Videobasierte Handschrifterkennung

Markus Wienecke

Dipl.-Inform. Markus Wienecke
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld

Genehmigte Dissertation zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.).
Von Markus Wienecke am 22. Oktober 2003
der Technischen Fakultät an der Universität Bielefeld vorgelegt.
Am 8. Dezember 2003 verteidigt und genehmigt.

Gutachter:

PD Dr.-Ing. Gernot A. Fink, Universität Bielefeld
Prof. Dr. Horst Bunke, Universität Bern

Prüfungsausschuss:

Prof. Dr. Helge Ritter, Universität Bielefeld
PD Dr.-Ing. Gernot A. Fink, Universität Bielefeld
Prof. Dr. Horst Bunke, Universität Bern
Dr.-Ing. Jannik Fritsch, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier nach DIN ISO 9706

Videobasierte Handschrifterkennung

Dissertation zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Technischen Fakultät der Universität Bielefeld
vorgelegt von

Markus Wienecke

22. Oktober 2003

Danksagung

An erster Stelle möchte ich meinem Betreuer Dr. Gernot A. Fink für die hervorragende fachliche Unterstützung danken. Seine zahlreichen Anregungen und wertvollen Ratschläge haben sehr zum Gelingen dieser Arbeit beigetragen.

Mein Dank gilt auch Prof. Dr. Horst Bunke für seine Bereitschaft, diese Niederschrift zu begutachten. Ihm möchte ich außerdem dafür danken, dass wir die am Institut für Informatik und angewandte Mathematik der Universität Bern erstellte IAM-Handschriftstichprobe für unsere Forschungen verwenden dürfen.

Außerdem danke ich Prof. Pietro Perona und Mario Munich vom California Institute of Technology, dass sie uns ihr System zur Stiftverfolgung in Videobildfolgen zur Verfügung stellten. Dieses System wurde im Rahmen eines Projektseminars erfolgreich weiterentwickelt. Den Seminarteilnehmern Birgit Möller, Frank Seifert, Marc Hanheide und Thomas Plötz danke ich für ihren enthusiastischen Einsatz.

Darüberhinaus möchte ich mich bei Birgit Möller und Thomas Plötz sowie bei Daniel Schlüter dafür bedanken, dass sie mir als Bürokollegen das Arbeiten in einer sehr angenehmen Atmosphäre ermöglichten und mir oftmals mit vielen Ideen und Tipps weiterhelfen konnten.

Inhaltsverzeichnis

1 Einleitung

Die Entwicklung von Verfahren zur automatischen Verarbeitung handschriftlicher Dokumente ist trotz der vielfach vorhandenen Möglichkeiten der elektronischen Kommunikation von wachsendem Interesse. Ein Beleg für diese Hypothese sind die zunehmenden Forschungsaktivitäten, die seit Beginn der 90er Jahre auf dem Gebiet der Schrifterkennung vorgenommen werden [Bun03]. Stift und Papier weisen somit offenbar einige vorteilhafte Eigenschaften auf, die die handschriftliche Kommunikation auch im Zeitalter der Computer und elektronischen Medien fortbestehen lassen.

Lesen und Schreiben zählen zu den ältesten Kulturtechniken der Menschheit, sodass der Umgang mit Stift und Papier im Gegensatz zu elektronischen Medien einer weitaus größeren Anzahl von Menschen vertraut ist. Zudem ist Papier billig, leicht und gut handhabbar. Es erlaubt die flexible Navigation innerhalb des Dokuments, ist außerdem universal einsetzbar und persistent – das Geschriebene bleibt mithin über Jahrzehnte oder Jahrhunderte erhalten. Weiterhin ist die Verwendung von Stift und Papier ideal für das Anfügen von Annotierungen in Dokumenten oder das Hervorheben bestimmter Textpassagen, sodass eine enge Verflechtung von Lesen und Schreiben ermöglicht wird. Dies sind nur einige Vorteile, die Sellen & Harper in ihrem Buch *"The myth of the paperless office"* (Der Mythos des papierlosen Büros) anführen [Sel02]. Die Autoren kommen darin zu dem Schluss, dass zum Lesen Papier das Medium der Wahl ist, auch wenn elektronische Kommunikationsmittel zur Verfügung stehen.

> *Paper [remains at present] the medium of choice for reading, even when the most high-tech technologies are to hand.*

Nach Ansicht von Plamondon haben die technologischen Errungenschaften der Vergangenheit die handschriftliche Kommunikation nicht zurückgedrängt, sondern im Gegenteil eher ihre Verbreitung gefördert [Pla95, Pla00]. So trugen beispielsweise die Erfindungen von Druckpresse und Schreibmaschine dazu bei, dass die Welt des geschriebenen Wortes nicht mehr einigen wenigen vorbehalten war. Ungleich mehr Menschen lernten Lesen und Schreiben und besaßen damit die Möglichkeit, sich an der schriftlichen Kommunikation zu beteiligen. Die Rolle der Handschrift unterlag somit einem Wandel, bei dem zwar die durchschnittliche Länge der handschriftlichen Dokumente abnahm, im Gegenzug jedoch die Anzahl der Menschen, die Handschrift verwenden, im gleichen Umfang zunahm.

Diesen Zusammenhang postuliert Plamondon auch für das "Computerzeitalter". Demnach würde die Bedeutung handschriftlicher Kommunikation durch den vermehrten Einsatz von Computern keinesfalls vermindert. Dies liegt an den Vorteilen von Stift und Papier gegenüber den herkömmlichen Computer-Eingabegeräten wie Tasta-

tur oder Maus, die in vielen Anwendungssituationen nur bedingt geeignet sind. So lassen sich insbesondere Skizzen, Tabellen, Formeln oder kurze Notizen handschriftlich schneller und bequemer anfertigen als per Tastatur oder Maus. Die stiftbasierte Eingabe ist außerdem platzsparend, sodass bei einem Großteil der Handheld-Computer diese Form der Eingabeschnittstelle favorisiert wird.

Durch die Verwendung von Handschrift zur Mensch-Maschine Kommunikation steigt damit der Bedarf an Systemen zur automatischen Handschrifterkennung. Um eine Akzeptanz dieser Systeme bei den Benutzern zu erreichen, werden hohe Anforderungen an die eingesetzten Verfahren gestellt. Im Vordergrund steht dabei eine möglichst geringe Fehlerrate, die das Erkennungssystem bei gleichzeitig möglichst kurzen Antwortzeiten aufweisen sollte. Die tolerierbare Fehlerrate hängt dabei stark von der Aufgabenstellung und von dem erwarteten "Nutzen" der Texterkennung ab. So werden nach der in [Fra95] beschriebenen Studie Erkennungsfehler bei Suchanfragen an eine Datenbank eher akzeptiert als beispielsweise bei Einträgen in ein Tagebuch.

Weiterhin sollte das Erkennungssystem auch in uneingeschränkten Szenarien einsetzbar sein, eine natürliche Interaktion erlauben und beliebige Schriftstile verarbeiten können. Wünschenswert ist außerdem die Unabhängigkeit von einem vorgegebenen Lexikon bzw. einer bestimmten Sprache. Zum gegenwärtigen Zeitpunkt sind geringe Fehlerraten jedoch nur in sehr eingeschränkten Szenarien erzielbar, wenn die Komplexität der Erkennungsaufgabe durch die Integration von Kontextwissen reduziert werden kann, beispielsweise bei der Erkennung von Postanschriften oder der Verarbeitung von Bankschecks.

Zur Signalaufnahme werden im überwiegenden Teil der Handschrifterkennungssysteme entweder Scanner eingesetzt, die ein Abbild des Geschriebenen liefern, oder Digitalisiertabletts, mit denen die Dynamik der Schreibbewegung aufgenommen wird. Diese spezialisierten Sensoren weisen als Eingabeschnittstelle für die Mensch-Maschine Kommunikation allerdings einige Nachteile auf. Beispielsweise bieten die Digitalisiertabletts nur bedingt eine natürliche Eingabeschnittstelle, da oftmals ein spezieller Stift erforderlich ist, sodass von Seiten des Benutzers eine gewisse Eingewöhnungszeit erforderlich ist. Wird dagegen ein Scanner zur Erfassung der Schrift eingesetzt, so können zwar ein normaler Stift und gewöhnliches Papier verwendet werden, jedoch ist durch den relativ zeitaufwendigen Scanvorgang keine schnelle Rückmeldung vom System und damit keine interaktive Kommunikation möglich.

Um die oben genannten Nachteile spezieller Sensoren zu vermeiden, erscheint es daher günstig, Videokameras zur Signalaufnahme zu verwenden. Damit wird zum einen das gewohnte Schreiben mit einem Stift auf Papier ermöglicht, zum anderen können durch die fortwährende Beobachtung des Schreibprozesses schnelle Rückmeldungen generiert werden, sodass beispielsweise interaktive Korrekturen durchgeführt werden können.

Mit Hilfe von Videokameras kann natürlich nicht nur die Schrift sondern auch die Gestik des Anwenders aufgenommen werden, sodass mit einem einzigen Sensor eine multimodale Eingabeschnittstelle realisiert werden kann. In Verbindung mit Mikrofonen zur Aufnahme der sprachlichen Äußerungen wird damit der Grundstein für die

Verwirklichung von Systemen zum *e-learning* oder *collaborative working* gelegt, indem Schrift, Gestik und Sprache miteinander in Beziehung gesetzt werden.

Nicht zuletzt stellen Videokameras mittlerweile preiswerte, kompakte und weit verbreitete Zubehörgeräte dar – nicht nur für Computer beispielsweise als sogenannte Webcams, sondern vermehrt auch für Mobiltelefone. Auch aus diesem Grund sind videobasierte Systeme eine bedenkenswerte Alternative zu herkömmlichen Benutzerschnittstellen.

Ziele und thematische Abgrenzung

Das Ziel dieser Arbeit ist die Untersuchung und Entwicklung videobasierter Methoden zur Handschrifterkennung. Durch die Verwendung einer handelsüblichen Videokamera zur Aufnahme des Schreibprozesses wird die Realisierung einer natürlichen Eingabeschnittstelle zur Mensch-Maschine Kommunikation angestrebt. Der Schreibprozess soll dabei möglichst wenigen Einschränkungen unterliegen, d.h. das Erkennungssystem sollte nicht auf bestimmte Anwendungsbedingungen, wie z.B. einen bestimmten Schreiber bzw. Schriftstil, festgelegt sein.

Für die Verarbeitung der Eingabedaten werden sowohl Verfahren untersucht, die auf der Erfassung der Schreibdynamik basieren, als auch Methoden, die im Gegensatz dazu auf einer bildhaften Repräsentation der Schriftdaten beruhen. Neben einer guten Erkennungsqualität, die in etwa vergleichbar mit herkömmlichen, auf einem Scanner bzw. Digitalisiertablett basierenden, Systemen sein sollte, werden außerdem kurze Antwortzeiten angestrebt, um ein interaktives Agieren mit dem System zu ermöglichen.

Die Untersuchung der Methoden zur Handschrifterkennung ist hier auf Lateinschrift deutscher bzw. englischer Sprache beschränkt. Da beispielsweise die Verarbeitung chinesischer Schrift, die nach dem Prinzip der Logographie arbeitet, oder die Erkennung arabischer Schrift gegenüber der Lateinschrift zum Teil andere Herangehensweisen erfordern, würde deren ausführliche Beschreibung den Rahmen dieser Arbeit sprengen.

Gliederung

Die vorliegende Arbeit ist wie folgt aufgebaut: Nach der Einleitung in diesem Kapitel werden im zweiten Kapitel die Konzepte vorgestellt, die beim Menschen bei der Generierung und Perzeption von Handschrift beteiligt sind. Vor diesem Hintergrund erfolgt im dritten Kapitel die Beschreibung der Verfahren, die in technischen Systemen zur Handschrifterkennung eingesetzt werden. Dabei wird ausgehend von der Signalaufnahme bis hin zur Adaption ein Überblick über die grundlegenden Methoden gegeben, die dem gegenwärtigen Stand der Forschung entsprechen.

Auf die Vorteile bei der Verwendung von Videokameras zur Signalaufnahme wird in Kapitel vier eingegangen. In diesem Kapitel werden außerdem die Anforderungen beschrieben, die bei der videobasierten Schrifterfassung an die eingesetzten Verarbeitungsschritte gestellt werden.

Im fünften Kapitel wird das realisierte Erkennungssystem vorgestellt, das auf der Extraktion der Schreibdynamik anhand von Videobildfolgen basiert. Das dort beschriebene System ist jedoch nur in geeigneten Szenarien einsetzbar, da zur Erfassung der Schreibdynamik der Stift stets in den aufgenommenen Bildern sichtbar sein muss. Aufgrund der häufigen Verdeckungen des Stifts durch den Schreiber ist diese Voraussetzung beispielsweise nicht erfüllt, wenn an einer Wandtafel (Whiteboard) geschrieben wird.

Ein videobasiertes Erkennungssystem, das die oben genannte Einschränkung der Anwendbarkeit nicht aufweist und daher auch zur Verarbeitung von Schreibvorgängen am Whiteboard eingesetzt werden kann, wird in Kapitel sechs vorgestellt. Dieses System beruht im Gegensatz zu dem in Kapitel fünf beschriebenen auf der Detektion von statischen Textregionen in der Videobildfolge.

In Kapitel sieben erfolgt dann die Evaluation der realisierten Erkennungssysteme, wobei als Bewertungskriterium die erzielte Fehlerrate auf Wort- bzw. Zeichenebene verwendet wird. Im achten Kapitel wird schließlich eine Zusammenfassung der wichtigsten Punkte dieser Arbeit gegeben.

2 Generierung und Perzeption von Handschrift

Lesen und Schreiben sind bemerkenswerte Fähigkeiten des Menschen. So ist die Generierung von Handschrift eine äußerst komplexe motorische Handlung, die bei geübten Schreibern weitgehend automatisiert abläuft. Der Schriftstil ist dabei sehr individuell, dennoch relativ unveränderlich über die Zeit oder in unterschiedlichen Schreibsituationen.

Ist der Mensch mit bestimmten Schriftstilen vertraut, so kann er die Schrift ohne große Mühen entziffern. Dies gelingt sogar dann, wenn sehr schnell geschrieben wurde und infolgedessen das Schriftbild stark variiert. Die Handschriftperzeption des Menschen ist daher äußerst leistungsfähig und den bisherigen technischen Systemen weit überlegen. Was liegt daher näher, als erst die Konzepte zu analysieren, die beim Menschen bei der Generierung und Perzeption von Handschrift mitwirken, bevor man mit der Entwicklung eines technischen Systems zur Handschrifterkennung beginnt?

2.1 Generierung von Handschrift

In diesem Abschnitt wird die Fragestellung untersucht, wie der Mensch Handschrift produziert. Es werden Modelle vorgestellt, die den Prozess der Handschrift*generierung* auf einer symbolischen Ebene einerseits, und auf einer kinematischen Ebene andererseits, beschreiben. Dies mag auf den ersten Blick vielleicht wenig hilfreich erscheinen bei der Entwicklung eines technischen Systems zur Handschrift*erkennung*, die Konzepte der Handschriftgenerierung sind jedoch in vielerlei Hinsicht auch bei der Erkennung nützlich.

So wird in der motorischen Theorie der Handschriftperzeption angenommen, dass der Mensch unbewußt Wissen über motorische Prozesse der Handschriftgenerierung bei der Perzeption einsetzt. Kann ein Wort z.B. nicht allein aufgrund der Form des Schriftbildes entziffert werden, wird dieser Theorie folgend versucht, dynamische Information über den Generierungsprozess aus dem statischen Schriftbild zu extrahieren und für die Erkennung nutzbar zu machen. Es findet somit bei der Perzeption von Handschrift ein Mitwirken von Generierungskonzepten statt [Zim82, Fre87, Pla98a].

Für die Realisierung technischer Systeme zur Handschrifterkennung ist das Einbringen von Wissen über Konzepte der Handschriftgenerierung ebenso hilfreich. Ein wichtiger Grund ist, dass das enorme Kontextwissen, das der Mensch bei der Handschrifterkennung unbewußt einsetzt, technischen Systemen nicht in diesem Umfang zur Verfügung steht. Beispielsweise nutzt der Mensch oftmals den Satzkontext aus,

um die Identifikation eines Wortes zu unterstützen. Diese top-down Verarbeitung bei der Perzeption ist jedoch für technische Systeme aufgrund mangeldem Kontextwissen nur sehr eingeschränkt anwendbar, sodass vielmehr eine zuverlässige bottom-up Verarbeitung angestrebt wird. Insbesondere kann in technischen Systemen durch Modellierung der Konzepte der Handschriftgenerierung die Robustheit der Verarbeitung in Bezug auf Schriftnormierung, Segmentierung und Merkmalsextraktion erhöht werden [Teu94]. Darüberhinaus wird die Möglichkeit eröffnet, aus dem statischen Schriftbild dynamische Bewegungsinformationen zu gewinnen und ebenfalls für die Erkennung nutzbar zu machen.

Das Blockdiagramm in Abbildung 2.1 zeigt auf sehr hoher Abstraktionsebene die Prozesse, die bei der Handschriftgenerierung beteiligt sind [Pla89]. Um ein Schriftsegment zu generieren, wird demnach im Gehirn zuerst das entsprechende motorische Programm erstellt. Über die Nervenbahnen werden dann die entsprechenden Muskeln in einer festgelegten Reihenfolge aktiviert, sodass daraus eine Bewegung der Stiftspitze resultiert.

Die Prozesse, die in Abbildung 2.1 dargestellt sind, werden von Forschern verschiedener Fachrichtungen, wie z.B. Psychologen, Neurologen und Informatikern, untersucht. Die Forschungen lassen sich je nach Interessengebiet in zwei gegensätzliche Richtungen unterteilen: Top-down Ansätze und bottom-up Ansätze [Pla89]. Steht eher die Analyse des motorischen Programms auf einer symbolischen Repräsentationsebene im Vordergrund, wobei die biophysikalischen Abläufe, die die Informationsüber-

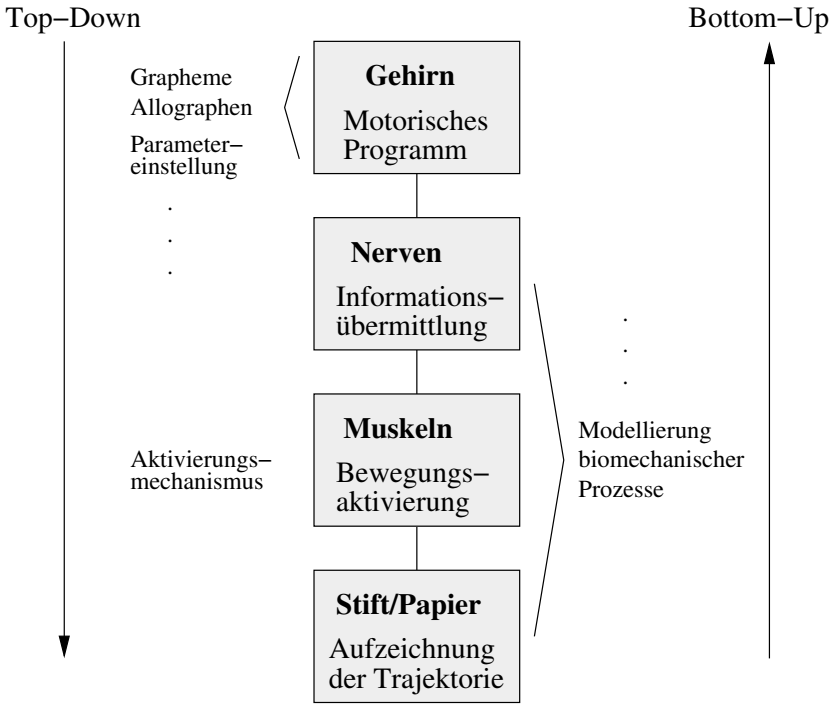


Abbildung 2.1: Blockdiagramm der Handschriftgenerierung (nach [Pla89])

tragung durch Nerven und die Aktivierung von Muskeln betreffen, als *Black Box* aufgefaßt werden, so entspricht dies einer top-down Strategie. Demgegenüber beschäftigt sich der bottom-up Ansatz mit den neuromuskulären Prozessen auf einer niedrigeren Ebene der Informationsverarbeitung und zielt auf eine kinematische Modellierung der Schreibbewegung ab.

2.1.1 Makroskopisches Modell

Kennzeichnend für das top-down Konzept der Bewegungsgenerierung ist ein makroskopisches Modell, bestehend aus einer Sequenz von Modulen, die aufgrund von Handschriftuntersuchungen und neurologischen Experimenten hypothetisiert werden [Ell82, Teu94]. Das Modell läßt sich unterteilen in Module einer höheren Ebene, die mittels im Gedächtnis gespeicherten Buchstabenmustern ein abstraktes motorisches Programm generieren, und Module einer niedrigeren Ebene, die dieses abstrakte Programm mit konkreten Bewegungsparametern versehen und die entsprechenden Muskeln aktivieren. Nach Ansicht von van Galen (siehe [Teu94]) ist diese Unterteilung aus Effizienzgründen motiviert. Da sich z.B. während des Schreibens einer Zeile die Orientierung der Hand graduell ändert, scheint es effizienter, die Bewegungsparameter des motorischen Programms zu adaptieren, anstatt sie ständig aus dem Gedächtnis abzurufen. Damit läßt sich auch das Phänomen erklären, dass das Schriftbild relativ unabhängig von den beteiligten Muskelgruppen ist (siehe [Ber76], S. 54). Dies äußert sich z.B. in ähnlichen Schriftbildern beim Schreiben an einer Tafel und beim “normalen” Schreiben auf Papier.

Den Ausgangspunkt des makroskopischen Modells der Handschriftgenerierung, dargestellt in Abbildung 2.2, bildet das zu schreibende Graphem, das in einem Zwischenspeicher für die weitere Verarbeitung aufbewahrt wird. Ein Graphem legt in diesem Zusammenhang fest, welcher Buchstabe geschrieben werden soll, es beschreibt jedoch nicht seine Form oder weitere Details wie z.B. Groß- oder Kleinschreibung. Die Graphemrepräsentation wird dann mit Hilfe des Allographengedächtnisses in eine allographische Beschreibung überführt. Allographen entsprechen unterschiedlichen Ausführungen eines Buchstabens, wobei zwar die Form, nicht jedoch die absolute Größe oder der zeitliche Ablauf bei der Produktion des Buchstabens spezifiziert wird (siehe Abbildung 2.3). Es wird angenommen, dass Handschrift in eine Sequenz von Basiseinheiten, sogenannten *Strokes*, zerlegt werden kann, wobei einzelne Strokes jeweils ca. 100 Millisekunden andauern. Die zeitliche Reihenfolge dieser Strokes wird mittels eines weiteren Moduls, in dem die Bewegungsmuster abgelegt sind, festgelegt. Das abgerufene Bewegungsmuster wird dann in einem weiteren Zwischenspeicher vorgehalten, bis die Bewegung initiiert wird.

Das Bewegungsmuster kann zu diesem Zeitpunkt als abstraktes motorisches Programm aufgefasst werden, das den zu schreibenden Buchstaben ideal, d.h. invariant gegenüber äußeren Einflüssen repräsentiert, das aber noch an die konkrete Schreibsituation adaptiert werden muss. So sind eine Reihe von Bewegungsparametern zu spezifizieren, wie z.B. die absolute Schriftgröße, die Startposition und weitere muskel-

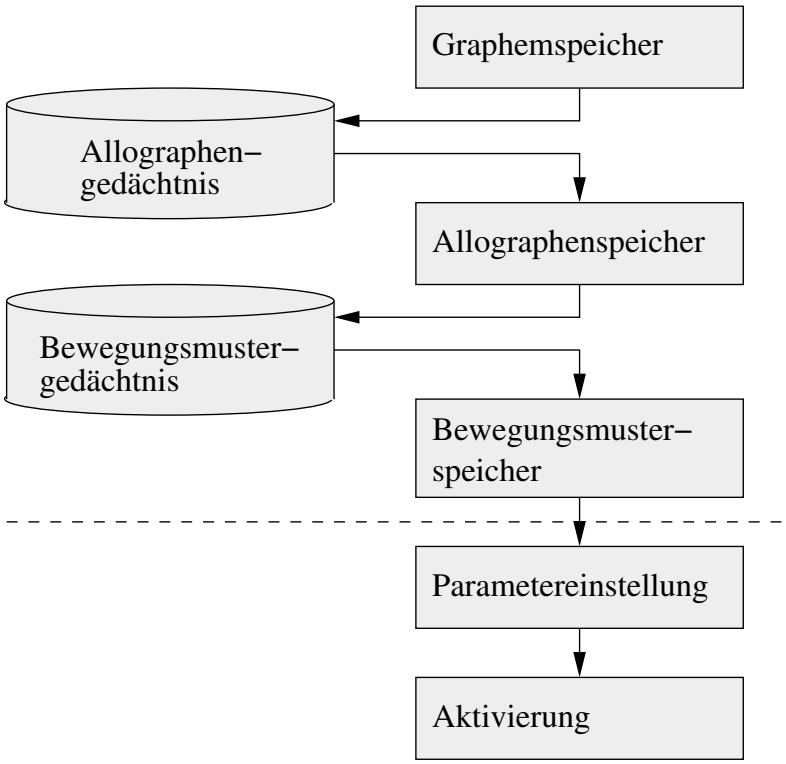


Abbildung 2.2: Makroskopisches Modell der Handschriftgenerierung

spezifische Parameter. Diese Adaption wird in zwei weiteren Modulen vorgenommen [Teu94]: Das erste Modul versieht das motorische Programm mit globalen Parametern, wie z.B. Schriftposition und -größe. Diese Parameter werden als muskelunabhängig beschrieben, da die Schriftgröße in gewissen Grenzen variiert werden kann, ohne dass sich die Funktionen der beteiligten Muskeln ändern. Die Aufgabe des zweiten Moduls ist nun die Aktivierung der entsprechenden Muskeln und das Einstellen muskelspezifischer Parameter, wodurch Orientierung und Neigung der Schrift beeinflusst werden können.

Bei Betrachtung des beschriebenen Modells fällt auf, dass keinerlei Rückkopplung zwischen den Modulen vorgesehen ist. Es wird also die Hypothese zugrundegelegt, dass das Schreiben als ballistische Bewegung aufgefasst werden kann, die ohne unmittelbare Positionsrückmeldung ausgeführt wird. Das motorische Programm legt somit schon zu Beginn eines u.U. mehrere Strokes umfassenden Schriftabschnitts die gesamte Trajektorie der entsprechenden Bewegung fest. Dies wird z.B. daran deutlich, dass eine Veränderung der Reibung zwischen Stift und Papier zu einer sofortigen Veränderung der Schriftgröße führt. Erst nach mehreren Strokes stellt sich die ursprüngliche Schriftgröße aufgrund visueller oder taktiler Rückmeldung wieder ein.

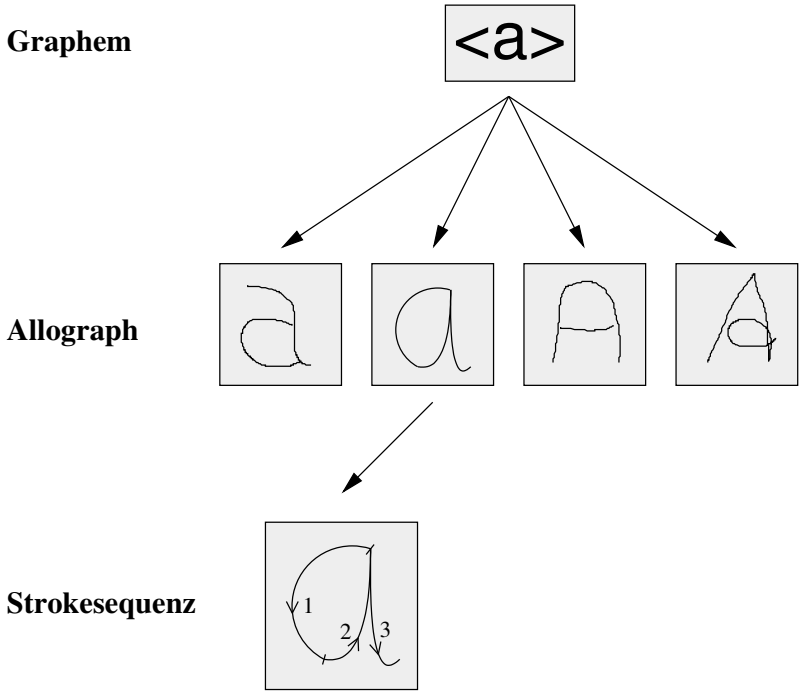


Abbildung 2.3: Graphem-Allograph-Stroke

2.1.2 Kinematisches Modell

Im Gegensatz zum top-down Konzept beschäftigt sich der bottom-up Ansatz mit den biomechanischen Prozessen auf neuromuskulärer Ebene und hat vor allem die kinematische Modellierung der Stiftbewegungen zum Ziel. Die Modelle, die dazu vorgeschlagen wurden, lassen sich in zwei Klassen einteilen: oszillatorische und diskrete Modelle [Pla00]. Die oszillatorischen Modelle gehen von einer Schwingung als Basisbewegung aus und modellieren eine komplexe Schreibbewegung durch Anpassung der Amplitude, Frequenz und Phasenverschiebung der verwendeten Wellenfunktion [Hol81]. Ein einzelner Stroke wird dabei als Spezialfall einer unterbrochenen Schwingung aufgefasst. Im Gegensatz dazu betrachten die diskreten Modelle eine Schreibbewegung als Sequenz bzw. zeitliche Überlagerung einzelner Strokes [Pla00, Mor82].

Ein Vertreter der Klasse der diskreten Modelle ist das *delta-lognormal Modell* von Plamondon, das in einer umfangreichen Vergleichsuntersuchung seine Leistungsfähigkeit in Bezug auf die kinematische Modellierung von Schreibbewegungen unter Beweis gestellt hat und daher im folgenden näher erläutert werden soll [Pla93, Pla98b].

Das Modell, dargestellt in Abbildung 2.4, beschreibt die Generierung eines Strokes als das Zusammenwirken eines agonistischen und eines antagonistischen neuromuskulären Systems zur Geschwindigkeitssteuerung der Stiftspitze. Die synergetische Aktivierung beider neuromuskulärer Systeme durch die Eingabesignale $\boldsymbol{D}_1$ und $\boldsymbol{D}_2$ zum Zeitpunkt t_0 führt zu einer Trajektorie in Form eines Kreisbogens, die von der Startposition P_0, Richtung θ_0 und Krümmung C_0 ausgeht.

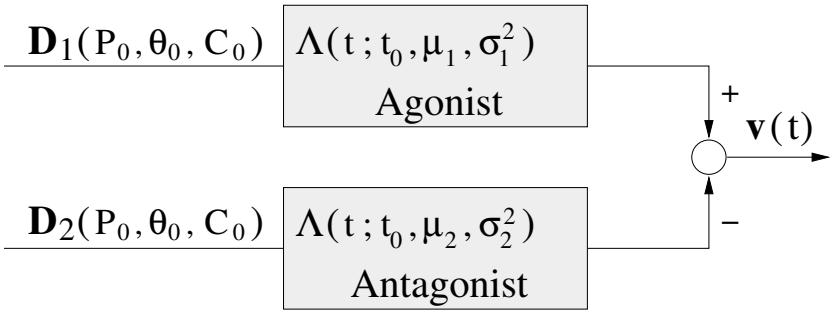


Abbildung 2.4: Delta-Lognormal Modell

Die Impulsantwort der einzelnen Systeme wird asymptotisch beschrieben durch eine logarithmisch-normalverteilte Funktion mit den Zeitkonstanten μ_j und σ_j^2:

$$\Lambda(t; t_0, \mu_j, \sigma_j^2) = \frac{1}{\sigma_j \sqrt{2\pi}(t - t_0)} \exp\left(\frac{-[ln(t - t_0) - \mu_j]^2}{2\sigma_j^2}\right) \tag{2.1}$$

Die Ausgabe des delta-lognormal Modells ist somit die Differenz der mit den Amplituden der Eingabesignale gewichteten Impulsantworten der einzelnen Systeme. Für den Betrag des Geschwindigkeitsvektors gilt demnach:

$$|\boldsymbol{\nu}(t)| = |\boldsymbol{D}_{1_{(P_0,\theta_0,C_0)}} \Lambda(t; t_0, \mu_1, \sigma_1^2) - \boldsymbol{D}_{2_{(P_0,\theta_0,C_0)}} \Lambda(t; t_0, \mu_2, \sigma_2^2)| \tag{2.2}$$

Ein Gütemaß zur Bewertung von Modellen zur Strokegenerierung ist, dass die in Experimenten beobachteten Eigenschaften der Trajektorien durch die Modelle möglichst exakt beschrieben werden können. Zu diesen Eigenschaften zählen im wesentlichen das asymmetrische, glockenförmige Geschwindigkeitsprofil eines Strokes und der Kompromiss zwischen der Geschwindigkeit und Genauigkeit der Bewegung. Dieser Kompromiss bedeutet, dass je schneller die Bewegung ausgeführt wird, desto weniger die resultierende Trajektorie mit dem geplanten Verlauf übereinstimmt.

Abbildung 2.5 zeigt, dass die resultierende Geschwindigkeit, die sich anhand des delta-lognormal Modells ergibt, das geforderte glockenförmige Profil aufweist. Ebenso geht der Kompromiss zwischen Geschwindigkeit und Genauigkeit aus dem Modell hervor. So lässt sich die Dauer eines Strokes in der einfachsten Form folgendermaßen abschätzen [Pla98b]:

$$T_S = K\left(\frac{D}{\Delta D}\right)^{\alpha} \tag{2.3}$$

Hierbei bezeichnet T_S die Strokedauer, D ist die Strokeamplitude mit

$$D = |\boldsymbol{D}_{1_{(P_0,\theta_0,C_0)}} - \boldsymbol{D}_{2_{(P_0,\theta_0,C_0)}}| \tag{2.4}$$

Der absolute Fehler der Strokeamplitude, d.h. die Differenz zum geplanten Bewegungsziel, wird durch ΔD gekennzeichnet, während α und K von μ_j und σ_j^2 abhängige

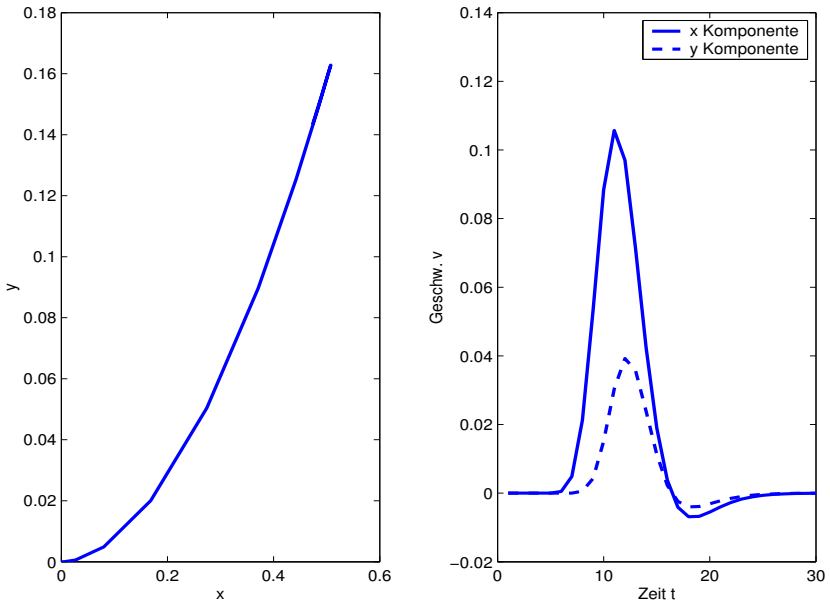


Abbildung 2.5: Durch das delta-lognormal Modell (Gleichung 2.2) generierter Stroke: Links ist der Stroke in Ortskoordinaten dargestellt. Rechts ist der zeitliche Verlauf der Geschwindigkeit in horizontaler bzw. vertikaler Richtung abgebildet.

Konstanten bezeichnen. Man erkennt, dass bei einer vorgegebenen Amplitude D, eine kürzere Bewegungszeit T_S nur durch Inkaufnahme eines größeren Fehlers ΔD zu erreichen ist.

Aus der in Gleichung 2.3 dargestellten Vorhersage der Bewegungsdauer eines Strokes lässt sich darüberhinaus der beobachtete Effekt der Bewegungsantizipation beim Menschen ableiten. Dieser Effekt besteht darin, dass sobald ein Stroke initiiert wird, d.h. $\boldsymbol{D}_1$ und $\boldsymbol{D}_2$ die neuromuskulären Systeme aktivieren, der Mensch die Dauer T_S des Strokes bei einem absoluten Fehler ΔD zum geplanten Bewegungsziel abschätzen kann. So sind die Trajektorien einzelner Strokes nicht nur zu Beginn der Bewegung festgelegt, sondern darüberhinaus kann der nächste Stroke schon initiiert werden, bevor der vorherige abgeschlossen wurde. Der Prozess der Handschriftgenerierung ist also zumindest bei geübten Schreibern nicht als Aneinanderreihung sondern vielmehr als zeitliche *Überlagerung* einzelner Strokes aufzufassen. In vektorieller Schreibweise lässt sich die resultierende Geschwindigkeit, die sich für die Stiftspitze durch Überlagerung einzelner Strokes ergibt, folgendermaßen beschreiben [Pla98b]:

$$\boldsymbol{\nu}(t) = \sum_{i=1}^{n} \boldsymbol{\nu}_i(t - t_{0i}) \tag{2.5}$$

Hierbei ist $\boldsymbol{\nu}_i(t - t_{0i})$ der aus Gleichung 2.2 bekannte Verlauf der Geschwindigkeit des Strokes i. Die jeweiligen Startzeitpunkte t_{0i} definieren die zeitliche Überlagerung

benachbarter Strokes. Der Effekt der zeitlichen Überlagerung von zwei Strokes ist in der Abbildung 2.6 dargestellt.

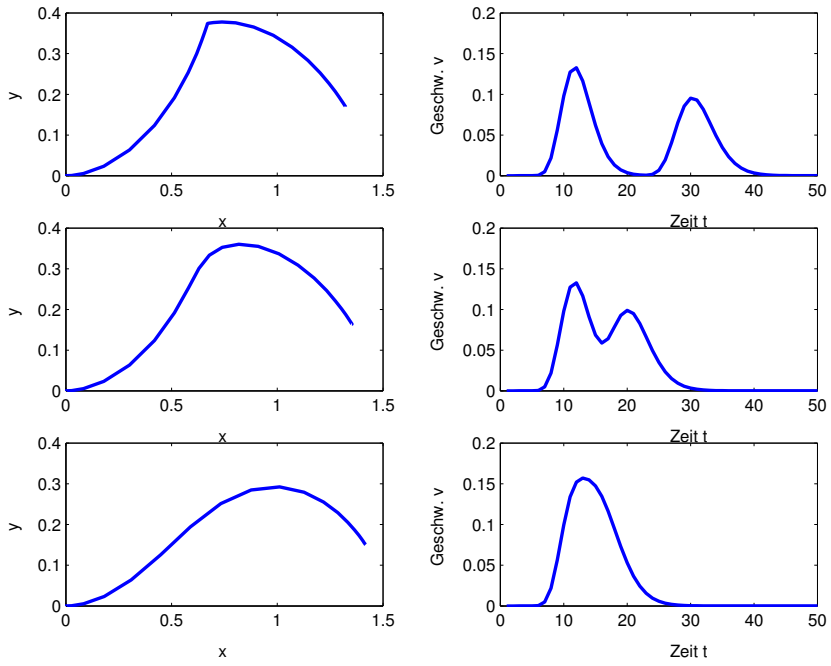


Abbildung 2.6: Mit Hilfe von Gleichung 2.5 berechnete Überlagerung von zwei Strokes: Links ist die resultierende Kurve in Ortskoordinaten dargestellt, rechts ist der Geschwindigkeitsbetrag über die Zeit aufgetragen. Der Grad der Überlagerung nimmt von oben nach unten zu.

Diese kinematische Beschreibung der Handschriftgenerierung durch das delta-lognormal Modell von Plamondon knüpft somit an das zuerst vorgestellte, makroskopische top-down Konzept an. Während das makroskopische Modell die Erstellung des motorischen Programms beschreibt und die muskulären Prozesse weitgehend ausblendet, geht das delta-lognormal Modell von einem motorischen Ablaufplan aus und modelliert die kinematischen Aspekte der Handschriftgenerierung. Das motorische Programm ist hierbei als Sequenz geplanter Bewegungsziele zu verstehen, die durch Überlagerung einzelner Strokes verbunden werden. Die Bewegung der Stiftspitze wird dabei beschrieben durch die synergetische Aktivierung eines agonistischen und eines antagonistischen neuromuskulären Systems.

2.1.3 Zusammenfassung

Welche Einsichten können nun für die Entwicklung technischer Systeme zur Handschrifterkennung aus dem Verständnis der Handschriftgenerierung heraus gewonnen werden? Eine wesentliche Erkenntnis ist, dass der Schriftzug aus einer Überlagerung von Basiseinheiten, den Strokes, besteht. Es erscheint daher sinnvoll, die für die Erken-

nung genutzten Merkmale anhand der einzelnen Strokes zu berechnen, die wiederum mit Hilfe eines kinematischen Bewegungsmodells, wie z.B. des delta-log-normal Modells, aus dem Schriftzug extrahiert werden können. Diese Segmentierung in Strokes ist dabei besonders für die Systeme zur Handschrifterkennung relevant, die auf der Extraktion dynamischer Bewegungsinformationen basieren.

Neben der Segmentierung geben die Konzepte der Handschriftgenerierung auch Aufschluss über geeignete Normierungsmaßnahmen, die auf den Schriftzug angewendet werden können. Geht man davon aus, dass das abstrakte motorische Programm im Bewegungsmustergedächtnis als ideale Repräsentation des zu schreibenden Buchstabens aufgefasst werden kann, so könnten daraus bzgl. der konkreten Schreibsituation invariante Merkmale extrahiert werden. Daher sind gerade die Parameter zu normieren, die das abstrakte motorische Programm an die konkrete Schreibsituation anpassen und mithin die Invarianz der Merkmale verringern [Teu94]. Es ist also eine Normierung bezüglich der globalen, muskelunabhängigen Parameter wie der Schriftposition und -größe ebenso erforderlich wie die Normierung muskelabhängiger Parameter wie der Schriftneigung und der Orientierung der Basislinie.

2.2 Perzeption von Handschrift

Nachdem im vorigen Abschnitt die kognitiven und kinematischen Vorgänge, die sich bei der Handschriftgenerierung abspielen, betrachtet wurden, wird in diesem Abschnitt auf das Lesen, also das visuelle Erfassen der Bedeutung von Texten, näher eingegangen. So ist besonders das Lesen handschriftlicher Texte eine bemerkenswerte perzeptive Fähigkeit, weil die Schriftbilder insbesondere bei unterschiedlichen Schreibern stark variieren. Auf Grund der enorm leistungsfähigen Schriftperzeption des Menschen ist die Analyse der zugrundeliegenden Konzepte nach Ansicht vieler Forscher ein erfolgversprechender Ansatz, um die Performanz technischer Systeme zur Handschrifterkennung zu verbessern [Sch99, Bra95, Côt98].

2.2.1 Analyse der Augenbewegungen

Die Bewegungen der Augen sind die einzigen sichtbaren Merkmale, die beim Lesen zu beobachten sind. Um die beim Lesen involvierten kognitiven Prozesse zu untersuchen, wird daher zuerst der Frage nachgegangen, ob die Bewegungen der Augen im Zusammenhang mit dem zu lesenden Text stehen und ob daraus Rückschlüsse über die kognitiven Prozesse des Lesens gezogen werden können.

Die dazu vorgenommenen Experimente, die jedoch größtenteils nicht auf handschriftlichen Texten sondern auf gedruckter Schrift basieren, haben ergeben, dass der Blick nicht gleichmäßig über die Zeile wandert, sondern dass mittels sprunghafter Bewegungen, sogenannter *Sakkaden*, bestimmte *Fixationspunkte* der Textzeile angesteuert werden, an denen die visuellen Informationen extrahiert werden [Ray89]. Wie in Abbildung 2.7 deutlich wird, unterliegt sowohl die Dauer der Fixationen als auch die

Länge der Sakkadensprünge starken Schwankungen, wobei im Mittel der Blick auf einem Fixationspunkt 250ms verweilt und bei einer Sakkade, die ca. 20 bis 35ms andauert, durchschnittlich 7 bis 9 Buchstaben nach rechts springt. Rückwärtssprünge, sogenannte regressive Sakkaden (Fixation 12 in der Abbildung), machen dabei 10 bis 15% aller Sakkaden aus.

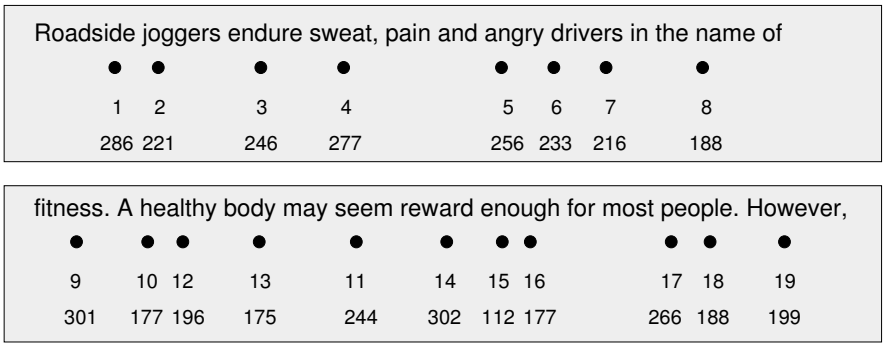


Abbildung 2.7: Fixationspunkte und -zeiten (in Millisekunden) für zwei Textzeilen. (nach [Ray89])

Der Textausschnitt, der während einer Fixationsphase in den fovalen Bereich des Auges fällt, also mit höchster Sehschärfe wahrgenommen wird, umfasst maximal sieben um den Fixationspunkt zentrierte Buchstaben. Es wird beim Lesen jedoch ein größerer perzeptiver Bereich genutzt, der in aller Regel das gerade fixierte Wort und bis zu 15 Buchstaben im parafovalen Bereich rechts mit einschließt. Damit können in einer Fixationsphase das gerade fixierte Wort und maximal ein bis zwei in Leserichtung benachbarte Wörter identifiziert werden. In den meisten Fällen wird jedoch nur das fixierte Wort vollständig identifiziert, wohingegen das benachbarte Wort höchstens partiell erkannt wird. Dabei kann z.B. nur der Anfangsbuchstabe erkannt werden, oder es werden vorläufige Annahmen, z.B. über das Vorhandensein von Ober- bzw. Unterlängen, über die folgenden Buchstaben gemacht. Diese Informationen werden in der darauffolgenden Fixationsphase genutzt, um die vollständige Identifikation des Wortes zu ermöglichen.

In diesem Zusammenhang ist mit Wortidentifikation der *lexikalische Zugriff* gemeint, d.h. die Entscheidung, um welches Wort es sich handelt. Dabei können einem mehrdeutigen Wort auch mehrere semantische Bedeutungen zugewiesen werden. Diese Ambiguität wird erst durch Integration von Kontextwissen in einem weiteren Schritt aufgelöst, der lexikalische Zugriff ist davon unbeeinflusst.

Doch welche Faktoren sind es, die die Schwankungen der Fixationszeiten und Sakkadenlängen verursachen? In der Literatur lassen sich einige Experimente finden, die zur Beantwortung dieser Frage unternommen wurden (siehe z.B. [Ray89]). Demzufolge werden Fixationsdauer und Sakkadenlänge unabhängig voneinander von unterschiedlichen Faktoren beeinflusst. Die Fixationsdauer und die Sakkadenlänge stehen sowohl unter direkter Kontrolle, ausgelöst durch die ausschließlich am aktuellen Fixationspunkt extrahierten Informationen, als auch unter kognitiver Kontrolle, bei der

weitere im Satzzusammenhang erschlossene Kontextinformationen eingehen. So wurde gezeigt, dass die Sakkadenlänge von der Länge des Wortes abhängig ist, das dem Fixationspunkt in Leserichtung benachbart ist, da bei längeren Wörtern der Blick weiter nach rechts zu springen tendiert als bei kürzeren Wörtern. Darüberhinaus besteht ein starker Zusammenhang zwischen Sakkadenlänge und der Häufigkeit des Vorkommens des benachbarten Wortes im Sprachgebrauch. Insbesonders kurze Wörter, die häufiger verwendet werden oder sich leicht aus dem Satzkontext vorhersagen lassen, werden oftmals mittels einer längeren Sakkade übersprungen und eher mit Hilfe visueller Informationen aus dem parafovalen Bereich identifiziert. Die regressiven Sakkaden kommen dann vor, wenn bei der Analyse des aktuellen Wortes festgestellt wird, dass das bisher Gelesene wohl missverstanden wurde und daher ein Rücksprung im Text notwendig ist. In ähnlicher Weise ist auch die Fixationsdauer von mehreren Faktoren abhängig. Kommt z.B. das gerade fixierte Wort häufig im Sprachgebrauch vor oder ist es leicht aus dem Zusammenhang zu erschließen, so kann die Wortidentifikation schneller geschehen und damit eine kürzere Fixationszeit erreicht werden.

2.2.2 Wortidentifikation

Der Prozess der Wortidentifikation ist also ein wesentlichen Faktor, der die Augenbewegungen während des Lesens in hohem Maße bestimmt. Die Wortidentifikation, d.h. der lexikalische Zugriff, wurde bisher jedoch als abstrakter Prozess aufgefasst, der nun näher beschrieben werden soll. Die zentrale Frage lautet also: *Wie* wird anhand seines Schriftbildes ein Wort identifiziert, d.h. sein Lexikoneintrag gefunden?

Dazu wurden in der Vergangenheit mehrere Theorien aufgestellt (siehe dazu [Ray89]). So wird z.B. die Wortidentifikation auf die Erkennung der einzelnen Buchstaben zurückgeführt, wobei die Buchstaben seriell von links nach rechts verarbeitet werden. Eine gegensätzliche Theorie besagt, dass Wörter eher als visuelle Schablonen, ähnlich wie Bilder, unter Umgehung der Einzelbuchstaben erkannt werden. Aufgrund vielfältiger Experimente herrscht mittlerweile allerdings die Meinung vor, dass die Wahrheit eher zwischen diesen beiden Theorien liegt.

So wird die erste Theorie, die auf der seriellen Buchstabenerkennung aufbaut, durch den *Word Superiority* Effekt weitgehend widerlegt. Dieser Effekt besteht in der kürzeren Erkennungszeit eines Buchstabens im Wortkontext gegenüber eines isolierten Buchstabens. Für die Identifikation eines gesamten Wortes wird sogar oftmals weniger Zeit benötigt als für einen einzelnen Buchstaben. Die Worterkennung kann damit nicht auf der seriellen Buchstabenerkennung basieren, denn eine Konsequenz dieser Theorie ist ja, dass die Worterkennung länger dauern müsste als die Buchstabenerkennung.

Die zweite Theorie, bei der die Wortidentifikation mit Hilfe visueller Schablonen vorgenommen wird, würde zwar den Word Superiority Effekt gut erklären, diese Theorie ist jedoch aus folgendem Grund wenig plausibel: Wäre die Worterkennung losgelöst von der Buchstabenerkennung, so müsste für jedes Wort eine visuelle Schablone vorliegen. Nach dieser Theorie müssten Probleme auftreten, wenn ein Wort gelesen werden soll, das in einem unterschiedlichen bzw. unbekannten Schriftstil geschrie-

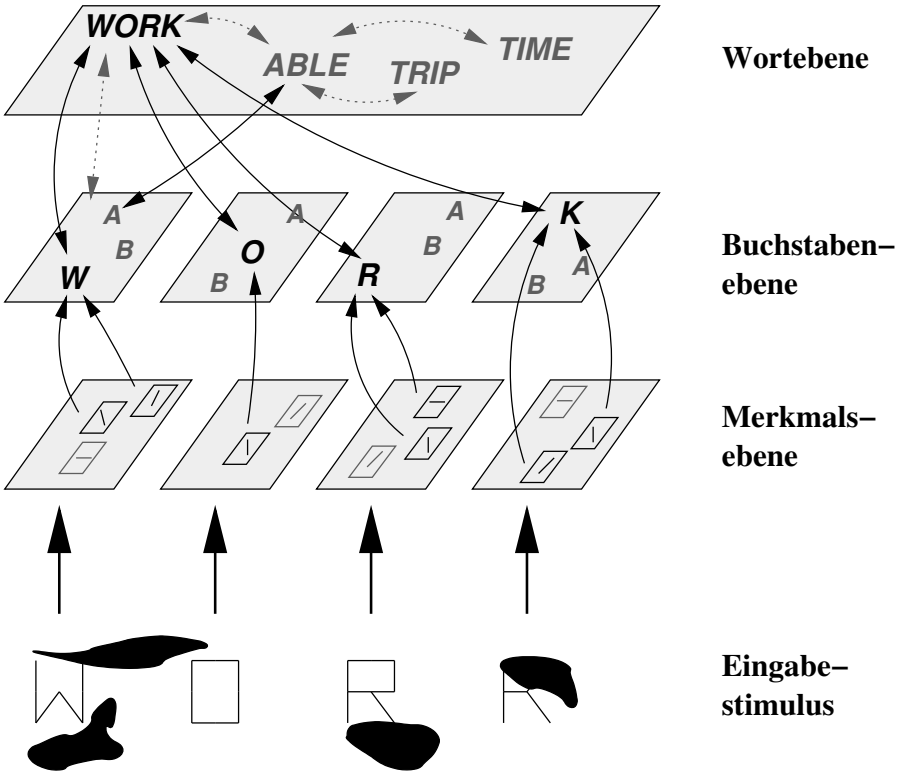


Abbildung 2.8: Interactive-Activation Modell: Die inhibitorischen Verknüpfungen sind im Gegensatz zu den exzitatorischen gepunktet dargestellt. Der Übersicht halber sind nicht alle Verschaltungen eingezeichnet [Côt98, McC81].

ben wurde. Die zu dieser Fragestellung vorgenommenen Experimente haben jedoch gezeigt, dass nach kurzer Eingewöhnungszeit kaum spürbare Verzögerungen bei der Wortidentifikation auftreten.

Die Modelle zur Wortidentifikation, die dagegen konform mit Experimenten zur visuellen Wahrnehmung gehen, weisen als gemeinsamen zentralen Bestandteil einen *Aktivierungsmechanismus* auf. Hierbei werden aufgrund visueller Stimuli die Evidenzen für Wort- bzw. Buchstabenkonzepte schrittweise erhöht, bis bei einem bestimmten Aktivierungsgrad das entsprechende Wort identifiziert werden kann. Vertreter dieser Aktivierungsmodelle sind u.a. das *logogen* Modell [Mor69], das *activation-verification* Modell [Paa82] und das *interactive-activation* Modell [McC81].

In Abbildung 2.8 ist das *interactive-activation* Modell dargestellt [McC81]. Diesem Modell der visuellen Wortperzeption liegt die Annahme zugrunde, dass die Verarbeitung in mehreren Abstraktionsebenen erfolgt, die jeweils über unterschiedliche Repräsentationsformen des Eingabesignals verfügen. Die Merkmalsebene bildet dabei die niedrigste Abstraktionsstufe, gefolgt von der Buchstaben- und der Wortebene. Innerhalb der Ebenen werden die einzelnen Merkmale, Buchstaben und Wörter durch jeweils eigene Knoten repräsentiert, die mit Knoten derselben Ebene über inhibitorische

und mit Knoten benachbarter Ebenen über exzitatorische und inhibitorische Verbindungen verknüpft sind. Die einzelnen Knoten besitzen ihrerseits einen momentanen Aktivierungsgrad, der über die Verschaltungen erhöht bzw. vermindert werden kann.

Weiterhin ist das Modell charakterisiert durch eine parallele Verarbeitungsstrategie, wobei sich die Parallelität auf zwei verschiedene Aspekte bezieht. So wird zum einen von einer räumlichen Parallelität ausgegangen, dass nämlich die visuellen Informationen eines Bereiches, der z.B. ein vier Buchstaben langes Wort umfasst, gleichzeitig verarbeitet werden können. Der zweite Aspekt besteht darin, dass die Verarbeitung auch auf unterschiedlichen Abstraktionsebenen gleichzeitig stattfindet.

Das wesentliche Merkmal des interactive-activation Modells ist jedoch die Modellierung der visuellen Wortidentifikation als interaktiver Prozess. Dabei interagieren top-down und bottom-up Verarbeitung in der Form, dass z.B. Knoten der Buchstabenebene über exzitatorische und inhibitorische Verschaltungen Knoten der Wortebene aktivieren bzw. unterdrücken, die ihrerseits wieder auf Knoten der Buchstabenebene rückwirken. Diese rekurrente Beeinflussung besteht im interactive-activation Modell jedoch aus Komplexitätsgründen nur zwischen Buchstaben- und Wortebene, zwischen Merkmals- und Buchstabenebene ist ausschließlich ein bottom-up Informationsfluss vorgesehen.

Wird dem interactive-activation Modell ein Eingabesignal präsentiert, so läuft der Prozess der Wortidentifikation folgendermaßen ab: Die visuellen Stimuli initiieren die Merkmalsextraktion, sodass die entsprechenden Knoten der Merkmalsebene aktiviert werden. Daraufhin wird der Aktivierungsgrad bestimmter Knoten der Buchstabenebene erhöht, während er bei anderen Knoten durch inhibitorische Einflüsse vermindert wird. Die Buchstabenknoten verstärken dann wiederum den Aktivierungsgrad der Wortknoten, die konsistent zu den Buchstabenknoten sind. Sind beispielsweise die Buchstabenknoten ’W’ und ’O’ aktiv, so würden u.a. die Wortknoten ’WORD’ und ’WORK’ aktiviert, wohingegen die Aktivierung der Wortknoten, die im Widerspruch zu den Buchstabenknoten stehen, abgeschwächt würde. Die Wortknoten wiederum versuchen sich über die inhibitorischen Verbindungen gegenseitig abzuschwächen und wirken außerdem auf die Knoten der Buchstabenebene zurück. Entsprechen die extrahierten Merkmale denen einer Buchstabensequenz und bildet diese ein Wort, welches in der Wortebene repräsentiert ist, so wird die Verarbeitung konvergieren und der Aktivierungsgrad der entsprechenden Worthypothese über konkurrierende Hypothesen dominieren.

In Abbildung 2.9 ist der Verlauf des Aktivierungsgrades verschiedener Wort- und Buchstabenknoten über die Zeit dargestellt, der sich für das Eingabesignal aus Abbildung 2.8 ergibt. Man erkennt, dass das System nach kurzer Zeit die Hypothese ’WORK’ favorisiert, während die konkurrierende Hypothese ’WORD’ aufgrund der geringen Aktivierung des Buchstabens ’D’ verworfen wird.

Die Modelle zur Wortidentifikation, wie z.B. das hier angeführte interactive-activation Modell, beschreiben detailliert die Generierung von Worthypothesen, sie setzen jedoch in aller Regel voraus, dass bereits Verfahren zur Extraktion von Merkmalen aus den visuellen Stimuli vorliegen. Neben der Wortidentifikation ist aber die

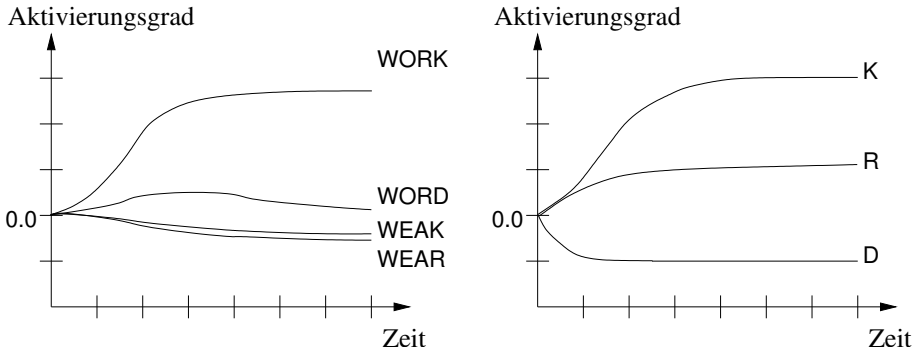


Abbildung 2.9: Aktivierungsgrad von Wort- und Buchstabenknoten (nach [McC81])

Merkmalsextraktion der zweite wichtige Aspekt, der von Systemen zur Handschrifterkennung modelliert werden muss. Im folgenden wird daher der Frage nachgegangen, welche Merkmale der Mensch aus den visuellen Mustern extrahiert und wie diese Merkmale zur Wortidentifikation genutzt werden können.

2.2.3 Visuelle Schriftmerkmale

Eine mögliche Vorgehensweise, um Informationen über die zum Lesen verwendeten Merkmale zu gewinnen, besteht darin, die Fehler beim Lesen zu analysieren, wobei als Fehler in diesem Sinne gilt, wenn ein anderes Wort als das geschriebene gelesen wird (siehe u.a. [Gib78]). Ein Vergleich des geschriebenen und irrtümlich gelesenen Wortes gibt dann Aufschluss über die verwendeten Merkmalsinformationen. Dieser Ansatz hat den Vorteil, dass das Eingabesignal, also der zu lesende Text, in keiner Weise verändert werden muss und somit keine Artefakte eingeführt werden. Es hat sich gezeigt, dass ein Wort häufig mit einem anderen verwechselt wird, wenn beide Wörter in etwa gleich lang sind und die gleichen Anfangs- und Endbuchstaben besitzen. Außerdem unterscheiden sich bei Verwechslungen die Konturen beider Wörter nur wenig, d.h. die jeweilige Abfolge von Ober- bzw. Unterlängen ist sehr ähnlich. Eine wichtige Erkenntnis ist, dass es sich bei diesen Merkmalen gerade um die Informationen handelt, die durch peripheres Sehen wahrgenommen werden, das jeweilige Wort somit nicht in den Fixationsbereich sondern in den parafovalen Bereich des Sehens fällt.

Oftmals werden auch manipulierte Texte zur Bestimmung relevanter Merkmale verwendet. Dabei wird das Eingabemuster z.B. dahingehend verändert, dass bestimmte Bereiche des Schriftzuges abgedeckt werden. Anhand der fehlerhaft gelesenen Wörter und der Lesegeschwindigkeit können dann wiederum Rückschlüsse über die zur Wortidentifikation verwendeten Merkmale gezogen werden. Auch dieser Ansatz hat ergeben, dass der Anfangs- und Endbuchstabe und die Wortkontur die wesentlichen Merkmale zum Lesen sind. Weitere relevante Merkmale sind außerdem Kreuzungspunkte und stark gekrümmte Abschnitte des Schriftzuges (siehe u.a. [Sch99]).

Um die Bedeutung der Wortkontur für die Erkennung genauer zu untersuchen, können die Buchstaben gemäß ihrer vertikalen Ausdehnung in drei Klassen eingeteilt werden: *Short letters*, die keine Ober- bzw. Unterlängen aufweisen, *tall letters*, die ausschließlich Oberlängen besitzen und *projecting letters*, die Unterlängen aufweisen. Die Kontur eines Wortes ergibt sich damit direkt anhand der Buchstabenabfolge. Das englische Wort 'bay' hätte demnach die Kontur *tall - short - projecting* und würde eher mit dem Wort 'beg' verwechselt als mit den Wörtern 'may' oder 'by', obwohl diese mehr gemeinsame Buchstaben aufweisen [Bra95]. Bouma hat diese Kategorien nach verfeinerten Kriterien weiter unterteilt, sodass die in der Tabelle 2.1 dargestellte Gruppierung in sieben Klassen resultierte [Bou71].

Short				Tall		Projecting
1	2	3	4	5	6	7
a s z x	e o c	r v w	n m u	d h k b	t i l f	g j p q y

Tabelle 2.1: Buchstabengruppierung nach Bouma

Demnach besitzt das Wort 'reading' die *Bouma-Kontur* 3215647. Für die gebräuchlichsten 20000 Wörter der englischen Sprache ergeben sich damit 18084 unterschiedliche Bouma Konturen, sodass die Wahrscheinlichkeit schon 90% beträgt, dass ein Wort ausschließlich anhand seiner Bouma Kontur korrekt identifiziert wird. Dabei wird jedoch die exakte Bestimmung der Bouma Kontur vorausgesetzt, was für maschinengeschriebene Dokumente noch eher durchführbar ist als für allgemeine handschriftliche Texte, wo schon die Segmentierung in einzelne Buchstaben ein ungelöstes Problem darstellt [Bra95].

2.2.4 Zusammenfassung

Welches sind nun also die zentralen Punkte, die die Schriftperzeption beim Menschen charakterisieren? Eine Feststellung ist, dass der Blick während des Lesens nicht gleichmäßig von links nach rechts über die Zeile wandert, sondern abhängig vom Prozess der Wortidentifikation bestimmte Fixationspunkte angesprungen werden. Die wesentliche Erkenntnis ist dabei, dass die Modelle zur Wortidentifikation, die die beobachteten Effekte der visuellen Wahrnehmung (z.B. word superiority Effekt) erklären, auf einer parallelen Verarbeitung der Buchstaben beruhen. Weiterhin sind diese Modelle durch einen Aktivierungsmechanismus gekennzeichnet, sodass ausgehend von der Merkmalsebene bestimmte Buchstaben als nächsthöhere Konzepte aktiviert werden, die wiederum die Konzepte der Wortebene aktivieren. Im Falle des vorgestellten interactive-activation Modells findet darüberhinaus sogar eine Interaktion von bottom-up und top-down Konzepten statt. Ein wesentliches Merkmal, das aus den visuellen Mustern extrahiert und zur Wortidentifikation herangezogen wird, ist die Wortkontur, d.h. die Abfolge von Ober- bzw. Unterlängen. Daneben kommt noch den Anfangs- und Endbuchstaben, Kreuzungspunkten und stark gekrümmten Segmenten eine größere Bedeutung zu.

3 Automatische Handschrifterkennung – Grundlagen und Stand der Forschung

Nachdem im vorigen Kapitel die Konzepte beschrieben wurden, die bei der Schriftperzeption beim Menschen mitwirken, sollen in diesem Abschnitt die Verfahren vorgestellt werden, die in technischen Systemen zur Handschrifterkennung eingesetzt werden. Betrachtet man dafür den gegenwärtigen Stand der Forschung, so muss man feststellen, dass sich videobasierte Systeme – das Thema dieser Arbeit – kaum finden lassen und eher in Ansätzen existieren. Demgegenüber dominieren die Systeme, die Scanner bzw. Digitalisiertabletts für die Aufnahme der Handschrift verwenden. Die einzelnen Verfahrensschritte sind jedoch oftmals unabhängig vom Aufnahmegerät, sodass sich die in den nicht-videobasierten Systemen eingesetzten Methoden häufig erfolgreich auch auf videobasierte Systeme übertragen lassen. Die besonderen Charakteristika der videobasierten Systeme, die über die im folgenden beschriebenen Verfahren hinausgehende Verarbeitungsschritte erfordern, werden dann ausführlich im nächsten Kapitel vorgestellt.

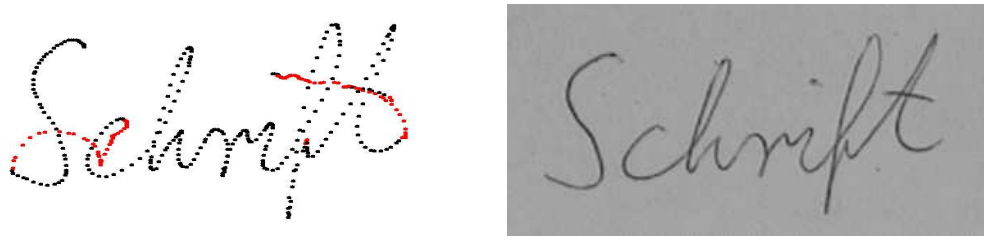
Begonnen wird dieses Kapitel mit einer überblicksweisen Vorstellung der unterschiedlichen Verarbeitungsstrategien, anhand derer sich die meisten Systeme zur Handschrifterkennung strukturieren lassen. Anschließend werden die grundlegenden Verarbeitungsschritte von der Signalaufnahme über die Vorverarbeitung, Segmentierung und Merkmalsextraktion bis hin zur Klassifikation und Adaption näher erläutert. Daran schließt sich eine knappe Vorstellung einiger Datensammlungen an, die zu Trainings- und Evaluationszwecken der Systeme verwendet werden, bevor im letzten Abschnitt eine kurze Zusammenfassung gegeben wird.

3.1 Verarbeitungsstrategien

In diesem Abschnitt wird eine Kategorisierung der bei der automatischen Handschrifterkennung eingesetzten Verarbeitungsstrategien vorgenommen. Die Einordnung wird dabei sowohl hinsichtlich der Signalaufnahme und Datenrepräsentation durchgeführt als auch in Bezug auf die verwendete Segmentierungsstrategie des Eingabesignals.

3.1.1 online vs. offline

Die Methoden zur Signalaufnahme und -verarbeitung, die bei der automatischen Handschrifterkennung eingesetzt werden, lassen sich in *online* und *offline* Verfahren unterscheiden. Die online Verfahren ermitteln *während* des Schreibprozesses in bestimmten



(a) (b)

Abbildung 3.1: (a) Stifttrajektorie für die online Handschrifterkennung (rot: Stift nicht aufgesetzt). (b) Schriftbild für die offline Handschrifterkennung.

Abtastintervallen die Stiftkoordinaten, die schritthaltend mit dem Schreibvorgang weiterverarbeitet werden können. Dagegen ist der *offline* Ansatz dadurch gekennzeichnet, dass *nach* Beendigung des Schreibprozesses ein Abbild des Textdokumentes aufgenommen und entsprechend weiterverarbeitet wird (siehe Abbildung 3.1).

Die Repräsentation der Eingabedaten bei der online Handschrifterkennung erfolgt in Form einer zeitlich geordneten Sequenz von Koordinatenpunkten, die die Trajektorie der Stiftspitze beschreibt[1]. Dabei handelt es sich also um eine raum-zeitliche Darstellung, die die Extraktion dynamischer Bewegungsinformationen erlaubt. Im Gegensatz dazu basiert die Verarbeitung bei der offline Handschrifterkennung auf einer bildhaften Repräsentation der Schrift. Anhand dieses statischen Abbildes der Schrift können dynamische Bewegungsinformationen nicht ohne weiteres ermittelt werden.

3.1.2 analytisch vs. holistisch

Orthogonal zu der Klassifikation in online bzw offline Systeme lässt sich außerdem eine Unterscheidung der Verarbeitungsstrategien in *holistische* bzw. *analytische* Verfahren vornehmen. Im *holistischen* Ansatz wird das Wort global betrachtet, d.h. die Erkennung ist wortbasiert und hängt nicht von der Identifikation von Wortuntereinheiten ab, wie z.B. Buchstaben oder Strokes. Im Gegensatz dazu ist der *analytische* Ansatz gerade dadurch gekennzeichnet, dass das Eingabesignal in eine Sequenz von kleineren Basiseinheiten zerlegt wird, sodass die Worterkennung entscheidend auf der Erkennung jener Basiseinheiten beruht.

In Abbildung 3.2 sind die Verarbeitungsstrategien gegenübergestellt. Beginnend mit der Signalaufnahme, die im Bereich der online Erkennung die zeitliche Abfolge der Stiftpositionen bzw. im offline Bereich ein Abbild des geschriebenen Textes erfasst, folgt die Vorverarbeitung des aufgenommenen Signals. Ziel der Vorverarbeitung ist die Qualitätsverbesserung des Signals. Das schließt z.B. die Glättung und Rauschunterdrückung ebenso ein, wie die Normalisierung von Schriftneigung und –rotation.

[1] Die Trajektorie der Stiftspitze wird im folgenden der Einfachheit halber auch als Stifttrajektorie bezeichnet.

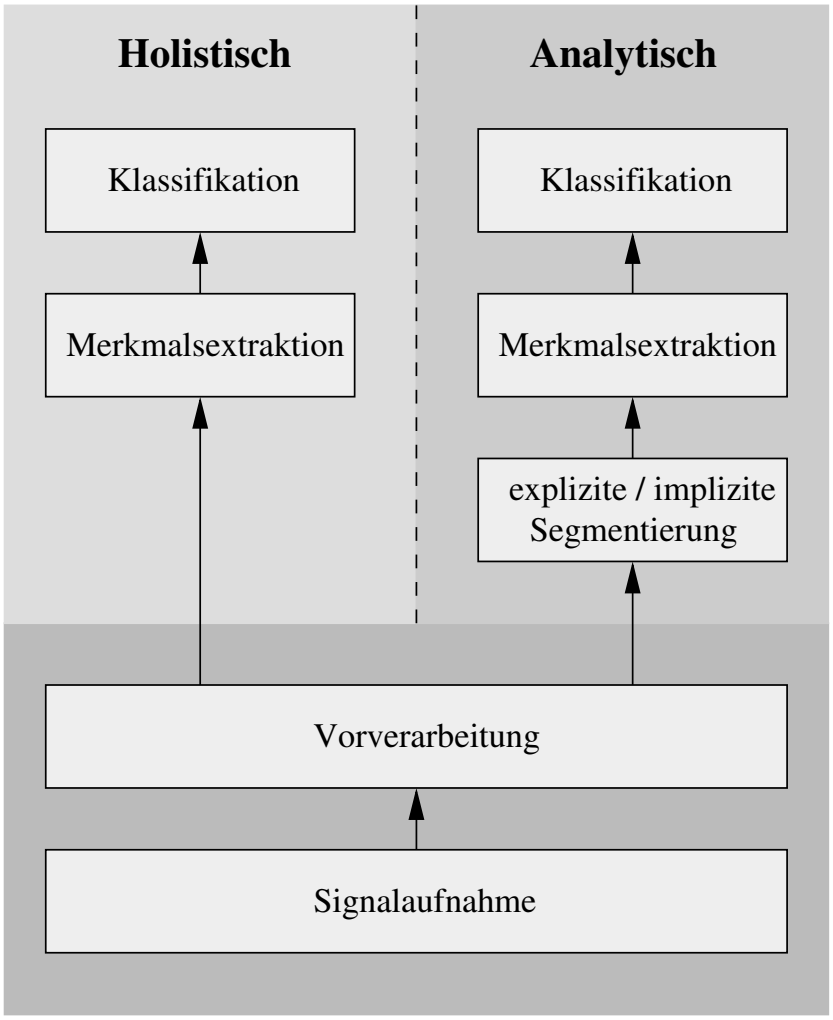


Abbildung 3.2: Verarbeitungsstrategien zur automatischen Handschrifterkennung

Im Falle der holistischen Erkennungsstrategie werden nun anhand des vorverarbeiteten Signals Merkmale berechnet, die in der darauffolgenden Phase zur Klassifikation eingesetzt werden. Die Merkmalsberechnung und Erkennung erfolgt dabei auf Wortebene, sodass die fehleranfällige Segmentierung des Wortes in Untereinheiten vermieden werden kann. Der entscheidende Nachteil der holistischen Variante ist jedoch die Abhängigkeit der Erkennung von einem vorgegebenen, statischen Lexikon. Da die Basiseinheit für die Merkmalsextraktion und die Erkennung das gesamte Wort ist, muss für jedes Wort eine eigene Repräsentation vorliegen, die für die Erkennung verwendet wird. Dadurch ist eine Erweiterung des Lexikons mit einigem Aufwand verbunden.

Demgegenüber stehen die analytischen Ansätze, bei denen die Worterkennung auf der Erkennung von Wortuntereinheiten basiert. Die Verarbeitung beginnt ebenfalls mit den Modulen Signalaufnahme und Vorverarbeitung, die unabhängig von der gewählten Erkennungsstrategie sind. Anschließend findet jedoch eine Segmentierung des vorver-

arbeiteten Signals in Abschnitte statt, anhand derer Merkmalsextraktion und Klassifikation vorgenommen werden.

Je nach verwendetem Erkennungsalgorithmus unterscheidet man bei der Segmentierung zwei gegensätzliche Richtungen: Bei der *expliziten* Segmentierung wird das Wort in einzelne, symbolische Basiseinheiten, meistens Buchstaben, zerlegt, die isoliert voneinander klassifiziert werden. Die Segmentierung heißt dagegen *implizit*, wenn sie als Nebenprodukt aus der Erkennung hervorgeht, daher auch der Name *erkennungsbasierte Segmentierung*. Dieser Ansatz kommt ohne ein kompliziertes Segmentierungsverfahren aus, da das Eingabesignal systematisch in viele kurze Segmente zerlegt wird, sodass z.B. ein Buchstabe durchaus in mehrere Segmente zerfallen kann. Die plausibelste Segmentierung des Signals in Buchstaben wird hierbei im Zuge der Klassifikation ermittelt, indem die möglichen Segmentierungshypothesen analysiert werden.

3.1.3 Vergleich mit der Schriftperzeption beim Menschen

Die hierarchische Verarbeitungsstrategie, bei der die Module von der Signalaufnahme bis hin zur Klassifikation sequentiell abgearbeitet werden, ist kennzeichnend für die meisten Systeme zur automatischen Handschrifterkennung. Dieser Ansatz entspricht jedoch nicht der Art und Weise, wie der Mensch bei der Schriftperzeption vorgeht (siehe Abschnitt 2.2). Da die bisherigen technischen Systeme dem Menschen in der Erkennungsleistung aber weit unterlegen sind, beschäftigen sich einige Forscher mit der Entwicklung von Erkennungssystemen, die eher nach der perzeptionsorientierten Methode vorgehen und sehr stark an die von Psychologen aufgestellten Modelle zum lexikalen Zugriff angelehnt sind [Bra95][Côt98].

Die perzeptionsorientierten Systeme sind dadurch charakterisiert, dass zuerst in einem bottom-up Prozess robuste Merkmale wie z.B. Ober- und Unterlängen oder sogenannten Key-Letters im Eingabesignal identifiziert werden, um damit bestimmte Wörter des Lexikons zu aktivieren. Die Positionen der Merkmale im Eingabesignal dienen daraufhin als Ankerpunkte, um im anschließenden top-down Prozess, ausgehend von den aktivierten Wörtern, hypothetisierte Merkmale anhand des Signals verifizieren zu können. Diese beiden Verarbeitungsschritte, der Aktivierungs- und Verifikationsprozess, können dabei solange iteriert werden, bis der Aktivierungsgrad eines Wortkandidaten deutlich dominiert und somit als Klassifikationsergebnis feststeht [Côt98]. Die Ermöglichung der Interaktion von bottom-up und top-down Prozessen ist allerdings gleichzeitig mit einer Erhöhung der Berechnungskomplexität verbunden, sodass die perzeptionsorientierte Vorgehensweise zur Zeit allenfalls für kleine Lexika realisierbar scheint.

3.2 Signalaufnahme

Am Anfang des Verarbeitungsprozesses von Systemen zur automatischen Handschrifterkennung steht naturgemäß die Signalaufnahme. Die dabei überwiegend eingesetzten Aufnahmegeräte sind Digitalisiertabletts im online Bereich und Scanner im offline Bereich. Videokameras spielen als Signalaufnahmegeräte für die Handschrifterkennung im Gegensatz dazu nur eine äußerst kleine Rolle.

3.2.1 Digitalisiertabletts

Die den Digitalisiertabletts zugrundeliegende Technologie hat sich in den vergangenen Jahrzehnten stets weiterentwickelt. Insbesonders die Verbesserungen von Auflösung und Abtastrate haben die Verwendung von Digitalisiertabletts für die online Handschrifterkennung ermöglicht. So erreichen moderne Geräte Auflösungen von 2500 Linien pro Zoll und Abtastraten von mehr als 200 Hz. Je nach Hersteller und Verwendungszweck werden unterschiedliche Techniken zur Ermittlung der Stiftposition eingesetzt, wovon an dieser Stelle zwei exemplarisch herausgegriffen werden sollen.

Ein sehr populäres Verfahren basiert auf elektromagnetischer Resonanz. Dabei senden unter der Tablettoberfläche horizontal und vertikal ausgerichtete Antennen elektromagnetische Wellen aus, die einen Schwingkreis im Stiftinneren zu Schwingungen anregen. Im zweiten Schritt schalten die Antennen des Tabletts vom Sende- in den Empfangsmodus, empfangen das Schwingungssignal und ermöglichen so die Lokalisation des Stifts. Darüberhinaus kann das Schwingungssignal durch Anpressdruck und Neigung des Stifts modifiziert werden, sodass auch diese Informationen übermittelt werden können. Diese Technik der elektromagnetischen Resonanz hat den weiteren Vorteil, dass der Stift auch dann lokalisiert werden kann, wenn er sich während einer *pen-up* Phase in geringer Entfernung über der Tablettoberfläche befindet. Ein Nachteil ist sicherlich die erzwungene Verwendung spezieller Stifte.

Bei einem alternativen Verfahren, das keine speziellen Stifte voraussetzt, besteht die Tablettoberfläche aus einer Anordnung einer elektrisch leitenden Schicht und einer Widerstandsschicht. An die Widerstandsschicht, die sich in geringem Abstand über der leitenden Schicht befindet, wird ein elektrisches Potential angelegt. Wird nun mit dem Stift Druck auf das Tablett ausgeübt, so führt dies zu einer punktuellen Berührung der beiden Schichten, sodass an der leitenden Schicht eine Spannung abgegriffen werden kann, anhand derer die Position der Stiftspitze hervorgeht. Diese Methode kommt zwar ohne speziellen Stift aus, dafür ist es allerdings auch nicht möglich, die Stiftposition in pen-up Phasen zu ermitteln.

Ein wahrer Technologieschub wurde seit einigen Jahren im Bereich der online Systeme durch die Integration von Eingabe- und Ausgabeeinheit ausgelöst. Bei diesen Geräten kann direkt auf das Display, z.B. einer Flüssigkristallanzeige (LCD), geschrieben werden, sodass die Schriftspur als *elektronische Tinte* oder direkt das Erkennungsergebnis im Display dargestellt wird. Dieses Konzept der elektronischen Tinte ist die Basis für stiftbasierte Computer, die als *Personal Digital Assistants* (PDAs) ei-

ne wachsende Popularität aufweisen und vielleicht eine neue Ära des *tablet computing* einläuten.

3.2.2 Scanner

Im Bereich der offline Handschrifterkennung werden überwiegend Scanner für die Signalaufnahme verwendet, um ein digitales Abbild der Vorlage zu erstellen. Die Einsatzgebiete sind vielfältig und reichen vom Lesen von Postanschriften zur Briefsortierung bis hin zur Digitalisierung historischer Dokumente. Die Scanner weisen daher je nach Anwendungsgebiet einige unterschiedliche Merkmale auf, ihre prinzipielle Arbeitsweise ist jedoch identisch.

Der Scanvorgang basiert auf der zeilenweisen Abtastung der Vorlage. Dazu wird die Vorlage beleuchtet und das reflektierte Licht über Spiegel- und Linsensysteme auf eine Zeile von CCD-Elementen gelenkt. Die CCD-Elemente (Charge Coupled Devices) sind lichtempfindliche Halbleiterbauelemente, die in Abhängigkeit von der Lichtintensität unterschiedlich hohe elektrische Spannungen liefern. Je höher also die gemessene elektrische Spannung ist, desto mehr Licht wurde reflektiert und desto heller ist der betreffende Bereich der Vorlage.

Zu den wichtigen Beurteilungskriterien für Scanner gehören vor allem Auflösung und Farbtiefe. Die Auflösung wird in horizontaler Richtung bestimmt durch die Anzahl der photosensitiven Elemente auf der Abtasteinheit, während die Schrittweite, mit der die CCD-Zeile die Vorlage abtastet, die vertikale Auflösung festlegt. Die Farbtiefe wird durch die Anzahl der Quantisierungsstufen bestimmt, die für die Abbildung der gemessenen Intensitätswerte auf einen diskreten Wertebereich zur Verfügung stehen.

Moderne Geräte erreichen physikalische, nicht interpolierte Auflösungen von mehr als 1000 dpi[2] und verwenden oftmals Farbtiefen von 12 Bit, so das pro Farbkanal 2^{12} Quantisierungsstufen genutzt werden können. Würde man also ein Dokument in Din A4 Größe mit 1000 dpi Auflösung und 12 Bit Farbtiefe scannen, so bedeutete dies einen Speicheraufwand von 132 MByte pro Farbkanal. Um diesen selbst für heutige Rechner hohen Speicheraufwand zu vermeiden, wird meist mit niedrigeren Auflösungen und Farbtiefen gearbeitet. Üblicherweise wird für die Handschrifterkennung eine Auflösung von 300 dpi und eine Farbtiefe von 8 Bit verwendet, sodass für ein Grauwertbild einer Din A4 Seite ein Speicheraufwand von ca. 8 MByte erforderlich ist.

3.2.3 Videokameras

Eine weitere Möglichkeit für die Signalaufnahme besteht in der Verwendung von Videokameras. Damit kann eine besonders natürliche Form der Eingabeschnittstelle realisiert werden, denn es sind weder spezielle Stifte notwendig, noch muss auf eine sensitive Tablettoberfläche geschrieben werden. Dass es dennoch bisher erst wenige Systeme gibt, die Videokameras als Eingabeschnittstelle für die Handschrifterkennung

[2] dpi = dots per inch, Punkte pro Zoll

verwenden, hat mehrere Ursachen, wobei insbesondere die geringere Signalqualität bzgl. Auflösung und Abtastrate im Vergleich zu Scannern und Digitalisiertabletts und die Notwendigkeit spezieller Vorverarbeitungsschritte vor allem im Bereich der online Erkennung hervorzuheben sind. Eine detaillierte Vorstellung der Vor- und Nachteile der videobasierten Sensorik bei der Handschrifterkennung findet sich im vierten Kapitel. An dieser Stelle sollen hingegen nur die Funktionsweise und einige technische Eigenschaften von Videokameras beschrieben werden.

In den meisten handelsüblichen Videokameras werden CCD-Sensoren für die Bildaufnahme eingesetzt. Bei der US-Videonorm RS 170 (NTSC) besteht dieser CCD-Chip aus einer Matrix von 768×494 Photoelementen, der europäischen Norm CCIR (PAL) entsprechen dagegen Chips mit 756×582 Elementen. Die Abmessungen der Sensorelemente variieren dabei je nach Größe des CCD-Chips zwischen $6.5 \times 6\mu m$ und $11 \times 13\mu m$. Darüberhinaus sind auch Kameras mit weitaus höheren Auflösungen mit bis zu 2048×2048 und mehr Bildpunkten verfügbar.

Neben der Auflösung ist die maximal erzielbare Bildrate ein weiteres wesentliches Merkmal einer Videokamera. Die Bildrate gibt an, wieviel Bilder die Kamera pro Sekunde aufnehmen kann. Der amerikanischen Videonorm entsprechen hierbei 30 Bilder pro Sekunde, der europäischen Norm 25 Bilder pro Sekunde. Diese Bildraten sind einerseits ausreichend, um beim menschlichen Betrachter den Eindruck einer kontinuierlichen Bewegung hervorzurufen, andererseits ist jedoch auch ein Flimmern wahrzunehmen. Um dieses Flimmern zu reduzieren, sehen die Videonormen die Interlace-Technik vor. Dabei wird jedes Bild in zwei Halbbilder zerlegt, wobei das eine Halbbild aus den geradzahligen Zeilen, das andere Halbbild dementsprechend aus den ungeradzahligen Zeilen besteht. Somit können Frequenzen von 60 (NTSC) bzw. 50 (PAL) Halbbildern pro Sekunde erreicht werden.

3.3 Vorverarbeitung

An die Aufnahme schließt sich als zweiter Schritt die Vorverarbeitung des Signals an. Die Eingabemuster werden in dieser Phase mittels einer Reihe geeigneter Transformationen qualitativ verbessert und in eine Repräsentation überführt, die für die Weiterverarbeitung vorteilhaft ist und somit bestmögliche Erkennungsergebnisse erlaubt.

Gängige Vorverarbeitungsschritte, die sowohl in den online als auch in den offline Systemen eingesetzt werden, sind Verfahren zur Rauschunterdrückung und Schriftnormalisierung. Während bei der Rauschunterdrückung vor allem aufnahmebedingte Störungen in den Daten eliminiert werden sollen, ist das Ziel der Schriftnormalisierung die Kompensation der schreiber- und situationsspezifischen Variabilität der Handschrift. Damit soll vor allem die Klassifikationsaufgabe erleichtert werden, da durch die Schriftnormalisierung die Varianz der Muster abnimmt, die derselben Klasse angehören.

Da die Verfahren zur Vorverarbeitung allerdings auch fehlschlagen können und in diesen Fällen meistens keine korrekte Erkennung mehr möglich ist, wird in einigen

Systemen weitgehend auf die Vorverarbeitung des Signals verzichtet. Damit muss jedoch entweder ein erhöhter Trainingsaufwand in Kauf genommen werden, oder es müssen äußerst robuste, bezüglich der Schriftvariabilität invariante Merkmale für die Erkennung verwendet werden.

3.3.1 Offline Systeme

Im Gegensatz zu den auf Digitalisiertabletts basierenden online Systemen, bei denen das Handschriftsignal direkt vom Sensor abgegriffen werden kann, liefern die Sensoren im offline Bereich ein Bild des gescannten Dokuments. Dabei handelt es sich üblicherweise um ein Grauwertbild, sodass zu Beginn der Verarbeitung die Schrift mit Hilfe von Binarisierungsverfahren vom Hintergrund separiert werden muss. Oftmals findet daran anschließend auch eine Skelettierung des Bildes statt, sodass die Strichstärke auf die Breite von einem Pixel ausgedünnt wird.

Etwaige Störpixel, die durch den Scanprozess oder die Binarisierung verursacht wurden, werden durch Schritte der Rauschunterdrückung eliminiert. Werden Formulare, wie z.B. Bankschecks, verarbeitet, so betrifft dies auch die Trennung vorgedruckter Muster vom Schriftsignal.

Bei der Verarbeitung komplexer Dokumente, die mehrere Textbereiche oder sowohl Text- als auch Grafikbereiche in beliebiger Anordnung enthalten, müssen vor der Weiterverarbeitung erst die für die Schrifterkennung relevanten Textbereiche im Bild lokalisiert werden. Da dieser Schritt aber in den eigenständigen Forschungsbereich der Dokumentenanalyse fällt, wird in dieser Arbeit nicht weiter darauf eingegangen. Stattdessen wird das Augenmerk auf Systeme gelegt, die ausschließlich einzelne Wörter oder Zeilen verarbeiten und daher keine komplexe Analyse der Dokumentenstruktur erfordern.

Anhand der extrahierten Wörter bzw. Textzeilen wird bei der Mehrzahl der Systeme neben der Rauschunterdrückung, Binarisierung bzw. Skelettierung außerdem eine Reihe von Normalisierungsschritten durchgeführt, um die schreiber- und situationsbedingte Variabilität der Schrift zu kompensieren. Üblicherweise wird dabei die Schrift horizontal ausgerichtet und die Schriftgröße sowie Schriftneigung korrigiert.

Binarisierung

Die Binarisierung von Grauwertbildern, d.h. die Trennung des Vordergrunds (Schrift) vom Hintergrund, ist ein elementarer Verarbeitungsschritt bei der offline Handschrifterkennung. Eine einfache und weit verbreitete Methode zur Binarisierung ist die Verwendung von Intensitätsschwellwerten. Geht man davon aus, dass die Schrift dunkel gegenüber dem Hintergrund ist, so klassifizieren die Schwellwertverfahren ein Pixel als Vordergrundpixel, falls seine Intensität kleiner als der Schwellwert ist. Andernfalls wird der Pixel dem Hintergrund zugeschlagen.

Bei den Schwellwertverfahren werden grundsätzlich zwei Strategien unterschieden: Globale und lokale Methoden. Die globalen Verfahren sind dadurch ausgezeichnet,

dass zur Binarisierung des Bildes nur ein einziger (globaler) Schwellwert herangezogen wird. Demgegenüber stehen die lokalen Verfahren, die für jeden Pixel einen Schwellwert anhand lokaler Bildinformation in der Nachbarschaft des betrachteten Pixels berechnen.

Der Ausgangspunkt für die Berechnung eines globalen Schwellwerts ist häufig das Intensitätshistogramm des Bildes. Dieses Histogramm beschreibt die Grauwertverteilung des Bildes, indem für jeden Grauwert i die relative Häufigkeit P_i der Pixel, die den entsprechenden Grauwert aufweisen, aufgetragen wird. Enthält das Bild Schriftpixel, so ergibt sich üblicherweise ein bimodales Histogramm. Um einen Schwellwert zu finden, der die beiden Moden des Histogramms optimal separiert, sind mehrere Verfahren vorgeschlagen worden (siehe [Har92]), darunter z.B. auch die sogenannte *Otsu-Methode* [Ots79].

Bei der Otsu-Methode wird der optimale Schwellwert so gewählt, dass die Summe der gewichteten Varianzen der beiden Histogrammgruppen minimiert wird. Die erste Gruppe ist dabei durch den Bereich des Histogramms mit $i \leq t$ bestimmt, für die zweite Gruppe gilt dementsprechend $i > t$. Das jeweilige Gruppengewicht wird aus der Summe der relativen Häufigkeiten P_i der entsprechenden Gruppe gebildet:

$$q_1(t) = \sum_{i=1}^{t} P_i \quad , \quad q_2(t) = \sum_{i=t+1}^{N} P_i \quad . \tag{3.1}$$

Damit gilt für die Mittelwerte der beiden Gruppen

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP_i}{q_1(t)} \quad , \quad \mu_2(t) = \sum_{i=t+1}^{N} \frac{iP_i}{q_2(t)} \quad , \tag{3.2}$$

und dementsprechend für die Varianzen

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P_i}{q_1(t)} \quad , \quad \sigma_2^2(t) = \sum_{i=t+1}^{N} [i - \mu_2(t)]^2 \frac{P_i}{q_2(t)} \quad . \tag{3.3}$$

Nach der Definition von Otsu ergibt sich der optimale Schwellwert $\hat{t}$ zur Trennung der beiden Histogrammgruppen aus dem Minimum der Summe der gewichteten Gruppenvarianzen $\sigma_w^2(t)$ (Intragruppenvarianz). Für $\hat{t}$ gilt somit:

$$\hat{t} = \underset{t}{\operatorname{argmin}} \left\{ \sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \right\} . \tag{3.4}$$

Die Intragruppenvarianz $\sigma_w^2(t)$ ist mit der Gesamtvarianz σ^2 und der Intergruppenvarianz $\sigma_b^2(t)$ folgendermaßen verknüpft:

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t) \quad . \tag{3.5}$$

Hierbei ist

$$\sigma^2 = \sum_{i=1}^{N} (i - \mu)^2 P_i \qquad \text{mit} \quad \mu = \sum_{i=1}^{N} iP_i$$

und

$$\sigma_b^2(t) = q_1(t)(1 - q_1(t))(\mu_1(t) - \mu_2(t))^2 \quad .$$

Da die Gesamtvarianz für das Bild konstant ist, kann die Bestimmung des optimalen Schwellwerts $\hat{t}$ alternativ zu Gleichung 3.4 auch durch die Maximierung der Intergruppenvarianz $\sigma_b^2(t)$ vorgenommen werden:

$$\hat{t} = \underset{t}{\operatorname{argmax}} \left\{ \sigma_b^2(t) \right\} \quad . \tag{3.6}$$

Ein weiteres Verfahren, dass häufig zur Binarisierung eingesetzt wird, ist die Methode von Kittler und Illingworth (siehe z.B. [Har92]). Dabei wird angenommen, dass sich die beobachtete Grauwertverteilung durch die Mischung zweier Gaußverteilungen beschreiben lässt. Die Mischverteilung hat die Gestalt

$$f_i(t) = \frac{q_1(t)}{\sqrt{2\pi}\sigma_1(t)} e^{-\frac{1}{2}\left(\frac{i-\mu_1(t)}{\sigma_1(t)}\right)^2} + \frac{q_2(t)}{\sqrt{2\pi}\sigma_2(t)} e^{-\frac{1}{2}\left(\frac{i-\mu_2(t)}{\sigma_2(t)}\right)^2}, \tag{3.7}$$

mit den Gruppengewichten q_1, q_2, Mittelwerten μ_1, μ_2 und Varianzen σ_1, σ_2 aus den Gleichungen 3.1-3.3. Die Aufgabe besteht nun darin, die Parameter der Mischverteilung durch Variation des Schwellwerts t so zu bestimmen, dass die Kullback-Leibler Distanz

$$J(P; f(t)) = \sum_{i=1}^{N} P_i \log \left[\frac{P_i}{f_i(t)} \right] \tag{3.8}$$

minimiert wird. Die Kullback-Leibler Distanz stammt aus der Informationstheorie und misst die Verschiedenheit zweier Verteilungen über der gleichen Zufallsvariable. Es gelten:

1. $J(P; f) \geq 0$ für alle Wahrscheinlichkeitsverteilungen P und f
2. $J(P; f) = 0$ genau dann, wenn $P = f$

Derjenige Wert von t, für den die Kullback-Leibler Distanz von P und $f(t)$ minimal ist, wird dann als optimaler Schwellwert angenommen.

Die globalen Verfahren sind anwendbar, wenn für das gesamte Bild die Trennung des Vordergrunds vom Hintergrund mit Hilfe eines einzigen Schwellwerts durchgeführt werden kann. Dies ist jedoch nicht immer gegeben. So kann es beispielsweise durch Verschmutzung oder Alterungsprozesse des Papiers dazu kommen, dass in manchen Bildbereichen die Hintergrundpixel eine niedrigere Intensität aufweisen als die Vordergrundpixel. Um dennoch eine Binarisierung des Bildes durchzuführen, muss für jeden Bildpunkt ein lokaler Schwellwert bestimmt werden.

Eine Methode zur adaptiven Schwellwertbestimmung wurde von Niblack vorgeschlagen [Nib86]. Bei diesem Verfahren wird der Schwellwert lokal in einem Fenster, das um den aktuell betrachteten Pixel zentriert ist, bestimmt. Der Schwellwert wird dabei anhand des Mittelwerts μ und der Standardabweichung σ der Intensitäten in dem Fenster wie folgt berechnet:

$$t(x, y) = \mu(x, y) + k\sigma(x, y) \tag{3.9}$$

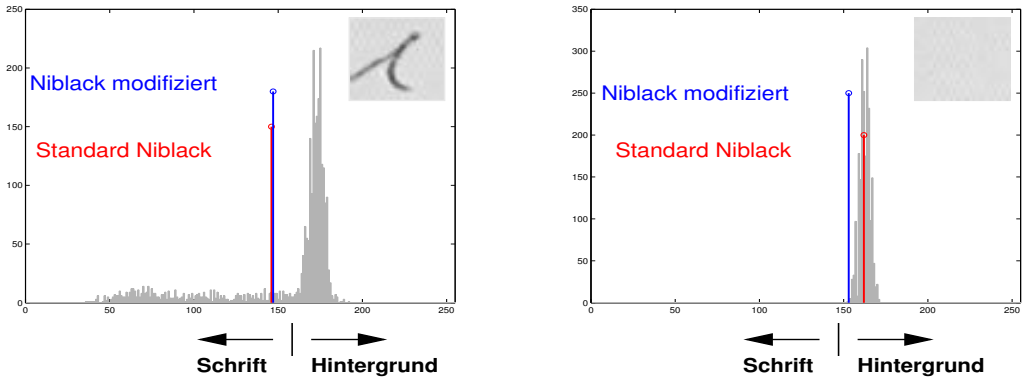


Abbildung 3.3: Intensitätshistogramme und Schwellwerte der lokalen Binarisierungsmethoden. Links ist das Histogramm einer Bildregion dargestellt, die Schriftpixel enthält. Rechts ist das Histogramm eines Hintergrundbereichs dargestellt.

In obiger Formel bezeichnet k einen vom jeweiligen Anwendungsfall abhängigen Parameter, der sich im wesentlichen danach richtet, ob der Anteil der Vordergrund- oder der Hintergrundpixel im Bild dominiert. Bei der Verarbeitung von Schriftdokumenten, bei denen die Schrift dunkel gegenüber dem Hintergrund ist und deutlich weniger Schriftpixel als Hintergrundpixel im Bild vorhanden sind, ist k als kleine negative Zahl zu wählen. Damit wird der Schwellwert also ausgehend vom Mittelwert der Intensitätsverteilung in Abhängigkeit von der Standardabweichung in Richtung niedrigerer Intensitätswerte verschoben.

Bei der oben vorgestellten Methode taucht eine Schwierigkeit auf, wenn in dem betrachteten Bildfenster keine Schriftpixel vorhanden sind. Dann ist die Standardabweichung der Intensitätsverteilung sehr klein, und der Schwellwert liegt nahe am Mittelwert, also im Bereich der Hintergrundintensitäten (siehe Abbildung 3.3). Dies führt zu Rauschen in den Bildbereichen, die keine Vordergrundpixel aufweisen.

In [Zha01b, Zha01a] wird eine Variante der oben vorgestellten Methode zur Binarisierung vorgeschlagen, die besser auf die Charakteristika von Schriftdokumenten abgestimmt ist . Dabei ist die Verschiebung des Schwellwerts ausgehend vom Mittelwert der Verteilung antiproportional zur Standardabweichung. Darüberhinaus wird die Standardabweichung normiert, indem durch den Dynamikbereich der Standardabweichungen dividiert wird. Die so modifizierte Formel zur lokalen Schwellwertbestimmung ist in Gleichung 3.10 dargestellt.

$$t(x,y) = \mu(x,y) + k\left(\mu(x,y)\left(1 - \frac{\sigma(x,y)}{R}\right)\right) \tag{3.10}$$

Der Parameter R bezeichnet dabei den Dynamikbereich der Standardabweichungen σ.

Anhand von Abbildung 3.3 erkennt man, dass beide Varianten angewendet auf einen Bildausschnitt, der Schriftpixel enthält, einen ähnlichen Schwellwert finden. Enthält der Bildausschnitt jedoch ausschließlich Hintergrundpixel (rechter Teil der Abbil-

Abbildung 3.4: Ausgangsbild (a), globale Binarisierung mit der Otsu-Methode (b), lokale Binarisierung mit der Niblack Methode (c) und lokale modifizierte Niblack-Binarisierung (d).

dung 3.3), so liefert die herkömmliche Methode einen Schwellwert, der zu weit im Bereich der Hintergrundintensitäten liegt. Im Gegensatz dazu findet das modifizierte Verfahren einen Schwellwert, der weitaus mehr Pixel korrekt dem Hintergrund zuweist.

Die resultierenden Binärbilder der lokalen Schwellwertverfahren sind in Abbildung 3.4 dargestellt. Das Ausgangsbild weist eine große Variation der Intensitäten auf, sodass kein globaler Schwellwert zur Binarisierung gefunden werden kann. Die Abbildung zeigt, dass die modifizierte Niblack-Methode deutlich weniger Rauschen im Binärbild hervorruft als die herkömmliche Variante.

Skelettierung

An die Binarisierung der Grauwertbilder schließt sich oftmals die Skelettierung an, um die Strichstärke der Schrift auf ein Pixel breite Linien auszudünnen. Außer zur Kompensation der Schriftvariabilität in Bezug auf die Strichstärke ist die Skelettierung insbesondere dann sinnvoll, wenn die Schrifterkennung auf strukturellen Merkmalen basiert. Strukturelle Merkmale, wie z.B. Kreuzungspunkte, Endpunkte oder Schleifen können dann leicht anhand des skelettierten Bildes ermittelt werden.

Für die Skelettierung gibt es keine mathematische Definition des Resultatsbildes. Stattdessen werden üblicherweise die folgenden qualitativen Anforderungen an die Skelettierung gestellt (siehe Abbildung 3.5) [Kle92]:

1. Das Skelett muss aus Linien der Breite von einem Bildpunkt bestehen.
2. Die topologischen Zusammenhänge müssen denjenigen des Ausgangsbildes entsprechen.
3. Die Skelettlinien müssen etwa in der Mitte der Objekte verlaufen.
4. Die Anzahl irrelevanter Skelettäste muss möglichst klein sein.
5. Das Ergebnis der Skelettierung muss stabil sein, d.h. unter der wiederholten Anwendung des Verfahrens sollte das skelettierte Bild unverändert bleiben.

Zur Skelettierung von Binärbildern – ähnlich wie die Binarisierung eine klassische Bildverarbeitungsoperation – ist eine Vielzahl von Verfahren entwickelt worden (siehe u.a. [Pav90, Gon91, Jäh97]). Eine verbreitete Vorgehensweise ist dabei die iterierte, topologieerhaltende Erosion der Bildobjekte bis auf ein Pixel Breite. Dies kann z.B. durch Einsatz morphologischer Operatoren erreicht werden. Das Ziel der morphologischen Operatoren ist die Detektion bestimmter Formen in Binärbildern anhand von *Hit-Miss-Masken*. Dabei wird je nach Aufgabenstellung eine Hit-Miss-Maske definiert und auf jeden Bildpunkt des Originalbildes angewendet. Ein Pixel wird im Resultatsbild nun genau dann gesetzt, wenn alle Werte der Maske mit denjenigen des Originalbildes übereinstimmen.

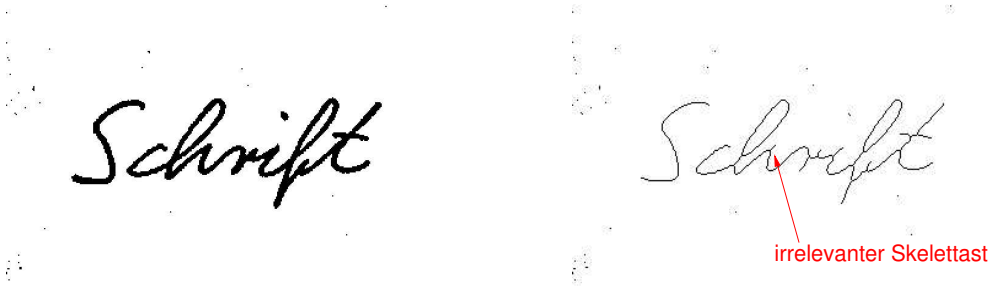
Ein einfaches Beispiel ist die Verwendung eines morphologischen Operators zur Detektion isolierter Pixel. Nimmt man für den Bildhintergrund den Wert 0 und für den Vordergrund den Wert 1 an, so ist die zugehörige Hit-Miss-Maske wie folgt definiert [Jäh97]:

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.11}$$

Um einen morphologischen Skelettierungsoperator zu realisieren, wird so vorgegangen, dass im ersten Schritt ein Satz von Hit-Miss-Masken erstellt wird, die genau die Pixel detektieren, die sich am Rand des Bildobjekts befinden und deren Entfernung nicht die Topologie des Bildobjekts zerstört[3].

$$M_1 = \begin{bmatrix} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & x & 1 \\ 0 & 1 & 1 \\ 0 & x & 1 \end{bmatrix},$$
$$M_3 = \begin{bmatrix} 1 & 1 & 1 \\ x & 1 & x \\ 0 & 0 & 0 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1 & x & 0 \\ 1 & 1 & 0 \\ 1 & x & 0 \end{bmatrix}. \tag{3.12}$$

[3] In den Hit-Miss-Masken können die Felder, die mit x markiert sind, sowohl den Vordergrund als auch den Hintergrund bezeichnen.



(a) (b)

Abbildung 3.5: Binärbild (a) und skelettiertes Bild (b).

Im zweiten Schritt werden dann die so ermittelten Pixel vom Bildobjekt entfernt, sodass das resultierende Bildobjekt anschließend auf jeder Seite um einen Pixel ausgedünnt ist. Um Objekte zu erhalten, die nur noch einen Pixel breit sind, muss diese Prozedur daher solange iteriert werden, bis keinerlei Veränderung im Resultatsbild auftritt [Jäh97].

Neben der klassischen Vorgehensweise, die auf der oben vorgestellten iterierten Erosion der Bildobjekte basiert, sind auch perzeptionsorientierte Verfahren entwickelt worden, die besonders zur Verdünnung von Handschrift geeignet sind und sich daran orientieren, wie der Mensch Schrift skelettiert [Bar88, Cho92]. Das perzeptionsorientierte Vorgehen besteht darin, dass die Schriftlinien Stroke für Stroke verfolgt und skelettiert werden, sodass das Schriftskelett im Gegensatz zu den iterativen Verfahren in einem Durchgang ermittelt werden kann.

Das *Skelettierung durch Linienverfolgung* Verfahren basiert auf zwei Zeigern P_L, P_R, die entlang der linken bzw. rechten Kante des Schriftzuges verschoben werden und einem Fenster flexibler Größe, das die beiden durch die Zeiger referenzierten Punkte umfasst. Die Verfolgung einer Linie wird erreicht, indem ausgehend von einer initialen Position die Zeiger schrittweise den Kanten des Schriftzuges folgen. Endpunkte oder Kreuzungspunkte der Linie können anhand des durch P_L und P_R aufgespannten Fensters detektiert werden. Wird ein Kreuzungspunkt festgestellt, so wird das Verfahren rekursiv auf die entsprechenden Verzweigungen angewandt. Die resultierende Skelettlinie ist definiert durch die Verbindungslinie der Mittelpunkte aufeinanderfolgender Fenster. Abbildung 3.6 veranschaulicht das vorgestellte Verfahren.

Konturverfolgung

In einigen Systemen wird vor der Weiterverarbeitung von der bildhaften Repräsentation abgesehen und stattdessen eine kompaktere Darstellung in Form von Kettencodes gewählt, die durch Konturverfolgungsverfahren ermittelt werden können [Kim97, Mad99b]. Die Schriftkontur ist dabei definiert als die Menge aller Schriftpixel, die mindestens einen Nachbarn im Hintergrund haben.

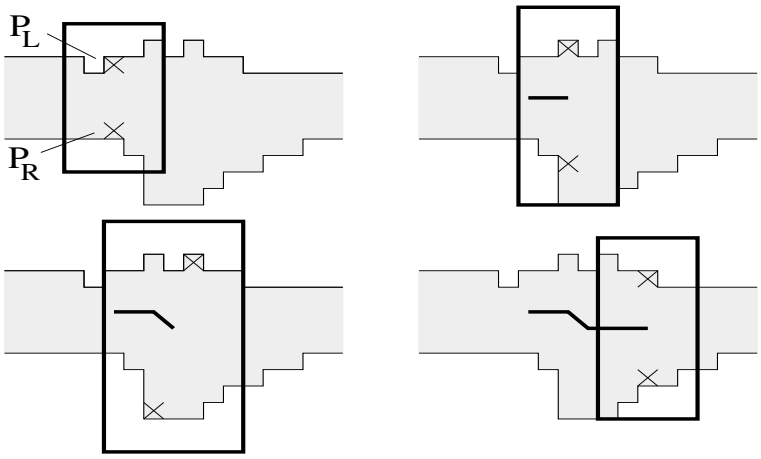


Abbildung 3.6: Skelettierung durch Linienverfolgung [Bar88]

Der Ausgangspunkt der Verfahren zur Konturverfolgung ist das binarisierte Schriftbild. Der erste Schritt besteht nun darin, einen initialen Konturpunkt zu finden, der als Anfangspunkt für die Konturverfolgung dient. Anschließend wird von diesem Anfangspunkt ausgehend die Schriftkontur in einer definierten Richtung (Uhrzeigersinn oder Gegenuhrzeigersinn) verfolgt, bis man wieder am Anfangspunkt angelangt ist. Man erhält somit eine Sequenz von Einzelschritten, wobei jeder Schritt in eine von acht Richtungen geht[4]. Werden die Richtungen durch die Zahlen $0, 1, \ldots, 7$ beschrieben, so kann die Kontur der Schrift durch eine Zahlenfolge, dem sogenannten *Kettencode*, dargestellt werden:

$$r_1, r_2, \ldots, r_N \qquad \text{mit } 0 \leq r_i \leq 7 \quad .$$

Werden zusätzlich die absoluten Koordinaten beispielsweise des Anfangspunktes angegeben, so ist neben der Kontur der Schrift auch ihre Lage eindeutig spezifiziert [Kle92].

Rauschunterdrückung

Durch die Bildaufnahme, Binarisierung und Normalisierung der Schrift kann es zum Auftreten von Störpixeln in den Bildern kommen. Um diese Störungen in den Bildern zu eliminieren, werden Verfahren zur Rauschunterdrückung eingesetzt. Eine verbreitete Methode zur Glättung des Bildes ist beispielsweise die Faltung mit einer Gaußfunktion [Sen92]:

$$g(x, y) = e^{-a^2x^2 - b^2y^2} \quad .$$

Die Gaußfunktion hat die gewünschte Eigenschaft, dass ihre Fouriertransformierte wiederum eine Gaußfunktion ist, und somit die Amplitudendämpfung des Signals

[4] Hier wird eine Achter-Nachbarschaft der Pixel angenommen. Pixel können damit horizontal, vertikal oder diagonal benachbart sein. Im Gegensatz dazu gelten die Pixel bei einer Vierer-Nachbarschaft als benachbart, wenn die Pixel horizontal oder vertikal aneinander angrenzen.

mit der Frequenz monoton zunimmt. Im diskreten lässt sich die Gaußfunktion durch die Binomialverteilung approximieren, die sich im eindimensionalen Fall anhand des Pascal'schen Dreiecks darstellen lässt. Aufgrund der Separierbarkeitseigenschaft der Gaußfunktion ergibt sich die zweidimensionale Faltungsmaske aus der Multiplikation der eindimensionalen Binomialmasken in x bzw. y-Richtung. Die 3×3 Gauß- bzw. Binomialmaske ergibt sich damit zu:

$$\frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \cdot \frac{1}{4}\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad .$$

Lineare Filter, wie z.B. der Gaußfilter, sind zur Rauschunterdrückung geeignet, wenn davon ausgegangen werden kann, dass die Bildpunkte zwar fehlerhafte, aber dennoch brauchbare Informationen tragen. Weist das Bild jedoch binäres Rauschen auf, bei dem die Grauwerte einzelner Pixel völlig verändert sind, so ist der Medianfilter besser zur Unterdrückung der "Ausreißerpixel" geeignet.

Mit dem Medianfilter, der zur Klasse der Rangordnungsfilter gehört, wird eine Sortierung der Grauwerte innerhalb der Maske vorgenommen und anschließend der mittlere Wert (Median) selektiert. Der Zentralpixel der Maske wird dann durch den Median ersetzt. Bei einem Medianfilter mit der Maskengröße 3×3 können somit einzelne Ausreißerpixel wirkungsvoll eliminiert werden, während konstante Nachbarschaften und Kanten unbeeinflusst bleiben. In Abbildung 3.7 ist die Medianfilterung einer durch einen "Ausreißer" gestörten Bildkante dargestellt.

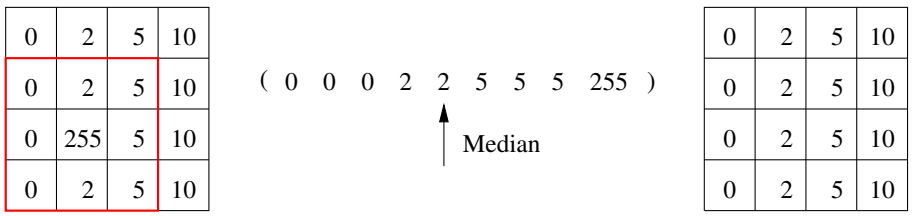


Abbildung 3.7: Anwendung des 3×3 Medianfilters auf einen Bildausschnitt. Links ist die Ausgangsmaske dargestellt, in der Mitte die sortierte Liste der Grauwerte, rechts das Ergebnis.

Bei der Normalisierung der Schrift anhand binarisierter Bilder tritt außerdem oftmals ein "Zerfasern" der Schriftkontur auf. Um diese Artefakte zu unterdrücken und eine glatte Kontur wiederherzustellen, werden beispielsweise in [Che94] morphologische Operatoren eingesetzt. Diese Operatoren basieren ähnlich wie die schon angesprochene Skelettierung auf Hit-Miss-Masken. Weisen die Schriftpixel des Ausgangsbilds I den Wert Eins auf, die Hintergrundpixel den Wert Null, so kann mit der Hit-Miss-Maske M, mit

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad ,$$

durch die folgende Transformation eine *Erosion* der Schriftkontur durchgeführt werden [Jäh97]:

$$I'_{x,y} = I \ominus M = \bigwedge_{k=-1}^{1} \bigwedge_{l=-1}^{1} I_{x+k,y+l} \wedge M_{k,l} \quad .$$

Bei der Erosion wird ein Pixel nur gesetzt, wenn im Ausgangsbild der Zentralpixel und alle benachbarten Pixel gesetzt sind. Die Schriftkontur wird somit um die Randpixel verdünnt, zudem verschwinden Bereiche, die kleiner als die Maske sind, völlig.

Die zur Erosion duale Operation ist die *Dilatation*. Bei der Dilatation wird die Schrift ausgedehnt und kleine Löcher oder Sprünge werden gefüllt:

$$I'_{x,y} = I \oplus M = \bigvee_{k=-1}^{1} \bigvee_{l=-1}^{1} I_{x+k,y+l} \wedge M_{k,l} \quad .$$

Um einerseits Konturen zu glätten, andererseits aber die Strichdicke zu erhalten, wird oftmals eine Kombination aus Erosion und Dilatation eingesetzt. Diese Operatoren sich nicht kommutativ. Wird erst die Erosion und dann die Dilatation ausgeführt, so spricht man von einem *Opening*. Bei einem *Closing* werden die Operationen dagegen in der umgekehrten Reihenfolge ausgeführt:

Opening: $(I \ominus M) \oplus M$

Closing: $(I \oplus M) \ominus M$

In den Arbeiten [Kim97, Kim99] wird die Glättung der Schriftkontur anhand des Kettencodes vorgenommen. Dazu wird über den Kettencode, also der Sequenz der Konturrichtungen, ein Fenster geschoben, das jeweils drei aufeinanderfolgende Kettencode-Einträge umfasst. Auf Basis dieser drei Richtungswerte wird dann mit Hilfe einer Reihe von Heuristiken eine Glättung durchgeführt, indem einzelne Komponenten weggelassen oder modifiziert werden.

Ermittlung von Referenzlinien

Die Schriftnormalisierung, d.h. insbesondere die Korrektur von Schriftorientierung und -größe basiert bei einer Vielzahl von Systemen zur Handschrifterkennung auf der Ermittlung von Referenzlinien der Schrift. Diese Linien leiten sich aus dem für die Lateinschrift charakteristischen *Vier-Linien-Schema* ab, das die Schrift in die drei Zonen der Oberlängen, Mittellängen und Unterlängen einteilt (Abbildung 3.8). Außer für die Schriftnormalisierung ist die Referenzlinienschätzung häufig auch für die spätere Merkmalsextraktion erforderlich. So können Raumrelationen der Merkmale zu den Referenzlinien ermittelt werden, wodurch die Komplexität der Klassifikationsaufgabe deutlich verringert wird [Cai00].

Die Verfahren, die zur Berechnung der Referenzlinien eingesetzt werden, lassen sich in zwei Gruppen einteilen: Histogrammbasierte Methoden einerseits und Verfahren, die auf der Approximation von Geraden an die Schriftkontur beruhen, andererseits.

Abbildung 3.8: Vier-Linien-Schema der Schrift.

Die Idee der histogrammbasierten Methoden ist die Ausnutzung der charakteristischen Form der Dichtefunktion der y-Koordinaten der Schrifttrajektorie. Diese Dichte weist bei horizontal ausgerichteter Schrift zwischen der Schriftgrundlinie und der Mittellinie ein hohes Plateau auf, während die Dichte im Ober- und Unterlängenbereich im Vergleich dazu deutlich geringer ist. Die Dichtefunktion wird durch das horizontale Projektionshistogramm beschrieben, in dem für jede Zeile des Bildes die relative Häufigkeit der Schriftpixel aufgetragen ist (siehe Abbildung 3.9).

Um das Plateau im Histogramm zu lokalisieren besteht eine Möglichkeit in der Anwendung eines Schwellwerts auf die relativen Häufigkeiten des Histogramms, sodass die Zeilen zur Mittelzone zugeschlagen werden, deren Histogrammeintrag größer als der Schwellwert ist. In [Boz89] wird dieser Schwellwert mit Hilfe von Heuristiken bestimmt. Es ist jedoch auch möglich, ohne Heuristiken auszukommen, indem das varianzminimierende Verfahren, das ursprünglich zur Binarisierung von Grauwertbildern vorgeschlagen wurde (Gleichung 3.4), für die Berechnung des Schwellwerts eingesetzt wird. Dieses Verfahren bestimmt anhand einer bimodalen Verteilung einen Schwellwert, sodass die beiden Moden optimal (im Sinne minimaler Varianz) separiert werden können. In diesem Fall wird nun nicht die Grauwertverteilung, sondern die Verteilung der Anzahl der Schriftpixel pro Bildzeile zugrundegelegt [Vin00].

Eine weitere Möglichkeit zur Detektion des Plateaus ist die Verwendung der ersten Ableitung des Histogramms [Bun95, Cai00]. Die erste Ableitung, d.h. die lokale Änderung der relativen Häufigkeiten, besitzt bei einem idealisierten Histogramm am Übergang von der Ober- zur Mittelzone ein globales Maximum und am Übergang von der Mittel- zur Unterzone ein globales Minimum, sodass die Mittelzone und damit die Schriftgrundlinie und Mittelinie leicht zu identifizieren sind.

Ein Nachteil der histogrammbasierten Methoden zur Referenzlinienschätzung ist allerdings, dass sie nur bei horizontal ausgerichteter Schrift anwendbar sind. Ist dies nicht der Fall, weist das Projektionshistogramm nicht das charakteristische Profil auf, sodass die Mittelzone weder durch Anwendung von Schwellwerten auf die relativen Häufigkeiten direkt noch durch Verwendung der ersten Ableitung bestimmt werden kann. Eine weitere Schwierigkeit entsteht beim Auftreten längerer horizontaler Linien im Bereich der Ober- bzw. Unterlängen, da sie im Projektionshistogramm hohe Ausschläge verursachen, welches somit deutlich von der Idealform abweicht.

Abbildung 3.9: Horizontales Projektionshistogramm der Schrift.

Ein alternatives Verfahren zur Referenzlinienschätzung ist die Anpassung von Geraden an die Schriftkontur. Diese Methode basiert auf der Beobachtung, dass die lokalen vertikalen Extrempunkte der Schriftkontur in der Regel auf Höhe der Referenzlinien der Schrift liegen. Betrachtet man beispielsweise nur Kleinbuchstaben ohne Ober- und Unterlängen, so ergibt sich die Schriftgrundlinie aus den lokalen Minima und die Mittellinie aus den lokalen Maxima der Schriftkontur. Dementsprechend liegt die obere Begrenzungslinie in Höhe der Maxima der Oberlängen und die untere Begrenzungslinie in Höhe der Minima der Unterlängen. Zur Ermittlung der Referenzlinien müssen daher zuerst die entsprechenden lokalen Extremstellen gefunden werden, sodass daraufhin die Referenzlinien als Ausgleichsgeraden durch die Extrempunkte berechnet werden können (siehe Abbildung 3.10).

Ein häufig eingesetztes Verfahren zur Berechnung von Ausgleichsgeraden ist die lineare Regression. Dabei wird die vertikale Komponente der Referenzlinie als normalverteilte Zufallsgröße Y mit dem Erwartungswert

$$E(Y) = a + bx \tag{3.13}$$

angenommen, wobei x die horizontale Komponente der Referenzlinie bezeichnet. Die Zufallsgröße Y hängt nach obiger Beziehung 3.13 also im Mittel linear von dem festen x-Wert ab [Bro01]. Die unbekannten Parameter a und b können durch Minimierung des quadratischen Fehlers anhand der Stützpunkte (x_i, y_i) bestimmt werden.

$$\sum_{i=1}^{n} [y_i - (a + bx_i)]^2 = min! \tag{3.14}$$

Für die Geradenparameter erhält man daraus die Schätzwerte:

$$\hat{b} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad , \quad \hat{a} = \bar{y} - \hat{b}\bar{x} \tag{3.15}$$

mit

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{3.16}$$

Abbildung 3.10: Approximation einer Geraden an die Schriftgrundlinie. Die Punkte stellen die verwendeten lokalen Minima der Schriftkontur dar.

Entscheidend für die Güte der Approximation ist dabei die Zuordnung der extrahierten Stützpunkte zu den zu schätzenden Referenzlinien. So sollten z.B. die Minima, die Unterlängen oder Störpixeln zuzurechnen sind, von der Berechnung der Schriftgrundlinie ausgeschlossen werden. Diese Zuordnung ist jedoch bei uneingeschränkter, nicht notwendigerweise horizontal ausgerichteter Schrift in der Regel nicht mit Hilfe einfacher Abstandsbetrachtungen durchführbar. Aus diesem Grund wird häufig ein zweistufiges Vorgehen favorisiert, bei dem im ersten Schritt eine grobe Schätzung der jeweiligen Referenzlinie vorgenommen wird, die dann im zweiten Schritt durch die Elimination von "Ausreißerpunkten" entsprechend verfeinert wird [Sen98].

In [Ari02] wird eine Variante dieses Verfahrens beschrieben, bei dem die "Ausreißer" nicht eliminiert werden, sondern mit einem geringeren Gewicht in die Minimierung des quadratischen Abstands eingehen.

$$\sum_{i=1}^{n} \frac{[y_i - (a + bx_i)]^2}{\mu_i^2} = min! \tag{3.17}$$

Das Gewicht μ_i basiert dabei auf dem mittleren Winkel, den die Verbindungsgeraden des lokalen Minimums (Maximums) (x_i, y_i) und allen anderen Minima (Maxima) $(x_j, y_j),\ \ j \neq i$ mit der Horizontalen einschließen. Diesem Ansatz liegt die Idee zugrunde, dass der mittlere Winkel an einem "Ausreißerpunkt" deutlich größer ist als der mittlere Winkel an den übrigen Punkten, sodass bei einem "Ausreißerpunkt" der entsprechende Term zu der gewichteten Summe 3.17 weniger beiträgt.

Eine weitere Möglichkeit um zu verhindern, dass sich "Ausreißerpunkte" stark auf die Berechnung der Ausgleichsgeraden auswirken, wird in [Sch97] vorgeschlagen. Dabei wird die Minimierung des quadratischen Fehlers durch die Minimierung des Absolutbetrags des Fehlers ersetzt:

$$\sum_{i=1}^{n} |y_i - (a + bx_i)| = min! \tag{3.18}$$

Üblicherweise wird bei der Referenzlinienschätzung davon ausgegangen, dass die Linien jeweils als Gerade approximiert werden können, die *global* auf dem gesamten Schriftabschnitt berechnet wird. Diese Annahme gilt jedoch nur bei kurzen Schriftabschnitten, die ein einzelnes Wort umfassen. Enthält der Schriftabschnitt dagegen mehrere Wörter, die in unterschiedlichen Orientierungen und/oder mit einem vertikalen

Versatz zueinander geschrieben wurden, so ist eine *lokale* Bestimmung der Referenzlinien erforderlich.

Ein Beispiel für ein lokales Verfahren zur Referenzlinienschätzung wird in [Mad99a] vorgestellt, das als Vorverarbeitungsschritt in einem System zur Erkennung von Straßennamen eingesetzt wird. Bei diesem Verfahren wird die Basislinie des gesamten Schriftabschnitts stückweise durch Geraden approximiert, die jeweils auf Basis der lokalen Konturminima durch linearer Regression berechnet werden. Die Entscheidung, welche Konturminima in die Berechnung der lokalen Basislinie eingehen, wird anhand des Gestaltgesetzes der "guten Fortsetzung" vorgenommen.

Korrektur der Schriftorientierung

Die Schriftorientierung ist definiert als der Winkel zwischen der Schriftgrundlinie und der Horizontalen. Da die Schriftorientierung bei unterschiedlichen Schreibern oftmals stark variieren kann, insbesonders wenn keine vorgedruckten Referenzlinien vorhanden sind, kommt der Korrektur der Schriftorientierung gerade bei Systemen zur schreiberunabhängigen Handschrifterkennung eine hohe Bedeutung zu.

Oftmals basiert die Orientierungskorrektur auf den im Vorhinein geschätzten Referenzlinien der Schrift [Bro83, Sen92, Sen98]. Zur Bestimmung der Schriftgrundlinie eignen sich hier vor allem die Verfahren, die auf der Approximation von Geraden an die lokalen vertikalen Minima der Schriftkontur beruhen. Diese Verfahren gestatten es im Gegensatz zu den histogrammbasierten Methoden, die Referenzlinien auch bei nicht horizontal orientierter Schrift in einem Schritt zu approximieren (siehe vorheriger Abschnitt). Ist der Steigungswinkel der approximierten Schriftgrundlinie schließlich ermittelt, so wird durch Rotation des Schriftbildes die Grundlinie horizontal ausgerichtet.

Die Verfahren, die die Referenzlinien im Gegensatz dazu anhand des Schrifthistogramms schätzen, basieren auf der Detektion bestimmter Eigenschaften des horizontalen Projektionshistogramms, wie z.B. einem ausgeprägten Plateau im Bereich der Schriftmittelzone, die bei horizontal ausgerichteter Schrift auftreten. Um die histogrammbasierten Methoden zur Korrektur der Schriftorientierung einsetzen zu können, ist es daher notwendig, das Schriftbild schrittweise zu rotieren und das resultierende Histogramm nach jedem Rotationsschritt auf das Auftreten des entsprechenden Histo-

Abbildung 3.11: Korrektur der Schriftorientierung durch Schätzung der Schriftgrundlinie.

grammmerkmals zu überprüfen. Derjenige Rotationswinkel, unter dem das Merkmal am stärksten ausgeprägt ist, wird dann als die Schriftorientierung angenommen.

In [Côt98] ist ein Verfahren zur Schätzung der Schriftorientierung beschrieben, das die *Entropie* als Maß für die "Kompaktheit" des horizontalen Projektionshistogramms verwendet. Die Entropie ist folgendermaßen definiert:

$$E = -\sum_i p_i \log p_i \tag{3.19}$$

Hierbei bezeichnet p_i die relative Häufigkeit des Auftretens eines Schriftpixels in der i-ten Zeile des Schriftbildes. Der Grundgedanke des Verfahrens ist, dass horizontal ausgerichtete Schrift durch ein kompaktes Plateau im Projektionshistogramm gekennzeichnet ist, das somit eine geringere Entropie aufweist als eher gleichverteilte Histogrammeinträge beliebig ausgerichteter Schrift. Um die Schriftorientierung zu bestimmen, wird nun das Schriftbild schrittweise rotiert und für jeden Rotationswinkel die Entropie anhand des Projektionshistogramms berechnet. Der Rotationswinkel, der zu minimaler Entropie führt, wird dann als Schätzwert für die Schriftorientierung angenommen (siehe Abbildung 3.12).

In [Cai00] wird ein ähnliches Verfahren vorgestellt, das ebenfalls auf dem horizontalen Projektionshistogramm des Schriftbildes beruht. Hier wird als Merkmal der Betrag der ersten Ableitung des geglätteten Histogramms verwendet. Analog zu obigem Verfahren wird angenommen, dass der Rotationswinkel, der den maximalen Betrag der ersten Ableitung hervorgerufen hat, der Orientierung der Schrift entspricht.

Korrektur der Schriftgröße

Ein weiterer Vorverarbeitungsschritt, der häufig vorgenommen wird, betrifft die Normalisierung der Schriftgröße. Dabei wird i.d.R. die Schrift sowohl in vertikaler als auch in horizontaler Richtung skaliert [Cai00, Mar00b].

Die Grundlage für die Skalierung in vertikaler Richtung bilden üblicherweise die Referenzlinien der Schrift. Anhand des Abstands zwischen der Basislinie und der Mittellinie wird die Korpushöhe der Schrift geschätzt, also die vertikale Ausdehnung von

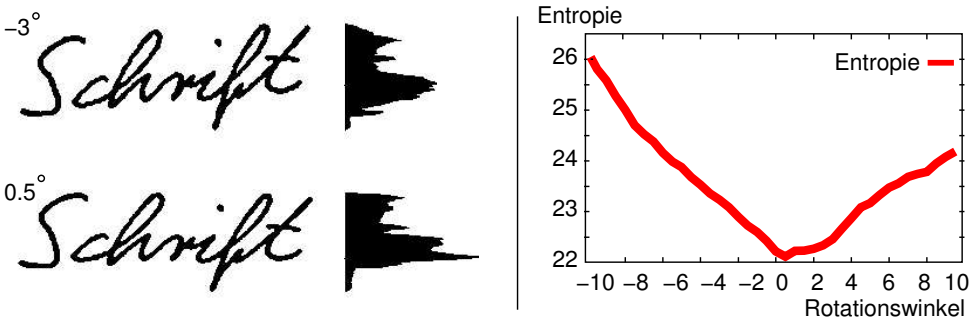


Abbildung 3.12: Korrektur der Schriftorientierung durch iterative Rotation und Berechnung der Entropie des Histogramms.

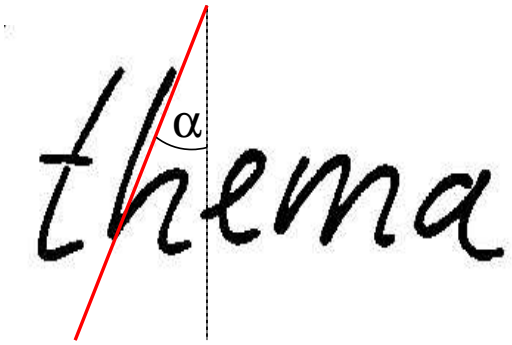


Abbildung 3.13: Neigungswinkels α der Schrift.

Buchstaben ohne Ober- bzw. Unterlänge [Cai00]. Die geschätzte Korpushöhe wird dann auf einen vorgegebenen Wert normiert, der beispielsweise aus der Mittelung über die Trainingsstichprobe hervorgegangen sein kann [Mar00b].

Die Skalierung in horizontaler Richtung basiert auf der geschätzten Buchstabenbreite. Als Maß wird hierfür oftmals das Verhältnis der mittleren Anzahl von schwarz-weiß (Schrift-Hintergrund) Übergängen in der Mittelzone der Schrift zur Breite der Bounding Box der Schrift verwandt [Mar00b]. Neben der Anzahl der schwarz-weiß Übergänge kann die Skalierung beispielsweise auch auf der Anzahl der lokalen Minima der Schriftkontur basieren [Mad99b].

Korrektur der Schriftneigung

Die Schriftneigung ist definiert als der mittlere Winkel, den die eher vertikal verlaufenden Schriftabschnitte mit dem Lot einschließen (siehe Abbildung 3.13). Da dieser Winkel zwischen unterschiedlichen Schreibern stark variieren kann, wird i.d.R. eine Normierung ausgeführt, sodass die Schrift anschließend einen Neigungswinkel von Null Grad aufweist, also senkrecht ausgerichtet ist. Die Normierung wird dabei durch eine Transformation vorgenommen, die einer *Scherung* des Bildes entspricht. Mit dem Neigungswinkel α und den Bildkoordinaten x und y ist die Scherung folgendermaßen definiert:

$$\begin{aligned} x' &= x - y \tan \alpha \\ y' &= y \end{aligned} \tag{3.20}$$

Häufig wird zur Bestimmung des Neigungswinkels in zwei Schritten vorgegangen [Mar00a, Boz89, Kim97]: Da der Neigungswinkel definiert ist als der Winkel zwischen den annähernd vertikalen Segmenten des Schriftzuges und dem Lot, besteht der erste Schritt in der Extraktion dieser Segmente aus dem Schriftbild. Im zweiten Schritt wird dann anhand der extrahierten Bildbereiche der Neigungswinkel bestimmt.

Dieses zweistufige Vorgehen wird beispielsweise in [Boz89] befolgt (siehe Abbildung 3.14). Dabei werden zuerst horizontale Streifen aus dem binarisierten Schriftbild

ausmaskiert, die längere zusammenhängende horizontale Schriftbereiche aufweisen. Anschließend werden die übrigen Streifen dann vertikal in Abschnitte aufgeteilt, anhand deren jeweils zwei Schwerpunkte berechnet werden – einer bzgl. der oberen Hälfte und einer bzgl. der unteren Hälfte des Abschnitts. Die Steigung der Verbindungslinie der beiden Schwerpunkte definiert dann den Neigungswinkel des entsprechenden Abschnitts. Die resultierende Neigung des gesamten Schriftbildes ergibt sich schließlich aus der Mittelung über die Neigungswinkel aller Abschnitte.

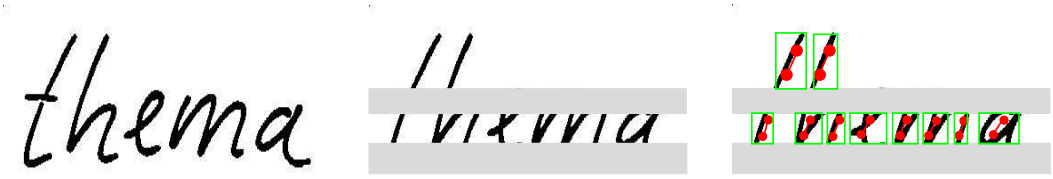
Eine einfache Möglichkeit zur Berechnung des Neigungswinkel anhand einer Kettencode-Repräsentation der Schrift ergibt sich aus der Verwendung des Histogramms der Kettencode-Richtungen (siehe u.a. [Din00]). Mit der Bezeichnung n_i für die Anzahl der Kettencode-Elemente der Richtung $i \times 45°, i = 0, \ldots, 3$, gemäß Abbildung 3.15 ergibt sich der mittlere Neigungswinkel durch

$$\alpha = \arctan \frac{n_1 - n_3}{n_1 + n_2 + n_3}. \tag{3.21}$$

Mit Hilfe der obigen Formel lässt sich eine hinreichend genaue Schätzung des Neigungswinkels im Bereich $-45°, \ldots, +45°$ durchführen. In [Din00] wurde darüberhinaus gezeigt, dass sich durch Verwendung einer verfeinerten Richtungsauflösung die Genauigkeit der Schätzung weiter verbessern lässt.

Eine Neigungskorrektur auf Basis eines Histogramms wird auch in [Sen98] vorgeschlagen. Hierbei wird ein Kantenfilter, in diesem Fall der Canny-Operator, auf das Intensitätsbild angewandt. Damit wird den Kantenpixeln jeweils eine Gradientenrichtung zugewiesen, die für das gesamte Bild in einem Richtungshistogramm akkumuliert werden. Anhand derjenigen Gradientenrichtung, die dem globalen Maximum des Richtungshistogramms entspricht, wird dann der Neigungswinkel ermittelt.

Eine weitere Möglichkeit zur Neigungskorrektur besteht in der Anwendung iterativer Projektionsmethoden [Bus97, Vin00, Kav02]. Das Vorgehen ist dabei mit der auf Seite 42 beschriebenen Orientierungskorrektur vergleichbar, wobei hier jedoch anstatt der horizontalen Projektion die vertikale Projektion verwendet wird. Die Grundannahme dieser Verfahren ist, dass das vertikale Projektionshistogramm bei neigungs-



(a) (b) (c)

Abbildung 3.14: Bestimmung des Neigungswinkels nach [Boz89]. Ausgangsbild (a), Ausmaskierung von Streifen, die horizontale Anteile des Schriftzuges enthalten (b) und Verbindungsgeraden der Abschnittsschwerpunkte (c).

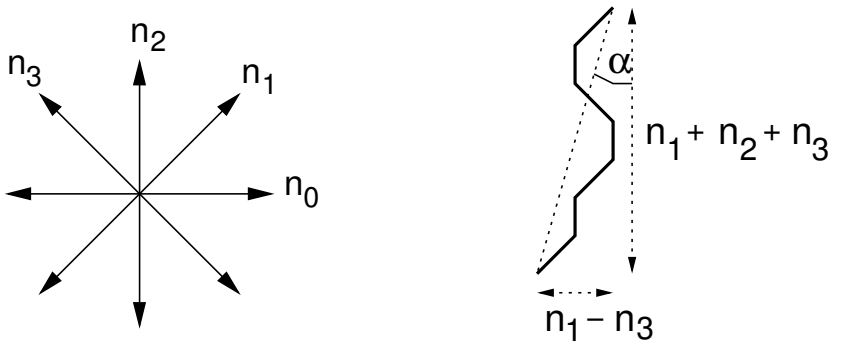


Abbildung 3.15: Bestimmung des Neigungswinkels anhand des Kettencodes [Din00].

korrigierter Schrift besonders ausgeprägte Merkmale aufweist, sodass durch iterative Scherung des Bildes mittels gradueller Änderung des Scherwinkels der optimale Neigungswinkel der Schrift ermittelt werden kann.

Das in [Bus97] vorgestellte Verfahren basiert beispielsweise auf der Beobachtung, dass das vertikale Projektionshistogramm bei neigungskorrigierter Schrift maximal kompakt ist und somit minimale Entropie

$$E = -\sum_i p_i \log p_i$$

aufweist, wobei in obiger Formel p_i den Histogrammeintrag für die Bildspalte i bezeichnet. Derjenige Scherwinkel, der zu minimaler Entropie des korrespondierenden Histogramms führt, wird somit als optimaler Neigungswinkel angenommen.

3.3.2 Online Systeme

In diesem Abschnitt werden die Verfahren zur Signalvorverarbeitung vorgestellt, die in Systemen zur online Handschrifterkennung eingesetzt werden. Im Gegensatz zum offline Bereich liegt hier keine bildhafte Information vor, sondern der Schriftzug wird durch die Stifttrajektorie in Form einer zeitlich geordneten Sequenz von Koordinatenpunkten repräsentiert.

$$\ldots, (t_k, \boldsymbol{x}_k), (t_{k+1}, \boldsymbol{x}_{k+1}), \ldots \qquad t_k < t_{k+1} \tag{3.22}$$

Hierbei bezeichnet t_k den Zeitpunkt der Abtastung und $\boldsymbol{x}_k$ den Vektor der aufgenommen Daten. Der Vektor $\boldsymbol{x}_k$ enthält die Stiftkoordinaten x_k und y_k und gegebenenfalls weitere Daten, wie z.B. den Anpressdruck oder die Stiftneigung.

$$\boldsymbol{x}_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \tag{3.23}$$

Die Vorverarbeitung der Stifttrajektorie gliedert sich ähnlich wie im offline Bereich in die beiden Phasen Rauschunterdrückung und Schriftnormalisierung. Um etwaige

Störungen in den Daten zu reduzieren, die z.B. durch die Digitalisierung hervorgerufen wurden, wird neben der Elimination von "Ausreißerpunkten" häufig eine Interpolation und Glättung der Trajektorie vorgenommen. In der zweiten Vorverarbeitungsphase wird die extrahierte Handschrift normalisiert, um die schreiber- und situationsspezifische Variabilität der Schrift zu kompensieren. So wird häufig sowohl die Schreibgeschwindigkeit, als auch die Neigung, Orientierung und Skalierung der Schrift durch geeignete Verfahren korrigiert.

Elimination von "Ausreißerpunkten"

Die "Ausreißer" sind Punkte der Trajektorie, die nicht durch die Stiftbewegungen sondern durch Störungen erzeugt wurden. Diese können einerseits durch fehlerhaft arbeitende Hardware hervorgerufen werden oder andererseits speziell bei drucksensitiven Digitalisiertabletts dadurch verursacht werden, dass ein Finger auf die Eingabefläche drückt. Die fehlerhaft aufgenommenen Punkte rufen in der Regel hohe Geschwindigkeits- und Beschleunigungsvariationen im Signal hervor, sodass Ausreißer mit Hilfe von Schwellwerten detektiert werden können, die sich an den maximal erreichbaren Beschleunigungen bei natürlichen Schreibbewegungen orientieren [Tap90][Gue93].

Interpolation

Neben dem Auftreten von "Ausreißern" kann bei der Signalaufnahme auch der Fall eintreten, dass z.B. durch einen zu geringen Anpressdruck des Stifts nicht alle Punkte des Schriftzuges vom Digitalisiertablett aufgenommen werden. In diesem Fall weisen zeitlich benachbarte Trajektorienpunkte einen großen räumlichen Abstand auf. Dieser Effekt tritt auch auf, wenn Digitalisiertabletts eingesetzt werden, die die Stiftposition in pen-up Phasen nicht detektieren können oder wenn die Schreibgeschwindigkeit im Verhältnis zur Abtastrate des Aufnahmegeräts sehr hoch ist. Um Lücken in den Trajektoriendaten zu schließen, können mit Hilfe von Interpolationsverfahren Zwischenpunkte berechnet werden, die auf einer Kurve liegen, die durch die aufgenommenen Datenpunkte verläuft.

Die einfachste Möglichkeit zur Berechnung von Zwischenpunkten stellt die lineare Interpolation dar. Dabei werden zwei benachbarte Datenpunkte $\boldsymbol{x}_k$ und $\boldsymbol{x}_{k+1}$ durch eine Gerade miteinander verbunden.

$$\boldsymbol{p}(t) = \boldsymbol{x_k} + \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{t_{k+1} - t_k}(t - t_k) \tag{3.24}$$

Die Position eines Zwischenpunktes $\boldsymbol{x}_s = \boldsymbol{p}(t_s), \;\text{ mit }\; t_k \leq t_s \leq t_{k+1}$, ist daher nur abhängig von seinen beiden unmittelbar benachbarten Punkten. In [Sch95a] wird die lineare Interpolation beispielsweise eingesetzt, um die Trajektorie während einer pen-up Phase zu berechnen.

Vielfach ist es jedoch wünschenswert, einen größeren Kontext als nur zwei Nachbarpunkte zur Interpolation zu verwenden. Dies führt dann zur Interpolation durch

Polynome. Werden dabei n Datenpunkte herangezogen, so besitzt das Interpolationspolynom den Grad $n-1$. Zur Ermittlung des Interpolationspolynoms $\boldsymbol{p}(t)$ eignet sich z.B. das *Neville-Schema*, das mittels Rekursion die Berechnung des Interpolationspolynoms vom Grad n auf den Grad $n-1$ zurückführt.

Bezeichne $\boldsymbol{p}_k^{k+n}(t)$ das Polynom n-ten Grades, das die Daten $(t_i, \boldsymbol{x}_i), i = k, k+1, \ldots, k+n$ interpoliert. Dann gilt folgende Rekursion [Opf94]:

$$\boldsymbol{p}_k^k(t) = \boldsymbol{x}_k \tag{3.25}$$

$$\boldsymbol{p}_k^{k+n}(t) = \frac{(t-t_k)\boldsymbol{p}_{k+1}^{k+n}(t) - (t-t_{k+n})\boldsymbol{p}_k^{k+n-1}(t)}{t_{k+n}-t_k} \tag{3.26}$$

Nach kurzer Umformung von 3.26 wird deutlich, dass sich für $n=1$ der Spezialfall der linearen Interpolation ergibt.

Die Erhöhung des Polynomgrades mit der Anzahl der Datenpunkte, die zur Interpolation herangezogen werden, führt jedoch dazu, dass die Interpolationskurve außerhalb der Punkte $\boldsymbol{x}_k$ stark von den Daten abweicht. Eine Lösung dieses Problems liegt in der stückweisen Interpolation, indem für jedes Teilintervall $[t_k, t_{k+1}]$ ein Polynom berechnet wird und diese zur Interpolationsfunktion $\boldsymbol{g}(t)$ zusammengesetzt werden. Stellt man darüberhinaus Glattheitsbedingungen an $\boldsymbol{g}(t)$, so führt dies zum Konzept des *Splines*.

Ein Spline ist eine aus Polynomen stückweise zusammengesetzte Funktion, die an den Intervallgrenzen t_k, den sogenannten Knotenpunkten, bestimmten Glattheitsbedingungen genügen muss. Bei den üblicherweise verwendeten Polynomen dritten Grades ist der *kubische Interpolationsspline* $\boldsymbol{g}(t)$ demnach eindeutig durch folgende Eigenschaften festgelegt [Bro01]:

1. $\boldsymbol{g}(t)$ durchläuft die Datenpunkte, d.h. $\boldsymbol{g}(t_k) = \boldsymbol{x}_k$ (Interpolationsbedingung).
2. $\boldsymbol{g}(t)$ ist in jedem Teilintervall $[t_k, t_{k+1}]$ ein Polynom vom Grad ≤ 3.
3. $\boldsymbol{g}(t)$ und seine ersten beiden Ableitungen sind stetig im gesamten Intervall $[t_1, t_N]$
4. $\boldsymbol{g}(t)$ erfüllt vorgegebene Randbedingungen bei t_1 und t_N. Wählt man beispielsweise $\boldsymbol{g}''(t_1) = \boldsymbol{g}''(t_N) = 0$, so erhält man einen *natürlichen* Spline.

Es lässt sich zeigen, dass der natürliche Interpolationsspline $\boldsymbol{g}(t)$, der den obigen Anforderungen genügt, unter allen zweimal stetig differenzierbaren Funktionen $\boldsymbol{f}(t)$ die minimale Gesamtkrümmung hat. Da die Krümmung einer Kurve in erster Näherung proportional zur zweiten Ableitung ist, gilt:

$$\int_{t_1}^{t_N} [\boldsymbol{g}''(t)]^2 dt \leq \int_{t_1}^{t_N} [\boldsymbol{f}''(t)]^2 dt \tag{3.27}$$

Die Berechnung des eindeutigen natürlichen Interpolationssplines entspricht daher exakt der Aufgabe, eine Interpolationskurve mit minimaler Krümmung zu finden, die

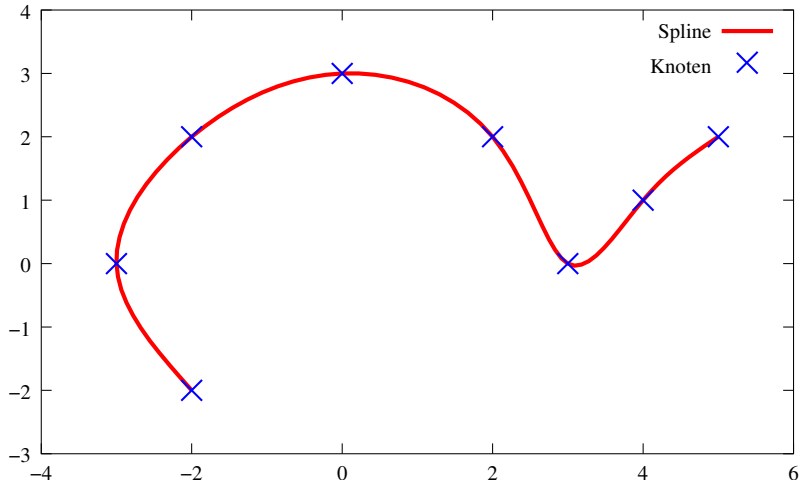


Abbildung 3.16: Kubischer Interpolationsspline.

also das Integral der quadrierten zweiten Ableitung minimiert. Näheres hierzu lässt sich z.B. in [Gre94] nachlesen.

Zur Berechnung des kubischen Interpolationssplines $\boldsymbol{g}(t)$ wird häufig folgende Darstellung für die Polynome der Teilintervalle $[t_k, t_{k+1}]$ gewählt [Bro01]:

$$\boldsymbol{g}(t) = \boldsymbol{g}_k(t) = \boldsymbol{a}_k + \boldsymbol{b}_k(t - t_k) + \boldsymbol{c}_k(t - t_k)^2 + \boldsymbol{d}_k(t - t_k)^3 \quad (k = 1, 2, \ldots, N - 1). \tag{3.28}$$

Anhand der Interpolationsbedingung und den Glattheitsbedingungen an den inneren Knoten lässt sich nun ein lineares Gleichungssystem aufstellen, mit dessen Hilfe die Polynomkoeffizienten $\boldsymbol{a}_k, \boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k$ bestimmt werden können. In der Abbildung 3.16 ist der natürliche kubische Interpolationsspline veranschaulicht, der sich aus den dargestellten Datenpunkten ergibt.

Glättung

Ein weiterer Vorverarbeitungsschritt, der in online Systemen sehr häufig angewandt wird, ist die Glättung der aufgenommenen Stifttrajektorie. Damit soll hochfrequentes Rauschen unterdrückt werden, das z.B. durch Digitalisierungsfehler hervorgerufen wird und die Handschrift überlagert. Im Gegensatz zur Interpolation wird hier nicht verlangt, dass die geglättete Schrifttrajektorie exakt durch die aufgenommenen Datenpunkte verläuft. Es wird vielmehr verlangt, dass die eventuell mit Messfehlern behafteten Daten möglichst glatt durch eine Kurve approximiert werden.

Eine sehr populäre weil schnelle Methode zur Glättung ist das *local weighted averaging*. Bei diesem Verfahren wird der aktuelle Trajektorienpunkt $\boldsymbol{x}_k$ ersetzt durch einen gewichteten Mittelwert, der in einem Fenster der Größe $2n + 1$, das um den aktuellen Punkt zentriert ist, berechnet wird:

$$\boldsymbol{x}'_k = \sum_{j=-n}^{n} \alpha_j \boldsymbol{x}_{k+j} \tag{3.29}$$

Gewichtet man jeden Datenpunkt gleich, so erhält man für die Gewichte α_j:

$$\alpha_j = \frac{1}{2n+1} \qquad \forall j : -n \leq j \leq n \tag{3.30}$$

Mit dieser Parameterwahl wird ein Rechteckfilter realisiert, der jedoch kein gutes Glättungsfilter darstellt. Anhand der Fouriertransformierten der Rechteckfunktion wird deutlich, dass die Amplitudendämpfung des Signals nicht mit der Frequenz monoton zunimmt, wie es eigentlich erwünscht wäre, sondern oszilliert [Jäh97].

Ein besseres Glättungsverhalten kann erreicht werden, wenn die Filterparameter α_j anhand der Gaußverteilung gewählt werden [Jäh97].

$$\alpha_j = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{j^2}{2\sigma^2}} \qquad \forall j : -n \leq j \leq n \tag{3.31}$$

Die Gaußfunktion hat die Eigenschaft, dass ihre Fouriertransformierte wiederum eine Gaußfunktion ist, sodass die Amplitudendämpfung monoton mit der Frequenz des Signals ansteigt. Bei entsprechender Wahl der Varianz σ bleiben somit nur die tiefen Frequenzen erhalten, die dem Handschriftsignal entsprechen, wohingegen höherfrequentes Rauschen ausgefiltert wird.

Oftmals wird jedoch auch eine spezielle Wahl der Parameter α_j getroffen, um die Trajektorie einerseits möglichst gut zu glätten, andererseits aber besondere Merkmale der Kurve, wie z.B. scharfe “Zacken”, zu erhalten. Würde beispielsweise die Trajektorie eines ’V’ stark geglättet, so ähnelte sie einem ’U’, da anstelle der charakteristischen Ecke eine mehr oder weniger starke Rundung treten würde. Dieses Phänomen lässt sich z.B. mit folgender Parametrisierung berücksichtigen [Jae01]:

$$\alpha_j = \begin{cases} \frac{c}{2n+c} & \text{if } j = k \\ \frac{1}{2n+c} & \text{otherwise} \end{cases} \qquad \forall j : -n \leq j \leq n \quad c \geq 1 \tag{3.32}$$

Je größer man hier den Gewichtsparameter c des aktuellen Trajektorienpunkts wählt, desto näher verläuft die Kurve an diesem Punkt und desto besser bleibt eine scharfe “Zacke” in der Trajektorie erhalten.

Eine weitere Möglichkeit zur Glättung von Trajektorien besteht in der Approximation der Datenpunkte durch einen *Ausgleichsspline* $\boldsymbol{g}(t)$. Da bei der Approximation im Gegensatz zur Interpolation nicht verlangt wird, dass die Datenpunkte $\boldsymbol{x}_k$ exakt durchlaufen werden, sondern vielmehr eine Balance zwischen einer guten Datenanpassung und einer ausreichenden Glattheit der Kurve gefordert ist, wird zur Splineberechnung die Interpolationsforderung ersetzt durch eine Extremwertaufgabe, bei der folgender Ausdruck bezüglich der Funktion $\boldsymbol{g}(t)$ zu minimieren ist [Bro01]:

$$\sum_{k=1}^{n} \left[\frac{\boldsymbol{x}_k - \boldsymbol{g}(t_k)}{\sigma_k} \right]^2 + \lambda \int_{t_1}^{t_n} [\boldsymbol{g}''(u)]^2 du = \min! \tag{3.33}$$

Hierbei misst der erste Term den gewichteten Abstand zwischen den Daten und der Schätzfunktion, während der zweite Term die Gesamtkrümmung der Kurve beschreibt.

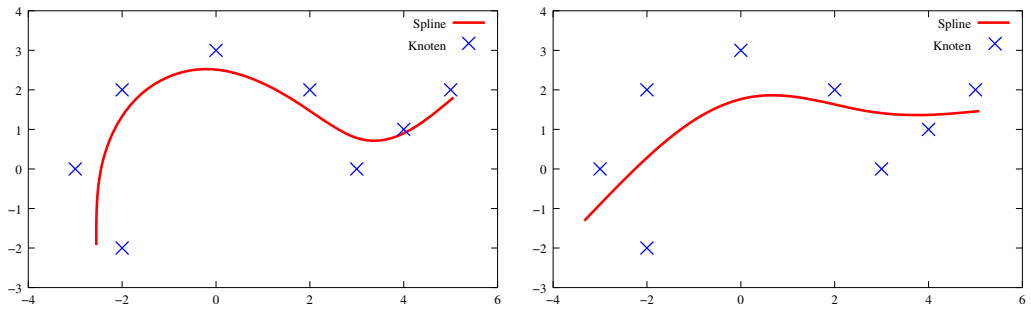


Abbildung 3.17: Kubische Ausgleichssplines. Rechts mit größerem Glättungsparameter λ.

Die Parameter σ_k bezeichnen die Standardabweichungen der Messfehler. Da die Stetigkeitseigenschaften von $\boldsymbol{g}(t)$, $\boldsymbol{g}'(t)$ und $\boldsymbol{g}''(t)$ wie im Fall der Interpolation weiterhin gelten müssen, entspricht die Minimierung des Ausdrucks 3.33 einer Extremwertaufgabe mit Nebenbedingungen. Die Lösung erfolgt üblicherweise mit Hilfe einer Lagrange-Funktion [Bro01]. Näheres zur Berechnung des Ausgleichssplines findet sich u.a. in [Rei67], [dB78], [Gre94] und [Win98].

Es zeigt sich, dass die optimale Funktion $\hat{\boldsymbol{g}}(t)$, die den obigen Ausdruck 3.33 minimiert und somit die Daten optimal approximiert, ein kubischer Ausgleichspline ist. Weiterhin lässt sich zeigen, dass $\hat{\boldsymbol{g}}(t)$ eindeutig ist [Rei67][Gre94].

Die Glattheit der Spline-Approximation kann dabei durch den Parameter λ variiert werden. Für $\lambda = 0$ erhält man eine Interpolationskurve, die alle Datenpunkte durchläuft, wohingegen für $\lambda > 0$ eine Approximationskurve berechnet wird, die umso glatter ist, je größer man den Wert für λ wählt. Für $\lambda \to \infty$ erhält man als Schätzfunktion eine Ausgleichsgerade. In der Abbildung 3.17 sind zwei kubische Ausgleichssplines mit unterschiedlichen Parametern λ dargestellt.

Es sei hier nur am Rande erwähnt, dass die Ausgleichsspline ähnlich wie in 3.29 auch als gewichteter Mittelwertfilter dargestellt werden kann. Dies liegt an der quadratischen Natur der Extremwertaufgabe 3.33, aus der sich unmittelbar ergibt, dass die Datenpunkte linear in die Berechnung des optimalen Ausgleichssplines eingehen. Näheres hierzu findet sich in [Gre94].

Ermittlung von Referenzlinien

Ähnlich wie im Bereich der offline Handschrifterkennung lassen sich die Verfahren zur Referenzlinienschätzung auch bei den online arbeitenden Systemen in die Gruppe der histogrammbasierten Methoden einerseits, und die Gruppe der Verfahren einteilen, die andererseits auf der Anpassung geometrischer Modelle an die Schriftkontur beruhen.

Die Lokalisation der Mittelzone der Schrift bei den histogrammbasierten Methoden geschieht durch die Detektion des charakteristischen Plateaus im horizontalen Projektionshistogramm. Im online Bereich erhält man dieses Histogramm, indem die Schrift mit einem Zeilenraster überlagert wird, und für jede Zeile die Häufigkeit, mit der die

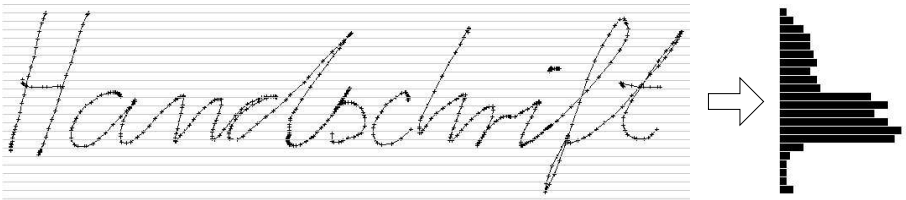


Abbildung 3.18: Überlagerung der Schrifttrajektorie mit einem Zeilenraster. Rechts ist das daraus resultierende Projektionshistogramm dargestellt. Dabei ist auf der x-Achse die Anzahl der Schnittpunkte zwischen Trajektorie und der jeweiligen Rasterzeile aufgetragen.

Trajektorie die Zeile kreuzt, in einem sogenannten Projektionshistogramm aufgetragen wird [Bro83] (siehe Abbildung 3.18). Das charakteristische Plateau des Histogramms, das die Mittelzone der Schrift kennzeichnet, kann dann mit Hilfe eines Schwellwerts lokalisiert werden, der die minimale Anzahl von Schnittpunkten zwischen Rasterzeile und Schrifttrajektorie festlegt.

Im überwiegenden Teil der online Systeme werden die Referenzlinien jedoch durch Approximation von Geraden an die Schriftkontur geschätzt [Sch95a]. Als Stützpunkte dienen dabei die lokalen vertikalen Extrema des Schriftzuges, also die Punkte, an denen die vertikale Richtungskomponente das Vorzeichen wechselt oder eine pen-up Phase beginnt.

Im einfachsten Fall können die Referenzlinien als Ausgleichsgeraden durch die Stützpunkte mit Hilfe der linearen Regression berechnet werden. Da der entscheidende Schritt hierbei die Zuordnung der extrahierten Extrema zu der jeweiligen zu schätzenden Referenzlinie ist, wird ähnlich wie im offline Bereich die Referenzlinienschätzung in einem zweistufigen Prozess vorgenommen [Gue93, Sch95a]. In [Sch95a] wird beispielsweise im ersten Schritt jeweils eine lineare Regressionsgerade durch alle lokalen Minima bzw. Maxima berechnet. Anschließend werden die Extrema entfernt, die den größten Abstand zu der jeweiligen Regressionsgerade aufweisen. Mit den übrigen Extremstellen wird dann im zweiten Schritt eine genauere Schätzung der Referenzlinien vorgenommen.

Eine flexiblere Referenzlinienschätzung wird in [Ben94] vorgestellt. Dabei werden die vier Referenzlinien in Form von Parabeln an die Stifttrajektorie angepasst. Außerdem wird das Problem der Mehrdeutigkeit bei der “harten” Zuordnung der lokalen Extrema zu den jeweiligen Referenzlinien in einem probabilistischen Rahmen behandelt. Dazu wird, ausgehend von einem Parametersatz $\boldsymbol{\theta}$, der die vier Parabeln beschreibt, der Abstand eines Extremums zu den jeweiligen Referenzlinien als Wahrscheinlichkeit dafür aufgefasst, dass das Extremum aus dem mit $\boldsymbol{\theta}$ parametrisierten Modell hervorging. Mit Hilfe des EM-Algorithmus [Dem77] wird dann der initiale Parametersatz $\boldsymbol{\theta}$ iterativ optimiert, sodass die Anpassung der Parabeln an die Schrifttrajektorie sukzessive verbessert wird.

Korrektur der Schriftorientierung und -größe

Die geschätzten Referenzlinien der Stifttrajektorie, die i.d.R. wie oben skizziert durch die Approximation von Ausgleichsgeraden [Gue93, Sch95a] oder der Anpassung von Parabeln [Ben94, Jae01] an die lokalen Minima bzw. Maxima der Trajektorie ermittelt werden, bilden üblicherweise die Basis sowohl für die horizontale Ausrichtung der Stifttrajektorie als auch für die Skalierung der Schrift. Da von allen Referenzlinien die Basislinie auf Grund der größeren Anzahl von Stützpunkten am sichersten geschätzt werden kann, wird die Korrektur der Schriftorientierung oftmals anhand des Winkels vorgenommen, den die Basislinie mit der Horizontalen einschließt [Gue93, Sch95a].

Die Korrektur der Schriftgröße basiert i.d.R. auf der geschätzten Basis- und Mittellinie der Schrift. Dabei wird der Abstand zwischen diesen Linien, also die Korpushöhe der Schrift, ermittelt und auf einen vorgegebenen Wert skaliert (siehe u.a. [Bro83, Bei95, Sch95a, Jae01]).

Das Vorgehen bei der Orientierungs- und Größenkorrektur ist damit ähnlich wie bei den entsprechenden Methoden im Bereich der offline Handschrifterkennung, die auf dem statischen Schriftbild basieren. Daraus folgt jedoch auch, das im Gegensatz zur Glättung oder Interpolation der Trajektorie die Korrektur der Schriftorientierung nicht schritthaltend mit der Signalaufnahme durchgeführt werden kann, da zur sicheren Schätzung der Referenzlinien ein längerer Schriftabschnitt erforderlich ist. Damit erhöht sich jedoch die Antwortzeit, also die Zeitspanne zwischen Signalaufnahme und Vorliegen des Erkennungsergebnisses, sodass in einigen online Systemen auf die Korrektur von Schriftorientierung und -größe verzichtet wird [Dol97].

Korrektur der Schriftneigung

Die Verfahren, die zur Neigungskorrektur der Stifttrajektorie eingesetzt werden, sind mit den entsprechenden Methoden aus dem Bereich der offline Handschrifterkennung vergleichbar. Die Neigungskorrektur wird dabei ebenfalls global durchgeführt, also anhand eines Winkels, der für den gesamten betrachteten Textabschnitt (i.d.R. ein Wort) gilt.

Das in [Gue93] beschriebene Verfahren basiert beispielsweise auf der in [Boz89] vorgestellten Vorgehensweise (siehe Seite 43). Dabei wird der Schriftzug gemäß der geschätzten Basis- und Mittellinie in drei horizontale Streifen (Ober-, Mittel- und Unterbereich) aufgeteilt, die dann anhand der Schriftkomponenten vertikal in Abschnitte unterteilt werden. Die Verbindungsgerade durch den unteren und oberen Abschnittsschwerpunkt definiert die lokale Neigung des Abschnitts. Der Neigungswinkel des gesamten Worts wird durch Mittelung über die lokalen Neigungswinkel der Abschnitte berechnet.

Eine ähnliche Methode zur Neigungskorrektur wird in [Bro83] beschrieben. Dort werden zwei horizontale Schnittgeraden durch den Mittelbereich des Schriftzuges gelegt, wobei die mittlere Steigung an den Kreuzungspunkten der Stifttrajektorie mit den Schnittgeraden als Neigungswinkel angenommen wird.

In [Jae01] wird ein Verfahren zur Neigungskorrektur der Stifttrajektorie vorgestellt, das auf einem Winkelhistogramm beruht. Dieses Histogramm wird aus den Steigungswinkeln der Verbindungsgeraden aufeinanderfolgender Trajektorienpunkte gebildet, wobei die Histogrammeinträge dabei mit dem Abstand zwischen den beiden betrachteten Punkten gewichtet werden. Der resultierende Neigungswinkel ergibt sich dann anhand des Maximums des Histogramms.

Neuabtastung

Einen weiteren Bestandteil der Vorverarbeitung stellt in vielen Systemen zur online Handschrifterkennung die Neuabtastung (*Resampling*) der Stifttrajektorie dar. Das Ziel dieses Schritts ist die Kompensation unterschiedlicher Schreibgeschwindigkeiten. Da die Digitalisiertabletts die Stiftkoordinaten in gleichmäßigen Zeitintervallen ermitteln, weisen die Trajektorienpunkte bei einer langsamen Schreibbewegung einen kleineren räumlichen Abstand zueinander auf als bei einer schnellen Schreibbewegung. Die Schreibgeschwindigkeit ist jedoch ein schreiberspezifisches Merkmal, sodass die Trajektorie vor der Weiterverarbeitung häufig in eine von der Schreibgeschwindigkeit unabhängige Repräsentation überführt wird.

Dazu wird die Trajektorie neu abgetastet, sodass anschließend der jeweilige *räumliche* Abstand zwischen allen aufeinanderfolgenden Punkten einem vorgegebenen Wert entspricht. Dieser Wert ist meistens festgelegt auf einen Bruchteil der geschätzten Korpushöhe der Schrift, also der Distanz zwischen Basis- und Mittellinie [Sch95a, Jae01]. In [Kos97] wird dagegen mit einem variablen Abstand gearbeitet, indem bei der Neuabtastung die lokale Krümmung der Stifttrajektorie berücksichtigt wird. Damit ist an stärker gekrümmten Abschnitten der Trajektorie der resultierende Punktabstand geringer als in weniger gekrümmten Bereichen. Dieses Verfahren stellt damit einen Kompromiss dar zwischen der Glattheit der Trajektorie und dem Speicheraufwand, also der Anzahl benötigter Datenpunkte.

Häufig wird die Neuabtastung im Zuge der Interpolation bzw. Glättung schritthaltend mit der Aufnahme der Stifttrajektorie ausgeführt. Ist beispielsweise der Abstand zwischen einem gerade aufgenommenen Punkt und seinem Vorgänger größer als der für die Neuabtastung vorgegebene Abstand, so werden entsprechend der Interpolationsvorschrift zusätzliche Punkte eingefügt, die den vorgegebenen Abstand zueinander aufweisen. Ist andererseits der Abstand kleiner, so wird der neu aufgenommen Punkt nicht in der Trajektorie gespeichert.

Behandlung von "delayed strokes"

Die Verarbeitung der Stifttrajektorie bei der online Handschrifterkennung erfolgt grundsätzlich sequentiell, die Koordinatenpunkte werden also in derselben Reihenfolge abgearbeitet, in der sie auch aufgenommen wurden. Dabei wird vorausgesetzt, dass bei der Schriftgenerierung ein Buchstabe erst vollständig geschrieben wurde, bevor mit dem nächsten Buchstaben fortgefahren wurde. Diese Annahme ist jedoch nicht

immer erfüllt. So wird oftmals der i-Punkt oder der t-Strich nicht unmittelbar nach dem Basisbuchstaben angefügt, sondern mittels sogenannter *delayed strokes* erst nachdem weitere Buchstaben geschrieben wurden oder das Wortende erreicht wurde. Im Deutschen trifft dies auf diakritische Zeichen zu, wie z.B. den Umlautpunkten.

Um trotz des Auftretens von delayed strokes eine sequentielle Verarbeitung der Trajektorie durchführen zu können, sind unterschiedliche Vorgehensweisen vorgeschlagen worden, die auf verschiedenen Ebenen – Vorverarbeitung, Merkmalsextraktion oder Modellierung – ansetzen. So werden beispielsweise in [Sch93] delayed strokes am Wortende während der Vorverarbeitung detektiert und aus der Eingabesequenz entfernt, sodass die Weiterverarbeitung mit unvollständigen Eingabedaten erfolgt. In [Tap82] wird dagegen während der Vorverarbeitungsphase eine Umsortierung der Eingabedaten vorgenommen, d.h. die delayed strokes werden an die "richtige" Position – unmittelbar nach dem Basisbuchstaben – eingefügt. Eine Schwierigkeit dabei ist, dass es oftmals nicht eindeutig ist, zu welchem Basisbuchstaben der delayed stroke gehört. Ist die "harte" Zuordnung während der Vorverarbeitung fehlerhaft, so sind falsche Erkennungsergebnisse kaum zu vermeiden [Sen99].

Eine weitere Möglichkeit zur Behandlung von delayed strokes wird u.a. in [Sch95a, Jae01] beschrieben. Diese Systeme sind dadurch gekennzeichnet, dass die Merkmalsrepräsentation um ein sogenanntes "hat-feature" erweitert wird, mit dem an der entsprechenden Position der Eingabesequenz das Vorhandensein eines delayed strokes angezeigt wird. Die delayed strokes werden dabei im Vorhinein anhand einfacher Heuristiken detektiert und anschließend aus der Eingabesequenz entfernt. Der Vorteil dieser Methode ist, dass durch die Berücksichtigung der delayed strokes als eine Komponente des Merkmalsvektors frühe "harte" Entscheidungen vermieden werden.

Das in [Sen99] dargestellte Verfahren umgeht ebenfalls eine frühe unumkehrbare Zuordnung der delayed strokes zu den Basisbuchstaben, indem die möglichen Kombinationen von delayed strokes und Basisbuchstaben erst in der Erkennungsphase durch den Buchstabenklassifikator bewertet werden. Durch eine Zuordnungstabelle wird verhindert, dass ein delayed stroke mehreren Basisbuchstaben zugeordnet wird. Außerdem wird mit Hilfe von Straftermen sichergestellt, dass bei dieser Prozedur alle delayed strokes berücksichtigt werden.

In [Hu96, Hu00] werden die delayed strokes auf der Modellierungsebene behandelt. Sie werden dabei als spezielle Zeichen aufgefasst, die wie "normale" Buchstaben auf Basis von Hidden Markov Modellen modelliert werden. Somit kann die fehleranfällige Detektion in der Vorverarbeitungsphase komplett umgangen werden. Für jedes Wort des Lexikons, das potentiell delayed strokes aufweisen kann, müssen dann allerdings mehrere Repräsentationen vorliegen, je nachdem wann der delayed stroke ausgeführt wurde. Um die damit verbundene drastische Erhöhung der Anzahl der Worthypothesen einzuschränken, wird in [Hu00] eine vereinfachte Modellierung eingesetzt, bei der delayed strokes nur direkt hinter dem Basisbuchstaben, unmittelbar davor oder am Ende des Wortes vorkommen. Anhand der resultierenden Wortkandidaten wird anschließend die optimale Lösung durch einer verfeinerte delayed stroke Modellierung ermittelt.

3.4 Segmentierung

Eine häufig vorgenommene Kategorisierung von Systemen zur automatischen Handschrifterkennung richtet sich nach der *Segmentierung* des vorverarbeiteten Schriftsignals (siehe u.a. [Ste99]). Mit Segmentierung ist die Unterteilung des Signals in Basiseinheiten gemeint, die Wörtern, Buchstaben oder Strokes entsprechen, und als Grundlage für den nachfolgenden Klassifikationsschritt dienen. Der Begriff der Segmentierung bezieht sich hier also nicht auf die Binarisierung, die in der Bildverarbeitung allgemein ja als Segmentierungsoperation aufgefasst wird, in Bezug auf die Handschrifterkennung jedoch als Vorverarbeitungsmaßnahme gilt.

Ist das betrachtete System zur Handschrifterkennung nicht auf die Verarbeitung isolierter Wörter beschränkt, so wird nach der Vorverarbeitung des Signals häufig eine Wortsegmentierung der Textzeile vorgenommen. Basiert die Erkennung auf einer holistischen Strategie, werden also Wortmodelle für die Erkennung verwendet, so findet anschließend keine weitere Segmentierung der Wörter in Buchstaben oder Strokes statt. Die Nachteile der holistischen Strategie, vor allem die Abhängigkeit von einem vorgegebenen Lexikon, vermeidet der analytische Ansatz, indem die Klassifikation anhand von Wortuntereinheiten vorgenommen wird. Dabei werden zwei Grundrichtungen unterschieden: Explizite und implizite Segmentierung. Bei der expliziten Segmentierung wird vor der Klassifikation das Signal in symbolische Einheiten – meistens Buchstaben – zerlegt, die anschließend isoliert voneinander klassifiziert werden. Der implizite Segmentierungsansatz kann dagegen frühe, unumkehrbare Segmentierungsentscheidungen dadurch vermeiden, dass die Segmentierung erst im Zuge der Klassifikation vorgenommen wird. Der implizite Ansatz wird daher auch häufig erkennungsbasierte Segmentierung genannt.

Die Segmentierung ist damit eng mit der Klassifikation verknüpft, und die Kategorisierung der Segmentierungsstrategie richtet sich danach, wie die “Intelligenz” zwischen Segmentierung und Klassifikation verteilt ist. So geht bei der expliziten Segmentierung schon ein Großteil des Wissens im Segmentierungsschritt selbst ein. Hier wird mit aufwendigen Verfahren versucht, einzelne Buchstaben oder Strokes zu segmentieren, sodass die Signalabschnitte vor der Klassifikation eindeutig festgelegt sind. Bei der impliziten Segmentierung steckt dagegen die gesamte “Intelligenz” in der Klassifikation. Daneben wird auch häufig ein Ansatz gewählt, der weder eine rein explizite noch eine rein implizite Segmentierungsstrategie aufweist. Bei dieser als *Übersegmentierung* bezeichneten Strategie wird das Signal ähnlich wie bei der expliziten Segmentierung vor der Klassifikation in Abschnitte zerlegt, die nun jedoch nicht eins-zu-eins Buchstaben entsprechen müssen, sondern eher als Buchstabenhypothesen aufzufassen sind. Damit kann ein Buchstabe durchaus in mehrere Segmente zerfallen. Die beste Segmentierung wird dann wie bei der impliziten Segmentierung im Zuge der Klassifikation durch Bewertung der Segmentierungshypothesen ermittelt.

3.4.1 Explizite Segmentierung

Das Ziel der expliziten Segmentierung ist die Aufspaltung des Signals in Segmente, die im anschließenden Klassifikationsschritt eins-zu-eins auf Buchstaben oder Strokes abgebildet werden. Die Segmentierung ist damit schon vor der Erkennung eindeutig festgelegt. Die Klassifikation der Basiseinheiten kann somit isoliert voneinander erfolgen, sodass eine Vielzahl unterschiedlicher Mustererkennungsmethoden eingesetzt werden kann. Die isolierte Klassifikation hat jedoch den Nachteil, dass Kontextinformationen, die z.B. in Form eines Lexikons oder statistischen Sprachmodellen vorliegen, erst in einem Nachverarbeitungsschritt eingesetzt werden können. Die größte Schwierigkeit bei der expliziten Segmentierung ist allerdings die wechselseitige Abhängigkeit von Erkennung und Segmentierung: Um einen Buchstaben zu erkennen, muss man ihn erst korrekt segmentiert haben, die Segmentierung ist jedoch erst möglich, wenn man den Buchstaben erkannt hat. Auf dieses Paradoxon wurde zum ersten Mal von Sayre hingewiesen [Say73].

Werden Textzeilen verarbeitet, die mehrere Wörter enthalten, so werden im ersten Schritt der expliziten Segmentierung die einzelnen Wörter der Schriftzeile extrahiert. Erst im darauffolgenden Schritt wird dann die Segmentierung der Worthypothesen in Buchstaben bzw. Strokes vorgenommen.

Wortsegmentierung der Schriftzeile

Die Segmentierung einer Schriftzeile in Wörter ist bei uneingeschränkter Handschrift aufgrund der hohen Variabilität der Wortabstände im Vergleich zu maschinengeschriebener Schrift ungleich schwieriger. Die Verfahren, die sowohl im online als auch im offline Bereich überwiegend zur Wortsegmentierung eingesetzt werden, basieren auf den räumlichen Abständen benachbarter Zusammenhangskomponenten der Schrift. Dabei wird davon ausgegangen, dass die Abstände der Komponenten innerhalb eines Wortes kleiner sind als die Abstände zwischen Wörtern. Zur Berechnung der Abstände können z.B. die folgenden Distanzmaße eingesetzt werden [Sen94a, Mar00a, Yan98]:

1. Minimaler euklidischer Abstand zwischen den Zusammenhangskomponenten.

2. Minimaler horizontaler Abstand zwischen den umschließenden Rechtecken (*Bounding Box*) der Zusammenhangskomponenten.

3. Minimaler horizontaler Abstand zwischen den konvexen Hüllen der Zusammenhangskomponenten.

4. Minimaler horizontaler Abstand zwischen den Zusammenhangskomponenten in der Mittelzone der Schrift.

Sind für alle benachbarten Zusammenhangskomponenten die Distanzen bestimmt, so müssen diese den Klassen *Abstände zwischen Wörtern* und *Abstände innerhalb*

von Wörtern zugeordnet werden. Üblicherweise wird diese Zuordnung anhand eines Schwellwerts vorgenommen. In [Yan98] basiert dieser Schwellwert beispielsweise auf der geschätzten Buchstabenbreite. Damit wird eine Wortgrenze hypothetisiert, wenn der Abstand zwischen benachbarten Zusammenhangskomponenten größer als der Schwellwert ist.

Die Wortsegmentierung nur auf Grund der Abstände zwischen benachbarten Zusammenhangskomponenten durchzuführen, ist bei uneingeschränkter Handschrift oftmals jedoch fehleranfällig. In [Par02] wird deshalb ein Verfahren vorgeschlagen, das die Schriftcharakteristik stärker berücksichtigt. Anstatt der Extraktion von Zusammenhangskomponenten wird hier eine Segmentierung der Zeile in Buchstabenhypothesen vorgenommen. Die Wortsegmentierung erfolgt dann anhand des Abstands zwischen den Schwerpunkten der Pixelverteilungen zweier benachbarter Buchstabenhypothesen.

Auch die Verfahren, die in [Kim98, Kim99] beschrieben werden, führen die Wortsegmentierung anhand vorsegmentierter Buchstabenhypothesen durch. In diesem Fall wird auf Basis zweier benachbarter Buchstabenhypothesen ein acht-dimensionaler Merkmalsvektor berechnet, der die Pixelverteilung innerhalb der Bounding Boxen der Buchstaben beschreibt. Die Klassifikation des Merkmalsvektors, d.h. die Entscheidung, ob dieser eine Wortgrenze beschreibt oder nicht, erfolgt mit Hilfe eines neuronalen Netzes.

In [Sen94a] wird ein Verfahren beschrieben, das die räumliche Abstandsinformation mit der Detektion von Satzzeichen kombiniert, um eine zuverlässigere Wortsegmentierung zu erzielen. Die Satzzeichen (Punkt und Komma) werden dabei anhand geometrischer Formmerkmale detektiert. Um eine Unterscheidung der Satzzeichen von i-Punkten und Apostrophs zu ermöglichen, werden weitere räumliche Merkmale verwendet, die die Abstände der Zeichen zu den Schriftgrundlinien beschreiben.

Buchstabensegmentierung im offline Bereich

Die Methoden zur expliziten Segmentierung der Schrift in Buchstaben haben ihren Ursprung in der Verarbeitung maschinengeschriebener Dokumente. Hier kann die Segmentierung oftmals auf einfachen Verfahren basieren, wie z.B. der Detektion von Leerräumen zwischen den Buchstaben oder der Analyse der vertikalen Projektion der Schrift.

Eine weitere Möglichkeit zur Buchstabensegmentierung, die besser für fließende Handschrift geeignet ist, besteht in der Analyse der Wortkontur. Die Segmentgrenzen werden dabei zwischen aufeinanderfolgenden Minima bzw. Maxima der Kontur gesucht, wobei hierfür oftmals aufwendige Heuristiken zum Einsatz kommen (siehe z.B. [Cas96]). Akzeptable Resultate lassen sich bei diesen Verfahren vor allem dann erzielen, wenn die Schrift bezüglich Orientierung und Neigung normalisiert wurde und Vorwissen über das zu segmentierende Signal vorliegt (z.B. Anzahl der Zeichen bei Postleitzahlen).

Allgemein lässt sich sagen, dass die explizite Segmentierung in Buchstaben nur bei maschinengeschriebener Schrift oder bei Blockschrift eines kooperativen Schreibers durchführbar ist. Bei uneingeschränkter Handschrift ist die Isolation von Buchstaben vor der Klassifikation nicht möglich. Aus diesem Grund wird heute von einer rein expliziten Segmentierung abgesehen, sondern verstärkt Verfahren eingesetzt, die die endgültige Segmentierungsentscheidung erst im Zuge der Klassifikation treffen.

Strokesegmentierung im online Bereich

Ähnlich wie im offline Bereich wird auch bei der online Erkennung uneingeschränkter Handschrift die explizite Segmentierung des Signals in Buchstaben vermieden. Stattdessen basiert die Erkennung oftmals auf Strokes, also den elementaren Schreibbewegungen [Sch93]. Diese Strokes lassen sich im Gegensatz zu kompletten Buchstaben sehr robust unter Berücksichtigung der Schreibdynamik aus dem Signal extrahieren.

Ein Stroke ist durch einen glockenförmigen Verlauf der Geschwindigkeit charakterisiert (siehe Abschnitt 2.1.2), sodass als Segmentierungskriterium häufig die lokalen Minima der Schreibgeschwindigkeit herangezogen werden [Sch90]. In [Gue98] wird ein *Analyse-durch-Synthese* Ansatz vorgeschlagen, in dem die Strokesegmentierung mit Hilfe des *Delta-Log-Normal-Modells* (siehe Seite 9) der Handschriftgenerierung durchgeführt wird. Mit diesem modellbasierten Ansatz zur Segmentierung können auch im Signal “versteckte” Strokes, die durch eine zeitliche Überlagerung hervorgerufen wurden, extrahiert und anschließend für die Klassifikation genutzt werden.

3.4.2 Implizite Segmentierung

Die explizite Buchstabensegmentierung, d.h. die vor der Klassifikation durchgeführte, eindeutige Unterteilung des Signals in Abschnitte, die eins-zu-eins Buchstaben entsprechen, ist bei fließender Handschrift nicht möglich. Im holistischen Erkennungsansatz kann dieser Segmentierungsschritt vermieden werden, indem ganze Wörter als Basiseinheiten für die Erkennung verwendet werden. Die holistischen Verfahren sind jedoch stark von einem vorgegebenen Lexikon abhängig, sodass analytische Verarbeitungsmethoden mehr Flexibilität bieten.

Möchte man frühe, unumkehrbare Segmentierungsentscheidungen vermeiden und dennoch auf die Flexibilität analytischer Verfahren nicht verzichten, so müssen die Prozesse der Segmentierung und Erkennung stärker miteinander verschränkt werden. Man spricht dann von impliziter bzw. erkennungsbasierter Segmentierung, wenn die endgültige Segmentierung erst im Zuge der Klassifikation ermittelt wird. Vor der Klassifikation findet dabei keinerlei Segmentierung in symbolische Einheiten wie z.B. Buchstaben oder Strokes statt, vielmehr wird eine systematische Unterteilung des Signals vorgenommen. Da bei der impliziten Segmentierung keine isolierte Klassifikation der Basiseinheiten vorgenommen wird, besteht die Möglichkeit, Kontextinformationen schon während der Erkennung gewinnbringend zu nutzen.

Offline Verfahren

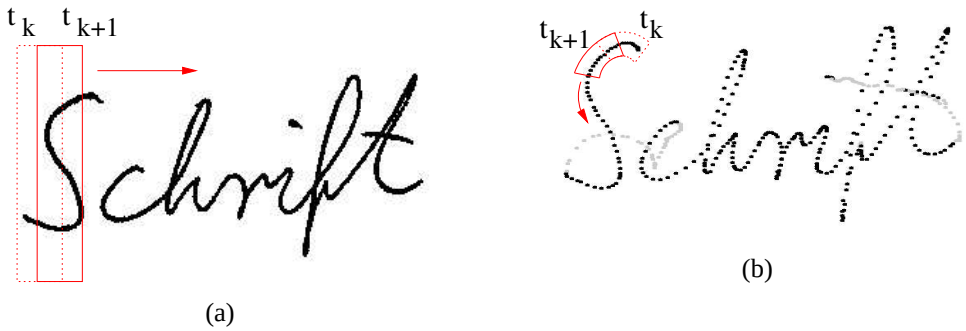
Eine einfache Möglichkeit der systematischen Unterteilung des Schriftbildes bei neigungs- und orientierungskorrigierter Schrift bieten sogenannte *Sliding-Window Verfahren* (siehe u.a. [Kal93, Cho95, Sch97, Mar98, Bra99, Vin00]). Dabei wird ein Fenster vorgegebener Breite von links nach rechts über die Zeile geschoben, wobei die Höhe des Fensters üblicherweise der der Zeile entspricht. Die Breite des Fensters variiert von System zu System und hängt von der geschätzten Buchstabenbreite der Schrift ab. Aus der Wahl der Fensterbreite und der Schrittweite, mit der das Fenster verschoben wird, gehen somit die potentiellen Segmentierungsgrenzen hervor. Die Abbildung 3.19 veranschaulicht die Sliding-Window Technik.

Online Verfahren

Analog zu den Sliding-Window Verfahren im offline Bereich, bei denen einzelne Bildspalten oder Gruppen aufeinanderfolgender Spalten extrahiert werden, werden im online Bereich die einzelnen Trajektorienpunkte oder Gruppen zeitlich benachbarter Trajektorienpunkte als Segmente definiert (siehe u.a. [Nat93, Mak94, Sta94, Sch95a, Kos97]). Im online Bereich sind Systeme, die auf impliziter Segmentierung basieren, weit verbreitet – nicht zuletzt deshalb, weil diese Verarbeitungsstrategie von der Spracherkennung her bekannt ist und vorhandene Spracherkennungssysteme mit relativ geringem Aufwand auch für die online Handschrifterkennung eingesetzt werden können [Sta94].

3.4.3 Übersegmentierung

Einige Systeme zur Handschrifterkennung weisen eine Segmentierungsstrategie auf, die weder als rein explizit noch als rein implizit bezeichnet werden kann, sondern mit der eher ein Mittelweg zwischen diesen beiden extremen Paradigmen beschritten wird. Dieser Ansatz ist dadurch gekennzeichnet, dass vor der Klassifikation eine



(a) (b)

Abbildung 3.19: Systematische Unterteilung des Schriftbildes (a) bzw. der Stifttrajektorie (b) durch *Sliding Window* Verfahren

Segmentierung in *Buchstabenhypothesen* durchgeführt wird, die im Vergleich zur expliziten Segmentierung jedoch nicht unbedingt eins-zu-eins Buchstaben entsprechen müssen. Vielmehr wird eine Übersegmentierung der Schrift vorgenommen, sodass ein Buchstabe durchaus in mehrere Segmente zerfallen kann, ein Segment jedoch nicht mehrere Buchstaben umfassen sollte. Die segmentierten Buchstabenhypothesen werden dann ähnlich wie bei der rein impliziten Segmentierung im Klassifikationsschritt kombiniert, sodass die endgültige Segmentierung in Buchstaben auch erst im Zuge der Klassifikation ermittelt wird.

Im Gegensatz zur expliziten Segmentierung in Buchstaben werden somit weniger strenge Anforderungen an die Segmentierung gestellt, sodass die Segmentierungshypothesen häufig anhand einfacher, robuster Kriterien aufgestellt werden können. Der Vorteil gegenüber der rein impliziten Segmentierung liegt darin, dass die Anzahl hypothetisierter Segmentgrenzen deutlich reduziert ist und damit der Rechenaufwand im Klassifikationsschritt sinkt [Dun92].

Offline Verfahren

Die Übersegmentierung der Schrift ist Kennzeichen zahlreicher Systeme zur offline Handschrifterkennung (siehe u.a. [Boz89, Gad95, Kim97, EY99]). Bei der Verarbeitung fließender Handschrift basieren die Segmentgrenzen dabei üblicherweise auf der Detektion von Ligaturen, also den Verbindungen zwischen den Buchstaben.

In [Kim97] wird ein Verfahren vorgeschlagen, das Ligaturen anhand des vertikalen Abstands zwischen der oberen und unteren Wortkontur bestimmt. Ist dieser Abstand geringer als die mittlere Strichbreite der Schrift, so wird eine Ligatur und dementsprechend die zugehörige Segmentgrenze hypothetisiert. Zusätzlich werden Heuristiken auf Basis konkaver bzw. konvexer Abschnitte der oberen bzw. unteren Schriftkontur angewendet, um die Zahl potentieller Segmentierungshypothesen einzuschränken. So erfordern die durch den nachfolgenden Klassifikationsschritt vorgegebenen Randbedingungen an die Segmentierung, dass ein Buchstabe in maximal vier Segmente zerfallen darf, ein Segment jedoch keinesfalls mehrere Buchstaben umfassen sollte.

Häufig dienen auch die lokalen vertikalen Minima der Schriftkontur als Ausgangspunkt für die Segmentierung. In [Boz89, EY99] wird beispielsweise in der Umgebung des jeweiligen Konturminimums nach einer vertikalen Segmentgrenze gesucht, die mehrfache Kreuzungen des Schriftzuges und somit das Durchschneiden von Schleifen vermeidet.

Während die oben vorgestellten Verfahren voraussetzen, dass die Schrift bereits bezüglich Orientierung und Neigung normalisiert ist, sodass die Segmentgrenzen als vertikale Geraden beschrieben werden können, wird in [Ari02] ein Verfahren zur Segmentierung von nicht-normalisierter Schrift vorgeschlagen. Dabei wird die Bestimmung einer nichtlinearen Segmentgrenze innerhalb einer hypothetisierten Segmentierungsregion als Pfadsuche realisiert, sodass die Segmentierungsregion von oben nach unten durchtrennt wird und der resultierende Pfad bezüglich einer Kostenfunktion minimal ist (siehe auch [Lee96]). Die Kostenfunktion ist dementsprechend so formuliert,

dass der Segmentierungspfad nicht auf dem Schriftzug verläuft und diesen möglichst selten kreuzt. Die Pfadsuche kann dabei effizient mit Hilfe der dynamischen Programmierung implementiert werden.

Online Verfahren

Im online Bereich basiert der überwiegende Teil der Systeme auf der impliziten Segmentierung der Stifttrajektorie. Daneben gibt es jedoch auch einige Systeme, die auf der Segmentierung des Signals in einzelne Strokes basieren, da sich diese robust aus dem Signal extrahieren lassen. Ob diese Systeme in die Kategorie "explizite Strokesegmentierung" oder "Übersegmentierung von Buchstaben" einzuordnen sind, hängt davon ab, wie im nachfolgenden Klassifikationsschritt die Buchstaben anhand der Strokes hypothetisiert werden.

Gibt es eine direkte Zuordnung der Strokes zu den Buchstaben, sodass die Erkennung eines Buchstabens direkt von der vollständigen Erkennung der Strokesequenz abhängig ist, so ist dieser Ansatz durch explizite Strokesegmentierung charakterisiert. Das Hauptaugenmerk liegt somit auf der Erkennung von Strokes. Die Buchstaben werden nicht probabilistisch sondern auf symbolischer Ebene aus den Strokes zusammengesetzt [Sch90, Sch93].

Basiert die Erkennung von Buchstaben dagegen nicht auf der symbolischen Zuordnung vorher erkannter Strokes, sondern werden stattdessen statistische Buchstabenmodelle verwendet, sodass die Kombination von Strokes implizit mittels eines probabilistischen Verfahrens vorgenommen wird, so kann man den Segmentierungsschritt eher als Übersegmentierung von Buchstaben auffassen. Beispiele hierfür sind die in [Bei95, Dol97] beschriebenen Verfahren. Dabei wird das Signal anhand der Minima der vertikalen Geschwindigkeitskomponente in Strokes zerlegt. Die Erkennung erfolgt dann mit Hilfe von Buchstabenmodellen in einem probabilistischen Rahmen.

3.5 Merkmalsextraktion

Anhand des vorverarbeiteten und gegebenenfalls segmentierten Eingabesignals werden in dieser Phase der Verarbeitung Merkmale bestimmt, die dann im nachfolgenden Schritt zur Klassifikation verwendet werden. Die Eingabemuster liegen dabei zum Zeitpunkt der Merkmalsextraktion i.d.R. noch in der gleichen Repräsentation vor wie das aufgenommene Signal, d.h. als Bilder bei den offline Systemen und als Trajektorien bei den online Systemen. Mit der Merkmalsextraktion wird das Eingabesignal in eine Repräsentation überführt, die zugleich möglichst diskriminativ, robust und kompakt ist, sodass damit die Voraussetzungen für eine erfolgreiche Klassifikation des Eingabemusters gegeben sind.

Die Forderung, dass die extrahierten Merkmale möglichst diskriminativ sein sollen, bedeutet, dass sich die Merkmale stark unterscheiden, wenn sie von Eingabemustern stammen, die unterschiedlichen Klassen zuzuordnen sind. Gleichzeitig jedoch sollten

sich die Merkmale wenig unterscheiden, wenn sie von Eingabemustern stammen, die derselben Klasse zuzuordnen sind. Diese Eigenschaft führt also dazu, dass sich die Klassengebiete im Merkmalsraum gut separieren lassen und innerhalb der Klassen eine relativ kleine Streuung der Merkmale vorliegt.

Um eine kleine Streuung der Merkmale innerhalb derselben Klasse zu erreichen, ist es erforderlich, dass die Merkmale robust gegenüber Schwankungen der zur selben Klasse gehörenden Eingabemuster sind. Weisen die Daten z.B. kleine Störungen auf, so sollte dies keinen oder nur geringen Einfluss auf die resultierenden Merkmale haben. Der Prozess der Merkmalsextraktion ist somit stark von der durchgeführten Vorverarbeitung des Signals abhängig. Wird beispielsweise keinerlei Vorverarbeitung in Bezug auf Größen- oder Neigungsnormalisierung der Schrift vorgenommen, so müssen diesbezüglich invariante Merkmale bestimmt werden.

Eine weitere wichtige Eigenschaft der Merkmalsextraktion ist die möglichst kompakte Repräsentation des Eingabesignals. Da bei hochdimensionalen Merkmalsvektoren auch entsprechend mehr Parameter des Klassifikators geschätzt werden müssen, ist eine kompakte Repräsentation insbesondere dann von Vorteil, wenn die Trainingsdaten für den Klassifikator nur in sehr begrenztem Umfang vorhanden sind.

Im Gegensatz zur Spracherkennung hat sich im Bereich der Handschrifterkennung bisher jedoch kein Merkmalssatz etabliert. Vielmehr findet man beinahe bei jedem System einen unterschiedlichen Satz von Merkmalen vor. Da der Versuch einer vollständigen Beschreibung der verwendeten Merkmale damit zum Scheitern verurteilt ist, wird im folgenden lediglich eine Kategorisierung der Merkmale durchgeführt. Anschließend werden mit der Hauptkomponenten- und linearen Diskriminanzanalyse zwei verbreitete Verfahren zur Merkmalsoptimierung vorgestellt, die eine möglichst kompakte und diskriminative Repräsentation erzeugen.

3.5.1 Merkmale: High-level vs. low-level

Eine in der Literatur häufig vorgenommene Kategorisierung der zur Handschrifterkennung genutzten Merkmale richtet sich danach, ob die Merkmale *global*, also anhand des gesamten Wortes extrahiert werden, oder ob die Merkmalsextraktion *lokal* auf Basis kürzerer Schriftsegmente erfolgt [Hu97, Hu00, Con00]. Im ersten Fall spricht man dann von *high-level* Merkmalen, während die segmentweise extrahierten Merkmale als *low-level* Merkmale bezeichnet werden[5].

High-level Merkmale

Die Extraktion von high-level Merkmalen beruht auf der Detektion von strukturellen Elementen der Schrift, den sogenannten *Primitiven*. Diese Elementarmuster der Schrift sind beispielsweise Oberlängen, Unterlängen, Schleifen, Kreuzungspunkte oder Endpunkte. Das zu erkennende Wort wird damit als komplexes Muster betrachtet, das

[5] In [Vin02a] wird die segmentweise Merkmalsextraktion noch weiter in *medium-level* und *low-level* unterschieden, je nachdem, ob die Segmente ganze Buchstaben oder nur Buchstabenteile umfassen.

aus Primitiven zusammengesetzt ist, die durch Nachbarschaftsrelationen untereinander verknüpft sind. Diese Art der Repräsentation findet vor allem im Bereich der *strukturellen* bzw. *syntaktischen* Mustererkennung Anwendung, bei der eine Analogie zwischen der Struktur eines Musters und der Syntax einer Sprache gezogen wird [Jai00]. Die Muster werden dabei als Sätze einer Sprache aufgefasst, die anhand grammatikalischer Regeln gebildet werden, wobei die Primitive als Alphabet und die Relationen als Regeln der Grammatik betrachtet werden.

Die Vorteile der high-level Merkmale sind ihre hohe Aussagekraft und guten Diskriminanzeigenschaften. Weiterhin ist die Detektion von Primitiven unabhängig von der Orientierung, Neigung und Größe der Schrift. Diese Robustheit gilt jedoch nicht für uneingeschränkte Handschrift unterschiedlicher Schreiber, wenn also eine hohe Schriftvariabilität vorliegt, sodass die Muster einer Klasse stark in ihrer Form voneinander abweichen. In diesem Fall ist die Detektion von Primitiven äußerst fehleranfällig, sodass zumindest im Bereich schreiberunabhängiger Systeme von einer ausschließlichen Verwendung von high-level Merkmalen zur Erkennung abgesehen wird.

Low-level Merkmale

Demgegenüber steht eine Vielzahl von Systemen, die low-level Merkmale für die Schrifterkennung nutzen. Diese Merkmale werden lokal anhand der im Segmentierungsschritt ermittelten Signalabschnitte extrahiert und pro Segment zu einem Merkmalsvektor fester Größe zusammengefasst. Die daraus resultierende Sequenz von Merkmalsvektoren kann dann beispielsweise mit Hilfe statistischer Methoden klassifiziert werden. Im Gegensatz zu Primitiven sind die low-level Merkmale für sich genommen weitaus weniger informativ, sie lassen sich jedoch robust aus dem Signal bestimmen.

Offline Bereich

Die Verfahren, die im offline Bereich low-level Merkmale extrahieren, basieren zu einem Großteil auf Bildsegmenten, die in binarisierter Form vorliegen. Oftmals kommen dann einfache *geometrische Merkmale* zum Einsatz, um die lokale Pixelverteilung zu beschreiben. In [Mar00a, Mar00b] werden beispielsweise im betrachteten Bildfenster u.a. die folgenden Merkmale bestimmt: Die Position und Orientierung der oberen bzw. unteren Schriftkontur, die Anzahl der Schriftpixel zwischen der oberen und unteren Kontur und die Anzahl der schwarz-weiß (Schrift-Hintergrund) Übergänge in vertikaler Richtung.

Eine andere Möglichkeit zur Beschreibung der lokalen Pixelverteilung ist das sogenannte *Zoning* (dt. Unterteilung in Zonen). Dabei wird das jeweilige Segment horizontal und vertikal in Zonen vorgegebener Größe eingeteilt und anschließend in jeder Zone die Anzahl der Schriftpixel bestimmt. Die relativen Häufigkeiten der Schriftpixel

in den einzelnen Zonen werden dann als Merkmale verwendet. Beispiele hierfür finden sich u.a. in [Che94, Bra99, Vin00]).

In [Che94] wird das betrachtete Bildsegment darüberhinaus durch *Kennlinien* beschrieben. Den Ausgangspunkt bilden hierbei horizontale und vertikale Geraden, die durch den Schwerpunkt des Bildsegments verlaufen. Die Merkmale basieren dann auf der Anzahl der Schnittpunkte zwischen den Geraden und dem Schriftzug. Ein ähnlicher Ansatz wird durch die Verwendung von Histogrammen verfolgt. In [Kav02] werden beispielsweise neben dem horizontalen und vertikalen Projektionshistogramm des Segments auch sogenannte *radiale Histogramme* verwendet. Dabei werden die Schriftpixel entlang einer Geraden vom Mittelpunkt bis zum Rand des Segments gezählt, wobei sich die Orientierung der Geraden schrittweise ändert, sodass ein voller Kreis beschrieben wird.

Eine weitere Möglichkeit zur Charakterisierung der Pixelverteilung bietet die Repräsentation durch *Momente* [Che94]. Werden die Koordinaten der Schriftpixel mit x und y bezeichnet, so ist das Moment m_{pq} definiert durch

$$m_{pq} = \frac{1}{N} \sum_{i=1}^{N} x_i^p y_i^q, \tag{3.34}$$

wobei N die Gesamtzahl der Schriftpixel bezeichnet. Mit den Schwerpunkten $\bar{x}_i = \frac{1}{N} \sum_{i=1}^{N} x_i$ und $\bar{y}_i = \frac{1}{N} \sum_{i=1}^{N} y_i$ ergibt sich beispielsweise das Zentralmoment μ_{pq} zu

$$\mu_{pq} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x}_i)^p (y_i - \bar{y}_i)^q. \tag{3.35}$$

Auf Basis der Zentralmomente lässt sich schließlich bei geeigneter Normierung ein Satz von Merkmalen angeben, der rotations-, translations- und größeninvariant ist.

Neben obigen Verfahren, die binarisierte Bildsegmente voraussetzen, gibt es außerdem Methoden, die die Merkmalsextraktion anhand des skelettierten Schriftbildes durchführen. In [Bun95] wird das Skelett der Schrift als Graph repräsentiert, wobei die singulären Punkte des Skeletts, also die End- und Kreuzungspunkte, die Graphknoten und die regulären Skelettpunkte die Kanten des Skelettgraphen darstellen. Anhand dieser Kanten, die in einer definierten Reihenfolge aus dem Graphen extrahiert werden, wird dann eine Reihe von Merkmalen berechnet, darunter z.B. die Position der Kante relativ zu den Referenzlinien der Schrift.

In [Kim97] basiert die Merkmalsextraktion auf einer Kettencode-Darstellung der Schriftkontur. Das Bildsegment wird dabei ähnlich wie beim Zoning in Teilbereiche zerlegt, wobei die Häufigkeiten der auftretenden Kettencode-Richtungen in den einzelnen Teilbereichen den Merkmalsvektor bilden.

Die hier vorgestellten Verfahren, die im offline Bereich zur Merkmalsextraktion eingesetzt werden, stellen nur eine Auswahl verschiedenster Methoden dar, die sich in der Literatur finden lassen. Neben den beschriebenen Verfahren existieren noch eine Reihe weiterer, und auch Kombinationen unterschiedlichster low-level Merkmale werden oftmals verwendet [Che94, Mar00b].

Online Bereich

Im online Bereich ist die als Sequenz von Koordinatenpunkten vorliegende Stifttrajektorie die Grundlage für die Extraktion von low-level Merkmalen. Bei einer Vielzahl von Systemen, die auf impliziter Segmentierung basieren, wird dabei für jeden einzelnen Koordinatenpunkt der Trajektorie ein zugehöriger Merkmalsvektor berechnet, der eine möglichst gute lokale Charakterisierung der Stifttrajektorie liefern soll.

Oftmals basieren die Merkmale auf den lokalen Richtungs- und Krümmungseigenschaften der Stifttrajektorie. In den Systemen [Sta94, Mak94] werden beispielsweise Merkmalsvektoren verwendet, die lediglich aus dem Winkel, den zwei benachbarte Trajektorienpunkte gegenüber der Horizontalen einschließen, und der Differenz dieses Winkels bestehen. In [Nat93] wird u.a. die Position und die Krümmung am jeweiligen Koordinatenpunkt verwendet. Um einen größeren zeitlichen Kontext zu berücksichtigen, werden darüberhinaus in einigen Systemen die Merkmale vorangegangener Trajektorienpunkte in den aktuellen Merkmalsvektor integriert. Dies kann beispielsweise dadurch geschehen, dass die Differenz der zum vorangegangenen Zeitpunkt extrahierten Merkmale und der zum aktuellen Zeitpunkt extrahierten Merkmale gebildet wird.

Ein weiteres Merkmal, das in online Systemen häufig eingesetzt wird, beschreibt, ob der Stift auf der Schreiboberfläche aufgesetzt ist (pen-down) oder von der Oberfläche abgehoben wurde (pen-up). Üblicherweise wird dazu ein *binäres* Merkmal verwendet, sodass auch bei Digitalisiertabletts, die den Anpressdruck des Stifts bzw. in pen-up Phasen den Abstand von der Schreiboberfläche in einem Intervall messen können, die entsprechenden Messwerte auf die Zustände pen-up bzw. pen-down abgebildet werden (u.a. in [Kos97, Rig98, Jae01]).

In einigen Systemen wird darüberhinaus ein spezielles Merkmal zur Beschreibung diakritischer Zeichen verwendet. Dabei werden zuerst die sogenannten *delayed strokes* (siehe Seite 53) in der Vorverarbeitungsphase mittels einfacher Heuristiken detektiert. Die delayed Strokes werden dann aus dem Eingabesignal entfernt und an ihrer Stelle werden die Koordinatenpunkte, die zu den Zeichen gehören, die durch den delayed Stroke komplettiert wurden, mit einem speziellen Merkmal, dem *hat-feature*, versehen.

Mit dem *hat-feature* werden damit räumlich benachbarte Koordinatenpunkte in Beziehung zueinander gesetzt, die einen großen zeitlichen Abstand zueinander haben. Der gleiche Ansatz wird in einigen Systemen mit den *Kontext-Bitmap* Merkmalen verfolgt. Kontext-Bitmaps sind Bilder, die entlang der Trajektorie geschoben werden, wobei das Zentrum der Kontext-Bitmap der jeweilige Koordinatenpunkt ist. Bei einer entsprechendem Größe der Bitmap können damit räumliche Kontextinformationen in die lokale Merkmalsrepräsentation eingebracht werden. Die Merkmale, die anhand der Kontext-Bitmap berechnet werden, basieren dabei häufig auf einfachem Zoning. Beispiele für die Verwendung von Kontext-Bitmaps finden sich u.a. in [Kos97, Rig98, Jae01].

Integration von high-level und low-level Merkmalen

Um gleichzeitig die guten Diskriminanzeigenschaften der globalen high-level Merkmale und die Robustheit der lokalen low-level Merkmale auszunutzen, werden in einigen Systemen beide Merkmalsarten kombiniert [Sen98, EY99, Hu00].

Eine Möglichkeit der Kombination von low-level und high-level Merkmalen im Bereich der online Handschrifterkennung wird in [Hu97] beschrieben. In diesem System werden neben vier low-level Merkmalen, die im wesentlichen auf der vertikalen Position, der Richtung und der Krümmung lokaler Trajektorienabschnitte basieren, ebenfalls drei high-level Merkmale mit in den Merkmalsvektor eingebunden. Dazu werden im ersten Schritt die high-level Merkmale, in diesem Fall sind dies Schleifen, Kreuzungspunkte und sogenannte Cusps (dt. etwa Zacken, Bereiche hoher Krümmung) auf der Trajektorie lokalisiert. Auf Basis der Lokalisationsergebnisse wird dann im zweiten Schritt die Integration der high-level Merkmale wie folgt vorgenommen: Liegt beispielweise der aktuell zur Merkmalsextraktion betrachtete Trajektorienpunkt auf einer Schleife des Schriftzuges, so wird das Vorhandensein dieses high-level Merkmals vermerkt, indem an die entsprechende Position des lokalen Merkmalsvektors eine Eins eingetragen wird. In ähnlicher Weise wird bei Kreuzungspunkten und Cusps verfahren. Da diese gegenüber Schleifen jedoch auf der Trajektorie eher vereinzelt vorkommen, wird anstelle eines binären Flags der räumliche Abstand vom aktuellen Trajektorienpunkt zum nächstgelegene Kreuzungspunkt bzw. Cusp als Merkmal verwendet.

3.5.2 Merkmalstransformationen

Die Entscheidung, welche Merkmale aus dem Eingabemuster extrahiert werden sollen, wird üblicherweise anhand heuristischer Kriterien vorgenommen. Hierbei spielen vor allem das Expertenwissen und die Erfahrung des Systementwicklers eine maßgebliche Rolle. Häufig wird dabei zuerst ein Basissatz von Merkmalen zugrundegelegt, der schrittweise durch neue Merkmale erweitert wird, um bessere Klassifikationsleistungen zu erreichen. Bei diesem Vorgehen kann es jedoch leicht dazu kommen, dass die Komponenten der u.U. hochdimensionalen Merkmalsvektoren miteinander korreliert sind. Um kompakte und unkorrelierte Merkmalsvektoren zu erhalten, ist somit eine Optimierung des Merkmalssatzes erforderlich. Gängige Verfahren sind vor allem die Hauptkomponentenanalyse und die lineare Diskriminanzanalyse, die u.a. in [Fin03], Seite 139ff und [Sch95b], Seite 113ff, ausführlich beschrieben sind.

Hauptkomponentenanalyse

Mit der Hauptkomponentenanalyse kann eine Dekorrelation und zusätzlich eine Dimensionsreduktion der Merkmale erreicht werden. Der Ausgangspunkt für die Hauptkomponentenanalyse ist das Streuungsverhalten der Merkmalsvektoren, welches durch

die Streumatrix $\boldsymbol{S}_x$ beschrieben wird.

$$\boldsymbol{S}_x = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \tag{3.36}$$

In obiger Formel bezeichnet $\bar{\boldsymbol{x}}$ den Mittelwert der Merkmalsvektoren.

$$\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = \bar{\boldsymbol{x}}, \quad \dim(\boldsymbol{x}) = D \tag{3.37}$$

Sind die Komponenten der Merkmalsvektoren korreliert, so besitzt die Streumatrix $\boldsymbol{S}_x$ nicht Diagonalgestalt. Das Ziel der Hauptkomponentenanalyse, die Dekorrelation der Merkmale, entspricht der Anwendung einer Transformation $\boldsymbol{\Phi}$ auf die mittelwertbereinigten Merkmalsvektoren

$$\boldsymbol{y} = \boldsymbol{\Phi}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{3.38}$$

sodass die Streumatrix $\boldsymbol{S}_y$ der transformierten Merkmale Diagonalgestalt aufweist:

$$\boldsymbol{S}_y = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{y}_i\boldsymbol{y}_i^T = \boldsymbol{\Phi}^T\boldsymbol{S}_x\boldsymbol{\Phi} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix}. \tag{3.39}$$

Durch Anwendung der Transformation wird eine Rotation des Merkmalsraums durchgeführt, sodass die Richtung maximaler Varianz mit der ersten Koordinatenachse übereinstimmt.

Die Herleitung der Transformation $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_D]$ zur Diagonalisierung der symmetrischen Matrix $\boldsymbol{S}_x$ erfolgt durch Lösung des folgenden Eigenwertproblems:

$$\boldsymbol{S}_x\boldsymbol{\phi}_k = \lambda_k\boldsymbol{\phi}_k\,, \qquad k = 1, \ldots, D. \tag{3.40}$$

Demnach werden die Spaltenvektoren der Transformationsmatrix $\boldsymbol{\Phi}^T$ durch die normierten Eigenvektoren $\boldsymbol{\phi}_k$ der Streumatrix $\boldsymbol{S}_x$ gebildet, die zu den jeweiligen Eigenwerten λ_k gehören.

Eine Dimensionsreduktion der Merkmale wird erreicht, wenn die Transformationsmatrix nicht aus allen D Eigenvektoren aufgebaut wird, sondern nur eine Teilmenge betrachtet wird. Sind die Eigenvektoren/-werte so numeriert, dass die Eigenwerte eine absteigende Folge bilden, so werden die ersten d Eigenvektoren berücksichtigt, die somit zu den d größten Eigenwerten gehören.

$$\tilde{\boldsymbol{\Phi}} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_d], \quad d < D. \tag{3.41}$$

Da die Eigenwerte die Varianzen der transformierten Merkmale bezeichnen, wird durch die Transformation

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{\Phi}}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{3.42}$$

eine Abbildung der Merkmalsvektoren in den d-dimensionalen Unterraum durchgeführt, der den größten Varianzanteil umfasst.

Die Hauptkomponentenanalyse optimiert somit die Merkmalsrepräsentation, indem anhand der Streuungseigenschaften der Merkmale eine Dekorrelation und ggf. eine Dimensionsreduktion durchgeführt wird. Es werden dabei alle Merkmalsvektoren unabhängig von der Klasse betrachtet, zu der die Vektoren jeweils zuzuordnen sind, sodass die Diskriminanzeigenschaften der Merkmalsrepräsentation nicht unbedingt verbessert werden.

Lineare Diskriminanzanalyse

Neben der Dekorrelation und Dimensionsreduktion ist das vorrangige Ziel der linearen Diskriminanzanalyse die Verbesserung der Separierbarkeit der Klassengebiete im Merkmalsraum. Daher geht im Gegensatz zur Hauptkomponentenanalyse statistische Klasseninformation in die Berechnung der Merkmalstransformation mit ein, sodass notwendigerweise eine bereits klassifizierte Lernstichprobe vorhanden sein muss.

Die zu bestimmende Transformation soll einerseits die Streuung der Merkmalsvektoren, die zur selben Klasse gehören, gering halten, während andererseits der Abstand der Klassenzentren im Merkmalsraum maximiert werden soll. Die Optimierung der Diskriminanzeigenschaften der Merkmalsrepräsentation basiert daher auf der Intraklassen-Streuungsmatrix $\boldsymbol{S}_w$ (engl. within-class-scatter) und der Interklassenstreuungsmatrix $\boldsymbol{S}_b$ (engl. between-class-scatter).

$$\boldsymbol{S}_w = \sum_{k=1}^{K} p_k \sum_{i=1}^{n_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^T \tag{3.43}$$

$$\boldsymbol{S}_b = \sum_{k=1}^{K} p_k (\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}})^T \tag{3.44}$$

Hierbei bezeichnet $\bar{\boldsymbol{x}}$ den globalen Mittelwert der Merkmale, K die Anzahl der Klassen, p_k die a-priori Wahrscheinlichkeit und $\bar{\boldsymbol{x}}_k$ den Mittelwert der jeweiligen Klasse.

Die Optimierung wird nun üblicherweise anhand des folgenden Kriteriums vorgenommen [Fuk72].

$$\text{Spur}\left\{\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\right\} \rightarrow \max! \tag{3.45}$$

Die gesuchte Transformationsmatrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_d]$ ergibt sich wiederum durch die Lösung eines Eigenwertproblems.

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\boldsymbol{\phi}_k = \lambda_k\boldsymbol{\phi}_k\,, \qquad k = 1, \ldots, d. \tag{3.46}$$

Die Transformationsmatrix $\boldsymbol{\Phi}$ besteht somit aus den d ($d < D$) Eigenvektoren der Matrix $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$, die zu den d größten Eigenwerten gehören.

Aus Gründen der numerischen Stabilität wird die Berechnung der Transformationsmatrix $\boldsymbol{\Phi}$ oftmals in einem zweistufigen Prozess durchgeführt (siehe z.B. [Fin03], Seite 149 und [Sch95b], Seite 117). Dabei wird im ersten Schritt durch ein sogenanntes

Whitening die Intraklassenstreuungsmatrix $\boldsymbol{S}_w$ auf Einheitsgestalt gebracht, die somit invariant gegenüber weiteren orthonormalen Transformationen ist. Anschließend erfolgt dann eine Hauptkomponentenanalyse der Klassenzentren, um die Separierbarkeit der Klassengebiete zu verbessern.

3.6 Klassifikation

Ausgehend von der im vorherigen Schritt erlangten Merkmalsrepräsentation wird im Klassifikationsschritt den Eingabesignalen ihre Bedeutung zugewiesen. Mit der Klassifikation erfolgt somit der Übergang vom Signal zu seiner symbolischen Repräsentation. Die Klassifikation lässt sich daher als Abbildung g auffassen, die Merkmalsvektoren $\boldsymbol{x}$ aus dem Merkmalsraum C zu Symbolen ω_k aus einem endlichen Symbolvorrat Ω zuordnet[6].

$$g : C \rightarrow \Omega, \quad \Omega = \{\omega_1, \ldots, \omega_K\} \tag{3.47}$$

Der Symbolvorrat Ω besteht bei der Handschrifterkennung im Falle einer analytischen Verarbeitungsstrategie üblicherweise aus Buchstaben, bei einer holistischen Vorgehensweise dagegen aus Wörtern.

Die Methoden, die zur Klassifikation eingesetzt werden können, lassen sich in die folgenden Kategorien gruppieren: Template-Matching, syntaktische, statistische und konnektionistische Verfahren [Jai00]. Diese unterschiedlichen Ansätze werden im folgenden kurz charakterisiert, bevor mit den Hidden Markov Modellen und speziellen Neuronalen Netzen Vertreter statistischer bzw. konnektionistischer Klassifikatoren vorgestellt werden, die im Bereich der Handschrifterkennung von überragender Bedeutung sind. Das Hauptaugenmerk liegt dabei auf den Hidden Markov Modellen, da sie auch das Grundgerüst der in dieser Arbeit realisierten Erkennungssysteme bilden.

Template Matching

Die Klassifikation durch Template-Matching ist ein vergleichsweise einfaches Verfahren, das auf dem Vergleich der zu klassifizierenden Eingabemuster mit im Vorhinein ermittelten Referenzmustern (Prototypen, Templates) basiert (vgl. [Jai00]). Die einzelnen Templates sind dabei die prototypischen Repräsentanten der jeweiligen Klasse ω_k, sodass ein Eingabemuster schließlich derjenigen Klasse zugeordnet wird, dessen Template die größte Ähnlichkeit zum Eingabemuster aufweist.

Aufgrund der Variabilität der Eingabemuster ist der Vergleich mit starren Template-Modellen häufig nachteilig, sodass stattdessen deformierbare Template-Modelle und elastische Matching-Verfahren verwendet werden. Ein effizientes Verfahren zum elastischen Matching, das auf Dynamischer Programmierung beruht, ist das *Dynamic Time Warping* (dt. Dynamische Zeitverzerrung). Näheres hierzu findet man u.a. in [Sch95b].

[6] Anstatt von der Abbildung des Merkmalsvektors $\boldsymbol{x}$ auf das Symbol ω_k zu sprechen, sagt man auch häufig, dass die *Klasse* ω_k bestimmt wird, zu der der Merkmalsvektor $\boldsymbol{x}$ gehört.

Syntaktische Klassifikation

Bei der syntaktischen Klassifikation wird das Eingabesignal als komplexes Muster betrachtet, das aus untereinander verknüpften Elementarmustern, den Primitiven, zusammengesetzt ist. Der Kerngedanke dabei ist, dass eine Analogie hergestellt wird zwischen der Struktur des Musters und der Syntax einer Sprache [Jai00]. Die Muster werden somit als Sätze der Sprache, die Primitive als Alphabet und die Struktur des Musters als Grammatik aufgefasst. Die Erkennung von Zeichen oder Wörtern geschieht also anhand einer Reihe von Regeln, die die für das jeweilige Zeichen bzw. Wort charakteristischen Beziehungen der Primitive untereinander beschreiben.

Der syntaktische, regelbasierte Ansatz wurde jedoch relativ schnell auf Grund der Schwierigkeiten bei der Detektion der Primitive und der Formulierung bzw. automatischen Extraktion robuster Regeln aufgegeben. Erst in jüngster Zeit erfuhr dieser Ansatz eine Wiederbelebung durch das Konzept der *Fuzzy-Logik*, also der Formulierung unscharfer Regeln. Die Bedeutung der syntaktischen Klassifikation im Vergleich zu den übrigen Methoden ist insbesondere im Bereich der Handschrifterkennung jedoch äußerst gering.

Statistische Klassifikation

Bei der statistischen Klassifikation wird davon ausgegangen, dass die observierten Muster durch einen stochastischen Prozess mit der Wahrscheinlichkeitsdichte $p(\boldsymbol{x}, \omega_k)$ generiert wurden. Dabei bezeichnet $\boldsymbol{x}$ die Merkmalsrepräsentation des Musters und ω_k seine zugehörige Klasse. Die Verbunddichte $p(\boldsymbol{x}, \omega_k)$ des als stationär angenommenen mustererzeugenden Prozesses lässt sich mit Hilfe der *Bayes-Regel* wie folgt ausdrücken:

$$p(\boldsymbol{x}, \omega_k) = P(\omega_k|\boldsymbol{x})p(\boldsymbol{x}) = p(\boldsymbol{x}|\omega_k)P(\omega_k). \tag{3.48}$$

Hierbei ist $P(\omega_k|\boldsymbol{x})$ die a posteriori Wahrscheinlichkeit der Klasse ω_k bei Vorliegen des Merkmalsvektors $\boldsymbol{x}$. Die a priori Wahrscheinlichkeit $P(\omega_k)$ gibt die Wahrscheinlichkeit der Klasse ω_k an, bevor überhaupt eine Beobachtung $\boldsymbol{x}$ vorliegt. Die klassenbedingte Dichte des Auftretens von $\boldsymbol{x}$, wenn die zugehörige Klasse ω_k bekannt ist, ist durch $p(\boldsymbol{x}|\omega_k)$ gegeben, und die Dichte der Merkmale unabhängig von ihrer Bedeutung ist durch $p(\boldsymbol{x})$ bezeichnet.

Die Klassifikation, d.h. die Bestimmung der Klasse ω_k des beobachteten Merkmalsvektors $\boldsymbol{x}$, wird auf Grundlage der obigen Formel vorgenommen. Stellt man 3.48 nach der a posteriori Wahrscheinlichkeit um, so erhält man:

$$P(\omega_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_k)P(\omega_k)}{p(\boldsymbol{x})} \tag{3.49}$$

Damit erhält man den theoretisch optimalen Klassifikator, wenn der Merkmalsvektor $\boldsymbol{x}$ derjenigen Klasse $\hat{\omega}_k$ zugeordnet wird, die zu maximaler a posteriori Wahrscheinlichkeit $P(\omega_k|\boldsymbol{x})$ führt, wobei der Nenner des obigen Ausdrucks, die Dichte der Merkmale

$p(\boldsymbol{x})$, bei der Maximumbildung unberücksichtigt bleiben darf, da dieser unabhängig von der Klasse ω_k ist.

Die optimale Klassifikation nach 3.49 lässt sich in der Praxis jedoch nur approximativ erreichen, da die wahren Werte der Wahrscheinlichkeiten $P(\omega_k|\boldsymbol{x})$, $P(\omega_k)$ und der Dichte $p(\boldsymbol{x}|\omega_k)$ nicht bekannt sind, sondern vielmehr aus einer möglichst repräsentativen Trainingsstichprobe geschätzt werden müssen. Abgesehen von den a priori Wahrscheinlichkeiten $P(\omega_k)$, die beispielsweise aus einer anwendungsspezifischen Textstichprobe geschätzt oder bei keinerlei Vorwissen als gleichverteilt angenommen werden können, besteht die Hauptaufgabe bei der Realisierung eines statistischen Klassifikators darin, anhand der Trainingsstichprobe entweder

(a) die a posteriori Wahrscheinlichkeiten $P(\omega_k|\boldsymbol{x})$ oder

(b) die klassenbedingten Dichten $p(\boldsymbol{x}|\omega_k)$

zu schätzen [Sch96]. Folgt man dem Ansatz (a), so führt dies zum Konzept der *MAP-Schätzung* (Maximum a posteriori Schätzung), während der Ansatz (b) als *Maximum Likelihood Schätzung* bezeichnet wird.

Die MAP-Schätzung lässt sich als Funktionsapproximationsaufgabe auffassen. Die Aufgabe besteht darin, die Parameter $\boldsymbol{w}$ einer Funktion $\boldsymbol{d_w}(\boldsymbol{x})$ so zu optimieren, dass die Funktion dem *Zielvektor* $\boldsymbol{y}(\boldsymbol{x})$ möglichst ähnlich ist. Der Zielvektor ist dabei ein K-dimensionaler Einheitsvektor und beschreibt die Klassenzugehörigkeit des Merkmalsvektors. Entstammt der Merkmalsvektor $\boldsymbol{x}$ beispielsweise der Klasse ω_k, so enthält der Zielvektor an der Position k eine Eins. Die Parameter der sogenannten *Unterscheidungsfunktion* $\boldsymbol{d_w}(\boldsymbol{x})$ lassen sich nun durch Minimierung des mittleren quadratischen Fehlers bestimmen:

$$E\{|\boldsymbol{d_w}(\boldsymbol{x}) - \boldsymbol{y}(\boldsymbol{x})|^2\} = \min_{\boldsymbol{d_w}(\boldsymbol{x})} ! \tag{3.50}$$

Für die parametrischen Funktionen kommen beispielsweise Polynome, radiale Basisfunktionen oder die sigmoiden Aktivierungsfunktionen neuronaler Netze in Betracht.

Die Wahl der parametrischen Funktionen $\boldsymbol{d_w}(\boldsymbol{x})$ ist dabei mit entscheidend für die Leistungsfähigkeit des Klassifikators. Weisen die Funktionen durch eine Vielzahl von Parametern ein hohes Maß an Flexibilität auf, so besteht die Gefahr des *Overfittings* der zu approximierenden Funktion an die Gegebenheiten des Trainingssets. Die Folge ist eine verminderte Generalisierungsfähigkeit, d.h. die Übertragbarkeit auf die konkreten Anwendungsbedingungen ist nicht mehr gegeben. Ist andererseits jedoch die Flexibilität der parametrischen Funktionen zu sehr eingeschränkt, so führt dies dazu, dass die systematische Datenvariabilität, also das Vorhandensein unterschiedlicher Ausprägungen von Mustern derselben Klasse, nicht ausreichend modelliert werden kann. Beim Entwurf des Klassifikators muss damit eine Abwägung zwischen der Genauigkeit der Approximation und der Generalisierungsfähigkeit vorgenommen werden.

Im Gegensatz zur MAP-Schätzung geht man bei der Maximum-Likelihood-Schätzung von statistischen Modellen aus, die die klassenbedingte Dichte $p(\boldsymbol{x}|\omega_k)$

repräsentieren. Die Lernaufgabe besteht in diesem Fall darin, die Parameter der statistischen Modelle so zu schätzen, dass die Wahrscheinlichkeit maximiert wird, mit der die observierten Daten von den so parametrisierten Modellen generiert wurden. Insbesondere zur Klassifikation von Observationsfolgen, wenn also die Merkmalsvektoren wie bei der Sprach- oder Handschrifterkennung als Sequenzen vorliegen, hat sich dabei die Verwendung von *Hidden-Markov-Modellen* (siehe Abschnitt 3.6.1) als äußerst erfolgreich erwiesen.

Konnektionistische Klassifikation

Die Verfahren zur konnektionistischen Klassifikation versuchen die in Nervensystemen durchgeführte Informationsverarbeitung mit Hilfe künstlicher neuronaler Netze nachzubilden. Der zentrale Baustein ist dabei das Modellneuron, dessen biologisches Vorbild im wesentlichen drei Bestandteile umfasst [Zel97]: Neben dem *Zellkörper* sind dies die *Dendriten*, die die Eingaben aufnehmen, und das *Axon*, welches die Ausgabe des Neurons weiterleitet. Indem sich das Axon verzweigt und über *Synapsen* mit anderen Neuronen in Kontakt tritt werden Verbindungen zwischen Neuronen etabliert (siehe Abbildung 3.20).

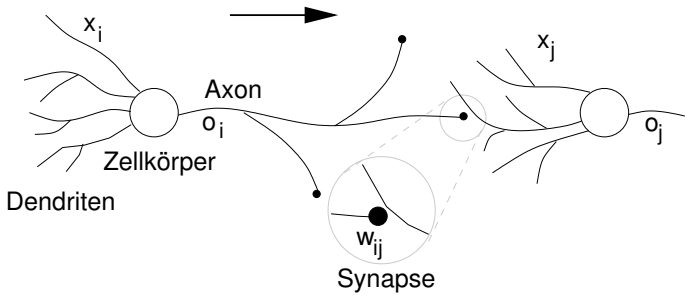


Abbildung 3.20: Verschaltung zweier Neuronen (angelehnt an [Zel97]). Der Pfeil gibt die Richtung des Informationsflusses an.

Die Funktionsweise eines Modellneurons wird üblicherweise wie folgt beschrieben: Im ersten Schritt werden die mit den Synapsenstärken w_i gewichteten Eingaben x_i aufsummiert. Anhand dieser Summe der gewichteten Eingaben wird dann mit Hilfe der *Aktivierungsfunktion* f_{act} der sogenannte *Aktivierungszustand* des Neurons bestimmt. Auf den Aktivierungszustand wird schließlich die Ausgabefunktion f_{out} angewendet, um die Ausgabe des Neurons zu bestimmen. Mathematisch lässt sich die Funktionsweise eines Modellneurons damit wie folgt ausdrücken:

$$o = f_{out}\left(f_{act}\left(\sum_{i=1}^{D} w_i x_i\right)\right) = f_{out}\left(f_{act}(\boldsymbol{w}^T\boldsymbol{x})\right). \tag{3.51}$$

Eine verbreitete Wahl für die Aktivierungsfunktion stellen sigmoide Funktionen dar, wie beispielsweise der *tangens hyperbolicus*, $f_{act} = \tanh(s)$, oder die *logistische*

Funktion, $f_{act} = (1 + \exp(-s))^{-1}$. Dagegen wird für die Ausgabefunktion des Modellneurons häufig die Identität gewählt, sodass Ausgabe und Aktivierung des Neurons übereinstimmen. Unter dieser Annahme vereinfacht sich Gleichung 3.51 zu:

$$o = f_{act}\left(\sum_{i=1}^{D} w_i x_i\right) = f_{act}(\boldsymbol{w}^T \boldsymbol{x}). \tag{3.52}$$

Die Leistungsfähigkeit konnektionistischer Methoden ergibt sich jedoch erst durch die Verknüpfung einzelner Neuronen zu komplexen Netzwerken. Dabei ist die am häufigsten eingesetzte Verschaltungsart die Vorwärtskopplung, die die sogenannten *Multi-Layer-Perzeptrons* (MLPs) charakterisieren.

Ein Beispiel eines MLPs mit vier Neuronenschichten und drei Schichten trainierbarer Verbindungsgewichte ist in Abbildung 3.21 dargestellt. Es handelt sich hier um ein vollständig ebenenweise verbundenes Netzwerk, bei dem jedes Neuron der Ebene l alle Neuronen der Ebene $l + 1$ als Nachfolger besitzt. Die Neuronen der Eingabeschicht werden dabei als *Eingabeneuronen* (*input units*) bezeichnet, die Neuronen der Ausgabeschicht entsprechend als *Ausgabeneuronen* (*output units*) und die Neuronen der inneren Schichten als *verdeckte Neuronen* (*hidden units*).

Das j-te Neuron der Schicht l des MLPs berechnet somit anhand des Eingabevektors $\boldsymbol{x}^l$ und des Gewichtsvektors $\boldsymbol{w}_j^l$ den Ausgangswert (mit $f_{out} =$ Identität)

$$o_j^l = f_{act}\left(\sum_{i=1}^{M_l} w_{ji}^l x_i^l\right) = f_{act}\left(((\boldsymbol{w}_j^l)^T \boldsymbol{x}^l\right).$$

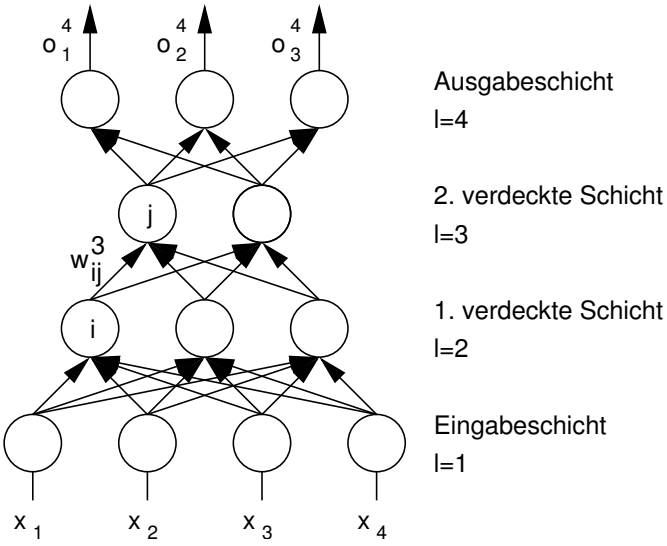


Abbildung 3.21: Multi-Layer-Perzeptron (MLP).

Fasst man die Ausgabewerte der Neuronen der Schicht l zusammen, so ergibt sich mit der Gewichtsmatrix $\boldsymbol{W}^l$ für den Ausgabevektor der l-ten Schicht

$$\boldsymbol{o}^l = \boldsymbol{f}_{act}\left((\boldsymbol{W}^l)^T \boldsymbol{x}^l\right),$$

wobei der Ausgabevektor $\boldsymbol{o}^l$ der Schicht l wiederum als Eingabevektor $\boldsymbol{x}^{l+1}$ der nachfolgenden Schicht $l+1$ fungiert.

Die Klassifikation von Merkmalsvektoren mit Hilfe von MLPs basiert auf der MAP-Schätzung der a posteriori Wahrscheinlichkeit $P(\omega_k|\boldsymbol{x})$ (siehe Seite 71, Gleichung 3.50). Das MLP, das somit auch als statistischer Klassifikator aufgefasst werden kann, dient dabei zur Approximation der Unterscheidungsfunktion $\boldsymbol{d}_{\boldsymbol{w}}(\boldsymbol{x})$. Die Approximationsaufgabe besteht darin, die Gewichtsparameter $\boldsymbol{W}^l$ des MLPs anhand der Trainingsstichprobe so zu optimieren, dass die Netzausgabe $\boldsymbol{o}^L$ mit dem jeweiligen Zielvektor $\boldsymbol{y}(\boldsymbol{x})$ möglichst gut übereinstimmt. Die Bestimmung der optimalen Gewichtsparameter erfolgt dabei durch Minimierung des mittleren quadratischen Fehlers

$$E\{|\boldsymbol{o}^L - \boldsymbol{y}(\boldsymbol{x})|^2\} = \min_{\boldsymbol{W}^1,\ldots,\boldsymbol{W}^L} !$$

Da keine geschlossene Lösung dieser Optimierungsaufgabe bekannt ist, werden hier meist Gradientenabstiegsverfahren eingesetzt.

Die in Abbildung 3.21 vorgestellte Architektur des MLPs eignet sich gut zur Klassifikation von Mustern, die sich jeweils mit einem Merkmalsvektor fester Dimension beschreiben lassen. Zur Klassifikation von Merkmalsvektorfolgen, wie sie im Bereich der Schrifterkennung vorliegen, werden dagegen modifizierte Netzwerkarchitekturen eingesetzt, vorwiegend sogenannte *Time-Delay Neural Networks* (TDNNs, siehe u.a. [Man94, Sen94b, Jae01]), die in Abschnitt 3.6.2 beschrieben werden.

3.6.1 Hidden Markov Modelle

Hidden-Markov-Modelle (HMMs) sind das dominierende Konzept zur statistischen Klassifikation von Beobachtungsfolgen variabler Länge. Inspiriert vom überaus erfolgreichen Einsatz von HMMs im Bereich der automatischen Sprachverarbeitung erlangen HMM-basierte Systeme auch im Handschriftbereich wachsende Bedeutung.

Ausgehend von der Definition von HMMs werden im folgenden wichtige Modellierungsaspekte betrachtet und die grundlegenden Verwendungskonzepte zur Klassifikation vorgestellt. Anschließend wird auf die drei fundamentalen Aufgabenstellungen im Umgang mit HMMs eingegangen – die Berechnung der Produktionswahrscheinlichkeit, die Dekodierung und das Training. Die vorliegende Darstellung orientiert sich dabei vorwiegend an den Arbeiten [Sch95b, Fin03].

Definitionen

Ein diskreter stochastischer Prozess $\boldsymbol{q}$ ist eine Folge von Zufallsvariablen, die jeweils einen Wert aus einer endlichen Zustandsmenge $S = \{s_1, s_2, \ldots, s_N\}$ annehmen:

$$\boldsymbol{q} = q_1, q_2 \ldots, q_T\,, \quad q_t \in S. \tag{3.53}$$

Der Parameter t des stochastischen Prozesses kann beispielsweise als Zeit interpretiert werden. Weist der Prozess $\boldsymbol{q}$ die Markov-Eigenschaft auf, dass die Wahrscheinlichkeit des Zustands $q_t = s_i$ nur vom direkten Vorgängerzustand $q_{t-1} = s_j$ abhängt,

$$P(q_t = s_j | q_{t-1} = s_i, \ldots, q_0 = s_k) = P(q_t = s_j | q_{t-1} = s_i), \tag{3.54}$$

und ist der Prozess darüberhinaus stationär, also von der absoluten Zeit unabhängig, so wird der Prozess als *homogene Markov-Kette* bezeichnet.

Die bedingten Wahrscheinlichkeiten $P(q_t = s_j | q_{t-1} = s_i)$ werden *Übergangswahrscheinlichkeiten* genannt, und geben die Wahrscheinlichkeit an, dass die Zufallsvariable $\boldsymbol{q}$ zum Zeitpunkt t den Wert s_j annimmt, wenn sie zum vorhergehenden Zeitpunkt den Wert s_i aufwies. Bei einer homogenen Markov-Kette können diese Übergangswahrscheinlichkeiten in Form einer quadratischen $N \times N$ Matrix dargestellt werden:

$$\boldsymbol{A} = [a_{ij}] \quad \text{mit} \quad a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \tag{3.55}$$

wobei die Stochastizitätsbedingungen

$$a_{ij} \geq 0 \quad \forall i, j \quad \text{und} \quad \sum_{j=1}^{N} a_{ij} = 1 \quad \forall i \tag{3.56}$$

gelten.

Die Initialisierung der Markov-Kette wird durch den Vektor der Anfangswahrscheinlichkeiten $\boldsymbol{\pi}$ bestimmt:

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N), \quad \pi_i = P(q_1 = s_i) \quad \text{mit} \quad \sum_{i=1}^{N} \pi_i = 1. \tag{3.57}$$

Die durch die Parameter $\boldsymbol{\pi}$ und $\boldsymbol{A}$ vollständig charakterisierte Markov-Kette bildet die erste Stufe eines Hidden Markov Modells.

Die zweite Stufe eines HMMs ist ein Prozess, der in Abhängigkeit vom aktuell eingenommenen Zustand eine Emission erzeugt. Nur diese Emissionsfolge eines HMMs ist beobachtbar, die Folge der eingenommenen Zustände dagegen nicht. Auf Grund dieser Eigenschaft, der dem Beobachter "verborgenen" Zustandsfolge, wird der zweistufige stochastische Prozess als *Hidden* Markov Modell bezeichnet.

Die Emissionsfolge $\boldsymbol{O} = o_1, \ldots, o_T$ besteht dabei entweder aus Symbolen $v_k \in \{v_1, \ldots, v_D\}$ aus einem endlichen Symbolvorrat oder Vektoren $\boldsymbol{x} \in \mathbb{R}^D$ aus einem D-dimensionalen Vektorraum. Im ersten Fall, man spricht dabei von *diskreten HMMs*, lassen sich die Emissionswahrscheinlichkeiten in Form einer $N \times D$ Matrix darstellen:

$$\boldsymbol{B} = [b_{jk}] \quad \text{mit} \quad b_{jk} = P(o_t = v_k | q_t = s_j) \tag{3.58}$$

und

$$b_{jk} \geq 0 \quad \forall j, k \quad \text{und} \quad \sum_{k=1}^{N} b_{jk} = 1 \quad \forall j. \tag{3.59}$$

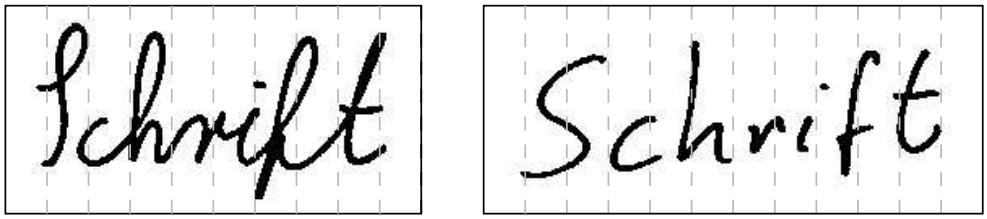


Abbildung 3.22: Längen- und Formvariation der Schrift. Die segmentierten Abschnitte, anhand derer die Merkmalsvektoren extrahiert werden, sind in grau angedeutet.

Im Falle vektorwertiger Emissionen *kontinuierlicher HMMs* wird analog ein N-dimensionaler Dichte-Vektor definiert:

$$\boldsymbol{B} = [b_j] \quad \text{mit} \quad b_j(\boldsymbol{x}) = p(o_t = \boldsymbol{x} | q_t = s_j) \tag{3.60}$$

und

$$b_j(\boldsymbol{x}) \geq 0 \quad \forall j \quad \text{und} \quad \int_{\mathbb{R}^D} b_j(\boldsymbol{x}) d\boldsymbol{x} = 1 \quad \forall j. \tag{3.61}$$

Ein Hidden Markov Modell ist damit durch das Tripel

$$\boldsymbol{\lambda} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}) \tag{3.62}$$

definiert. Es beschreibt einen zweistufigen stochastischen Prozess, dessen erste Stufe eine homogene Markov-Kette bildet, bei der – ausgehend vom Initialisierungsvektor – die Zufallsvariable gemäß der Übergangswahrscheinlichkeiten einen Zustand aus einer endlichen Zustandsmenge einnimmt. Dieser Zustand ist jedoch nicht beobachtbar. Vielmehr wird in Abhängigkeit des eingenommenen Zustands eine Emission generiert, die entweder ein Symbol oder ein Vektor sein kann. Diese Emissionsgenerierung wird ebenfalls durch einen stochastischen Prozess beschrieben, der die zweite Stufe des HMMs bildet.

Modellierung

Durch den zweistufigen Aufbau von HMMs eignen sie sich sehr gut zur Klassifikation von Merkmalsvektorfolgen variierender Länge, wie sie typischerweise im Sprach- bzw. Handschriftbereich vorliegen. In Abbildung 3.22 sind beispielsweise zwei Schriftbilder aus dem Offline-Handschriftbereich dargestellt, die die Längen- bzw. Formvariabilität der Schrift verdeutlichen. So unterscheidet sich sowohl der Schriftstil – Schreibschrift gegenüber Blockschrift – als auch die horizontale Ausdehnung und damit die Anzahl der extrahierten Merkmalsvektoren.

Zur Modellierung dieser Variabilität eignet sich das in Abbildung 3.23 gezeigte HMM. Das abgebildete HMM besitzt sieben Zustände, die jeweils einem Buchstaben des modellierten Wortes entsprechen. Für die Zustände sind sowohl Selbstübergänge

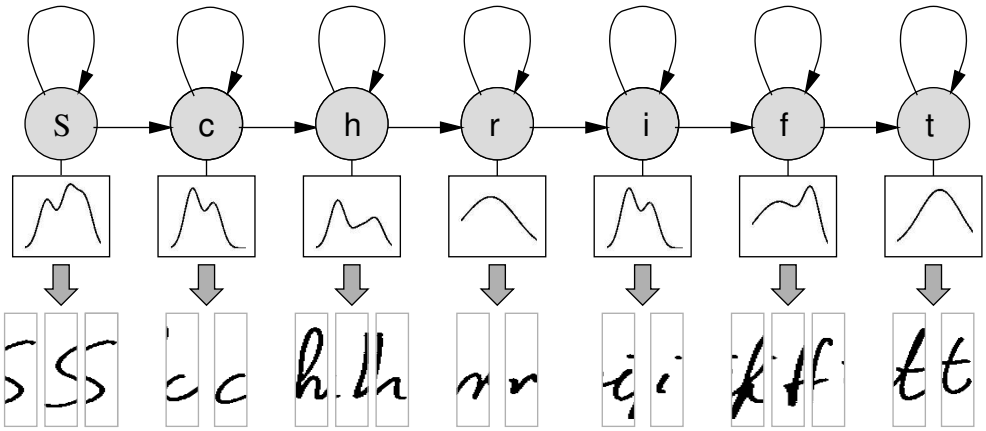


Abbildung 3.23: HMM zur Modellierung des Wortes “Schrift”. Nur die von Null verschiedenen Übergangswahrscheinlichkeiten sind eingezeichnet.

möglich als auch der Übergang in den Zustand, der dem im Wort folgenden Buchstaben entspricht. Die Längenvariabilität der Schrift kann also durch die Selbstübergänge modelliert werden, da sie das Verbleiben in einem Zustand und damit die Generierung mehrerer Emissionen über mehrere Abschnitte des Signals hinweg ermöglichen.

Der Observationsvektor wird in Abhängigkeit des eingenommenen Zustands gemäß der zugehörigen Wahrscheinlichkeitsdichte emittiert. Dabei sind unterschiedliche Realisierungen (Schreibschrift bzw. Blockschrift) zugelassen.

Während mit Hilfe der ersten Stufe des HMMs, der Zustandsübergänge, die Längenvariabilität der Schrift erfasst wird, gelingt mit der zweiten Stufe, der stochastischen Emissionsgenerierung, die Modellierung der Form- bzw. Schriftstilunterschiede.

Modelltopologie

Ein wichtiger Schritt beim Entwurf von HMMs ist die Festlegung der Modelltopologie, d.h. der Gestalt der Matrix $\boldsymbol{A}$ der Übergangswahrscheinlichkeiten. Eine Übersicht gängiger Modelltopologien ist in Abbildung 3.24 dargestellt.

Die höchst mögliche Flexibilität der Modelle in Bezug auf die Zustandsübergänge wird erreicht, wenn die Matrix $\boldsymbol{A}$ voll besetzt ist. HMMs, die eine solche Topologie aufweisen, werden *ergodisch* genannt. Sind die zu modellierenden Prozesse jedoch durch eine bestimmte Struktur gekennzeichnet, sodass die beobachteten Ereignisse in einer zeitlichen oder räumlichen Abfolge auftreten, können eingeschränkte Modelltopologien verwendet werden, bei denen bestimmte Übergangswahrscheinlichkeiten zu Null gesetzt werden.

Das in Abbildung 3.23 dargestellte HMM weist eine *lineare* Topologie auf. Hier treten von Null verschiedene Übergangswahrscheinlichkeiten nur bei Selbstübergängen und bei Übergängen der Form $s_i \rightarrow s_{i+1}$ auf. *Bakis-Modelle* sind darüberhinaus dadurch gekennzeichnet, dass außerdem Transitionen der Art $s_i \rightarrow s_{i+2}$ zugelassen sind und damit einzelne Zustände übersprungen werden können. Soll auch das Über-

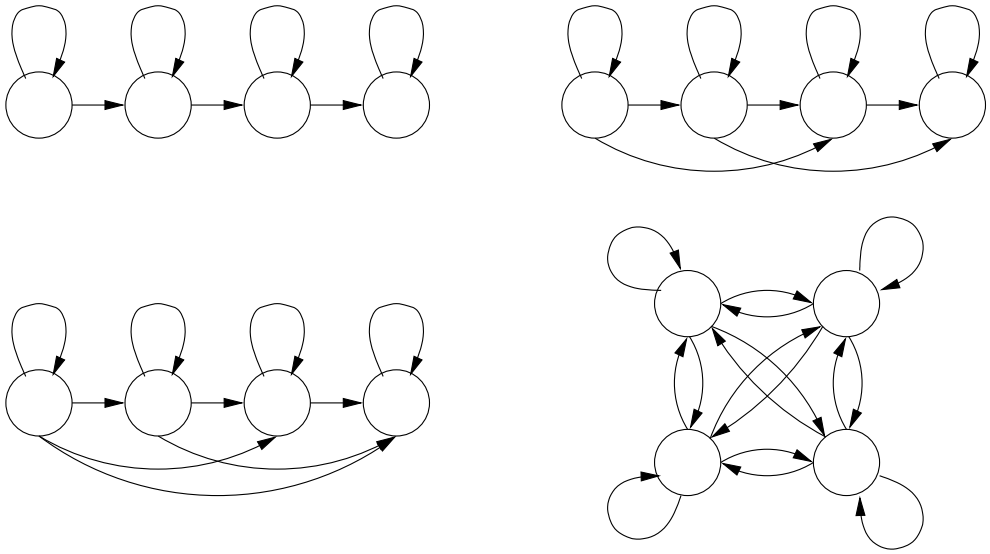


Abbildung 3.24: HMM-Topologien. Von links oben nach rechts unten: Linear, Bakis, Links-Rechts, Ergodisch.

springen mehrerer Zustände ermöglicht werden, sodass beispielsweise vom Startzustand direkt in den Endzustand gesprungen werden kann, bietet sich die Verwendung von *Links-Rechts-Modellen* an. Sie erlauben Zustandsübergänge der Art $s_i \rightarrow s_j$, für alle $i \leq j$.

Bei der Festlegung der Modelltopologie ist zu beachten, dass eine gesteigerte Flexibilität durch eine erhöhte Zahl möglicher Zustandsübergänge auch einen Mehraufwand beim Training und bei der Dekodierung der Modelle nach sich zieht. Die Wahl der Topologie resultiert daher i.d.R. aus einer Abwägung zwischen Flexibilität und Handhabbarkeit der Modelle.

Emissionsmodellierung

Neben der Bestimmung der Modelltopologie ist vor allem die Emissionsmodellierung eine der zentralen Aufgabenstellungen beim Entwurf von HMMs. Im Bereich der Handschrifterkennung hat sich dabei die Verwendung diskreter HMMs, bei denen die Observationen also aus einem diskreten, endlichen Symbolvorrat entstammen, kaum durchgesetzt. Dies liegt vor allem daran, dass die extrahierten Schriftmerkmale i.d.R. Elemente des $\mathbb{R}^D$ und damit kontinuierlicher Natur sind. Daraus folgt, dass zum Einsatz diskreter HMMs ein Verfahren zur Vektorquantisierung vorgeschaltet werden muss, um die kontinuierlichen Merkmale auf diskrete Symbole abzubilden. Mit der Vektorquantisierung geht jedoch ein nicht kompensierbarer Informationsverlust einher, der zu einer Verringerung der Klassifikationsleistung der HMMs führt.

Die Quantisierungsstufe und mithin der Quantisierungsfehler kann durch die direkte Verwendung der extrahierten Merkmalsvektoren in kontinuierlichen HMMs vermieden werden. Die zustandsabhängige, kontinuierliche Emissionsverteilungsdichte $b_j(\boldsymbol{c})$

wird dabei üblicherweise durch eine Gauß'sche Mischverteilungsdichte

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{K_j} c_{jk} g_{jk}(\boldsymbol{x}) = \sum_{k=1}^{K_j} c_{jk} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{jk}, \boldsymbol{K}_{jk}) \tag{3.63}$$

beschrieben, da sich mit ihr bei geeigneter Wahl der Anzahl K_j der Mischungskomponenten jede beliebige Dichtefunktion approximieren lässt. In obiger Formel bezeichnen die c_{jk} die Mischungsgewichte, die den Bedingungen

$$\sum_{k=1}^{K_j} c_{jk} = 1 \quad \forall j \quad \text{und} \quad c_{jk} \geq 0 \quad \forall j,k \tag{3.64}$$

genügen müssen. Die Parameter $\boldsymbol{\mu}_{jk}$ und $\boldsymbol{K}_{jk}$ beschreiben den Mittelwertvektor bzw. die Kovarianzmatrix der entsprechenden Gaußdichte.

Da bei dieser Art der Emissionsmodellierung jedoch eine sehr hohe Anzahl von Parametern bestimmt werden muss – für jeden Zustand s_j die Mischungskoeffizienten, Mittelwertvektoren und Kovarianzmatrizen von K_j Gaußdichten – ist es vorteilhaft, für alle Zustände einen *gemeinsamen* Satz von Gaußdichten zur Emissionsmodellierung zu verwenden:

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{K} c_{jk} g_k(\boldsymbol{x}) = \sum_{k=1}^{K} c_{jk} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{K}_k). \tag{3.65}$$

Die so beschriebene Emissionsmodellierung ist das Kennzeichen sogenannter *semikontinuierlicher HMMs*. Sie können auch als diskrete HMMs mit integrierter "weicher" Vektorquantisierung aufgefasst werden, indem die Mischungskoeffizienten c_{jk} als Ausgabewahrscheinlichkeiten des diskreten Modells interpretiert werden, die im Gegensatz dazu jedoch mit Hilfe der Dichtewerte $g_k(\boldsymbol{x})$ gewichtet werden.

Verwendung zur Klassifikation

HMMs sind generative Modelle, die durch mehrstufige stochastische Prozesse Observationsfolgen mit der Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ erzeugen. Wie können nun aber HMMs eingesetzt werden, um eine gegebene Folge von Merkmalsvektoren zu klassifizieren?

Die Grundlage für die Klassifikation ist die Bayes-Regel (siehe Gleichung 3.48). Wird für jede Musterklasse ω_k ein separates HMM $\boldsymbol{\lambda}_k$ definiert, so ergibt sich anhand der Bayes-Regel die a posteriori Wahrscheinlichkeit zu:

$$P(\boldsymbol{\lambda}_k|\boldsymbol{O}) = \frac{P(\boldsymbol{O}|\boldsymbol{\lambda}_k)P(\boldsymbol{\lambda}_k)}{P(\boldsymbol{O})} \tag{3.66}$$

Die Klassifikationsentscheidung fällt damit zu Gunsten derjenigen Klasse ω_k aus, deren zugehöriges HMM $\boldsymbol{\lambda}_k$ die a posteriori Wahrscheinlichkeit $P(\boldsymbol{\lambda}_k|\boldsymbol{O})$ maximiert. Da

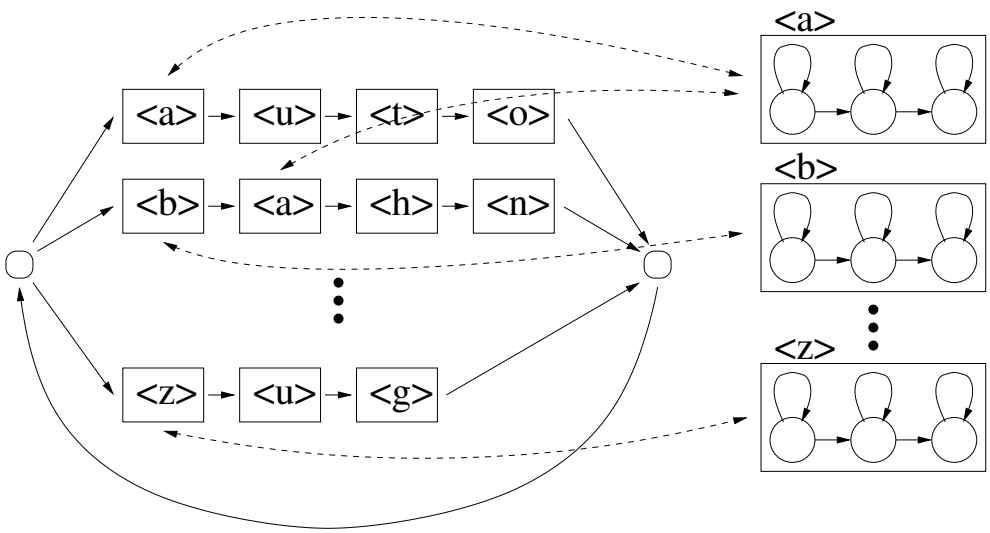


Abbildung 3.25: Zusammengeschaltete Buchstabenmodelle zur Verbundworterkennung.

der Nenner des obigen Ausdrucks unabhängig von der Klassenentscheidung ist, kann er bei der Maximierung ignoriert werden. Die Entscheidungsregel lautet demnach:

$$\boldsymbol{\lambda}_l = \arg\max_{\boldsymbol{\lambda}_k} P(\boldsymbol{O}|\boldsymbol{\lambda}_k)P(\boldsymbol{\lambda}_k) \tag{3.67}$$

Liegt keinerlei Wissen über die a priori Wahrscheinlichkeit $P(\boldsymbol{\lambda}_k)$ vor, so wird hierfür i.d.R. eine Gleichverteilung angenommen. Die Klassifikationsentscheidung erfolgt dann ausschließlich anhand der Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda}_k)$, sodass damit ein Maximum-Likelihood-Klassifikator vorliegt.

Diese sogenannte *modelldiskriminante* Vorgehensweise, bei der jeder Musterklasse ein einzelnes HMM zugeordnet und die Klassifikation anhand der maximalen Produktionswahrscheinlichkeit vorgenommen wird, kann zur Klassifikation von Signalen eingesetzt werden, die nicht weiter zu segmentieren sind. Im Bereich der Handschrifterkennung wird diese holistische Strategie daher in Systemen zur Einzelworterkennung angewandt. Da jedoch für jedes Wort ein zugehöriges HMM benötigt wird, ist die holistische Strategie nur für kleine Wortschätze handhabbar.

Statt der Verwendung von Wortmodellen werden daher meistens kürzere Signalabschnitte modelliert, die üblicherweise Buchstaben entsprechen. Zur Erkennung von Wörtern werden die Buchstabenmodelle dann zu einem komplexen Verbundmodell zusammengeschaltet (siehe Abbildung 3.25). Wird ein solches Verbundmodell zur Klassifikation eingesetzt, so kommt die Verwendung der Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda}_k)$ zur Entscheidungsfindung nicht mehr in Betracht. Stattdessen muss der wahrscheinlichste Pfad durch das Verbundmodell ermittelt werden, der zu der beobachteten Observationsfolge geführt hat. Bei diesem sogenannten *pfaddiskriminanten* Ansatz wird deshalb diejenige Zustandsfolge $\boldsymbol{s}^*$ bestimmt, die im Modell $\boldsymbol{\lambda}$ die Produktionswahrscheinlichkeit $P(\boldsymbol{O}, \boldsymbol{s}^*|\boldsymbol{\lambda})$ maximiert. Da sich anhand der Zustandsfolge

die zugehörigen Buchstabenmodelle eindeutig identifizieren lassen, wird mithin eine implizite Segmentierung der Observationsfolge vorgenommen.

Die pfaddiskriminante Vorgehensweise ist durch die Verwendung eines gemeinsamen Satzes von Buchstaben-HMMs zur Modellierung der Wörter des Erkennungslexikons im Vergleich zum modelldiskriminanten Ansatz deutlich flexibler. So können neue Wörter leicht durch entsprechende Verschaltungen bereits bestehender Buchstaben-HMMs modelliert werden, sodass sich diese Technik auch für größere Wortschätze eignet.

Berechnung der Produktionswahrscheinlichkeit

Im vorherigen Abschnitt wurde beschrieben, dass bei der modelldiskriminanten Klassifikation die Klassenentscheidung anhand der Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ erfolgen kann. Sie gibt für das gegebene HMM $\boldsymbol{\lambda}$ die Wahrscheinlichkeit an, die Observationsfolge $\boldsymbol{O}$ zu generieren und kann damit als Bewertungsmaß für die Genauigkeit der Modellierung verwendet werden.

Die Berechnung der Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ kann effizient, d.h. linear in Bezug auf die Länge der Observationsfolge, mit Hilfe der dynamischen Programmierung vorgenommen werden. Dazu können einerseits die *Vorwärtswahrscheinlichkeiten*

$$\alpha_t(j) = P(o_1, \ldots, o_t, q_t = s_j|\boldsymbol{\lambda}) \tag{3.68}$$

oder andererseits die *Rückwärtswahrscheinlichkeiten*

$$\beta_t(j) = P(o_{t+1}, \ldots, o_T|q_t = s_j, \boldsymbol{\lambda}) \tag{3.69}$$

verwandt werden. Die Vorwärtswahrscheinlichkeit $\alpha_t(j)$ gibt also die Wahrscheinlichkeit an, die Beobachtungsfolge $o_1, \ldots, o_t$ zu observieren und zum Zeitpunkt t im Zustand s_j zu sein. Demgegenüber gibt $\beta_t(j)$ die Wahrscheinlichkeit an, ab dem Zeitpunkt $t+1$ die Beobachtungsfolge $o_{t+1}, \ldots, o_t$ zu beobachten, falls man zum Zeitpunkt t im Zustand s_j ist. Gemäß obiger Definitionen gilt also:

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i). \tag{3.70}$$

Aufgrund der Markov-Eigenschaft des stochastischen Prozesses können die $\alpha_t(j)$ bzw. $\beta_t(j)$ rekursiv anhand ihres jeweiligen Vorgängerwertes berechnet werden. Für die Initialisierung der Berechnung setzt man:

$$\alpha_1(i) = \pi_i b_i(o_1) \tag{3.71}$$

bzw.

$$\beta_T(i) = 1. \tag{3.72}$$

1. Initialisierung:

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad \beta_T(j) = 1$$

2. Rekursion:

$$\forall t : t = 1, \ldots, T-1 \qquad \forall t : t = T-1, \ldots, 1$$

$$\alpha_{t+1}(j) = \sum_i^N \{\alpha_t(i) a_{ij}\} b_j(o_{t+1}) \qquad \beta_t(i) = \sum_j^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

3. Terminierung:

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{i=1}^N \alpha_T(i) \qquad P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{j=1}^N \pi_j b_j(o_1) \beta_1(j)$$

Abbildung 3.26: Berechnung der Produktionswahrscheinlichkeit. Links mittels der Vorwärtswahrscheinlichkeiten, rechts mittels der Rückwärtswahrscheinlichkeiten.

Die Vorwärtswahrscheinlichkeit des folgenden Zeitpunktes kann dann rekursiv bestimmt werden durch:

$$\alpha_{t+1}(j) = \sum_i^N \{\alpha_t(i) a_{ij}\} b_j(o_{t+1}). \tag{3.73}$$

Analog gilt für die Rückwärtswahrscheinlichkeiten:

$$\beta_t(i) = \sum_j^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j). \tag{3.74}$$

Für die Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ erhält man somit am Ende der Berechnungen:

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{i=1}^N \alpha_T(i) \tag{3.75}$$

bzw.

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{j=1}^N \pi_j b_j(o_1) \beta_1(j). \tag{3.76}$$

Beide Verfahren, d.h. die Verwendung der Vorwärtswahrscheinlichkeiten einerseits oder die Verwendung der Rückwärtswahrscheinlichkeiten andererseits, sind gleichwertig, der wesentliche Unterschied liegt in der Rekursionsrichtung. In Abbildung 3.26 sind die Verfahren im Überblick dargestellt.

Berechnung der optimalen Zustandsfolge

Bei der pfaddiskriminanten Verwendung von HMMs ist es erforderlich, die internen Abläufe eines HMMs, d.h. die Folge der eingenommenen Zustände, aufzudecken. Das Ziel dieser *Dekodierungsaufgabe* ist also die Bestimmung derjenigen Zustandsfolge, die die Wahrscheinlichkeit

$$P(\boldsymbol{q}|\boldsymbol{O},\boldsymbol{\lambda}) = \frac{P(\boldsymbol{O},\boldsymbol{q}|\boldsymbol{\lambda})}{P(\boldsymbol{O}|\boldsymbol{\lambda})} \tag{3.77}$$

bei gegebenem Modell $\boldsymbol{\lambda}$ und vorliegender Observationsfolge $\boldsymbol{O}$ maximiert. Da der Nenner des obigen Ausdrucks von der gesuchten optimalen Zustandsfolge $\boldsymbol{q}^*$ unabhängig ist ergibt sich:

$$\boldsymbol{q}^* = \underset{\boldsymbol{q}}{\operatorname{argmax}}\, P(\boldsymbol{O},\boldsymbol{q}|\boldsymbol{\lambda}) \tag{3.78}$$

Die Bestimmung dieser optimalen Zustandsfolge kann wiederum effizient mit Hilfe der dynamischen Programmierung erfolgen. Dabei wird in ähnlicher Weise vorgegangen wie bei der Berechnung der Vorwärtswahrscheinlichkeit, mit der Ausnahme, dass die Summation durch eine Maximumbildung ersetzt wird. Anstelle der Vorwärtswahrscheinlichkeit $\alpha_t(j)$ tritt somit die Wahrscheinlichkeit

$$\vartheta_t(i) = \max_{q_1,\ldots,q_{t-1}} P(o_1,\ldots,o_t,q_1,\ldots,q_{t-1},q_t = s_i|\boldsymbol{\lambda}), \tag{3.79}$$

die die maximale Wahrscheinlichkeit bezeichnet, auf dem optimalen Pfad die Beobachtungsfolge $o_1,\ldots,o_t$ zu erzeugen und zum Zeitpunkt t im Zustand s_i zu sein. Zusätzlich wird eine Rückverzeigerungsmatrix $[\psi_t(j)]$ zur Extraktion der optimalen Zustandsfolge aufgebaut, da diese erst nach Abschluss der Berechnungen bei Vorliegen der gesamten Beobachtungsfolge ermittelt werden kann. Das Verfahren zur Bestimmung der optimalen Zustandsfolge, das als *Viterbi-Algorithmus* bekannt ist, lässt sich nun wie folgt skizzieren:

Zu Beginn des Verfahrens zum Zeitpunkt $t = 1$ erfolgt die Initialisierung der $\vartheta_j(i)$ und der Rückwärtszeiger $\psi_t(j)$:

$$\vartheta_1(i) = \pi_i b_i(o_1) \quad \text{und} \quad \psi_1(j) = 0. \tag{3.80}$$

Die Wahrscheinlichkeit $\vartheta_{t+1}(i)$ des jeweils folgenden Zeitpunktes lässt sich dann rekursiv bestimmen durch:

$$\vartheta_{t+1}(j) = \max_i\{\vartheta_t(i)a_{ij}\}b_j(o_{t+1}). \tag{3.81}$$

Geichzeitig wird im Rückwärtszeiger $\psi_{t+1}(j)$ für das entsprechende $\vartheta_{t+1}(j)$ der optimale Vorgängerzustand vermerkt:

$$\psi_{t+1}(j) = \underset{i}{\operatorname{argmax}}\, \vartheta_t(i)a_{ij}. \tag{3.82}$$

1. Initialisierung: $t = 1$

$$\vartheta_1(j) = \pi_j b_j(o_1) \quad \text{und} \quad \psi_1(j) = 0 \quad \forall j : j = 1, \ldots, N$$

2. Rekursion: $\forall t : t = 1, \ldots, T-1 \quad \text{und} \quad \forall j : j = 1, \ldots, N$

$$\vartheta_{t+1}(j) = \max_i \{\vartheta_t(i) a_{ij}\} b_j(o_{t+1}) \quad \text{und} \quad \psi_{t+1}(j) = \operatorname*{argmax}_i \vartheta_t(i) a_{ij}$$

3. Terminierung:

$$P(\boldsymbol{O}, \boldsymbol{q}^* | \boldsymbol{\lambda}) = \max_i \vartheta_T(i) \quad \text{und} \quad q_T^* = \operatorname*{argmax}_i \vartheta_T(i)$$

4. Rückverfolgung: $\forall t : t = T-1, \ldots, 1$

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

Abbildung 3.27: Viterbi-Algorithmus

Am Ende der Berechnungen zum Zeitpunkt T ergibt sich für die optimale Produktionswahrscheinlichkeit:

$$P(\boldsymbol{O}, \boldsymbol{q}^* | \boldsymbol{\lambda}) = \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q} | \boldsymbol{\lambda}) = \max_i \vartheta_T(i). \tag{3.83}$$

Aus der Rückverzeigerungsmatrix $[\psi_t(j)]$ lässt sich dann die zugehörige optimale Zustandsfolge $\boldsymbol{q}^*$ bestimmen. Beginnend mit dem optimalen Endzustand $q_T^* = \operatorname*{argmax}_i \vartheta_T(i)$ ergibt sie sich aus:

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{3.84}$$

In Abbildung 3.27 sind die Schritte des Viterbi-Algorithmus zusammengefasst dargestellt.

Aus Gleichung 3.81 wird deutlich, dass in jedem Rekursionsschritt N^2 Maximierungen zur Bewertung des lokal optimalen Pfades ausgeführt werden müssen. Der Viterbi-Algorithmus ist somit durch eine quadratische Komplexität in Bezug auf die Anzahl der Zustände gekennzeichnet. Zur Beschleunigung der Dekodierung werden in der Praxis daher meistens Verfahren zur Suchraumeinschränkung eingesetzt, wobei das wohl verbreitetste das sogenannte *Beam-Search* Verfahren ist.

Die Idee des Beam-Search Verfahrens ist, die Auswertung der Pfadbewertungen auf eine relativ kleine Menge *aktiver Zustände* zu beschränken, anstatt alle möglichen

Vorgängerzustände zu betrachten. Die Menge der zum Zeitpunkt t aktiven Zustände ist folgendermaßen definiert:

$$\mathcal{A}_t = \{i | \vartheta_t(i) \geq B\vartheta_t^*\} \quad \text{mit} \quad \vartheta_t^* = \max_j \vartheta_t(j). \tag{3.85}$$

Dabei bezeichnet ϑ_t^* die maximale Bewertung auf dem optimalen partiellen Pfad und der Faktor B, die sogenannte *Beambreite*, eine positive Konstante kleiner Eins. Die zum Zeitpunkt t aktiven Zustände sind also diejenigen, deren Bewertung $\vartheta_t(i)$ größer als ein Schwellwert ist, der von der maximal erzielbaren Bewertung zu diesem Zeitpunkt abhängt.

Die Gleichung 3.81 zur Bewertung des lokal optimalen Pfades wird nun dahingehend verändert, dass die Maximierung nicht mehr über alle möglichen Vorgängerzustände ausgeführt wird, sondern nur noch über die Menge der aktiven Zustände:

$$\vartheta_{t+1}(j) = \max_{i \in \mathcal{A}_t} \{\vartheta_t(i) a_{ij}\} b_j(o_{t+1}). \tag{3.86}$$

Training

Die beiden vorangegangenen Abschnitte befassten sich mit der Auswertung von HMMs, d.h. zum einen mit der Berechnung der Produktionswahrscheinlichkeit und zum anderen mit der Aufdeckung der wahrscheinlichsten Zustandsfolge, die bei der Emissionsgenerierung eingenommen wurde. Dabei wurde stets von einem gegebenen Modell ausgegangen. In diesem Abschnitt wird nun der Frage nachgegangen, wie die Parameter eines HMMs $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ automatisch anhand einer Trainingsstichprobe geschätzt werden können.

Da keine analytische Lösung zur Berechnung der HMM-Parameter bekannt ist, werden iterative Verfahren eingesetzt, die von einem initialen Modell $\boldsymbol{\lambda}^0$ ausgehen und anhand der Trainingsdaten schrittweise verbesserte Modelle $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots$ berechnen. Das wohl am häufigsten verwendete Verfahren hierzu ist der *Baum-Welch-Algorithmus*. Dieser ist eine Variante des EM-Verfahrens (siehe [Dem77]), das ein allgemeines Verfahren zur Maximum-Likelihood-Parameterschätzung stochastischer Prozesse beschreibt. Als Optimierungskriterium zur schrittweisen Schätzung der HMM-Parameter wird im Baum-Welch-Algorithmus die Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ verwendet[7]. Für das entsprechende verbesserte Modell $\widehat{\boldsymbol{\lambda}} = (\widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}})$ gilt damit:

$$P(\boldsymbol{O}|\widehat{\boldsymbol{\lambda}}) \geq P(\boldsymbol{O}|\boldsymbol{\lambda}).$$

Der Baum-Welch-Algorithmus zur Berechnung optimierter Modellparameter $\widehat{\boldsymbol{\lambda}}$ stützt sich im wesentlichen auf die Vorwärts- und Rückwärtsvariablen $\alpha_t(j)$ bzw.

[7] Bei dieser Darstellung wird davon ausgegangen, dass nur eine Beobachtungsfolge $\boldsymbol{O}$ zur Verfügung steht. In der Praxis liegen jedoch meist mehrere Beobachtungsfolgen vor, sodass dann zur Parameterschätzung über die Beobachtungsfolgen gemittelt wird.

$\beta_t(j)$, anhand derer die Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ des Ausgangsmodells bestimmt wird. Mit Hilfe der $\alpha_t(j)$ und $\beta_t(j)$ wird zunächst die Variable

$$\begin{aligned}\gamma_t(i,j) &= P(q_t = s_i, q_{t+1} = s_j|\boldsymbol{O},\boldsymbol{\lambda}) = \frac{P(q_t = s_i, q_{t+1} = s_j, \boldsymbol{O}|\boldsymbol{\lambda})}{P(\boldsymbol{O}|\boldsymbol{\lambda})} \\ &= \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \end{aligned} \tag{3.87}$$

definiert, die die a posteriori Wahrscheinlichkeit eines Übergangs $s_i \rightarrow s_j$ des Ausgangsmodells $\boldsymbol{\lambda}$ zum Zeitpunkt t bei vorliegender Observationsfolge $\boldsymbol{O}$ bezeichnet. Außerdem wird die Variable

$$\gamma_t(i) = P(q_t = s_i|\boldsymbol{O},\boldsymbol{\lambda}) = \sum_{j=1}^{N}\gamma_t(i,j) \tag{3.88}$$

definiert, die die Wahrscheinlichkeit angibt, zum Zeitpunkt t im Zustand s_i zu sein. Damit können nun die Anfangswahrscheinlichkeiten $\hat{\boldsymbol{\pi}}$, die Übergangswahrscheinlichkeiten $\widehat{\boldsymbol{A}}$ und im Falle diskreter HMMs auch die Emissionswahrscheinlichkeiten $\widehat{\boldsymbol{B}}$ des verbesserten Modells $\hat{\boldsymbol{\lambda}}$ geschätzt werden. Die Anfangswahrscheinlichkeiten ergeben sich direkt zu:

$$\hat{\pi}_i = P(q_1 = s_i|\boldsymbol{O},\boldsymbol{\lambda}) = \gamma_1(i). \tag{3.89}$$

Die erwarteten Zustandsübergangswahrscheinlichkeiten ergeben sich aus der Summierung der Einzelübergangswahrscheinlichkeiten über die Zeit und anschließender Normierung auf die Gesamtzahl der vom Zustand s_i ausgehenden Übergänge:

$$\hat{a}_{ij} = \frac{\sum\limits_{t=1}^{T-1} P(q_t = s_i, q_{t+1} = s_j|\boldsymbol{O},\boldsymbol{\lambda})}{\sum\limits_{t=1}^{T-1} P(q_t = s_i|\boldsymbol{O},\boldsymbol{\lambda})} = \frac{\sum\limits_{t=1}^{T-1}\gamma_t(i,j)}{\sum\limits_{t=1}^{T-1}\gamma_t(i)}. \tag{3.90}$$

Zur Bestimmung der diskreten Observationswahrscheinlichkeiten wird ähnlich vorgegangen. Dabei wird die Anzahl der Emissionen des betreffenden Symbols im jeweiligen Zustand über die Länge der Observationsfolge aufsummiert und anhand der Gesamtzahl der im selben Zustand generierten Emissionen normiert:

$$\hat{b}_{jk} = \frac{\sum\limits_{t=1}^{T} P(q_t = s_j, o_t = v_k|\boldsymbol{O},\boldsymbol{\lambda})}{\sum\limits_{t=1}^{T} P(q_t = s_j|\boldsymbol{O},\boldsymbol{\lambda})} = \frac{\sum\limits_{t:o_t=v_k}\gamma_t(j)}{\sum\limits_{t=1}^{T}\gamma_t(j)}. \tag{3.91}$$

Im Falle kontinuierlicher Emissionsmodellierung durch Mischverteilungen müssen dagegen sowohl die Parameter $\boldsymbol{\mu}_{jk}$ und $\boldsymbol{K}_{jk}$ der einzelnen Gaußdichten als auch die Mischungsgewichte c_{jk} geschätzt werden. Da die Mischungsgewichte als Ausgabewahrscheinlichkeiten eines diskreten HMMs interpretiert werden können, lassen sich

die c_{jk} in ähnlicher Weise bestimmen. Dazu definiert man sich die Variable $\xi_t(j,k)$, die die Wahrscheinlichkeit angibt, dass zum Zeitpunkt t im Zustand j die k-te Mischungskomponente an der Emission von o_t beteiligt war:

$$\xi_t(j,k) = P(q_t = s_j, K_t = k|\boldsymbol{O}, \boldsymbol{\lambda}) = \frac{\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} c_{jk} g_{jk}(o_t) \beta_t(j)}{P(\boldsymbol{O}|\boldsymbol{\lambda})}. \tag{3.92}$$

Die Mischungsgewichte c_{jk} ergeben sich somit aus der Summierung der $\xi_t(j,k)$ über die Zeit und Normierung auf die Gesamtzahl der im Zustand j generierten Emissionen:

$$\hat{c}_{jk} = \frac{\sum_{t=1}^{T} P(q_t = s_j, K_t = k|\boldsymbol{O}, \boldsymbol{\lambda})}{\sum_{t=1}^{T} P(q_t = s_j|\boldsymbol{O}, \boldsymbol{\lambda})} = \frac{\sum_{t=1}^{T} \xi_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{3.93}$$

Für die Bestimmung der Mittelwertvektoren und Kovarianzmatrizen der Gaußdichten muss berücksichtigt werden, dass die Observationen $\boldsymbol{x}_t$ probabilistisch mit der Wahrscheinlichkeit $\xi_t(j,k) = P(q_t = s_j, K_t = k|\boldsymbol{O}, \boldsymbol{\lambda})$ in die Berechnung eingehen. Damit gilt:

$$\widehat{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k) \boldsymbol{x}_t}{\sum_{t=1}^{T} \xi_t(j,k)} \tag{3.94}$$

$$\widehat{\boldsymbol{K}}_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k) (\boldsymbol{x}_t - \widehat{\boldsymbol{\mu}}_{jk})(\boldsymbol{x}_t - \widehat{\boldsymbol{\mu}}_{jk})^T}{\sum_{t=1}^{T} \xi_t(j,k)} \tag{3.95}$$

Wird dagegen eine semi-kontinuierliche Emissionsmodellierung verwendet, so müssen zur Berechnung der $\widehat{\boldsymbol{\mu}}_k$ und $\widehat{\boldsymbol{K}}_k$ die $\xi_t(j,k)$ durch die Randverteilungen $\xi_t(k) = \sum_{j=1}^{N} \xi_t(j,k)$ ersetzt werden.

3.6.2 Time Delay Neural Networks (TDNNs)

In diesem Abschnitt werden mit den sogenannten *Time Delay Neural Networks* (TDNNs) Vertreter der konnektionistischen Informationsverarbeitung vorgestellt, die ursprünglich zur Spracherkennung vorgeschlagen wurden [Lan90], mittlerweile aber auch in einigen Systemen zur Handschrifterkennung zum Einsatz kommen [Man94, Sen94b, Jae01]. Eine ausführlichere Beschreibung von TDNNs findet sich u.a. in [Zel97].

TDNNs gehören zur Klasse vorwärtsgerichteter Neuronaler Netze. Im Gegensatz zu MLPs, die zur Klassifikation einzelner Merkmalsvektoren fester Dimension eingesetzt werden, eignen sich TDNNs zur Erkennung von Mustern, die durch unterschiedlich

lange Folgen von Merkmalsvektoren beschrieben werden, wie es beispielsweise im Bereich der Handschrifterkennung der Fall ist.

Die Fähigkeit, Folgen von Merkmalsvektoren verarbeiten zu können, wird erreicht, indem die Neuronen durch Zeitverzögerungsglieder (engl. time delay units) erweitert werden (siehe Abbildung 3.28). Jede Eingabe wird damit über mehrere Zeitschritte entsprechend der Anzahl der Verzögerungsglieder dem Nachfolgerneuron präsentiert. Der Übersichtlichkeit halber werden zur Darstellung von TDNNs die Eingabekomponenten meistens vertikal angeordnet, während die Zeitverzögerungen horizontal dargestellt werden. Aus dieser Sichtweise können die Zeitverzögerungen auch als ein zeitliches Fenster betrachtet werden, das über die Eingabevektoren geschoben wird. Dieses Konzept ist dabei nicht ausschließlich auf die Eingabeschicht beschränkt, sondern es kann vielmehr auch auf die verdeckten Schichten übertragen werden. Die Abbildung 3.29 zeigt ein Beispiel eines mehrschichtigen TDNNs mit zwei inneren Schichten in der für TDNNs üblichen Darstellung.

TDNNs sind weiterhin dadurch gekennzeichnet, dass die Verbindungen zwischen den Schichten nicht vollständig sind. So sind z.B. die Neuronen des Eingabefensters mit den Neuronen der ersten verdeckten Schicht nicht vollständig verbunden. Um Muster unabhängig von ihrer Position in der Merkmalsvektorfolge zu erkennen, ist es vielmehr erforderlich, dass jedes Neuron der nachfolgenden Schicht nur mit einem kleinen Ausschnitt der Vorgängerschicht, dem sogenannten *rezeptiven Feld*, verknüpft ist, wobei die Gewichte der Zeitverzögerungsneuronen, also der Neuronen einer Zeile, identisch sind. Damit stellen die Neuronen einer Zeile den zeitlichen Verlauf der Aktivierung eines einzigen Neurons über die zeitliche Dauer der Vorgängerschicht dar.

Die Ausgabeneuronen integrieren schließlich die zeitlichen Ausgaben der letzten verdeckten Schicht. Indem an dieser Stelle die quadrierten Ausgaben aufsummiert werden, wird – im Vergleich zur einfachen Summe – die Ausgabe des TDNNs eher durch die Aktivierungszustände mit den größten Werten bestimmt.

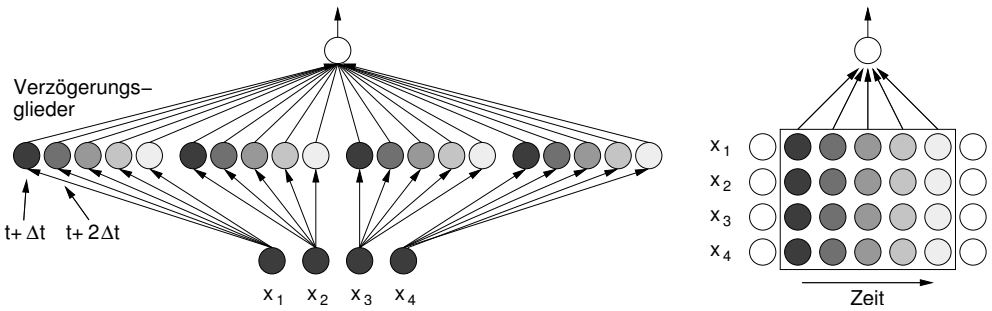


Abbildung 3.28: Schicht eines TDNN-Netzes (angelehnt an [Zel97]). Die Zeitverzögerungsglieder sind durch unterschiedliche Graustufen veranschaulicht. Rechts ist die für TDNNs übliche Darstellung gewählt, bei der die Zeitverzögerungen als ein zeitliches Fenster aufgefasst werden, das über die Eingabevektoren geschoben wird.

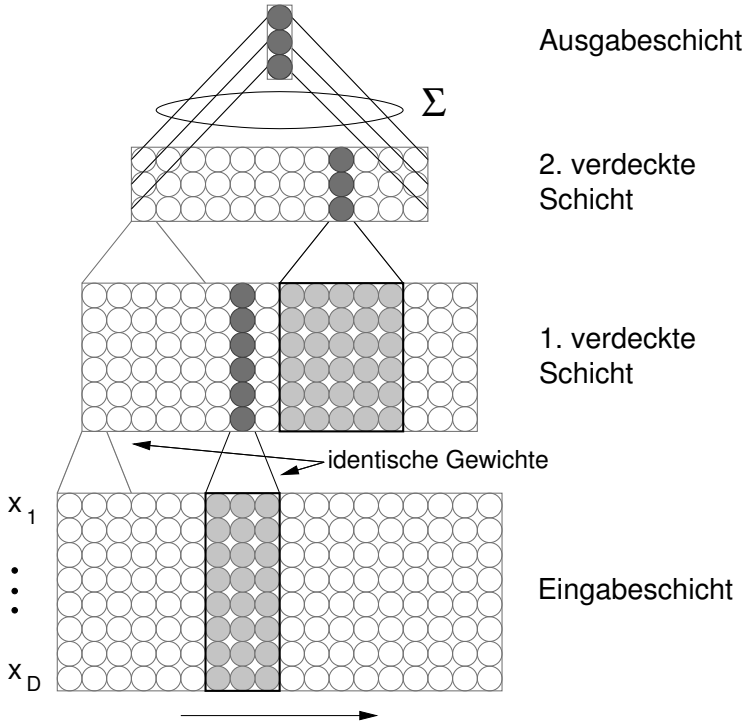


Abbildung 3.29: Mehrschichtiges TDNN. Die rezeptiven Felder der dunkel eingefärbten Neuronen sind hellgrau unterlegt. Die Ausgabeneuronen summieren die Ausgaben der letzten verdeckten Schicht über die Zeit auf.

Das Training der TDNNs kann wie bei den MLPs durch Gradientenabstiegsverfahren vorgenommen werden. Die Herleitung der Backpropagation Lernregel zur Parameteroptimierung von TDNNs kann u.a. in [Zel97] nachgelesen werden.

Die vorgestellten TDNNs sind in der Lage, einzelne Muster in einer Folge von Merkmalsvektoren zu erkennen. Häufig betrachtet man jedoch komplexe Muster, die aus einer Folge von Teilmustern aufgebaut sind. So lässt sich beispielsweise im Bereich der online Handschrifterkennung ein Buchstabe als ein komplexes Muster auffassen, das aus einer Sequenz von Strokes besteht. Ein herkömmliches TDNN würde nun lediglich eine Folge einzelner Strokes in der Merkmalsvektorfolge detektieren, es erlaubt dabei jedoch nicht unbedingt einen Rückschluss auf die entsprechenden Buchstaben. Um ein solches komplexes Muster zu erkennen, also z.B. einen aus Strokes zusammengesetzten Buchstaben, ist es somit darüberhinaus erforderlich, einen Abgleich zwischen der durch das TDNN ermittelten Folge von Strokes und den “wahren” Strokesequenzen der Buchstaben durchzuführen, sodass damit der wahrscheinlichste Buchstabe bestimmt werden kann.

Eine Möglichkeit zur Erkennung von bestimmten Mustersequenzen bieten die sogenannten *Multi-State-TDNNs* (MS-TDNNs) [Haf92]. Diese MS-TDNNs stellen eine Erweiterung herkömmlicher TDNNs um einen zusätzlichen Verarbeitungsschritt dar, in dem durch Dynamic Time Warping ein Abgleich der beobachteten Sequenz von

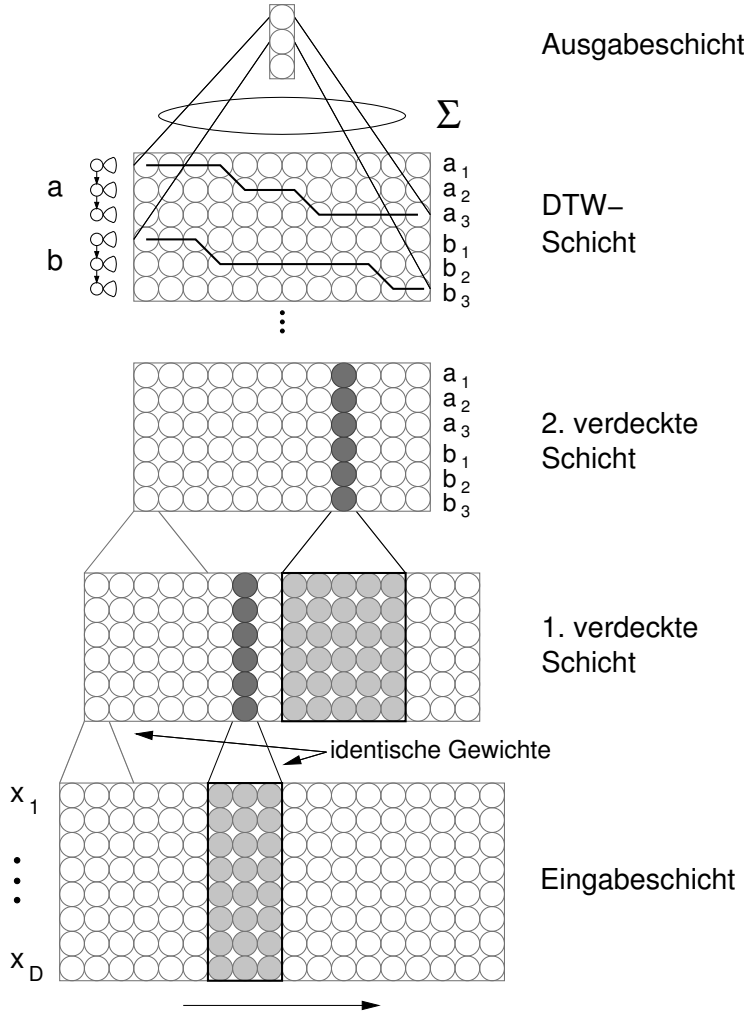


Abbildung 3.30: MS-TDNN (angelehnt an [Jae01]). Vor der Ausgabeschicht ist eine DTW-Schicht eingefügt, welche einen Abgleich der beobachteten Sequenz von Teilmustern (States) mit vorgegebenen State-Sequenzen komplexer Muster durchführt.

Teilmustern (States) mit den vorgegebenen State-Sequenzen komplexer Muster durchgeführt wird. In den Ausgabeneuronen der MS-TDNNs wird dann nicht mehr über die einzelnen Zeilen der letzten verdeckten Schicht summiert, sondern über den optimalen Pfad, der durch das Dynamic Time Warping bestimmt wurde (siehe Abbildung 3.30).

3.7 Sprachmodellierung durch n-Gramm-Modelle

Üblicherweise liegt den Systemen zur Handschrifterkennung ein vorgegebenes Lexikon von Wörtern zugrunde, welches im Vorhinein anhand des betreffenden Anwendungsszenarios bestimmt wird. Mit Hilfe des Lexikons kann so die Sequenz der vom

Erkenner erzeugten Buchstabenhypothesen auf die gültigen Wörter eingeschränkt werden. Dies kann z.B. durch Zusammenschalten der Buchstabenmodelle zu entsprechenden Verbundmodellen erreicht werden (siehe Seite 79).

Eine weitere Möglichkeit zur Vermeidung unwahrscheinlicher Buchstabensequenzen besteht in der Integration von statistischen n-Gramm Sprachmodellen in den Erkennungsprozess (siehe u.a. [Fin03]). Dabei wird die Eigenschaft von Texten ausgenutzt, dass bestimmte Buchstabenfolgen gegenüber anderen mit einer größeren Wahrscheinlichkeit vorkommen. Mit den Bezeichnungen $P(\boldsymbol{w})$ für die sprachmodellbasierte Wahrscheinlichkeit der Buchstabensequenz $\boldsymbol{w}$ und $P(\boldsymbol{x}|\boldsymbol{w})$ für die Observationswahrscheinlichkeit auf Basis der Buchstabenmodelle besteht die Erkennungsaufgabe dann darin, diejenige Buchstabenfolge $\hat{\boldsymbol{w}}$ zu finden, die die Wahrscheinlichkeit für das kombinierte Modell gemäß

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w})P(\boldsymbol{x}|\boldsymbol{w}) \tag{3.96}$$

maximiert. Die Wahrscheinlichkeit $P(\boldsymbol{w})$ der Buchstabenfolge lässt sich dabei mit Hilfe der Bayes-Regel folgendermaßen umformen:

$$P(\boldsymbol{w}) = \prod_{t=1}^{T} P(w_t|w_1, \ldots, w_{t-1}) \quad . \tag{3.97}$$

In der Praxis wird der Kontext meistens auf $n-1$ Vorgänger reduziert, häufig $n = 2$ (Bi-Gramm) oder $n = 3$ (Tri-Gramm), sodass sich näherungsweise folgender Zusammenhang ergibt:

$$P(\boldsymbol{w}) \approx \prod_{t=1}^{T} P(w_t|w_{t-n+1}, \ldots, w_{t-1}) \quad . \tag{3.98}$$

Der Buchstabe w_t bildet zusammen mit seinen $n-1$ Vorgängern ein n-Tupel, das damit als n-Gramm bezeichnet wird. Ein statistisches n-Gramm Modell beschreibt somit die Wahrscheinlichkeit für das Auftreten eines Buchstabens im Kontext von $n-1$ vorangegangenen Buchstaben. Man spricht dabei auch von der Wahrscheinlichkeit für das *Ereignis* $(w_1, w_2, \ldots, w_n)$:

$$P(w_n|w_1, w_2, \ldots, w_{n-1}) \quad .$$

Die n-Gramm Wahrscheinlichkeiten lassen sich prinzipiell direkt aus einer Trainingsstichprobe anhand der Vorkommenshäufigkeiten der n-Tupel und der möglichen Kontexte $w_1, \ldots, w_{n-1}$ bestimmen.

$$P(w_n|w_1, w_2, \ldots, w_{n-1}) := f(w_1, w_2, \ldots, w_n) = \frac{c(w_1, w_2, \ldots, w_n)}{c(w_1, w_2, \ldots, w_{n-1})} \quad . \tag{3.99}$$

Dabei tritt jedoch das Problem auf, dass nicht alle theoretisch möglichen n-Gramme in der Trainingsstichprobe repräsentiert sind. Wendet man das Modell dann auf Testdaten an, die ein im Training nicht beobachtetes Tupel aufweisen, so ergibt sich für dieses

die Wahrscheinlichkeit Null. Aus diesem Grund werden die Wahrscheinlichkeitsverteilungen, die sich aus dem Auszählen der Vorkommenshäufigkeiten ergeben haben, nachbearbeitet, um auch nicht beobachteten Ereignissen robuste Auftrittswahrscheinlichkeiten zuzuweisen.

Die Nachbearbeitung gliedert sich üblicherweise in zwei Schritte. Im ersten Schritt wird eine Umverteilung von Wahrscheinlichkeitsmasse von beobachteten auf unbeobachtete Ereignisse vorgenommen. Zur Gewinnung von Wahrscheinlichkeitsmasse werden die empirischen Häufigkeiten beobachteter Ereignisse vermindert, bei dem populären Verfahren des *absolute discounting* beispielsweise um einen konstanten Betrag β. Die veränderte relative Häufigkeit ergibt sich damit zu:

$$f^*(w_1, w_2, \ldots, w_n) = \frac{c(w_1, w_2, \ldots, w_n) - \beta}{c(w_1, w_2, \ldots, w_{n-1})} \quad . \tag{3.100}$$

Im zweiten Schritt wird die so gewonnene Wahrscheinlichkeitsmasse verwendet, um durch Einbeziehung allgemeinerer Verteilungen robuste Schätzwerte für nicht beobachtete Ereignisse zu bestimmen. Eine allgemeinere Verteilung ergibt sich dabei i.d.R durch Kürzung des n-Gramm Kontextes, sodass beispielsweise ein Tri-Gramm auf ein Bi-Gramm reduziert wird. Bei der Methode des *backing off* wird die allgemeinere Verteilung dann mit in die Berechnung der n-Gramm Wahrscheinlichkeit einbezogen, wenn die verminderte relative Häufigkeit $f^*(w_1, w_2, \ldots, w_n)$ verschwindet. Die Umverteilung der im ersten Schritt gewonnenen Wahrscheinlichkeitsmasse erfolgt dabei proportional zur allgemeineren Verteilung.

3.8 Adaptionsverfahren bei HMM-basierten Systemen

Erkennungssysteme, die unabhängig von einem bestimmten Schreiber oder einer konkreten Anwendungssituation trainiert wurden, bleiben i.d.R. in ihrer Erkennungsleistung hinter den Systemen zurück, die speziell für bestimmte Bedingungen optimiert wurden. Während allgemeine Systeme in vielfältigen Anwendungsfällen befriedigende Leistungen erzielen, kann mit einem speziellen System allerdings nur unter den dafür vorgesehenen Bedingungen eine gute Erkennungsleistung erreicht werden, unter geringfügig abweichenden Bedingungen verschlechtern sich die Ergebnisse dagegen häufig drastisch.

Um die Robustheit und Flexibilität allgemeiner Systeme hinsichtlich ihrer Anwendbarkeit in unterschiedlichen Situationen beizubehalten und gleichzeitig die Erkennungsleistung spezieller Systeme unter den entsprechenden Bedingungen zu erzielen, werden üblicherweise Adaptionsverfahren eingesetzt. Damit können die Parameter allgemeiner Systeme an einen bestimmten Schreiber bzw. eine konkreten Anwendungssituation angepasst werden, sodass im günstigsten Fall die Ergebnisse spezieller Systeme erreicht werden.

Adaptionsverfahren bei HMM-basierten Systemen können auf unterschiedlichen Ebenen des Erkennungssystems eingesetzt werden. So kann beispielsweise mit Hil-

fe von Normalisierungsverfahren eine Merkmalsrepräsentation angestrebt werden, die unabhängig von speziellen Anwendungsbedingungen ist. Eine weitere Möglichkeit ist die Adaption der HMM Parameter. Hier haben sich vor allem Verfahren durchgesetzt, die die Adaption auf Basis der Emissionsmodellierung vornehmen, wobei die Topologie und die Übergangswahrscheinlichkeiten der HMMs unverändert bleiben. Zwei verbreitete Adaptionsverfahren werden im folgenden näher erläutert. Die Darstellung ist dabei angelehnt an die Arbeiten [Leg95] und [Fin03], Seite 177ff.

3.8.1 Maximum Likelihood Linear Regression

Das Maximum Likelihood Linear Regression (MLLR) Verfahren adaptiert anhand einer Adaptionsstichprobe die Parameter der auf Gaußdichten basierenden Emissionsmodellierung, wobei das Optimierungskriterium wie beim Baum-Welch Training die Maximierung der Produktionswahrscheinlichkeit ist. Üblicherweise werden bei der MLLR-Adaption jedoch nur die Mittelwertvektoren der Gaußdichten transformiert, die Kovarianzmatrizen bleiben meistens unberücksichtigt.

Durch die Definition von Regressionsklassen kommt die MLLR-Adaption mit einer relativ kleinen Adaptionsstichprobe aus. In diesen Regressionsklassen werden jeweils mehrere Gaußdichten zusammengefasst, sodass auch die Mittelwertvektoren derjenigen Gaußdichten transformiert werden, für die keine Adaptionsdaten vorliegen. Die Definition der Regressionsklassen kann dabei rein datengetrieben vorgenommen werden, beispielsweise anhand des Abstands zwischen den Mittelwertvektoren der einzelnen Gaußdichten.

Die Adaption des Mittelwertvektors der Gaußdichte des Zustands s wird durch eine lineare Transformation vorgenommen[8]:

$$\boldsymbol{\mu}'_s = \boldsymbol{W}_s \hat{\boldsymbol{\mu}}_s \quad , \quad \hat{\boldsymbol{\mu}}_s = [\omega_s, \mu_{1_s}, \ldots, \mu_{d_s}]^T \quad . \tag{3.101}$$

Dabei bezeichnet $\boldsymbol{W}_s$ eine $d \times (d+1)$ dimensionale Transformationsmatrix und $\hat{\boldsymbol{\mu}}_s$ den um den Regressionsoffset ω_s (üblicherweise $\omega_s = 1$) erweiterten ursprünglichen Mittelwertvektor.

Die Berechnung der Transformationsmatrix $\boldsymbol{W}_s$ für den Mittelwertvektor der Gaußdichte des Zustands s erfolgt nach der Maximum-Likelihood Methode durch Maximierung der Produktionswahrscheinlichkeit. Dazu wird die folgende Hilfsfunktion definiert,

$$Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \sum_{\boldsymbol{q} \in \Theta^T} P(\boldsymbol{O}, \boldsymbol{q}|\boldsymbol{\lambda}) \log(P(\boldsymbol{O}, \boldsymbol{q}|\boldsymbol{\lambda}')) \tag{3.102}$$

bei der $\boldsymbol{\lambda}$ bzw. $\boldsymbol{\lambda}'$ die HMM-Parameter zweier aufeinanderfolgender Iterationen der Maximierung darstellen und Θ^T die Menge aller Zustandsfolgen der Länge T bezeich-

[8] Für die Herleitung der Transformationsmatrix wird im folgenden von einer unimodalen Emissionsmodellierung ausgegangen, sodass also jedem HMM Zustand genau eine Gaußdichte zugeordnet ist.

net. Durch Einsetzen der Produktionswahrscheinlichkeit, Ableiten und Nullsetzen ergibt sich die folgende Gleichung zur Schätzung der Transformationsmatrix $\widehat{\boldsymbol{W}}_s$:

$$\sum_{t=1}^{T} \gamma_t(s) \boldsymbol{K}_s^{-1} \boldsymbol{x}_t \hat{\boldsymbol{\mu}}_s^T = \sum_{t=1}^{T} \gamma_t(s) \boldsymbol{K}_s^{-1} \widehat{\boldsymbol{W}}_s \hat{\boldsymbol{\mu}}_s \hat{\boldsymbol{\mu}}_s^T \quad . \tag{3.103}$$

Dabei bezeichnet $\gamma_t(s)$ die Wahrscheinlichkeit, zum Zeitpunkt t im Zustand s zu sein, $\boldsymbol{K}_s$ die Kovarianzmatrix der Gaußdichte und $\boldsymbol{x}_t$ die Observation zum Zeitpunkt t.

Werden die Dichten mehrerer Zustände zu einer Regressionsklasse zusammengefasst, so muss zur Schätzung der Transformationsmatrix über die entsprechenden Zustände summiert werden:

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_t(s_r) \boldsymbol{K}_{s_r}^{-1} \boldsymbol{x}_t \hat{\boldsymbol{\mu}}_{s_r}^T = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_t(s_r) \boldsymbol{K}_{s_r}^{-1} \widehat{\boldsymbol{W}}_s \hat{\boldsymbol{\mu}}_{s_r} \hat{\boldsymbol{\mu}}_{s_r}^T \quad . \tag{3.104}$$

In der obigen Gleichung bezeichnet R die Anzahl der Zustände der Regressionsklasse.

Eine Vereinfachung ergibt sich unter der Annahme identischer Kovarianzmatrizen aller einer Regressionsklasse zugeordneter Gaußdichten. In diesem Fall verschwinden die Kovarianzmatrizen aus Gleichung 3.104:

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_t(s_r) \boldsymbol{x}_t \hat{\boldsymbol{\mu}}_{s_r}^T = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_t(s_r) \widehat{\boldsymbol{W}}_s \hat{\boldsymbol{\mu}}_{s_r} \hat{\boldsymbol{\mu}}_{s_r}^T \quad . \tag{3.105}$$

Nimmt man außerdem eine *eindeutige* Zuordnung der Observationen $\boldsymbol{x}_t$ zu den HMM-Zuständen an (beispielsweise durch den Viterbi-Algorithmus), so ergibt sich:

$$\begin{aligned} \sum_{t=1}^{T} \boldsymbol{x}_t \hat{\boldsymbol{\mu}}_{q_t}^T \delta_{s,q_t} &= \widehat{\boldsymbol{W}}_s \sum_{t=1}^{T} \hat{\boldsymbol{\mu}}_{q_t} \hat{\boldsymbol{\mu}}_{q_t}^T \delta_{s,q_t} \\ \delta_{s,q_t} &= \begin{cases} 1, \text{ wenn } q_t \in \{s_1, \ldots, s_R\} \\ 0, \text{ sonst} \end{cases} \end{aligned} \tag{3.106}$$

Definiert man die Matrizen

$$\begin{aligned} \boldsymbol{X} &= [\hat{\boldsymbol{\mu}}_{q_1}, \ldots, \hat{\boldsymbol{\mu}}_{q_t}] \\ \boldsymbol{Y} &= [\boldsymbol{x}_1 \delta_{s,q_1}, \ldots, \boldsymbol{x}_t \delta_{s,q_t}] \quad , \end{aligned}$$

so kann die Gleichung 3.106 wie folgt geschrieben werden:

$$\boldsymbol{Y}\boldsymbol{X}^T = \widehat{\boldsymbol{W}}_s \boldsymbol{X}\boldsymbol{X}^T \quad .$$

Die Matrix $\widehat{\boldsymbol{W}}_s$ ergibt sich folglich zu:

$$\widehat{\boldsymbol{W}}_s = \boldsymbol{Y}\boldsymbol{X}^T (\boldsymbol{X}\boldsymbol{X}^T)^{-1} \quad .$$

Dieses Vorgehen entspricht damit exakt der Methode der Minimierung des mittleren quadratischen Fehlers.

3.8.2 MAP Adaption

Wie bereits erläutert, stellt die Bayes-Regel die Grundlage der statistischen Klassifikation dar (vgl. Seite 70):

$$P(\boldsymbol{\lambda}|\boldsymbol{O}) = \frac{P(\boldsymbol{O}|\boldsymbol{\lambda})P(\boldsymbol{\lambda})}{P(\boldsymbol{O})} \quad .$$

Liegt keinerlei Vorwissen über die a priori Wahrscheinlichkeit $P(\boldsymbol{\lambda})$ vor, so wird diese als konstant angenommen und die Parameterschätzung des HMMs erfolgt anhand der Produktionswahrscheinlichkeit $P(\boldsymbol{O}|\boldsymbol{\lambda})$ (Maximum Likelihood Schätzung).

Ist dagegen die a priori Wahrscheinlichkeit gegeben, so kann das Training der HMM Parameter anhand der a posteriori Wahrscheinlichkeit $P(\boldsymbol{\lambda}|\boldsymbol{O})$ vorgenommen werden (MAP Schätzung):

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\boldsymbol{\lambda})P(\boldsymbol{\lambda}) \quad .$$

Üblicherweise ist vor dem Training des Klassifikators die a priori Wahrscheinlichkeit allerdings nicht bekannt. Im Falle der Parameteradaption eines bestehenden Klassifikators kann die a priori Wahrscheinlichkeit jedoch auf Basis der Parameter $\boldsymbol{\lambda}$ des zu adaptierenden HMMs bestimmt werden. So ergibt sich beispielsweise für die MAP-Adaption des Mittelwertvektors der k-ten Komponente der Mischverteilung im Zustand s des HMMs durch Anwendung des EM-Algorithmus folgende Berechnungsvorschrift (für eine ausführliche Herleitung siehe z.B. [Gau92]):

$$\boldsymbol{\mu}'_{sk} = \frac{\tau\boldsymbol{\mu}_{sk} + \sum_{t=1}^{T} \xi_t(s,k)\boldsymbol{x}_t}{\tau + \sum_{t=1}^{T} \xi_t(s,k)} \quad . \tag{3.107}$$

Hierbei bezeichnet $\boldsymbol{\mu}_{sk}$ den Mittelwertvektor der k-ten Mischungskomponente des Zustands s des zu adaptierenden HMMs und $\xi_t(s,k)$ die in Gleichung 3.92 definierte Wahrscheinlichkeit, mit der zum Zeitpunkt t im Zustand s die k-te Mischungskomponente an der Emission von $\boldsymbol{x}_t$ beteiligt war. Mit dem Parameter τ kann eine Gewichtung zwischen dem a priori Mittelwert $\boldsymbol{\mu}_{sk}$ ($\tau \to \infty$) und dem ausschließlich auf Basis der Adaptionsdaten berechneten Mittelwert $\widehat{\boldsymbol{\mu}}_{sk}$ ($\tau = 0$, vgl. Gleichung 3.94) vorgenommen werden.

$$\widehat{\boldsymbol{\mu}}_{sk} = \frac{\sum_{t=1}^{T} \xi_t(s,k)\boldsymbol{x}_t}{\sum_{t=1}^{T} \xi_t(s,k)}$$

Der Fall $\tau = 0$ entspricht damit einer Maximum Likelihood Schätzung, wenn also kein a priori Wissen vorliegt.

Ein Nachteil der MAP-Adaption im Vergleich zum MLLR-Verfahren ist, dass eine umfangreichere Adaptionsstichprobe benötigt wird. Dies liegt vor allem daran, dass bei der MAP-Methode für alle zu adaptierenden Gaußdichten Daten vorhanden sein

müssen, während beim MLLR-Verfahren durch die Definition von Regressionsklassen *globale* Transformationen geschätzt werden können, so dass auch diejenigen Gaußdichten adaptiert werden können, für die keine Observationen vorliegen. Ist jedoch für die MAP-Adaption eine ausreichend große Adaptionsstichprobe verfügbar, so kann gegenüber dem MLLR-Verfahren eine asymptotisch bessere Erkennungsleistung erzielt werden, da die Parameter der einzelnen Gaußdichten *individuell* adaptiert werden können [Fis99].

3.9 Stichprobendatenbanken

Eine wesentliche Grundvoraussetzung für die Anwendbarkeit statistischer Erkennungssysteme ist die Verfügbarkeit möglichst umfangreicher Stichproben für das Training und die Evaluation des Klassifikators. Um außerdem die Erkennungsraten unterschiedlicher Systeme miteinander vergleichen zu können, ist es vorteilhaft, wenn standardisierte Stichprobendatenbanken verwendet werden [Guy97, Mar02].

Im Gegensatz zum Gebiet der Spracherkennung, wo schon seit langem standardisierte Trainings- und Evaluationsstichproben eingesetzt werden, basieren die Systeme zur Handschrifterkennung oftmals auf eigens für die Entwicklung des jeweiligen Systems aufgenommenen Daten. Damit ist die Vergleichbarkeit der Erkennungsergebnisse unterschiedlicher Systeme jedoch kaum gegeben. Erst seit relativ kurzer Zeit gibt es auch im Handschriftbereich Bemühungen, geeignete Stichproben aufzunehmen und zu verbreiten, sodass sich auch hier Standards etablieren können.

3.9.1 Offline Bereich

Einige Handschriftstichproben, die im Bereich der offline Erkennung Verwendung finden, stammen u.a. von den Datenbanken, die von den Instituten CEDAR[9], NIST[10] und CENPARMI[11] verbreitet werden. Diese Datenbanken enthalten größtenteils jedoch nur isolierte Zeichen oder einzelne Wörter und sind damit für die Entwicklung von Systemen zur Erkennung uneingeschränkter handschriftlicher Texte nur bedingt geeignet.

Eine Datenbank, die handschriftliche englische Texte enthält, ist in [Sen98] beschrieben. Diese Datenbank enthält insgesamt 25 eingescannte Seiten handgeschriebener und annotierter Sätze, wobei ein Vokabular von ca. 1300 Wörtern verwendet wurde. Die Texte wurden dem Lancaster-Oslo/Bergen Korpus (LOB) entnommen, einer Sammlung englischsprachiger Texte [Joh78]. Da die Handschriftstichprobe jedoch nur durch einen einzigen Schreiber erstellt wurde, ist diese Datenbank für die Realisierung schreiberunabhängiger Systeme nicht geeignet.

[9] Center of Excellence for Document Analysis and Recognition, University at Buffalo, State University of New York.

[10] National Institute of Standards and Technology, Gaithersburg, Md., USA.

[11] Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Canada.

Um die Entwicklung schreiberunabhängiger Systeme zur Erkennung uneingeschränkter Handschrift zu unterstützen, wurde am Institut für Informatik und angewandte Mathematik der Universität Bern die IAM-Datenbank erstellt, die ebenfalls auf handschriftlichen Texten aus dem LOB Korpus basiert [Mar99, Mar02]. Insgesamt haben bisher ca. 500 Schreiber zur Erstellung der Datenbank beigetragen, die zur Zeit etwa 1500 Textseiten (ca. 10000 annotierte Zeilen) umfasst. Das verwendete Vokabular besteht aus über 10000 Wörtern. Die IAM-Datenbank ist zu Forschungszwecken frei erhältlich und wird mittlerweile von einigen Forschergruppen benutzt (u.a. [Vin02b, Kav02]).

3.9.2 Online Bereich

Der Bereich der online Handschrifterkennung war lange Zeit das "Stiefkind" der Mustererkennung in Bezug auf standardisierte, öffentlich verfügbaren Stichprobendatenbanken[Guy97]. Um diesen Missstand zu beheben, wurde im September 1992 das UNIPEN-Projekt ins Leben gerufen, das den Aufbau einer großen, öffentlich zugänglichen Datenbank zur online Handschrifterkennung zum Ziel hatte[Guy94]. Zahlreiche Arbeitsgruppen aus industrieller und öffentlicher Forschung haben dabei einen gemeinsamen Standard zum Datenaustausch, das sogenannte UNIPEN-Format, entwickelt und Stichprobendaten zur Verfügung gestellt. Für die Datenaufnahme sind unterschiedlichste Digitalisiertabletts verwendet worden, sodass die Qualität der Daten stark variiert. Leider ist auch die Annotierung der Daten oftmals fehlerhaft. Mittlerweile ist ein Ausschnitt der Datenbank, der sogenannte `Train_r01_v07` Teilbereich, auf einer CD-ROM erschienen (siehe `http://unipen.nici.kun.nl/cdroms/`).

3.9.3 Online+Offline

Eine Stichprobendatenbank, die sowohl online als auch offline Informationen enthält, ist die Ireste On/Off – kurz IRONOFF – Datenbank[VG99]. Sie besteht aus isolierten Zahlen, Buchstaben und Wörtern, wobei die Wortstichprobe englisch- und französischsprachige Wörter umfasst. Die Besonderheit bei dieser Datenbank ist, dass sowohl die Stifttrajektorie mit Hilfe eines Digitalisiertabletts als auch das fertige Bild desselben Schriftzugs mit Hilfe eines Scanners aufgenommen wurde. Da somit die Schreibdynamik sowie das statische Grauwertbild zur Verfügung stehen, ist diese Stichprobendatenbank daher besonders für Untersuchungen zur Integration von online und offline Erkennung geeignet. Außerdem kann sie verwendet werden, um die Rekonstruktion der dynamischen Bewegungsinformation anhand des statischen Schriftbildes zu bewerten. Da die Datenbank jedoch größtenteils isolierte Zahlen und Buchstaben enthält, und das Vokabular der Wortstichprobe mit 197 recht eingeschränkt ist, ist sie für Experimente zur Erkennung handschriftlicher Texte kaum einsetzbar.

3.10 Zusammenfassung

In diesem Kapitel sind die Grundlagen und der gegenwärtige Stand der Forschung der automatischen Handschrifterkennung vorgestellt worden. Dabei wurde sowohl auf online Systeme eingegangen, die auf den dynamischen Bewegungsinformationen des Schreibprozesses basieren, als auch auf offline Systeme, die das statische Schriftbild für die Erkennung nutzen.

Die üblicherweise eingesetzten Verarbeitungsschritte von der Signalaufnahme bis hin zur Klassifikation und Adaption wurden erläutert. Besonderes Augenmerk wurde dabei auf die Schritte zur Signalvorverarbeitung gelegt, da diese Verfahren zur Verbesserung der Signalqualität und Kompensation der schreiber- und situationsspezifischen Variabilität der Schrift einen großen Einfluss auf die Erkennungsleistung der Systeme besitzen.

Für die bei analytischen Systemen erforderliche Segmentierung des Signals in Untereinheiten lässt sich sagen, dass sich die implizite Segmentierungsstrategie und Übersegmentierungsverfahren durchgesetzt haben. Die endgültige Segmentierung wird damit erst im Zuge der Klassifikation ermittelt, so dass frühe unumkehrbare Segmentierungentscheidungen vermieden werden können. Für die Merkmalsrepräsentation der Schrift hat sich bisher kein fester Satz von Merkmalen etabliert. Häufig wird von einer großen Anzahl heuristisch gewählter Merkmale ausgegangen und durch Optimierungsverfahren, wie der Hauptkomponenten- oder linearen Diskriminanzanalyse, ein kompakter und möglichst diskriminativer Satz von Merkmalen erzeugt. Die bevorzugten Verfahren zur Klassifikation der Merkmalsvektorfolgen basieren auf Hidden Markov Modellen oder Time Delay Neural Networks. Insbesondere die HMMs haben sich dabei auf Grund der hervorragenden Modellierungseigenschaften von Beobachtungsfolgen unterschiedlicher Länge für die Handschrifterkennung als geeignet erwiesen.

4 Handschrifterkennung mittels videobasierter Sensorik

Die Systeme, die zum gegenwärtigen Stand zur automatischen Handschrifterkennung eingesetzt werden, verwenden nahezu ausschließlich spezielle Hardware zur Signalaufnahme. So werden fast ausnahmslos für die online Erkennung Digitalisiertabletts und für die offline Erkennung Scanner eingesetzt. Aufgrund der Nachteile, die diese Sensoren als Mensch-Maschine Schnittstelle aufweisen, schlugen einige Forscher vor, die Signalaufnahme stattdessen mit Hilfe von Videokameras durchzuführen (u.a. [Mun96, SF96]).

Die Verwendung einer Videokamera zur Signalaufnahme stellt jedoch auch spezielle Anforderungen an die nachfolgenden Verarbeitungsschritte des Handschrifterkennungssystems. Dabei sind insbesondere für die online Handschrifterkennung zusätzliche Verarbeitungsschritte erforderlich, um anhand der aufgenommenen Bildsequenz die Stifttrajektorie zu extrahieren.

4.1 Motivation

Wie bereits in der Einleitung dieser Arbeit angeklungen ist, schränken die spezialisierten Sensoren zur Signalaufnahme wie Digitalisiertabletts oder Scanner, die in den herkömmlichen Systemen zur Handschrifterkennung eingesetzt werden, die Anwendungsmöglichkeiten dieser Systeme deutlich ein. So erfordern beispielsweise die bei der online Verarbeitung üblicherweise eingesetzten, auf elektromagnetischer Resonanz basierenden Digitalisiertabletts die Verwendung spezieller Stifte, sodass in dieser Hinsicht das Ziel einer möglichst *natürlichen Eingabeschnittstelle* kaum erreicht ist. Demgegenüber erlauben die im offline Bereich zur Signalaufnahme eingesetzten Scanner zwar das Schreiben mit einem normalen Stift auf normalem Papier, als Eingabeschnittstelle zur Mensch-Maschine Kommunikationen sind Scanner allerdings nur bedingt geeignet. Da hier das gesamte handschriftliche Dokument erst nach der Fertigstellung eingescannt und entsprechend weiterverarbeitet wird, sind die *Interaktionsmöglichkeiten* mit dem Rechner, z.B. das Anbringen von Korrekturen, stark eingeschränkt.

Diese Nachteile können durch den Einsatz von Videokameras zur Signalaufnahme weitestgehend vermieden werden. Videokameras bieten eine natürliche Form der Eingabeschnittstelle, da mit einem beliebigen Stift auf einem beliebigen Untergrund, d.h. beispielsweise auch an Whiteboards, geschrieben werden kann. Durch die fortwährende Beobachtung des Schreib*prozesses* können außerdem die Erkennungsergebnisse

schritthaltend mit dem Schreibvorgang generiert werden, sodass dem Benutzer die Möglichkeit geboten wird, interaktiv mit dem System zu agieren.

Videobasierte Handschrifterkennungssysteme bestehen bisher jedoch eher in Ansätzen. Die Signalaufnahme mittels Videokamera und die Extraktion der Stifttrajektorie für die *online* Handschrifterkennung wird in den Arbeiten [Mun02, Bun99] behandelt. Während in [Mun02] dabei ausschließlich auf die Erfassung der Stifttrajektorie eingegangen wird, werden in [Bun99] anhand der extrahierten Trajektorie auch Erkennungsexperimente durchgeführt, wobei dazu allerdings ein bereits bestehendes Klassifikationsmodul eingesetzt wird. Das Hauptaugenmerk liegt in beiden Arbeiten jedoch eindeutig auf den Methoden zur Extraktion der Stifttrajektorie (siehe Abschnitt 4.3), geeignete Verfahren zur Weiterverarbeitung und Erkennung werden nicht thematisiert.

Die bestehenden Systeme zur videobasierten *offline* Erkennung sind i.d.R. auf maschinengeschriebene Dokumente eingeschränkt [Li00, Mir01, Cla02]. Der Fokus dieser Arbeiten liegt meist auf der Extraktion und Gruppierung der Textregionen, wohingegen zur Erkennung dann häufig eine gegebene OCR-Software eingesetzt wird.

Weitere videobasierte Systeme sind außerdem als sogenannte *whiteboard scanner* entwickelt worden, die der wachsenden Popularität von Whiteboards in Büro- und Besprechungsräumen Rechnung tragen [SF96, Bla98, Sau99]. Mit diesen Systemen wird allerdings keine Schrifterkennung durchgeführt. Die Erkennungsaufgabe beschränkt sich dabei auf einige Symbole, die vom Benutzer an das Whiteboard geschrieben werden können, um bestimmte Aktionen zu veranlassen, wie z.B. das Ausdrucken des Tafelinhalts oder das “Ausschneiden” bestimmter Bereiche.

4.2 Anforderungen

Dass trotz der Argumente für den Einsatz videobasierter Sensorik zur Handschrifterkennung solche Systeme bisher nur in Ansätzen realisiert wurden, liegt vor allem an der geringeren Signalqualität von Videokameras im Vergleich zu Scannern bzw. Digitalisiertabletts. Die Faktoren, die beim Einsatz von Videokameras zu der geringeren Signalqualität führen, sowie die Anforderungen, die sich daraus für die Verarbeitungsschritte in videobasierten Systemen ergeben, werden im folgenden näher erläutert.

Videokamera vs. Scanner

Die Grauwertverteilung der mit einer Videokamera aufgenommenen Bilder ist aufgrund von Schattenwurf und schwankenden Beleuchtungsbedingungen oftmals ungleichmäßig. Dies kann sogar dazu führen, dass in schwach beleuchteten Bildregionen der Hintergrund eine niedrigere Intensität aufweist, als die Schrift in stärker beleuchteten Regionen. Diese Schwierigkeit tritt bei der Verwendung von Scannern zur offline Handschrifterkennung nicht auf, da diese über eine eigene, äußerst homogene Beleuchtung verfügen.

Scanner bieten darüberhinaus den Vorteil, dass die räumliche Auflösung, die üblicherweise bei mindestens 300 dpi liegt, deutlich höher ist als bei einer herkömmlichen, der PAL-Norm entsprechenden Videokamera. Diese erreicht bei der Aufnahme einer DIN A4 Seite mit einer Standard-Bildgröße von 756×576 eine Auflösung von maximal 70 dpi. Durch die geringere Auflösung können damit feinere Strukturen der Schrift nur grob abgebildet werden, sodass oftmals bei den Schriftkonturen ein "Zerfasern" festzustellen ist. Ein weiterer Punkt ist, dass die aufnahmebedingten geometrischen Verzerrungen, die bei einem Scanner auftreten können, weniger stark ausgeprägt sind als bei der Verwendung einer Videokamera. Während bei einem Scanner das Dokument i.d.R. nur rotiert oder translatiert ist, können bei einer Videokamera darüberhinaus noch Linsenverzeichnungen oder perspektivische Verzerrungen bedingt durch den Aufnahmewinkel auftreten.

Um also an Stelle eines Scanners eine Videokamera für die offline Handschrifterkennung einsetzen zu können, müssen die zur Bildvorverarbeitung und Merkmalsextraktion verwendeten Verfahren robust in Bezug auf die geringere Signalqualität sein. Neben der auf Grund der geringeren Auflösung erforderlichen Glättung der Bilder sind hierbei vor allem adaptive Verfahren zur Binarisierung notwendig, um ungleichmäßige Beleuchtungsbedingungen zu kompensieren.

Im Gegensatz zu einem Scanner, der nach Beendigung des Schreibvorgangs *ein Bild* des gesamten Dokuments "auf Knopfdruck" aufnimmt, kann mit einer Videokamera eine *Bildsequenz* des gesamten Schreibprozesses beobachtet werden, sodass eine automatische, mit dem Schreibprozess schritthaltende, Erkennung realisiert werden kann. Da die Erkennungsergebnisse somit vergleichsweise zeitnah vorliegen wird eine erhöhte Interaktivität der Mensch-Maschine Schnittstelle erreicht. Dazu ist es allerdings erforderlich, zu jedem Zeitschritt die jeweils im aktuell aufgenommenen Bild neu hinzugekommmenen Textregionen zu detektieren, sodass der enthaltene Schriftabschnitt klassifiziert und zum Gesamterkennungsergebnis schrittweise integriert werden kann.

Videokamera vs. Digitalisiertablett

Um dagegen die Dynamik der Schreibbewegung für die Klassifikation zu nutzen, also eine online Erkennung im Sinne der Nomenklatur der Handschrifterkennung durchzuführen, ist es erforderlich, zu jedem Zeitschritt die Position des Stiftes zu bestimmen. Im Gegensatz zur Verwendung von Digitalisiertabletts ist bei der videobasierten online Erkennung zur Extraktion der Stifttrajektorie jedoch eine Reihe zusätzlicher Verarbeitungsschritte notwendig.

So muss zuerst anhand der Bildfolge detektiert werden, wann der Schreibprozess beginnt. Diese Initialisierung ist bei den Digitalisiertabletts trivial, da sie erst dann Koordinaten übermitteln, wenn der Stift in die Nähe der Tablettoberfläche kommt. Neben der Bestimmung der Stiftkoordinaten können die üblicherweise eingesetzten Digitalisiertabletts außerdem die pen-up/down Informationen, also ob der Stift aufgesetzt ist oder sich knapp über der Oberfläche befindet, direkt ermitteln. Bei der videobasier-

ten Vorgehensweise müssen dagegen Bildverarbeitungsverfahren eingesetzt werden, um während des Schreibprozesses anhand des zum jeweiligen Abtastzeitpunkt aufgenommenen Bildes die Position des Stifts und die pen-up/down Informationen zu bestimmen. Daraus folgt unmittelbar eine wesentliche Voraussetzung speziell für videobasierte *online* Systeme, dass nämlich der Stift während des Schreibprozesses stets in den aufgenommenen Bildern sichtbar sein muss.

Darüberhinaus sind die räumliche Auflösung und die Abtastfrequenz von Digitalisiertabletts im Vergleich zu handelsüblichen Videokameras höher. Moderne Tabletts erreichen Auflösungen von bis zu 2500 Linien pro Zoll und Abtastraten von mehr als 200 Hz. Bei Videokameras liegt die räumliche Auflösung dagegen bei den schon angesprochenen 70 dpi und die Abtastfrequenz beträgt im Interlace-Verfahren 50 Hz. Aus diesen Gründen ist die resultierende Trajektorie in videobasierten Systemen weniger glatt, sodass eine stärkere Glättung und robuste Verfahren zur Weiterverarbeitung erforderlich sind.

Speicheraufwand

Eine weitere Schwierigkeit bei videobasierten Systemen ergibt sich daraus, dass das Aufnehmen von Trainingsmaterial für die Lernphase des Klassifikators vergleichsweise aufwendig ist. So fällt bei der Aufnahme einer einminütigen Bildsequenz in Interlace Technik (50 Bilder pro Sekunde) mit der Auflösung von 756×288 ein unkomprimiertes Datenvolumen von ca. 650 MByte an. Dies ist selbst aus heutiger Sicht eine unhandliche Größe, sodass es aus dieser Sicht günstiger ist, Trainingsmaterial zu verwenden, das auf den Daten von Digitalisiertabletts bzw. Scannern basiert. Daraus ergibt sich jedoch ein *Mismatch* zwischen den Trainings- und Anwendungsbedingungen, der durch entsprechende Vorverarbeitungsschritte oder darüberhinaus durch Adaption der Klassifikationsparameter an die Anwendungsbedingungen ausgeglichen werden kann.

4.3 Extraktion der Schreibdynamik aus Videobildfolgen

Soll anhand einer Videobildfolge eine online Schrifterkennung auf Basis der Schreibdynamik durchgeführt werden, so sind im Gegensatz zur Verwendung von Digitalisiertabletts zusätzliche Verarbeitungsschritte notwendig, um die Stifttrajektorie und damit die Dynamik der Schreibbewegung zu extrahieren. Dazu lassen sich in der Literatur zwei unterschiedliche Ansätze finden, die im folgenden näher vorgestellt werden [Mun02, Bun99].

4.3.1 Template-Matching Verfahren von Munich & Perona

Das System von Munich & Perona [Mun96, Mun00, Mun02] basiert auf einem Template-Matching Ansatz, mit dem der Stift während der Schreibbewegung in der Videobildfolge verfolgt wird. Anhand der resultierenden Trajektorie wurden von den Autoren Experimente zur Unterschriftenverifikation durchgeführt, die die Robustheit und Genauigkeit dieses Verfahrens zur Stiftverfolgung unter Beweis stellten. Zur Handschrifterkennung wurde das Verfahren bisher jedoch nicht eingesetzt.

Systemaufbau

Der Systemaufbau ist so gestaltet, das eine möglichst natürliche Eingabeschnittstelle zur Mensch-Maschine Kommunikation realisiert wird. Die verwendetet Kamera ist eine handelsübliche Videokamera, die in ca. 30cm Höhe über der Schreiboberfläche angebracht ist und im Interlace Modus 60 Bilder pro Sekunde (NTSC-Standard) mit einer Auflösung von 640×240 Pixel aufnimmt. Dabei ist weder eine spezielle Beleuchtung noch eine Kalibrierung der Kamera notwendig.

Es kann mit einem beliebigen Stift auf normalem Papier geschrieben werden. Die einzigen Einschränkungen sind, dass die Kamera stets freie Sicht auf die Stiftspitze hat, und die Schrift einen ausreichenden Kontrast zum Papier aufweist.

Initialisierung

Zur Initialisierung des Systems wird ein semiautomatisches Verfahren eingesetzt. Dabei muss der Benutzer den Stift in einem vorgegebenen Bereich aufsetzen, der ihm zusammen mit dem aufgenommenen Bild auf einem Monitor angezeigt wird. Durch die Auswertung von Differenzbildern wird die durch die Stiftbewegung hervorgerufene Aktivität in dem Bereich festgestellt. Nachdem die Stiftbewegung in den vorgegebenen Bereich detektiert wurde, muss der Benutzer den Stift für eine kurze Zeit ruhig halten, sodass das Template der Stiftspitze, das für die weitere Verfolgung des Stifts verwendet wird, extrahiert werden kann. Anschließend wird ein akustisches Signal ausgegeben, das anzeigt, dass der Schreibprozess begonnen werden kann.

Das Template der Stiftspitze ist ein 25×25 Pixel großes Bild, das um den Kontaktpunkt der Stiftspitze mit dem Papier zentriert ist. Um diesen Kontaktpunkt zu ermitteln wird die in die Bildebene projizierte Stiftspitze als Dreieck modelliert, dessen Seitenlinien anhand der im Bildausschnitt gefundenen Kanten ermittelt werden. Im ersten Schritt werden dazu Kantenpixel mit dem Canny-Kantendetektor extrahiert. Unter der Annahme, dass der Kontrast zwischen der Stiftspitze und dem Papier deutlich höher ist als der Kontrast zwischen Stift- und Hautfarbe, werden nur die Kantenpixel betrachtet, die am Übergang zwischen Stift und Papier auftreten. Diese Kantenpixel werden dann durch zwei Geraden approximiert, wobei die eine Gerade die obere Stiftkante und die andere Gerade die untere Stiftkante beschreibt. Der Übergang von der Stiftspitze zum Papier auf der Mittelachse der beiden Geraden wird dann als Kontaktpunkt zwischen Stiftspitze und Papier und damit als Mittelpunkt des Templates angenommen.

Stiftverfolgung

Das in der Initialisierungsphase ermittelte Template der Stiftspitze ist die Grundlage für die weitere Stiftverfolgung während des Schreibprozesses. Dazu wird in jedem aufgenommenen Bild die Position der größten Übereinstimmung zwischen Bild und Template bestimmt. Als Maß für die Übereinstimmung wird die normalisierte Kreuzkorrelation verwendet [Cox95]:

$$d(x,y) = \frac{\sum\limits_{k,l} P_{x+k,y+l} \cdot T_{k,l}}{\sqrt{\sum\limits_{k,l} P^2_{x+k,y+l} \cdot \sum\limits_{k,l} T^2_{k,l}}}. \tag{4.1}$$

Hierbei bezeichnen x, y die Matching-Koordinaten im aktuellen Bild P und k, l die Abmessungen des Templates T. Sind die Koordinaten der größten Übereinstimmung zwischen Bild und Template ermittelt, so erfolgt anschließend, ähnlich wie bei der Initialisierung, die "Feinlokalisation" des Kontaktpunktes zwischen Stiftspitze und Papier. Unterschreitet dagegen die Korrelation einen vorgegebenen Schwellwert, so wird der Tracking-Vorgang abgebrochen.

Um die Stiftverfolgung in Echtzeit realisieren zu können, kann das Template-Matching auf Grund der Zeitkomplexität jedoch nicht auf dem gesamten Bild durchgeführt werden. Vielmehr ist es erforderlich, den Suchraum für das Template-Matching einzuschränken. Aus diesem Grund wird ein Kalman-Filter eingesetzt, der anhand eines kinematischen Bewegungsmodells eine Schätzung der Position der Stiftspitze für das jeweils folgende Bild vornimmt.

Kalman-Filter

Ein Kalman-Filter eignet sich zur Modellierung eines linearen, zeitinvarianten Systems, das mit einem Zufallsprozess überlagert ist. Der Kalman-Filter stellt dabei einen rekursiven, varianzminimierenden Schätzer des Systemzustands dar, sodass anhand des bis zum aktuellen Zeitpunkt beobachteten Systemverhaltens Schätzungen über das zukünftige Systemverhalten abgeleitet werden können.

Das mit einem Zufallsprozess überlagerte lineare System, das mit Hilfe eines Kalman-Filters modelliert werden kann, lässt sich durch die folgenden Gleichungen beschreiben:

$$\boldsymbol{s}_{k+1} = \boldsymbol{\Phi}\boldsymbol{s}_k + \boldsymbol{\omega}_k \tag{4.2}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{s}_k + \boldsymbol{\mu}_k \tag{4.3}$$

Die erste Gleichung (4.2), die sogenannte Modellgleichung, beschreibt die zeitliche Entwicklung des Systemverhaltens. Der neue Systemzustand $\boldsymbol{s}_{k+1}$ ergibt sich demnach aus der Multiplikation der Übergangsmatrix $\boldsymbol{\Phi}$ mit dem aktuellen Systemzustand $\boldsymbol{s}_k$ und einer stochastischen Störgröße $\boldsymbol{\omega}_k$. Die Störgröße $\boldsymbol{\omega}_k$ wird dabei als weißes

unkorreliertes Rauschen mit bekannter Kovarianz angenommen:

$$E\{\boldsymbol{\omega}_k \boldsymbol{\omega}_i^T\} = \begin{cases} \boldsymbol{Q}_k & , \quad i = k \\ 0 & , \quad i \neq k \end{cases} \tag{4.4}$$

Die zweite Gleichung (4.3), die sogenannte Messgleichung, beschreibt den Zusammenhang zwischen den Messwerten $\boldsymbol{z}_k$ und dem nicht beobachtbaren Systemzustand $\boldsymbol{s}_k$. Dieser Zusammenhang wird als linear angenommen, wobei die Messunsicherheit durch den Vektor $\boldsymbol{\mu}_k$ repräsentiert ist. Die Messunsicherheit wird ebenfalls durch unkorreliertes weißes Rauschen beschrieben:

$$E\{\boldsymbol{\mu}_k \boldsymbol{\mu}_i^T\} = \begin{cases} \boldsymbol{R}_k & , \quad i = k \\ 0 & , \quad i \neq k \end{cases} \quad \text{und} \tag{4.5}$$

$$E\{\boldsymbol{\omega}_k \boldsymbol{\mu}_i^T\} = 0 \quad \text{für alle } k, i \tag{4.6}$$

Mit Hilfe des Kalman-Filters wird nun ausgehend von einem initialen Schätzwert $\hat{\boldsymbol{s}}_k^-$ des Systemzustands anhand des Messwertes $\boldsymbol{z}_k$ eine verbesserte Schätzung $\hat{\boldsymbol{s}}_k$ des Systemzustands vorgenommen. Der verbesserte Schätzwert ergibt sich dabei aus folgender Gleichung

$$\hat{\boldsymbol{s}}_k = \hat{\boldsymbol{s}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{s}}_k^-), \tag{4.7}$$

wobei $\boldsymbol{K}_k$ den sogenannten *Kalman-Faktor* bezeichnet. Die Aufgabenstellung der Kalman-Filterung ist demnach, den Mischfaktor $\boldsymbol{K}_k$ so zu bestimmen, dass der resultierende verbesserte Schätzwert möglichst gut dem "wahren" Systemzustand entspricht. Dazu ist die Fehlerkovarianz, also der mittlere quadratische Fehler

$$\boldsymbol{P}_k = E\{(\boldsymbol{s}_k - \hat{\boldsymbol{s}}_k)(\boldsymbol{s}_k - \hat{\boldsymbol{s}}_k)^T\} \tag{4.8}$$

zwischen dem verbesserten Schätzwert $\hat{\boldsymbol{s}}_k$ und dem "wahren" Wert $\boldsymbol{s}_k$ bezüglich des Kalman-Faktors $\boldsymbol{K}_k$ zu minimieren.

Nach längerer Rechnung (siehe u.a. [Bra75]) ergibt sich für den Kalman-Faktor

$$\boldsymbol{K}_k = \frac{\boldsymbol{P}_k^- \boldsymbol{H}^T}{\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R}} \quad , \tag{4.9}$$

wobei $\boldsymbol{P}_k^-$ die Kovarianzmatrix des mittleren quadratischen Fehlers zwischen dem initialen Schätzwert $\hat{\boldsymbol{s}}_k^-$ und dem "wahren" Wert $\boldsymbol{s}_k$ bezeichnet. Für die Fehlerkovarianzmatrix $\boldsymbol{P}_k$ der verbesserten Schätzung erhält man mit Gleichung 4.9

$$\boldsymbol{P}_k = (\boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{H} \boldsymbol{P}_k^-) \quad . \tag{4.10}$$

Durch Anwendung des Kalman-Faktors $\boldsymbol{K}_k$ kann damit gemäß Gleichung 4.7 die initiale Schätzung des Systemzustands unter Berücksichtigung des Messwertes verbessert werden. Der so korrigierte Systemzustand $\hat{\boldsymbol{s}}_k$ und die verbesserte Schätzung

der Fehlerkovarianz $\boldsymbol{P}_k$ bilden dann die Grundlage für neue initiale Vorhersagen von Systemzustand und Fehlerkovarianz zum folgenden Zeitpunkt:

$$\hat{\boldsymbol{s}}_{k+1}^{-} = \boldsymbol{\Phi}\hat{\boldsymbol{s}}_k \tag{4.11}$$

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{\Phi}\boldsymbol{P}_k\boldsymbol{\Phi}^T + \boldsymbol{Q} \quad . \tag{4.12}$$

Der Kalman-Filter stellt somit ein rekursives Schätzverfahren dar, das sich in eine Vorhersage- und eine Korrekturphase gliedert. In der Vorhersagephase werden anhand der Modellgleichung und des korrigierten Systemzustands Schätzungen für den neuen Systemzustand und die neue Fehlerkovarianz vorgenommen (4.11, 4.12). In der Korrekturphase werden diese Schätzungen dann unter Berücksichtigung des Messwertes verbessert (4.7, 4.10).

Einsatz des Kalman-Filters zur Stiftverfolgung

Die Motivation für den Einsatz des Kalman-Filters zur Stiftverfolgung ist, den Suchraum für das Template-Matching einzuschränken, sodass eine Echtzeit-Verarbeitung gewährleistet werden kann. Dazu wird ein kinematisches Bewegungsmodell verwendet, um in der Vorhersagephase des Kalman-Filters eine Schätzung der Stiftposition für den jeweils folgenden Zeitpunkt vorzunehmen. In der Korrekturphase wird dann das Template-Matching zur Messung der Stiftposition durchgeführt, wobei dann nur ein relativ kleiner Bildbereich, der um die vorhergesagte Stiftposition zentriert ist, betrachtet wird.

Das in [Mun00, Mun02] verwendete diskrete kinematische Bewegungsmodell beschreibt eine konstante Beschleunigung $\boldsymbol{a}$ des Stiftes, die mit weißem Rauschen $\boldsymbol{\omega}$ überlagert ist. Mit $\Delta t = t_{k+1} - t_k$ sind Position, Geschwindigkeit und Beschleunigung des Stifts demnach gegeben durch:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \boldsymbol{v}_k\Delta t + \frac{1}{2}\boldsymbol{a}_k\Delta t^2 \\ \boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + \boldsymbol{a}_k\Delta t \\ \boldsymbol{a}_{k+1} &= \boldsymbol{a}_k + \boldsymbol{\omega}_{a_k} \end{aligned} \tag{4.13}$$

Gemessen wird die Stiftposition. Die Messgleichung lautet also:

$$\boldsymbol{z}_k = \boldsymbol{x}_k + \boldsymbol{\mu}_k \quad . \tag{4.14}$$

Fasst man Position, Geschwindigkeit und Beschleunigung im Zustandsvektor $\boldsymbol{s}$ zusammen, so ergeben sich Modell- und Messgleichung des Kalman-Filters zu

$$\begin{aligned} \boldsymbol{s}_{k+1} &= \boldsymbol{\Phi}\boldsymbol{s}_k + \boldsymbol{\omega}_k \\ \boldsymbol{z}_k &= \boldsymbol{H}\boldsymbol{s}_k + \boldsymbol{\mu}_k \quad , \end{aligned} \tag{4.15}$$

wobei

$$\boldsymbol{s}_k = \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{v}_k \\ \boldsymbol{a}_k \end{pmatrix}, \qquad \boldsymbol{\omega}_k = \begin{pmatrix} 0 \\ 0 \\ \boldsymbol{\omega}_{a_k} \end{pmatrix},$$

und

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & \Delta t & \frac{1}{2}\Delta t \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}.$$

Pen-up/down Klassifikation

Anhand der aus der Stiftverfolgung resultierenden Trajektorie kann nicht unterschieden werden, ob der Stift aufgesetzt ist und sich in einer Schreibphase befindet (pen-down), oder ob sich der Stift über der Schreiboberfläche zu einer neuen Position bewegt (pen-up). In [Mun00, Mun02] wurde daher vorgeschlagen, die auf dem Papier zurückgelassene Tintenspur[1] als Merkmal für die pen-up/down Unterscheidung zu verwenden.

Die Detektion der Tintenspur erfolgt durch ein lokales Verfahren, indem für jeden Punkt der Stifttrajektorie der korrespondierende Grauwert im Bild mit den Grauwerten umgebender Pixel verglichen wird (siehe Abbildung 4.1). Diese umgebenden Pixel lie-

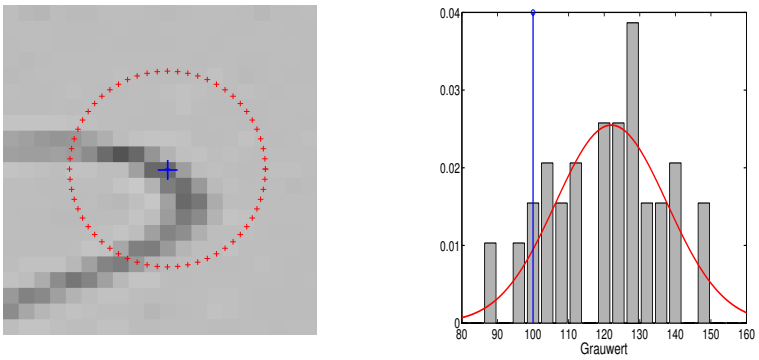


Abbildung 4.1: Lokale pen-up/down Detektion (nach [Mun00, Mun02]): Links ist ein Bildausschnitt zur pen-up/down Detektion abgebildet. Die extrahierte Stiftposition ist in blau angedeutet, während die betrachteten Umgebungspixel rot dargestellt sind. Die rechte Abbildung zeigt das Grauwerthistogramm der Umgebungspixel und die daraus geschätzte Normalverteilungsdichte. Der Grauwert an der Stiftposition ist in blau eingezeichnet. Die pen-up Wahrscheinlichkeit ergibt sich aus der Fläche unter der Dichtefunktion zwischen $-\infty$ und dem Grauwert an der Stiftposition.

gen auf einem Kreis, der um die extrahierte Stiftposition zentriert ist. Dabei wird davon ausgegangen, dass die Grauwerte dieser als pen-up angenommenen Pixel normalverteilt sind, sodass anhand ihres Grauwerthistogramms die Parameter einer Normalverteilungsdichte berechnet werden. Die pen-up Wahrscheinlichkeit an der betrachteten

[1] Der Einfachheit halber wird hier die Bezeichnung “Tinte” allgemein für die auf dem Papier zurückbleibende Schriftspur verwendet.

Stiftposition ergibt sich dann aus der Integration der Normalverteilungsdichte zwischen $-\infty$ und dem entsprechenden Grauwert an der Stiftposition.

Die pen-up/down Detektion kann jedoch erst erfolgen, wenn der Stift ausreichend weit von der betrachteten Position entfernt ist, sodass die Tintenspur nicht vom Stift oder der Hand des Schreibers verdeckt ist. In [Mun00, Mun02] wird zur Detektion, ob zum betreffenden Zeitpunkt an der betrachteten Stiftposition die pen-up/down Unterscheidung durchgeführt werden kann, ein kegelförmiges Modell verwendet, das den von Stift und Hand überdeckten Bildbereich beschreibt. Demzufolge wird mit der pen-up/down Unterscheidung solange gewartet, bis sich der Kegel außerhalb des betrachteten Bildausschnitts befindet.

Um die Robustheit der pen-up/down Bestimmung zu verbessern, sind in [Mun00, Mun02] weitere Verarbeitungsschritte eingesetzt worden, die jedoch nicht in das Echtzeit-System integriert wurden. Zum einen wird ein Hidden-Markov-Modell verwendet, das die beiden Zustände pen-up und pen-down besitzt, um die Übergänge zwischen diesen Zuständen zu modellieren. Mit Hilfe des Viterbi-Algorithmus kann somit anhand einer beobachteten Folge von pen-up Wahrscheinlichkeiten, die durch das in Abbildung 4.1 dargestellte Verfahren ermittelt wurden, die wahrscheinlichste Sequenz von pen-up bzw. pen-down Zuständen bestimmt werden.

Zum anderen wird eine Segmentierung der Trajektorie in Strokes durchgeführt, sodass die lokalen, an jedem Punkt der Trajektorie vorliegenden pen-up Wahrscheinlichkeiten zu einer stroke-weisen pen-up/down Unterscheidung zusammengefasst werden können. Als Segmentierungskriterien werden dabei die Geschwindigkeit der Stiftspitze und die Krümmung der Trajektorie verwendet.

4.3.2 Differenzbildverfahren von Bunke & Kollegen

Im Gegensatz zu dem oben beschriebene Verfahren von Munich und Perona, das auf der Stiftverfolgung mittels Template-Matching basiert, verwendet das von Bunke und Kollegen vorgestellte Verfahren [Bun99, vS98] einen Differenzbildansatz, um direkt anhand der Tintenspur die Stifttrajektorie zu extrahieren. Auf Basis der extrahierten Trajektorien wurden außerdem mit Hilfe eines bestehenden Erkennungssystems ([Sch95a]) Experimente zur Handschrifterkennung durchgeführt.

Systemaufbau

Der Schreibprozess wird mit einer Videokamera aufgenommen, die in ca. 35cm Entfernung von der Schreibposition angebracht ist. Dabei beträgt der Winkel zwischen der optischen Achse der Kamera und der Schreiboberfläche ca. 45 Grad. Die Bilder weisen eine Auflösung von 384×288 Pixel bei einer Bildrate von 19 Bildern pro Sekunde auf. Die Szene wird über die weiße Zimmerdecke durch vier Fotolampen indirekt beleuchtet, um die Entstehung scharfer Schatten zu vermeiden. Der verwendete Stift ist ein Filzstift, der bis auf die Stiftspitze weiß eingefärbt ist.

Extraktion der Stifttrajektorie

Die Grundidee des Verfahrens lässt sich anhand von Abbildung 4.2 beschreiben: Berechnet man das Differenzbild der zum Zeitpunkt t und $t + 1$ aufgenommenen Bilder, so erhält man daraus im idealen Fall die Tintenspur, die zwischen den beiden Zeitpunkten produziert wurde. Bei einer ausreichend hohen Abtastrate enthält jedes Differenzbild nur maximal ein gesetztes Pixel, das der eingenommenen Stiftposition entspricht. Ist kein Pixel im Differenzbild gesetzt, so hat sich der Stift entweder nicht bewegt oder er war nicht aufgesetzt (pen-up). Im Normalfall besteht die Differenz jedoch häufig aus mehreren Pixeln, anhand derer dann *eine* Stiftposition zu bestimmen ist. Als resultierende Stiftposition wird in diesem Fall diejenige gewählt, die den mittleren quadratischen Abstand zu den übrigen extrahierten Pixeln minimiert.

Dieses Vorgehen weist in der Praxis jedoch die Schwierigkeit auf, dass Differenzpixel nicht nur durch die Tintenspur hervorgerufen werden können, sondern auch durch die Bewegung des Stifts, der Hand oder durch Schattenwurf. Diesem Problem wird zum einen durch die Verwendung eines weißen Stifts begegnet, zum anderen durch die sorgfältig kontrollierten Beleuchtungsbedingungen. Außerdem werden bei der Differenzbildberechnung nur die Pixel betrachtet, die ihre "Farbe" von weiß auf schwarz ändern und in einer bestimmten Anzahl darauffolgender Bilder auch schwarz bleiben.

Eine weitere Maßnahme, um die durch Schattenwurf oder Handbewegungen hervorgerufenen Störpixel zu eliminieren, besteht darin, den betrachteten Bereich im Differenzbild auf eine *Region of interest* (ROI) einzuschränken. Dazu wird nach Beendigung des Schreibvorgangs anhand des letzten Bildes der Sequenz der Schriftzug extrahiert, wobei sichergestellt sein muss, dass dieses Bild nicht durch den Stift, die Hand

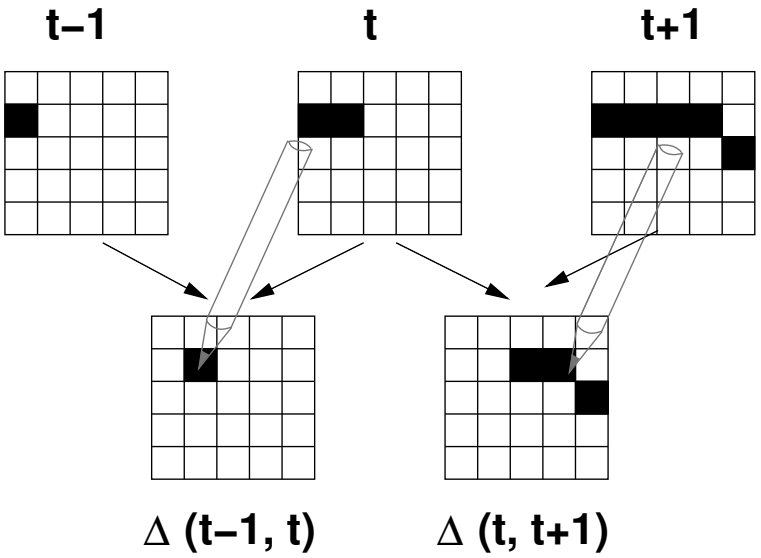


Abbildung 4.2: Differenzbildansatz (aus [Bun99]). In der oberen Zeile sind die aufgenommenen Bilder, unten die daraus resultierenden Differenzbilder dargestellt.

des Schreibers oder durch Schattenwurf gestört ist. Die ROI ergibt sich dann durch Anwendung eines 3×3 Dilatationsoperators auf die extrahierten Schriftpixel. Da die Bestimmung der ROI anhand des letzten Bildes der Sequenz vorgenommen wird, kann somit auch der Erkennungsprozess erst nach Beendigung des Schreibprozesses beginnen.

Außerdem können Schwierigkeiten dadurch auftreten, dass die Tintenspur kurzzeitig durch den Stift oder die Hand verdeckt sein kann. Das hat dann zur Folge, dass ein Pixel, das in Wahrheit einer einmalig eingenommenen Stiftposition entspricht, in mehreren Differenzbildern auftauchen kann. Diesen Artefakten kann jedoch mit Hilfe eines *aggregierten* Differenzbildes entgegengewirkt werden, sodass nur noch diejenigen Pixel in das jeweilige Differenzbild übernommen werden, die vorher noch in keinem anderen Differenzbild gesetzt waren.

Größere Probleme bereiten jedoch die Situationen, in denen längere Abschnitte der Tintenspur durch den Stift verdeckt sind *während* sie geschrieben werden. Dies ist beispielsweise bei Schleifen der Fall und führt zu einer erschwerten Rekonstruktion der Trajektorie, da diese Abschnitte zu einem späteren Zeitpunkt “auf einen Schlag” im Differenzbild sichtbar werden. Eine weitere Schwierigkeit tritt auf, wenn ein bereits geschriebener Linienzug erneut überstrichen wird. Diese Bewegung kann anhand der Differenzbilder nicht detektiert werden.

4.3.3 Beurteilung der Verfahren

Eine objektive Beurteilung der oben vorgestellten Verfahren zur Erfassung der Stifttrajektorie ist nur schwer möglich, da sie nicht unter vergleichbaren Bedingungen getestet wurden. Dennoch sollen hier die Vor- bzw. Nachteile der Verfahren diskutiert werden, um herauszufinden, inwiefern sie für den Einsatz zur videobasierten Handschrifterkennung geeignet sind.

Das Differenzbildverfahren von Bunke & Kollegen bietet den Vorteil, dass durch die Verfolgung der Tintenspur die pen-up/down Unterscheidung implizit als “Nebenprodukt” abfällt und hierfür keine zusätzlichen Verarbeitungsschritte notwendig sind. Jedoch weist das Differenzbildverfahren einige Nachteile auf. Hier sind beispielsweise die Schwierigkeiten beim Erfassen der Trajektorie zu nennen, die durch Verdeckungen der Tintenspur und mehrfaches Überstreichen des Linienzuges hervorgerufen werden. Weiterhin wird, um das Auftreten von Störpixeln in den Differenzbildern zu vermeiden, neben der Verwendung eines weißen Stifts die sorgfältige Kontrolle der Beleuchtung vorausgesetzt. Damit wird jedoch die Forderung nach einer möglichst natürlichen Mensch-Maschine Schnittstelle deutlich eingeschränkt. Ein weiterer Nachteil ist, dass die Stifttrajektorie erst dann extrahiert werden kann, wenn der Schreibprozess beendet ist, da die region of interest der Differenzbilder anhand des letzten Bildes der Sequenz bestimmt wird. Damit muss ein längerer zeitlicher Versatz zwischen dem Beginn des Schreibprozesses und dem Vorliegen des Erkennungsresultats in Kauf genommen werden, sodass die Interaktivität des Verfahrens eingeschränkt ist.

Im Gegensatz dazu unterliegt das Template-Matching System von Munich & Perona weniger starken Einschränkungen. Zwar muss hier die Stiftspitze auch stets in den aufgenommenen Bildern sichtbar sein, jedoch ist keine spezielle Beleuchtung notwendig. Weiterhin kann mit einem beliebigen Stift geschrieben werden, sofern der Kontrast zwischen Stift und Papier höher ist als zwischen Stift- und Hautfarbe. Ein Nachteil des Verfahrens stellt die semi-automatische Initialisierung dar, bei der der Benutzer den Stift in einem vorgegebenen Bereich aufsetzen muss. Darüberhinaus sind im Gegensatz zum Differenzbildverfahren von Bunke & Kollegen zusätzliche Verarbeitungsschritte notwendig, um die pen-up/down Unterscheidung durchzuführen.

4.4 Zusammenfassung

In diesem Kapitel wurde der Einsatz videobasierter Sensorik zur Handschrifterkennung motiviert. Zusammengefasst liegen die Vorteile insbesondere darin, dass eine möglichst natürliche Mensch-Maschine-Schnittstelle realisiert werden kann, die einen interaktiven Umgang mit dem System erlaubt.

Doch die Verwendung von Videokameras bringt auch Nachteile mit sich. Hier ist vor allem die Qualität der aufgenommenen Daten zu nennen, die gegenüber der Verwendung spezieller Sensoren deutlich schlechter ist. Neben den oftmals ungleichmäßig beleuchteten Bildern betrifft dies sowohl die geringere räumliche Auflösung, als auch die gegenüber Digitalisiertabletts geringere Abtastrate. Damit werden an die Verarbeitungsschritte in videobasierten Systemen insbesondere in Bezug auf die Robustheit höhere Anforderungen gestellt, um eine ähnliche Erkennungsleistung wie in herkömmlichen Systemen zu erreichen.

Bei der Verwendung einer Videokamera an Stelle eines Digitalisiertabletts zur online Handschrifterkennung sind außerdem zusätzliche Verarbeitungsschritte notwendig, um anhand der Bildfolge die Stifttrajektorie zu extrahieren. Dazu sind zwei unterschiedliche Verfahren vorgestellt worden. Das in Abschnitt 4.3.1 beschriebene Verfahren lokalisiert die Stiftspitze durch Template-Matching, das Verfahren in Abschnitt 4.3.2 verwendet dagegen Differenzbilder, um die zeitliche Entwicklung der Tintenspur zu verfolgen. Insbesondere das Template-Matching Verfahren erscheint vielversprechend, da es weniger Einschränkungen in Bezug auf die geforderte, natürliche Mensch-Maschine Schnittstelle unterliegt.

5 Videobasierte online Handschrifterkennung

In diesem Kapitel wird ein System zur videobasierten online Handschrifterkennung vorgestellt, das auf der Erfassung der Schreibdynamik basiert. Mit diesem System soll eine möglichst natürliche Eingabeschnittstelle zur Mensch-Maschine-Kommunikation realisiert werden, sodass außer einer handelsüblichen Videokamera keinerlei weitere spezialisierte Sensorik notwendig ist. Vielmehr soll mit einem herkömmlichen Stift auf normalem Papier geschrieben werden können.

Das Verfahren zur Extraktion der Stifttrajektorie aus der Videobildfolge ist weitestgehend angelehnt an das von Munich & Perona vorgeschlagene Vorgehen [Mun02]. Während in ihrer Arbeit die Trajektorie zwar zur Unterschriftenverifikation, nicht jedoch zur Schrift*erkennung* verwendet wird, werden hier darüberhinaus Verfahren zur Vorverarbeitung, Segmentierung, Merkmalsextraktion und Klassifikation eingesetzt, sodass damit ein komplettes Erkennungssystem vorliegt. Eine Beschreibung der eingesetzten Verfahren findet sich auch in [Fin01, Wie01].

5.1 Systemaufbau

Der Systemaufbau zur videobasierten online Handschrifterkennung ist in der Abbildung 5.1 veranschaulicht. Zur Aufnahme des Schreibprozesses wird eine herkömmliche Videokamera eingesetzt (Sony EVI-D31), die in ca. 50cm Höhe über der Schreiboberfläche angebracht ist, wobei der Winkel zwischen der optischen Achse der Kamera und der Schreiboberfläche ca. 65 Grad beträgt. Die Größe des beobachteten Schreibfelds beträgt dabei ca. 18×15 cm, sodass ungefähr die Hälfte einer Din A4 Seite abgedeckt ist. Die Videokamera arbeitet im Interlace-Modus und nimmt der PAL-Norm entsprechend 50 Halbbilder pro Sekunde mit der Auflösung 756×288 Pixel auf.

Die Beleuchtung der Szene wird durch zwei handelsübliche Schreibtischlampen vorgenommen, in denen jeweils zwei elf Watt leistende Leuchstoffröhren eingebaut sind. Das abgebildete Digitalisiertablett wird zum einen verwendet, um Trainings- und Validierungsdaten aufzunehmen, zum anderen verhindert es durch elektrostatische Anziehung ein “Verrutschen” des Papiers, sodass das Tablett als Schreibunterlage auch für die videobasierte Erkennung eingesetzt wird.

Als Schreibmittel eignen sich vor allem die Stifte, die eine kegelförmige Stiftspitze besitzen, die dunkler als das Papier ist. Dies trifft beispielsweise auf Bleistifte und üblicherweise auch auf Kugelschreiber zu. Liegt ein Template der Stiftspitze vor, so ist auch die Verwendung beliebiger Stifte möglich.

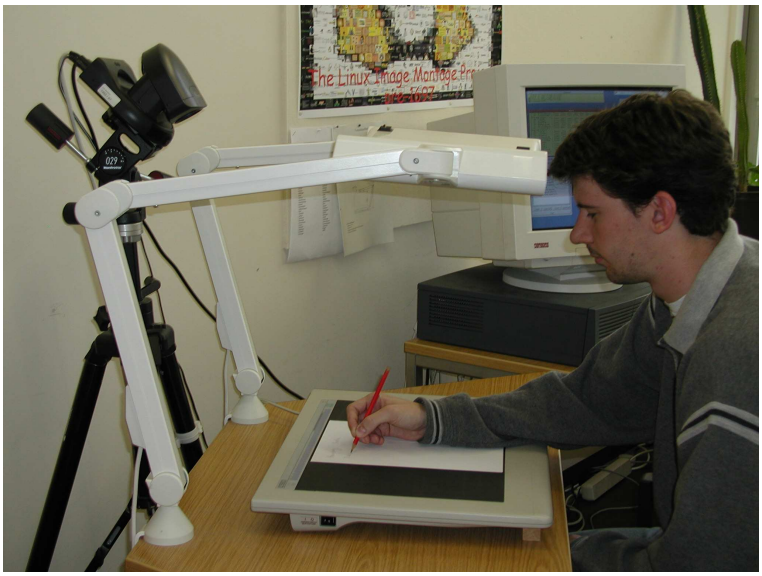


Abbildung 5.1: Systemaufbau zur videobasierten online Handschrifterkennung. Das Digitalisiertablett wird nur zur Fixierung des Papiers und zur Aufnahme von Trainingsdaten eingesetzt.

5.2 Extraktion der Stifttrajektorie

Das realisierte Verfahren zur Extraktion der Stifttrajektorie anhand der aufgenommenen Videobildfolge basiert auf einem Template-Matching Ansatz. Es ist angelehnt an das in Abschnitt 4.3.1 vorgestellte System von Munich & Perona, das uns freundlicherweise vom California Institute of Technology zur Verfügung gestellt wurde. Das System wurde im Rahmen einer studentischen Projektarbeit vollständig neu implementiert, wobei einige Verfahrensschritte modifiziert wurden.

Die durchgeführten Modifikationen am Verfahren von Munich & Perona betreffen zum einen die Initialisierung, die nun deutlich weniger Benutzerkooperation erfordert, zum anderen wurde das dem Kalman-Filter unterliegende kinematische Modell der Schreibbewegung verändert. Außerdem wurden noch Änderungen an der pen-up/down Unterscheidung durchgeführt.

5.2.1 Initialisierung

Im Ausgangssystem von Munich & Perona muss der Benutzer den Stift in einem vorgegebenen Bereich der Schreibfläche aufsetzen, um den Beginn eines Schreibprozesses anzuzeigen. Erst dann kann ein Template der Stiftspitze extrahiert werden, das zur weiteren Stiftverfolgung verwendet wird. Dieses Vorgehen zur Initialisierung erfordert jedoch entweder, dass der Bereich auf dem Papier markiert und die Kamera entsprechend kalibriert ist, oder dass der Benutzer die Stiftbewegung in den vorgegebenen Bildbereich am Monitor beobachten muss.

Um dagegen die notwendige Benutzerkooperation auf ein Mindestmaß reduzieren zu können, wurde eine zweistufige Aufmerksamkeitssteuerung zur Initialisierung entwickelt, mit der der Beginn eines Schreibvorgangs automatisch erkannt werden kann. Im ersten Teilschritt wird dabei das Auftreten einer signifikanten Szenenänderung detektiert, die beispielsweise dann vorliegt, wenn eine Hand ins Bild kommt. Wenn dies der Fall ist, wird im zweiten Teilschritt die initiale Lokalisierung der Stiftspitze durchgeführt. Dazu wird ein Satz generischer Templates verwendet, mit denen unterschiedliche Stiftformen abgedeckt werden. Ist die initiale Stiftposition schließlich ermittelt, so wird daraufhin der 20×20 Pixel große Bildbereich, der um diese Stiftposition zentriert ist, ausgeschnitten und als Template für die weitere Stiftverfolgung verwendet.

Die einzige Einschränkung bei dieser Vorgehensweise ist, dass der Benutzer während der Stiftsuche, die bei der Verwendung von drei initialen Templates ca. drei bis vier Sekunden dauert, den Stift ruhig halten muss.

Detektion von Szenenänderungen

Signifikante Szenenänderungen, die beispielsweise auftreten, wenn sich die Hand in die Szene bewegt, werden mit Hilfe von Differenzbildern detektiert. Um zu verhindern, dass das System auch auf Schwankungen der Beleuchtung reagiert, die sich global auf die gesamte Szene auswirken, wird das Bild in vier Quadranten eingeteilt, anhand derer die Differenzberechnung vorgenommen wird. Für die einzelnen Quadranten Q_i wird dann jeweils die Summe der pixelweise berechneten Grauwertdifferenzen des aktuellen Bildes I_t zum vorherigen Bild I_{t-1} bestimmt:

$$d_{Q_i} = \sum_{x,y \in Q_i} |I_t(x,y) - I_{t-1}(x,y)| \quad \text{mit } i = 1, \ldots, 4. \tag{5.1}$$

Eine Handbewegung wird nun festgestellt, wenn obiger Wert für einen oder mehrere Quadranten einen vorgegebenen Schwellwert übersteigt. Nun wird solange gewartet, bis der Stift an der Schreibposition aufgesetzt ist und mithin die Summen der Grauwertdifferenzen wieder unter den Schwellwert sinken. Dadurch ist sichergestellt, dass der Stift ruhig an einer Position verharrt, sodass anschließend mit der Lokalisierung der Stiftspitze fortgefahren werden kann.

Liegen die Summen der Grauwertdifferenzen d_{Q_i} dagegen in allen vier Quadranten gleichzeitig über dem Schwellwert, so wird davon ausgegangen, dass die Ursache nicht eine Handbewegung sondern vielmehr eine globale Beleuchtungsänderung der Szene war.

Initiale Lokalisierung der Stiftspitze

Der zweite Teilschritt zur Initialisierung, der durchgeführt wird, wenn eine durch eine Handbewegung hervorgerufene Szenenänderung detektiert wurde, ist die Lokalisierung der Stiftspitze. Das dazu verwendete Verfahren basiert auf einem Template-Matching Ansatz, bei dem mehrere Templates unterschiedlicher Stiftspitzen mit dem

aufgenommenen Bild verglichen werden. Als optimale initiale Stiftposition wird dann die Position angenommen, an der die größte Übereinstimmung zwischen Bild und Template gemessen wird.

Das Maß für die Übereinstimmung des jeweiligen Templates T_i mit dem aktuellen Bild I basiert auf dem euklidischen Abstand der Grauwerte korrespondierender Pixel:

$$d_i(x,y) = \sum_{u=0}^{M_u} \sum_{v=0}^{M_v} \left[I(x+u, y+v) - T_i(u,v) \right]^2 \ . \tag{5.2}$$

Hierbei bezeichnen M_u und M_v die horizontale bzw. vertikale Ausdehnung der Templates. Die Lokalisation der Stiftspitze entspricht somit einer Minimierung über die Abstände $d_i(x,y)$.

Um die Robustheit des Verfahrens in Bezug auf schwankende Beleuchtungsbedingungen zu erhöhen, werden sowohl die einzelnen Templates als auch das aufgenommene Bild einer Grauwertnormalisierung unterzogen. Dabei wird der Dynamikbereich der Grauwerte auf das gesamte zur Verfügung stehende Intervall $[0, \ldots, 255]$ ausgedehnt.

Das Template-Matching Verfahren, das zur initialen Stiftfindung auf dem gesamten Bild durchgeführt werden muss, ist äußert rechenaufwendig. So ist die Anzahl der erforderlichen pixelweisen Vergleichsoperationen bei einer Bildgröße von $N_x \times N_y$ und einer Templategröße von $M_u \times M_v$ gegeben durch:

$$K = N_x N_y M_u M_v \ . \tag{5.3}$$

Um diesen Aufwand einzuschränken und somit die Antwortzeit des Systems zu verkürzen, wird das Template-Matching in eine Grob- und eine Feinsuche unterteilt. Dabei wird die Beobachtung ausgenutzt, dass bei einem kleinen Versatz des Templates bezüglich der optimalen Position die Grauwertdistanz $d_i(x,y)$ noch nahe am globalen Minimum liegt. Daher wird bei der Grobsuche so vorgegangen, dass das Template nicht Pixel für Pixel sondern in Sprüngen von n Pixeln über das Bild geschoben wird. Ausgehend von derjenigen Grobposition, die im Vergleich zu den übrigen die minimale Grauwertdistanz aufweist, wird anschließend die Feinsuche durchgeführt. Bei der Feinsuche wird dann das Template-Matching pixelweise vorgenommen, wobei der Suchbereich jedoch auf die $n \times n$ Umgebung um die Grobposition beschränkt ist. Daraus ergibt sich eine verbesserte Effizienz von

$$\begin{aligned} K &= K_{\text{grob}} + K_{\text{fein}} \\ &= \frac{1}{n^2} N_x N_y M_u M_v + n^2 M_u M_v \ . \end{aligned} \tag{5.4}$$

Um eine robuste Stiftlokalisation zu gewährleisten, darf bei der Grobsuche der Versatz n jedoch nicht zu groß gewählt werden. Bei den Experimenten hat sich für n ein Wert von zehn Pixeln bei einer Templategröße von 40×40 Pixel als geeignet erwiesen.

Die erfolgreiche Lokalisation der Stiftspitze wird nun hypothetisiert, wenn die minimale Distanz $d_i(x,y)$ nach abgeschlossener Feinsuche unter einem vorgegebenen

Schwellwert liegt. Dann wird der Bildbereich, der um die extrahierte Position zentriert ist, ausgeschnitten und als aktuelles Template der Stiftspitze für die weitere Stiftverfolgung verwendet. Überschreitet die minimale Distanz zwischen Template und Bild dagegen den Schwellwert, so wird angenommen, dass sich entweder kein Stift in der Szene befindet oder dass die Stiftspitze verdeckt ist. In diesem Fall schaltet das System in den Startzustand, und es wird mit dem ersten Schritt, der Detektion von Szenenänderungen, fortgefahren.

5.2.2 Stiftverfolgung

Nachdem die initiale Stiftposition ermittelt worden ist, muss während des Schreibprozesses der Stift in der Videobildfolge verfolgt werden. Dies geschieht, ähnlich wie bei der Initialisierung auch, durch ein Template-Matching Verfahren. Da bei der Bildaufnahmerate von 50 Bildern pro Sekunde für jedes Bild zur Bestimmung der Position der Stiftspitze jedoch nur ein Zeitraum von 20 Millisekunden bleibt, ist eine Einschränkung des Suchbereichs für das zeitintensive Template-Matching unumgänglich. Deshalb wird auf Basis eines kinematischen Bewegungsmodells die erwartete Stiftposition für den jeweils folgenden Zeitpunkt mit Hilfe eines Kalman-Filters geschätzt. Somit kann der Suchraum für das Template Matching auf einen kleinen, um die geschätzte Stiftposition zentrierten Bereich beschränkt werden. Die Vorgehensweise ist dabei eng an das Ausgangssystem von Munich & Perona angelehnt (siehe Abschnitt 4.3.1).

Vorhersage der erwarteten Stiftposition

Um ein geeignetes Modell der Stiftbewegung zu erhalten, wurden im Rahmen einer studentischen Projektarbeit unterschiedliche Modelle anhand ihrer Vorhersagegenauigkeit miteinander verglichen. Darunter war zum einen das von Munich & Perona verwendete Modell, das auf konstanter Beschleunigung basiert, zum anderen ein Modell, in dem die Geschwindigkeit der Stiftbewegung als konstant aufgefasst wird, sowie ein Modell, das die zum aktuellen Zeitpunkt gemessene Stiftposition als Vorhersage für das folgende Bild verwendet. Die anhand von 10 Testtrajektorien vorgenommene Untersuchung zeigte, dass das Modell konstanter Geschwindigkeit sowohl in Bezug auf den maximalen als auch den mittleren Vorhersagefehler das beste Verfahren darstellt (siehe Tabelle 5.1).

Aufgrund der Vergleichsuntersuchung wird hier davon abgesehen, die Stiftbewegung, wie ursprünglich von Munich & Perona vorgeschlagen, durch ein Modell konstanter Beschleunigung zu beschreiben. Stattdessen wird das Modell konstanter Geschwindigkeit verwendet. Die Modellgleichung ist damit gegeben durch (vgl. Abschnitt 4.3.1, Gleichung 4.13):

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \boldsymbol{v}_k + \boldsymbol{\omega}_{x_k} \\ \boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + \boldsymbol{\omega}_{v_k} \end{aligned} \tag{5.5}$$

Traj.	horizontal			vertikal		
	K_v	K_a	D	K_v	K_a	D
1	4.1 (39)	7.5 (84)	4.3 (49)	2.0 (26)	3.1 (49)	1.8 (27)
2	3.8 (35)	6.2 (66)	4.7 (36)	2.0 (24)	2.9 (41)	2.1 (22)
3	3.8 (65)	7.6 (125)	3.5 (66)	1.8 (24)	2.6 (40)	1.7 (24)
4	4.1 (76)	8.2 (153)	4.2 (83)	2.0 (11)	2.6 (18)	1.9 (11)
5	4.2 (33)	8.3 (63)	4.3 (41)	2.2 (14)	3.1 (24)	2.2 (14)
6	3.2 (52)	5.9 (96)	3.2 (57)	1.7 (9)	2.2 (13)	1.5 (8)
7	2.8 (25)	3.8 (42)	4.2 (57)	1.3 (13)	1.5 (18)	1.7 (13)
8	2.5 (106)	4.6 (181)	3.3 (107)	1.2 (14)	1.6 (19)	1.4 (12)
9	2.1 (27)	3.1 (48)	3.1 (36)	1.3 (10)	1.6 (15)	1.4 (10)
10	1.8 (20)	2.5 (34)	2.3 (22)	1.1 (8)	1.3 (11)	1.0 (9)

Tabelle 5.1: Mittlerer (maximaler) Vorhersagefehler der Stiftposition in Pixel. K_v bezeichnet das Kalman-Filter-Modell mit konstanter Geschwindigkeit, K_a das Modell mit konstanter Beschleunigung und D das Modell, das die zum aktuellen Zeitpunkt gemessene Stiftposition als Vorhersage für den folgenden Zeitschritt verwendet.

Hierbei bezeichnen $\boldsymbol{x}_k$ und $\boldsymbol{v}_k$ Position bzw. Geschwindigkeit des Stifts und $\boldsymbol{\omega}_{x_k}$ bzw. $\boldsymbol{\omega}_{v_k}$ die Modellunsicherheiten.

Da nur die Stiftposition beobachtet werden kann, lautet die Messgleichung unter Berücksichtigung der Messunsicherheit $\boldsymbol{\mu}_k$ wie in Gleichung 4.14:

$$\boldsymbol{z}_k = \boldsymbol{x}_k + \boldsymbol{\mu}_k \quad . \tag{5.6}$$

Werden im Zustandsvektor $\boldsymbol{s}$ Position und Geschwindigkeit zusammengefasst, so ergeben sich die Modell- und Messgleichung zu:

$$\begin{aligned} \boldsymbol{s}_{k+1} &= \boldsymbol{\Phi}\boldsymbol{s}_k + \boldsymbol{\omega}_k \\ \boldsymbol{z}_k &= \boldsymbol{H}\boldsymbol{s}_k + \boldsymbol{\mu}_k \quad , \end{aligned} \tag{5.7}$$

mit

$$\boldsymbol{s}_k = \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{v}_k \end{pmatrix} , \qquad \boldsymbol{\omega}_k = \begin{pmatrix} \boldsymbol{\omega}_{x_k} \\ \boldsymbol{\omega}_{v_k} \end{pmatrix} ,$$

und

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} , \qquad \boldsymbol{H} = \begin{pmatrix} 1 & 0 \end{pmatrix} .$$

Die Kovarianzmatrix des Modellfehlers $\boldsymbol{Q}_k = E\{\boldsymbol{\omega}_k \boldsymbol{\omega}_k^T\}$ wird als zeitlich konstant angenommen und ergibt sich auf Basis der Arbeit von [Koh97] zu:

$$\boldsymbol{Q} = \frac{a^2 \Delta t}{6} \begin{pmatrix} 2\Delta t^2 & 3\Delta t \\ 3\Delta t & 6 \end{pmatrix} \quad . \tag{5.8}$$

Hierbei bezeichnet a eine Schätzung der maximalen im Signal vorkommenden Beschleunigung der Stiftspitze.

Die Kovarianzmatrix des Messfehlers $\boldsymbol{R} = E\{\boldsymbol{v}_k \boldsymbol{v}_k^T\}$ wird durch Auswertung der Template-Matching Resultate auf Basis von einigen per Hand annotierten Stifttrajektorien ermittelt. Der initiale Zustandsvektor $\hat{\boldsymbol{s}}_0^-$ ergibt sich aus der Stiftposition, die während der Initialisierung ermittelt wurde. Anhand des initialen Zustandsvektors und der Messfehlerkovarianz lässt sich außerdem die initiale Schätzfehlerkovarianzmatrix $\boldsymbol{P}_0^-$ bestimmen.

Damit liegen alle für die Kalman-Filterung notwendigen Größen vor, sodass der Vorgang der Stiftverfolgung initiiert werden kann.

Messung der Stiftposition

Durch die Anwendung des Kalman-Filters liegt zu jedem Zeitpunkt eine Schätzung der erwarteten Stiftposition vor. Zur *Messung* der Stiftposition kann somit der Suchbereich für das Template-Matching auf eine Bildregion eingeschränkt werden, die um die geschätzte Stiftposition zentriert ist. In dem realisierten System wurde die Größe des Suchbereichs dabei auf 120×60 Pixel gesetzt, sodass eine robuste Echtzeitverarbeitung möglich ist.

Das Template-Matching Verfahren basiert auf dem Template der Stiftspitze, welches während der Initialisierung ausgeschnitten wurde. Wie in der Initialisierungsphase ist der Matching-Vorgang zur Effizienzsteigerung in eine Grob- und eine Feinsuche unterteilt. Zur weiteren Rechenzeitersparnis wird die Berechnung der Differenz aus Gleichung 5.2 abgebrochen, wenn die bis dato berechnete Teilsumme bereits größer ist als der bisher kleinste Wert $d(x, y)$. Da der Template-Matching Vorgang an der vorhergesagten Stiftposition begonnen wird, erfolgt die Überprüfung der korrekten Stiftposition im Mittel relativ früh, sodass ein Großteil der Differenzberechnungen eingespart werden kann.

5.2.3 Pen-up/down Unterscheidung

Durch das Template-Matching wird die Position der Stiftspitze bestimmt, nicht jedoch, ob sich der Stift in einer Schreibphase befindet und auf dem Papier aufgesetzt ist (pen-down Phase), oder der Stift in geringer Entfernung über dem Papier zu einem neuen Aufsetzpunkt bewegt wird (pen-up Phase). Um die pen-up/down Unterscheidung herbeizuführen, ist es daher ein naheliegender Ansatz, die ermittelte Stiftposition mit der auf dem Papier zurückgelassenen Tintenspur in Beziehung zu setzen. Eine Stiftposition wird demnach als pen-down gekennzeichnet, wenn an seiner Position im aufgenommenen Bild Schriftpixel zu finden sind. Um Verdeckungen durch den Stift oder die Hand des Schreibers zu vermeiden, kann die pen-up/down Unterscheidung an der betrachteten Stiftposition allerdings erst dann vorgenommen werden, wenn die aktuelle Stiftposition ausreichend weit entfernt ist.

In unserem System konnte die Methode, die von Munich & Perona zur pen-up/down Unterscheidung vorgeschlagen wurde, jedoch keine befriedigenden Ergebnisse liefern. Das liegt hauptsächlich daran, dass dabei eine exakte Bestimmung der Stiftposition vorausgesetzt wird. Diese muss genau auf der Tintenspur im aufgenommenen Bild liegen. Da die Tintenspur jedoch nur maximal ein bis drei Pixel breit ist, kann auf Grund von Messunsicherheiten, beispielsweise hervorgerufen durch ein leichtes Verrutschen des Papiers, diese Voraussetzung in unserem System nicht immer erfüllt werden. Weiterhin wird in dem System von Munich & Perona davon ausgegangen, dass die Vergleichspixel in der Umgebung der betrachteten Stiftposition dem Hintergrund zuzuordnen sind. Auch diese Annahme ist nicht notwendigerweise erfüllt, da in der Umgebung natürlich auch weitere Schriftpixel vorkommen können.

Demgegenüber wird hier ein Verfahren zur pen-up/down Unterscheidung eingesetzt, das geringe Ungenauigkeiten bei der Lokalisierung der Stiftspitze gestattet. Dabei wird, um etwaige Schriftpixel in der Umgebung der Stiftposition zu detektieren, eine 30×30 Pixel große Bildregion, die um die Stiftposition zentriert ist, mit Hilfe der Otsu-Methode (siehe Seite 29) binarisiert. Falls anhand der Binarisierung Schriftpixel erkannt werden, die nicht weiter als zwei Pixel von der extrahierten Stiftposition entfernt sind, wird angenommen, dass es sich um eine pen-down Position handelt, andernfalls wird die Position als pen-up gekennzeichnet.

Hier wird die globale Otsu-Binarisierung eingesetzt, da die lokalen Verfahren (siehe Seite 30) einen zu hohen Zeitaufwand erfordern. Da die Otsu-Binarisierung jedoch selbst dann eine Trennung von Vorder- und Hintergrund durchführt, wenn das Bild augenscheinlich keine Vordergrundpixel enthält und somit ein unimodales Histogramm aufweist, muss die Binarisierung einer “reinen” Hintergrundregion vermieden werden, da sonst ein verrauschtes Binärbild resultieren würde. Hierfür wird die Intergruppenstreuung σ_b^2 betrachtet, die im Zuge der Binarisierung berechnet wird. Diese Intergruppenstreuung ist für ein unimodales Histogramm im Vergleich zu einem bimodalen Histogramm sehr gering. Unterschreitet also die Intergruppenstreuung σ_b^2 einen vorgegebenen Schwellwert (hier: $\sigma_b^2 \leq 20$), so wird die entsprechende Stiftposition als pen-up gekennzeichnet.

Zur Vermeidung von Verdeckungen durch den Stift oder die Hand des Schreibers wird mit einem festen zeitlichen Versatz von 60 Zeitschritten (1.2 Sekunden) gearbeitet, der sich experimentell ergeben hat. Die pen-up/down Unterscheidung erfolgt also 60 Zeitschritte nachdem die betreffende Stiftposition als aktuelle Position extrahiert wurde. Im Gegensatz dazu wird im System von Munich & Perona ein Modell des von Stift und Hand überdeckten Bereichs verwendet, um zu entscheiden, ob für die betreffende Stiftposition die pen-up/down Unterscheidung vorgenommen werden kann. Dieser Ansatz hat jedoch zur Folge, das u.U. bis zum Ende des Schreibprozesses gewartet werden muss, bis also keine Verdeckungen mehr vorliegen. Damit ist eine mit dem Schreibprozess schritthaltende online Verarbeitung jedoch nicht mehr gegeben. Aus diesem Grund wird von dieser Vorgehensweise abgesehen und stattdessen die pen-up/down Unterscheidung mit einem festen zeitlichen Versatz durchgeführt. Somit können Verdeckungen zwar nicht vollständig vermieden werden, dafür kann der

Erkennungsprozess schon während des Schreibprozesses initiiert werden, sodass die Erkennungsergebnisse so zeitnah wie möglich vorliegen.

Eine weitere Schwierigkeit, die sich sowohl in diesem System als auch bei dem Verfahren von Munich & Perona ergibt, wird durch das mehrfache Überstreichen von Koordinatenpunkten mit dem Stift hervorgerufen. War der Stift bei der ersten Bewegung aufgesetzt und hat demzufolge eine Tintenspur hinterlassen, so kann beim zweiten Überstreichen der betreffenden Koordinaten nicht unterschieden werden, ob der Stift aufgesetzt war oder nicht. Da an den Stiftpositionen dann ja eine Tintenspur detektiert werden kann, würden diese Punkte somit als pen-down gekennzeichnet, auch wenn der Stift nicht aufgesetzt war. Da die nachfolgende Texterkennung jedoch strokebasiert ist, und die pen-up/down Informationen der einzelnen Stiftpositionen pro Stroke gemittelt werden, haben einzelne Fehler bei der pen-up/down Bestimmung kaum Einfluss auf das pen-up/down Merkmal des gesamten Strokes.

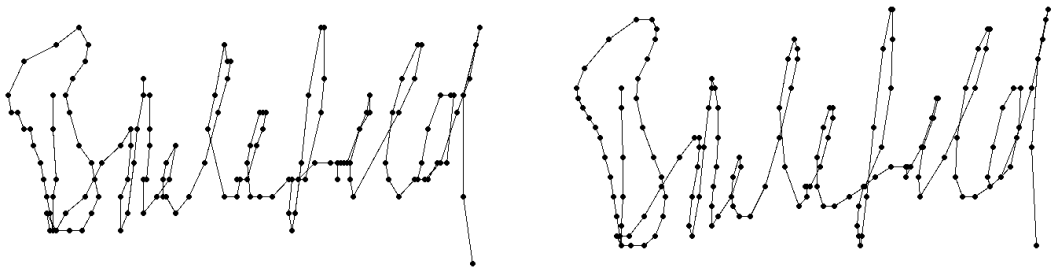
5.3 Vorverarbeitung

Nach der Stiftverfolgung liegt die mit pen-up/down Informationen annotierte Stifttrajektorie vor. Auf diese Trajektorie werden nun nacheinander einige Vorverarbeitungsschritte angewandt. Im ersten Schritt wird dabei eine Glättung des Schriftzuges durchgeführt, gefolgt von der Korrektur aufnahmebedingter geometrischer Verzerrungen und der Neuabtastung (Resampling) der Trajektorie, mit der eine von der Schreibgeschwindigkeit unabhängige Repräsentation erreicht wird.

Diese Vorverarbeitungsmaßnahmen arbeiten schritthaltend mit dem Schreibprozess, d.h. sie werden angestoßen, sobald die ersten mit pen-up/down Informationen annotierten Trajektorienpunkte vorliegen. Dies ist notwendig, damit eine schritthaltende Erkennung realisiert werden kann. Aus diesem Grund wird während der Vorverarbeitung auch keine Korrektur der Schriftgröße, -orientierung und -neigung durchgeführt, da diese Schritte nicht strokeweise, sondern nur auf der Basis eines längeren Schriftzuges robust angewandt werden können.

5.3.1 Glättung

Die aus der videobasierten Stiftverfolgung resultierende Trajektorie ist auf Grund der im Vergleich zu Digitalisiertabletts niedrigeren Abtastfrequenz und räumlichen Auflösung von geringerer Qualität. Ein wichtiger Vorverarbeitungsschritt, um aufnahmebedingtes Rauschen zu eliminieren, besteht in der Glättung der Trajektorie. Dazu wird ein *local averaging* Verfahren (siehe Seite 48) eingesetzt, bei dem mit Hilfe eines Glättungsfilters jeder Trajektorienpunkt $\boldsymbol{x}_k$ durch einen gewichteten Mittelwert $\boldsymbol{x}'_k$ ersetzt wird. Die Berechnung wird dabei durch Faltung der Filtergewichte mit den Trajektorienpunkten vorgenommen, wobei ein $2n + 1$ Punkte umfassender Abschnitt der



(a) (b)

Abbildung 5.2: (a) Ausgangstrajektorie, (b) geglättete Trajektorie.

Trajektorie betrachtet wird, der um den jeweiligen Punkt $\boldsymbol{x}_k$ zentriert ist:

$$\boldsymbol{x}'_k = \sum_{j=-n}^{n} \alpha_j \boldsymbol{x}_{k+j} \quad .$$

In dem hier realisierten System werden die Gewichtsparameter α_j des Glättungsfilters anhand der Binomialverteilung gewählt. Der Binomialfilter stellt im diskreten eine Approximation an den Gaußfilter dar, der auf Grund der Forminvarianz der Gaußverteilung unter der Fouriertransformation ein gutes Glättungsverhalten aufweist.

Da das "Nutzsignal" der Handschrift im Vergleich zum Rauschen eine niedrigere Frequenz besitzt, wird die Breite der Filtermaske entsprechend klein gewählt, sodass die niedrig-frequenten Anteile erhalten bleiben und nur die hochfrequenten Anteile eliminiert werden. In unserem Fall wird $n = 1$ gesetzt, die Breite der Filtermaske beträgt somit drei. Die Gewichtsparameter des Binomialfilters lauten demnach:

$$\alpha_{k-1} = \frac{1}{4}, \quad \alpha_k = \frac{1}{2}, \quad \alpha_{k+1} = \frac{1}{4} \quad .$$

In Abbildung 5.2 sind die Ausgangstrajektorie und die nach der Filterung resultierende Trajektorie gegenübergestellt.

5.3.2 Verzerrungskorrektur

In diesem Vorverarbeitungsschritt werden aufnahmebedingte geometrische Verzerrungen korrigiert und die extrahierten Stiftkoordinaten vom Bild- in das Schreibfeldkoordinatensystem transformiert. Die Faktoren, die eine Korrektur der Schriftgeometrie erforderlich machen, sind die folgenden:

1. Die Position der Kamera. Sie ist an der dem Schreiber gegenüberliegenden Seite des Tisches angebracht, sodass die Schrift in den aufgenommenen Bildern um 180 Grad rotiert ist.

2. Der Blickwinkel der Kamera. Der Winkel zwischen der optischen Achse der Kamera und dem Schreibfeld beträgt ca. 65 Grad (siehe Abbildung 5.3). Das

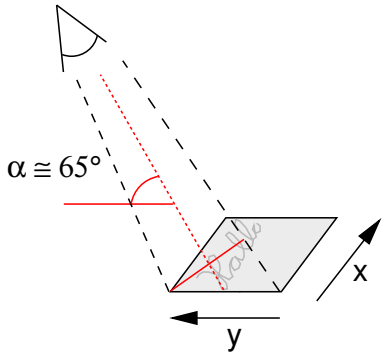


Abbildung 5.3: Schematische Darstellung der Bildaufnahme.

führt dazu, dass die extrahierte Stifttrajektorie in vertikaler Richtung gestaucht ist.

3. Der Interlace-Modus der Kamera. Um eine möglichst hohe Bildrate zu erzielen, wird die Videokamera im Interlace-Modus betrieben, sodass 50 Halbbilder pro Sekunde aufgenommen werden. Da das sogenannte EVEN-Halbbild nur aus den geradzahligen Zeilen besteht, und das sogenannte ODD-Halbbild dementsprechend nur aus den ungeradzahligen Zeilen, ist die vertikale Ausdehnung der Schrift in den aufgenommenen Halbbildern halbiert.

Um eine 180 Grad Rotation der Stifttrajektorie vorzunehmen, sind die Koordinaten anhand folgender Vorschrift zu transformieren:

$$\begin{aligned} x'_k &= B_w - x_k \\ y'_k &= B_h - y_k \quad . \end{aligned} \tag{5.9}$$

Dabei bezeichnet B_w die Bildbreite von 756 Pixel und B_h die Bildhöhe von 288 Pixel.

Zur Korrektur der vertikalen Stauchung der Schrift, die zum einen durch den Blickwinkel der Kamera von ca. 65 Grad und zum anderen durch das Interlace-Verfahren gegenüber der tatsächlichen Schriftgröße vermindert ist, sind die y-Koordinaten der Stifttrajektorie entsprechend zu skalieren:

$$\begin{aligned} x''_k &= x'_k \\ y''_k &= \frac{2y'_k}{\sin(65°)} \quad . \end{aligned} \tag{5.10}$$

Diese Korrektur stellt allerdings nur eine Näherung dar, da von der Annahme ausgegangen wird, dass der Blickwinkel von ca. 65 Grad für das gesamte Schreibfeld konstant ist. Genauer betrachtet gilt dieser Winkel jedoch nur für die Mitte des Schreibfeldes, am oberen Rand beträgt der Winkel ca. 72 Grad, am unteren Rand ca. 59 Grad. Da die Schrifthöhe im Verhältnis zur Höhe des Schreibfeldes jedoch klein ist, machen sich diese Differenzen des Blickwinkels in der resultierenden Trajektorie kaum bemerkbar,

sodass die Annahme eines über die Ausdehnung des Schreibfeldes konstanten Winkels gerechtfertigt ist.

Ebenso ist durch die perspektivische Verzerrung (Trapezverzerrung) die horizontale Ausdehnung des aufgenommenen Schreibfeldes nicht konstant, es ist in den Bildern vielmehr am oberen Rand knapp 2cm schmaler als am unteren Rand. Auch diese Differenz macht sich in den extrahierten Trajektorien nicht bemerkbar, sodass auf eine Korrektur verzichtet wurde.

Nach der Korrektur der Schriftgeometrie erfolgt eine Transformation der Stiftkoordinaten vom Bild- in das Schreibfeldkoordinatensystem. Dabei wird außerdem der Koordinatenursprung von der linken *oberen* Ecke (Bildkoordinaten) in die linke *untere* Ecke (Schreibfeldkoordinaten) verschoben. Als Maßeinheit für die Schreibfeldkoordinaten werden Mikrometer (μm) verwendet. Mit der Schreibfeldbreite $S_w = 176000\mu m$, der Schreibfeldhöhe $S_h = 149000\mu m$ und der durch Gleichung 5.10 skalierten Bildhöhe $B'_h = 2B_h/\sin(65)$ Pixel ergeben sich die neuen Koordinaten zu:

$$\begin{aligned} x'''_k &= \frac{S_w}{B_w} x''_k \\ y'''_k &= S_h - \frac{S_h}{B'_h} y''_k = S_h - \frac{S_h \sin(65)}{2B_h} y''_k \quad . \end{aligned} \tag{5.11}$$

Fasst man die obigen drei Korrektur- bzw. Transformationsschritte (5.9-5.11) zu einer Berechnungsvorschrift zusammen, so ergeben sich die neuen Koordinaten des Trajektorienpunktes wie folgt:

$$\begin{aligned} x'''_k &= \frac{S_w}{B_w}(B_w - x_k) \\ y'''_k &= \frac{S_h y_k}{B_h} \quad . \end{aligned} \tag{5.12}$$

Man erkennt, dass der Aufnahmewinkel von 65 Grad hier nicht mehr explizit auftaucht. Er geht vielmehr implizit in die beobachtbare Schreibfeldhöhe S_h ein, die ja direkt mit dem Aufnahmewinkel verknüpft ist.

5.3.3 Neuabtastung

Bei der zu diesem Zeitpunkt vorliegenden Trajektorie variieren die räumlichen Abstände zwischen den einzelnen Punkten je nach der Schreibgeschwindigkeit, die von Schreiber zu Schreiber große Unterschiede aufweisen kann. So sind bei einer konstanten Abtastrate die Abstände umso kleiner, je langsamer geschrieben wird. Die Schreibgeschwindigkeit ist jedoch ein schreiberspezifisches Merkmal, das nicht zur Klassenunterscheidung bei der Handschrifterkennung beiträgt. Aus diesem Grund wird eine Neuabtastung (engl. Resampling) der Trajektorie durchgeführt, um eine von der Schreibgeschwindigkeit unabhängige Repräsentation der Trajektorie zu erreichen.

Neben der Schreibgeschwindigkeit ist außerdem die Abtastrate des jeweiligen Aufnahmegeräts, in diesem Fall also der Videokamera, ein wesentlicher Faktor, der die

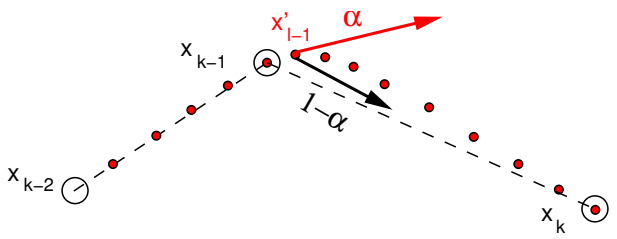


Abbildung 5.4: Neuabtastung der Stifttrajektorie.

räumlichen Abstände zwischen den Trajektorienpunkten beeinflusst. Diese Abstände sind umso geringer, je höher die Abtastrate ist. Die Neuabtastung der Stifttrajektorie dient somit auch dazu, unterschiedliche Abtastraten der Aufnahmegeräte zu kompensieren.

Die üblicherweise eingesetzten Resampling-Verfahren (siehe Seite 53) basieren auf einer *konstanten* Schrittlänge, d.h. die räumlichen Abstände zwischen aufeinanderfolgenden Punkten sind an allen Stellen der Trajektorie gleich. Da jedoch die gekrümmten Abschnitte der Trajektorie für die Erkennung besonders "interessant" sind, ist es vorteilhaft, den Abstand zwischen den Punkten entsprechend der lokalen Krümmung zu wählen, sodass an den gekrümmten Abschnitten der Trajektorie eine ausreichend hohe Auflösung sichergestellt wird. Dies wird in dem realisierten Verfahren dadurch erreicht, dass die Schrittweite an die lokale Krümmung angepasst wird. Die vorgegebene Zieldistanz zwischen aufeinanderfolgenden Punkten wird somit nur an den geraden Abschnitten der Trajektorie erreicht, an stärker gekrümmten Abschnitten werden die Punkte der lokalen Krümmung entsprechend dichter gesetzt.

Um außerdem die Glattheit der resultierenden Trajektorie zu erhöhen, wird anstelle der üblicherweise verwendeten linearen Interpolation zur Ermittlung der neuen Trajektorienpunkte $\boldsymbol{x}'_l$ ein Verfahren eingesetzt, das mittels eines *Impulsterms* die Bewegungsrichtung zum vorherigen Punkt $\boldsymbol{x}'_{l-1}$ zu einem gewissen Grad beibehält (siehe Abbildung 5.4). Durch dieses *Impuls-Resampling* werden also sprunghafte Richtungsänderungen an den neuabgetasteten Trajektorienpunkten vermieden. Dies ist insbesondere dann günstig, wenn wie in diesem Fall die Abtastfrequenz bei der Signalaufnahme gering ist, sodass große räumliche Abstände zwischen den Trajektorienpunkten vorliegen.

Anhand der in Abbildung 5.5 dargestellten Berechnungsvorschrift kann ein Impuls-Resampling mit der maximalen Schrittweite eins durchgeführt werden. Dabei bezeichnet $\boldsymbol{x}_k$ den k-ten Punkt der Ausgangstrajektorie, $\boldsymbol{x}'_l$ den l-ten Punkt der neuabgetasteten Trajektorie, $\boldsymbol{w}_l$ den momentanen Schrittvektor und α den Impulsfaktor. Ausgehend vom Startpunkt $\boldsymbol{x}'_0 = \boldsymbol{x}_0$ ergeben sich die neuen Trajektorienpunkte $\boldsymbol{x}'_l$ durch einen iterativen Prozess, wobei sich die jeweiligen Schrittvektoren $\boldsymbol{w}_l$ aus dem vorherigen Schrittvektor und der normierten Differenz zum "Zielpunkt" $\boldsymbol{x}_k$ berechnen:

$$\boldsymbol{w}_l = (1-\alpha)\frac{\boldsymbol{x}_k - \boldsymbol{x}'_{l-1}}{\|\boldsymbol{x}_k - \boldsymbol{x}'_{l-1}\|} + \alpha \boldsymbol{w}_{l-1} \qquad \text{mit } 0 \leq \alpha < 1 \quad . \tag{5.13}$$

1. Initialisierung: $k = 0, \quad l = 0$

$$\boldsymbol{x}'_0 = \boldsymbol{x}_0 \, , \quad \boldsymbol{w}_0 = 0$$

2. Wiederhole für alle $\boldsymbol{x}_k$: $k > 0$

2a. Wiederhole bis $\|\boldsymbol{x}_k - \boldsymbol{x}'_l\| < 1$:

$$\begin{aligned} \boldsymbol{w}_l &= (1-\alpha)\frac{\boldsymbol{x}_k - \boldsymbol{x}'_{l-1}}{\|\boldsymbol{x}_k - \boldsymbol{x}'_{l-1}\|} + \alpha \boldsymbol{w}_{l-1} \qquad \text{mit } 0 \leq \alpha < 1 \\ \boldsymbol{x}'_l &= \boldsymbol{x}'_{l-1} + \boldsymbol{w}_l \, , \quad l\,{+}{+} \end{aligned}$$

Abbildung 5.5: Neuabtastung mit der Schrittweite 1.

Den Spezialfall der linearen Interpolation erhält man, wenn $\alpha = 0$ gesetzt wird. Anhand der Dreiecksungleichung

$$\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

wird deutlich, dass die Schrittweite der Länge eins ($\|\boldsymbol{w}_l\| = 1$) nur an den geraden Abschnitten der Trajektorie erreicht wird, also wenn die Richtung zum "Zielpunkt" mit der momentanen Bewegungsrichtung übereinstimmt:

$$\frac{\boldsymbol{x}_k - \boldsymbol{x}'_{l-1}}{\|\boldsymbol{x}_k - \boldsymbol{x}'_{l-1}\|} = \boldsymbol{w}_{l-1} \quad .$$

Gilt diese Gleichung nicht, so ist bei $\alpha > 0$ der Abstand zwischen den neuabgetasteten Punkten kleiner als eins. Damit weist das Verfahren die gewünschte Eigenschaft auf, dass an den gekrümmten Abschnitten der Trajektorie die neuabgetasteten Punkte entsprechend "dichter" gesetzt werden.

Soll die Trajektorie mit einer größeren Schrittweite als eins neu abgetastet werden, so ist der Schritt 2 des Verfahrens aus Abbildung 5.5 zu modifizieren. Der wesentliche Unterschied ist, dass bei der Abtastung nun nur jeder N-te Punkt in die resultierende Trajektorie übernommen wird (siehe Abbildung 5.6). Anhand der durchgeführten Experimente hat sich eine maximale Schrittweite von 200 für die Neuabtastung als geeignet erwiesen. An geraden Abschnitten der Trajektorie liegen die Punkte somit $200\mu m$ auseinander, ist die Trajektorie gekrümmt, so ist der Abstand entsprechend geringer. In der Abbildung 5.7 ist eine Trajektorie nach der Neuabtastung dargestellt.

1. Initialisierung: $k = 0, \quad l = 0$

$$\boldsymbol{x}'_0 = \boldsymbol{x}_0\,, \quad \boldsymbol{w}_0 = 0\,, \quad \boldsymbol{s}_0 = \boldsymbol{x}_0\,, \quad n = 0;$$

2. Wiederhole für alle $\boldsymbol{x}_k$: $k > 0$

2a. Wiederhole bis $\|\boldsymbol{x}_k - \boldsymbol{x}'_l\| < N$:

$$\begin{aligned} \boldsymbol{w}_n &= (1-\alpha)\frac{\boldsymbol{x}_k - \boldsymbol{s}_{n-1}}{\|\boldsymbol{x}_k - \boldsymbol{s}_{n-1}\|} + \alpha \boldsymbol{w}_{n-1} \qquad \text{mit } 0 \leq \alpha < 1 \\ \boldsymbol{s}_n &= \boldsymbol{s}_{n-1} + \boldsymbol{w}_n\,, \quad n\,{+}{+} \\ \boldsymbol{x}'_l &= \boldsymbol{s}_n\,, \quad l\,{+}{+} \quad \textbf{falls } n \text{ modulo } N = 0 \end{aligned}$$

Abbildung 5.6: Neuabtastung mit Schrittweite N.

5.4 Segmentierung

Nach Durchführung der Vorverarbeitungsmaßnahmen wird die Stifttrajektorie nun einem Segmentierungsschritt unterzogen, sodass anschließend die Segmente der Trajektorie vorliegen, anhand derer die Merkmalsberechnung vorgenommen werden kann. In Kapitel 2.1 wurde bereits darauf hingewiesen, dass die Handschrift aus Basiseinheiten (Strokes) zusammengesetzt ist, die sich robust auf Basis der Schreibdynamik extrahieren lassen. Diese Strokes sind durch einen glockenförmigen Verlauf des Betrags der Schreibgeschwindigkeit gekennzeichnet, sodass eine Möglichkeit zur Segmentierung von Strokes die Detektion von lokalen Minima der Schreibgeschwindigkeit darstellt. Die so vorgenommene Bestimmung der Segmentgrenzen ist außerdem invariant gegenüber der Schriftgröße, sodass bei einer geeigneten Merkmalsrepräsentation eine von der Schriftgröße unabhängige Erkennung durchgeführt werden kann [Dol97].

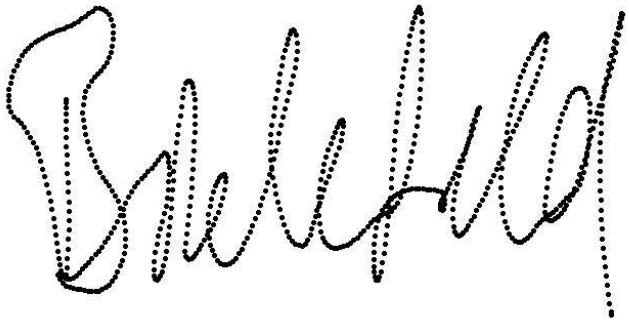


Abbildung 5.7: Trajektorie nach der Neuabtastung

Um auch Überlagerungen von Strokes auflösen zu können, wurde in [Gue98] eine modellbasierte Segmentierung mittels des *Delta-Log-Normal-Modells* (siehe Abschnitt 2.1.2) vorgeschlagen. Da hierbei jedoch die Modellparameter iterativ in einem relativ aufwendigen *Analyse-durch-Synthese* Ansatz geschätzt werden müssen, wurde aus Effizienzgründen von der modellbasierten Segmentierung abgesehen.

Da die Informationen über die Schreibgeschwindigkeit auf Grund der Neuabtastung der Trajektorie zu diesem Zeitpunkt jedoch nicht mehr vorliegen, basiert das hier eingesetzte Segmentierungsverfahren auf der lokalen Krümmung der Stifttrajektorie. Die Verwendung der Krümmung als Segmentierungskriterium ist deshalb gerechtfertigt, da die Krümmung eng mit der Schreibgeschwindigkeit verknüpft ist. So gilt für stückweise elliptische Bewegungen folgender Zusammenhang (siehe u.a. [Pla98b]):

$$|\boldsymbol{v}(t)| = a \left(\frac{1}{C(t)} \right)^{\frac{2}{3}} \tag{5.14}$$

Dabei bezeichnet $\boldsymbol{v}(t)$ den Betrage der Geschwindigkeit, $C(t)$ die Krümmung und a einen konstanten Faktor. Je höher also der Betrag der Schreibgeschwindigkeit ist, desto niedriger ist die lokale Krümmung der Trajektorie. Lokale Minima der Schreibgeschwindigkeit entsprechen somit lokalen Maxima der Krümmung und umgekehrt.

Die Berechnung der lokalen Krümmung C_k am Punkt $\boldsymbol{x}_k$ der Trajektorie basiert auf dem Winkel ρ_k, der durch die drei aufeinanderfolgenden Punkte $\boldsymbol{x}_{k-1}, \boldsymbol{x}_k$ und $\boldsymbol{x}_{k+1}$ gegeben ist. Mit den Vektoren

$$\boldsymbol{d}_k = \frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|}, \quad \boldsymbol{d}_{k+1} = \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|}$$

berechnet sich der Kosinus des eingeschlossenen Winkels ρ_k durch das Skalarprodukt:

$$\cos \rho_k = \boldsymbol{d}_k \cdot \boldsymbol{d}_{k+1} \quad .$$

Die Krümmung C_k ergibt sich daraus wie folgt:

$$C_k = 1 - \cos \rho_k \quad . \tag{5.15}$$

Der Wertebereich der Krümmung ist somit durch das Intervall $[0 \ldots 2]$ gegeben. Die Krümmung ist Null, wenn die Richtungen von $\boldsymbol{d}_k$ und $\boldsymbol{d}_{k+1}$ übereinstimmen. Sind die Richtungen entgegengesetzt, so ist C_k maximal, eine Krümmung von eins ergibt sich, wenn $\boldsymbol{d}_k$ und $\boldsymbol{d}_{k+1}$ einen rechten Winkel bilden.

In dem hier eingesetzten Verfahren werden nun diejenigen Punkte der Trajektorie als Segmentgrenzen markiert, an denen die lokale Krümmung C_k einen vorgegebenen Schwellwert $C_{\text{max}} = 0.03$ überschreitet. Da darüberhinaus diejenigen Punkte, an denen die vertikale Komponente der Schreibrichtung das Vorzeichen wechselt, auch bei einer geringeren Krümmung als Segmentgrenzen angenommen werden können, wird in diesen Fällen mit einem zweiten, kleineren Schwellwert $C_{\text{max\_vert}} = 0.001$ gearbeitet. Damit wird eine Segmentgrenze detektiert, wenn die folgende Bedingung gilt:

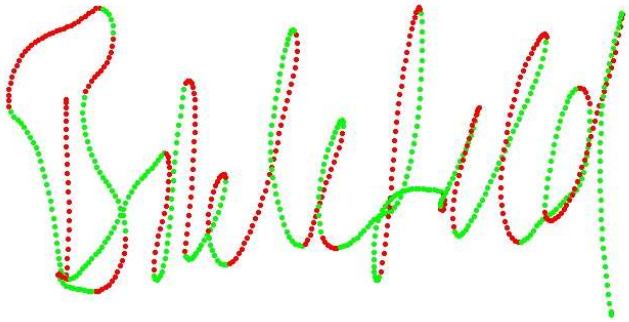


Abbildung 5.8: Segmentierte Strokes der Trajektorie.

1. $C_k > C_{\text{max}} = 0.03$ ODER

2. $(y_k - y_{k-1}) \cdot (y_{k+1} - y_k) < 0$ UND $C_k > C_{\text{max\_vert}} = 0.001$.

Die resultierende Segmentierung in Strokes, die sich anhand dieser Kriterien ergibt, ist an einem Beispiel in Abbildung 5.8 dargestellt.

5.5 Merkmalsextraktion

Anhand der im vorherigen Verarbeitungsschritt extrahierten Segmente der Stifttrajektorie wird nun eine Reihe von Merkmalen bestimmt, die anschließend zur Klassifikation genutzt werden. Da zur Ermöglichung einer schritthaltenden Erkennung während der Vorverarbeitung keine Normalisierung der Schriftgröße durchgeführt wurde, wird eine Merkmalsrepräsentation angestrebt, die invariant bezüglich der Schriftgröße ist.

Die Grundlage der Merkmalsberechnung bilden die Abschnitte der Stifttrajektorie, die zum einen aus den Segmenten selbst bestehen, sich zum anderen aus der 50%-igen Überdeckung zweier aufeinanderfolgender Segmente ergeben. Diese Definition der Trajektorienabschnitte wurde u.a. von Dolfing & Haeb-Umbach [Dol97] vorgeschlagen und ist dadurch motiviert, dass gerade die Bereiche der Segment*grenzen*, die somit durch eine starke Krümmung gekennzeichnet sind, eine wichtige Rolle für die Erkennung spielen.

Auf Basis der so definierten Abschnitte der Trajektorie wird jeweils ein 17-dimensionaler Merkmalsvektor extrahiert. Die einzelnen Komponenten dieses Vektors, die im folgenden näher beschrieben werden, sind größtenteils an die Merkmale angelehnt, die in [Jae01] zur Erkennung verwendet werden.

Richtung

Als Richtungsmerkmal wird der Sinus und Kosinus der *mittleren Schreibrichtung* des Abschnitts s der Trajektorie verwendet:

$$\cos\alpha_s = \frac{1}{N_s}\sum_{k=1}^{N_s}\cos\alpha_k \tag{5.16}$$

$$\sin\alpha_s = \frac{1}{N_s}\sum_{k=1}^{N_s}\sin\alpha_k \quad . \tag{5.17}$$

Dabei bezeichnet N_s die Anzahl der Punkte $\boldsymbol{x}_k$ im Segment s, und $\cos\alpha_k$ bzw. $\sin\alpha_k$ beschreiben den Kosinus bzw. Sinus der *lokalen* Schreibrichtung am Punkt $\boldsymbol{x}_k$ des Abschnitts s:

$$\cos\alpha_k = \frac{\Delta x_k}{\Delta s_k}\ ,\ \sin\alpha_k = \frac{\Delta y_k}{\Delta s_k} \quad .$$

Mit

$$\begin{aligned}\Delta x_k &= x_{k-1} - x_{k+1}\\ \Delta y_k &= y_{k-1} - y_{k+1}\\ \Delta s_k &= \sqrt{\Delta x_k^2 + \Delta y_k^2} \quad .\end{aligned}$$

Die Verwendung von Sinus und Kosinus anstatt des Winkels selbst als Merkmal ist auf Grund der Periodizität von Sinus und Kosinus vorteilhaft, da so der "Sprung" von $359°$ auf $0°$ vermieden werden kann.

Krümmung

In ähnlicher Weise ergibt sich der Sinus und Kosinus der *mittleren Krümmung* des Abschnitts s der Trajektorie aus den lokalen Krümmungen:

$$\cos\beta_s = \frac{1}{N_s}\sum_{k=1}^{N_s}\cos\beta_k \tag{5.18}$$

$$\sin\beta_s = \frac{1}{N_s}\sum_{k=1}^{N_s}\sin\beta_k \quad . \tag{5.19}$$

Die lokale Krümmung $\cos\beta_k$ bzw. $\sin\beta_k$ beschreibt die Differenz der Schreibrichtung an den Punkten $\boldsymbol{x}_{k-1}$, $\boldsymbol{x}_{k+1}$ und lässt sich somit anhand der lokalen Richtungen $\cos\alpha_k$ bzw. $\sin\alpha_k$ wie folgt bestimmen:

$$\begin{aligned}\cos\beta_k &= \cos\alpha_{k-1}\cos\alpha_{k+1} + \sin\alpha_{k-1}\sin\alpha_{k+1}\\ \sin\beta_k &= \cos\alpha_{k-1}\sin\alpha_{k+1} - \sin\alpha_{k-1}\cos\alpha_{k+1} \quad .\end{aligned}$$

Pen-up/down Merkmal

Während der Extraktion der Stifttrajektorie wird für jeden einzelnen Trajektorienpunkt die pen-up/down Detektion durchgeführt. Die Trajektorienpunkte werden dabei jeweils mit einem binären Wert p_k annotiert ($p_k = 0$ (pen-down), $p_k = 1$ (pen-up)).

Zur abschnittsweisen Berechnung des pen-up/down Merkmals wird nun über die Werte p_k eines Abschnitts gemittelt:

$$p_s = \frac{1}{N_s} \sum_{k=1}^{N_s} p_k \quad . \tag{5.20}$$

Das Ergebnis p_s ist somit ein kontinuierlicher Wert im Intervall $[0, \ldots, 1]$.

Seitenverhältnis

Das Seitenverhältnis der *bounding box* des Abschnitts s beschreibt das Verhältnis von Höhe h_s zur Breite w_s des Abschnitts. Um das auf das Intervall $[-1, \ldots, 1]$ normierte Seitenverhältnis zu berechnen, wird die folgende Berechnungsvorschrift angewendet.

$$r_s = \frac{2h_s}{w_s + h_s} - 1 \quad . \tag{5.21}$$

Curliness

Das Merkmal *Curliness* beschreibt die Abweichung der Trajektorie im Abschnitt s von einer Geraden. Die Curliness basiert auf dem Verhältnis der Trajektorienlänge L_s zur größten Ausdehnung der bounding box $\max(w_s, h_s)$:

$$\phi_s = \frac{L_s}{\max(w_s, h_s)} - 2 \text{ , mit } \quad L_s = \sum_{k=1}^{N_s - 1} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| \quad . \tag{5.22}$$

Mit dieser Definition umfasst der Wertebereich der Curliness das Intervall $[-1, \ldots, N_s - 3]$, wobei in der Praxis kaum Werte auftreten, die größer als eins sind [Jae01].

Delta-Merkmale

Um einen größeren Kontext mit in die Merkmalsberechnung einzubeziehen, werden mit den Delta-Merkmalen approximierte Ableitungen der Merkmale 5.16-5.22 berechnet.

$$\Delta \cos \alpha_s = \cos \alpha_s - \cos \alpha_{s-1} \tag{5.23}$$

$$\Delta \sin \alpha_s = \sin \alpha_s - \sin \alpha_{s-1} \tag{5.24}$$

$$\Delta \cos \beta_s = \cos \beta_s - \cos \beta_{s-1} \tag{5.25}$$

$$\Delta \sin \beta_s = \sin \beta_s - \sin \beta_{s-1} \tag{5.26}$$

$$\Delta p_s = p_s - p_{s-1} \tag{5.27}$$
$$\Delta r_s = r_s - r_{s-1} \tag{5.28}$$
$$\Delta \phi_s = \phi_s - \phi_{s-1} \tag{5.29}$$

Delayed-Merkmale

Auf Basis der Arbeit von Dolfing [Dol97] werden außerdem sogenannte *delayed* Merkmale berechnet, die die Lage- und Größenrelationen der Trajektorienabschnitte über eine größere Distanz Δs hinweg beschreiben.

Die Lagerelation wird durch den Winkel ($\angle$) der Verbindungsgeraden der Abschnittsschwerpunkte (center of gravity, cog) ausgedrückt:

$$\sin d_\theta = \sin \angle(\text{cog}_s, \text{cog}_{s-\Delta s}) \tag{5.30}$$
$$\cos d_\theta = \cos \angle(\text{cog}_s, \text{cog}_{s-\Delta s}) \quad . \tag{5.31}$$

Die Größenrelation basiert auf der Änderung der vertikalen Ausdehnung h_s der Trajektorienabschnitte s:

$$d_{\text{size}} = \frac{h_s}{h_s + h_{s-\Delta s}} \quad . \tag{5.32}$$

Anhand der Experimente hat sich für Δs ein Wert von zwei als geeignet herausgestellt.

5.6 Statistische Modellierung und Erkennung

Die Klassifikation der Merkmalsvektorfolgen basiert auf Hidden Markov Modellen (HMMs), wobei die Konfiguration, Parameteroptimierung und Dekodierung der HMMs mit Hilfe der ESMERALDA-Entwicklungsumgebung vorgenommen wird [Fin99].

Auf Grund der im Vergleich zu Wortmodellen größeren Flexibilität im Hinblick auf die Erweiterbarkeit des Erkennungslexikons werden hier Buchstabenmodelle zur Klassifikation eingesetzt. Um eine Worterkennung durchzuführen, werden die einzelnen Buchstabenmodelle dann zu Verbundmodellen zusammengeschaltet (siehe Abbildung 3.25). Diese Vorgehensweise, die Verwendung eines gemeinsamen Satzes von Buchstabenmodellen, hat gegenüber der Verwendung von Wortmodellen bei umfangreicheren Lexika außerdem den Vorteil, dass für die einzelnen HMMs mehr Trainingsdaten zur Verfügung stehen.

Modelltopologie

Die verwendeten HMMs sind durch eine Bakis-Topologie gekennzeichnet (siehe Abbildung 3.24). Von Null verschiedene Zustandsübergangswahrscheinlichkeiten treten damit nur bei Selbstübergängen und bei Zustandsübergängen der Form $s_i \rightarrow s_{i+1}$ und

$s_i \rightarrow s_{i+2}$ auf. Diese Topologie bietet im Vergleich zu linearen Modellen mehr Flexibilität, da auch Zustände übersprungen werden können. Diese Eigenschaft ist insbesondere bei schreiberunabhängigen Systemen vorteilhaft, wenn beispielsweise den Buchstaben in der Trainingsmenge längere Merkmalsvektorfolgen zuzuordnen sind als den entsprechenden Buchstaben in der Testmenge.

Die Anzahl der Zustände der HMMs hängt von dem zu modellierenden Buchstaben ab. Sie richtet sich nach der minimalen Länge der Merkmalsvektorfolge, die der betreffende Buchstabe in der Initialisierungsstichprobe aufweist. So besteht beispielsweise das HMM für den Buchstaben `a` aus sieben Zuständen, während das HMM für das `m` zehn Zustände umfasst.

Emissionsmodellierung

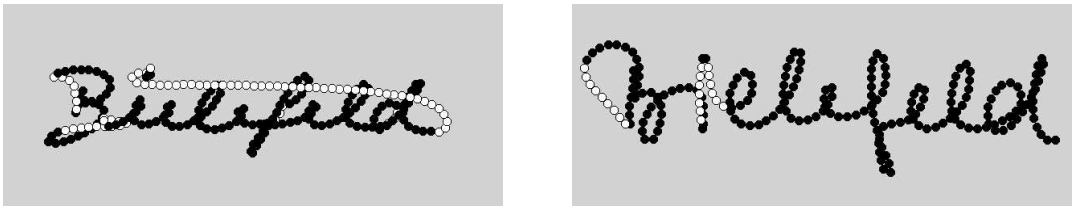
Neben der Modelltopologie ist die Wahl der Emissionsmodellierung ein wesentlicher Aspekt beim Entwurf von HMMs. Auf Grund der guten Approximationseigenschaft bei gleichzeitig handhabbarem Parameterumfang wird hier eine semikontinuierliche Emissionsmodellierung auf Basis einer Gauß'schen Mischverteilungsdichte verwendet (siehe Gleichung 3.65). Die Parameter der Mischverteilungsdichte, d.h. die Anzahl der Gaußdichten, ihre jeweiligen Mittelwertvektoren und Kovarianzmatrizen, werden anhand der Merkmalsvektorfolgen der Trainingsstichprobe mit Hilfe des LBG-Algorithmus zur Vektorquantisierung geschätzt. In diesem Fall resultiert daraus eine Mischverteilungsdichte, die 152 Komponenten umfasst. Um die Anzahl der zu schätzenden Parameter weiter einzuschränken, finden hier nur diagonale Kovarianzmatrizen Verwendung.

Modellierung diakritischer Zeichen

Die diakritischen Zeichen (delayed strokes), also der Punkt beim "i" bzw. "j", der "t"-Strich, und die Umlautpunkte, werden als spezielle Buchstaben aufgefasst und mittels zweier "Diakritika-HMMs" modelliert. Dieses Vorgehen ist angelehnt an die Arbeit von Hu & Kollegen [Hu00] und vermeidet die fehleranfällige Detektion von delayed strokes in der Vorverarbeitungsphase.

Da hier keinerlei Umsortierung der Strokes vorgenommen wird, müssen unterschiedliche Realisierungen in Bezug auf den zeitlichen Zusammenhang zwischen dem diakritischen Zeichen und dem zugehörigen Basisbuchstaben berücksichtigt werden. So können die Diakritika theoretisch zu einem beliebigen Zeitpunkt angefügt werden. Um die Komplexität einzuschränken, werden hier jedoch nur die beiden Fälle betrachtet, in denen die delayed strokes entweder unmittelbar nach dem Basisbuchstaben oder erst am Ende des Wortes realisiert werden.

In das Erkennungslexikon werden somit für jedes Wort, das diakritische Zeichen enthält, mindestens zwei Einträge aufgenommen, sodass die obigen beiden Realisierungsmöglichkeiten abgedeckt werden. Enthält das entsprechende Wort mehr



B ı e l e f e l d · B ı · e l e f e l d

Abbildung 5.9: Varianten zur Modellierung diakritischer Zeichen

als ein diakritisches Zeichen, werden außerdem Mischformen der beiden Realisierungsmöglichkeiten berücksichtigt.

Initialisierung, Training & Dekodierung der HMMs

Die Initialisierung der HMMs wird mittels einer Teilmenge der Trainingsstichprobe durchgeführt, die von Hand auf Buchstabenebene annotiert wurde. Die so gewonnene Initialisierungsstichprobe, die insgesamt 144 manuell annotierte Wörter von zwei unterschiedlichen Schreibern umfasst, ist dabei jedoch nur für die Realisierung eines *ersten* funktionsfähigen Erkennungssystems erforderlich. Mit Hilfe dieses Systems kann eine semiautomatische Annotierung der Daten auf Buchstabenebene vorgenommen werden, sodass im weiteren ein größerer Datenumfang zur Initialisierung genutzt werden kann.

Das Training der HMMs wird mit Hilfe des Baum-Welch Algorithmus vorgenommen (siehe Seite 85). Dazu können auf Wortebene annotierte Daten verwendet werden, da vor jeder Trainingsiteration ein Viterbi-Schritt durchgeführt wird, um eine Zuordnung der Signalabschnitte zu den entsprechenden Buchstabenmodellen vorzunehmen.

Zur Dekodierung des Erkennungsmodells wird der Standard Viterbi Beam-Search Algorithmus verwendet. Aus Effizienzgründen ist dabei das Erkennungsmodell, das eine Parallelschaltung der einzelnen Verbundmodelle (siehe Abbildung 3.25) darstellt, unter Ausnutzung der Präfixäquivalenzen der Verbundmodelle als Baum organisiert. Ein Ausschnitt des baumförmigen Erkennungsmodells ist in Abbildung 5.10 veranschaulicht. Untersuchungen aus dem Bereich der Sprachverarbeitung haben gezeigt, dass durch die baumförmige Organisation eine bis zu siebenfache Beschleunigung der Beam-Search Dekodierung erreicht werden kann (siehe u.a. [Sch95b], Seite 257).

5.7 Adaption

Idealerweise liegen zum Parametertraining eines schreiberunabhängigen Systems große Mengen von Videobilddaten vor, die während einer Vielzahl unterschiedlicher Schreibprozesse aufgenommen wurden. Da bei der Aufzeichnung einer einminütigen Videobildsequenz jedoch schon ein Datenvolumen von ca. 650 MByte anfällt,

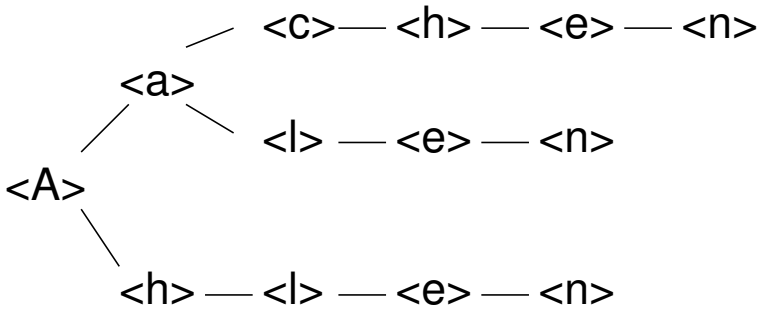


Abbildung 5.10: Beispiel eines baumförmigen Erkennungsmodells.

das selbst bei der heutigen Speichertechnologie noch recht unhandlich ist, wurde nur die Evaluierungsstichprobe mit der Videokamera aufgenommen, die Stichprobe zur Initialisierung und Parameteroptimierung der HMMs wurde dagegen mit Hilfe eines WACOM-Digitalisiertabletts erstellt. Durch diesen *Mismatch* zwischen den Trainings- und den Testbedingungen wird die Komplexität der Erkennungsaufgabe erhöht. Daher wird neben der Normalisierung in der Vorverarbeitungsphase zur Verbesserung der Signalqualität (Glättung, Korrektur der aufnahmebedingten geometrischen Verzerrung, Impuls-Neuabtastung) außerdem eine Adaption der HMM-Parameter vorgenommen.

Aufgrund des geringen Umfangs der videobasierten Stichprobe wird hier von einer MAP-Adaption abgesehen und stattdessen eine MLLR-Adaption durchgeführt, wobei nur die Mittelwertvektoren der Gauß'schen Mischverteilungsdichte transformiert werden. Dabei wird von den in Abschnitt 3.8.1 beschriebenen Vereinfachungen ausgegangen, dass die Kovarianzmatrizen der Gaußdichten identisch sind und die Zuordnung der Observationen zu den HMM-Zuständen in eindeutiger, nicht-probabilistischer Weise vorgenommen wird. Weiterhin wird wie in [Fis99] nur eine Regressionsklasse definiert, sodass die Mittelwertvektoren aller Gaußdichten mit einer Transformationsmatrix adaptiert werden.

Mit diesen Vereinfachungen ergibt sich die Gleichung 3.106 zur Schätzung der Transformationsmatrix zu:

$$\sum_{t=1}^{T} \boldsymbol{x}_t \hat{\boldsymbol{\mu}}_{q_t}^T = \widehat{\boldsymbol{W}} \sum_{t=1}^{T} \hat{\boldsymbol{\mu}}_{q_t} \hat{\boldsymbol{\mu}}_{q_t}^T \quad . \tag{5.33}$$

Die Transformationsmatrix $\widehat{\boldsymbol{W}}$ ist damit gegeben durch:

$$\widehat{\boldsymbol{W}} = \left(\sum_{t=1}^{T} \boldsymbol{x}_t \hat{\boldsymbol{\mu}}_{q_t}^T \right) \left(\sum_{t=1}^{T} \hat{\boldsymbol{\mu}}_{q_t} \hat{\boldsymbol{\mu}}_{q_t}^T \right)^{-1} \quad . \tag{5.34}$$

Das Adaptionsverfahren wird dabei sowohl im überwachten als auch im unüberwachten Modus durchgeführt. Der Unterschied liegt darin, dass bei der überwachten Adaption neben der Observationsfolge auch ihre korrekte Transkription vorgegeben wird, wohingegen bei der unüberwachten Adaption die u.U. fehlerhaften Erkennungs-

ergebnisse zur Berechnung der Transformationsmatrizen verwendet werden. Die überwachte Adaption ist somit aufgrund der erforderlichen Transkription nur bedingt anwendbar. Dagegen bietet die unüberwachte Adaption mehr Flexibilität, jedoch sind die dabei zu erwartenden Verbesserungen der Erkennungsergebnisse geringer.

5.8 Zusammenfassung

In diesem Kapitel wurden die Verfahren vorgestellt, die in dem realisierten System zur videobasierten online Handschrifterkennung eingesetzt werden. Die Grundlage der online Erkennung ist die Extraktion der Stifttrajektorie aus den Videobildfolgen. Dazu wird ein Template-Matching Ansatz verwendet, der angelehnt ist an das System von Munich & Perona, wobei gegenüber diesem einige Verfahrensschritte – insbesondere im Hinblick auf die Initialisierung – modifiziert wurden.

Aufgrund der im Vergleich zu Digitalisiertabletts geringeren räumlichen und zeitlichen Auflösung wird die Qualität der extrahierten Stifttrajektorie mit Hilfe einiger Vorverarbeitungsschritte verbessert. Dabei wird die aufnahmebedingte geometrische Verzerrung korrigiert, sowie eine Glättung durch Binomialfilterung und Impuls-Neuabtastung der Trajektorie durchgeführt.

Anhand der vorverarbeiteten Trajektorie wird anschließend eine Segmentierung in Strokes vorgenommen, die im darauffolgenden Schritt die Grundlage für die Extraktion von Merkmalen bilden. Die Klassifikation der Merkmalsvektorfolgen erfolgt auf Basis von Hidden Markov Modellen. Die Behandlung von diakritischen Zeichen (delayed strokes) geschieht auf der Modellierungsebene durch Verwendung von “Diakritika-HMMs” unter Berücksichtigung mehrerer Realisierungsmöglichkeiten derjenigen Wörter im Lexikon, die diakritische Zeichen enthalten. Um den Mismatch zwischen den Trainingsbedingungen (tablettbasierte Daten) und den Anwendungsbedingungen (videobasierte Daten) zu kompensieren, wird eine Adaption der HMM-Parameter durchgeführt.

Die Beurteilung der Leistungsfähigkeit der Verfahrensschritte erfolgt anhand einer Reihe von Experimenten, wobei als Bewertungskriterium die erzielten Fehlerraten auf der videobasierten Teststichprobe verwendet wird. Die detaillierte Beschreibung der durchgeführten Experimente und die Vorstellung der Ergebnisse wird im Evaluationskapitel, Abschnitt 7.2, vorgenommen.

6 Inkrementelle videobasierte offline Handschrifterkennung

Das im vorherigen Kapitel beschriebene System zur videobasierten online Handschrifterkennung basiert auf der Erfassung der Schreibdynamik. Die Grundlage jenes Systems ist die Extraktion der Stifttrajektorie, also die Ermittlung des zeitlichen Verlaufs der Stiftposition anhand der Videobildfolge. Die Voraussetzung dabei ist, dass die Stiftspitze stets in den aufgenommenen Bildern sichtbar ist. Damit geht jedoch eine bedeutende Einschränkung der Anwendbarkeit einher, da diese Vorbedingung bei natürlichen Schreibvorgängen nicht immer erfüllt ist, bzw. nur durch eine spezielle, u.U. schreiberabhängige Positionierung der Kamera erfüllt werden kann.

Ein Beispiel für ein Szenario, bei dem die videobasierte online Handschrifterkennung kaum anwendbar ist, stellt das Schreiben an einer Wandtafel (Whiteboard) dar. Hier ist der Stift häufig vom Schreiber verdeckt, sodass das Geschriebene erst dann sichtbar wird, wenn bereits an einem anderen Bereich des Whiteboards geschrieben wird. In diesem Fall kann die Dynamik der Schreibbewegung nicht mit einer Videokamera erfasst werden, sodass vielmehr Methoden aus dem Bereich der offline Handschrifterkennung anzuwenden sind, die also auf dem statischen Schriftbild basieren.

In diesem Kapitel wird nun ein System vorgestellt, das auch in uneingeschränkteren Szenarien, wie z.B. der Erkennung von Tafelanschrift, eingesetzt werden kann (siehe auch [Wie02, Wie03]). Das System ist gekennzeichnet durch eine inkrementelle Verarbeitungsstrategie, bei der – ähnlich wie im online System – der Schreibprozess fortwährend mit einer Videokamera beobachtet wird. Die Verarbeitung basiert hier jedoch nicht auf der Schreibdynamik, sondern vielmehr auf den Abbildern der zum jeweiligen Zeitpunkt neu erfassten Schriftabschnitte.

6.1 Anwendungsszenario & Systemaufbau

Das Anwendungsszenario des realisierten Systems zur inkrementellen videobasierten offline Handschrifterkennung ist die Erkennung von Tafelanschrift. Diese Anwendung ist einerseits auf Grund der wachsenden Popularität von Whiteboards in Büro- und Besprechungsräumen von hohem Interesse, andererseits ist hier das videobasierte online System durch die häufigen Stiftverdeckungen nicht anwendbar.

Das in diesem Kapitel vorgestellte System stellt damit eine Erweiterung der in Abschnitt 4.1 kurz angesprochenen videobasierten Whiteboard-Scanner dar. Jene Systeme werden üblicherweise zur Erkennung von wenigen “Kommando-Symbolen” eingesetzt, die jeweils einer bestimmte Aktion zugeordnet sind, wie z.B. das Abfotografieren

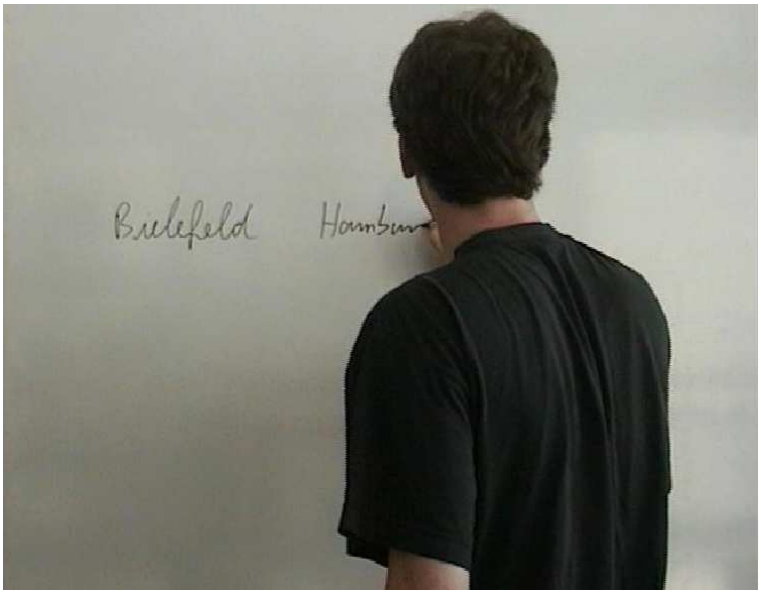


Abbildung 6.1: Momentaufnahme eines Schreibvorgangs am Whiteboard

des Whiteboards. Im Gegensatz dazu kann mit dem hier beschriebenen System direkt der an das Whiteboard geschriebene Text erkannt werden.

Die Kamera ist in dem realisierten System in einem Abstand von ca. dreieinhalb Metern frontal vor dem Whiteboard positioniert, sodass die aufnahmebedingten geometrischen Verzerrungen so weit wie möglich minimiert werden. Die Bildaufnahmerate der Kamera beträgt fünf Bilder pro Sekunde. Der Zoomfaktor ist so eingestellt, dass der beobachtbare Bereich an dem Whiteboard ca. 70cm breit und ca. 50 cm hoch ist. Diese Einstellung ist ein Kompromiss aus einer möglichst großen Schreibfläche einerseits und einer möglichst guten Lesbarkeit auch bei kleinen Schriftgrößen andererseits[1]. Bei dem verwendeten Bildformat von 756×576 Pixel liegt die Bildauflösung damit bei ca. 30 dpi – ein Zehntel der Auflösung, die bei einem Scanner üblicherweise angewendet wird. Eine spezielle Beleuchtung ist bei der Bildaufnahme nicht erforderlich, d.h. es kann sowohl mit Tageslicht als auch mit künstlicher Bürobeleuchtung gearbeitet werden. Als Stifte eignen sich herkömmliche Board-Marker. In der Abbildung 6.1 ist eine Momentaufname eines Schreibvorgangs dargestellt.

6.2 Inkrementelle Verarbeitungsstrategie

Im Gegensatz zu der herkömmlichen offline Handschrifterkennung, bei der nach Beendigung des Schreibvorgangs das fertiggestellte Dokument auf Knopfdruck aufgenom-

[1] Die Größe der verfügbaren Schreibfläche könnte durch Mosaiktechniken erhöht werden. Da die rechenintensive Erstellung von Mosaikbildern jedoch zu längeren Antwortzeiten und damit einer eingeschränkten Interaktivität führen würde, wurde in dem vorliegenden System davon abgesehen.

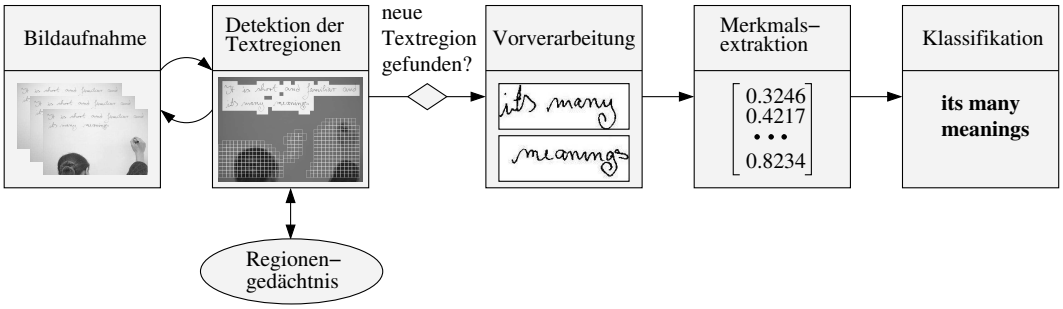


Abbildung 6.2: Architektur des Systems zur videobasierten inkrementellen offline Handschrifterkennung

men und weiterverarbeitet wird, ist das hier beschriebene System durch eine schritthaltende Verarbeitung charakterisiert. Dabei wird der Schreibvorgang fortwährend beobachtet, sodass der Erkennungsprozess gestartet werden kann, sobald eine Schriftregion im Bild detektiert wird. Die einzelnen Schriftabschnitte werden damit schrittweise, in der Reihenfolge ihres Auftauchens in der Bildfolge, klassifiziert und zum Gesamterkennungsergebnis integriert.

Dieses schrittweise Vorgehen kann damit als *inkrementelle offline* Erkennung bezeichnet werden. Das Attribut “offline” soll in diesem Zusammenhang darauf hinweisen, dass die Grundlage für die Erkennung das Schriftbild ist und nicht die Dynamik der Schreibbewegung. Das Verfahren lässt sich somit zwischen der “reinen” online Erkennung einordnen, die auf der Dynamik der Schreibbewegung basiert, und der “reinen” offline Erkennung, die auf dem vollständigen Dokumentenabbild beruht und nach der Beendigung des Schreibprozesses durchgeführt wird.

Die Architektur des offline Erkennungssystems ist in Abbildung 6.2 veranschaulicht. In einem Verarbeitungszyklus erfolgt nach der Bildaufnahme die Detektion der Textregionen. Wird dabei im Verlauf des Schreibvorgangs eine neue Textregion detektiert, so wird daraufhin der Erkennungsprozess für diesen Schriftabschnitt initiiert. Außerdem wird dieser Schriftabschnitt im Regionengedächtnis gespeichert, um zu verhindern, dass für eine einmal klassifizierte Textregion der Erkennungsprozess zu späteren Zeitpunkten erneut durchgeführt wird. Nach der Vorverarbeitung des Schriftabschnitts zur Schriftnormalisierung erfolgt die Merkmalsextraktion und schließlich die Klassifikation.

6.3 Detektion der Textregionen

Der erste Verarbeitungsschritt nach der Bildaufnahme ist die Detektion derjenigen Bildbereiche, die Schrift enthalten. Um dies möglichst schnell und robust vornehmen zu können, wird ein zweistufiges Verfahren eingesetzt. Mit dem ersten Verarbeitungsschritt wird dabei eine grobe, dafür schnelle Partitionierung des Bildes in Textbereiche und Hintergrund- bzw. Störbereiche vorgenommen. Im zweiten Schritt werden dann

die Schriftkomponenten (die Tintenspur) benachbarter Textbereiche zusammengefasst, sodass schließlich Bildregionen resultieren, die einzelnen Wörtern bzw. Textzeilen entsprechen.

6.3.1 Partitionierung des Bildes

Das Ziel des ersten Verarbeitungsschritts zur Detektion von Textregionen ist eine schnelle Vorauswahl von Bildbereichen, die Schrift enthalten. Dazu wird das aufgenommene Grauwertbild $I(x, y)$ in 40×40 Pixel große Blöcke eingeteilt, die sich jeweils um 20 Pixel überlappen. Anhand dieser Blöcke werden dann dreidimensionale Merkmalsvektoren zur Unterscheidung von Text-, Hintergrund- und Störbereichen extrahiert, wobei die einzelnen Komponenten der Merkmalsvektoren durch die folgenden charakteristischen Eigenschaften der Regionen motiviert sind.

Die Textbereiche sind vorwiegend dadurch gekennzeichnet, dass sie zusammenhängende Kantenpixel – hervorgerufen durch einen Teil der Schriftkontur – aufweisen, sich über wenige Zeitschritte kaum verändern und außerdem ihre mittlere Grauwertintensität im Bereich der Hintergrundintensität liegt. Im Gegensatz dazu enthalten die Hintergrundbereiche, die einem "leeren" Whiteboard entsprechen, beispielsweise keinerlei Kanteninformationen. Die Störregionen, die i.d.R. durch den Körper des Schreibers oder durch Schattenwurf hervorgerufen werden, verändern sich über die Zeit und weisen zudem meistens eine niedrigere Intensität auf als Text- oder Hintergrundbereiche.

Diese Charakteristika der Textregionen im Gegensatz zu Hintergrund- und Störregionen motivieren die Extraktion folgender Merkmale:

1. Anzahl der Kantenpixel: Die Kantenpixel werden durch Anwendung des Sobel-Operators auf die einzelnen Bildblöcke $B^k(u, v)$ bestimmt. Die Sobelfilter zur Detektion von Bildkanten in horizontaler bzw. vertikaler Richtung sind definiert durch:

$$S_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad , \quad S_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad .$$

Durch die Faltung dieser Sobelmasken mit den Bildblöcken $B^k(u, v)$ ergeben sich somit die Bilder $E_x^k(u, v)$ und $E_y^k(u, v)$, die die horizontalen bzw. vertikalen Komponenten der Kanten enthalten:

$$E_x^k(u, v) = B^k(u, v) * S_x(u, v) \quad , \quad E_y^k(u, v) = B^k(u, v) * S_y(u, v) \quad .$$

Die Gradientenstärke berechnet sich daraus durch:

$$|E^k(u, v)| = \sqrt{(E_x^k(u, v))^2 + (E_y^k(u, v))^2} \quad .$$

Als Merkmal wird nun die Anzahl der Kantenpixel verwendet, deren Gradientenstärke einen vorgegebenen Schwellwert $\theta_{|E|}$ übersteigt:

$$\xi_{edge}^k = \sum_{|E^k(u,v)|>\theta_{|E|}} 1 \quad , \tag{6.1}$$

wobei in den vorgenommenen Experimenten der Schwellwert $\theta_{|E|}$ auf 20 gesetzt wurde.

2. Mittlerer Grauwert: Der mittlere Grauwert eines Bildblocks berechnet sich durch:

$$\xi^k_{grey} = \frac{1}{N_u N_v} \sum_{u=1}^{N_u} \sum_{v=1}^{N_v} B^k(u,v) \quad , \tag{6.2}$$

wobei N_x bzw. N_y die horizontale bzw. vertikale Ausdehnung der Bildblöcke bezeichnet.

3. Pixelweise Differenz: Die Bewegungsinformation wird anhand des Differenzbildes zweier aufeinanderfolgender Bilder bestimmt:

$$\xi^k_{diff} = \frac{1}{N_u N_v} \sum_{u=1}^{N_u} \sum_{v=1}^{N_v} |B^k_t(u,v) - B^k_{t-1}(u,v)| \quad . \tag{6.3}$$

Die Entscheidung, ob der betrachtete Bildblock nun einer Text-, Hintergrund- oder Störregion zuzuordnen ist, erfolgt durch Anwendung von Schwellwerten auf die extrahierten Merkmale. Ein Textblock wird damit hypothetisiert, wenn

Text: $(\xi^k_{edge} > \theta_{edge})$ UND $(\xi^k_{grey} > \theta_{grey})$ UND $(\xi^k_{diff} < \theta_{diff})$.

Dementsprechend wird ein Noise-Block detektiert, wenn die folgende Bedingung erfüllt ist:

Noise: $(\xi^k_{grey} \leq \theta_{grey})$ ODER $(\xi^k_{diff} \geq \theta_{diff})$.

Als Hintergrundbereiche werden diejenigen Bildblöcke angenommen, die weder als Text noch als Noise klassifiziert werden.

Die in den obigen Bedingungen verwendeten Schwellwerte wurden experimentell bestimmt. Dabei haben sich folgende Werte als geeignet erwiesen:

$$\theta_{edge} = \min\{N_y, N_x\} \quad , \quad \theta_{grey} = 0.6\mu_{grey} \quad , \quad \theta_{diff} = 5 \quad .$$

Hierbei bezeichnet N_x bzw. N_y die Breite bzw. Höhe eines Bildblocks, in diesem Fall also jeweils 40 Pixel. Der Parameter μ_{grey} stellt den mittleren Grauwert des gesamten Bildes dar. Dieser wird von Zeit zu Zeit an die aktuellen Beleuchtungsverhältnisse angepasst. Dazu wird er neu berechnet, wenn das Bild "ruhig" ist, also keine Störregionen enthält.

In Abbildung 6.3 sind die detektierten Textblöcke sowie die Störbereiche anhand einer Momentaufnahme eines Schreibvorgangs dargestellt.

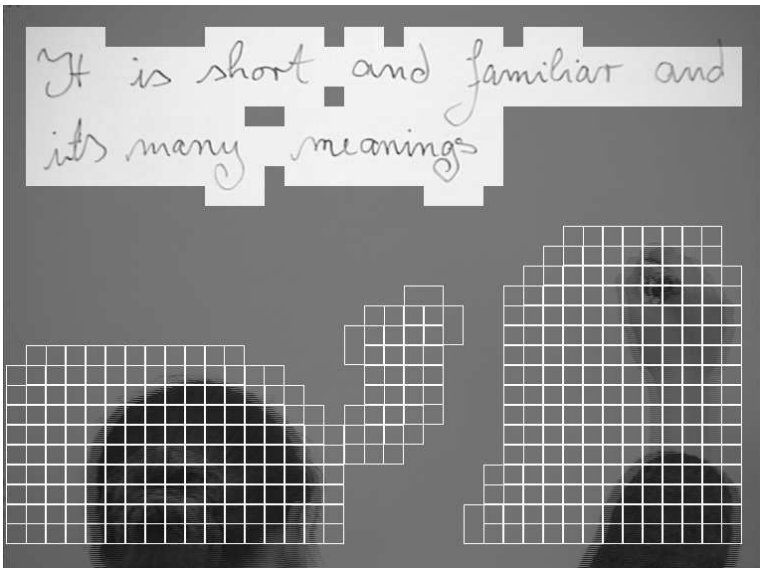


Abbildung 6.3: Darstellung der Textbereiche (hell) und Störbereiche (umrahmt).

6.3.2 Gruppierung der Schriftkomponenten

Nach der im vorherigen Schritt vorgenommenen groben Partitionierung des Bildes in Text-, Hintergrund- und Noise-Blöcke werden nun anhand der Textblöcke Bildregionen ermittelt, die einzelnen Wörtern oder Textzeilen entsprechen. Dazu werden erst die Zusammenhangskomponenten der Schrift, also die einzelnen Komponenten der Tintenspur, extrahiert und anschließend über benachbarte Textblöcke hinweg aggregiert.

Binarisierung der Textblöcke

Zur Extraktion der zusammenhängenden Schriftpixel eines Textblocks ist eine vorherige Binarisierung des Blocks erforderlich. Da innerhalb des 40×40 Pixel großen Blocks nur geringe Schwankungen der Vorder- bzw. Hintergrundintensität auftreten, wird anstelle zeitaufwendiger adaptiver Verfahren die schnelle Otsu-Methode (siehe Seite 29) zur Berechnung des Binarisierungsschwellwerts für den jeweiligen Block eingesetzt.

Extraktion zusammenhängender Schriftkomponenten

Die Bestimmung der Zusammenhangskomponenten (engl. connected components) anhand des binarisierten Bildblocks erfolgt durch das Standard Connected-Components-Labeling, das u.a. in [Har92], Seite 33, beschrieben ist. Das Verfahren, das die Zusammenhangskomponenten mit numerischen Markierungen versieht, lässt sich wie folgt skizzieren ([Kle92], Seite 243):

Im ersten Durchlauf über das Bild (von links oben nach rechts unten) werden die Schriftpixel markiert. Dabei wird geprüft, ob mindestens einer der vier bereits bearbei-

teten Nachbarpixel markiert ist. Ist dies nicht der Fall, so wird dem betrachteten Pixel die niedrigste noch nicht vergebene Marke zugewiesen. Sind dagegen ein oder mehrere Nachbarpixel bereits mit Marken versehen, dann wird das betrachtete Pixel mit der kleinsten Markierung der Nachbarpixel gekennzeichnet. Gleichzeitig wird in einer Äquivalenztabelle festgehalten, dass die Markierungen der benachbarten Pixel der selben Zusammenhangskomponente zugeordnet sind. Dies ist erforderlich, da diejenigen Bildsegmente, die nach unten und/oder links gerichtete Konkavitäten aufweisen, im ersten Durchlauf mehr als eine Markierung zugewiesen bekommen.

Die mehrfachen Markierungen werden im zweiten Durchlauf anhand der Äquivalenztabelle aufgelöst, sodass schließlich jede Zusammenhangskomponente der Schrift genau eine Markierung aufweist. Die Äquivalenz*klassen* werden dabei auf Basis der transitiven Hülle der Menge der Äquivalenzen unter Berücksichtigung der Reflexivität, Symmetrie und Transitivität bestimmt.

Gruppierung der Schriftkomponenten

Nachdem die Zusammenhangskomponenten der Tintenspur extrahiert wurden, werden diejenigen Komponenten, die ein Wort bzw. eine Textzeile bilden, zusammengefasst. Das Gruppierungskriterium basiert dabei auf dem Abstand der Bounding Boxen der Schriftkomponenten. Damit werden Komponenten zusammengefasst, wenn ihr gegenseitiger horizontaler bzw. vertikaler Abstand kleiner als ein vorgegebener Schwellwert ist. Der Bildausschnitt, der sich aus der Bounding Box der gruppierten Schriftkomponenten ergibt, wird dann für den Erkennungsprozess verwendet.

Mit den Bezeichnungen x^i_{ul}, y^i_{ul} und x^i_{lr}, y^i_{lr} für die Koordinaten der oberen linken (upper left) bzw. unteren rechten (lower right) Ecke der Bounding Box der i-ten Komponente ergibt sich der horizontale bzw. vertikale Abstand zwischen zwei Komponenten zu:

$$\begin{aligned} d_x &= \max(x^i_{ul}, x^j_{ul}) - \min(x^i_{lr}, x^j_{lr}) \\ d_y &= \max(y^i_{ul}, y^j_{ul}) - \min(y^i_{lr}, y^j_{lr}) \quad . \end{aligned}$$

Die Gruppierung der Komponenten erfolgt somit, wenn der horizontale Abstand kleiner als der Schwellwert θ_x ist und außerdem die vertikale Überlappung mindestens ein Fünftel der Ausdehnung der kleineren Komponente beträgt:

$$(d_x < \theta_x) \quad \text{UND} \quad (d_y < -0.2 \min((y^i_{lr} - y^i_{ul}), (y^j_{lr} - y^j_{ul}))) \quad .$$

Für den horizontalen Abstandsschwellwert θ_x können unterschiedliche Werte vorgegeben werden, je nachdem, ob Bildregionen einzelner Wörter oder ganzer Zeilen extrahiert werden sollen. Im Falle der Wortextraktion hat sich für θ_x ein Wert von 20 Pixel als geeignet erwiesen, sollen dagegen Zeilen extrahiert werden, so muss der Schwellwert auf mindestens 80 Pixel gesetzt werden, um die Schriftkomponenten auch über größere Wortzwischenräume hinweg zusammenfassen zu können.

6.4 Regionengedächtnis

Die Detektion der Textregionen erfolgt im Bildtakt, also anhand jedes aufgenommenen Bildes. Da ein einmal geschriebener Textabschnitt jedoch so lange am Whiteboard verbleibt bis er vom Schreiber weggewischt wird, ist es ausreichend, wenn der Erkennungsprozess für eine extrahierte Textregion nur einmal gestartet wird. Um zu verhindern, dass der Erkennungsprozess auch zu allen späteren Zeitpunkten erneut durchgeführt wird, werden die extrahierten Textregionen, sobald sie das erste Mal im Bild auftauchen, in einem Regionengedächtnis gespeichert. Eine Textregion wird somit nur dann klassifiziert, wenn sie bisher noch nicht im Regionengedächtnis vermerkt ist.

Um zu entscheiden, ob eine extrahierte Region bereits im Regionengedächtnis eingetragen wurde, wird die betreffende Region paarweise mit allen gespeicherten Regionen verglichen. Falls sich keine Überschneidung mit einer gespeicherten Region ergibt, so handelt es sich bei der extrahierten Region um einen noch nicht klassifizierten Schriftabschnitt. Somit wird der Erkennungsprozess gestartet und die Region im Gedächtnis eingetragen. Überschneidet sich jedoch die betrachtete Region $R_{act}(x, y)$ mit einer gespeicherten Region $R_{mem}(x, y)$, so basiert die Entscheidung auf der Differenz der (blockweise) binarisierten Regionen. Diese Differenz wird anhand der Exklusiv-Oder (XODER) Verknüpfung bestimmt:

$$d(x, y) = R_{act}(x, y) \text{ XODER } R_{mem}(x, y) \quad . \tag{6.4}$$

Liegt die Anzahl der gesetzten Pixel im Bild $d(x, y)$ – normiert auf die Größe des von beiden Regionen aufgespannten rechteckigen Bereichs – unter einem vorgegebenen Schwellwert,

$$\frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} d(x, y) < \theta_{mem} \, , \qquad \text{hier: } \theta_{mem} = 0.03 \, , \tag{6.5}$$

so wird angenommen, dass die betrachtete Region der im Gedächtnis gespeicherten Region entspricht. Anderenfalls wird diese Region als noch nicht klassifiziert gekennzeichnet und der Erkennungsprozess angestoßen.

Mit Hilfe des Regionengedächtnisses wird außerdem die Erkennung von Korrekturen ermöglicht. Wenn der Schreiber also beispielsweise ein Wort oder Teile eines Wortes, das bereits klassifiziert wurde, wegwischt und neu schreibt, so wird für die aktualisierte Textregion die Erkennung erneut durchgeführt.

Dies wird ebenfalls dadurch erreicht, dass zu jedem Zeitschritt die im Gedächtnis gespeicherten Regionen mit den aktuell extrahierten Regionen abgeglichen werden (6.4-6.5). Lässt sich dabei ein gespeicherter Bildausschnitt nicht an der gleichen Position im aktuellen Bild wiederfinden, und befindet sich außerdem keine Störregion innerhalb dieses Bildbereichs, so wird angenommen, dass der entsprechende Textabschnitt weggewischt oder modifiziert wurde. In diesem Fall wird die gespeicherte Region aus dem Gedächtnis entfernt. Wurde der Textabschnitt vom Schreiber nicht weggewischt sondern modifiziert, so wird dann im folgenden Zeitschritt der Erkennungsprozess für diesen Textabschnitt erneut gestartet.

6.5 Vorverarbeitung

Nachdem die zu klassifizierenden Textregionen aus der Bildfolge extrahiert wurden, werden einige Vorverarbeitungsschritte durchgeführt, um die schreiber- und situationsbedingte Variabilität der Schrift zu kompensieren. Im Vergleich zur Verarbeitung eingescannter Dokumente kommt diesen Normalisierungsmaßnahmen bei der Erkennung von Tafelanschrift eine größere Bedeutung zu, da die Schrift häufig stärker verzerrt ist. Dies liegt vor allem daran, dass jegliche Referenzlinien fehlen und das Schreiben an einer Tafel für viele Personen oftmals ungewohnt ist. Durch die videobasierte Aufnahme verschärft sich diese Problematik auf Grund schwankender Beleuchtungsbedingungen (z.B. hervorgerufen durch Schattenwurf) und der geringeren Auflösung zusätzlich.

6.5.1 Adaptive Binarisierung

Der Ausgangspunkt für die Vorverarbeitung ist die als Graustufenbild vorliegende Textregion. Da die weiteren Verarbeitungsschritte auf Binärbildern basieren, erfolgt im ersten Schritt eine Binarisierung der Textregion.

Hier stellt sich die Frage, warum erneut eine Trennung des Vordergrunds (Tintenspur) vom Hintergrund vorgenommen wird, wo doch schon zur Detektion der Textregionen die entsprechenden Textblöcke binarisiert wurden. Die Antwort ist, dass die blockweise Binarisierung, die zur Einsparung von Rechenzeit mit einem festen Schwellwert pro Block durchgeführt wird, in inhomogen beleuchteten Bereichen Teile der Tintenspur häufig fälschlicherweise dem Hintergrund zuschlägt (siehe Abbildung 6.4(a) und 6.4(b)). Diese Fehler sind bei der Detektion der Textregionen unkritisch, solange ein gesamter Textblock nicht vollständig als Hintergrund angenommen wird. Für die Texterkennung hätten fehlende Bereiche der Tintenspur jedoch fatale Auswirkungen in Form falsch erkannter Buchstaben, sodass die Textregion deshalb mit einem lokalen Verfahren binarisiert wird.

Zur lokalen Binarisierung wird hier die modifizierte Niblack-Methode eingesetzt, mit der für jeden Pixel der optimale Schwellwert separat in einem Bildfenster berechnet wird, das um den betrachteten Bildpunkt zentriert ist und pixelweise über das Bild geschoben wird. Dieses Verfahren ist bereits auf Seite 31 behandelt worden, zur besseren Übersicht wird die Berechnungsvorschrift (Gleichung 3.10) für den lokalen Schwellwert $t(x, y)$ hier wiederholt:

$$t(x,y) = \mu(x,y) + k\left(\mu(x,y)\left(1 - \frac{\sigma(x,y)}{R}\right)\right)$$

In obiger Formel bezeichnet $\mu(x, y)$ den mittleren Grauwert des Fensters, $\sigma(x, y)$ die Standardabweichung und R den Dynamikbereich der Grauwerte. Der Parameter k ist vom Anwendungsfall abhängig und bei der Schrifterkennung als kleine negative Zahl zu wählen. Aus den Experimenten hat sich für k der Wert -0.06 und für den Dynamikbereich R der Wert 100 als geeignet herausgestellt. Die Fenstergröße beträgt

(a) (b) (c)

Abbildung 6.4: Vergleich der (blockweisen) Otsu-Binarisierung (b) mit der adaptiven (modifizierten) Niblack-Binarisierung (c). Der Pfeil markiert Bereiche der Tintenspur, die bei der blockweisen Otsu-Binarisierung dem Hintergrund zugeschlagen wurden, wohingegen mit der modifizierten Niblack-Methode die Tintenspur korrekt segmentiert wurde.

51×51 Pixel. Das mit dieser Parametrisierung binarisierte Ausgangsbild ist in Abbildung 6.4(c) dargestellt.

6.5.2 Ermittlung von Referenzlinien

Die Bestimmung der Referenzlinien der Schrift ist eine unabdingbare Voraussetzung sowohl für die Schriftnormalisierung als auch für die spätere Merkmalsextraktion. Da sich jedoch der vertikale Versatz und die Orientierung einzelner Wörter innerhalb einer Textzeile oftmals deutlich unterscheiden (siehe Abbildung 6.5(a)), sind bei der Verarbeitung ganzer Textzeilen adaptive Verfahren zur Referenzlinienschätzung erforderlich.

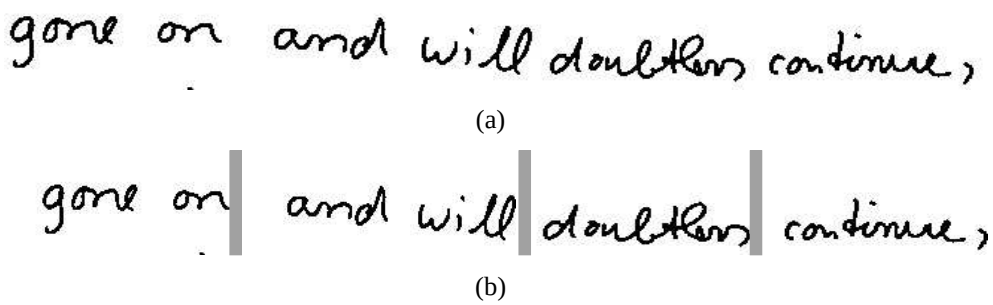
Um eine *robuste* Bestimmung von Basis- und Mittellinie der Schrift durchführen zu können, werden die folgenden vereinfachenden Annahmen zugrundegelegt:

1. Die Korpushöhe der Schrift ist in einer Textzeile annähernd konstant, sodass von einer parallel zur Basislinie verlaufenden Mittellinie ausgegangen werden kann.

2. Veränderungen in Position und Orientierung der Schrift und damit der Referenzlinien treten nur an Wortgrenzen bzw. bei längeren Lücken innerhalb der Textzeile auf.

3. Innerhalb eines zusammenhängenden Abschnitts lässt sich die Basislinie durch eine Gerade approximieren.

Da die Basislinie im Vergleich zu den übrigen Referenzlinien am sichersten geschätzt werden kann, basiert die Schriftnormalisierung, insbesondere die Korrektur der Schriftorientierung und des vertikalen Versatzes, daher ausschließlich auf der lokalen Basislinie eines Textabschnitts. Die Schätzung der Mittellinie ist dagegen nur für die Merkmalsextraktion erforderlich. Somit ist es ausreichend, wenn die Mittellinie erst nach Durchführung aller Vorverarbeitungsschritte bestimmt wird.



(a)

(b)

Abbildung 6.5: (a) Unterschiede in Orientierung und vertikaler Position einzelner Wörter innerhalb einer Textzeile. (b) Aufteilung der Textzeile in einzelne Abschnitte.

Extraktion von Textabschnitten

Zur adaptiven Schätzung der Basislinie wird die Textregion anhand der Zwischenräume im Schriftbild in Abschnitte aufgeteilt, anhand derer die lokale Basislinie durch eine Gerade approximiert wird. Als Zwischenräume werden dabei diejenigen Bereiche der Textregion hypothetisiert, die mindestens zehn aufeinanderfolgende Bildspalten umfassen, die ausschließlich Hintergrundpixel enthalten. Bei dieser Vorgehensweise sind allerdings kurze Schriftabschnitte, die beispielsweise aus einzelnen Buchstaben bestehen, zu vermeiden, da diese eine verläßliche Schätzung der Basislinie nicht ermöglichen. Aus diesen Grund wird daher außerdem verlangt, dass ein Schriftabschnitt eine vorgegebene Mindestlänge, in diesem Fall 100 Pixel, aufweist. Ein Beispiel für die Aufteilung einer Textzeile in Abschnitte ist in Abbildung 6.5(b) veranschaulicht.

Schätzung lokaler Basislinien

Um eine möglichst robuste und genaue Bestimmung der lokalen Basislinie zu erreichen, wird die Approximation in einem dreistufigen Prozess vorgenommen, durch den die Genauigkeit der Approximation sukzessive verbessert wird:

1. Im ersten Schritt wird die grobe Position der noch als horizontal angenommenen Basislinie ermittelt. Dazu wird das horizontale Projektionshistogramm des Schriftabschnitts berechnet. Um unabhängig von der Strichdicke zu sein, wird hier anstatt der relativen Häufigkeit der Schriftpixel die relative Häufigkeit der horizontalen schwarz-weiß (Schrift-Hintergrund) Übergänge im Histogramm aufgetragen. Die Detektion des charakteristischen Plateaus im Histogramm, das kennzeichnend für den Mittelbereich der Schrift ist, erfolgt durch Anwendung eines Schwellwerts, der automatisch mit Hilfe der Otsu-Methode anhand der Verteilung der Schriftpixel pro Bildzeile bestimmt wird. Die untere Begrenzung des Mittelbereichs wird dann als grobe Schätzung für die y-Koordinate der Basislinie der Schrift verwendet.

2. Die im ersten Schritt geschätzte horizontale Basislinie wird im zweiten Schritt dazu verwendet, die für die Geradenapproximation relevanten Konturminima von den “Ausreißer-Minima”, die beispielsweise den Unterlängen zuzuordnen sind, zu unterscheiden. Dies wird dadurch erreicht, dass nur diejenigen Konturminima betrachtet werden, die sich in einem “Korridor” entlang der geschätzten Basislinie befinden. Der Radius des Korridors entspricht dabei der Breite des Plateaus im Projektionshistogramm. Anhand der Konturminima, die innerhalb des Korridors liegen, wird durch lineare Regression eine verbesserte Schätzung der Basislinie vorgenommen (siehe Gleichungen 3.14-3.16).

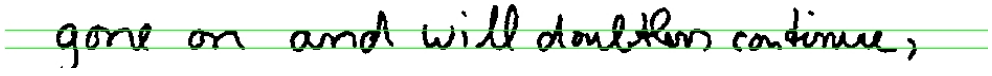
3. Im dritten und letzten Schritt wird die Lage der Ausgleichsgeraden erneut korrigiert, indem weitere “Ausreißer-Minima” von der Berechnung ausgeschlossen werden. In diesem Fall sind die Punkte betroffen, deren Abstand mindestens doppelt so groß ist wie der mittlere Punktabstand zur geschätzten Basislinie. Anhand der resultierenden Stützpunkte erfolgt die abschließende Berechnung der Ausgleichsgeraden.

Man erhält somit für jeden Textabschnitt eine lokale Schätzung der Basislinie (siehe Abbildung 6.6(a)), anhand derer die Normalisierungsschritte vorgenommen werden können. Nachdem diese durchgeführt wurden, werden die einzelnen Abschnitte wieder zusammengefügt, sodass die nun horizontalen Basislinien untereinander keinen Versatz mehr aufweisen. Nach der Skalierung der gesamten Textzeile wird zum Abschluss der Vorverarbeitung die Mittellinie und damit die Korpushöhe der Schrift geschätzt, sodass Merkmale, die invariant gegenüber der Schriftgröße sind, extrahiert werden können.

Schätzung der (globalen) Mittellinie

Die Bestimmung der für die Berechnung der Korpushöhe notwendigen Mittellinie der Schrift erfolgt anhand der lokalen Maxima der Schriftkontur. Da auf Grund der geringeren Anzahl von Stützpunkten eine robuste lokale Schätzung der Mittellinie im

(a)



(b)

Abbildung 6.6: (a) Lokale Basislinien. (b) Resultierende Textzeile mit Basis- und Mittellinie nach Abschluss der Vorverarbeitung.

Gegensatz zur Basislinie nicht vorgenommen werden kann, wird stattdessen im Anschluss an die Schriftnormalisierung eine globale, horizontale Mittellinie für die gesamte Textzeile berechnet. Dazu werden die y-Koordinaten der lokalen Maxima der Schriftkontur mit Hilfe der Otsu-Methode in zwei Gruppen geclustert, wobei für jeden Cluster anschließend der Mittelwert berechnet wird. Der größere der beiden Mittelwerte wird dann als y-Koordinate der Mittellinie angenommen (Der Ursprung des Koordinatensystem liegt links oben im Bild). In Abbildung 6.6(b) ist die geschätzte Mittellinie der Textzeile eingezeichnet.

6.5.3 Korrektur der Orientierung und des Versatzes

Die Korrektur der Orientierung und des vertikalen Versatzes der Schrift erfolgt lokal auf der Grundlage der geschätzten Basislinien der in Abschnitte aufgeteilten Textzeile. Wie oben erläutert wurde, werden die Basislinien der Schriftabschnitte jeweils durch eine Ausgleichsgerade approximiert, die wie folgt dargestellt werden kann:

$$y(x) = a + bx \quad .$$

Dabei bezeichnet der Parameter a den y-Achsenabschnitt und damit die vertikale Position der Schrift, während der Parameter b die Steigung der Geraden und somit die Orientierung der Schrift darstellt. In Abbildung 6.7 ist an einem Beispiel die Ausgleichsgerade, die die Minima der Schriftkontur approximiert, dargestellt.

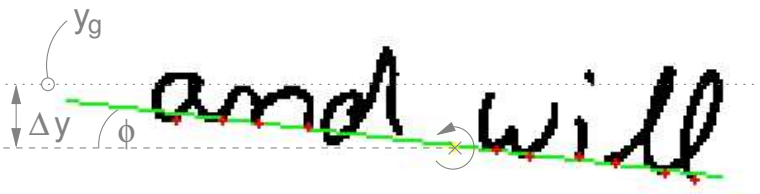


Abbildung 6.7: Konturminina (rot) und Ausgleichsgerade (grün). Außerdem dargestellt: Rotationswinkel ϕ und vertikaler Versatz Δy zur globalen Basislinie y_g.

Der erste Korrekturschritt besteht darin, eine Rotation des Schriftabschnitts vorzunehmen, sodass anschließend die Basislinie horizontal ausgerichtet ist. Der Rotationswinkel ϕ ist dabei gegeben durch

$$\phi = \arctan b \quad .$$

Als Drehpunkt dient der Mittelwertvektor $(\bar{x}, \bar{y})$ der Konturminima, sodass nach Durchführung der Rotation die Basislinie der Gleichung $y(x) = \bar{y}$ genügt. Die Berechnungsvorschrift für die Rotation lautet demnach:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix} + \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \tag{6.6}$$

Im zweiten Schritt wird eine Translation der Schrift in vertikaler Richtung durchgeführt, um die Basislinie auf eine vorgegebene Bildhöhe zu verschieben, sodass ein etwaiger vertikaler Versatz zwischen den einzelnen Schriftabschnitten kompensiert wird. Als Zielwert wird dabei die y-Koordinate der globalen Basislinie y_g verwendet, die im Vorhinein anhand des Projektionshistogramms der gesamten Textzeile ermittelt wurde. Die neuen Bildkoordinaten ergeben sich also zu:

$$\begin{pmatrix} x'' \\ y'' \end{pmatrix} = \begin{pmatrix} x' \\ y' + \Delta y \end{pmatrix}, \quad \text{mit } \Delta y = y_g - \bar{y}. \tag{6.7}$$

6.5.4 Korrektur der Neigung

Die Korrektur der Schriftneigung erfolgt ebenfalls lokal auf Basis der einzelnen Schriftabschnitte der Textzeile, wobei die Schriftneigung innerhalb eines Abschnittes als konstant angenommen wird. Das eingesetzte Verfahren basiert auf dem Histogramm der Gradientenrichtungen des Kantenbildes und ist angelehnt an die in [Sen98] beschriebene Methode.

Da vor allem die eher vertikalen Segmente der Tintenspur den Neigungswinkel der Schrift definieren, werden diese Bereiche im ersten Verarbeitungsschritt anhand des binarisierten Ausgangsbildes extrahiert. Dies erfolgt durch die zeilenweise Anwendung eines Filters, der direkt aufeinanderfolgende Schriftpixel, die also ein horizontales Schriftsegment beschreiben, bis auf den jeweils ersten Pixel unterdrückt.

Im zweiten Schritt wird ein Kantenfilter auf das so gefilterte Bild angewandt, um die Gradienteninformationen der Kantenpixel zu erhalten (siehe Abbildung 6.8(b)). Hier wird dafür der Canny-Kantendetektor eingesetzt, vor allem auf Grund seiner guten Lokalisationseigenschaften von verrauschten Kanten und der Richtungsisotropie des Gradienten[2]. Der Canny-Operator kombiniert die Gaußglättung mit der Differentiation des Bildes. Der resultierende Gradientenbetrag $m(x, y)$ ist dabei gegeben durch

$$m(x, y) = \|\nabla(G_\sigma * I)\| = \sqrt{\left(\frac{\partial(G_\sigma * I)}{\partial x}\right)^2 + \left(\frac{\partial(G_\sigma * I)}{\partial y}\right)^2},$$

wobei der Term $G_\sigma * I$ die Gaußfilterung (mit Standardabweichung σ) des Ausgangsbildes bezeichnet. Die Gradientenrichtung ergibt sich entsprechend zu:

$$\vartheta(x, y) = \arctan \frac{\left(\frac{\partial(G_\sigma * I)}{\partial y}\right)}{\left(\frac{\partial(G_\sigma * I)}{\partial x}\right)} \quad .$$

Im Anschluss an die Glättung und Differentiation des Bildes sieht der Canny-Kantendetektor die Ausdünnung der Kanten vor. Dies geschieht mit Hilfe der *Non-Maximum-Suppression* Methode, bei der ein Kantenpixel nur gesetzt wird, wenn sein

[2] Hier wurde die frei erhältliche Implementation des Canny-Operators der Robot Vision Group, Dept. of Artificial Intelligence, University of Edinburgh, eingesetzt.

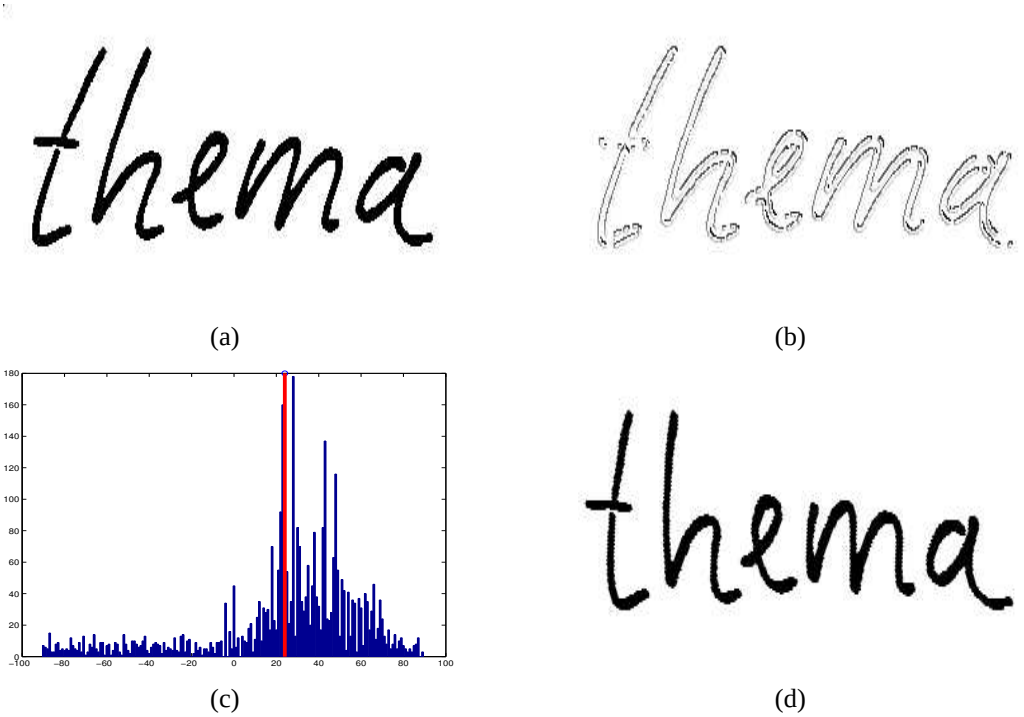


Abbildung 6.8: (a) Ausgangsbild, (b) Gradientenrichtungsbild, (c) Histogramm der Gradientenrichtungen und Mittelwert und (d) Resultat der Neigungskorrektur

Gradientenbetrag ein lokales Maximum in Gradientenrichtung darstellt:

$$m(x,y) \geq m(x+\delta x, y+\delta y) \quad \text{UND} \quad m(x,y) \geq m(x-\delta x, y-\delta y),$$

mit dem Einheitsvektor in Gradientenrichtung

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = \frac{\nabla(G_\sigma * I)}{\|\nabla(G_\sigma * I)\|}.$$

Der letzte Schritt des Canny-Kantendetektors besteht in der Unterdrückung nicht signifikanter Kantenpixel. Anstelle eines einfachen Schwellwerts wird beim Canny-Operator ein Hysterese-Verfahren verwendet, das auf zwei Schwellwerten, θ_L und θ_H, basiert. Dabei wird ein Kantenpixel gesetzt, wenn seine Gradientenstärke $m(x,y)$ größer/gleich θ_H ist. Entsprechend wird ein Kantenpixel unterdrückt, wenn die Gradientenstärke kleiner als θ_L ist. Diejenigen Kantenpixel, deren Gradientenstärke zwischen θ_H und θ_L liegt, werden nur dann gesetzt, wenn sie jeweils über einen Pfad mit einem Pixel verbunden sind, dessen Gradientenstärke über θ_H liegt und außerdem die Gradientenstärken aller Pixel auf dem Pfad größer/gleich θ_L sind.

Das Verhalten des Canny-Operators kann also mit drei Parametern beeinflusst werden: Der Standardabweichung σ der Gaußmaske und den Hystereseschwellwerten θ_L

und θ_H. Hier wurde die folgende Parametrisierung verwendet: $\sigma = 1.0$, $\theta_L = 1$ und $\theta_H = 255$.

Nach der Kantendetektion mit Hilfe des Canny-Operators werden die Gradientenrichtungen signifikanter Kantenpixel in einem Histogramm akkumuliert (siehe Abbildung 6.8(c)). Der Mittelwert der durch das Histogramm approximierten Verteilung wird dann als Neigungswinkel α der Schrift angenommen, der anschließend durch eine Scherung des Bildes korrigiert wird (Abbildung 6.8(d)):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x - y \tan \alpha \\ y \end{pmatrix}. \tag{6.8}$$

6.5.5 Korrektur der Größe

Im Anschluss an die Normalisierung der Neigung der Schrift erfolgt die Korrektur der Schriftgröße. Die Schätzung der Schriftgröße basiert auf der Annahme, dass die Anzahl der lokalen Konturminima der Schrift näherungsweise proportional zur Anzahl der Buchstaben ist [Mad99b]. Der durchschnittliche Abstand zwischen benachbarten lokalen Konturminima, der sich aus dem Verhältnis der Breite der Bounding Box des Schriftabschnitts zur Anzahl der Konturminima ergibt, kann somit als Richtwert für die mittlere Buchstabenbreite interpretiert werden. Diese mittlere Buchstabenbreite wird nun durch eine Skalierung des Bildes auf einen vorgegebenen Wert gebracht.

Im Gegensatz zu den vorherigen Normalisierungsschritten wird die Schätzung der Schriftgröße allerdings global anhand der gesamten Textzeile durchgeführt. Die lokale Korrektur wird vermieden, da kurze Schriftabschnitte nur wenige Konturminima aufweisen, sodass eine verlässliche Schätzung der Schriftgröße kaum möglich ist. Eine verbesserte Robustheit des Verfahrens wird außerdem dadurch erreicht, dass anstatt der Anzahl der Konturminima der Mittelwert aus der Anzahl der Konturminima und -maxima für die Berechnung des Skalierungsfaktors verwendet wird.

Mit den Bezeichnungen C für die vorgegebene Buchstabenbreite, W für die Breite der Bounding Box des Schriftabschnitts und $N_{\min}$ und $N_{\max}$ für die Anzahl der Konturminima bzw. -maxima ergibt sich der Skalierungsfaktor demnach zu:

$$s = C \, \frac{N_{\min} + N_{\max}}{2W} \quad .$$

Mit Hilfe des Faktors s wird das Bild sowohl in horizontaler als auch in vertikaler Richtung skaliert. Die vorgegebene Buchstabenbreite C ist hier auf 25 Pixel festgelegt worden.

6.6 Segmentierung

Nach Durchführung der Vorverarbeitung liegt das binarisierte und bezüglich der Schriftvariabilität normalisierte Bild der Textregion vor. Da zur Schrifterkennung eine analytische Strategie auf Basis von HMMs eingesetzt wird, muss die Textregion

vor der Klassifikation in Untereinheiten segmentiert werden. Anhand der Sequenz von Untereinheiten wird dann eine Merkmalsvektorfolge extrahiert, die mittels der HMMs klassifiziert wird. Um frühe unumkehrbare Segmentierungsentscheidungen zu vermeiden, wird eine systematische Unterteilung der Textregion durchgeführt, sodass die Segmentierung der Schrift in Buchstaben *implizit* im Zuge der Klassifikation erfolgt.

Zur systematischen Unterteilung der Textregion wird eine Sliding-Window Methode verwendet, bei der ein Fenster von links nach rechts über die Textzeile geschoben wird. Die Breite des Fensters beträgt hier vier Pixel, die Höhe entspricht der der Textregion. Die Verschiebung des Fensters erfolgt mit einem Überlapp von zwei Pixeln.

6.7 Merkmalsextraktion

Anhand des jeweiligen Bildausschnitts, welcher durch das sliding window definiert wird und hier mit $f_s(i, j)$ bezeichnet wird, erfolgt in diesem Schritt die Extraktion von Merkmalen. Die resultierende Merkmalsrepräsentation sollte dabei möglichst robust, kompakt und diskriminativ sein, um bestmögliche Erkennungsergebnisse zu erhalten.

Die Entscheidung, welche Art von Merkmalen für diese Erkennungsaufgabe geeignet ist, wurde auf der Grundlage der in [Mar00a] beschriebenen Untersuchungen vorgenommen. Dort sind die Erkennungsleistungen von Systemen gegenübergestellt worden, die einerseits auf geometrischen Merkmalen bzw. andererseits auf einfachen Zoning-Merkmalen basieren. Dabei konnte auch durch eine Hauptkomponentenanalyse der Zoning-Merkmale nicht die Performanz der geometrischen Merkmale erreicht werden, sodass auch hier die gewählte Merkmalsrepräsentation geometrischer Natur ist. Insgesamt umfasst der Merkmalsvektor achtzehn Komponenten, aufgeteilt in neun Basismerkmale (siehe Abbildung 6.9) und neun Delta-Merkmale, die jeweils die Ableitung des Basismerkmals darstellen. Die Berechnungsvorschriften für die einzelnen Merkmale sind im folgenden detailliert beschrieben.

Position der Schriftkontur

Wie in Abschnitt 2.2.3 erläutert wurde, ist die Schriftkontur ein wichtiges Merkmal zur visuellen Schriftperzeption beim Menschen. Daher wird hier innerhalb des sliding windows der mittlere Abstand der oberen sowie der unteren Schriftkontur zur Basislinie gemessen und als Merkmal verwendet (siehe Abbildung 6.9(a), 6.9(b)). Die Abstände werden dabei auf die Korpushöhe der Schrift normiert, um eine von der Schriftgröße unabhängige Repräsentation zu erreichen. Neben der oberen und unteren Schriftkontur wird außerdem der mittlere Abstand des Schwerpunktes zur Basislinie, normiert auf die Korpushöhe, als weiteres Merkmal verwendet (Abbildung 6.9(c)).

Mit den Bezeichnungen $y_i^u,\ \ y_i^l,\ \ y_i^{cog},\ \ (i = 1, \ldots, 4)$ für die y-Koordinaten der oberen und unteren Schriftkontur, bzw. des Schwerpunktes innerhalb des sliding windows f_s, der Position der Basislinie y_B und der Korpushöhe c_H ergeben sich für die

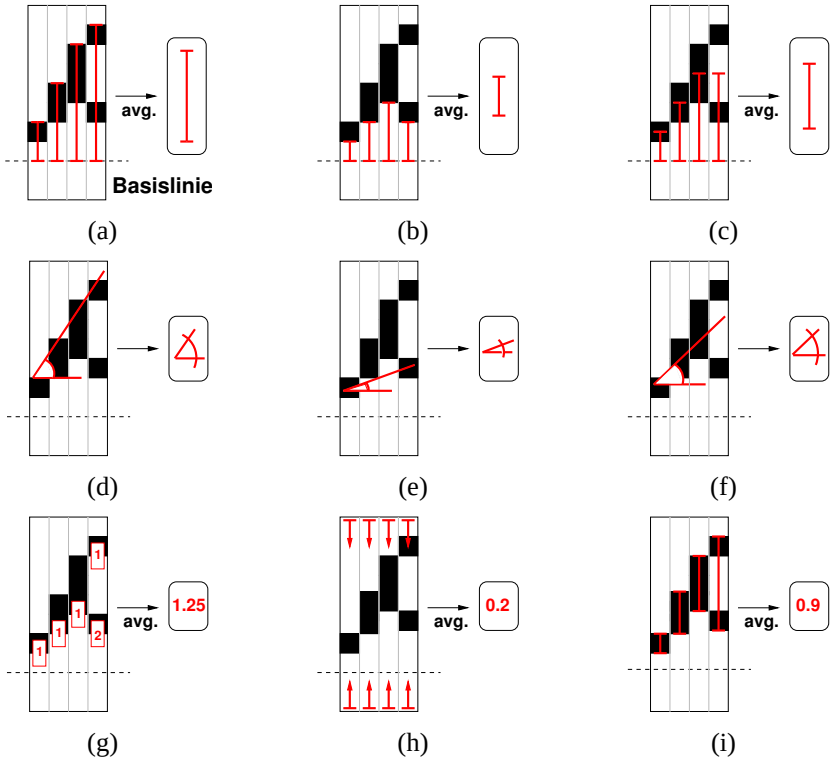


Abbildung 6.9: Extraktion geometrischer Merkmale innerhalb des sliding window. (a) Mittlerer Abstand der oberen Schriftkontur zur Basislinie. (b) Mittlerer Abstand der unteren Schriftkontur. (c) Mittlerer Abstand der y-Koordinaten der Spaltenschwerpunkte. (d) Orientierung der oberen Kontur. (e) Orientierung der unteren Kontur (f) Orientierung des Verlaufs der Spaltenschwerpunkte. (g) Mittlere Anzahl vertikaler Schrift-Hintergrund Übergänge (h) Mittlere Anzahl der Schriftpixel pro Bildspalte. (i) Mittlere Anzahl der Schriftpixel zwischen oberer und unterer Schriftkontur.

Positionsmerkmale die folgenden Berechnungsvorschriften:

$$p_s^u = \frac{y_B - \frac{1}{4}\sum_{i=1}^{4} y_i^u}{c_H} \tag{6.9}$$

$$p_s^l = \frac{y_B - \frac{1}{4}\sum_{i=1}^{4} y_i^l}{c_H} \tag{6.10}$$

$$p_s^{cog} = \frac{y_B - \frac{1}{4}\sum_{i=1}^{4} y_i^{cog}}{c_H} \quad . \tag{6.11}$$

Orientierung der Schriftkontur

Neben dem Abstand der Schriftkontur zur Basislinie wird außerdem die Orientierung der Kontur als Merkmal verwendet (Abbildung 6.9(d), 6.9(e)). Als Schätzwert für die

Orientierung wird der Steigungswinkel der Approximationsgeraden verwendet, die durch lineare Regression an die Kontur angenähert wurde. Auch hier wird neben der oberen und unteren Kontur der Verlauf des Schwerpunkts innerhalb des sliding windows berücksichtigt (Abbildung 6.9(f)). Damit werden die folgenden drei Merkmale extrahiert:

$$\alpha_s^u = \arctan(m^u), \quad m^u = \frac{\sum_{i=1}^{4}(x_i^u - \bar{x}^u)(y_i^u - \bar{y}^u)}{\sum_{i=1}^{n}(x_i^u - \bar{x}^u)^2} \tag{6.12}$$

$$\alpha_s^l = \arctan(m^l), \quad m^l = \frac{\sum_{i=1}^{4}(x_i^l - \bar{x}^l)(y_i^l - \bar{y}^l)}{\sum_{i=1}^{n}(x_i^l - \bar{x}^l)^2} \tag{6.13}$$

$$\alpha_s^{cog} = \arctan(m^{cog}), \quad m^{cog} = \frac{\sum_{i=1}^{4}(x_i^{cog} - \bar{x}^{cog})(y_i^{cog} - \bar{y}^{cog})}{\sum_{i=1}^{n}(x_i^{cog} - \bar{x}^{cog})^2} \tag{6.14}$$

Vertikale Schrift-Hintergrund Transitionen

Als weiteres Merkmal zur Beschreibung der Schriftgeometrie wird ähnlich wie in [Mar00a] die Anzahl der vertikalen Schrift-Hintergrund Transitionen innerhalb des sliding windows verwendet. Die Gesamtzahl wird dabei durch die Breite des sliding windows dividiert, sodass hier die mittlere Anzahl der vertikalen Schrift-Hintergrund Übergänge pro Bildspalte betrachtet wird (Abbildung 6.9(g)).

$$n_s^{tr} = \frac{1}{4}\sum_{i=1}^{4}\sum_{j=1}^{H-1} tr(i,j) \tag{6.15}$$

mit

$$tr(i,j) = \begin{cases} 1\,, & \text{wenn } f(i,j) = 1 \wedge f(i,j+1) = 0 \\ 0\,, & \text{sonst} \end{cases} .$$

Hierbei bezeichnet $f(i,j)$ die binarisierte Bildinformation und H die Höhe des sliding windows.

Relation Schriftpixel / Gesamtpixel

Um die "Schwärzung" des jeweiligen Bildausschnitts zu berücksichtigen, wird zusätzlich zu den obigen Merkmalen das Verhältnis der Anzahl der Schriftpixel zur Gesamtzahl der Pixel innerhalb des sliding windows betrachtet (Abbildung 6.9(h)).

$$rel_s^{(1)} = \frac{1}{4H}\sum_{i=1}^{4}\sum_{j=1}^{H}[f(i,j) = 1] \quad . \tag{6.16}$$

Darüberhinaus wird auch die Relation der Anzahl der Schriftpixel zur Zahl der Pixel zwischen oberer und unterer Schriftkontur gebildet (Abbildung 6.9(i)):

$$rel_s^{(2)} = \frac{1}{4}\sum_{i=1}^{4}\left(\frac{1}{y_i^l - y_i^u}\sum_{j=y_i^u}^{y_i^l}[f(i,j)=1]\right) \quad . \tag{6.17}$$

Delta-Merkmale

Zur Berücksichtigung eines größeren räumlichen Kontextes werden analog zum online System Delta-Merkmale berechnet, die jeweils eine Approximation der Ableitung der einzelnen Merkmale 6.9-6.17 darstellen:

$$\Delta p_s^u = \frac{1}{2}\left(p_{s+1}^u - p_{s-1}^u\right) \tag{6.18}$$

$$\Delta p_s^l = \frac{1}{2}\left(p_{s+1}^l - p_{s-1}^l\right) \tag{6.19}$$

$$\Delta p_s^{cog} = \frac{1}{2}\left(p_{s+1}^{cog} - p_{s-1}^{cog}\right) \tag{6.20}$$

$$\Delta \alpha_s^u = \frac{1}{2}\left(\alpha_{s+1}^u - \alpha_{s-1}^u\right) \tag{6.21}$$

$$\Delta \alpha_s^l = \frac{1}{2}\left(\alpha_{s+1}^l - \alpha_{s-1}^l\right) \tag{6.22}$$

$$\Delta \alpha_s^{cog} = \frac{1}{2}\left(\alpha_{s+1}^{cog} - \alpha_{s-1}^{cog}\right) \tag{6.23}$$

$$\Delta n_s^{tr} = \frac{1}{2}\left(n_{s+1}^{tr} - n_{s-1}^{tr}\right) \tag{6.24}$$

$$\Delta rel_s^{(1)} = \frac{1}{2}\left(rel_{s+1}^{(1)} - rel_{s-1}^{(1)}\right) \tag{6.25}$$

$$\Delta rel_s^{(2)} = \frac{1}{2}\left(rel_{s+1}^{(2)} - rel_{s-1}^{(2)}\right) \quad . \tag{6.26}$$

Merkmalsoptimierung

Um die Merkmalsrepräsentation zu optimieren, werden die Merkmalsvektoren einer linearen Diskriminanzanalyse (LDA) unterzogen. Mit dem in Abschnitt 3.5.2 beschriebenen Verfahren wird eine lineare Transformation auf die Merkmalsvektoren angewandt, um eine Dekorrelation der Merkmale bei einer gleichzeitigen Verbesserung der Klassenseparabilität zu erreichen.

Die Berechnung der Transformation erfolgt anhand der Trainingsstichprobe. Da in die LDA Transformation Klasseninformationen miteinbezogen werden, ist allerdings eine annotierte Trainingstichprobe erforderlich. Daher wird im ersten Schritt ein “normales” Training mit nichttransformierten Merkmalen durchgeführt, sodass mit dem resultierenden Erkenner die Trainingsstichprobe auf HMM-Zustandsebene annotiert werden kann. Anhand der annotierten Trainingstichprobe wird daraufhin die LDA-Transformation ermittelt. Die Transformation wird dann auf alle Merkmalsvektoren

angewandt, sodass anschließend ein neues Training mit transformierten Merkmalen vorgenommen werden kann.

6.8 Statistische Modellierung und Erkennung

Die statistische Modellierung und Erkennung wird ähnlich wie bei dem online System mit Hilfe der ESMERALDA-Entwicklungsumgebung auf Basis von HMMs vorgenommen. Insgesamt werden in diesem Erkennungssystem 77 HMMs verwendet, wobei 52 HMMs zur Modellierung der Buchstaben eingesetzt werden, 12 für Satzzeichen und Klammern und 10 für Ziffern. Die drei übrigen HMMs werden zur Modellierung der Wortzwischenräume verwendet. Da die Länge der Wortzwischenräume erheblich variiert (zwischen vier und 550 Pixel) werden diese in die Gruppen kurze, mittlere und lange Abstände eingeteilt und mit einem separaten HMM pro Gruppe modelliert. Zur Worterkennung auf Basis eines Lexikons werden die einzelnen Buchstabenmodelle wie im online System zu Verbundmodellen zusammengeschaltet (siehe Abbildung 3.25).

Die verwendeten HMMs weisen Bakis-Topologie und eine semi-kontinuierliche Emissionsmodellierung auf Basis Gauß'scher Mischverteilungsdichten auf. Die Mittelwertvektoren und Kovarianzmatrizen der Gaußdichten werden mit Hilfe des k-means Algorithmus anhand der Merkmalssequenzen der Trainingsstichprobe geschätzt, wobei auch hier diagonale Kovarianzmatrizen verwendet werden.

Das Training der HMMs erfolgt auf Basis des Standard Baum-Welch Algorithmus (siehe Seite 85). Zur Dekodierung des Erkennungsmodells wird der Viterbi Beam-Search Algorithmus eingesetzt (siehe Seite 84).

Neben der lexikonbasierten Erkennung anhand von Verbundmodellen wird außerdem eine lexikonfreie Erkennung durchgeführt. Dabei werden statistische Sprachmodelle in den Erkennungsprozess integriert, sodass Schätzwerte für die Auftrittswahrscheinlichkeiten von Buchstabensequenzen vorliegen, mit denen die Generierung unwahrscheinlicher Buchstabenhypothesen vermieden werden kann. Hier kommt ein Tri-Gramm Sprachmodell zum Einsatz, das anhand der Trainingsstichprobe geschätzt wird. Das Problem nicht beobachteter Tri-Gramme wird durch Umverteilung von Wahrscheinlichkeitmasse mit Hilfe des absolute discounting und Rückzug auf allgemeinere Verteilungen (backing-off) behandelt (siehe Abschnitt 3.7).

6.9 Adaption

Wie bei dem online System auch, wird das videobasierte Material aufgrund des geringen Umfangs nur zur Evaluation des Erkennungssystems eingesetzt. Training und Validierung erfolgen vielmehr mit Hilfe von scannerbasierten Daten. Um den Mismatch zwischen den Trainings- und Anwendungsbedingungen des Erkennungssystems zu kompensieren, wird eine Adaption der HMM-Parameter vorgenommen.

Die Adaption wird durch MLLR-Schätzung der Mittelwertvektoren der Mischverteilungsdichte durchgeführt. Dabei wird nur eine Regressionsklasse verwendet, sodass die Mittelwertvektoren aller Gaußdichten durch ein und dieselbe Transformation adaptiert werden. Die Berechnungsvorschrift für die Transformationsmatrix entspricht damit der Gleichung 5.34 in Kapitel 5.

6.10 Zusammenfassung

In diesem Kapitel wurde das realisierte System zur videobasierten Erkennung von Tafelanschrift vorgestellt. Im Gegensatz zu dem in Kapitel 5 vorgestellten System basiert es nicht auf der Erfassung der Schreibdynamik, sondern auf der Detektion von statischen Textregionen in der Videobildfolge. Die Verarbeitung der detektierten Bildbereiche wird dabei inkrementell durchgeführt, d.h. gegenüber einem scannerbasierten offline System, bei dem ein Bild nach Beendigung des Schreibvorgangs aufgenommen wird, erfolgt hier die Aufnahme einer Bildsequenz des gesamten Schreibprozesses, sodass die Erkennung schrittweise durchgeführt werden kann.

Die Voraussetzung der schrittweisen Verarbeitung ist die Detektion der im jeweils aktuellen Bild neu hinzugekommenen Textregionen. Dazu wird ein zweistufiges Verfahren eingesetzt, das im ersten Schritt eine schnelle Unterscheidung zwischen Text,- Hintergrund- und Störbereichen vornimmt. Im zweiten Schritt erfolgt dann die Zusammenfassung der Schriftkomponenten zu Textregionen, die einzelnen Wörtern oder Teilen von Textzeilen entsprechen.

Anhand der extrahierten Textregionen werden einige Vorverarbeitungsschritte durchgeführt. Aufgrund der inhomogenen Beleuchtung erfolgt im ersten Schritt eine lokale Binarisierung. Um die Variabilität der Schrift zu kompensieren, werden anschließend Orientierung, Neigung und Größe der Schrift normalisiert, wobei die Orientierungs- und Neigungskorrektur ebenfalls mit Hilfe lokaler Methoden vorgenommen werden.

Die vorverarbeiteten Textregionen sind der Ausgangspunkt für die Merkmalsextraktion. Dabei wird innerhalb eines sliding windows, das von links nach rechts über die Textregion bewegt wird, ein Satz geometrischer Merkmale extrahiert. Die Klassifikation erfolgt wie im online System auf Basis von Hidden Markov Modellen. Da die HMMs mit scannerbasierten Daten trainiert werden, wird auch hier eine MLLR-Adaption durchgeführt, um das Ungleichverhältnis zwischen den Trainings- und den Anwendungsbedingungen des Erkennungssystems zu kompensieren.

Die Evaluation des realisierten Systems zur inkrementellen videobasierten Erkennung von Tafelanschrift wird im folgenden Kapitel in Abschnitt 7.3 beschrieben.

7 Evaluation

In diesem Kapitel erfolgt die Evaluation der in den beiden vorangegangenen Kapiteln vorgestellten Systeme zur videobasierten online bzw. zur inkrementellen videobasierten offline Handschrifterkennung. Da für den Anwender letztendlich die Klassifikationsleistung der Systeme ausschlaggebend für ihren möglichen Einsatz ist, wird die Evaluation der Systeme auf Basis der erzielten Fehlerraten anhand videobasierter Teststichproben vorgenommen.

7.1 Evaluationsmaß und Konfidenzintervalle

Um die Klassifikationsleistung eines Systems zur automatischen Handschrifterkennung sinnvoll beurteilen zu können, ist ein möglichst aussagekräftiges Evaluationsmaß erforderlich. Häufig wird hierfür die *Wortakkuratheit* verwendet, die auf der Anzahl der Erkennungsfehler auf Wortebene im Vergleich zu dem Referenztext basiert. Da das Erkennungsresultat jedoch nicht notwendigerweise die gleiche Anzahl von Wörtern aufweist wie der Referenztext, muss zur Berechnung der Wortakkuratheit vorher eine wortweise Zuordnung von Referenztext und Erkennungsresultat – i.d.R durch dynamische Programmierung – vorgenommen werden. Anhand der resultierenden Zuordnung kann die Wortakkuratheit dann durch die folgende Vorschrift berechnet werden:

$$\text{WA} = 1 - \frac{N_{sub} + N_{del} + N_{ins}}{N_{all}} \quad . \tag{7.1}$$

Dabei bezeichnet N_{sub} die absolute Häufigkeit der vertauschten Wörter gegenüber dem Referenztext, N_{del} und N_{ins} die Anzahl der irrtümlich ausgelassenen bzw. eingefügten Wörter. Der Term N_{all} bezeichnet die Gesamtzahl der Wörter im Referenztext.

Ein weiteres Evaluationsmaß ist die *Wortfehlerrate* (word error rate, WER). Diese kann auf Basis der (in Prozent angegebenen) Wortakkuratheit wie folgt berechnet werden:

$$\text{WER} = 100\% - \text{WA} \quad . \tag{7.2}$$

Mit der Wortfehlerrate lässt sich somit der Anteil fehlerhaft erkannter Wörter angeben. Dieses Maß wird häufig verwendet, um Verbesserungen der Erkennungsleistung von Systemen zu verdeutlichen, die bereits eine gute Wortakkuratheit aufweisen.

Die auf der Teststichprobe gemessene Erkennungsleistung eines Systems stellt jedoch nur eine Schätzung der tatsächlichen Erkennungsleistung auf der Grundgesamtheit dar, für deren Ermittlung theoretisch unendlich viele Versuche durchgeführt werden müssten [Pau95]. Stattdessen wird für die anhand der Teststichprobe ermittelte Er-

kennungswahrscheinlichkeit $\hat{p}$ ein *Konfidenzintervall* $[p_u, p_o]$ berechnet. Dieses Konfidenzintervall gibt einen Bereich an, innerhalb dessen die tatsächliche Erkennungswahrscheinlichkeit p mit einer vorgegebenen statistischen Sicherheit, dem sogenannten *Konfidenzniveau*, liegt.

Da die vorliegenden Erkennungsexperimente jeweils als Bernoulli-Prozess aufgefasst werden können, bei dem nur die beiden Ereignisse A (erkannt) und $\bar{A}$ (nicht erkannt) eintreten können, ist das Konfidenzintervall für die Wahrscheinlichkeit p demnach anhand der Binomialverteilung zu bestimmen. Bei genügend großem Stichprobenumfang n kann die Wahrscheinlichkeit p allerdings approximativ als normalverteilt angesehen werden [Sch00]:

$$\frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \approx N(0,1) \quad . \tag{7.3}$$

Die obere bzw. untere Grenze des Konfidenzintervalls zum Konfidenzniveau $1 - \alpha$ ergibt sich daraus wie folgt (siehe u.a. [Sch00]):

$$p_u = \frac{n}{n+z^2}\left(\hat{p} + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}\right) \tag{7.4}$$

$$p_o = \frac{n}{n+z^2}\left(\hat{p} + \frac{z^2}{2n} + z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}\right) \quad . \tag{7.5}$$

Hierbei bezeichnet z das $(1 - \frac{\alpha}{2})$-Quantil der Standard-Normalverteilung, dessen Wert in Tabellen nachgeschlagen werden kann. Für die hier durchgeführten Experimente wird ein Konfidenzniveau $1 - \alpha$ von 95% verwendet, sodass also mit einer statistischen Sicherheit von 95% die tatsächliche Erkennungswahrscheinlichkeit innerhalb des Konfidenzintervalls liegt.

Auf Basis des Konfidenzintervalls lassen sich außerdem Aussagen darüber treffen, ob Veränderungen der Fehlerraten, beispielsweise durch modifizierte Systemparameter, signifikant sind. Liegt demzufolge die veränderte Fehlerrate außerhalb des Konfidenzintervalls, so wird die Veränderung als signifikant aufgefasst, anderenfalls wird von einer zufälligen Schwankung ausgegangen.

7.2 Online System

In diesem Abschnitt werden die Experimente beschrieben, die zur Evaluation des in Kapitel 5 vorgestellten Systems zur videobasierten online Handschrifterkennung durchgeführt wurden. Der den Experimenten zugrundeliegende Textkorpus besteht aus deutschen Städtenamen, wobei das verwendete Lexikon 201 Wörter umfasst. Wegen der unterschiedlichen Realisierungsmöglichkeiten der diakritischen Zeichen – beispielsweise direkt nach dem Basisbuchstaben oder erst am Ende des Wortes – vergrößert sich das Lexikon auf insgesamt 335 Einträge. Bei der Stichprobenerstellung

wurden die Wörter isoliert voneinander aufgenommen, sodass die Segmentierung von Textzeilen in Wörter entfällt und somit eine Einzelworterkennungsaufgabe vorliegt.

Wie bereits in Kapitel 5.7, Seite 134, erläutert wurde, wurde auf Grund der äußerst speicherintensiven videobasierten Datenaufnahme zum Training des Systems eine Stichprobe mit Hilfe eines WACOM-Digitalisiertabletts erstellt. Insgesamt haben dabei 13 Schreiber jeweils das komplette Lexikon, also 201 Städtenamen, geschrieben, sodass ein tablettbasierter Korpus von insgesamt $13 \times 201 = 2613$ Wörtern vorliegt. Für die Evaluation des Systems wurde dann eine kleinere Stichprobe mittels Videokamera erstellt. Hierbei wurden weitere vier Schreiber gebeten, jeweils die komplette Liste aller Wörter des Lexikons zu schreiben, sodass die videobasierte Teststichprobe $4 \times 201 = 804$ Wörter umfasst. Für jeden Schreibvorgang wurde dabei sowohl die Stifttrajektorie aufgenommen als auch – zur Evaluation der pen-up/down Unterscheidung – das Schriftbild nach Beendigung eines Schreibvorgangs[1]. Bei der tablettbasierten sowie bei der videobasierten Aufnahme wurden den Schreibern keinerlei Einschränkungen in Bezug auf den Schriftstil auferlegt.

Robustheit der Stiftverfolgung

Anhand der videobasierten Teststichprobe wird im ersten Schritt die Robustheit der Stiftverfolgung untersucht. Bei der Betrachtung der aufgenommenen Trajektorien stellt man fest, dass bei ca. 4.1% der insgesamt 804 Trajektorien die Stiftverfolgung bzw. die initiale Stiftlokalisierung fehlschlugen. So wurde bei ca. 1.7% der Trajektorien die Stiftverfolgung während des Schreibprozesses abgebrochen, da die Stiftspitze nicht im Suchbereich gefunden wurde bzw. das Template-Matching eine zu geringe Übereinstimmung lieferte. In ca. 2.4% der Fälle versagte die initiale Stiftlokalisierung. Diese Fehler wurden beispielsweise dadurch hervorgerufen, dass irrtümlich ein Schatten oder ein Bereich der Hand als Stiftposition angenommen wurde.

Tablettbasiertes Referenzsystem

Um die Erkennungsleistungen des videobasierten Systems gegenüber einem tablettbasierten System einordnen zu können, wird zuerst ein Referenzsystem evaluiert, das ausschließlich anhand der Tablettdaten trainiert und getestet wurde. Gegenüber den in Kapitel 5 beschriebenen Schritten zur Verarbeitung videobasierter Daten weist dieses tablettbasierte Referenzsystem einige Unterschiede auf. So werden die Korrektur der geometrischen Verzerrung, die Glättung und das Impuls-Resampling in dem videobasierten System durchgeführt, um die Videodaten so zu verbessern, dass annähernd die Qualität von Tablettdaten erreicht wird. In einem tablettbasierten System können die-

[1] Idealerweise müsste für jeden Schreibvorgang die gesamte Bildsequenz abgespeichert werden. Bei einer durchschnittlichen Dauer von ca. 3.8s pro Schreibvorgang und einer Bildaufnahmerate von 50 Bildern pro Sekunde würde dabei jedoch bei 804 Schreibvorgängen ein Datenvolumen von mindestens 34 GByte anfallen. Indem nur das jeweils letzte Bild eine Schreibvorgangs abgespeichert wird, kann das Datenvolumen auf “handliche” 176 MByte verringert werden.

se Schritte daher entfallen, bzw. im Resamplingschritt kann eine lineare Interpolation (Impuls-Term $\alpha = 0$) durchgeführt werden. Zudem ist auch die pen-up/down Klassifikation hier nicht erforderlich, da diese Informationen direkt vom Digitalisiertablett geliefert werden.

Zur Evaluation des Referenzsystems wird der verfügbare tablettbasierte Korpus in eine Trainings- und eine Teststichprobe aufgeteilt, wobei die Trainingsstichprobe aus den Daten von sieben Schreibern, also 7×201 Wörtern besteht, während die restlichen Daten, also 6×201 Wörter, für den Test verwendet werden. Die Evaluationsergebnisse, die sich für das tablettbasierte Referenzsystem ergeben, sind in der Tabelle 7.1 dargestellt. Der $\pm$-Term gibt dabei die Größe des Konfidenzintervalls zum Konfidenzniveau von 95% an.

Schreiber	Wortfehlerrate
a	$1.0\% \pm 1.6$
b	$0.0\% \pm 1.0$
c	$1.0\% \pm 1.6$
d	$16.9\% \pm 5.2$
e	$0.5\% \pm 1.3$
f	$8.0\% \pm 3.8$
Gesamt	$4.6\% \pm 1.2$

Tabelle 7.1: Wortfehlerraten des tablettbasierten Referenzsystems

Videobasiertes Basissystem

Nachdem die Erkennungsleistung des tablettbasierten Systems ermittelt wurde, wird nun ein videobasiertes Basissystem evaluiert. Um dieses System mit dem tablettbasierten vergleichen zu können und eine Bewertungsgrundlage für die in Kapitel 5 vorgestellten Methoden zur Verbesserung der videobasierten Trajektoriendaten zu schaffen, wird hier zunächst von der Integration dieser Verfahren abgesehen. Demzufolge wird sowohl auf die Glättung der Trajektorie durch Binomialfilterung bzw. Impuls-Resampling verzichtet, als auch auf die Korrektur der geometrischen Verzerrung, die durch den Blickwinkel der Kamera bedingt ist. Zudem wird keine pen-up/down Klassifikation durchgeführt, vielmehr werden alle Trajektorienpunkte als pen-down angenommen.

Zur Evaluation des Basissystems anhand der videobasierten Teststichprobe (4×201 Wörtern) wird die Trainingsstichprobe zur Optimierung der HMM-Parameter erweitert. Der tablettbasierte Korpus wird dazu in eine Trainings- und Validierungsmenge aufgeteilt, sodass die Trainingsmenge 11×201 Wörter und die Validierungsmenge demzufolge 2×201 Wörter umfasst. Der Parametersatz, der zu den besten Ergebnissen auf der Validierungsmenge führt, wird dann zur Evaluation der videobasierten Teststichprobe eingesetzt.

Schreiber	Wortfehlerrate
1	27.4% ± 6.1
2	48.4% ± 6.8
3	28.4% ± 6.2
4	65.2% ± 6.5
Gesamt	42.4% ± 3.4

Tabelle 7.2: Ergebnisse des videobasierten Basissystems

Die Ergebnisse des Basissystems auf der videobasierten Teststichprobe zeigt die Tabelle 7.2. Die Wortfehlerrate beträgt auf der gesamten Stichprobe demnach 42.4%. Im Vergleich dazu erreicht das tablettbasierte Referenzsystem eine Wortfehlerrate von 4.6% (7 Schreiber Training, 6 Schreiber Test) und 2.7%, wenn es auf der erweiterten Trainingsstichprobe trainiert wird (11 Schreiber Training, 2 Schreiber Test).

Aus der Tabelle geht außerdem hervor, dass die Erkennungsleistung des Systems je nach Schreiber stark variiert. So fallen die Ergebnisse bei den Schreibern zwei und vier gegenüber den übrigen beiden deutlich ab. Bei Betrachtung der aufgenommenen Stifttrajektorien überrascht dieses Resultat jedoch nicht. Beispielsweise unterscheidet sich der Schriftstil von Schreiber zwei deutlich von denen, die zum Training des Systems zur Verfügung standen. Während die Trainingsschreiber überwiegend fließende Handschrift verwendeten, hat der Schreiber zwei die Buchstaben isoliert voneinander geschrieben. Bei Schreiber vier ist dagegen die Schreibgeschwindigkeit vergleichsweise hoch. Die mittlere Dauer der Schreibprozesse beträgt bei diesem Schreiber ca. 3.4s, während der Mittelwert über alle Schreiber bei ca. 3.8s liegt. Diese hohe Schreibgeschwindigkeit führt bei der geringen Abtastrate von 50 Hz zu sprunghaften Richtungsänderungen der Trajektorie, sodass lokal hohe Krümmungswerte auftreten, die häufig in einer fehlerhaften Strokesegmentierung resultieren.

Anhand der Ergebnisse des videobasierten Basissystems wird somit deutlich, dass die Qualität der videobasierten Daten bei weitem nicht ausreicht, um eine annähernd vergleichbare Erkennungsleistung zu erzielen, wie sie das tablettbasierte Referenzsystem aufweist. In den folgenden Experimenten wird daher untersucht, ob die in Kapitel 5 beschriebenen Verfahren die Qualität der videobasierten Daten entsprechend verbessern und damit eine Verringerung der Fehlerraten erzielt werden kann.

Integration der Pen-Up/Down Klassifikation

In diesem Experiment wird die Erkennungsleistung des videobasierten Basissystems gemessen, das um die in Abschnitt 5.2.3 vorgestellte Komponente zur pen-up/down Klassifikation erweitert wurde. Ein Trajektorienpunkt wird demnach als pen-up klassifiziert, wenn in dem Bild des Schriftzuges in einem Abstand von maximal zwei Bildpunkten von der gemessenen Stiftposition kein Schriftpixel detektiert werden kann oder die Intergruppenstreuung des Grauwerthistogramms einen vorgegebenen Schwellwert unterschreitet (siehe Abschnitt 5.2.3, Seite 119).

Die dort beschriebene pen-up/down Unterscheidung basiert auf der Analyse lokaler Bildauschnitte, die zur Vermeidung von Verdeckungen mit einem festen zeitlichen Versatz extrahiert werden. Da bei der Erstellung der Teststichprobe neben der Trajektorie jedoch nur ein einziges Bild des jeweiligen Schreibvorgangs aufgenommen wurde, das den gesamten Schriftzug nach Beendigung des Schreibprozesses zeigt, kann hier die schritthaltende pen-up/down Bestimmung nicht durchgeführt werden. Vielmehr muss die pen-up/down Unterscheidung für alle Trajektorienpunkte am Ende des Schreibprozesses vorgenommen werden. Da in dem jeweiligen Bild stets der gesamte Schriftzug sichtbar ist, tritt das Problem von Verdeckungen hier nicht auf. Die Erkennungsergebnisse, die durch die Integration der pen-up/down Unterscheidung erzielt werden können, sind in der Tabelle 7.3 dargestellt.

Schreiber	Wortfehlerrate
1	23.4% $\pm$ 5.8
2	39.3% $\pm$ 6.7
3	24.4% $\pm$ 5.9
4	60.7% $\pm$ 6.7
Gesamt	36.9% $\pm$ 3.3

Tabelle 7.3: Ergebnisse der videobasierten Erkennung mit integrierter pen-up/down Unterscheidung

Vergleicht man die erzielte Wortfehlerrate mit der des videobasierten Basissystems, so wird deutlich, dass sich durch die Integration der pen-up/down Unterscheidung eine signifikante Verringerung der Wortfehlerrate von 42.4% auf 36.9% ergibt. Bei Schreiber zwei ist die Verbesserung der Erkennungsleistung von 48.4% auf 39.3% besonders auffällig. Dies ist auch zu erwarten, da bei diesem Schreiber aufgrund der isoliert geschriebenen Buchstaben der Anteil von pen-up Segmenten am größten ist.

Glättung durch Impuls-Resampling und Binomialfilterung

Da auf Grund der im Vergleich zu Digitalisiertabletts geringeren Aufnahmerate der Videokamera die Stifttrajektorie sprunghafte Richtungsänderungen an den Trajektorienpunkten aufweist, wird in diesem Schritt neben der pen-up/down Klassifikation zusätzlich eine Glättung der Trajektorie durchgeführt. Dazu wird sowohl das Resampling mit Impulsterm in das Basissystem integriert (Abschnitt 5.3.3), als auch die in Abschnitt 5.3.1 beschriebene Binomialfilterung der aufgenommenen Datenpunkte.

Der vorgegebene Punktabstand für die Neuabtastung beträgt 200μm. Für den Impulsfaktor α hat sich ein Wert von 0.9975 als optimal herausgestellt. Durch die Integration des so parametrisierten Impuls-Resampling in das videobasierte Basissystem mit pen-up/down Klassifikation konnten die in Tabelle 7.4, mittlere Spalte, dargestellten Erkennungsergebnisse erreicht werden. Wird vor der Neuabtastung zusätzlich eine Binomialfilterung der Trajektorie mit Hilfe der Filtermaske $\frac{1}{4}[\ 1\ 2\ 1\ ]$ vorgenommen,

	Wortfehlerrate	
Schreiber	Impuls-Resampling (IR)	IR plus Binomialfilterung
1	12.4% ± 4.6	10.4% ± 4.3
2	29.8% ± 6.3	20.9% ± 5.6
3	15.4% ± 5.0	13.9% ± 4.8
4	30.3% ± 6.3	47.8% ± 6.8
Gesamt	22.1% ± 2.9	23.3% ± 2.9

Tabelle 7.4: Videobasierte Erkennung mit pen-up/down Unterscheidung und Impuls-Resampling bzw. mit pen-up/down Unterscheidung, Impuls-Resampling und Binomialfilterung

so ergeben sich die in der rechten Spalte von Tabelle 7.4 gezeigten Erkennungsergebnisse. Aus dem Vergleich mit den Ergebnissen aus Tabelle 7.3 geht hervor, dass die Glättung der Trajektorie durch das Impuls-Resampling gegenüber dem Resampling mit linearer Interpolation zu einer signifikanten Verbesserung der Erkennungsleistung bei allen vier Schreibern führt. Auf der gesamten Stichprobe kann die Wortfehlerrate von 36.9% (Tabelle 7.3) auf 22.1% reduziert werden. Wird die Trajektorie außerdem vor dem Resampling durch Binomialfilterung geglättet, so kann die Wortfehlerrate bei den Schreibern eins bis drei weiter verbessert werden, bei Schreiber vier ergibt sich dagegen eine deutliche Steigerung der Fehlerrate.

Korrektur der geometrischen Verzerrung

Wird neben der pen-up/down Unterscheidung, dem Impuls-Resampling und der Binomialfilterung zusätzlich die vertikale Stauchung der Trajektorie korrigiert, die durch den Blickwinkel der Kamera von ca. $65°$ hervorgerufen wird (siehe Abschnitt 5.3.2), so werden die in der Tabelle 7.5 dargestellten Ergebnisse erreicht. Anhand der Tabelle kann man eine Verringerung der Wortfehlerrate ablesen. Diese Verbesserung liegt jedoch innerhalb des Konfidenzintervalls, sodass sie nicht als signifikant aufzufassen ist.

Schreiber	Wortfehlerrate
1	9.9% ± 4.2
2	16.4% ± 5.1
3	13.9% ± 4.8
4	43.3% ± 6.8
Gesamt	20.9% ± 2.8

Tabelle 7.5: Videobasierte Erkennung mit pen-up/down Unterscheidung, Impuls-Resampling, Binomialfilterung und Korrektur der aufnahmebedingten Verzerrung.

Adaption

Um den Mismatch zwischen den Trainings- und den Anwendungsbedingungen zu verringern, wird neben den Vorverarbeitungsmaßnahmen außerdem eine Adaption der HMM-Parameter vorgenommen. Wie in Abschnitt 5.7 beschrieben wurde, erfolgt hier eine MLLR-Adaption der Mittelwertvektoren der Emissionsmodellierung. Auf Grund des geringen Umfangs der videobasierten Stichprobe wird nur eine Regressionsklasse definiert, sodass die Mittelwertvektoren aller Gaußdichten gemeinsam durch eine Transformation adaptiert werden.

Im ersten Fall erfolgt eine schreiberabhängige Adaption im überwachten Modus, d.h. die korrekte Transkription der zur Adaption verwendeten Trajektorien steht zur Verfügung. Für die Adaption wird das videobasierte Material des jeweiligen Schreibers in zwei Hälften geteilt, sodass die eine (korrekt gelabelte) Hälfte zur Adaption, die andere Hälfte dann zur Evaluation verwendet werden kann. Die Ergebnisse der schreiberabhängigen, überwachten Adaption sind in der Tabelle 7.6 dargestellt. Zum Vergleich sind in der mittleren Spalte die Ergebnisse des videobasierten Systems ohne Adaption abgebildet, die auf der gleichen Hälfte des Testkorpus erzielt wurden. Es wird deutlich, dass durch die überwachte Adaption eine leichte Verringerung der Wortfehlerrate von 21.8% auf 18.8% erreicht wird.

	Wortfehlerrate	
Schreiber	50% Test	Adaption 50% Test
1	9.9% $\pm$ 5.9	9.9% $\pm$ 5.9
2	16.8% $\pm$ 7.3	17.8% $\pm$ 7.4
3	15.8% $\pm$ 7.1	9.9% $\pm$ 5.9
4	44.5% $\pm$ 9.5	37.6% $\pm$ 9.3
Gesamt	21.8% $\pm$ 4.0	18.8% $\pm$ 3.8

Tabelle 7.6: Ergebnisse der schreiberabhängigen, überwachten Adaption auf 50% der Teststichprobe. Die mittlere Spalte zeigt zum Vergleich die Ergebnisse ohne Adaption auf 50% der Teststichprobe.

Im zweiten Experiment wird die Adaption ebenfalls schreiberabhängig durchgeführt, im Unterschied zum ersten Experiment wird jedoch keine manuell erstellte Transkription eingesetzt. Die Adaption erfolgt hier vielmehr unüberwacht, wobei die (u.U. fehlerhafte) Transkription automatisch durch eine vorgeschaltete Viterbi Beam-Search Dekodierung ermittelt wird. Die Evaluation wird auf der gesamten Teststichprobe vorgenommen.

Die erreichten Ergebnisse der unüberwachten Adaption zeigt die Tabelle 7.7. Vergleicht man das Gesamtergebnis von 18.3% mit der Wortfehlerrate von 20.9% aus Tabelle 7.5, so stellt man eine Verbesserung der Erkennungsleistung fest, die jedoch knapp unterhalb der Signifikanzschwelle bleibt. Eine signifikante Verringerung der Wortfehlerrate von 43.3% auf 33.3% ergibt sich nur für den vierten Schreiber. Dies

Schreiber	Wortfehlerrate
1	9.0% ± 4.0
2	21.4% ± 5.6
3	9.4% ± 4.1
4	33.3% ± 6.5
Gesamt	18.3% ± 2.7

Tabelle 7.7: Ergebnisse der schreiberabhängigen, unüberwachten Adaption.

deutet darauf hin, dass insbesondere bei diesem Schreiber sich die Lage der Merkmalsvektoren im Raum deutlich von der Merkmalsrepräsentation der "Trainingsschreiber" unterscheidet.

7.3 Offline System

Nachdem im vorherigen Abschnitt die Erkennungsleistung des videobasierten online Systems dargestellt wurde, erfolgt hier die Evaluation des in Kapitel 6 vorgestellten Systems zur inkrementellen videobasierten offline Erkennung von Tafelanschrift.

Ähnlich wie bei dem online System wird auch hier das Training der Klassifikationsparameter nicht anhand des videobasierten Materials vorgenommen – dieses wird auf Grund des geringen Umfangs ausschließlich für die Adaption und die Evaluation eingesetzt. Das Training wird vielmehr auf Basis eines Ausschnitts der IAM-Datenbank (siehe Abschnitt 3.9.1) mit eingescannten handschriftlichen Formularen durchgeführt.

Zur Erstellung der videobasierten Stichprobe haben 6 Personen beigetragen. Sie haben jeweils einen Textausschnitt – aus der Kategorie F01 des LOB-Korpus – an das Whiteboard geschrieben. Die Schreiber wurden dabei gebeten, auf eine gute Separierbarkeit der Zeilen zu achten, ansonsten wurden ihnen keinerlei Einschränkungen bzgl. des Schriftstils auferlegt. Insgesamt wurden so 79 Textzeilen geschrieben. Dies entspricht einer Gesamtzahl von 497 Wörtern. Die Aufnahme der Schreibprozesse erfolgte im Vollbildmodus mit einer Bildrate von 5 Hz, daraus resultiert ein Speicherbedarf für die Bildsequenzen von insgesamt 2,1 GByte.

Zur Evaluation der Erkennungsleistung werden unterschiedliche Experimente durchgeführt. Dabei wird jeweils sowohl die Wortfehlerrate auf Basis eines Lexikons gemessen, als auch die Zeichenfehlerrate, die ohne Verwendung eines Lexikons aus der Sequenz der Buchstabenhypothesen resultiert[2]. Darüberhinaus wird bei der lexikonfreien Erkennung ein statistisches Tri-Gramm Sprachmodell (auf Zeichenebene) in den Dekodierungsprozess integriert, um die Erzeugung unwahrscheinlicher Buchstabenhypothesen zu vermeiden und somit die Zeichenfehlerrate zu verbessern.

[2] Die Zeichenfehlerrate ist analog zur Wortfehlerrate definiert, mit dem Unterschied, dass als Grundlage einzelne Zeichen anstelle von Wörtern verwendet werden.

Robustheit der Textregionenextraktion

Die Voraussetzung für die Erkennung der Schrift am Whiteboard ist die Detektion der entsprechenden Textregionen. Das dazu eingesetzte Verfahren sollte robust sein, d.h. es sollten möglichst alle Textregionen als solche erkannt und extrahiert werden, wobei gleichzeitig Stör- bzw. Hintergrundbereiche unterdrückt werden sollten.

Die Messung der Robustheit der Textregionenextraktion wurde auf Basis der aufgenommen Teststichprobe durchgeführt. Dabei wurden mit dem in Abschnitt 6.3 beschriebenen Verfahren anhand der 79 geschriebenen Textzeilen insgesamt 124 Textregionen extrahiert. Keine einzige Textregion wurde irrtümlich als Hintergrund- oder Störregion erkannt, wobei umgekehrt allerdings 17 Hintergrund- bzw. Störregionen fälschlicherweise als Textregionen detektiert wurden. Somit beträgt der Anteil fehlerhaft detektierter Textregionen 14%.

Scannerbasiertes Referenzsystem

Wie bereits erwähnt erfolgt die Parameteroptimierung des offline Systems anhand von scannerbasiertem Material. Für das Training wird dazu der Abschnitt A01-D07 der IAM-Datenbank verwendet. Dieser Abschnitt umfasst ca. 4200 Textzeilen (ca. 36000 Wörter), die insgesamt von über 200 Personen geschrieben wurden. Die Validierung erfolgt anhand des Abschnitts E01-F04 der IAM-Datenbank. Dieser Teil enthält ca. 1050 Textzeilen (ca. 9500 Wörter) und wurde anhand der Daten von mehr als 50 Schreibern erstellt. Das verwendete Vokabular von Trainings- und Validierungsmenge zusammen besteht aus ca. 7000 Wörtern.

Die Segmentierung, Merkmalsextraktion und statistische Modellierung der IAM-Daten wird mit denselben Verfahren vorgenommen, die auch für das videobasierte System eingesetzt werden. Unterschiede bestehen dagegen in der Vorverarbeitung der Daten. So wird auf Grund der homogenen Beleuchtung im scannerbasierten System mit einem globalen Binarisierungsschwellwert gearbeitet, der mit der Otsu-Methode bestimmt wird. Die Korrektur der Schriftorientierung und -neigung erfolgt ebenfalls global, d.h. mit einem Rotations- bzw. Schwerwinkel pro Schriftzeile. Eine lokale Korrektur brachte keinerlei Verbesserungen der Erkennungsraten. Dies liegt darin begründet, dass bei der Erstellung der Stichprobe ein Unterlegblatt verwendet wurde, sodass horizontale Referenzlinien sichtbar waren und der Schriftstil somit innerhalb einer Zeile relativ homogen ist.

Die Fehlerraten, die mit diesem scannerbasierten System auf der Validierungsmenge erreicht werden, sind in der Tabelle 7.8 dargestellt. Die Zeichenfehlerrate beläuft sich damit auf 27.3%. Durch die Integration eines Tri-Gramm Sprachmodells in den Erkennungsprozess kann die Zeichenfehlerrate auf 19.4% verbessert werden. Das verwendete Sprachmodell wurde dabei anhand der Trainingsstichprobe ermittelt. Die Perplexität des Tri-Gramm Modells auf der Validierungsmenge beträgt 8.9. Auf Wortebene beträgt die Fehlerrate unter Verwendung eines ca. 7000 Wörter umfassenden Lexikons 37.4%.

Zeichenfehlerrate		Wortfehlerrate
ohne Sprachm.	Tri-Gramm	7k Lexikon
27.3% $\pm$ 0.4	19.4% $\pm$ 0.3	37.4% $\pm$ 0.9

Tabelle 7.8: Ergebnisse des scannerbasierten Referenzsystems.

Anhand von drei weiteren Experimenten wird die Erkennungsleistung des Systems gemessen, bei dem die Merkmalsrepräsentation einer LDA-Transformation unterzogen wird (siehe Tabelle 7.9). Im ersten Fall wird dabei die Dimension der Merkmalsvektoren beibehalten, wohingegen im zweiten und dritten Experiment die Dimension von 18 auf 16 bzw. von 18 auf 12 Merkmalskomponenten reduziert wird. Die Dimensionsreduktion geschieht durch Auswahl derjenigen Eigenvektoren, die zu den 16 bzw. 12 größten Eigenwerten gehören, wobei mit den 16 Eigenvektoren ca. 99% der Gesamtvarianz und mit 12 Eigenvektoren ca. 95% der Gesamtvarianz der Merkmalskomponenten umfasst werden. Die Ergebnisse zeigen jedoch, dass durch die LDA-Transformation die Erkennungsleistung nicht verbessert werden kann. Dies spricht dafür, dass die untransformierte Merkmalsrepräsentation bereits unkorreliert ist und gute Diskriminanzeigenschaften aufweist.

	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	7k Lexikon
LDA 18	28.1% $\pm$ 0.4	19.6% $\pm$ 0.3	38.8% $\pm$ 0.9
LDA 16	28.3% $\pm$ 0.4	20.0% $\pm$ 0.3	40.0% $\pm$ 0.9
LDA 12	28.8% $\pm$ 0.4	20.5% $\pm$ 0.3	40.1% $\pm$ 0.9

Tabelle 7.9: Ergebnisse des scannerbasierten Systems mit LDA-Transformation der Merkmale. Erste Zeile: LDA-Transformation ohne Dimensionsreduktion. Zweite und dritte Zeile: Dimensionsreduktion auf 16 bzw. 12 Merkmalskomponenten.

Videobasiertes Basissystem

Das anhand der scannerbasierten Stichprobe trainierte Erkennungssystem wird nun zur Klassifikation der videobasierten Teststichprobe eingesetzt. Im ersten Schritt wird dabei die Erkennungsleistung des Basissystems gemessen, d.h. wie im scannerbasierten Referenzsystem wird sowohl eine globale Binarisierung als auch eine globale Orientierungs- und Neigungskorrektur der Schrift durchgeführt.

Die Ergebnisse des videobasierten Basissystems sind in der Tabelle 7.10 dargestellt. Insgesamt wird eine Fehlerrate von 38.5% auf Zeichenebene erreicht. Durch Integration des Tri-Gramm Sprachmodells (Perplexität von 8.1 auf dem videobasierten Testset) in den Erkennungsprozess kann die Zeichenfehlerrate auf 30.8% verbessert werden. Auf Wortebene ergibt sich durch den Einsatz eines 400 Wörter umfassenden Lexikons eine Fehlerrate von 41.3%.

Schreiber	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	400 Wörter Lexikon
1	38.0% ± 3.0	34.6% ± 2.9	33.3% ± 6.3
2	40.8% ± 4.2	26.8% ± 3.9	52.9% ± 9.6
3	48.6% ± 4.4	42.7% ± 4.3	66.0% ± 9.5
4	37.0% ± 5.1	26.7% ± 4.7	43.1% ± 9.6
5	26.7% ± 3.8	19.6% ± 3.4	29.1% ± 7.8
6	39.9% ± 4.5	28.1% ± 4.1	28.4% ± 8.4
Gesamt	38.5% ± 1.6	30.8% ± 1.6	41.3% ± 3.6

Tabelle 7.10: Ergebnisse des videobasierten Basissystems.

Lokale Binarisierung

Die im videobasierten Basissystem eingesetzte globale Binarisierung setzt eine homogene Beleuchtung der Szene voraus, um die Schrift vom Hintergrund separieren zu können. Bei der videobasierten Erkennung von Tafelanschrift ist diese Voraussetzung aufgrund von Schattenwurf des Schreibers und der reflektierenden Tafeloberfläche häufig nicht gegeben. Um in diesen Fällen dennoch die Schrift vom Hintergrund trennen zu können, wird deshalb die in Abschnitt 6.5.1 vorgestellte Binarisierung mit lokalen Schwellwerten basierend auf der modifizierten Niblack-Methode in das Erkennungssystem integriert.

Wie man anhand der Erkennungsergebnisse in Tabelle 7.11 erkennen kann, hat die Integration der lokalen Binarisierung einen positiven Effekt. So verbessert sich die Zeichenfehlerrate von 38.5% (Basissystem) auf 34.1%. Mit Hilfe des Sprachmodells ergibt sich eine weitere Verringerung auf 27.3% korrekt erkannter Zeichen. Auch auf Wortebene verbessert sich die Erkennungsleistung. Hier ergibt sich eine Wortfehlerrate von 31.7%.

Schreiber	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	400 Wörter Lexikon
1	28.4% ± 2.7	21.0% ± 2.5	22.1% ± 5.3
2	34.9% ± 4.1	24.5% ± 3.8	47.1% ± 9.2
3	47.6% ± 4.4	47.6% ± 4.4	44.7% ± 9.3
4	35.1% ± 5.0	27.3% ± 4.8	25.9% ± 9.6
5	25.7% ± 3.8	19.0% ± 3.4	26.7% ± 8.0
6	39.1% ± 4.5	31.1% ± 4.3	32.1% ± 8.9
Gesamt	34.1% ± 1.6	27.3% ± 1.5	31.7% ± 3.3

Tabelle 7.11: Ergebnisse bei Integration der lokalen Binarisierung.

Lokale Orientierungs- und Neigungskorrektur

Das Fehlen jeglicher Referenzlinien am Whiteboard hat bei einigen Schreibern der Teststichprobe dazu geführt, dass sowohl die Basislinien der Wörter als auch die Schriftneigung innerhalb einer Textzeile oftmals stark variieren. Globale Verfahren zur Schriftnormalisierung, die einen Rotations- bzw. Scherwinkel pro Textzeile schätzen, sind in diesen Fällen nicht praktikabel, vielmehr sind stattdessen lokale Methoden vorzuziehen.

Im folgenden Experiment wird daher das Erkennungssystem evaluiert, bei dem neben der lokalen Binarisierung außerdem die in den Abschnitten 6.5.2-6.5.4 beschriebenen lokalen Verfahren zur Schriftnormalisierung integriert wurden. Demzufolge werden in der Vorverarbeitungsphase die extrahierten Textregionen erst gemäß der horizontalen Abstände zwischen den Schriftkomponenten in Abschnitte aufgeteilt, anhand derer die Schriftnormalisierung dann jeweils separat durchgeführt wird.

Schreiber	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	400 Wörter Lexikon
1	25.5% $\pm$ 2.6	18.2% $\pm$ 2.3	12.8% $\pm$ 4.1
2	39.4% $\pm$ 4.2	27.4% $\pm$ 3.9	37.6% $\pm$ 8.9
3	47.8% $\pm$ 4.4	45.1% $\pm$ 4.4	40.8% $\pm$ 9.2
4	28.9% $\pm$ 4.8	23.6% $\pm$ 4.6	17.2% $\pm$ 8.0
5	25.1% $\pm$ 3.8	20.0% $\pm$ 3.5	31.4% $\pm$ 8.3
6	36.3% $\pm$ 4.4	31.3% $\pm$ 4.3	25.9% $\pm$ 8.2
Gesamt	32.8% $\pm$ 1.6	26.3% $\pm$ 1.5	25.8% $\pm$ 3.1

Tabelle 7.12: Ergebnisse bei Integration der lokalen Schriftnormalisierung.

Mit Hilfe dieser lokalen Schriftnormalisierung werden die in der Tabelle 7.12 dargestellten Ergebnisse erreicht. Die auf der gesamten Stichprobe gemessene Zeichenfehlerrate kann somit von 34.1% (System ohne lokale Schriftnormalisierung) auf 32.8% verbessert werden. Durch Integration des Tri-Gramm Sprachmodells verringert sich dieser Wert auf 26.3%. Bei der lexikonbasierten Erkennung ergibt sich eine Verbesserung der Wortfehlerrate auf 25.8%.

LDA Transformation der Merkmale

Im folgenden wird das videobasierte System evaluiert, bei dem ebenfalls eine lokale Vorverarbeitung durchgeführt wird, jedoch im Unterschied zum vorhergehenden Experiment die Merkmalsvektoren einer LDA-Transformation unterzogen werden. Die verwendeten Transformationsmatrizen sind dabei diejenigen, die auch im scannerbasierten System zur LDA-Transformation der Merkmale berechnet wurden (vgl. Tabelle 7.9). Die erzielten Erkennungsergebnisse des videobasierten Systems sind in der Tabelle 7.13 dargestellt. Die erste Zeile zeigt die Fehlerraten bei LDA-Transformation ohne Dimensionsreduktion (LDA 18), während in der zweiten bzw. dritten Zeile der

	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	400 Wörter Lexikon
LDA 18	31.5% ± 1.6	25.5% ± 1.5	24.5% ± 3.0
LDA 16	31.7% ± 1.6	25.8% ± 1.5	26.5% ± 3.1
LDA 12	33.6% ± 1.6	25.6% ± 1.5	30.9% ± 3.2

Tabelle 7.13: Ergebnisse des videobasierten Systems auf der gesamten Stichprobe mit LDA-Transformation der Merkmale. Erste Zeile: LDA-Transformation ohne Dimensionsreduktion. Zweite und dritte Zeile: Dimensionsreduktion auf 16 bzw. 12 Merkmalskomponenten.

Tabelle (LDA 16 bzw. LDA 12) die Ergebnisse bei einer Dimensionsreduktion auf 16 bzw. 12 Merkmalskomponenten dargestellt sind.

Durch den Vergleich der Ergebnisse aus Tabelle 7.13 mit den Resultaten aus den vorherigen Experimenten wird deutlich, dass durch LDA-Transformation der Merkmale die Fehlerraten leicht verringert werden können, wobei die Verbesserungen hier jedoch unterhalb der Signifikanzgrenze liegen. Das System ohne Dimensionsreduktion schneidet dabei durchweg besser ab als die Systeme mit Dimensionsreduktion, womit die Ergebnisse des scannerbasierten Systems aus Tabelle 7.9 bestätigt werden. Die Ergebnisse des Systems LDA 18, das bei den LDA-Experimenten am besten abschnitt, sind in der Tabelle 7.14 für die einzelnen Schreiber separat aufgeführt.

Adaption

Für die im folgenden beschriebenen Adaptionsexperimente wird von dem videobasierten System mit lokaler Normalisierung und LDA-Transformation der Merkmale ausgegangen. Ähnlich wie im online System wird die Adaption durch das MLLR-Verfahren mit einer Regressionsklasse vorgenommen, d.h. die Mittelwertvektoren aller Gaußdichten werden mit Hilfe einer gemeinsamen Transformationsmatrix adaptiert.

Schreiber	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	400 Wörter Lexikon
1	21.8% ± 2.5	16.1% ± 2.3	8.7% ± 3.6
2	34.3% ± 4.0	25.3% ± 3.8	40.0% ± 8.4
3	50.6% ± 4.4	47.4% ± 4.4	47.6% ± 9.4
4	31.4% ± 4.8	24.5% ± 4.6	8.6% ± 6.9
5	24.6% ± 3.7	19.0% ± 3.4	24.4% ± 7.6
6	36.9% ± 4.5	30.4% ± 4.3	28.4% ± 8.8
Gesamt	31.5% ± 1.6	25.5% ± 1.5	24.5% ± 3.0

Tabelle 7.14: Ergebnisse des Systems LDA 18 (LDA-Transformation der Merkmale ohne Dimensionsreduktion).

Das Adaptionsverfahren wird dabei im überwachten Modus durchgeführt, d.h. die Berechnung der Transformationsmatrix erfolgt anhand einer korrekt annotierten Stichprobe. Demzufolge wird das zur Verfügung stehende videobasierte Material in zwei Hälften aufgeteilt, sodass die eine (annotierte) Hälfte zur Berechnung der Transformationsmatrix verwendet werden kann, während die andere Hälfte als Teststichprobe dient.

	Zeichenfehlerrate		Wortfehlerrate
	ohne Sprachm.	Tri-Gramm	400 Wörter Lexikon
ohne Adaption	33.2% $\pm$ 2.2	26.4% $\pm$ 2.1	25.3% $\pm$ 4.3
MLLR-Adaption	32.1% $\pm$ 2.2	26.2% $\pm$ 2.1	26.9% $\pm$ 4.3

Tabelle 7.15: Ergebnisse der MLLR-Adaption auf 50% der videobasierten Daten. Die erste Zeile zeigt zum Vergleich die Ergebnisse des Systems ohne Adaption auf derselben Teststichprobe.

Die Ergebnisse der Adaption werden in der Tabelle 7.15 gezeigt. Zum Vergleich sind in der ersten Zeile die Ergebnisse des Systems ohne Adaption dargestellt, die auf der Hälfte des videobasierten Materials erreicht werden. Man erkennt, dass sich die Zeichenfehlerrate geringfügig verbessert hat. Die Wortfehlerrate ist dagegen leicht gestiegen. Die Veränderungen sind in beiden Fällen jedoch nicht signifikant, sodass es sich dabei auch um zufällige Schwankungen handeln kann. Dies kann als Indiz dafür interpretiert werden, dass sich die Merkmalsrepräsentationen der scanner- und videobasierten Daten nicht so sehr unterscheiden, als dass durch eine globale Transformation eine signifikante Verbesserung der Erkennungsleistung zu erreichen wäre.

7.4 Zusammenfassung der Ergebnisse

In diesem Kapitel wurden die Evaluationsergebnisse der realisierten Systeme zur videobasierten online und offline Handschrifterkennung vorgestellt. Als Kriterium zur Messung der Erkennungsleistung wurde die Fehlerrate auf Wort- bzw. Zeichenebene verwendet.

Die Ergebnisse zeigen, dass robuste und leistungsfähige videobasierte Erkennungssysteme entwickelt werden konnten. Im online Bereich ist mit dem Verfahren zur Stiftverfolgung in Videobildfolgen eine zuverlässige Extraktion der Stifttrajektorie möglich. Durch die Vorverarbeitung der Trajektorie, insbesondere durch die pen-up/down Klassifikation und das Impuls-Resampling, konnte die Fehlerrate gegenüber dem Basissystem signifikant verringert werden. Das Erkennungssystem, bei dem außerdem eine Korrektur der aufnahmebedingten geometrischen Verzerrung der Schrift und eine MLLR-Adaption der HMM-Parameter durchgeführt wird, erreicht eine Wortfehlerrate von 18.3% bei einer Lexikongröße von 201 Wörtern. Die Fehlerrate bleibt damit hinter der des tablettbasierten Referenzsystems (4.6%) zurück, nichtdestotrotz

sind die erzielten Ergebnisse in Anbetracht der anspruchsvollen Erkennungsaufgabe vielversprechend. So wurde die Erkennung schreiberunabhängig durchgeführt und den Schreibern wurden keinerlei Einschränkungen hinsichtlich des Schriftstils auferlegt.

Auch das inkrementelle videobasierte offline System zur Erkennung von Tafelanschrift weist eine gute Performanz auf. Obwohl der Schwierigkeitsgrad der Erkennungsaufgabe gegenüber dem scannerbasierten Referenzsystem in vielerlei Hinsicht erhöht ist, beispielweise durch die geringere Bildauflösung, die variierenden Beleuchtungsbedingungen, das Fehlen von Referenzlinien am Whiteboard, können vor allem durch die lokalen Verfahren zur Vorverarbeitung der Textregionen annähernd die Ergebnisse des Referenzsystems erreicht werden. Das System mit lokaler Binarisierung und lokaler Schriftnormalisierung weist eine Fehlerrate auf Zeichenebene von 32.8% auf, dies entspricht einer Reduktion der Fehlerrate von ca. 15% gegenüber dem videobasierten Basissystem mit globaler Vorverarbeitung. Im Vergleich dazu beträgt die Zeichenfehlerrate des scannerbasierten Referenzsystems 27.3%. Durch LDA-Transformation der Merkmalsrepräsentation kann die Zeichenfehlerrate des videobasierten Systems auf 31.5% reduziert werden. Wird darüberhinaus ein Tri-Gramm Sprachmodell in den Erkennungsprozess integriert, so ergibt sich die Fehlerrate auf Zeichenebene zu 25.5% (verglichen mit 19.4% des Referenzsystems).

8 Zusammenfassung

Das oberste Ziel im Bereich der Mensch-Maschine Kommunikation ist es, den Umgang mit der Maschine so natürlich wie möglich zu gestalten. Da Stift und Papier gegenüber herkömmlichen Computer-Eingabegeräten wie Tastatur oder Maus eine Reihe vorteilhafter Eigenschaften aufweisen, wird neben der Sprache und Gestik auch die Verwendung von Handschrift als Eingabemodalität intensiv untersucht. Die Vorteile liegen insbesondere darin, dass sich kurze Notizen, Skizzen oder Tabellen handschriftlich schneller und bequemer anfertigen lassen als per Tastatur oder Maus. Aufgrund der platzsparenden Schnittstelle sind stiftbasierte Systeme außerdem auch für Handheld-Computer attraktiv.

Die Signalaufnahme erfolgt bei stiftbasierten Eingabesystemen derzeit fast ausschließlich mit Hilfe spezieller Sensoren. Bei den online Systemen, die auf der Erfassung der Schreibdynamik basieren, kommen zur Signalaufnahme Digitalisiertabletts zum Einsatz, während die offline Handschrifterkennungssysteme Scanner zur Bildaufnahme verwenden. Der Einsatz dieser speziellen Sensoren läuft jedoch der Forderung nach einer natürlichen Mensch-Maschine Schnittstelle zuwider und schränkt die Anwendungsmöglichkeiten der entsprechenden Systeme ein. In dieser Arbeit wurden daher Systeme zur Handschrifterkennung entwickelt, die anstelle von Scanner bzw. Digitalisiertablett eine Videokamera zur Schrifterfassung einsetzen, sodass damit eine natürliche Eingabeschnittstelle zur Mensch-Maschine Kommunikation zur Verfügung steht. Es wurde sowohl ein online Erkennungssystem vorgestellt, das auf der Extraktion der Schreibdynamik aus der Videobildfolge basiert, als auch ein offline System, das von einer statischen, bildhaften Repräsentation der Eingabedaten ausgeht.

Das realisierte videobasierte online System erlaubt das Schreiben mit einem gewöhnlichen Stift auf normalem Papier, wobei der Kameraaufbau so gestaltet wurde, dass Schreibvorgänge an einem Schreibtisch beobachtet werden können. Die Voraussetzung für die videobasierte online Erkennung ist die Extraktion dynamischer Bewegungsinformationen anhand der aufgenommenen Bildsequenz. Dazu wird ein Verfahren zur Verfolgung von Stiftbewegungen in Videobildfolgen verwendet, das auf einem Template-Matching Ansatz basiert und an das in [Mun02] vorgeschlagene System angelehnt ist. Gegenüber diesem wurde das hier eingesetzte Verfahren vor allem im Bereich der Initialisierung modifiziert, um die erforderliche Benutzerkooperation auf ein Minimum zu reduzieren.

Die aus der videobasierten Stiftverfolgung resultierenden Trajektorien sind jedoch gegenüber tablettbasierten Daten von erheblich geringerer Qualität. Mit Hilfe des Template-Matching Verfahrens kann beispielweise nicht festgestellt werden, ob der Stift gerade aufgesetzt ist (pen-down) oder sich knapp über der Schreiboberfläche

befindet (pen-up). Zudem ist sowohl die Abtastrate (50 Hz) als auch die räumliche Auflösung (ca. 70 dpi) der Videokamera im Vergleich zu einem Digitalisiertablett (200 Hz, 2500 dpi) sehr niedrig, sodass insbesondere bei einer hohen Schreibgeschwindigkeit sprunghafte Richtungsänderungen in der Stifttrajektorie auftreten.

Die videobasierten Stifttrajektorien werden daher einer Reihe von Vorverarbeitungsschritten unterzogen, um annähernd die Qualität von Tablettdaten zu erreichen. Dazu wird für jeden Trajektorienpunkt zunächst eine pen-up/down Unterscheidung durchgeführt, wobei das Schriftbild in der Umgebung der extrahierten Stiftposition analysiert wird. Die Stiftposition wird dann als pen-down gekennzeichnet, wenn in der betrachteten Umgebung Schriftpixel zu finden sind, die der auf dem Papier zurückgelassenen Tintenspur entsprechen, andernfalls wird der Trajektorienpunkt als pen-up markiert.

Zur Glättung der aufgenommenen Stifttrajektorie wird ein local averaging Verfahren eingesetzt, mit dem eine Binomialfilterung der Stiftkoordinaten durchgeführt wird. Außerdem wird die geometrische Verzerrung der Trajektorie korrigiert, die durch das Halbbildverfahren sowie den Aufnahmewinkel der Kamera hervorgerufen wird. Um die Trajektorie in eine Repräsentation zu transformieren, die unabhängig von der Schreibgeschwindigkeit und der Abtastrate ist, wird außerdem eine Neuabtastung vorgenommen. Durch Anwendung eines Impulsterms wird dabei gleichzeitig eine weitere Glättung der Trajektorie erreicht.

Die vorverarbeiteten Schriftdaten werden anschließend in einzelne Strokes segmentiert, anhand derer dann die Merkmalsberechnung erfolgt. Zur Klassifikation der resultierenden Merkmalsvektorfolgen werden Hidden Markov Modelle (HMMs) mit semikontinuierlicher Emissionsmodellierung auf Basis Gauß'scher Mischverteilungsdichten eingesetzt. Aufgrund des geringen Umfangs der videobasierten Stichprobe werden die Modelle mit Hilfe tablettbasierter Daten trainiert. Um den daraus resultierenden Mismatch zwischen den tablettbasierten Trainingsdaten und den videobasierten Testdaten zu kompensieren, wird neben den Vorverarbeitungsmaßnahmen außerdem eine MLLR-Adaption der HMM-Parameter vorgenommen.

Anhand der Evaluationsexperimente konnte gezeigt werden, dass das realisierte System zur videobasierten online Handschrifterkennung eine gute Performanz auf der schreiberunabhängigen Teststichprobe aufweist. So wird die Stifttrajektorie sehr robust und schritthaltend mit dem Schreibprozess aus den Videobildfolgen extrahiert. Durch die Vorverarbeitungsschritte kann die Qualität der aufgenommenen Daten erheblich verbessert werden. Dabei wird insbesondere durch Integration der pen-up/down Detektion und Glättung durch Impuls-Resampling eine deutliche Verbesserung der Erkennungsleistung erzielt. Mit zusätzlicher Adaption der HMM-Parameter wird bei einer Lexikongröße von 201 Wörtern eine Wortfehlerrate von 18.3% erreicht.

Das online System basiert auf der Erfassung der Schreibdynamik anhand der Videobildfolge. Die Voraussetzung dafür ist, dass die Stiftspitze stets in den aufgenommenen Bildern sichtbar ist. Für Schreibprozesse, die an einem Schreibtisch vorgenommen werden, kann diese Vorbedingung auch ohne große Einschränkung der Benutzbarkeit erfüllt werden. Wird jedoch beispielsweise an einer Wandtafel (Whiteboard) geschrie-

ben, so ist der Stift sehr häufig vom Schreiber verdeckt, sodass das beschriebene online System dann nicht anwendbar ist.

Für das Tafelszenario wurde daher ein System zur videobasierten offline Erkennung realisiert. Das offline System ist dadurch gekennzeichnet, dass ähnlich wie bei der online Erkennung der Schreibprozess fortwährend beobachtet wird, wobei hier jedoch anstatt der Stiftkoordinaten die Bildregionen extrahiert werden, die neu geschriebenen Textabschnitten entsprechen. Die Verarbeitung erfolgt inkrementell, d.h. der Erkennungsvorgang wird angestoßen, sobald eine bisher nicht detektierte Textregion in der Bildfolge sichtbar wird. Mit dieser Verarbeitungsstrategie werden kurze Antwortzeiten erreicht, sodass damit ein interaktiver Umgang mit dem System ermöglicht wird – eine wichtige Eigenschaft benutzerfreundlicher Systeme.

Zur schnellen und robusten Detektion der Textregionen wird ein zweistufiges Verfahren eingesetzt. Im ersten Schritt wird eine Unterscheidung von Text-, Hintergrund- und Störbereichen des Bildes durchgeführt, sodass im zweiten Schritt die Schriftkomponenten zu Textregionen zusammengefasst werden können, die einzelnen Wörtern bzw. Abschnitten einzelner Textzeilen entsprechen.

Die extrahierten Textregionen werden anschließend mit Hilfe von lokal arbeitenden Verfahren vorverarbeitet. Für die Binarisierung wird die modifizierte lokale Niblack-Methode eingesetzt, da die aufgenommenen Bilder oftmals durch Schattenwurf und schwankende Beleuchtungsbedingungen eine inhomogene Grauwertverteilung aufweisen, sodass die Trennung der Schrift vom Hintergrund i.d.R. nicht mittels eines globalen Schwellwerts vorgenommen werden kann. Weitere Vorverarbeitungsmaßnahmen sind erforderlich, um die geometrische Variabilität der Schrift zu kompensieren. So kann durch das Fehlen jeglicher Referenzlinien und das ungewohnte Schreiben an der Tafel die Basislinie einer Textregion häufig nicht durch eine einzelne Gerade approximiert werden. Daher erfolgt sowohl die Bestimmung der Basislinie zur Korrektur der Schriftorientierung als auch die Normalisierung der Schriftneigung anhand lokaler Methoden.

Die zur Klassifikation verwendeten, geometrischen Merkmale werden innerhalb eines sliding windows extrahiert, das von links nach rechts über die Textregion geschoben wird. Die Segmentierung der Textregion in Buchstaben erfolgt damit erst im Zuge der Erkennung, sodass frühe unumkehrbare Segmentierungsentscheidungen vermieden werden können. Zur Optimierung der Merkmalsrepräsentation wird eine LDA-Transformation durchgeführt. Die Klassifikation der Merkmalsvektorfolgen wird dann ähnlich wie im online System auf Basis von Hidden Markov Modellen vorgenommen, die aufgrund der geringen Menge zur Verfügung stehender videobasierter Daten mit Hilfe scannerbasierten Materials trainiert wurden.

Zur Evaluation des Systems wurde eine Reihe von Experimenten durchgeführt. Dabei wurde sowohl die Wortfehlerrate bei Verwendung eines Lexikons gemessen als auch die Fehlerrate auf Zeichenebene bei lexikonfreier Erkennung. Um die Resultate der lexikonfreien Erkennung zu verbessern, wurde ein statistisches Tri-Gramm Sprachmodell in den Erkennungsprozess integriert. Die Ergebnisse der Evaluation zeigen, dass ein robustes und leistungsfähiges System zur inkrementellen videobasierten

Erkennung von Tafelanschrift realisiert werden konnte. Durch die lokale Binarisierung und Schriftnormalisierung der extrahierten Textregionen wird dabei eine deutliche Verbesserung der Erkennungsqualität erreicht. So verringert sich die Fehlerrate auf Zeichenebene von 30.8% (Basissystem mit Tri-Gramm Sprachmodell) auf 26.3% (System mit lokaler Vorverarbeitung und Tri-Gramm Sprachmodell). Auf Wortebene wird durch die lokale Vorverarbeitung eine Reduktion der Fehlerrate von 41.3% auf 25.8% bei einem 400 Wörter umfassenden Lexikon erzielt. Wird zusätzlich die Merkmalsrepräsentation LDA-transformiert, so resultiert eine Fehlerrate von 25.5% auf Zeichenebene bzw. 24.5% auf Wortebene.

Abschließend betrachtet stellt die Verwendung videobasierter Sensorik einen vielversprechenden Ansatz zur automatischen Handschrifterkennung dar. Die in dieser Arbeit realisierten Systeme weisen jeweils eine gute Erkennungsqualität auf und bieten dabei eine natürliche Eingabeschnittstelle zur Mensch-Maschine Kommunikation. Dies gilt vor allem für das System zur inkrementellen offline Erkennung von Tafelanschrift, bei dem von Seiten des Benutzers keinerlei Anpassung an das System erforderlich ist. Aufgrund der wachsenden Popularität von Whiteboards in Büro- und Besprechungsräumen ist dieses Szenario besonders interessant, sodass ein steigendes Interesse an Systemen zur Erkennung von Tafelanschrift zu erwarten ist. Der Einsatz videobasierter Sensorik ist dabei auch deshalb von Vorteil, da neben der Texterkennung auch die Möglichkeit besteht, die Gestik das Benutzers auszuwerten, sodass eine multimodale Eingabeschnittstelle realisiert werden kann. In Verbindung mit Systemen zur Spracherkennung wird damit der Grundstein für die Entwicklung “intelligenter Räume” gelegt.

Literatur

[Ari02] N. Arica, F. Yarman-Vural: *Optical Character Recognition for Cursive Handwriting*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 24, Nr. 6, 2002, S. 801–813.

[Bar88] O. Baruch: *Line Thinning by Line Following*, *Pattern Recognition Letters*, Bd. 8, 1988, S. 271–276.

[Bei95] H. S. M. Beigi, K. Nathan, J. Subrahmoia: *On-Line Unconstrained Handwriting Recognition based on Probabilistic Techniques*, in *Proc. of the Iranian Conf. on Electrical Engineering*, Tehran, Iran, May 1995.

[Ben94] Y. Bengio, Y. Le Cun: *Word Normalization for On-Line Handwritten Word Recognition*, in *Proc. Int. Conf. on Pattern Recognition*, Bd. 2 von *12*, Jerusalem, Israel, October 1994, S. 409–413.

[Ber76] N. Bernstein: *The Co-ordination and Regulation of Movements*, Pergamon Press, 1976.

[Bla98] M. J. Black, A. D. Jepson: *A Probabilistic Framework for Matching Temporal Trajectories: Condensation-based Recognition of Gestures and Expressions*, in H. Burkhardt, B. Neumann (Hrsg.): *European Conf. on Computer Vision*, Freiburg, Germany, 1998, S. 909–924.

[Bou71] H. Bouma: *Visual Recognition of Isolated Lower Case Letters*, *Vision Research*, Bd. 11, 1971, S. 495–474.

[Boz89] R. M. Bozinovic, S. N. Srihari: *Off-Line Cursive Script Word Recognition*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 11, Nr. 1, 1989, S. 68–83.

[Bra75] K. Brammer, G. Siffling: *Kalman-Bucy-Filter*, Oldenbourg Verlag, München Wien, 1975.

[Bra95] P. E. Bramall, C. A. Higgins: *A Cursive Script-Recognition System based on Human Reading Models*, *Machine Vision and Applications*, Bd. 8, Nr. 4, 1995, S. 224–231.

[Bra99] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, G. Rigoll: *Performance Evaluation of a New Hybrid Modeling Technique for Handwriting Recognition Using Identical On-Line and Off-Line Data*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Bangalore, India, September 1999, S. 446–449.

[Bro83] M. Brown, S. Ganapathy: *Preprocessing Techniques for Cursive Script Word Recognition*, *Pattern Recognition*, Bd. 16, Nr. 5, 1983, S. 447–458.

[Bro01] I. M. Bronstein, K. A. Semendjajew, G. Musiol, H. Mühlig: *Taschenbuch der Mathematik*, Verlag Harri Deutsch, Thun und Frankfurt am Main, 5. Ausg., 2001.

[Bun95] H. Bunke, M. Roth, E. G. Schukat-Talamazzini: *Off-Line Cursive Handwriting Recognition Using Hidden Markov Models*, *Pattern Recognition*, Bd. 28, Nr. 9, 1995, S. 1399–1413.

[Bun99] H. Bunke, T. von Siebenthal, T. Yamasaki, M. Schenkel: *Online Handwriting Data Acquisition Using a Video Camera*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Bangalore, 1999, S. 573–576.

[Bun03] H. Bunke: *Recognition of Cursive Roman Handwriting – Past, Present and Future*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Bd. 1, Edinburgh, Scotland, 2003, S. 448–459.

[Bus97] R. Buse, Z.-Q. Liu, T. Caelli: *A Structural and Relational Approach to Handwritten Word Recognition*, *IEEE Trans. on Systems, Man, and Cybernetics*, Bd. 27, Nr. 5, October 1997, S. 847–861.

[Cai00] J. Cai, Z.-Q. Liu: *Off-Line Unconstrained Handwritten Word Recognition*, *Int. Journal of Pattern Recognition and Artificial Intelligence*, Bd. 14, Nr. 3, 2000, S. 259–280.

[Cas96] R. G. Casey, E. Lecolinet: *A Survey of Methods and Strategies in Character Segmentation*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 18, Nr. 7, 1996, S. 690–706.

[Che94] M.-Y. Chen, A. Kundu, J. Zhou: *Off-line Handwritten Word Recognition using a Hidden Markov Model Type Stochastic Network*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 16, Nr. 5, 1994, S. 481–496.

[Cho92] C. Chouinard, R. Plamondon: *Thinning and Segmenting Handwritten Characters by Line Following*, *Machine Vision and Applications*, Bd. 5, Nr. 3, 1992, S. 185–197.

[Cho95] W. Cho, s. Lee, J. H. Kim: *Modeling and Recognition of cursive Words with Hidden Markov Models*, *Pattern Recognition*, Bd. 28, 1995, S. 1941–1953.

[Cla02] P. Clark, M. Mirmehdi: *Recognising Text in Real Scenes*, *Int. Journal on Document Analysis and Recognition*, Bd. 4, 2002, S. 243–257.

[Con00] S. D. Connell: *Online Handwriting Recognition using Multiple Pattern Class Models*, Dissertation, Michigan State University, 2000.

[Côt98] M. Côté, E. Lecolinet, M. Cheriet, C. Suen: *Automatic Reading of Cursive Scripts using a Reading Model and Perceptual Concepts - The PERCEPTO System*, *Int. Journal on Document Analysis and Recognition*, Bd. 1, 1998, S. 3–17.

[Cox95] G. S. Cox: *Template Matching and Measures of Match in Image Processing*, Technischer Bericht, Department of Electrical Engineering, University of Cape Town, July 1995.

[dB78] C. de Boor: *A Practical Guide to Splines*, Springer, 1978.

[Dem77] A. P. Dempster, N. M. Laird, D. B. Rubin: *Maximum Likelihood from Incomplete Data via the EM Algorithm*, *Journal of the Royal Statistical Society, Series B*, Bd. 39, Nr. 1, 1977, S. 1–22.

[Din00] Y. Ding, F. Kimura, Y. Miyake, M. Shridhar: *Accuracy Improvement of Slant Estimation for Handwritten Words*, in *Proc. Int. Conf. on Pattern Recognition*, Bd. 4, Barcelona, 2000, S. 527–530.

[Dol97] J. G. A. Dolfing, R. Haeb-Umbach: *Signal Representations for Hidden Markov Model based On-Line Handwriting Recognition*, in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Bd. IV, München, 1997, S. 3385–3388.

[Dun92] C. Dunn, P. S. P. Wang: *Character Segmentation Techniques for Handwritten Text – A Survey*, in *Proc. Int. Conf. on Pattern Recognition*, Bd. 2, The Hague, The Netherlands, 1992, S. 577–580.

[Ell82] A. W. Ellis: *Spelling and Writing (and Reading and Speaking)*, in A. W. Ellis (Hrsg.): *Normality and Pathology in Cognitive Function*, Kap. 4, Academic Press, 1982, S. 113–146.

[EY99] A. El-Yacoubi, M. Gilloux, R. Sabourin, C. Suen: *An HMM-based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 21, Nr. 8, 1999, S. 752–760.

[Fin99] G. A. Fink: *Developing HMM-based Recognizers with ESMERALDA*, in V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (Hrsg.): *Lecture Notes in Artificial Intelligence*, Bd. 1692, Springer, Berlin Heidelberg, 1999, S. 229–234.

[Fin01] G. A. Fink, M. Wienecke, G. Sagerer: *Video-Based On-line Handwriting Recognition*, in *Proc. Int. Conf. on Document Analysis and Recognition*, 2001, S. 226–230.

[Fin03] G. A. Fink: *Mustererkennung mit Markov-Modellen*, Leitfäden der Informatik, B. G. Teubner, Stuttgart – Leipzig – Wiesbaden, 2003.

[Fis99] A. Fischer, V. Stahl: *Database and Online Adaptation for Improved Speech Recognition in Car Environments*, in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, 1999.

[Fra95] C. Frankish, R. Hull, P. Morgan: *Recognition Accuracy and User Acceptance of Pen Interfaces*, in *Proc. Conf. on Human Factors and Computing Systems*, 1995, S. 503–510.

[Fre87] J. J. Freyd: *Dynamic Mental Representations*, *Psychological Review*, Bd. 94, 1987, S. 427–438.

[Fuk72] K. Fukunaga: *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[Gad95] P. Gader, M. Whalen, M. Ganzberger, D. Hepp: *Handprinted Word Recognition on a NIST Data Set*, *Machine Vision and Applications*, Bd. 8, Nr. 1, 1995, S. 31–40.

[Gau92] L. Gauvain, C.-H. Lee: *MAP Estimation of Continuous Density HMM: Theory and Applications*, in *Proc. DARPA Speech & Nat. Lang.*, Morgan Kaufmann, Los Altos, CA, 1992.

[Gib78] E. J. Gibson, H. Levin: *The Psychology of Reading*, MIT Press, Cambridge, Massachusetts, 1978.

[Gon91] R. Gonzalez, P. Wintz: *Digital Image Processing*, Addison-Wesley, 1991.

[Gre94] P. J. Green, B. W. Silverman: *Nonparametric Regression and Generalized Linear Models*, Nr. 58 in Monographs on Statistics and Applied Probability, Chapman & Hall, London, 1994.

[Gue93] W. Guerfali, R. Plamondon: *Normalizing and Restoring On-Line Handwriting*, *Pattern Recognition*, Bd. 26, Nr. 3, 1993, S. 419–431.

[Gue98] W. Guerfali, R. Plamondon: *A New Method for the Analysis of Simple and Complex Planar Rapid Movements*, *Journal of Neuroscience Methods*, Bd. 82, Nr. 1, 1998, S. 35–45.

[Guy94] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet: *UNIPEN Project of On-line Data Exchange and Recognizer Benchmarks*, in *Proc. Int. Conf. on Pattern Recognition*, Bd. 2, Jerusalem, Israel, 1994, S. 29–33.

[Guy97] I. Guyon, R. M. Haralick, J. J. Hull, I. T. Phillips: *Data Sets for OCR and Document Image Understanding Research*, in H. Bunke, P. S. P. Wang (Hrsg.): *Handbook of Character Recognition and Document Image Analysis*, World Scientific, 1997, S. 779–799.

[Haf92] P. Haffner, A. Waibel: *Multi-State Time Delay Neural Networks for Continuous Speech Recognition*, in J. Moody, S. Hanson, R. Lipmann (Hrsg.): *Advances in Neural Information Processing Systems*, Bd. 4, Morgan Kaufmann, San Mateo, CA, 1992, S. 135–143.

[Har92] R. M. Haralick, L. G. Shapiro: *Computer and Robot Vision*, Bd. 1, Addison-Wesley, 1992.

[Hol81] J. M. Hollerbach: *An Oscillation Theory of Handwriting, Biological Cybernetics*, Bd. 39, 1981, S. 139–156.

[Hu96] J. Hu, M. K. Brown, W. Turin: *HMM based On-line Handwriting Recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 18, Nr. 10, 1996, S. 1039–1044.

[Hu97] J. Hu, A. S. Rosenthal, M. K. Brown: *Combining High-Level Features with Sequential Local Features for On-Line Handwriting Recognition*, in *Proc. Int. Conf. on Image Analysis and Processing*, Florence, Italy, September 1997, S. 647–654.

[Hu00] J. Hu, S. G. Lim, M. Brown: *Writer Independent On-line Handwriting Recognition Using an HMM Approach, Pattern Recognition*, Bd. 33, Nr. 1, 2000, S. 133–147.

[Jae01] S. Jaeger, S. Manke, J. Reichert, A. Waibel: *Online Handwriting Recognition: The NPen++ Recognizer*, *Int. Journal on Document Analysis and Recognition*, Bd. 3, 2001, S. 169–180.

[Jäh97] B. Jähne: *Digitale Bildverarbeitung*, Springer, 1997.

[Jai00] A. K. Jain, R. P. W. Duin, J. Mao: *Statistical Pattern Recognition: A Review*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 22, Nr. 1, 2000, S. 4–37.

[Joh78] S. Johannson, G. N. Leech, H. Goodluck: *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers*, Department of English, University of Oslo, Oslo, 1978.

[Kal93] A. Kaltenmeier, C. Fritz, P. Regel-Brietzmann, T. Caesar, J. Gloger, E. Mandler: *Hidden Markov Models – A Unified Approach to Recognition of Spoken and Written Language*, in S. J. Pöppl, H. Handels (Hrsg.):

Mustererkennung 1993, 15. DAGM Symposium, Lübeck, Informatik aktuell, Springer, Berlin, 1993, S. 191–198.

[Kav02] E. Kavallieratou, N. Fakotakis, G. Kokkinakis: *An Unconstrained Handwriting Recognition System*, *Int. Journal on Document Analysis and Recognition*, Bd. 4, Nr. 4, 2002, S. 226–242.

[Kim97] G. Kim, V. Govindaraju: *A Lexicon Driven Approach to Handwritten Word Recognition for Real-Time Applications*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 19, Nr. 4, 1997, S. 366–379.

[Kim98] G. Kim, V. Govindaraju: *Handwritten Phrase Recognition as Applied to Street Name Images*, *Pattern Recognition*, Bd. 31, Nr. 1, 1998, S. 42–51.

[Kim99] G. Kim, V. Govindaraju, S. N. Srihari: *An Architecture for Handwritten Text Recognition Systems*, *Int. Journal on Document Analysis and Recognition*, Bd. 2, 1999, S. 37–44.

[Kle92] R. Klette, P. Zamperoni: *Handbuch der Operatoren für die Bildbearbeitung: Bildtransformationen für die digitale Bildverarbeitung*, Vieweg, 1992.

[Koh97] M. Kohler: *Using the Kalman Filter to Track Human Interactive Motion – Modelling and Initialization of the Kalman Filter for Translational Motion*, Technischer Bericht 629, Informatik VII, University of Dortmund, January 1997.

[Kos97] A. Kosmala, J. Rottland, G. Rigoll: *Large Vocabulary On-Line Handwriting Recognition with Context Dependent Hidden Markov Models*, in *Mustererkennung 97, 19. DAGM-Symposium Braunschweig*, Informatik aktuell, Springer, 1997, S. 254–261.

[Lan90] K. Lang, A. Waibel: *A Time-Delay Neural Network Architecture for Isolated Word Recognition*, *Neural Networks*, Bd. 3, 1990, S. 23–43.

[Lee96] S.-W. Lee, D.-J. Lee, H.-S. Park: *A New Methodology for Gray-Scale Character Segmentation and Recognition*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 18, Nr. 10, 1996, S. 1045–1050.

[Leg95] C. J. Leggetter, P. C. Woodland: *Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models*, *Computer Speech & Language*, Bd. 9, 1995, S. 171–185.

[Li00] H. Li, D. Doermann, O. Kia: *Automatic Text Detection and Tracking in Digital Video*, *IEEE Transactions on Image Processing*, Bd. 9, Nr. 1, 2000, S. 147–156.

[Mad99a] S. Madhvanath, V. Govindaraju: *Local Reference Lines for Handwritten Phrase Recognition*, *Pattern Recognition*, Bd. 32, 1999, S. 2021–2028.

[Mad99b] S. Madhvanath, G. Kim, V. Govindaraju: *Chaincode Contour Processing for Handwritten Word Recognition*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 21, Nr. 9, September 1999, S. 928–932.

[Mak94] J. Makhoul, T. Starner, R. Schwartz, G. Chou: *On-Line Cursive Handwriting Recognition Using Hidden Markov Models and Statistical Grammars*, in *Human Language Technology*, Morgan Kaufmann, 1994, S. 432–436.

[Man94] S. Manke, M. Finke, A. Waibel: *Combining Bitmaps with Dynamic Writing Information for On-Line Handwriting Recognition*, in *Proc. Int. Conf. on Pattern Recognition*, Jerusalem, 1994, S. 596–598.

[Mar98] U.-V. Marti, H. Bunke: *Erkennung handgeschriebener Wortsequenzen*, in P. Levi, R.-J. Ahlers, F. May, M. Schanz (Hrsg.): *Mustererkennung 98, 20. DAGM-Symposium Stuttgart, Informatik aktuell*, Springer-Verlag, Berlin, 1998, S. 263–270.

[Mar99] U.-V. Marti, H. Bunke: *A Full English Sentence Database for Off-line Handwriting Recognition*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Bangalore, 1999, S. 705–708.

[Mar00a] U.-V. Marti: *Offline Erkennung handgeschriebener Texte*, Dissertation, University of Bern, Bern, Switzerland, Nov. 2000.

[Mar00b] U.-V. Marti, H. Bunke: *Handwritten Sentence Recognition*, in *Proc. Int. Conf. on Pattern Recognition*, Bd. 3, Barcelona, 2000, S. 467–470.

[Mar02] U.-V. Marti, H. Bunke: *The IAM-Database: An English Sentence Database for Offline Handwriting Recognition*, *Int. Journal on Document Analysis and Recognition*, Bd. 5, 2002, S. 39–46.

[McC81] J. L. McClelland, D. E. Rumelhart: *An Interactive Activation Model of Context Effects in Letter Perception: Part 1. An Account of Basic Findings*, *Psychological Review*, Bd. 88, Nr. 5, 1981, S. 375–407.

[Mir01] M. Mirmehdi, P. Clark, J. Lam: *Extracting Low Resolution Text with an Actvie Camera for OCR*, in J. Sanchez, F. Pla (Hrsg.): *Proc. 9th Spanish Symposium on Pattern Recognition and Image Processing*, 2001, S. 43–48.

[Mor69] J. Morton: *Interaction of Information in Word Recognition*, *Psychological Review*, Bd. 76, 1969, S. 165–178.

[Mor82] P. Morasso, F. A. Mussa Ivaldi: *Trajectory Formation and Handwriting: A Computational Model*, *Biological Cybernetics*, Bd. 45, 1982, S. 131–142.

[Mun96] M. E. Munich, P. Perona: *Visual Input for Pen-Based Computers*, in *Proc. Int. Conf. on Pattern Recognition*, Bd. 3, Vienna, Austria, 1996, S. 33–37.

[Mun00] M. E. Munich: *Visual Input for Pen-based Computers*, Dissertation, California Institute of Technology, Pasadena, California, Jan. 2000.

[Mun02] M. E. Munich: *Visual Input for Pen-based Computers*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 24, Nr. 3, March 2002, S. 313–328.

[Nat93] K. A. Nathan, J. R. Bellegarda, D. Nahamoo, E. J. Bellegarda: *On-Line Handwriting Recognition Using Continuous Parameter Hidden Markov Models*, in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Bd. 5, Minneapolis, 1993, S. 121–124.

[Nib86] W. Niblack: *An Introduction to Digital Image Processing*, Prentice-Hall, 1986.

[Opf94] G. Opfer: *Numerische Mathematik für Anfänger*, Vieweg, 1994.

[Ots79] N. Otsu: *A Threshold Selection Method from Gray-Level Histograms*, *IEEE Trans. on Systems, Man, and Cybernetics*, Bd. 9, 1979, S. 62–66.

[Paa82] K. R. Paap, S. L. Newsome, J. E. McDonald, R. W. Schvaneveldt: *An Activation-Verification Model for Letter and Word Recognition: The Word Superiority Effect.*, *Psychological Review*, Bd. 89, 1982, S. 573–594.

[Par02] J. Park, V. Govindaraju: *Use of Adaptive Segmentation in Handwritten Phrase Recognition*, *Pattern Recognition*, Bd. 35, 2002, S. 245–252.

[Pau95] E. Paulus, M. Lehning: *Die Evaluierung von Spracherkennungssystemen in Deutschland*, Technischer Bericht, Verbmobil-Report 70, TU Braunschweig, Juli 1995.

[Pav90] T. Pavlidis: *Algorithmen zur Grafik und Bildverarbeitung*, Heise, 1990.

[Pla89] R. Plamondon, F. J. Maarse: *An Evaluation of Motor Models of Handwriting*, *IEEE Trans. on Systems, Man, and Cybernetics*, Bd. 19, Nr. 5, 1989, S. 1060–1072.

[Pla93] R. Plamondon, A. M. Alimi, P. Yergeau, F. Leclerc: *Modeling Velocity Profiles of Rapid Movements: A Comparative Study*, *Biological Cybernetics*, Bd. 69, 1993, S. 119–128.

[Pla95] R. Plamondon: *A Renaissance for Handwriting*, *Machine Vision and Applications*, Bd. 8, 1995, S. 195–196.

[Pla98a] R. Plamondon: *Handwriting Analysis*, in G. Varile, A. Zampolli (Hrsg.): *Survey of the State of the Art in Human Language Technology*, Kap. 2.6, Cambridge Univ. Press, March 1998, S. 96–100.

[Pla98b] R. Plamondon, W. Guerfali: *The Generation of Handwriting with Delta-Lognormal Synergies, Biological Cybernetics*, Bd. 78, 1998, S. 119–132.

[Pla00] R. Plamondon, S. N. Srihari: *On-line and Off-Line Handwriting Recognition: A Comprehensive Survey, IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 22, Nr. 1, 2000, S. 63–83.

[Ray89] K. Rayner, A. Pollatsek: *The Psychology of Reading*, Prentice-Hall, 1989.

[Rei67] C. H. Reinsch: *Smoothing by Spline Functions*, *Numerische Mathematik*, Bd. 10, 1967, S. 177–183.

[Rig98] G. Rigoll, A. Kosmala, D. Willett: *An Investigation of Context-Dependent and Hybrid Modeling Techniques for very Large Vocabulary On-Line Cursive Handwriting Recognition*, in *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, Taejon, Korea, 1998, S. 429–438.

[Sau99] E. Saund: *Bringing the Marks on a Whiteboard to Electronic Life*, in *Proc. 2nd Int. Workshop on Cooperative Buildings, CoBuild'99*, Springer, Pittsburgh, 1999, S. 69–78.

[Say73] K. M. Sayre: *Machine Recognition of Handwritten Words: A Project Report, Pattern Recognition*, Bd. 5, Nr. 3, 1973, S. 213–228.

[Sch90] L. R. B. Schomaker, H.-L. Teulings: *A Handwriting Recognition System Based on Properties of the Human Motor System*, in *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, Montreal, Canada, 1990, S. 195–211.

[Sch93] L. Schomaker: *Using Stroke- or Character-based Self-Organizing Maps in the Recognition of On-line, Connected Cursive Script, Pattern Recognition*, Bd. 26, Nr. 3, 1993, S. 443–450.

[Sch95a] M. Schenkel, I. Guyon, D. Henderson: *On-line Cursive Script Recognition Using Time-Delay Neural Networks and Hidden Markov Models, Machine Vision and Applications*, Bd. 8, Nr. 4, 1995, S. 215–223.

[Sch95b] E. G. Schukat-Talamazzini: *Automatische Spracherkennung*, Vieweg, Wiesbaden, 1995.

[Sch96] J. Schürmann: *Pattern Classification*, John Wiley, New York, 1996.

[Sch97] M. Schüßler, H. Niemann: *Die Verwendung von Kontextmodellen bei der Erkennung handgeschriebener Wörter*, in E. Paulus, F. Wahl (Hrsg.): *Mustererkennung 1997, 19. DAGM-Symposium Braunschweig*, 1997, S. 262–269.

[Sch99] L. Schomaker, E. Segers: *Finding Features used in the Human Reading of Cursive Handwriting*, *Int. Journal on Document Analysis and Recognition*, Bd. 2, 1999, S. 13–18.

[Sch00] R. Schlittgen: *Einführung in die Statistik*, Oldenbourg, München Wien, 9. Ausg., 2000.

[Sel02] A. J. Sellen, R. H. R. Harper: *The Myth of the Paperless Office*, The MIT Press, Cambridge, MA, 2002.

[Sen92] A. W. Senior: *Off-line Handwriting Recognition: A Review and Experiments*, Technischer Bericht, Cambridge University Engineering Department, 1992.

[Sen94a] G. Seni, E. Cohen: *External Word Segmentation of Off-line Handwritten Text Lines*, *Pattern Recognition*, Bd. 27, Nr. 1, 1994, S. 41–52.

[Sen94b] G. Seni, N. Nasrabadi, R. Srihari: *An On-Line Cursive Word Recognition System*, in *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, Seattle, 1994, S. 404–410.

[Sen98] A. W. Senior, A. J. Robinson: *An Off-line Cursive Handwriting Recognition System*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 20, Nr. 3, 1998, S. 309–321.

[Sen99] G. Seni, J. Seybold: *Diacritical Processing for Unconstrained Online Handwriting Recognition Using a Forward Search*, *Int. Journal on Document Analysis and Recognition*, Bd. 2, 1999, S. 24–29.

[SF96] Q. Stafford-Fraser, P. Robinson: *BrightBoard: A Video-Augmented Environment*, in *Proc. Conf. on Human Factors and Computing Systems*, Vancouver, BC, Canada, 1996, S. 134–141.

[Sta94] T. Starner, J. Makhoul, R. Schwartz, G. Chou: *On-Line Cursive Handwriting Recognition Using Speech Recognition Methods*, in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Bd. 5, Adelaide, 1994, S. 125–128.

[Ste99] T. Steinherz, E. Rivlin, N. Intrator: *Offline Cursive Script Word Recognition – A Survey*, *Int. Journal on Document Analysis and Recognition*, Bd. 2, Nr. 2, 1999, S. 90–110.

[Tap82] C. C. Tappert: *Cursive Script Recognition by Elastic Matching*, *IBM J. Res. Develop.*, Bd. 26, Nr. 6, 1982, S. 765–771.

[Tap90] C. C. Tappert, C. Y. Suen, T. Wakahara: *The State of the Art in On-line Handwriting Recognition*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 12, Nr. 8, 1990, S. 787–808.

[Teu94] H.-L. Teulings: *Invariant Handwriting Features Useful in Cursive Script Recognition*, in S. Impedovo (Hrsg.): *Fundamentals in Handwriting Recognition*, Computer and Systems Sciences, Kap. 3, Springer, 1994, S. 179–198.

[VG99] C. Viard-Gaudin, P. M. Lallican, P. Binter: *The IRESTE On/Off (IRONOFF) Dual Handwriting Database*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Bangalore, India, September 1999, S. 455–458.

[Vin00] A. Vinciarelli, J. Luettin: *Off-Line Cursive Script Recognition Based on Continuous Density HMM*, in *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, Amsterdam, 2000, S. 493–498.

[Vin02a] A. Vinciarelli: *A Survey on Off-Line Cursive Word Recognition*, *Pattern Recognition*, Bd. 35, Nr. 07, 2002, S. 1433–1446.

[Vin02b] A. Vinciarelli, S. Bengio: *Writer adaptation techniques in HMM based Off-Line Cursive Script Recognition*, *Pattern Recognition Letters*, Bd. 23, 2002, S. 905–916.

[vS98] T. von Siebenthal: *Online-Erfassung von Handschrift mit einer Videokamera*, Diplomarbeit, Institut für Informatik und angewandte Mathematik, Universität Bern, 1998.

[Wie01] M. Wienecke, G. A. Fink, G. Sagerer: *A Handwriting Recognition System Based on Visual Input*, in *2nd International Workshop on Computer Vision Systems*, Vancouver, Canada, 2001, S. 63–72.

[Wie02] M. Wienecke, G. A. Fink, G. Sagerer: *Experiments in Unconstrained Offline Handwritten Text Recognition*, in *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, Ontario, Canada, August 2002.

[Wie03] M. Wienecke, G. A. Fink, G. Sagerer: *Towards Automatic Video-based Whiteboard Reading*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Edinburgh, Scotland, 2003, S. 87–91.

[Win98] B. Winkler: *Bootstrap-Methoden bei nichtparametrischer Regression*, Dissertation, Ludwig-Maximilians-Universität München, 1998.

[Yan98] B. Yanikoglu, P. Sandon: *Segmentation of Off-Line Cursive Handwriting Using Linear Programming*, *Pattern Recognition*, Bd. 31, Nr. 12, 1998, S. 1825–1833.

[Zel97] A. Zell: *Simulation neuronaler Netze*, Oldenbourg, München, 1997.

[Zha01a] Z. Zhang, C. Tan: *Recovery of Distorted Document Images from Bound Volumes*, in *Proc. Int. Conf. on Document Analysis and Recognition*, Seattle, September 2001, S. 429–433.

[Zha01b] Z. Zhang, C. Tan: *Restoration of Images Scanned from Thick Bound Documents*, in *Proc. Int. Conf. on Image Processing*, Thessaloniki, Greece, October 2001, S. 1074–1077.

[Zim82] A. Zimmer: *Do we see what makes our script so characteristic or do we only feel it? Modes of Sensory Control in Handwriting*, *Psychological Research*, Bd. 44, 1982, S. 165–174.