalsvektoren zu erhalten, werden folgende Forderungen gestellt:

$$\phi_\mathrm{t}(n+1) \geq \phi_\mathrm{t}(n) \qquad \text{und} \qquad \phi_\mathrm{r}(n+1) \geq \phi_\mathrm{r}(n) \qquad \text{für } 1 \leq n \leq N-1$$

Hierdurch ist es zudem möglich, die Berechnungseffizienz deutlich zu steigern, da weniger Pfade betrachtet werden müssen.

6.2.1.3 Lokale Stetigkeitsbedingungen

Bei geringen Redundanzen ist es durchaus sinnvoll, jeden Merkmalsvektor miteinander zu vergleichen, um keine Information unberücksichtigt zu lassen. Kombiniert mit der Forderung nach Monotonie ergeben sich die von Sakoe and Chiba [41] vorgeschlagenen einfachen lokalen Stetigkeitsbedingungen:

$$\phi_\mathrm{t}(n+1) - \phi_\mathrm{t}(n) \leq 1 \qquad \text{und} \qquad \phi_\mathrm{r}(n+1) - \phi_\mathrm{r}(n) \leq 1$$

Zur Anschauung besser geeignet ist die Beschreibung mittels relativer Pfade. Hierzu sei $\mathbb{P} = \{(1,0), (0,1), (1,1)\}$ die Menge der erlaubten Transitionen. So muss für den relativen Pfad

$$\triangle\phi = \phi(n-1) - \phi(n) \in \mathbb{P}$$

gelten. Dies bedeutet, dass mindestens ein Schritt vorwärts gegangen werden muss und maximal ein Schritt in beiden Merkmalsvektoren vorwärts gegangen werden darf. Mittels dieser Pfadübergänge ist weiterhin jeder Gitterpunkt erreichbar. In [36] werden weitaus komplexere lokale Stetigkeitsbedingungen speziell für die Spracherkennung formuliert. Diese Modelle basieren auf rein heuristischen Untersuchungen. Dies begründet sich damit, dass die Sprechgeschwindigkeit und die zeitlichen Variationen in einem Sprachmuster sich nur schwer modellieren lassen.

6.2.2 Algorithmen zur effizienten Berechnung

Für die Implementierung erfolgt die Darstellung der aufgespannten Gitterebene durch die *Viterbimatrix* $\mathbf{D}^* = \left[D^*_{i,j}\right]$. Der Eintrag (i, j) in der Viterbimatrix bezeichnet die akkumulierte Distanz der Merkmalsvektorenpaare $(\mathbf{t}_i, \mathbf{r}_j)$ ausgehend vom ersten Merkmalsvektorpaar $(\mathbf{t}_1, \mathbf{r}_1)$. Um die Anfangsbestimmung zu erfüllen, wird die Matrix um eine Anfangszeile und Anfangsspalte, entsprechend Abbildung 6.2, erweitert.

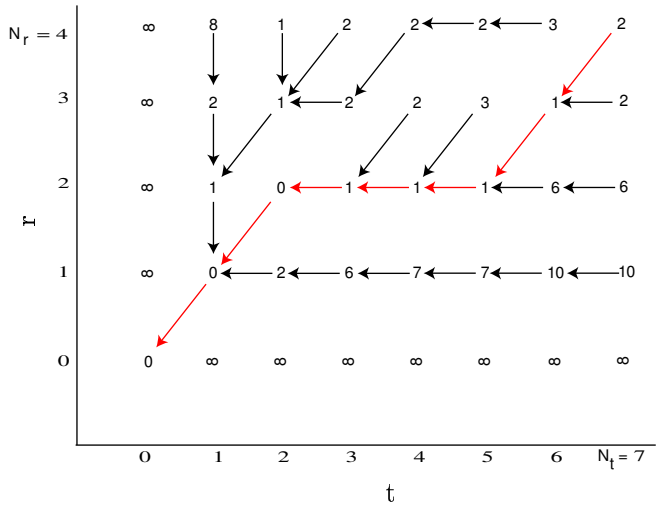


Abbildung 6.2: Die erweiterte lokale Distanzmatrix (*Viterbimatrix*) $\left[D^*_{i,j}\right]$. Ihre Berechnung erfolgt rekursiv mithilfe dynamischer Programmierungstechniken. Die Erweiterung der *Viterbimatrix* um eine Anfangszeile und Anfangsspalte, welche außer im Element $D^*_{0,0}$ mit ∞ inizialisiert werden, garantieren, dass der Viterbipfad beim jeweils ersten Merkmalsvektor beginnt.

Durch die zuvor gestellten Bedingungen an die Warpingpfade ist es zwar möglich, den Suchraum einzugrenzen, jedoch können mittels den lokalen Pfadübergängen immer noch alle Gitterpunkte erreicht werden. Es müssten somit in einem naiven Ansatz exponentiell viele Pfade exploriert werden, um die Viterbidistanz 6.2 und den korrespondierenden Viterbipfad bestimmen zu können. Die Berechungskomplexität dieses Verfahrens wächst proportional zur Größe der Viterbimatrix $\mathcal{O}\left(|\mathbb{P}|^{\tilde{N}}\right)$ bei einer mittleren Pfadlänge von $\tilde{N}$. Durch die Eigenschaft, dass sich jeder Viterbipfad aus optimalen Teilpfaden zusammensetzt, ist das Optimalitätsprinzip erfüllt und der dynamische Programmierungsansatz garantiert das Auffinden eines Viterbipfades in Zeit $\mathcal{O}\left(|\mathbb{P}|\,\tilde{N}^2\right)$. Um Teilpfade beginnend beim ersten Merkmalsvektorpaar aufbauen zu können, wird die Viterbidistanz rekursiv umgeschrieben. So lässt sich $D^*[d]\,(\mathrm{t},\mathrm{r})$ bestimmen, indem über alle lokalen Pfade, die im Punkt $(N_\mathrm{t}, N_\mathrm{r})$ enden, rekursiv minimiert wird. Es gilt:

$$\begin{aligned} D^*[d]\,(\mathrm{t},\mathrm{r}) &= D^*_{N_\mathrm{t},N_\mathrm{r}} \\ &= d(N_\mathrm{t},N_\mathrm{r}) + \min\left\{\begin{array}{c} D^*_{N_\mathrm{t}-1,N_\mathrm{r}} \\ D^*_{N_\mathrm{t}-1,N_\mathrm{r}-1} \\ D^*_{N_\mathrm{t},N_\mathrm{r}-1} \end{array}\right\} \end{aligned}$$

mit

$$\begin{aligned} D^*_{0,0} &= 0 \\ D^*_{1,1} &= d(\mathbf{t}_1,\mathbf{r}_1) \end{aligned}$$

$$D^*_{0,k} = D^*_{k,0} = \infty \qquad \text{für } 1 \le k \le N-1$$

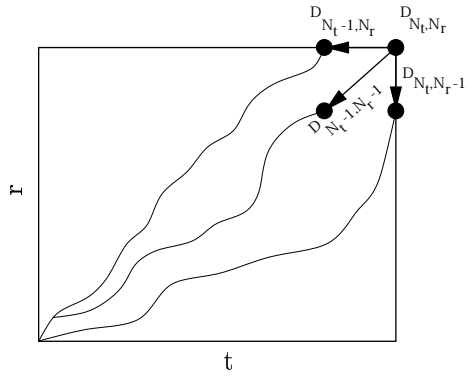


Abbildung 6.3: Rekursiver Aufbau der Viterbi-Matrix $\mathbf{D}^*$ mit den lokalen Pfadübergängen $\mathbb{P} = \{(1,0),(0,1),(1,1)\}$. Mithilfe des dynamischen Programmierungsansatz kann eine Berechnungskomplexität von $\mathcal{O}\left(N_\mathrm{t} \cdot N_\mathrm{r}\right)$ erreicht werden.

Eine weitere Effizienzsteigerung kann erzielt werden, wenn in jedem Expansionsschritt nur die erfolgversprechendsten Pfade weiterverfolgt werden. Dies kann jedoch auch zu suboptimalen Lösungen führen, da einmal getroffene Entscheidungen nicht mehr revidiert werden (Greedy Prinzip). Im nächsten Abschnitt wird diese Technik genauer betrachtet.

6.2.3 Strahlsuche

Zur weiteren Reduktion der Berechnungskomplexität wird die so genannte Strahlsuche (engl. *beam search*), wie es zum Beispiel in Schukat-Talamazzini [45] beschrieben ist, eingesetzt. Hierzu werden in einem Expansionsschritt nur solche Pfade weiterverfolgt, die sich innerhalb einer festgelegten Strahlbreite bewegen. Alle anderen Pfadhypothesen werden unwiderruflich verworfen. Durch diese Strategie müssen weniger Pfade betrachtet werden, man geht jedoch das Risiko ein, einen suboptimalen Gesamtpfad zu generieren. Dieser Fall tritt genau dann ein, wenn der optimale Pfad zu Anfang in Expansionsrichtung höhere akkumulierte Distanzen erzeugt, wodurch er aus dem Berechnungsschema herausfällt. Entsprechend der Symmetrie der gewählten lokalen Pfadübergänge $\mathbb{P}$ erfolgt die Berechnung innerhab der Viterbimatrix diagonalenweise. Hierdurch ist gesichert, dass jeder Teilpfad gleich weit vom angestrebten Endpunkt $(N_\mathrm{t}, N_\mathrm{r})$ entfernt liegt und ein Vergleich der aktiven Hypothesen untereinander zulässig ist. Innerhalb der n-ten Diagonalen wird das Strahlkriterium wie folgt bezüglich der kleinsten akkumulierten Pfaderweiterungsdistanz $s_n = \min\limits_{i+j=n} D^*_{i,j}$ definiert:

$$D^*_{i,j} < s_n + \text{beam} \quad \text{für } n = i + j$$

Alle Pfaderweiterungen, deren akkumulierte Distanz $D^*_{i,j}$ nicht innerhalb der Strahlbreite liegen, werden verworfen. Die Konstante beam wird vor Berechnungsbeginn fest gewählt und hängt zumeist von den gewählten Merkmalen und deren Wertebereich ab. Abbildung 6.4 zeigt die

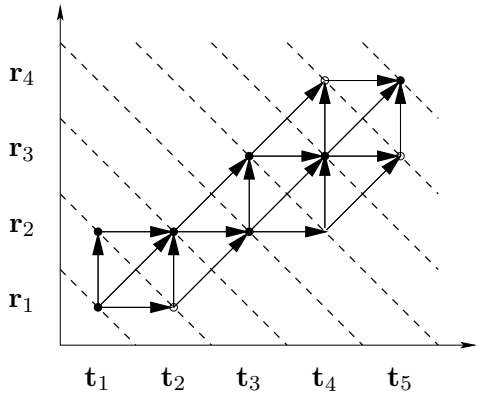


Abbildung 6.4: Vorwärtsgerichtete pfadsynchrone Strahlsuche anhand der Diagonalen in der Viterbimatrix. Es werden nur Pfadhypothese verfolgt, die ausgehend von der kleinsten akkumulierten sich innerhalb einer festgelegten Strahlbreite befinden.

vorwärtsgerichtete diagonalenweise Strahlsuche. Durch diese Suchstrategie kann die Berechnungskomplexität weiter auf $\mathcal{O}\left(|\mathbb{P}|\,\tilde{N}\right)$ verringert werden.

6.3 Modellierung der Referenzmodelle

Bis jetzt wurde die Annahme getroffen, dass es sich bei den Referenzmustern ebenfalls um eine Merkmalsvektorfolge handle. Diese Annahme genügte zur Beschreibung des DTW Verfahrens vollkommen aus. Da nun das Verfahren bekannt ist, kann eine genauere Beschreibung zur Generierung der Referenzmodelle stattfinden.

6.3.1 Generierung von Allographen

In der Handschrift, wo große Varianzen innerhalb gleicher Einzelzeichen (*Zeichenklasse*) vorliegen, kann es von Vorteil sein, die Modelle auf einer höheren Ebene zu partitionieren, bevor sie zu einem Modell zusammengefasst werden. Würde man dies nicht tun, bekäme man ein Modell, das zu allgemeine Parameter besäße und somit für eine Klassifikation unbrauchbar wäre. Durch eine Partitionierung, auch *Clustering* genannt, können innerhalb einer Zeichenklasse ähnliche Ausprägungen gefunden und diese zu einem Cluster zusammengefasst werden. Anschließend kann zu jedem Cluster ein Referenzmodell generiert werden. Eine Zeichenklasse, wie zum Beispiel die des Buchstabens 'b', wird somit durch mindestens ein Referenzmodell beschrieben. Will man jedoch genügend Schreibvariationen abdecken, so sind zumeist mehrere Modelle (*Allographen*) notwendig.

Zur Bestimmung ähnlicher Trainingsdatensätze kommt das zuvor beschriebene DTW Verfahren zum Einsatz. Um eine ganzheitliche Behandlung mit den später gezeigten Klassifikatordistanzen zu gewährleisten, wird eine andere lokale Distanz als die von Gleichung 6.1 verwendet. In der

Arbeit von Bockhorn [6] wurde hierzu eine entsprechende 'Distanzfunktion' $\tilde{d}:\mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}$ entwickelt. Sie ermöglicht es, die geschätzten Statistiken in den Referenzmodellen zu berücksichtigen:

$$\begin{aligned}\tilde{d}\left(\mathbf{t}_i, \mathbf{r}_j\right) &= \frac{F}{2}\ln\left(2\pi\right) + \frac{1}{2}\ln\left(|\mathbf{\Sigma}|\right) \\ &\quad + \frac{1}{2}\left(\mathbf{t}_i - \mathbf{r}_j\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{t}_i - \mathbf{r}_j\right) \\ &\quad + \ln\left(|\mathbb{P}|\right)\end{aligned}$$

F bezeichnet die Dimension der Merkmalsvektoren, $\mathbf{\Sigma}$ die $F \times F$-dimensionale Kovarianzmatrix beziehungsweise $|\mathbf{\Sigma}|$ die Determinante der regulären Kovarianzmatrix. $|\mathbb{P}|$ ist die Kardinalität der Menge der lokalen Pfadübergänge, welche in 6.2.1.3 genauer beschrieben wurden. Wie gezeigt werden kann, handelt es sich bei dieser Funktion nicht um eine Metrik im mathematischen Sinne. So gilt:

$$\tilde{d}(\mathbf{t}_i, \mathbf{r}_j) = 0 \Leftrightarrow \mathbf{t}_i = \mathbf{r}_j,$$

$$\tilde{d}(\mathbf{t}_i, \mathbf{r}_j) = \tilde{d}(\mathbf{r}_j, \mathbf{t}_i)$$

aber nicht die Dreiecksungleichung

$$\tilde{d}(\mathbf{t}_i, \mathbf{r}_j) \leq \tilde{d}(\mathbf{t}_i, \mathbf{z}) + \tilde{d}(\mathbf{z}, \mathbf{r}_j)$$

Abbildung 6.5 zeigt fünf Buchstaben der Zeichenklasse 'b' mit dem zugehörigen *Unähnlichkeitsdendrogramm.*

Ähnliche Schriftproben wurden zu Clustern zusammengefasst. Je nach Schnitt durch das Dendrogramm ergibt sich eine unterschiedliche Anzahl an Clustern. Die Grenze, ab welcher ein Schnitt durch das Dendrogramm erfolgt, wird durch einen zuvor festgelegten Wert D_{max} bestimmt und ist für alle Zeichenklassen gültig. Um eine robuste Parameterschätzung zu garantieren, werden Cluster mit zu kleinen Kardinalitäten verworfen. Weitere Details lassen sich in Bahlmann and Burkhardt [3] finden.

6.3.2 Statistische Modellierung eines Allographen

Wie schon in der Einleitung zu diesem Kapitel vorweggenommen wurde, arbeitet das zugrunde liegende Erkennungssystem mit *statistischen* Referenzmodellen. Das bedeutet, dass die beim Training erzeugten Referenzmuster zusätzliche statistische Merkmale besitzen. Durch das Clustering ist es nun möglich robust statistische Parameter zu schätzen und diese für die Klassifikation zu verwenden. Zur Visualisierung dieses Sachverhaltes eignet sich zum Beispiel das Merkmal der normierten Ortskoordinaten $\mathbf{P}_k$, wie sie in Kapitel 5 beschrieben wurden. Abbildung 6.6 zeigt mehrere übereinandergelegte Ziffern eines Clusters der Zeichenklasse '3'. Es sind deutlich die unterschiedlichen Verteilungen der Merkmale (hier Stiftbewegungen) zu erkennen. Im rechten Bild ist das aus den gemittelten Merkmalen generierte Referenzmodell für

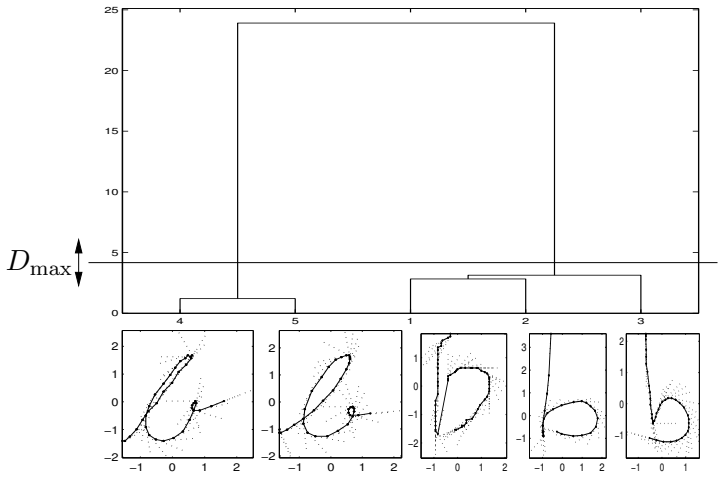


Abbildung 6.5: Unähnlichkeitsdendrogramm für fünf Schriftproben der Zeichenklasse 'b'. Ähnliche Schriftproben sind zu Cluster zusammengefasst. Je nach Wahl des Schwellwertes für den Abstand der Cluster ergibt sich eine feinere beziehungsweise gröbere Clusterunterteilung. Der hier gewählte Clusterschwellwert teilt die Zeichenklasse in zwei Cluster ein, wodurch sie später durch zwei Allographen repräsentiert werden.

die Ziffer 3 dargestellt. Jedoch sollen nun nicht nur die mittleren Merkmale extrahiert werden, sondern zusätzlich die Streuung eines Merkmals in dessen Umgebung. Zu diesem Zweck wird eine Gaußsche Normalverteilung der Merkmale angenommen, was, wie in [6] gezeigt werden konnte, für die Merkmale durchaus zutreffend ist. Anhand dieser Verteilungsfunktion kann nun der Eigenvektor und dessen Eigenwert der Kovarianzmatrix für korrespondierende Merkmalvektoren bestimmt werden. Die rechte Zeichnung veranschaulicht diesen Sachverhalt indem zu jedem eingezeichneten Merkmalsvektor dessen Streuungshauptrichtung und Streuungsstärke (durch eine Gerade visualisiert) eingezeichnet wurde.

Sei $\mathbb{C}^{lk}$ das k-te Cluster der Zeichenklasse $l \in \{1, \ldots, L\}$ für $k \in \{1, \ldots, K_l\}$, und bezeichne R^{lk} das zugehörige Referenzmodell, welches aus M_l Trainingsmuster generiert werden soll, so wird für die statistische Modellierung anstatt von $\mathrm{r}^{lk} = \left[\mathbf{r}_1^{lk}, \ldots, \mathbf{r}_{N_\mathrm{r}}^{lk}\right]$ eine Sequenz von $\mathrm{R}^{lk} = \left[\mathbf{R}_1^{lk}, \ldots, \mathbf{R}_{N_{\mathrm{R}^{lk}}}^{lk}\right]$ von Tupeln $\mathbf{R}_j^{lk} = \left(p_j^{lk}, P_j^{lk}\right)$ genommen. Dabei bezeichnet:

1. eine stetige Wahrscheinlichkeitsdichtefunktion (*Probability Density Function*) $p_j^{lk} : \mathbb{R}^F \rightarrow \mathbb{R}$, welche die Verteilung der Merkmale im Abtastpunkt j bestimmt und

2. eine diskrete Übergangswahrscheinlichkeit $P_j^{lk} : \mathbb{P} \rightarrow [0, 1]$, ausgehend vom Abtastpunkt j.

Für die Berechnung der Wahrscheinlichkeitsdichtefunktion wurde eine F-dimensionale multi-

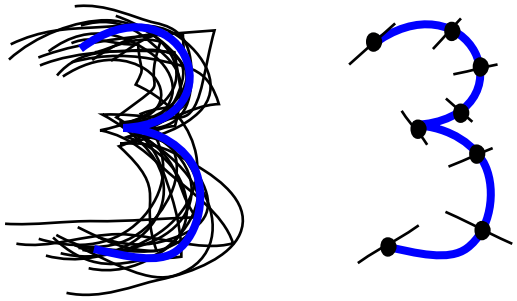


Abbildung 6.6: Links sieht man mehrere Zeichen der Zeichenklasse '3' anhand ihrer normierten Ortskoordinaten $\mathbf{P}_k$ übereinander gelegt. Es sind deutlich die Varianzen der Merkmale zu erkennen. Rechts davon ist das aus den gemittelten Merkmalen generierte Referenzmodell dargestellt. Es wurden zusätzlich zu einigen Merkmalsvektor die Hauptrichtung und Stärke der Streuung durch eine Gerade visualisiert.

variate Gaußverteilung benutzt.

$$p_j(\mathbf{x}) = \mathcal{N}_{\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j}(\mathbf{x}) = \frac{1}{(2\pi)^{F/2}\,|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right)$$

Hieraus ergibt sich die neue Merkmalsvektorfolge durch $\mathrm{R}^{lk} = \left[\mathbf{R}_1^{lk}, \ldots, \mathbf{R}_{N_{\mathrm{R}^{lk}}}^{lk}\right]$ mit der statistischen Größe $\mathbf{R}_j^{lk} = \left(p_j^{lk}, P_j^{lk}\right)$ für $1 \leq j \leq N_{\mathrm{R}^{lk}}$. Zur Berechnung dieser Modellparameter wird ein iteratives Verfahren angewandt, welches in abgeänderter Form als *Viterbi-Training* oder *segmentweise K-means* bekannt ist, siehe hierzu Juang and Rabiner [23] und Schukat-Talamazzini [45]. Für weitere Details wird auf Bockhorn [6] verwiesen.

6.4 CSDTW

Zur Klassifikation muss somit ein Merkmalvektorsequenz $\mathrm{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathrm{t}}]$ mit einer Menge von statistischen Referenzmodellen $\mathrm{R}^{lk} = \left[\mathbf{R}_1^{lk}, \ldots, \mathbf{R}_{N_{\mathrm{R}^{lk}}}^{lk}\right]$ verglichen werden. Die konsequente Erweiterung des DTW Verfahrens stellt das so genannte *cluster-generative statistical* DTW (*CSDTW*) dar. Hierzu wird die lokale Distanzfunktion d durch folgende Funktion ersetzt:

$$\begin{aligned}\hat{d}(\mathbf{t}_i, \mathbf{R}_j) \;=\; & \tfrac{F}{2}\ln(2\pi) + \tfrac{1}{2}\ln(|\boldsymbol{\Sigma}_j|) \\ & + \tfrac{1}{2}(\mathbf{t}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{t}_i - \boldsymbol{\mu}_j) \\ & - \ln(P_j(\Delta\boldsymbol{\phi}))\end{aligned}$$

Die Viterbidistanz ergibt sich nun durch

$$D^*[\hat{d}]\,(\mathrm{t},\mathrm{R}) = D_{\mathbf{\Phi}^*}[\hat{d}]\,(\mathrm{t},\mathrm{R}) = \min_{\mathbf{\Phi}} \left\{ D_{\mathbf{\Phi}}[\hat{d}]\,(\mathrm{t},\mathrm{R}) \right\}.$$

Man kann zeigen Bockhorn [6], dass $D^*[\hat{d}]\,(\mathrm{t},\mathrm{R})$ folgender Gleichung genügt:

$$D^*[\hat{d}]\,(\mathrm{t},\mathrm{R}) = -\log\left(P\,(\mathrm{t},\mathbf{\Phi}^*|\mathrm{R})\right)$$

wobei $P\,(\mathrm{t},\mathbf{\Phi}^*|\mathrm{R})$ die pfadoptimierte Wahrscheinlichkeit bezüglich des Viterbipfades $\mathbf{\Phi}^*$ist.

6.5 Zusammenfassung

Ausgehend vom DTW Verfahren wurde das CSDTW Verfahren erläutert. Diese Verfahren stellt den Kern des Klassifikators dar. Es ermöglicht eine Distanzberechnung unter Berücksichtigung von statistischen Merkmalsgrößen und konnte bereits erfolgreich für die Einzelzeichenerkennung Bahlmann and Burkhardt [3] eingesetzt werden.

KAPITEL 7

Klassifikation von Zeichensequenzen

Diese Kapitel widmet sich der Erkennung handgeschriebener Worte anhand von trainierten Zeichenmodellen. Abbildung 7.1 zeigt diesen Sachverhalt am Beispiel des Wortes 'in'. Hierzu werden die Referenzmodelle während des Erkennungsprozesses von links nach rechts an das zu erkennende Wort angelegt. Gesucht ist die ähnlichste *Referenzmustersequenz.* Die mathematische Grundlage bildet die statistische Entscheidungstheorie auf Basis der Bayes'schen Formel. Bezeichne $\Re$ die Menge aller Referenzmuster, so wird zu einer gegebenen Beobachtung t dasjenige *Superreferenzmuster* $R^{\tau,s} := \mathrm{R}_{\tau(1)} \oplus ... \oplus \mathrm{R}_{\tau(s)}$ mit unbekannter Sequenzlänge $s \in \mathbb{N}$ und Abbildungsfunktion $\tau : \mathbb{N} \rightarrow 1, .., |\Re|$ gesucht, das die maximale a-posteriori Wahrscheinlichkeit $P(R^{\tau,s} \mid \mathrm{t})$ besitzt:

$$\widehat{R}^{\tau,s} = \arg\max_{R^{\tau,s}} P(R^{\tau,s} \mid \mathrm{t})$$

Da in der Regel die a-posteriori Wahrscheinlichkeit $P(R^{\tau,s} \mid \mathrm{t})$ für alle Referenzmustersequenzen nicht vorliegt, werden folgende Umformungen durchgeführt:

$$\begin{aligned} \arg\max_{R^{\tau,s}} P(R^{\tau,s} \mid \mathrm{t}) &= \arg\max_{R^{\tau,s}} \frac{P(\mathrm{t}, R^{\tau,s})}{P(\mathrm{t})} \\ &= \arg\max_{R^{\tau,s}} \frac{P(R^{\tau,s}) P(\mathrm{t} | R^{\tau,s})}{P(\mathrm{t})} \end{aligned}$$

Der Nenner ist konstant und somit irrelevant bezüglich der Maximabildung. Somit lautet das zu optimierende Kriterium:

$$\widehat{R}^{\tau,s} = \arg\max_{R^{\tau,s}} P(R^{\tau,s}) P(\mathrm{t} \mid R^{\tau,s})$$

In diesem Ausdruck bezeichnet $P(R^{\tau,s})$ die a-priori-Wahrscheinlichkeit der zu erkennenden Referenzmustersequenz. Diese Wahrscheinlichkeit ist unabhängig vom Testmuster und beschreibt das linguistische Wissen über die Sprache hinsichtlich einer Zeichenfolge $R^{\tau,s}$. $P(\mathrm{t} \mid R^{\tau,s})$ stellt die bedingte Wahrscheinlichkeit oder auch Wahrscheinlichkeitsdichte dar und liefert zu einer beliebigen Referenzmusterfolge die Wahrscheinlichkeit, dass das Testmuster vorlag.

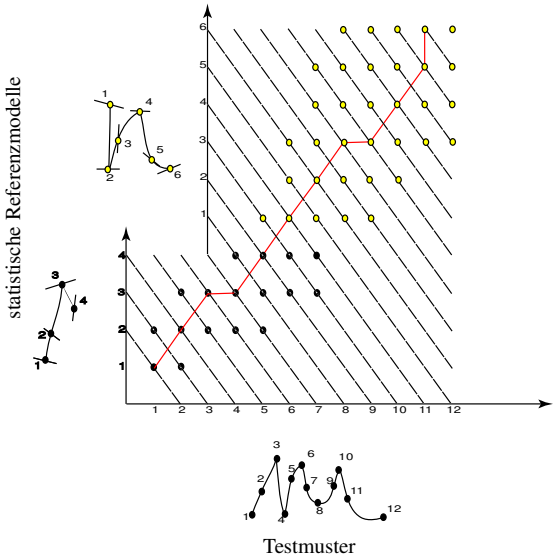


Abbildung 7.1: Worterkennung anhand von Einzelbuchstaben als Referenzmuster.

Aufgrund der Tatsache, dass bei einer Kardinalität von $|\Re| = 1000$, was bei einem Zeichensatz, bestehend aus Zahlen, Groß-, Kleinbuchstaben und Sonderzeichen, durchaus realistisch ist und einer mittleren Wortlänge von 10 Buchstaben, theoretisch 10^{1000} Referenzmustersequenzen zu evaluieren wären, ist es notwendig Suchstrategien zu entwerfen, die den Suchraum einschränken. Hierzu erfolgt die Suche nach der wahrscheinlichsten Zeichenfolge unter Zuhilfenahme von linguistischem Wissen über die zugrunde liegende Sprache.

Im vorliegenden Fall handelt es sich um ein Wörterbuch, das in einem so genannten *Lexikon Trie* abgespeichert wird. Zusätzlich könnte die Auftretenswahrscheinlichkeit eines Wortes in der Sprache hinzugezogen werden. Da diese Information jedoch nicht zur Verfügung steht, werden alle Worte als gleichwahrscheinlich angesehen. Die wissensbasierte Suche gehört zu den wichtigsten Mechanismen bei der effizienten Einschränkung des Hypothesenraumes. Ein entscheidender Nachteil der Lexikon basierten Suche liegt darin, dass nur Wörte erkannt werden können, die sich auch im Lexikon befinden. Eine Lösung dieses Problems stellen n-Gramme dar, auf die jedoch hier nicht weiter eingegangen wird.

7.1 Wissensquelle Lexikon Trie

Eine entscheidende Reduktion des Suchraumes lässt sich durch die Zuhilfenahme eines baumorganisierten Lexikons erzielen. Eine typischer Aufbau eines Lexikonbaums (*lexicon trie*) ist die Speicherung der Buchstaben im Innern der Knoten (siehe Abbildung 7.2). Beginnend beim Wurzelknoten ergibt sich ein Wort durch die Konkatenation der Einzelzeichen bis zum Erreichen eines Endknotens. Alle Blätter des Baumes sind Endknoten, aber auch innere Knoten kön-

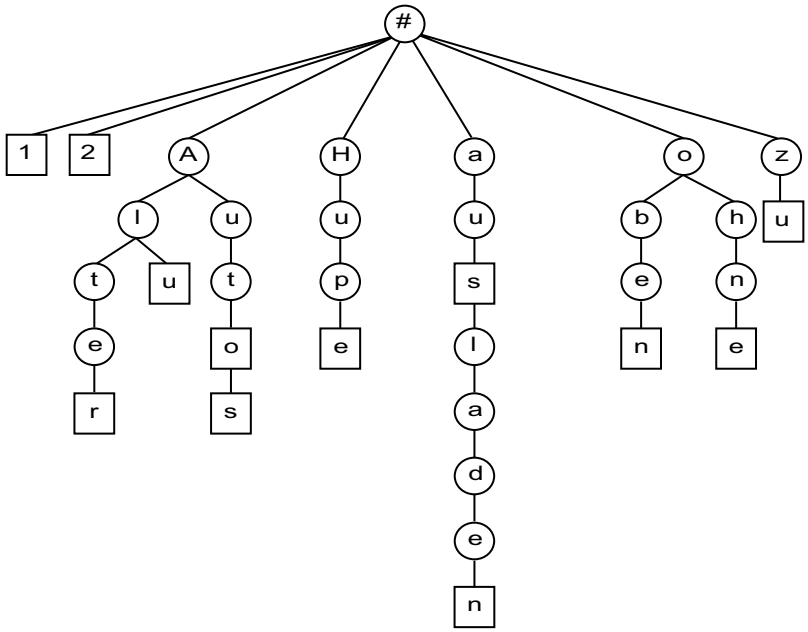


Abbildung 7.2: Dargestellt ist ein so genannter Lexikon Trie. Beginnend bei der Wurzel ergibt sich ein Wort durch die Konkatenation der Einzelzeichen bis zum Erreichen eines Endknotens, hier dargestellt durch ein Rechteck. Die Abspeicherung des Wörterbuches in einer Baumstruktur ermöglicht das Einsetzen effizienter Pruning Techniken bei der Viterbisuche auch für große Lexika.

nen Endknoten darstellen. Hierdurch ist es unter anderem möglich, dass ein Präfix eines Wortes bereits ein ganzes Wort repräsentiert.

Zu Beginn einer Klassifikation werden alle Referenzmuster aktiviert, die direkte Nachfolger der Wurzel des Baumes sind. Genauer, sei 'A' ein von der Wurzel durch eine Kante erreichbarer Knoten, so werden alle Allographen von 'A' als aktive Hypothesen initialisiert. Bei großem Vokabular bedeutet dies, dass zumeist alle in der Referenzmenge $\Re$ enthaltenen Referenzmuster aktiviert werden. Eine Initialisierungsmenge, welche einen enormen Berechnungsaufwand zu Beginn des DTW Prozesses darstellt. Bezogen auf den Einsatz von Real-World Anwendungen, mit einer Lexikongröße von 25000 Wörtern, kann dieser Berechnungsaufwand nur über eine kurze Zeit betrieben werden.

Somit müssen zusätzlich effiziente Suchtechniken eingesetzt werden, um eine schnelle Antwortzeit gewährleisten zu können. Diese Techniken sollten weitestgehend unabhängig von der Lexikongröße sein und die Erkennungsrate nicht gravierend beeinflussen. Eine Beschneidung des Hypothesenraumes (*pruning*), die sich bewähren konnte, wird in Manke et al. [30] beschrieben und kommt auch in dieser Arbeit zum Einsatz (siehe hierzu Abschnitt 7.3).

7.2 Berechnung der Viterbidistanz

Für die Worterkennung müssen Anpassungen vorgenommen werden, die es ermöglichen, innerhalb einer Referenzmustersequenz die Viterbidistanz bestimmen zu können. Die eigentliche Erweiterung stellt die Modellierung der Übergänge zwischen den Referenzmustern dar. Einige der ersten Ansätze zur Lösung des Worterkennungsproblems mittels DTW stammen aus der Spracherkennung und wurden von Sakoe [39] aufgezeigt.

Zwei Jahre später wurde der so genannte *Level-Building-Algorithmus* (*LBA*) von Myers und Rabiner [31] vorgestellt. Sie schlugen eine stufenweise Berechnung der Viterbimatrix vor, wobei zu vorab bestimmten Stufen alle aktiven Hypothesen berechnet und nur die n-besten Viterbipfade bezüglich jeder Stufe weiterverfolgt werden. Ein Nachteil des Verfahrens liegt in der vorherigen Bestimmung der Anzahl von Levels, bezüglich derer eine Art Zwischenbilanz der Viterbipfade bestimmt werden soll und der Rückwärtssuche am Ende des Berechnungsvorgangs um den global besten Pfad bestimmen zu können.

Bridle et al. [8] und Nakagawa [32] stellten innerhalb eines Jahres unabhängig voneinander ein Verfahren vor, das die Schwächen des Level-Building-Algorithmus beseitigte. Es handelt sich hierbei um den *One-Stage-Algorithmus* bei dem keine Rücksprünge bezüglich dem Testmuster notwendig sind.

Ney [34] präsentierte 1994 schließlich eine einfache Umsetzung des One-Stage-Algorithmus, welcher auch die Grundlage dieser Arbeit bildet. Bei diesem Verfahren kann die Berechnung der Viterbimatrix in einem Durchgang erfolgen. Durch die Einhaltung des Optimalitätsprinzips setzt sich die Viterbidistanz wiederum aus den akkumulierten Viterbidistanzen nun eines jeden Referenzmusters innerhalb der Referenzmustersequenz zusammen. Wurden die Übergänge korrekt modelliert, so kann die Referenzmustersequenz als ein komplettes Superreferenzmuster betrachtet werden, womit die Berechnung der Viterbidistanz auf den Vergleich zweier Buchstaben zurückgeführt ist.

7.2.1 Dynamisches Auffinden der Zeichenübergänge

Zur Beschreibung der Übergangsmodellierung wird im Weiteren davon ausgegangen, dass es sich beim Testmuster $\mathrm{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathrm{t}}]$ mit $\mathbf{t}_i \in \mathbb{R}^F$ für $1 \leq i \leq N_\mathrm{t}$ um ein Wort handelt. Der einfacheren Notation zuliebe sei angenommen, dass jede Zeichenklasse l nur durch einen Allographen repräsentiert wird, wodurch der Allographenindex entfallen kann. Dementsprechend bezeichne $\mathrm{R}^{l_{k_0}} = \left\{ \mathbf{R}_1^{l_{k_0}}, ..., \mathbf{R}_{N_{\mathrm{R}^{l_{k_0}}}}^{l_{k_0}} \right\} \in \Re$ das Referenzmodell zur Zeichenklasse $l_{k_0} \in \{1, \ldots, L_{k_0}\}$, aus der Menge der direkten Nachfolge-Zeichenklassen des Wurzelknotens k_0.

Beginnend beim ersten Viterbi-Matrixelement $(1, 1)$ erfolgt nun ein Vergleich beider Muster. Dieser Anfangsvergleich entspricht somit der in Abschnitt 6.2.1.1 geforderten Anfangsbedingung.

Zur Berechnung der Viterbimatrix kommt das in Abschnitt 6.2.3 beschriebene Strahlsuchverfahren zum Einsatz. Hierdurch werden nur solche Pfade Φ expandiert, die sich innerhalb einer

vorgegebenen Strahlbreite bewegen. Erreicht ein Pfad innerhalb des Strahlkriteriums das letzte Element des Referenzmusters $\phi_{\mathrm{R}^{l_{k_0}}}(n) = N_{\mathrm{R}^{l_{k_0}}}$, so wird das Matrixelement, welches durch das Tripel $\left(l_{k_0}, i, N_{\mathrm{R}^{l_{k_0}}}\right)$ eindeutig bestimmt werden kann, als potentielles *Übergangselement* markiert. Durch die diagonalenweise Entwicklung der Viterbimatrix und der Wahl der lokalen Pfadübergänge $\mathbb{P} = \{(1,0),(0,1),(1,1)\}$ kann das markierte Übergangselement maximal zwei Diagonalen von der aktuellen Diagonalen m entfernt liegen. Somit wird es spätestens nach zwei Diagonalen durch die Viterbisuche aktiviert. Durch seine Aktivierung kommt es zur eigentlichen Modellierung des Übergangs in ein neues Referenzmuster.

Während beim Buchstabenvergleich vom Matrixelement $\left(l_{k_0}, i, N_{\mathrm{R}^{l_{k_0}}}\right)$ nur noch der lokale Pfadübergang $(1,0)$ gewählt werden konnte, da ein Überschreiten der Merkmale in Referenzmusterrichtung nicht zulässig war, erfolgt nun eine Suchraumerweiterung gemäß dem Lexikonbaum. Sei $k_1^{l_{k_0}}$ der Knoten der Zeichenklasse l_{k_0} auf dem ersten Level im Lexikonbaum, so müssen alle Referenzmodelle, deren Zeichenklasse direkte Nachfolger des Knoten $k_1^{l_{k_0}}$ sind, als mögliche Übergangsmodelle betrachtet werden. Sei etwa $\mathrm{R}^{l_{k_1}}$ mit $l_{k_1} \in \{1,\ldots,L_{k_1}\}$ eine beliebige Erweiterung aus dieser Menge, so werden ausgehend von der akkumulierten Distanz $D^*_{l_{k_0},i,N_{\mathrm{R}^{l_{k_0}}}}$ die lokalen Pfadübergänge $\{(0,1),(1,1)\}$ berechnet, mittels derer die Gitterpunkte $\left(\mathbf{t}_i, \mathbf{R}_1^{l_{k_1}}\right)$ und $\left(\mathbf{t}_{i+1}, \mathbf{R}_1^{l_{k_1}}\right)$ in der Erweiterungsmatrix erreicht werden können.

$$\left(l_{k_0}, i, N_{\mathrm{R}^{l_{k_0}}}\right) \overset{\{(1,0),(1,1)\}}{\longrightarrow} \left\{ \begin{array}{c} (l_{k_1}, i, 1) \\ (l_{k_1}, i+1, 1) \end{array} \right\}$$

Befindet sich ein Pfadübergang innerhalb des Strahlkriteriums, so erfolgt eine Initialisierung der Hypothese. Falls dies nicht der Fall ist, wird eine weitere Überprüfung bezüglich eines späteren Übergangselement durchgeführt.

Abbildung 7.3 stellt diesen Sachverhalt noch einmal grafisch dar. Das Übergangselement $(l_{k_0}, 4, 4)$ der Viterbimatrix kann von der sechsten Diagonale durch den lokalen Pfadübergang (1,1) erreicht werden. Nach zwei Diagonalen wird es aktiviert, wodurch die lokalen Pfadübergänge $\{(0,1),(1,1)\}$ ins neue Referenzmuster $\mathrm{R}^{l_{k_1}}$ berechnet werden. Unter der Annahme, dass diese Pfadübergänge sich nicht innerhalb des Strahlkriteriums befinden (gestrichelte Übergänge), findet zu diesem Zeitpunkt keine Initialisierung des Erweiterungsmusters statt. In der nächsten Diagonalen findet sich jedoch bereits ein neues Übergangelement $(l_{k_0}, 5, 4)$. Ausgehend von diesem kann mittels dem lokalen Pfadübergang $(1,1)$ eine akkumulierte Distanz erzeugt werden, welche sich innerhalb des Strahlkriteriums befindet (durchgezogene Linie). Hierdurch kommt es zur Initialisierung der Hypothese $\mathrm{R}^{l_{k_1}}$.

Wurde die Hypothese erweitert, so wird die Referenzmustersequenz als ein Superreferenzmuster $R^{l_{k_0},l_{k_1}} = \mathrm{R}^{l_{k_0}} \oplus \mathrm{R}^{l_{k_1}}$ betrachtet. Das Zusammenfügen einzelner Referenzmuster zu einem Superreferenzmuster erfolgt solange, bis alle Merkmalsvektoren des Testmusters betrachtet wurden, beziehungsweise ein Blatt im Lexikonbaum erreicht wurde.

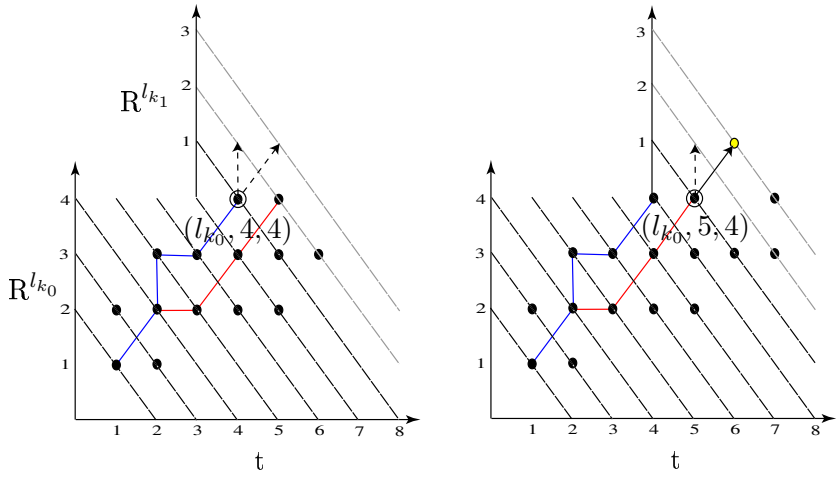


Abbildung 7.3: Modellierung des Übergangs zwischen zwei Referenzmustern. Ausgehend vom Übergangselement $(l_{k_0}, 4, 4)$ werden die lokalen Pfadübergänge $\{(0,1),(1,1)\}$ ins neue Referenzmuster $\mathrm{R}^{l_{k_1}}$ berechnet. Da jedoch beide Übergänge außerhalb der Strahlbreite liegen (gestichelte Übergänge), findet keine Initialisierung des Referenzmusters $\mathrm{R}^{l_{k_1}}$ zu diesem Zeitpunkt statt. Erst für das nächste Übergangselement $(l_{k_0}, 5, 4)$ erzeugt der lokale Übergang $(1,1)$ eine Distanz, welche innerhalb der Strahlbreite liegt, was zu einer Aktivierung von $\mathrm{R}^{l_{k_1}}$ führt.

7.2.2 Erweiterte Betrachtung der Übergangsmodellierung

Das bisherige Verfahren zur Übergangsmodellierung arbeitet mithilfe der lokalen Pfadübergänge wie sie auch innerhalb der Viterbimatrix Anwendung finden. Betrachtet man jedoch die Zeichenübergänge innerhalb eines Wortes genauer, so befinden sich je nach Schreibstil Verbindungsstriche (*Ligaturen*) zwischen den Zeichen. Sie entstehen, wenn der Schreiber ohne Absetzen des Stifts ein Teil oder das gesamte Wort fließend schreibt. Beim Training können solche Ligaturen nur dann gelernt werden, wenn ein entsprechender Datensatz von Einzelbuchstaben existiert. Dieser Datensatz stand jedoch zur Modellierung nicht zur Verfügung.

Ein Ansatz, der zur Lösung dieses Problems verfolgt werden könnte, ist die Modellierung von 'typischen' Referenzmodellen für Ligaturen. Sie dürften dann zwischen den Übergängen ohne eine Erhöhung der lokalen Distanz eingefügt werden. Der Zweck dieser Modelle bestünde somit nur darin, dass Ligaturen die Gesamtdistanz nicht belasten. Ein entscheidender Nachteil dieser Modellierung ist jedoch, dass hierdurch kleine Segmentabschnitte im Schriftzug unbeobachtet bleiben können, die gegebenfalls ein Einzelzeichen darstellen.

Um die Auslassung von Merkmalsvektoren besser kontrollieren zu können, wird eine anderer Ansatz verfolgt. Die Idee ist hierbei, dass ausgehend von einem Übergangelement nicht nur die lokalen Pfadübergänge $\{(0,1),(1,1)\}$ ins neue Referenzmuster erlaubt werden, sondern auch Übergänge der Art $\{(2,1),...,(k,1)\}$. Somit ist es möglich kontrolliert k Merkmalsvektoren aus dem Testmuster zu überspringen. Die Anzahl k der zu überspringenden Merkmalsvektoren im Testmuster ist natürlich abhängig vom jeweiligen Schreibstil und der Dichte der Neuabtastung.

Um eine realistische Auswahl für k während der Laufzeit bestimmen zu können, wird der Tangentensteigungswinkel θ ab dem aktuellen Merkmalsvektor des Testmusters betrachtet. Liegt eine starke Richtungsänderung vor, so wird $k = 2$ gewählt. Ansonsten wird k aus einem kleinen Framebereich gewählt, innerhalb dessen der Tangentensteigungswinkel relativ konstant bleibt, etwa $k \in \{3..10\}$. Dieses Vorgehen lässt sich damit begründen, dass Ligaturen zumeist den kürzesten Weg von einem Zeichen zum nächsten beschreiben. Im Idealfall eine Gerade, deren Tangentensteigungswinkel konstant ist, wohingegen starke Richtungsänderungen auf den Anfang eines neuen Buchstabens hindeuten.

Diese Idee wird auch häufig bei der Vorsegmentierung eines Wortes eingesetzt. Jedoch besitzt die hier gewählte analytische Vorgehensweise den Vorteil, dass der potentielle Zeichenübergang in einer Umgebung beginnend beim Übergangselement zu suchen ist.

7.2.3 Nachbetrachtungen Endpunktbedingungen DTW

Würden zwei 'gleichwertige' Muster miteinander verglichen, so könnte man durchaus die Forderung stellen, dass der Pfad beim jeweils ersten Merkmalsvektor beginnen und beim jeweils letzten Merkmalsvektor enden muss. Diese Restriktionen sind jedoch zu scharf, wenn es sich beim Referenzmuster um ein einzelnes Zeichen und beim Testmuster um ein Wort, bestehend aus einer Zeichenfolge, handelt. Hier kann die Anfangspunktbedingung nur für die erste Hypothese in einer Referenzmusterfolge gelten. Für jede weitere Hypothese sollte die Bedingung, wie zuvor beschrieben, gelockert werden.

Endpunktbedingungen werden überprüft, sobald die Referenzsequenz ein vollständiges Wort ergibt. In diesem Fall wird die Vollständigkeit des bisherigen Pfades überprüft und eine Relaxierung der Endpunktbedingung durchgeführt. Ziel ist es zwar den Endpunkt zu erreichen, jedoch kann über eine längere Referenzsequenz dieses Idealziel nicht immer eingehalten werden. Grund hierfür sind verzögerte Striche, wie zum Beispiel der i-Punkt oder t-Striche, die zumeist erst am Wortende gesetzt werden. Durch den Vergleich beider Muster von links nach rechts in der zeitlichen Historie können verzögerte Striche nur dann berücksichtigt werden, wenn sie sofort nach dem Schreiben der relevanten Buchstaben gesetzt werden. Einige, der in der Literatur beschriebenen Klassifikatoren umgehen dieses Problem, indem sie eine zweistufige Klassifikation durchführen. So beschreibt zum Beispiel Hu and Brown [19] ein Verfahren, welches in einem zweiten Klassifikationsdurchlauf auf der n-besten Hypothesenmenge eine detailliertere Untersuchung mit zusätzlicher Betrachtung von verzögerten Strichen durchführt, während im ersten Durchlauf das Testmuster im Vorverarbeitungsschritt weitestgehend von diesen befreit wurde.

7.3 Pruning Techniken

Zur weiteren Einschränkung des Suchraums werden so genannte Beschneidungstechniken (engl. *pruning*), wie sie in Schukat-Talamazzini [45] zu finden sind, in den Suchprozess integriert. Hierzu zählt insbesondere das Beschneiden auf Element- und Zeichenebene sowie die Limitierung der aktiven Elemente in der Viterbimatrix auf eine maximale Anzahl zulässiger Hypothesen

pro Zeitpunkt. Diese Pruningverfahren werden in vielen lexikonbasierten Worterkennungssystemen eingesetzt und gehören zu den so genannten Standard-Pruning-Methoden. Die Arbeit von Manke et al. [29] gibt einen guten Einblick in das Pruning auf Zeichenebene. In dieser Arbeit wurde eine eigene Pruningtechnik entwickelt, die alle drei Pruningebenen beinhaltet. Hierbei spielt der Lexikonbaum eine zentrale Rolle, wodurch dessen Knoten um zusätzliche Informationen erweitert werden müssen.

Zu Beginn der Klassifikation spielt das *Character-Pruning* eine wesentliche Rolle. Ist nur noch eine minimale Anzahl an Worthypothesen vorhanden, so beginnt das *Node-Pruning*. In dieser Phase dürfen keine weiteren Worthypothesen verworfen werden, jedoch werden innerhalb eines Knotens nur die besten Referenzmustersequenzen zu jeder Worthypothese weiterverfolgt.

Bezeichne im Weiteren

$$\mathcal{A}_n = \left\{ l \,\middle|\, \mathrm{R}^l \in \Re \text{ ist aktiv in der Diagonalen } n \right\}$$

die Menge der aktiven Referenzmuster und

$$\mathcal{O}_{l,n} = \{(l,i,j)\,|(i,j) \text{ ist aktiv in der Diagonalen } n\,\}$$

die Menge der aktiven Viterbimatrix Elemente des Referenzmusters R^l in der Diagonalen n. (Bemerkung: Betrachtet man ein beliebiges aktives Matrixelement (l,i,j), so gehören alle Elemente, deren Summe $i+j=n$ ist der n-ten Diagonalen an.)

Interessant für das Pruning ist nun die kleinste Viterbidistanz

$$\hat{s}_{l,n} = \min_{(l,i,j)\in\mathcal{O}_{l,n}\wedge(i+j=n)} D^*_{l,i,j}$$

, die innerhalb der Diagonalen n des Referenzmusters R^l erzielt werden konnte. Diese Distanz wird für jedes aktive Referenzmusters berechnet, wodurch sich schließlich die global kleinste akkumulierte Distanz bezüglich einer Diagonalen bestimmen lässt:

$$\hat{s}_n = \min_{l\in\mathcal{A}_n} \hat{s}_{l,n}$$

Sie stellt den Ausgangspunkt für das Pruning dar und bestimmt im Weiteren ob das jeweilige Referenzmuster in der nächsten Diagonalen aktiviert bleibt.

7.3.1 Character-Pruning

Zu Anfang der Klassifikation werden alle Allographen der jeweiligen Zeichenklassen, die direkte Nachfolgeknoten des Wurzelknotens sind, aktiviert. Für jeden aktiven Allographen werden die akkumulierten Distanzen zu den erreichbaren Matrixelementen innerhalb der Viterbimatrix berechnet. Hierbei werden nur solche Pfade weiterverfolgt, die dem Strahlsuchverfahren genügen.

Beim Übergang in die nächste Diagonale $n+1$ kommt es zur Überprüfung des Character-Pruning-Kriteriums. Es bleiben hierbei genau die Hypothesen l weiterhin aktiv, für die gilt:

$$\hat{s}_{l,n} < \hat{s}_n + \text{CharBeam}$$

Hierbei bezeichnet CharBeam eine Konstante, anhand derer die Beschneidungstoleranz eingestellt werden kann. Alle aktiven Hypothesen, die dieser Ungleichung nicht genügen, werden verworfen. Ein Einschnitt der besonders von den ersten Buchstaben im Wort abhängig ist, falls die Konstante CharBeam sehr klein gewählt wird. Erfolgt bereits hier ein Verwerfen der richtigen Hypothesen, so kann das ganze Wort nicht mehr korrekt erkannt werden. Dieser Fall tritt zumeist dann ein, wenn eine falsche Referenzlinienschätzung durchgeführt wurde. Auf der anderen Seite trägt eine frühzeitige Hypotheseneinschränkung zur Effizienz der Worterkennung bei.

7.3.2 Approximation aktiver Worthypothesen

Um die Anzahl aktiver Worthypothesen zum Diagonalenzeitpunkt n bestimmen zu können, wird jeder Knoten k im Lexikonbaum um einen Worthypothesenzähler w_k erweitert. Er gibt an, wie viele Worthypothesen sich unterhalb dieses Knotens befinden, beziehungsweise zählt die Anzahl der Endknoten, die vom aktuellen Knoten erreicht werden können. Für den Wurzelknoten entspricht der Zähler genau der Anzahl unterschiedlicher Wörter, die sich im Lexikon befinden.

Def: Ein Knoten im Baum heißt aktiv, wenn ein Allograph der entsprechenden Zeichenklasse des Knotens aktiv ist.

Eine erste Idee, die aktiven Worthyphothesen in der Diagonalen n zu berechnen, wäre die Summation über alle Worthypothesenzähler deren Knoten aktiv sind. Hierbei kommt es jedoch in folgenden Situationen zu Fehlern:

Während der Suche können sowohl der Vater als auch ein oder mehrere Söhne gleichzeitig aktiv sein. Diese Tatsache führt bei der oben genannten Methode zu Mehrfachzählungen von gleichen Worthypothesen zwischen dem Vater und dessen Söhnen. Denn jeder Sohnknoten im Baum enthält eine Untermenge von Worthypothesen des Vaters.

Ein effiziente Approximation der gesuchten aktiven Worthypothesen zum Zeitpunkt n, welche nur die aktiven Nachbarknoten betrachtet, ist die folgende:

Sei $\mathcal{K}_n$ die Menge aller aktiven Knoten zum Diagonalenzeitpunkt n, S_k die Menge der aktiven Söhne vom Knoten k und $\widehat{w}_n$ die gesuchte Anzahl aktiver Worthypothesen zum Zeitpunkt n somit gilt:

$$\sum_{k \in \mathcal{K}_n} w_k \geq \sum_{k \in \mathcal{K}_n} w_k - \sum_{s_k \in S_k} w_{s_k} \geq \widehat{w}_n$$

Hierbei können Mehrfachzählung durch Abziehen der Anzahl von Worthypothesen, die sich unterhalb des Kindknotens befinden, verhindert werden. Es gilt

$$\sum_{k \in \mathcal{K}_n} w_k - \sum_{s_k \in S_k} w_{s_k} = \widehat{w}_n$$

immer dann, wenn die aktiven Knoten innerhalb eines Zweiges eine 'Kette' bilden. Somit ist jeder tiefer liegende Knoten über den Sohn erreichbar und man zählt implizit nur den tiefsten

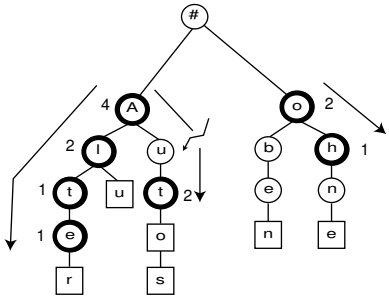


Abbildung 7.4: Approximative Bestimmung der aktiven Worthypothesen anhand aktiver Knoten im Lexikonbaum. Jeder Knoten enthält zusätzliche Information über die Anzahl der Wörter, die sich unterhalb des Knotens befinden (Worthypothesenzähler). Um feststellen zu können, wie viele Wörter zu einem Zeitpunkt aktiv sind, werden nur die Worthypothesenzähler der aktiven Knoten (schwarz unterlegt) betrachtet.

Knoten im Zweig. Der Fall einer Mehrfachzählung tritt genau dann ein, wenn beispielsweise der tiefste aktive Knoten keinen aktiven Vaterknoten besitzt, wodurch die Kette zu einem oberhalb liegenden aktiven Knoten unterbrochen ist. Abbildung 7.4 stellt diesen Sachverhalt nocheinmal grafisch dar. Die Worthypothesenzähler sind jeweils neben die Knoten geschrieben. Im Zweig der Worthypothesen 'Auto, Autos' kommt es durch den inaktiven Knoten 'u' zu einer Unterbrechung der Kette und somit zu einer Mehrfachzählung der beiden Wörter.

Ist nur noch eine Mindestanzahl an Worthypothesen aktiv, so wird kein Character-Pruning mehr durchgeführt. Alle noch aktiven Worthypothesen werden mittels einem Node-Pruning nur noch in ihren Referenzmustersequenzen eingeschränkt.

7.3.3 Node-Pruning

Ein Knoten im Lexikonbaum entspricht einer Zeichenklasse, da jede Zeichenklasse durch unterschiedlich viele Allographen repräsentiert wird, kommt es zu unterschiedlichen Referenzmustersequenzen für die gleiche Worthypothese. Das Node-Pruning bezeichnet das Beschneiden dieser unterschiedlichen Referenzmustersequenzen. Bezeichne $\mathcal{A}_{k,n}$ die Menge aller aktiven Allographen im Knoten k. So wird die kleinste akkumulierten Distanz im Knoten k wie folgt bestimmt:

$$\hat{s}_{k,n} = \min_{l \in \mathcal{A}_{k,n}} \hat{s}_{l,n}$$

Falls ein Superreferenzmuster das Node-Pruning-Kriterium

$$\hat{s}_{l,n} < \hat{s}_{k,n} + \text{NodeBeam}$$

nicht erfüllt, so wird es aus der aktiven Liste entfernt. Da immer ein minimales Superreferenzmuster pro aktiven Knoten existieren muss, werden keine Worthypothesen gelöscht.

7.3.4 Bemerkung zu den vorgestellten Pruningtechniken

Die hier vorgestellten Techniken passen sich automatisch an die Momentanqualität der besten Teillösungen innerhalb eines betrachteten Diagonalenzeitpunktes an. Ein entscheidender Nachteil bei der Wahl einer festen Pruningkonstanten liegt darin, dass bei dichtem Kandidatenfeld die Zahl der weiterhin aktiv gehaltenen Hypothesen unter Umständen groß bleibt. Um diesem Sachverhalt entgegenzusteuern, wird eine Kardinalitätsbeschränkung durchgeführt. Befindet sich die Anzahl aktiver Hypothesen oberhalb der festgesetzten Kardinalität, erfolgt ein temporäres Node-Pruning, wodurch die Allographenflut gesenkt wird. Führt diese Strategie zu keiner Reduktion der aktiven Hypothesen, so wird die Aktivenliste durch die Kardinalitätsbeschränkung beschnitten.

KAPITEL 8

Ergebnisse

Um die Leistungsfähigkeit des Systems testen zu können, wurde der UNIPEN Datensatz herangezogen. Da nur die Trainings-CD train_r01_v07 am Institut zur Verfügung steht, wurde sowohl das Training als auch die Klassifikation hierauf durchgeführt. Nachfolgend wird der Versuchsaufbau beschrieben.

8.1 Versuchsaufbau

8.1.1 Berechnung der Referenzmodelle

Das Training der Referenzmodelle fand innerhalb der Kategorien 1a, 1b, 1c, 2 und 3 statt. Hierzu wurde eine randomisierte Stichprobe generiert, die einen prozentualen Anteil der jeweiligen Kategorien ausmachte. Um die Schreiberunabhängigkeit des Systems zu garantieren, wurden die Trainingsdaten nur von Schreiber zusammengestellt, die keine Daten in der Kategorie 6 erstellt haben. Die Liste der relevanten Namesabkürzungen lautet: aga, apa, apb, app, ced, gmd, mot, syn, tos, uqb und val.

Diese Einzelzeichen besitzen keine Referenzlinieninformationen, die jedoch für die Berechnung des Merkmals der normierten y-Koordinate notwendig ist. Um diesem Sachverhalt entgegenzutreten wurden zwei Methoden verfolgt:

1. Die erste Methode führt eine automatische Referenzlinienbestimmung während des Trainings anhand des Labels durch. Der Nachteil dieser Methode liegt in der einheitlichen Bestimmung der Referenzlinie für gleiche Zeichen. So wurden beispielsweise die Referenzlinien (Grund- und Kernlinie) beim Buchstaben 'c' immer auf die beiden vertikalen

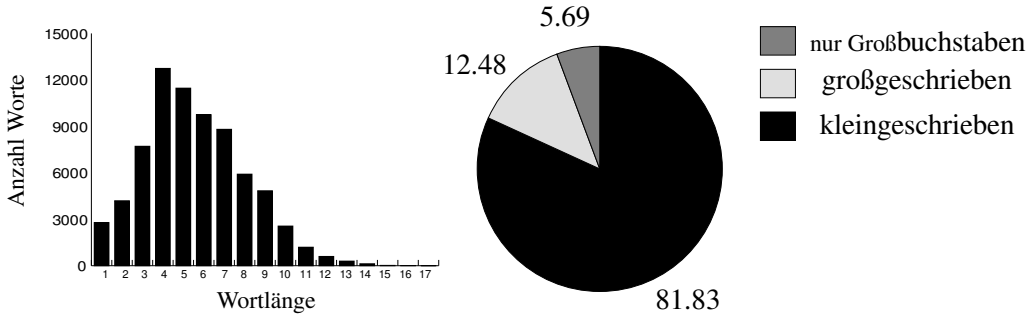


Abbildung 8.1: Wortlängenhistogramm der Kategorie 6 (isolated cursive or mixed-style words in mixed cases) des UNIPEN train_r01_v07 Datensatzes. Rechts davon ist die Verteilung der Worte bezüglich ihrer Groß- und Kleinschreibung zu sehen.

Extrema gesetzt. Hierdurch kann jedoch kein 'c' gelernt werden, dessen y-Koordinate aus der Kernzone herausragt, wodurch die geschätzten Modelle bezüglich des y-Merkmals im Randbereich keine Varianzen in Linienrichtung aufweisen. Außerdem entziehen sich einige Buchstaben einer einheitliche Referenzlinienbestimmung, wie beispielsweise bei der neudeutschen und altdeutschen Schreibweise des Buchstabens 'z'. Bedenkt man zudem, dass nicht alle Buchstaben innerhalb eines Wortes immer in den richtigen Linienabschnitten liegen, so ist das Training dieser Modelle zu idealisiert und führt teilweise zu großen Wiedererkennungsproblemen.

2. Bei der zweiten Methode werden die Modelle ohne eine Referenzlinienvorgabe trainiert. Es handelt sich hierbei um das gleiche Verfahren wie es zur Einzelzeichenerkennung verwendet wurde. Dadurch bleiben die Varianzen innerhalb gleicher Buchstaben im Grenzbereich erhalten. Die Referenzlinienschätzung wird erst nachträglich durchgeführt, indem die Modelle teils von Hand teils automatisch im y-Merkmal normiert werden. Hierbei bleibt zu beachten, dass beim Skalieren des y-Merkmals auch deren Varianzen mitskaliert werden müssen.

8.1.2 Zusammenstellung der Testmenge

Die Testdatensätze wurden nach dem gleichen Verfahren ausgewählt wie es in Hu et al. [21] beschrieben ist. Einziger Unterschied stellt die Verwendung des train_r01_v07 Datensatzes anstatt dem dort verwendeten devtest_r01_v02 Datensatz dar. Um eine adäquate Zusammenstellung der Testdaten durchführen zu können, wurde die Verteilungsstatistik der Kategorie 6 des train_r01_v07 Datensatzes berechnet. Abbildung 8.1 zeigt das Längenhistogramm und die Verteilung der Worte entsprechend ihrer Groß- und Kleinschreibung. Anhand dieser Statistik wurden drei Testdatensätze in den Größen 1500, 2000 und 3000 aus einer Teilmenge der Kategorie 6 zufällig zusammengestellt.

8.1.3 Zusammenstellung der Wörterbücher

Zu jedem der drei Testmengen wurde das entsprechende Wörterbuch anhand der Labels erzeugt. Fand zusätzlich eine Wortschatzerweiterung statt, so wurden hierzu zufällig aus dem Standard UNIX Wörterbuch weitere Wörter hinzugenommen.

8.1.4 Bewertungsmethoden

Die Klassifikationsrate berechnet sich aus dem Quotienten der Anzahl korrekt erkannter Muster und der Gesamtanzahl an Testmustern. Die Angabe erfolgt in Prozent:

$$\text{Top } 1 := C_{\text{corr}} = \frac{\#\text{Testmuster} - \#\text{Fehler}}{\#\text{Testmuster}} \cdot 100$$

Des Weiteren erfolgt eine Angabe der Klassifikationsrate unter Berücksichtigung der n-besten Treffer.

$$\text{Top N} := C_{\text{corr}}^{\text{N}} = \frac{\#\text{Treffer innerhalb der N-Besten}}{\#\text{Testmuster}} \cdot 100$$

Diese zusätzliche Angabe gibt einen guten Aufschluss darüber inwieweit eine zweite, intensivere Suche auf den n-besten Kandidaten weiteres Potential zur Steigerung der Erkennungsleistung liefert.

8.2 Klassifikationsergebnisse

In diesem Abschnitt werden die Erkennungsresultate vorgestellt. Zuerst findet eine Betrachtung der besten Ergebnisse unter allgemeinen Gesichtspunkten statt. Anschließend erfolgt eine spezifischere Sichtweise unter den Gesichtspunkten: Merkmale, Parametereinstellungen, Vorverarbeitung und der Identifizierung typischer Fehlklassifikationen.

Einen Überblick über die besten Erkennungsresultate zeigt Tabelle 8.1. Hierbei wurden die erstellten Testmengen bei unterschiedlichen Wörterbuchgrößen untersucht. Die Erweiterung des Vokabulars führte zu Erkennungsverlusten im Bereich der Top 1 wie auch der Top 10 Raten. Dass die Größe der Testbeispiele statistisch aussagekräftig ist, zeigten die geringen Abweichungen beim Übergang zum jeweils nächst größeren Datensatz. Die erzielten Resultate liegen weit unter den von Hu et al. [21] erzielten Ergebnissen (siehe zum Vergleich Tabelle 1.2). Hierbei ist jedoch zu bemerken, dass bei diesem System sieben verschiedene Merkmale gegenüber zwei Merkmalen zum Einsatz kamen und eine zweite, intensivere Suche auf den n-besten Suchergebnissen, unter Berücksichtigung von delayed strokes durchgeführt wurde. In Hu and Brown [19] wird beschrieben, dass mithilfe dieser Technik eine 6-fache Reduktion der Fehlerrate von 19.4% auf 3.1%, gegenüber einer 1-besten Klassifikation erzielt werden konnte. Es ist deutlich zu erkennen, dass diese Strategie auch hier zu einer weiteren Verbesserung der Erkennungsgenauigkeit führen könnte. Der Nachteil dieses Verfahrens liegt jedoch im zusätzlichen zeitlichen

		Erkennungsrate in %	
Vokabulargröße	Testbeispiele	Top 1	Top 10
1000	1500	71.8	82.2
2000	1500	69.6	79.0
1500	2000	69.8	76.2
2500	2000	68.2	75.6
2500	3000	67.4	72.8
3500	3000	66.0	69.2

Tabelle 8.1: Worterkennungsresultate auf kleinen und mittleren Lexika.

Aufwand, der gerade bei Realtime-Anwendungen eine nicht unerhebliche Komponente darstellt. So kann man sich vorstellen, dass es effektiver ist, die 5-besten Kandidaten dem Benutzer zur Auswahl anzuzeigen, anstatt ihn weiter auf ein möglicherweise besseres 1-bestes Erkennungsresultat warten zu lassen.

8.3 Auswertung der Ergebnisse

Zur spezifischeren Untersuchung wurden die Gesichtspunkte Merkmale, Parametereinstellungen, Vorverarbeitung und die Identifizierung typischer Fehlklassifikationen genauer betrachtet. Hierzu wurde eine Teilmenge des 1500 Datensatzes bestehend aus 500 Testbeispielen und einer Vokabulargröße von 1000 Wörtern genauer untersucht. Dieser Datensatz wurde zufällig aus dem Datensatz bestehend aus 1500 Wörtern anhand der UNIPEN Verteilungsstatistik, wie in Abbildung 8.1 angegeben, erstellt.

8.3.1 Auswahl der Referenzmodellsätze

Zur Bestimmung der besten Merkmalskombination wurde folgende Vorgehensweise verfolgt: Zuerst wurden Modelle auf einem Datensatz von 18.525 Groß- und 40.491 Kleinbuchstaben gelernt. Dieser Trainingsdatensatz wurde unter Verwendung der jeweiligen Kategorien zufällig zusammengestellt. Anschließend wurden die Modelle zur Klassifikation von 15.041 Testbuchstaben herangezogen. Die Testdaten wurden aus einer disjunkten Menge von Schreibern entnommen. Die Referenzmodelle, welche hierbei die besten Erkennungsresultate erzielen konnten, wurden weiterführend verwendet. In der nachfolgenden Tabelle sind die besten Merkmaleskombinationen mit den jeweils erzielten Fehlerraten aufgeführt.

Bezeichung	Neuabtastung	Merkmale	Fehlerrate in %	#Modelle
A1	keine	VY, θ	27.5	499
A1	ja	$VY, \tilde{\theta}$	28.4	410
B1	keine	$\mathbf{P}_y^{(\text{norm})}, \theta$	12.3	298
B2	ja	$\tilde{\mathbf{P}}_y^{(\text{norm})}, \tilde{\theta}$	14.1	250

Zu den hier erzielten Erkennungsleistungen ist zu bemerken, dass durch die Zusammenlegung von Groß- und Kleinbuchstaben und den kleineren Merkmalsätzen, es zu wesentlich schlechteren Erkennungsraten auf Zeichenebene kam wie sie in der Arbeit von Bahlmann and Burkhardt [3] erzielt werden konnten.

Das schlechte Ergebnis des Merkmals VY (zur Berechnung diese Merkmals sei auf Bahlmann and Burkhardt [3] verwiesen) ist daraufhin zurückzuführen, dass keine Größennormierung anhand der Kernhöhe durchgeführt wurde, so kam es zu typischen Verwechselungen von Groß- und Kleinbuchstaben wie beispielsweise des kleinen 'c's und dem großen 'C'. Da jedoch eine nachträgliche Normierung, wie sie in Abschnitt 8.1.1 beschrieben ist, stattfand, konnte der Modellsatz an Diskriminierungseigenschaften weiter hinzugewinnen.

8.3.2 Bestimmung des geeignetsten Modellsatzes

Um den geeignetsten Referenzmodellsatz bestimmen zu können wurden die Modellsätze unter gleichen Bedingungen getestet. Hierbei wurden die Testdaten einer Vorverarbeitung mit anschließender Neuabtastung unterzogen. Der Abstand zwischen den neuabgetasteten Koordinatenpunkte wurde mit $\frac{1}{10}$ der Kernhöhe fest gewählt. Als Glättungsfaktor wurde $p = 0.1$ bestimmt. Für die Bedeutung des Glättungsfaktors bei der Splinekurvenberechnung wird auf die Arbeit Simon [47] verwiesen. Dieser Faktor führt allgemein zu einer schwachen Glättung. Insgesamt zeigten sich diese Einstellungen in einem Vortest auf kleineren Datenmengen als gute Wahl. Die Berechnungen der Wortklassifikationen fanden auf einem AMD DURON 1.2 MHz mit 1 GHz Hauptspeicher statt. Nachfolgende Tabelle enthält Ergebnisse mit ungefähren Zeitangaben, welche auf Grundlage des angegebenen Rechnermodells ermittelt wurden.

	Ø Klassifikationszeit	Erkennungsrate in %	
Referenzmodellsatz	pro Wort [Sek]	Top 1	Top 10
A1	21.4	78.8	89.0
A2	18.0	72.2	78.0
B1	5.5	58.6	69.2
B2	3.9	56.8	62.4

Testbeispiele:	500
Vokabulargröße:	1000
Beam:	60
Char-Pruning:	200
Node-Pruning:	30
Zeilenkorrektur:	ja
Schriftneigung:	nein

Es zeigt sich hierbei schon deutlich, dass die Modellsätze B1 und B2, welche mittels der automatischen Referenzlinienschätzung erzeugt wurden, weniger geeignet waren, während die nachträglich normierten Modellsätze A1 und A2 zu besseren Werten führten. Zusätzlich sind die Modellsätze mit Neuabtastung in beiden Kategorien schlechter zu bewerten. Ein Nachteil von A1 und A2 ist die durchschnittliche Laufzeit, die auf die höhere Anzahl an Modellen zurückzuführen ist.

Im Weiteren wird nur noch der Modellsatz A1 betrachtet, dessen Ergebnisse anhand dieser Untersuchung am viel versprechendsten sind.

8.3.3 Einfluss der Beam-, Char-Pruning- und Node-Pruningkonstanten

Die Wahl dieser Konstanten nimmt sowohl erheblichen Einfluss auf die Geschwindigkeit als auch auf die Erkennungsleistung des Systems. So führen kleine Konstanten zu einer kurzen Erkennungszeit, wodurch jedoch die Erkennung umso stärker vom Anfangsteil eines Wortes bestimmt ist. Nachfolgend sind einige Einstellungen aufgelistet:

Konstanten			Ø Klassifikationszeit	Erkennungsrate in %	
Beam	Char	Node	pro Wort [Sek]	Top 1	Top 10
30	30	30	1.5	45.2	56.0
30	60	30	5.3	65.0	79.8
30	120	30	15	75.0	87.2
60	200	30	21.4	78.8	89.0
60	350	30	78.1	77.2	86.4
100	200	30	25.4	72.2	82.4

Testbeispiele:	500
Vokabulargröße:	1000
Zeilenkorrektur:	ja
Glättungsfaktor:	0.1
Neuabtastung:	$\frac{1}{10}$
Schriftneigung:	nein

Die kleinsten Parametereinstellungen führten zu einer schnelleren Erkennungzeit, jedoch traten hier die typischen Fehlerkennungen durch den links rechts Vergleich auf.

8.3.4 Einfluss der Vorverarbeitung auf die Erkennungsrate

Um den Einfluss der Vorverarbeitung bestimmen zu können, wurden die bisher besten Parametereinstellungen verwendet.

Testbeispiele	Vokabulargröße	Glättungsfaktor	Beam	Char	Node
500	1000	0.1	60	200	30

Vorverarbeitungskorrekturen				Ø Zeit	Erkennungsrate in %	
Zeilenneigung	Schriftneigung	Referenzlinie+Extrema	Neuabtastung	pro Wort [Sek]	Top 1	Top 10
nein	nein	nein	nein	14.2	69.2	74.2
nein	nein	ja	nein	14.2	72.8	80.2
nein	nein	ja	$\frac{1}{10}$	21.4	73.2	83.8
nein	ja	ja	$\frac{1}{10}$	20.9	72.0	84.4
ja	ja	ja	$\frac{1}{10}$	20.4	77.2	86.4
ja	nein	ja	$\frac{1}{10}$	21.4	78.8	89.0

Durch die Hinzunahme der einzelnen Vorverarbeitungsschritte konnte jeweils eine leichte Steigerung der Erkennungsrate erzielt werden. Einzig die Schriftneigungskorrektur führte zu leichten Verschlechterungen in den Erkennungszahlen. Die Referenzlinienschätzung mithilfe des Histogrammverfahrens und der Erweiterung durch die Extrema-Methode brachten leichte Verbesserungen. Die Neuabtastung mit $\frac{1}{10}$ der Kernhöhe wurde fest gewählt. Weitere Untersuchungen zum Einfluss der Neuabtastung auf die Erkennungsrate wurden nicht durchgeführt.

8.3.5 Typische Fehlklassifikationen

Die Untersuchung des besten Resultats auf 1500 Testdaten zeigte, dass großgeschriebene Worte wie auch Worte die nur aus Großbuchstaben bestehen besonders häufig zu Fehlklassifikationen führten. So wurden 74.7% aller kleingeschriebenen Worte richtig erkannt, während nur 64.2% der großgeschriebenen Worte und 47.1% der nur aus Großbuchstaben bestehenden Worte richtig erkannt werden konnten.

Bei Worten, die nur aus Großbuchstaben bestanden, lag es an der falschen Kernlinienschätzung, welche bei diesen Worten zumeist nicht vorhanden ist und somit nicht richtig geschätzt werden konnte. Besteht das Wort beispielsweise aus den Buchstaben 'CVD' so existiert keine Information zur Kernlinienschätzung. Hierdurch wurden die Buchstaben bei der Normierung auf Kernhöhe skaliert. Ein Ausweg versprach die entsprechende Normierung der Großbuchstabenmodelle auf Kernhöhe, dies führte jedoch zu größeren Fehlklassifikationen im Bereich von kleingeschriebenen Wörtern.

Bei großgeschriebenen Wörtern führten häufig überdimensionierte Anfangsbuchstaben zu Fehlerkennungen. Hier zeigte sich deutlich die Abhängigkeit des verwendeten normierten y-Merkmals von der Güte der Vorverarbeitung.

KAPITEL 9

Portierung der Worterkennung auf einen PDA

Den Abschluss dieser Arbeit bildet die Integration der Worterkennung in ein Compaq iPAQ 3660 PDA. Dieser ist mit einem 203 MHz StrongARM Prozessor (ohne Fließkomma Arithmetik), 16 MB Flash-Speicherkarte und einem 64 MB RAM Hauptspeicher ausgestattet. Sein Betriebsystem bildet die Linux Kernel Version 2.4.18 aus der *Familiar* Distribution [16]. Als grafische Benutzerschnittstelle wurde die *Open Palmtop Integrated Environment (OPIE)* [35] benutzt, welches ein GPL lizenzierte Umsetzung des von Trolltech entwickelten *Qtopia* [49] darstellt. Auf dieser Umgebung konnte bereits erfolgreich eine Integration der Zeichenerkennung unter der Bezeichnung *frog on hand*, wie sie in der Arbeit von Bahlmann and Burkhardt 3 beschrieben ist, durchgeführt werden. Durch den Einsatz von Strahlsuche und Pruningmethoden bei der Viterbiberechnung, benötigt die Erkennung eines Buchstabens etwa 0.3 Sekunden.

Die größte Herausforderung, die sich bei einer Portierung der Worterkennung ergibt, sind die begrenzten Ressourcen, die eine solche Umgebung zu Verfügung stellt. Um hier eine akzeptable Anwendung programmieren zu können sind mehrere Einschränkungen notwendig:

1. Zuerst wird der Eingabebereich mit Referenzlinien ausgestattet, wodurch sich die Vorverarbeitung vollständig erübrigt.
2. Desweiteren findet eine Unterabtastung der Stiftbewegungen statt, sodass weitaus weniger Merkmale verglichen werden müssen.
3. Eine weitere Beschleunigung lässt sich dadurch erreichen, dass die Referenzmodelle nur aus wenigen Allographen bestehen.
4. Zusätzlich kann durch die Wahl einer kleinen Strahlbreite und kleinen Pruningparameter im Character- und Node-Pruning Bereich, weitere Platzeinsparungen und Effizienzsteigerungen erzielt werden. Diese Maßnahmen führen natürlich unweigerlich zur Reduktion

der Erkennungsleistung. Dadurch, dass die Erkennung von links nach rechts erfolgt, führen unleserliche Wortanfänge zumeist zu falschen Suchpfaden im Lexikon Trie.

Die Erkennung eines Wortes erfolgt nicht während des Schreibvorgangs sondern erst nach Fertigstellen eines Wortes. Hierzu muss der Anwender jedes geschriebene Wort einzeln auf dem Display bestätigen. Es werden ihm die n-besten Resultate in einer Auswahlliste angezeigt, die er dann mittels dem Stift auswählen kann.

KAPITEL 10

Zusammenfassung und Ausblick

10.1 Zusammenfassung

In der vorliegenden Arbeit wurde ein Weg aufgezeigt, wie durch entsprechende Erweiterung der Buchstabenerkennung eine Worterkennung aufgebaut werden kann. Hierzu wurden neue Vorverarbeitungsverfahren vorgestellt, die eine Normierung der Schrift ermöglichen und den Ausgangspunkt für die Merkmalsextraktion bilden. Es erfolgte die Erweiterung des bestehenden Erkennungssystems unter zusätzlicher Hinzunahme von linguistischem Wissen und neuen Suchtechniken bei der Viterbisuche. Desweiteren wurde die Leistungsfähigkeit des Systems anhand des UNIPEN-Datensatz dokumentiert und abschließend eine Portierung auf einen PDA durchgeführt.

Um eine Merkmalsextraktion bei der Worterkennung zu ermöglichen, mussten neue Ansprüche an die Vorverarbeitung gestellt werden. Durch eine Neuabtastung anhand von Splinekurven war es möglich, eine robuste Korrektur des Schriftzuges mithilfe von Histogrammanalyseverfahren vorzunehmen. Zur Auswertung der Histogramme im Bereich der Zeilen- und Schriftneigungskorrektur wurden zwei aus der Literatur bekannte Verfahren implementiert und im Falle der Zeilenneigung einander gegenübergestellt. Es konnte hierbei gezeigt werden, dass die Wigner Ville Verteilung bessere Resultate nach einer Neuabtastung liefert. Zusätzlich konnte durch die Neuabtastung die Rückweisungsrate verringert werden. Wie die Testergebnisse zeigten, konnte der Schriftneigungskorrektur eine geringere Gewichtung beigemessen werden, was hauptsächlich auf die statistische Modellierung der Referenzmodelle zurückzuführen ist. Ein weitaus wichtigerer Teil stellt die Referenzlinienschätzung dar, von der die Güte des normierten y-Merkmals unmittelbar abhängig ist. So wurden bei der Schätzung gleich mehrere Modifikationen vorgestellt. Zum einen konnten durch eine Restriktion auf abwärtsgerichtete Stiftbewegungen die Histogrammauswertung positiv beeinflusst werden, zum anderen wurden weitere Verbesserungen

durch die zusätzliche Berechnung der lokalen Extrema und der Sättigung des Histogramms in deren Clusterzentren erzielt.

Im Bereich der Merkmalsextraktion wurden lokale Merkmale vorgestellt, die je nach Vorverarbeitung anhand einer Neuabtastung beziehungsweise ohne Neuabtastung berechnet wurden. Anschließend erfolgte die Erweiterung des bereits entwickelten CSDTW Verfahrens. Hierzu fanden Erweiterungen im Bereich der Zeichenübergänge statt, wie sie aus der Spracherkennung zur Modellierung von Wortübergänge zum Einsatz kommen. Zusätzlich wurde eine Ligaturmodellierung durch Hinzunahme neuer lokaler Transitionen im Übergangsbereich durchgeführt, deren Reichweite von der Änderung des Tangentensteigungswinkels abhängig gemacht wurde. Durch die Integration eines Lexikon Tries bei der Viterbisuche wurden nur solche Hypothesen an den Übergängen aktiviert, die ausgehend vom korrespondierenden Knoten im Baum erreicht werden konnten. Eine der Hauptschwierigkeiten der Worterkennung liegt in der enormen Hypothesenflut, die sich bei der Viterbisuche trotz linguistischem Wissen ansammelt. Um hier möglichst früh die Anzahl der aktiven Hypothesen zu dezimieren, wurden effiziente Suchstrategien integriert, wie sie aus der Spracherkennung bekannt sind. Hierbei wurde eine eigene Pruningstrategie verfolgt, um eine möglichst effiziente Berechnung auch bei dichtem Hypothesenfeld gewährleisten zu können.

Die ersten Untersuchungen auf dem UNIPEN-Datensatz können als zufriedenstellend bezeichnet werden, jedoch liegen die erzielten Werte noch weit von den Resultaten internationaler Systeme entfernt. Eine Analyse der n-besten Kandidaten zeigte, dass eine intensivere Suche innerhalb dieses Kandidatenfeldes weiteres Potential für eine Verbesserung der Erkennungsgenauigkeit bereitstellt.

Den Abschluss bildete die Integration der Worterkennung in die Eingabeumgebung eines PDAs. Hierbei spielte vor allem der begrenzte Ressourcenvorrat einen entscheidenden Faktor. So konnten aus Platz- und Rechenleistungsgründen nur eine begrenzte Anzahl an Hypothesen verfolgt werden. Zur Effizienzsteigerung wurden Referenzlinien im Eingabefeld vorgegeben, wodurch die Vorverarbeitung vollständig entfallen konnte.

10.2 Ausblick

Zur weiteren Verbesserung des Systems ist es notwendig Untersuchungen mit so genannten long range Merkmalen, wie sie im Überblick zur Merkmalsberechnung vorgestellt wurden, durchzuführen. Diese Merkmale ermöglichen eine vorausschauendere Hypothesenauswahl bei der Viterbisuche. Zusätzlich muss eine Berücksichtigung von zeitlich versetzten Strichen *(*delayed strokes*)* durchgeführt werden. Den entscheidenden Faktor einer Erkennungssteigerung dürfte jedoch die Erweiterung des Systems durch eine *two-pass* Ansatz zeigen, der eine intensivere Suche auf den n-besten Erkennungsergebnissen durchführt.

KAPITEL 11

Danksagung

An dieser Stelle will ich mich bei der sehr guten Betreuung durch meinen Mentor Claus Bahlmann bedanken.

Literaturverzeichnis

[1] URL http://lmb.informatik.uni-freiburg.de/.

[2] E. Anquitil and G. Lorette. Automatic generation of hierarchical fuzzy classification systems based on explicit fuzzy rules deduced from possibilistic clustering. In *Application to On-line Handwritten Character Recognition*, pages 259–264, Granada, 1996. 4.3

[3] C. Bahlmann and H. Burkhardt. The writer independent on-line handwriting recognition system *frog on hand* and cluster-generative statistical dynamic time warping. *Currently in review by IEETPAMI*, submitted 2003. 5.2, 6.3.1, 6.5, 8.3.1, 9

[4] R. Bellmann. *Dynamic Programming*. Princeton University Press, 1957. 6.2

[5] S. Bercu and G. Lorette. On-line handwritten word recognition: An approach based on Hidden Markov Models. *Proc. 3rd Int. Workshop Front. Handwriting Recognition (IWFHR)*, pages 385–390, 1993. 4.1

[6] D. Bockhorn. Bestimmung und Untersuchung von Signifikanzgewichtungen für die Erkennung von handgeschriebenen Buchstaben. Diploma thesis, Albert-Ludwigs-Universität Freiburg, Institut für Informatik, 2000. URL http://lmb.informatik.uni-freiburg.de/. 6.3.1, 6.3.2, 6.3.2, 6.4

[7] R. M. Bozinovic and S. N. Srihari. Off-line cursive script word recognition. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 11(1):68–83, Jan. 1989. 3.5

[8] J. S. Bridle, M. D. Brown, and R. M. Chamberlain. An algorithm for connected word recognition. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 899–902, Paris, May 1982. 7.2

[9] E. R. Brocklehurst and P. D. Kenward. Preprocessing for cursive script recognition. *NPL Report DITC 132/88*, Nov. 1988. 3.5.2.1

[10] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden Markov models. *Pattern Recognition*, pages 1399–1413, 1995. 3.5

[11] T. Caesar, J. Gloger, and E. Mandler. Preprocessing and feature extraction for a handwriting recognition system. In *Proc. 2nd Int. Conf. Doc. Anal. Recognition (ICDAR)*, pages 408–411, Tsukuba, Japan, 1993. 3.3

[12] M. Cheriet and C. Y. Suen. Automatic reading of cursive scripts using human knowledge. In *Proc. 4th Int. Conf. Doc. Anal. Recognition (ICDAR)*, pages 107–111, October 1997. 4.3

[13] V. Claus and A. Schwill, editors. *Duden Informatik: Ein Fachlexikon für Studium und Praxis*. Dudenverlag: Mannheim-Leipzig-Wien-Zürich, 3. edition, 2001. 1

[14] S. D. Connell. *Online Handwriting Recognition using Multiple Pattern Class Models*. PhD thesis, Michigan State University, 2000. 1.1, 1.6

[15] S. Edelman, T. Flash, and S. Ullman. Reading cursive handwriting by the alignment of letter prototypes. In *International Journal of Computer Vision (IJCV)*, volume 5, pages 303–331, 1990. 4.3

[16] Familiar. Linux distribution for compaq ipaq. URL `http://familiar.handhelds.org/`. 9

[17] R. Farag. Word-level recognition of cursive script. *IEEE Trans. on Comp.*, C28(2):172–175, February 1979. 4.1

[18] I. Guyon, L. Schumaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Proc. 12th Int. Conf. Pattern Recognition (ICPR)*, pages 29–33, Oct. 1994. 1.6, 2

[19] J. Hu and M. K. Brown. On-line handwriting recognition with constrained n-best decoding. In *Proc. 13th Int. Conf. Pattern Recognition (ICPR)*, Vienna, Austria, 1996. effizienteres n-best decoding. 7.2.3, 8.2

[20] J. Hu, M. K. Brown, and W. Turin. HMM based on-line handwriting recognition. *IEEE Trans. Pattern Anal. and Mach. Intell.*, pages 1039–1045, 1996. 1.6

[21] J. Hu, S. G. Lim, and M. K. Brown. Writer independent on-line handwriting recognition using an HMM approach. *Pattern Recognition*, 33:133–147, Jan. 2000. 1.6, 1.2, 4.2, 8.1.2, 8.2

[22] J. Hu, A. S. Rosenthal, and M. K. Brown. Combining high-level features with sequential local features for on-line handwriting recognition. In *Proc. 7th Int. Conf. on Intell. Syst. for Molecular Biology (ISMB)*, 1995. 1.6, 5.1

[23] B.-H. Juang and L. Rabiner. The segmental k-means algorithm for estimating parameters of hidden Markov models. *IEEE Trans. Acoust. Speech Signal Process.*, 38:1639–1641, 1990. 6.3.2

[24] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. New algorithms for skewing correction and slant removal on word-level. In *Proc. 6th IEEE International Conference on Electronics (ICECS)*, volume 2, pages 1159–1162, Pafos, Cyprus, September 1999. 3.3, 3.3.3, 3.4.1

[25] A. Kosmala. HMM-basierte Online Handschrifterkennung - ein integrierter Ansatz zur Text- und Formelerkennung, 2000. PhD thesis. 3.3

[26] A. Kosmala, J. Rottland, and G. Rigoll. Improved on-line handwriting recognition using context dependent Hidden Markov Models. In *Proc. 4th Int. Conf. Doc. Anal. Recognition (ICDAR)*, pages 641–644, Ulm, Germany, Aug. 1997. 1.6

[27] G. Lorette. Handwriting recognition or reading ? What is the situation at the dawn of the 3rd millenium. In *Proc. Internat. J. on Document Anal. and Recognition (IJDAR)*, volume 2, pages 2–12, 1999. 4.3

[28] S. Manke, M. Finke, and A. Waibel. Combining bitmaps with dynamic writing information for on-line handwriting recognition. In *Proc. 12th Int. Conf. Pattern Recognition (ICPR)*, pages 596–598, Jerusalem, Israel, 1994. 5.1

[29] S. Manke, M. Finke, and A. Waibel. NPen++: A writer independent, large vocabulary on-line cursive handwriting recognition system. In *Proc. 3rd Int. Conf. Doc. Anal. Recognition (ICDAR)*, Montreal, Canada, 1995. 1.6, 4.2, 7.3

[30] S. Manke, M. Finke, and A. Waibel. A fast search technique for large vocabulary on-line handwriting recognition. In *Proc. 5th Int. Workshop Front. Handwriting Recognition (IWFHR)*, Colchester, UK, 1996. 7.1

[31] C. S. Myers and L. R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. *IEEE Trans. Acoust. Speech Signal Process.*, pages 284–297, Apr. 1981. 7.2

[32] S. Nakagawa. A connected spoken word recognition methode by O(n) dynamic programming pattern matching algorithm. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 226–299, Bosten, USA, April 1983. 7.2

[33] K. S. Nathan, H. Beigi, H. Subrahmonia, G. J. Clary, and H. Maruyama. Real-time on-line unconstrained handwriting recognition using statistical methods. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 2619–2623, Detroit, MI, May 1995. 1.6

[34] H. Ney. The use of a One-Stage dynamic programming algorithm for connected word recognition. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume ASSP-32, pages 263–271, April 1984. 7.2

[35] OPIE. Open Palmtop Integrated Environment. URL http://opie.handhelds.org/. Graphical user interface for Linux Compaq iPAQ. 9

[36] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993. 6.2, 6.2.1.3

[37] G. Rigoll, A. Kosmala, and D. Willett. A new hybrid approach to large vocabulary cursive handwriting recognition. In *Proc. 12th Int. Conf. Pattern Recognition (ICPR)*, pages 1512–1514, Brisbane, 1994. 1.6

[38] A. S. Rosenthal, J. Hu, and M. K. Brown. Size and orientation normalization of on-line handwriting using hough transform. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1997. 3.5

[39] H. Sakoe. Two level DP matching—a dynamic programming based pattern matching algorithm for connected word recognition. *IEEE Trans. Acoust. Speech Signal Process.*, pages 588–595, Dec. 1979. 7.2

[40] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *IEEE 7th Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 20C-13, Budapest, Hungary, 1971. 6.2

[41] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.*, 26(1):43–49, Feb. 1978. 6.2.1.3

[42] K. Sayre. Machine Recognition of Handwritten Words: A Project Report. *Pattern Recognition*, 5(3):213–228, 1973. 4

[43] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time delay neural networks and hidden Markov models. In R. Plamondon, editor, *Machine Vision and Applications 8*, pages 215–223. Springer Verlag, 1995. 3.2

[44] L. Schomaker and H.-L. Teulings. Stroke- versus character-based recognition of on-line, connected cursive script. In J.-C. Simon and S. Impedovo, editors, *From Pixels to Features III Frontiers in Handwriting Recognition*, pages 313–325. North Holland, 1992. 4.2

[45] E. G. Schukat-Talamazzini. *Automatische Spracherkennung—Grundlagen, statistische Modelle und effiziente Algorithmen*. Friedr. Vieweg & Sohn, 1995. 6.2.3, 6.3.2, 7.3

[46] G. Seni, R. K. Srihari, and N. Nasrabadi. Large vocabulary recognition of on-line handwritten cursive words. *IEEE Trans. Pattern Anal. and Mach. Intell.*, pages 757–762, 1996. 1.6, 3.3, 3.5, 3.5.2.1

[47] K. Simon. Vorverarbeitung und Merkmalsextraktion in der Online-Handschrifterkennung. Student's thesis, Albert-Ludwigs-Universität Freiburg, Institut für Informatik, 2002. URL `http://lmb.informatik.uni-freiburg.de/`. 3.2, 8.3.2

[48] C. Tappert. Cursive script recognition by elastic matching. *IBM J. Res. Develop.*, 26(6): 765–771, Nov. 1982. 6.2

[49] Trolltech. Qtopia. URL `http://www.trolltech.com/products/qtopia/`. Graphical user interface for Linux Compaq iPAQ. 9