

# Methoden der Merkmalsgenerierung bei Offline Handschriften

*Studienarbeit*

Autor

Thomas Burkhardt

Betreuer

Stefan Manke

Institut für Logik, Komplexität  
und Deduktionssysteme

April 1997

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation</b> .....	<b>2</b>
<b>2</b>	<b>Übersicht</b> .....	<b>3</b>
<b>3</b>	<b>Vorverarbeitung</b> .....	<b>4</b>
3.1	Binary Connected Components (BCCs) .....	4
3.2	Basisliniennormierung .....	5
3.2.1	Bestimmung der Schreiblinien .....	6
3.2.2	Transformation .....	8
3.3	Scherwinkelnormierung .....	9
3.3.1	Detektion des Scherwinkels .....	9
3.3.2	Normierung des Scherwinkels .....	13
3.4	Skelettierung .....	14
3.5	Abstandnormalisierung .....	16
3.6	Approximation .....	18
<b>4</b>	<b>Merkmalsgenerierung</b> .....	<b>20</b>
4.1	Lokal-statistische Merkmale .....	20
4.2	Topologische Merkmale .....	21
4.3	Geometrisch-statistische Merkmale .....	22
<b>5</b>	<b>Experimente</b> .....	<b>27</b>
5.1	Rahmenbedingungen .....	27
5.1.1	Trainings- und Testdaten .....	27
5.1.2	Der Erkenner .....	27
5.2	Vorversuch .....	28
5.3	Versuche .....	28
5.4	Bewertung und Ausblick .....	29
<b>6</b>	<b>Zusammenfassung</b> .....	<b>31</b>
<b>7</b>	<b>Literatur</b> .....	<b>32</b>

# 1 Einleitung und Motivation

Seit der Erfindung des Computers ist der Mensch daran interessiert, die menschlichen kognitiven Fähigkeiten nachzubilden. Besonders hervorzuheben ist das Gebiet der maschinellen Erkennung von Schrift und Sprache, das durch neue Konzepte und nicht zuletzt durch immer leistungsfähiger werdende Computer ständig weiterentwickelt wurde.

Bei der Handschriftenerkennung unterscheidet man zwischen Online- und Offline-Erkennung. Die Eingabe der Handschrift für die Online-Erkennung erfolgt durch ein Eingabetablett. Hierbei wird während des Schreibens die Koordinatensequenz des Stiftes gespeichert. Als Eingabemedium der Offline-Erkennung dient ein Scanner, der das schon geschriebene Wort als Bitmap speichert. Während nun die Offline-Eingabe als örtliches Signal vorliegt, verfügt die Online-Erkennung zusätzlich noch über die zeitliche Komponente, die sogenannten dynamischen Schreibinformationen. Somit ist eine Online-Eingabe problemlos in eine Offline-Eingabe konvertierbar. Bedingt durch den Verlust der dynamischen Schreibinformationen bei dieser Umwandlung ist eine Umkehrung dieser Abbildung nur schwer und dann oftmals sehr ungenau möglich [Jäger96].

Durch diesen gravierenden Unterschied zwischen Online- und Offline-Eingabe ist die Online-Handschriftenerkennung der Offline-Handschriftenerkennung meist überlegen. Ein Vergleich wurde bereits in [SSE96] durchgeführt. Inzwischen gibt es schreiberunabhängige Online-Handschriftenerkennung, die bei einem Wörterbuch mit 20000 Einträgen sowohl Schreib- als auch Druckschrift mit einer Erkennungsrate von 91.4% erkennen [MFW95].

Für die Offline-Handschriftenerkennung wurden bisher nur wenige konkrete Ergebnisse veröffentlicht, da es sich hierbei noch um intensive Forschungsarbeit handelt.

Wegen der geringeren Erkennungsleistung von Offline-Erkennungssystemen sind praktische Anwendungsgebiete, wie z.B. die Erkennung von handgeschriebenen Faxen oder Briefen, noch wenig sinnvoll. Ein Einsatzgebiet ist jedoch die Verbesserung der Online-Erkennung [SSE96]. Durch Umwandlung der Online- in Offline-Handschriften kann man parallel zu dem Online-einen Offline-Erkennung die Eingabe erkennen lassen. Aus den Ergebnissen kann man dann das am wahrscheinlichsten geschriebene Wort auswählen.

Ziel dieser Arbeit war es, einen schreiberunabhängigen Offline-Einzelworterkennung zu entwickeln, der als Eingabe Online-Handschriften erhält. Hierzu mußten eine Vorverarbeitung und Methoden der Merkmalsgenerierung für Offline-Handschriften erarbeitet werden. Für die durchgeführten Experimente wurden Online-Handschriften mit natürlicher Druckschrift vom MIT benutzt.

Dieses Offline-Handschriftenerkennungssystem kann dann mit dem schreiberunabhängigen Online-Handschriftenerkennung NPen<sup>++</sup> [MFW95] vom Institut für Logik, Komplexität und Deduktionssysteme der Universität Karlsruhe (TH) in oben beschriebener Weise kombiniert werden.

## 2 Übersicht



Wie bereits erwähnt erhält das von mir entwickelte Offline-Handschriftenerkennungssystem als Eingabe Online-Handschriften. Diese werden jedoch im ersten Schritt der Vorverarbeitung in Offline-Handschriften konvertiert, wobei die dynamischen Schreibinformationen verloren gehen. Die daraus resultierende Bitmap ist jedoch für die Vorverarbeitung ungeeignet, da sehr viel redundante Hintergrundinformation gespeichert wird. Für die Vorverarbeitung sollte jedoch die eigentliche Handschrift innerhalb der Bitmap bekannt sein. Deshalb werden mit dem Datentyp der Binary Connected Components (BCCs) nur die Konturen der Handschrift gespeichert. Alle folgenden Vorverarbeitungsschritte, wie Basisliniennormierung, Scherwinkelnormierung, Skelettierung, Abstandnormalisierung und Approximation, basieren auf den erzeugten BCCs.

Für das vorverarbeitete Wort werden nun anhand zwei Verfahren Merkmalsvektoren generiert: die topologische und geometrisch-statistische Methode. Der Erkenner, ein Multi State - Time Delay Neural Network (MS-TDNN), klassifiziert die generierten Merkmale und sucht anhand eines Wörterbuchs das wahrscheinlichste Wort heraus.

Im Kapitel 3 wird die Vorverarbeitung der Offline-Handschriften erläutert. Diese wurde an Anlehnung bereits existierender Systeme, wie in [CGM93], entwickelt. In der Literatur bekannte Konzepte wie die BCCs, Scherwinkelnormierung und Skelettierung wurden hierzu umgesetzt. Es wurden auch neue Methoden für die Offline-Vorverarbeitung für einige Normierungsschritte entwickelt bzw. weiterentwickelt, wie die Basisliniennormierung anhand eines Algorithmus von Bengio und LeChun [BeCh94], die Abstandnormalisierung und die spezielle Approximation für Handschriften.

In Kapitel 4 werden zwei Methoden der Merkmalsgenerierung vorgestellt. Dabei wurde eine topologische Methode und eine geometrisch-statistische Methode (in Anlehnung an [CGM93]) entwickelt.

Die in Kapitel 5 durchgeführten Experimente wurden mit Online-Daten vom MIT durchgeführt, wobei ein Wörterbuch mit 50000 Einträgen und ein Wörterbuch mit 25000 Einträgen benutzt wurde.

### 3 Vorverarbeitung

Ziel der Vorverarbeitung ist es, eine beliebige Handschrift nach bestimmten Kriterien zu normalisieren, so daß redundante Informationen entfernt werden und sich wichtige Merkmale herauskristallisieren und erhalten bleiben. So wäre es wünschenswert, wenn auffällige schreiberabhängige Variationen einer beliebigen Handschrift (wie Schreiblinien, Scherwinkel, Schriftstärke und Abstand zwischen den Buchstaben) normiert werden, damit der Erkenner von dieser Norm ausgehen kann. Oft benötigen die Normierungsschritte eine genaue Analyse des Wortes und damit werden interessante Merkmale der Handschrift gewonnen. Bei der Basisliniennormierung wäre dies die mittlere Schreibhöhe und bei der Skelettierung z.B. Schnitt- und Endpunkte, die durch die Approximation noch konkreter hervorgehoben werden.

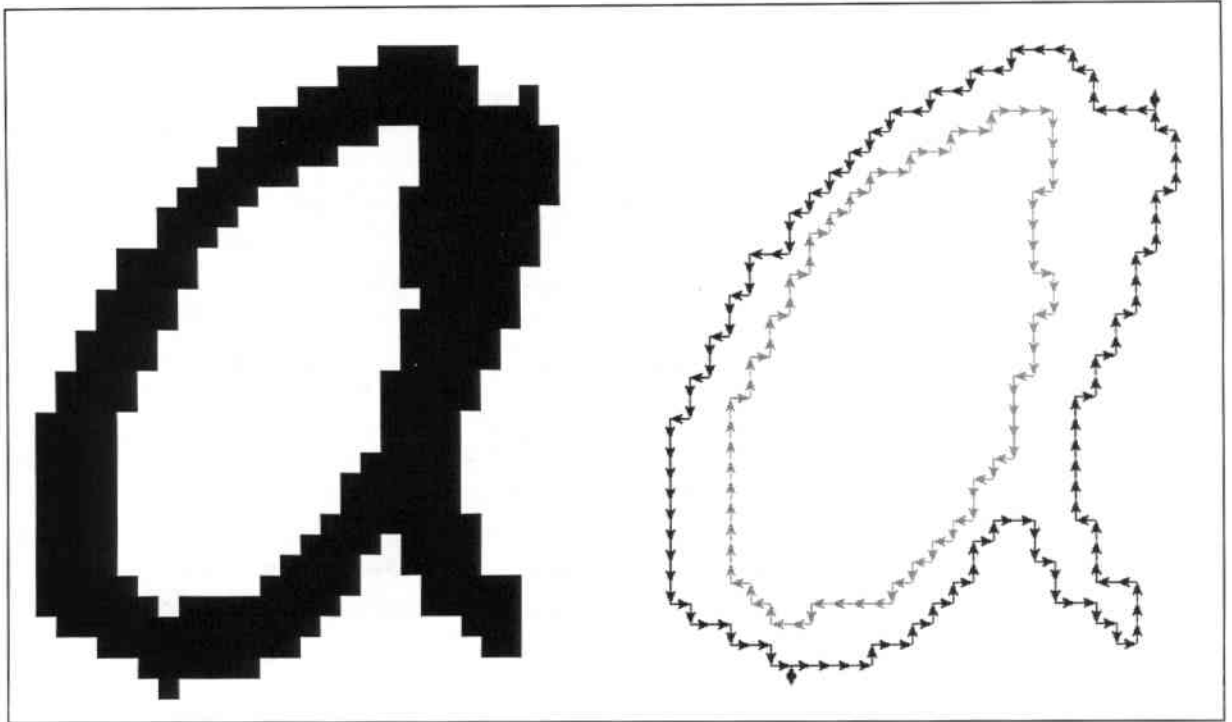
#### 3.1 Binary Connected Components (BCCs)

Im folgenden wird eine Handschrift, als Bitmap gespeichert, betrachtet, dabei gehen wir davon aus, daß ein Hintergrundpixel die Farbe weiß und ein Vordergrundpixel bzw. ein Pixel der Schrift die Farbe schwarz besitzt.

Da Bitmaps für die Vorverarbeitung der Handschrift aus oben genannten Gründen (siehe Kapitel 2) ungeeignet sind, ist es sinnvoll nur die Vordergrundpixel in der Merkmalsanalyse zu betrachten. Hierzu werden alle Gebiete bestimmt, die nur aus Vordergrundpixel bestehen und zusammenhängend sind. So ist es möglich, daß ein ganzes Wort sowohl aus nur einem Zusammenhangsgebiet als auch aus einer Vielzahl von Zusammenhangsgebieten besteht, dies hängt vom Schreibstil des Schreibers ab. Trotz der Vielfältigkeit ist es sehr wahrscheinlich, daß jedes Zusammenhangsgebiet mindestens einen fast vollständigen Buchstaben der Schrift repräsentiert.

Eine effiziente Art der Beschreibung von Zusammenhangsgebieten sind die sogenannten Binary Connected Components. Ein Zusammenhangsgebiet kann vollständig durch seine Vordergrund-Hintergrund-Übergänge beschrieben werden. Repräsentiert man die Pixel der Übergänge als Knoten eines Graphen, so erhält man gerichtete zweifach-zusammenhängende Graphen. Die Kanten des Graphen repräsentieren die Verbindung zwischen zwei Pixel. Einen solchen Graphen nennt man auch Kontur des Zusammenhangsgebietes. Man unterscheidet zwischen inneren und äußeren Konturen. Innere Konturen entstehen durch innere Löcher, die durch die Schrift, meist durch Schleifen, erzeugt werden. So enthalten die Zusammenhangsgebiete „B“ bzw. „b“ zwei bzw. eine innere Kontur. Eine äußere Kontur umschreibt den gesamten äußeren Rand eines Zusammenhangsgebietes. Somit erhält man eine Vater-Sohn-Beziehung zwischen den äußeren und inneren Konturen.

Abb. 1 soll die Korrespondenz zwischen Bitmap und BCC illustrieren.



**Abb. 1:** links) Bitmap, rechts) BCC

Eine BCC wird durch die adjazenten Pixel der Kontur beschrieben. Zwei adjazente Punkte eines BCC müssen direkt benachbart sein, d.h. ihr Abstand beträgt horizontal oder vertikal eins. Somit sind horizontale, vertikale und diagonale Verbindungen erlaubt. Des Weiteren sind BCCs orientiert. Betrachtet man die Verbindung zweier adjazenter Punkte als Vektor, so muß links von diesem Vektor das Innere des zugehörigen Zusammenhangsgebiets liegen.

Durch die BCCs als Beschreibungsmittel der Schrift erhält man die Zusammenhangsinformationen der Schrift. Es dient im folgenden allen Normierungsschritten als Grundlage. Die aufwendige Repräsentation durch ein Rasterbild wird nicht mehr benötigt, da durch die BCCs das ursprüngliche Rasterbild verlustfrei rekonstruiert werden kann.

### **3.2 Basisliniennormierung**

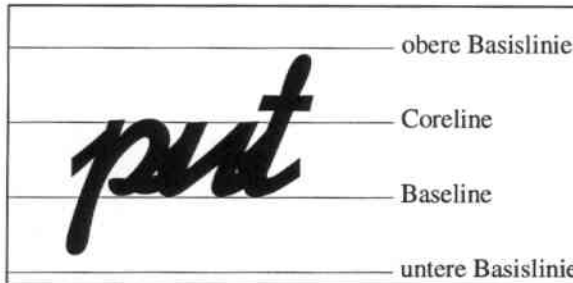
Natürliche Handschrift besteht aus einer Vielzahl von Variationen. So gibt es abhängig vom Schreiber gravierende Unterschiede in der Schreibhöhe, z.B. sind Kleinbuchstaben unverhältnismäßig klein im Vergleich zu Großbuchstaben. Auch die Lage der Schrift kann nicht als konstant angenommen werden. Besonders bei Einzelworterkennern kommt es oftmals vor, daß die Schrift nicht mehr horizontal geschrieben wurde.

Aus Untersuchungen [BeCh94] im Bereich der Online-Handschriftenerkennung wurde gezeigt, daß die Erkennungsleistung steigt, wenn man Schriften anhand ihrer Basislinien normalisiert und Merkmale relativ zu den Basislinien generiert werden.

Ziel der Basisliniennormalisierung ist es, eine Schrift so zu normalisieren, daß die Lage der Schrift konstant ist, d.h. in unserem Fall horizontal ist.

Die Basisliniennormalisierung besteht aus zwei Schritten:

1. Bestimmung der Schreiblinien (obere Basislinie, Coreline, Baseline, untere Basislinie)
2. Transformation der Schrift in ein Modell konstanter Lage (siehe Abb. 2)



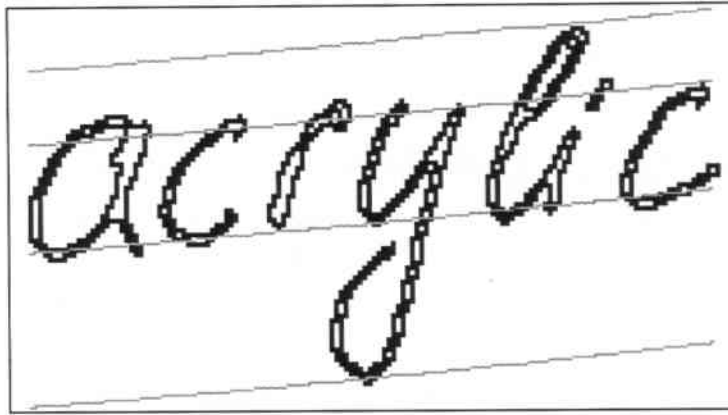
**Abb. 2: Schreiblinienmodell**

### 3.2.1 Bestimmung der Schreiblinien

Hierzu wird ein Verfahren von Yoshua Bengio und Yann Le Cun verwendet [BeCh94], das bereits in der Online-Handschriftenerkennung von NPen<sup>++</sup> erfolgreich eingesetzt wurde.

Das Verfahren basiert auf einem geometrischen Modell, das die Lage der Schrift durch die vier Schreiblinien beschreibt. Um das Modell der Schrift anzupassen, wird die Methode der „elastischen Anpassung“ benutzt. Bildlich kann man sich die vier Schreiblinien als parallele Parabeln vorstellen, die durch Federn unterschiedlicher Federkonstanten und Länge miteinander verbunden sind. Hierzu wird eine Energiefunktion aufgestellt die aus zwei Termen besteht. Der erste Term beschreibt die Qualität des Modells, d.h. wie gut das Modell zu der Schrift paßt. Dies wird anhand der Entfernung der Schreiblinien zu den lokalen Maxima und Minima der Schrift gemessen. Der zweite Term repräsentiert die Spannung innerhalb des Modells, d.h. wie sehr die Schreiblinien „gespannt“ werden müssen. Durch Umformung der Energiefunktion in ein stochastisches Modell, können die Parameter der Schreiblinien durch den Expectation Maximization Algorithmus [BeCh94] iterativ bestimmt werden.

In Abb. 3 werden die von dem Algorithmus bestimmten Schreiblinien dargestellt.



**Abb. 3: Ermittelte Schreibleinien**

Somit benötigt der Algorithmus als Eingabe nur die lokalen Maxima und Minima der Schrift. In der Online-Erkennung ist dies durch den hohen Quantifizierungsgrad eine triviale Aufgabe, so daß meist nur eine konstante Anzahl von Punkten betrachtet werden muß, um zu beurteilen, ob sich das Vorzeichen des Gradienten ändert.

In der Off-Line Erkennung ist jedoch der Quantifizierungsgrad vergleichbar gering und die Schrift kann nicht als Linienzug betrachtet werden, da die Offline-Schriften keine konstante Strichstärke der Größe eins besitzen. Deshalb betrachtet man ausschließlich die Kontur der Schrift, um lokale Extrema zu bestimmen. Die Konturpunkte können als diskrete Abtastpunkte der ursprünglichen Schrift betrachtet werden.

Allgemein liegt ein lokales Extremum vor, wenn sich das Vorzeichen des Gradienten in einem hinreichend kleinen Intervall ändert. Dies gilt auch im diskreten Fall, wenn man die Gradienten einer Punktfolge ermittelt. Im Falle der BCCs gibt es allerdings nur Gradienten mit  $0, +1, -1, +\infty, -\infty$ . Des weiteren sind durch den Quantifizierungsgrad lange Plateaus häufig, bei denen der Gradient ständig  $0$  ist. Somit kann nur am Anfang bzw. Ende des Plateaus entschieden werden, ob es sich bei dem betrachteten Plateau um ein lokales Extremum handelt. Ist z.B. am Anfang des Plateaus der Gradient aufsteigend und ist der Gradient am Ende absteigend, so handelt es sich um ein lokales Maximum.

Formal kann für eine Punktfolge  $p^{(i)}$  eines BCC ein lokales Extremum so definiert werden:

$$\begin{aligned} &\exists \text{ Intervall } [j, k] \text{ mit } j < k: \\ &\forall i \in [j, k]: \text{dir}(p^{(i)}, p^{(i+1)}) \text{ horizontal} \wedge \\ &(\text{dir}(p^{(j-1)}, p^{(j)}) \text{ aufsteigend} \wedge \text{dir}(p^{(k)}, p^{(k+1)}) \text{ absteigend}) \text{ bzw. umgekehrt} \end{aligned}$$

Hierbei klassifiziert *dir* einen Gradienten zwischen zwei Punkten nach horizontal ( $0$ ), aufsteigend ( $+1, +\infty$ ) und absteigend ( $-1, -\infty$ ).



### 3.2.2 Transformation

Das Verfahren zur Bestimmung der Schreiblinien liefert die vier parallelen Schreiblinien in der Form:  $f(x) = k(x - x_0)^2 + s(x - x_0) + y_0$

Die Schreiblinien unterscheiden sich nur durch die Abszisse  $y_0$ . Deshalb kann im weiteren eine beliebige Schreiblinie als Transformationsgrundlage betrachtet werden.

Ziel der Transformation ist es, die Schrift so zu transformieren, daß die Lage der Schreiblinien waagrecht ist. Dies entspricht einer Abbildung einer Parabel  $f$  auf eine waagrechte Gerade  $g(x) = y_0$ . Es gibt einige Möglichkeiten eine solche Abbildung zu definieren, z.B. könnte man die Parabel einfach vertikal auf die Gerade projizieren. Jedoch wäre eine solche Projektion nicht mehr längentreu. Beobachtungen zeigen, daß Schreiber, deren Schreiblinien parabelförmig sind, dazu neigen Aufwärtsstriche, wie beim „l“, senkrecht zur Parabel zu schreiben. Deshalb würde eine einfache Projektion die Schrift zwar normalisieren, jedoch auch verzerren. Die Forderung der Längentreue bedeutet, daß die Weglänge auf einer Parabel des Punktes  $P_1(0, f(0))$  nach  $P_2(u, f(u))$  gleich der Weglänge auf einer Gerade des Punktes  $Q_1(0, g(0))$  nach  $Q_2(v, g(v))$  ist, wenn  $u$  auf  $v$  abgebildet wird.

Mit diesem Wissen kann man sich eine längentreue Abbildung konstruieren.

Die Weglänge auf einer Parabel von  $P_1$  nach  $P_2$  entspricht:

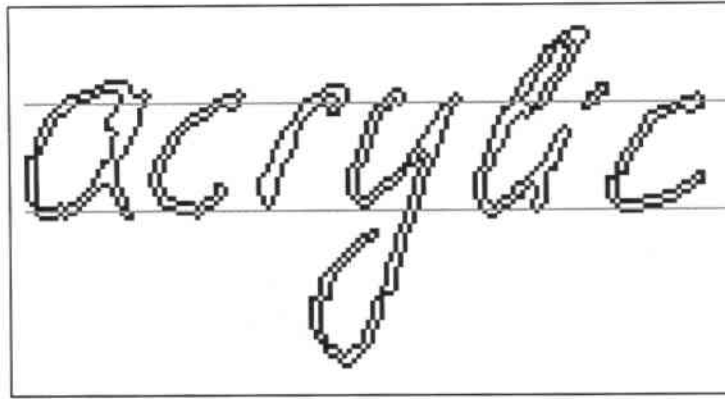
$$\int_0^u \sqrt{1 + f'(x)^2} dx = \int_0^u \sqrt{1 + (2a_2x + a_1)^2} dx \text{ mit } a_2 = k, a_1 = s - 2kx_0$$

dann erhält man für  $v$ :

$$v = \begin{cases} \frac{1}{4a_2} \left( (2a_2u + a_1) \sqrt{(a_1 + 2ua_2)^2 + 1} + \ln \left( (2a_2u + a_1) + \sqrt{(a_1 + 2ua_2)^2 + 1} \right) \right), & \text{für } a_2 \neq 0 \\ u\sqrt{1 + a_1^2}, & \text{für } a_2 = 0 \end{cases}$$

Abb. 4 entsteht aus der Anwendung dieser Abbildung auf Abb. 3.

Diese Abbildung ist jedoch sehr rechenintensiv, wenn jeder Konturpunkt transformiert werden muß. Zudem ist es eher die Ausnahme, daß Schreiber parabelförmig schreiben. Deshalb wird für eine hinreichend kleine Krümmung  $k$  die Parabel als Gerade behandelt. Diese Gerade muß dann nur noch in die Horizontale gedreht werden.



**Abb. 4: Basisliniennormalisiertes Wort**

Bei der Anwendung dieser Transformation auf BCCs können jedoch BCC Bedingungen verletzt werden, indem Punkte zu weit auseinander liegen oder benachbarte Punkte sogar auf ein und den selben Pixel abgebildet werden. Deshalb müssen die BCCs nachträglich korrigiert werden.

### 3.3 Scherwinkelnormierung

Die Scherwinkelnormierung ist ein wichtiger Schritt in der Vorverarbeitung der Handschriften. Der Neigungswinkel der Schrift ist von Schreiber zu Schreiber unterschiedlich. So kann es vorkommen, daß der Neigungswinkel so schräg ist, daß ein Buchstabe seinen direkt benachbarten vertikal überdeckt. Generiert man die Merkmale für den Erkenner spaltenweise, dann führt dieses Phänomen zu einer Abnahme der Erkennungsleistung. Auch bei anderen Ansätzen führt diese Variation der Handschrift zu schlechteren Erkennungsleistungen. Ziel der Normierung ist es schließlich, alle extremen Variationen zu eliminieren und die verschiedenen Handschriften so einheitlich wie möglich dem Erkenner zu übergeben. In diesem Fall bedeutet dies, den Scherwinkel einer Schrift auf  $90^\circ$  zu normieren.

Die Scherwinkelnormierung teilt sich auch in zwei Schritte auf:

1. Detektion des Scherwinkels
2. Normierung des Scherwinkels

#### 3.3.1 Detektion des Scherwinkels

Anstatt auf der Ebene des Rasterbildes zu arbeiten, wird hier der Scherwinkel anhand der BCCs ermittelt und korrigiert. Dadurch daß nur die Kontur einer Schrift betrachtet wird, reduziert sich die zu betrachtende Datenmenge drastisch. [BCGM95]

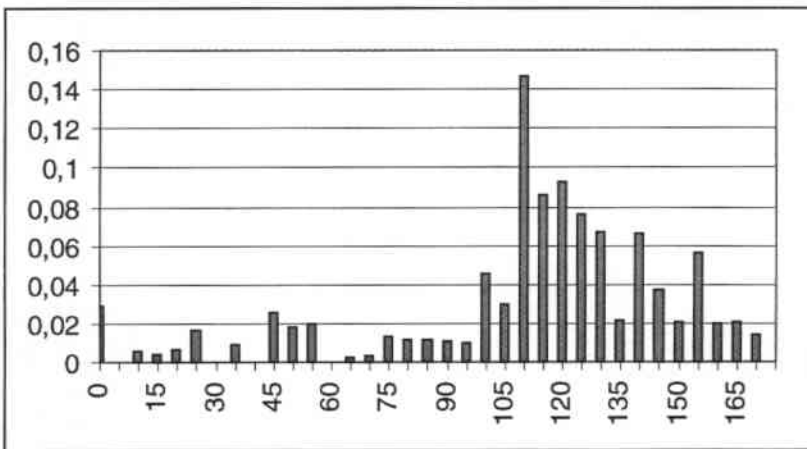
In dieser Arbeit wird der Scherwinkel durch einen statistischen Ansatz ermittelt. Jede Kante einer BCC wird durch ihren ungerichteten Gradient ( $0^\circ$ - $180^\circ$ ), gewichtet durch ihre Länge, in ein eindimensionales Histogramm eingetragen. Dieser Ansatz ist sehr verwandt mit der bei

Rasterbilder verwendeten Hough-Transformation [BaBr82][Glog92]. Anhand dieses Gradienten-Histogramms kann man dann den Scherwinkel schätzen.

Da die Kanten in den BCCs nur vier verschiedene Gradienten haben und nur eine Länge von eins besitzen, müssen die BCCs zur Ermittlung des Scherwinkels weiter aufbereitet werden. Durch den in [WaDa83] beschriebenen Approximations-Algorithmus werden die BCCs durch Geradensegmente variabler Länge und variabler Steigung approximiert. Als Beispiel zeigt Abb. 5 die Approximation von Abb. 4. Diese Kanten werden nun in das eindimensionale Histogramm eingetragen (Abb. 6).



**Abb. 5: Scherwinkel-Approximation**

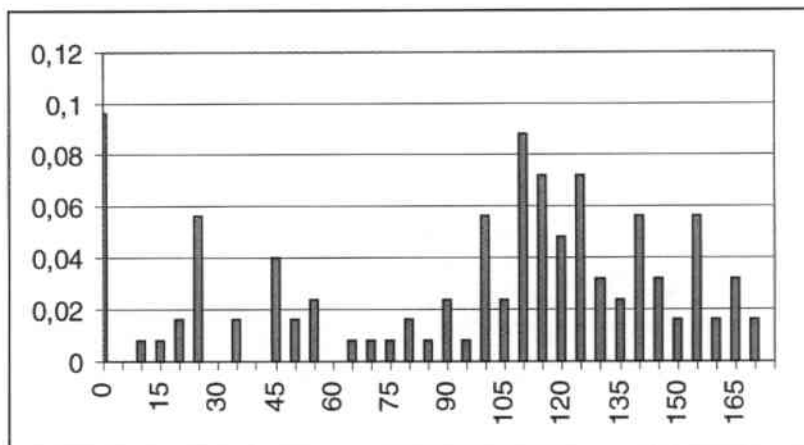


**Abb. 6: Histogramm**

Das Histogramm enthält am Ende für jeden Gradienten die Summe der Längen derjenigen Kanten, die genau diesen Gradienten haben. Anschließend wird das Histogramm noch mit der Gesamtlänge normalisiert. Um den Scherwinkel anhand des Histogramms zu ermitteln, kann man verschiedene Strategien verfolgen. Alle Strategien betrachten nur Gradienten im Bereich zwischen  $45^{\circ}$ - $135^{\circ}$ . Ein größeres Intervall würde das Ergebnis verzerren, da es kaum Schreiber

gibt, die mit einem Neigungswinkel größer  $45^\circ$  schreiben. Zusätzlich würden Kanten ins Gewicht fallen, die keinesfalls Aussagen über den Neigungswinkel der Schrift geben, z.B. diagonale Kanten beim „x“ und zusätzlich horizontale Kanten beim „Z“. In der „normierten“ Schrift gibt es auffällige Anhäufungen im Bereich um  $90^\circ$  (vertikale Striche),  $45^\circ$  bzw.  $135^\circ$  (diagonale Striche) und  $0^\circ$  bzw.  $180^\circ$  (horizontale Striche). Bei einer kursiven Schrift verlagert sich der Häufungspunkt der vertikalen Striche je nach Neigungswinkel, wobei die Häufungspunkte der diagonalen und horizontalen fast gleich bleiben. Ziel aller Strategien ist es den Häufungspunkt der vertikalen Striche zu suchen:

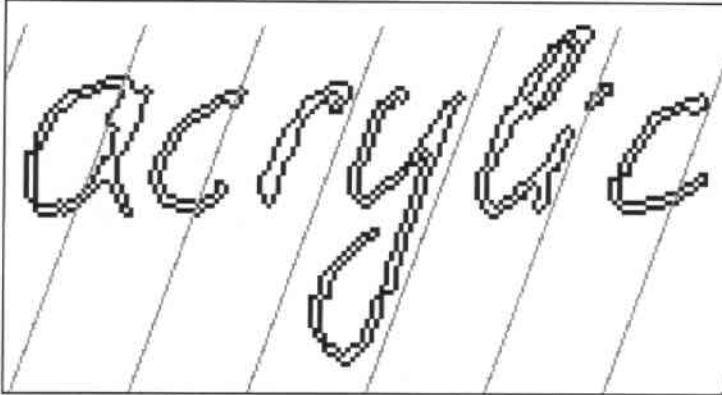
- *straight maximum*: Ermittelt das Maximum innerhalb des Histogramms und nimmt den Gradienten dieses Maximums als Scherwinkel an.
- *soft maximum*: Ermittelt das Maximum der Summe über jeweils drei benachbarte Gradienten. Der Gradient dieses Maximum wird als Scherwinkel angenommen. Hinsichtlich Verfälschung durch einzelne extrem lange Kanten ist diese Strategie sicherer als die straight maximum-Strategie.
- *low risc maximum*: Ermittelt wie die straight maximum-Strategie zunächst das Maximum des Histogramms. Jedoch wird nun derjenige Gradient ausgewählt, dessen Wert innerhalb einer definierten Toleranz unterhalb des Maximum liegt und am nächsten zu  $90^\circ$  liegt. Damit erreicht man, daß zumindest nicht-kursive Wörter fälschlich normalisiert werden. Auch wird somit oft ausgeschlossen, daß man als Maximum den Häufungspunkt der diagonalen Striche ermittelt. Der Preis dafür ist, daß es passieren kann, daß man einen Scherwinkel ermittelt der etwas kleiner ist als der eigentlich richtige.
- *weighted maximum*: Anstatt nur die Kantenlänge zu betrachten, wird hier zusätzlich noch die Anzahl der Kanten hinzugezogen, die in den Histogrammeintrag aufgenommen wurden (Abb. 7 zeigt dieses Frequenz-Diagramm von Abb. 4). Das Maximum bzw. Vergleichswert wird dann aus der Kantenlänge multipliziert mit der Anzahl der betrachteten Kanten gebildet. Ansonsten arbeitet diese Strategie wie die low risc maximum-Strategie. Somit bleiben die Vorteile der low risc maximum-Strategie erhalten und einzelne verfälschende Kanten werden nicht so stark gewichtet.



**Abb. 7: Frequenz Histogramm**

In der Praxis liefern die low risc maximum- und weighted maximum-Strategie sehr gute Ergebnisse. Die Ergebnisse sind jedoch auch abhängig vom Grad der Approximation und der Quantifizierung der Gradienten. Für die Approximation gilt, je niedriger der Approximationsgrad desto bessere Ergebnisse, da der Konturverlauf so getreu wie möglich repräsentiert werden sollte. Als Quantifizierungsgrad hat sich  $5^\circ$  pro Histogrammeintrag bewährt. Allerdings zeigen sich in der Praxis zwischen Approximation und Quantifizierung Abhängigkeiten.

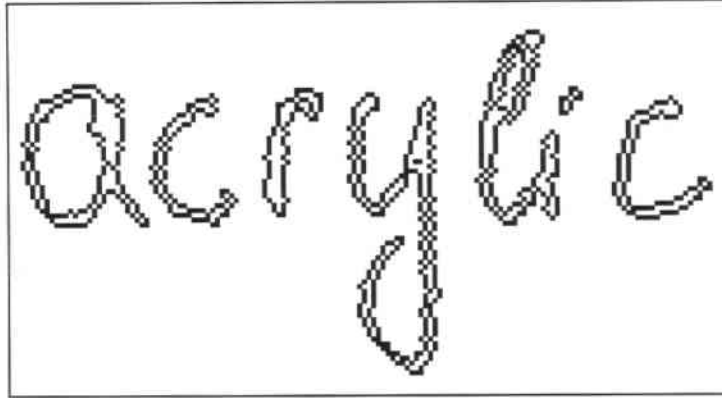
Abb. 8 zeigt den durch die weighted maximum-Strategie gefundenen Scherwinkel.



**Abb. 8: Ermittelter Scherwinkel**

Eines der Hauptprobleme dieser Methode liegt darin, daß davon ausgegangen wird, daß ein Wort vollständig mit dem selben Neigungswinkel geschrieben wird. Allerdings kann man oft beobachten, daß nur Wortteile kursiv geschrieben werden. Dieses Problem könnte behoben werden, indem man für jede BCC den Scherwinkel isoliert bestimmt und korrigiert. Jedoch führen gerade besonders kleine Kantenerhebungen zu falschen Ergebnissen, da die statistischen Daten zur korrekten Auswertung des Histogramms nicht ausreichen.

### 3.3.2 Normierung des Scherwinkels



**Abb. 9: Scherwinkelnormalisiertes Wort**

Die Normierung der Schrift durch den ermittelten Neigungswinkel  $\alpha$  entspricht geometrisch einer Scherung um  $90^\circ - \alpha$  in positiver Richtung. Dies entspricht der Projektion der Geraden  $y = x \tan \alpha$  auf die Achse  $x = 0$ . Es gilt für jeden Konturpunkt  $v(x, y)$ :

neuer Konturpunkt  $v'(x', y')$  mit  $x' = x - \frac{y}{\tan \alpha}$  und  $y' = y$

Durch die Anwendung dieser Transformation auf BCCs kommt es zu Verletzungen der BCC-Bedingungen. Innerhalb einer Zeile tauchen keine Probleme auf, da die Zeile nur verschoben wird. Jedoch bei Verbindungen zweier benachbarter Konturpunkte zweier verschiedener Zeilen können Seiteneffekte auftreten. So ist z.B. möglich, daß die Punkte zu weit auseinander liegen und interpoliert werden müssen. Besonders die entstandenen Spitzen müssen geglättet werden.

Das Ergebnis der Schernormierung von Abb. 8 kann man in Abb. 9 betrachten.

### 3.4 Skelettierung



Abb. 10: Unverarbeitete Handschrift



Abb. 11: Skelettierte Handschrift

Bisher hat die Handschrift nur physikalische Vorverarbeitungsstufen durchlaufen, wodurch Pixel für Pixel transformiert wurde. Für eine sinnvolle Erkennung sollte man jedoch von physikalischen pixelhaften Beschreibung auf eine abstraktere Darstellungsform übergehen [CoCo82]. Dies geschieht durch die Skelettierung und die anschließende Approximation. Die Skelettierung hat die Aufgabe die Schriftstärke zu normieren, die stark von dem Schreibmedium abhängig ist. Hierbei wird die variable Schriftstärke auf ein Pixel reduziert.

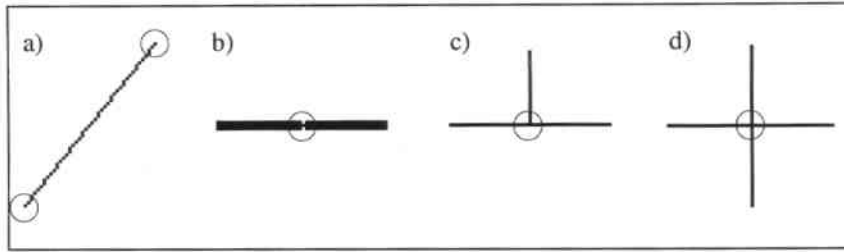
Hierzu wurde der Algorithmus von Kwok [Kwok88] benutzt, der sich besonders gut auf BCCs anwenden läßt.

Zuvor noch einige Begriffsdefinitionen:

Einen Pixel in einem Zusammenhangsgebiet, der nach dem Entfernen das Zusammenhangsgebiet in zwei Teilgebiete spaltet, nennt man *Schnittpunkt*.

Ein Pixel in einer BCC, der nur einmal in der BCC vorkommt und bei dem sich die Richtung der Kontur um  $180^\circ$  ändert, wird *Endpunkt* genannt.

Pixel, die weder Schnittpunkte noch Endpunkte sind, heißen *nicht-sichere Punkte*.



**Abb. 12: a) Endpunkte, b) Schnittpunkt, c) dreifacher Schnittpunkt, d) vierfacher Schnittpunkt**

Der Algorithmus arbeitet iterativ. In jeder Iteration wird die gesamte Kontur durchlaufen und eine neue „dünnere“ Kontur wird aufgebaut. Dabei wird für jeden Punkt entschieden, ob es sich um einen Schnittpunkt, Endpunkt oder nicht-sicheren Punkt handelt. Nicht-sichere Punkte werden aus der Kontur nach entsprechenden Regeln [Kwok88] entfernt. Somit bilden die Schnittpunkte und Endpunkte das Skelett. Die Klassifizierung der Konturpunkte erhält man einfach durch ein Bitmap, in das vor der ersten Iteration die Kontur eingezeichnet wird. Dabei enthält jeder Pixel die Vielfachheit des entsprechenden Konturpunktes. Wird ein Punkt in der Iteration besucht, wird sein korrespondierender Pixel in der Bitmap um eins erhöht. Beträgt der Wert eines Konturpunktes größer gleich zwei, dann handelt es sich um einen Schnittpunkt. Dadurch daß der Algorithmus immer drei Punkte der alten Kontur betrachtet, um die neue Kontur geeignet aufzubauen, sind Endpunkte durch ihren Richtungswechsel auch einfach zu bestimmen.

In Abb. 11 kann man beispielsweise das Skelett von der unverarbeiteten Handschrift in Abb. 10 betrachten.

Am Ende des Algorithmus besteht die Kontur nur noch aus Schnittpunkten und Endpunkten. Und das dazugehörige Gebiet hat die konstante Stärke von eins. Zusätzlich verfügt man über die abstrakten Merkmale wie Schnittpunkte und Endpunkte, die für den Erkenner von Nutzen sein können. So kann man die Schnittpunkte nach ihrer Vielfachheit weiter klassifizieren (Abb. 12). Einen Schnittpunkt, der nur zweimal in einer BCC vorkommt, verbindet zwei Gebiete oder zwei Striche. Weitaus interessanter sind Schnittpunkte höherer Ordnung: Ein dreifacher Schnittpunkt verbindet drei Gebiete und markiert die Berührung zweier Striche, ein vierfacher Schnittpunkt verbindet vier Gebiete und markiert den Schnitt zweier Striche.



### 3.5 Abstandnormalisierung

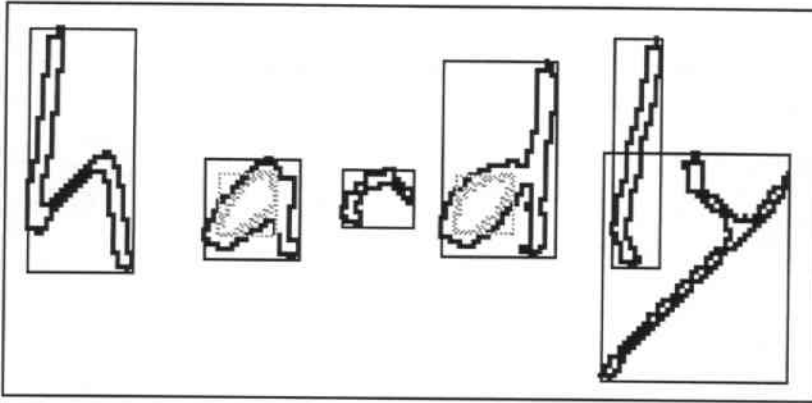


Abb. 13: Ungleiche Buchstabenzwischenräume

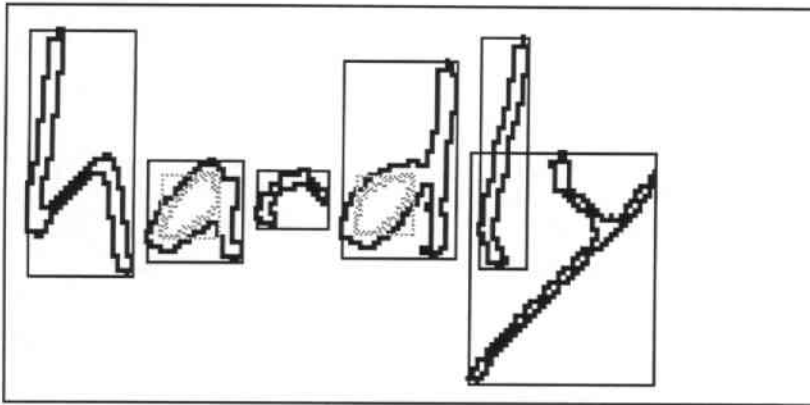


Abb. 14: Abstandnormalisiertes Wort

Nachdem alle Normierungen an dem Bitmap abgeschlossen sind, sollten noch zusätzlich die Abstände zwischen isolierten Buchstaben normiert werden. Da der Erkenner sowohl Druckschrift als auch Schreifschrift erkennen soll, liegt es nahe die Druckbuchstaben möglichst dicht aneinander zu reihen, damit er nicht durch zu große Abstände zwischen den Buchstaben verwirrt wird. Allerdings kann der Erkenner durch wenige Leerspalten schon erkennen, daß mit gewisser Wahrscheinlichkeit ein neuer Buchstabe zu klassifizieren ist.

Das Problem der Einführung von Äquidistanzen zwischen isolierten Buchstaben läßt sich auf die Abstandsnormierung von benachbarten Zusammenhangsgebieten reduzieren, da ein Zusammenhangsgebiet meistens mindestens einen Buchstaben repräsentiert.

Zwei Zusammenhangsgebiete sind *benachbart*, wenn sich die umschreibenden Rechtecke bezüglich der x-Achse nicht überdecken. Zwei Zusammenhangsgebiete sind folglich *verschachtelt*, wenn sie nicht benachbart sind.

Somit ist bei verschachtelten Zusammenhangsgebieten nicht entscheidbar, ob sie zusammen einen Buchstaben repräsentieren (z.B. „i“) oder ob sich zwei Buchstaben nur bezüglich der x-

Achse überdecken (z.B. „ly“). Um diesem Manko aus dem Weg zu gehen, werden verschachtelte Zusammenhangsgebiete nur als Gesamtheit betrachtet. D.h. ihre umschreibenden Rechtecke werden verschmolzen und die darin liegenden Zusammenhangsgebiete werden als eine zusammengehörige Gruppe betrachtet. Im weiteren braucht man nur noch die Vater-Zusammenhangsgebiete (Zusammenhangsgebiete 1. Ordnung) betrachten, da diese mit ihren Söhnen per Definition verschachtelt sind.

Damit lautet der Algorithmus wie folgt:

Bbox entspricht dem umschreibenden Rechteck des letzten betrachteten Zusammenhangsgebietes und ist initial leer.

t entspricht dem Translationswert und ist initial 0.

d entspricht dem beliebigen konstanten Abstand.

sortiere alle Zusammenhangsgebiete (1. Ordnung) aufsteigend nach dem x-Minimum des umschreibenden Rechtecks.

für alle Zusammenhangsgebiete (aufsteigend):

falls das Zusammenhangsgebiet mit Bbox verschachtelt ist,  
dann

    verschmelze das umschreibende Rechteck mit BBox

sonst

    dakt ist der x-Abstand zwischen Bbox und dem umschreibenden Rechteck und ist für das erste Zusammenhangsgebiet 0

    aktualisiere den Translationswert  $t = t + d - dakt$

    weise Bbox das umschreibende Rechteck des Zusammenhangsgebiet zu

    verschiebe das Zusammenhangsgebiet und alle untergeordneten um t

Nach Beendigung des Algorithmus sind alle benachbarten Zusammenhangsgebiete äquidistant, wie es am Beispiel von Abb. 13 und Abb. 14 ersichtlich ist.

### 3.6 Approximation

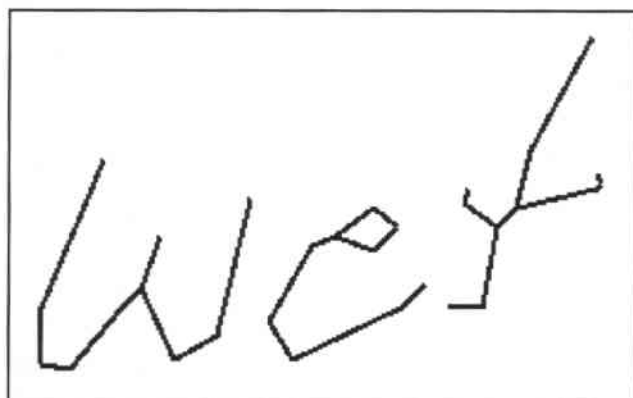


Abb. 15: Approximierte Handschrift

Die anschließende Approximation überführt die immer noch pixelbehaftete Darstellung des Skeletts in eine abstraktere Form. Das Skelett wird durch Liniensegmente unter Einhaltung der signifikanten Merkmale, wie Endpunkte und mehrfache Schnittpunkte, approximiert. Dadurch wird auch nicht die Struktur des Skeletts zerstört.

Für die Implementierung wurde ein schneller Algorithmus von Wall und Danielsson [WaDa83] benutzt, der ursprünglich für digitalisierte Kurven entwickelt wurde. Da er auf der Grundlage von Kurvenpunkten arbeitet, ist er auch problemlos auf BCCs anwendbar, die abstrakt betrachtet auch Stützpunkte einer Kurve sind. Zusätzlich wurde der Algorithmus auf die Eigenheiten von Handschriften angepaßt.

Der originale Algorithmus lautet wie folgt: Das längste erlaubte Liniensegment wird dadurch gefunden, indem man ausgehend von einem initialen Konturpunkt die folgenden Konturpunkte solange extrapoliert, bis ein gewisses Testkriterium nicht mehr erfüllt wird. Das approximierte Liniensegment entspricht dann dem Segment von dem initialen Konturpunkt zu dem Punkt, der das Testkriterium als letztes bestanden hat. Das hier benutzte Testkriterium enthält einen beliebig konstanten Parameter  $T$ .  $T$  bestimmt den Grad der Approximation. Die geometrische Interpretation des Parameters  $T$  entspricht der maximalen erlaubten Flächenabweichung pro Längeneinheit des approximierten Liniensegments.

Als zusätzlicher Parameter wurde *decs* eingeführt. Gerade bei Handschriften kommen oft enge Spitzen im Konturverlauf vor, die oft interessante Merkmale repräsentieren. So kann es vorkommen, daß die Länge des am Ende approximierten Liniensegments nicht größer ist als alle zuvor ermittelten. Somit würde die Spitze einfach übersehen werden. Mit *decs* kann man angeben, wie oft hintereinander die Länge des Liniensegments abnehmen darf, bevor man mit der Approximation abbricht. Nach dem Abbruch wählt man den betrachteten Konturpunkt als Endpunkt des Liniensegments, der die Länge maximiert.

Der bisher vorgestellte Algorithmus eignet sich nur wenig für Handschriften bzw. BCCs, da es möglich ist, daß sich approximierte Liniensegmente überschneiden können. Somit wird das Zusammenhangsgebiet verzerrt und hat oft nur noch wenig Ähnlichkeit mit der ursprünglichen

Handschrift. Für Anwendungen wie z.B. die Schernormierung ist dies allerdings nicht relevant, da die approximierten Handschrift nicht weiterverarbeitet wird.

Um ein Skelett zu approximieren, das als BCC vorliegt, sollte man zusätzlich beachten, daß man fast nur Schnittpunkte in Approximation betrachtet. D.h. ein Pixel kommt mindestens zweimal in der Konturfolge vor. Um Schnitte mit Liniensegmenten oder unbeabsichtigte Schleifen zu vermeiden, darf jeder Pixel nur einmal in Approximation betrachtet werden. Trifft man also während der Approximation auf einen schon betrachteten Pixel, wird diese abgebrochen und man erhält das Liniensegment bis zu diesem Pixel. Der nächste initiale Konturpunkt ist dann der benutzte Pixel, der der Vorgänger des nächsten unbenutzten Pixels ist.

Beachtet man noch zusätzlich mehrfache Schnittpunkte und Endpunkte als Abbruchkriterium für die Approximation, indem bei deren Antreffen das Liniensegment schließt, dann erhält man eine gute Annäherung an das originale Skelett, wie in Abb. 15, die aus der Approximation von Abb. 11 entsteht. Somit bleiben auch noch die signifikanten Punkte erhalten.

## 4 Merkmalsgenerierung

Ausgehend von der approximierten vorverarbeiteten Handschrift werden zwei Methoden der Merkmalsgenerierung betrachtet: die erste basiert auf den topologischen und die zweite auf den geometrisch-statistischen Eigenschaften der Handschrift. Bei beiden Methoden wird ein fest definiertes Fenster entlang der x-Achse über die Handschrift geschoben. Aus dem Bereich innerhalb des Fensters werden dann jeweils die Merkmale gewonnen. Eine weitere Möglichkeit wäre die Merkmalsgenerierung entlang der temporalen Achse, wie bei Online-Systemen, allerdings wäre dafür noch zusätzlich eine aufwendige Bestimmung der Schriftführung notwendig [Jäger96]. Unklar ist dann die Kombination der temporalen und der topologischen Merkmale. Im Rahmen dieser Studienarbeit wurde diese Alternative allerdings nicht erarbeitet.

Für den Einsatz auf TDNN- und HMM-Klassifikatoren ist es erforderlich, daß entlang der x-Achse Merkmalsvektoren konstanter Dimension generiert werden. Zusätzlich werden den Merkmalsvektoren der beiden Methoden noch eine konstante Anzahl lokal-statistischer Merkmale hinzugefügt, die schon während der Vorverarbeitung generiert wurden.

### 4.1 Lokal-statistische Merkmale

In dem durch das Fenster definierten Bereich der Handschrift wird das Auftreten von Schnittpunkten, Endpunkten, Schleifen und Zusammenhangsgebieten bestimmt. Diese lokal-statistischen Merkmale sollen dem Erkenner eine zusätzliche Hilfestellung zu den topologischen oder geometrisch-statistischen Merkmalen geben. Um durch die Erkennungsleistung durch zu große Merkmalsvektoren zu beeinträchtigen, werden nur vier dieser Merkmale betrachtet, obwohl man noch weitere Merkmale hinzunehmen könnte. Ein simples Verfahren um die Häufigkeit eines Ereignisses als beschränktes numerisches Merkmal zu repräsentieren, ist die reziproke Häufigkeit. Somit ist das Merkmal beschränkt durch die Schranken 0 und 1.

Nach der Skelettierung sind Schnitt- und Endpunkte bereits markiert, und somit leicht zu ermitteln. Schleifen müssen vor der Approximation bestimmt werden, da während der Approximation jedes Vater-Zusammenhangsgebiet mit all seinen untergeordneten Zusammenhangsgebieten verschmilzt. Schleifen sind durch untergeordnete Konturen charakterisiert, denn hat ein Vater eine untergeordnete Kontur, dann hat das Vater-Zusammenhangsgebiet ein Loch, was man als Schleife interpretiert. Gezählt werden alle Schleifen, die zum Teil oder ganz innerhalb des Fensters liegen. Für die Häufigkeit der Zusammenhangsgebiete gilt dasselbe für die Zählweise. Allerdings beschränkt man sich nur auf die Vater-Zusammenhangsgebiete, da wie schon erwähnt, die untergeordneten Zusammenhangsgebiete nach Approximation entfallen.

## 4.2 Topologische Merkmale

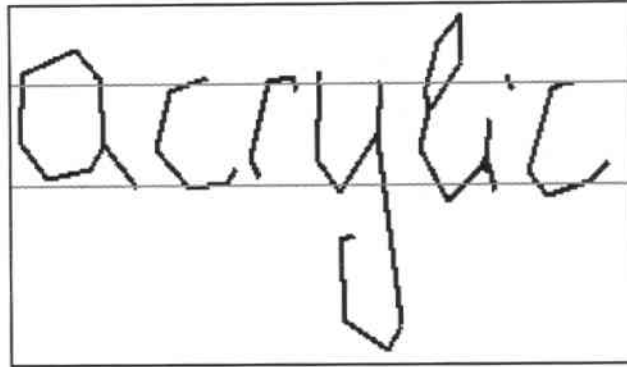


Abb. 16: Vorverarbeitete Handschrift

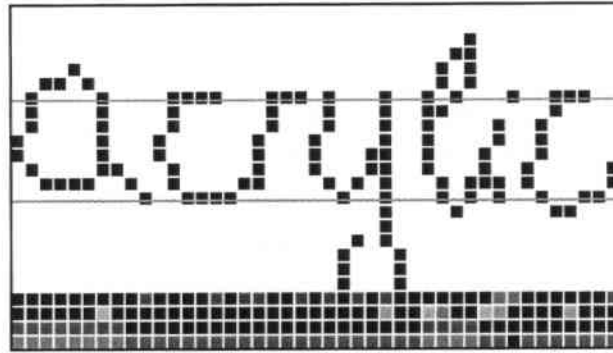


Abb. 17: Merkmalsatz 1

Die topologischen Merkmale werden nicht direkt aus der approximierten Handschrift gewonnen, sondern spaltenweise aus dem korrespondierenden Bitmap.

Die als Kantenmenge vorliegende Handschrift, wird zunächst in ein Bitmap konstanter Höhe konvertiert. Dabei wird die Coreline und die Baseline in dem Bitmap fest gewählt. Die Handschrift wird dann so skaliert, daß dessen Coreline und Baseline, bekannt durch die Basislinienerkennung, auf die des Bitmaps abgebildet wird. Somit wird der mittlere Basislinienbereich einer Handschrift immer durch den selben y-Achsen Abschnitt des Bitmaps repräsentiert.

Die Auswahl der Größe und Position des mittleren Basislinienbereichs im Bitmap sollte sehr sorgfältig geschehen. Gemessen am Informationsgrad (Pixeln) ist dieser Bereich in einer durchschnittlichen Handschrift dichter besetzt als der untere oder obere Bereich. Somit ist der mittlere Basislinienbereich für eine erfolgreiche Klassifikation am wichtigsten.

Anhand verschiedener Versuche haben sich folgende Größen für die Praxis als geeignet erwiesen: eine Bitmaphöhe von 20 Pixel, Coreline an y-Position 6 und Baseline an y-Position 13 erwiesen. Hier entspricht der mittlere Bereich, insgesamt 8 Pixel, etwas mehr als einem Drittel der Bitmaphöhe. Für den oberen Bereich verbleiben 5 und für den unteren Bereich 6 Pixel.

Es stellt sich nun die Frage, was zu tun ist, wenn der obere oder untere Bereich größer ist als der mittlere Bereich. Bei einer Skalierung der gesamten Handschrift, würde der obere oder untere Bereich nicht mehr in den vorhergesehenen Bitmapbereich passen. Es bieten sich zwei Alternativen an: entweder werden zu große Bereiche einfach abgeschnitten, oder man geht zur stückweisen bereichsbezogenen Skalierung („Squashing“) über. Die stückweise Skalierung hat jedoch die Eigenschaft, das Schriftbild stark zu verzerren, besonders wenn kritische Normierungen wie Basislinien- oder Scherwinkelnormierung nicht optimal gearbeitet haben. Somit entscheidet man sich für das Abschneiden überstehender Bereiche und verliert einen Teil der Handschrift. Jedoch sind diese Teile im Allgemeinen überwiegend dünn besetzt und tragen nicht viel Information zur Klassifikation bei. Des Weiteren wird die gesamte Handschrift skaliert und somit wird auch das Schriftbild nicht verzerrt.

Ein topologisches Merkmal entspricht dann dem Pixel an einer  $y$ -Position des Bitmap. Jedes dieser Merkmale ist binär, entweder 0 für Hindergrundinformation oder 1 für Schriftinformation. Die Merkmalsvektoren werden spaltenweise generiert, somit entspricht ein Merkmalsvektor genau einer Spalte des Bitmap. Folglich ist dessen Dimension gleich der Höhe des Bitmap.

Um die lokal-statistischen Merkmale noch hinzufügen zu können, muß man für jede Spalte im Bitmap noch ein korrespondierendes Fenster in der vorverarbeiteten Handschrift definieren. Dieses läßt sich jedoch durch Umkehrung der Skalierung leicht bestimmen. Jedoch wird dieses Fenster nicht für überstehende Bereiche abgeschnitten. Zusammen mit den topologischen Merkmalen bilden die vier lokal-statistischen Merkmale den Merkmalsatz 1 der Dimension: Höhe des Bitmap + 4.

Anschaulich kann man den Merkmalsatz 1 in Abb. 17 betrachten, der der vorverarbeiteten Handschrift in Abb. 16 entspricht.





### 4.3 Geometrisch-statistische Merkmale

Der geometrisch-statistische Ansatz bestimmt die Merkmale direkt aus der approximierten Handschrift. Da die Handschrift als Kantenmenge vorliegt, liegt es nahe die Merkmale aus den Kantenbeziehungen zu gewinnen. Die Art und Weise wie Kanten verbunden sind, gibt einen detaillierteren geometrischen Aufschluß als die topologische Betrachtungsweise.





Die Analyse der Kanten soll Aufschluß über die Geometrie der Handschrift geben, deshalb wird im folgenden erst die Art der Verbindung zwischen Kanten und dann die Lage der Kante betrachtet.

Im folgenden werden wir von der noch physikalischen Darstellung der Handschrift auf eine geometrische Darstellungsweise übergehen. Diese Abstraktion wird die Handschrift noch weiter vereinfachen und die spezifische Merkmalsgenerierung erleichtern.

Betrachtet man eine hinreichend approximierte Handschrift, dann kann man zur Bestimmung der Merkmale ein Modell benutzen, das Kanten einer gewissen Art benutzt. In unserem Modell sind das folgende Kanten:

			
1. horizontal	2. vertikal	3. diagonal+	4. diagonal-

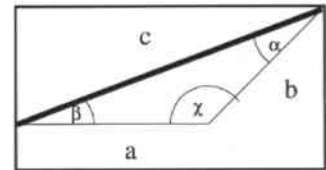
Dieses Kantenmodell kann jede Handschrift repräsentieren. Da es nur vier verschiedene Kanten gibt, ist auch die Art der Verbindung zwischen zwei Kanten beschränkt. Betrachtet man den zwischen zwei Kanten eingeschlossenen Winkel, dann gibt es nur vier Kategorien: Weiterleitung, Bogen, Ecke, Spitze:

			
1. Weiterleitung	2. Bogen	3. Ecke	4. Spitze

In diesem Kantenmodell besitzt man acht Merkmale: die vier verschiedenen Kantenverbindungen (Verbindungsmerkmale) und die vier verschiedenen Lagen (Lagemerkmale). Jedoch bedarf es noch einer numerischen Darstellung dieser Merkmale und der Überführung der approximierten Handschrift in das Modell.

Allgemein ist es sehr unwahrscheinlich, daß eine Handschrift nach der Approximation nur aus vier verschiedenen Kanten besteht. Daher ist eine Überführung in das Kantenmodell notwendig. Man definiert man den numerischen Wert der Lagemerkmale für eine horizontale Kante durch:  $horizontal = 1$ ,  $vertikal = 0$ ,  $diagonal+ = 0$ ,  $diagonal- = 0$ . Analog definiert man vertikale und diagonale Kanten. Dann kann man eine Kante beliebiger Lage als Anteil der vier Basislagen darstellen. Eine Möglichkeit wäre, die beliebige Kante durch die zwei Basislagen zu approximieren, die sie unmittelbar begrenzen.

Beispiel: Sei  $c$  eine beliebige Kante mit Lage  $0^\circ \leq \beta \leq 45^\circ$ , dann wird  $c$  unmittelbar von der horizontalen ( $a$ ) und positiv diagonalen ( $b$ ) Basislage begrenzt. Bildet man nun das Dreieck dieser drei Kanten, dann kann man durch den Sinussatz den Anteil der Horizontalen und der positiv Diagonalen an  $c$  berechnen.  $\chi$  ist immer  $135^\circ$ ,  $\beta$  ist durch die Lage von  $\chi$  bekannt und  $\alpha$  ergibt sich durch  $190^\circ - \chi - \beta$ .



Dann ergibt sich für den horizontalen Anteil:  $h(\beta) = \frac{\sin \alpha}{\sin \chi} = \frac{\sin(45^\circ - \beta)}{\sin(135^\circ)}$ , für  $0^\circ \leq \beta \leq 45^\circ$

Und für den positiv diagonalen Anteil:  $d^+(\beta) = \frac{\sin \beta}{\sin \chi} = \frac{\sin \beta}{\sin(135^\circ)}$ , für  $0^\circ \leq \beta \leq 45^\circ$



Der Anteil an vertikalen und negativ diagonalen ist offensichtlich 0, da sie die Kante nicht unmittelbar begrenzen.

Dieser Ansatz hat den Vorteil, daß man jede Kante in unser Kantenmodell überführen kann. Betrachtet man diese Überführung geometrisch, dann wird aus der relativ glatten Approximation eine sehr rauhe Kurve. Man sollte daher nur Kanten mit in das Modell übernehmen, die annähernd eine Basislage haben, ansonsten sollen sie nur wenig gewichtet werden. Dies gilt insbesondere für Kanten der Lagen:  $22,5^\circ$ ,  $67,5^\circ$ ,  $112,5^\circ$ ,  $157,5^\circ$ . Eine solche Einteilung erfolgt

mit der Funktion  $\varphi_{\mu,\sigma^2}(\beta) = e^{-\frac{(\beta-\mu)^2}{2\sigma^2}}$ , wobei  $\beta$  die Lage der Kante,  $\mu$  eine Basislage und  $\sigma^2$  die

Varianz ist.  $\sigma$  kann durch die Verwandtschaft von  $\varphi$  mit der Gauß-Verteilung als ein Streuungsmaß interpretiert werden. In Abb. 18 werden die Anteile der Basislagen an beliebigen Lagen graphisch dargestellt.

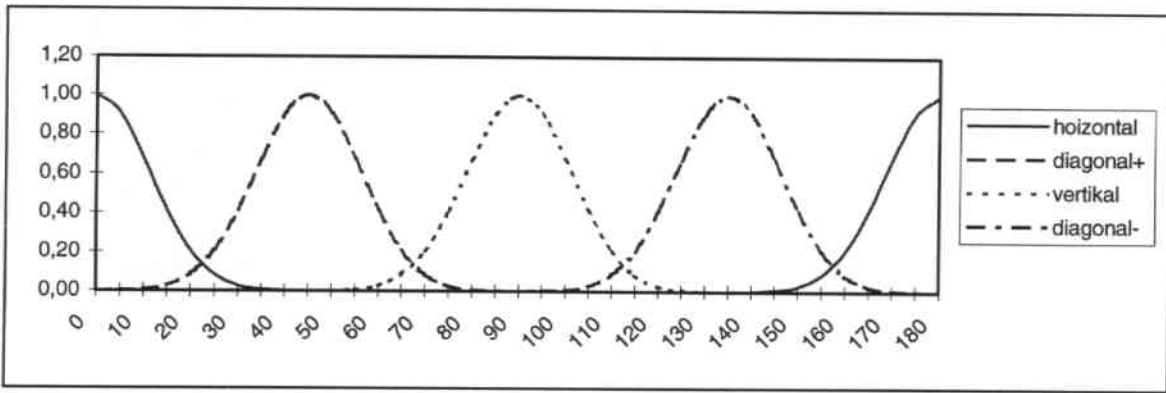


Abb. 18: Anteile der Basislagen

Diese Überführung bildet Kanten in der Nähe einer Basislage mit hohen Anteilen und sonst mit niedrigen Anteilen ab. Somit ist kein weiterer Approximationsschritt, wie oben beschrieben, nötig und es werden nur Kanten in das Modell übernommen, die annähernd dem Modell entsprechen.

Analog gilt dies auch für die vier Basisverbindungen.

Will man die Merkmale anstatt kantenweise aus einem Fenster bzw. aus einer durch ein Fenster definierte Kantenmenge gewinnen, dann muß man die Merkmale neu definieren. So sollen die Lagemerkmale eines Fensters die Häufigkeit der Kanten verschiedener Lagen repräsentieren. Da es aber in einem Fenster durchaus vorkommen kann, daß es viele kurze Kanten gibt, kann dies unser Merkmal verzerren, wenn man einfach die Kantenmerkmale aufsummiert. Deshalb wird zusätzlich die Länge einer Kante betrachtet. Benutzt man diese als Gewicht in der Summe, dann kann man die relativ kurzen nicht repräsentativen Kanten vernachlässigen.

Für die Verbindungsmerkmale gilt dies nicht. Denn selbst Verbindungen zwischen kurzen Kanten sind für die Merkmale relevant, da sie wie lange Kanten die Geometrie wiedergeben. Somit kann man für die Verbindungsmerkmale einfach die entsprechenden Kantenmerkmale aufsummieren.

Man definiert dann für die Lage- bzw. Verbindungsmerkmale:

$$\text{Lagemerkmal}_i^{(F)} := \sum \text{Länge}(k) * \text{Lagemerkmal}_i$$

$$\text{Verbindungsmerkmal}_i^{(F)} := \sum \text{Verbindungsmerkmal}_i$$

Wie oben beschrieben werden die Lage- und Verbindungsmerkmale aus einem fest definierten Fenster über der approximierten Handschrift gewonnen. Dieser Ausschnitt der Handschrift ist so hoch wie die Handschrift und kann beliebig aber fest breit definiert werden. Dieses Fenster wird dann entlang der x-Achse über die Handschrift geschoben, so daß ein Fenster immer an das letzte angrenzt (siehe Abb. 19). Maßgebend für eine gute Erkennungsleistung ist die Breite des Ausschnitts. Denn je größer das Fenster ist, desto mehr Kanten werden betrachtet und die Merkmale nehmen mehr Information auf. Bei steigender Kantenanzahl werden die Lage- wie die Verbindungsmerkmale immer gleichverteilter und tragen zur Klassifikation nicht mehr bei. Somit wäre es ideal, wenn man nur wenige Kanten in diesem Ausschnitt hätte. Hinzu kommt die Tatsache, daß der mittlere Basislinienbereich informationsträchtiger ist als die anderen Bereiche. Es liegt also nahe, den Ausschnitt in Interessengebiete aufzuteilen. Hierzu teilen wir das Fenster gemäß den bekannten Basislinien in 5 Bereiche (siehe Abb. 19) auf: oberer Bereich, Corelinebereich, mittlerer Bereich, Baselinebereich und unterer Bereich. Um die Merkmalsgenerierung zu stabilisieren, dürfen sich diese Gebiete um einen gewissen Teil überlappen. Dies hat den Vorteil, daß man nicht völlig auf die exakte Arbeitsweise der Basisliniennormierung angewiesen ist. Nun kann man die Merkmale je nach Interessengebiet ermitteln:

- im oberen Bereich sind horizontale und vertikale Kanten interessant, z.B. t-Striche und i-Punkte. Von Interesse sind auch Spitzen und Bögen.
- im Corelinebereich, mittleren Bereich und Baselinebereich sind alle Eigenschaften von Interesse. Somit werden horizontale, vertikale und diagonale Lagen und Spitzen, Ecken und Bögen als Verbindungen bewertet.
- im unteren Bereich sind horizontale und vertikale Kanten und Spitzen und Bögen interessant, um beispielsweise Buchstaben mit Unterlängen zu klassifizieren: „y“ „g“

Die Unterteilung von Diagonalen in positive und negative wurde nur zum Zweck des Kantenmodells vorgenommen. Allerdings kann man bei der Merkmalsgenerierung wieder von dieser Tatsache abstrahieren und diese beiden Merkmale in eines zusammenfassen, da es von geringerer Interesse ist, welche Richtung die Diagonale hat. Ein weiteres technisches Detail, das es zu erwähnen gilt, ist die Normierung der Merkmale. Theoretisch sind die Merkmale über einer Kantenmenge numerisch nach oben unbeschränkt. Dies ergibt sich aus der Aufsummierung der Kantenmerkmale. Praktisch überschreiten die Merkmale jedoch selten Schranken, die durch die Größe des betrachteten Ausschnitts bestimmt werden können. Da man die Ausschnitte so gewählt hat, daß nur wenige Kanten einer Art (wenn möglich jeweils nur eine) ist eine Normierung der Lagemerkmale durch die Größe des Ausschnitts plausibel. Diese Annahme erübrigt sogar die Normierung der Verbindungsmerkmale, da diese nicht mit der Länge gewichtet werden und somit durch die obere Schranke von 1 beschränkt sein sollten.

Somit erhalten wir zusammen mit den lokal-statistischen Merkmalen, die über den gesamten Ausschnitt gebildet werden, den Merkmalsatz 2 mit der Dimension 30.

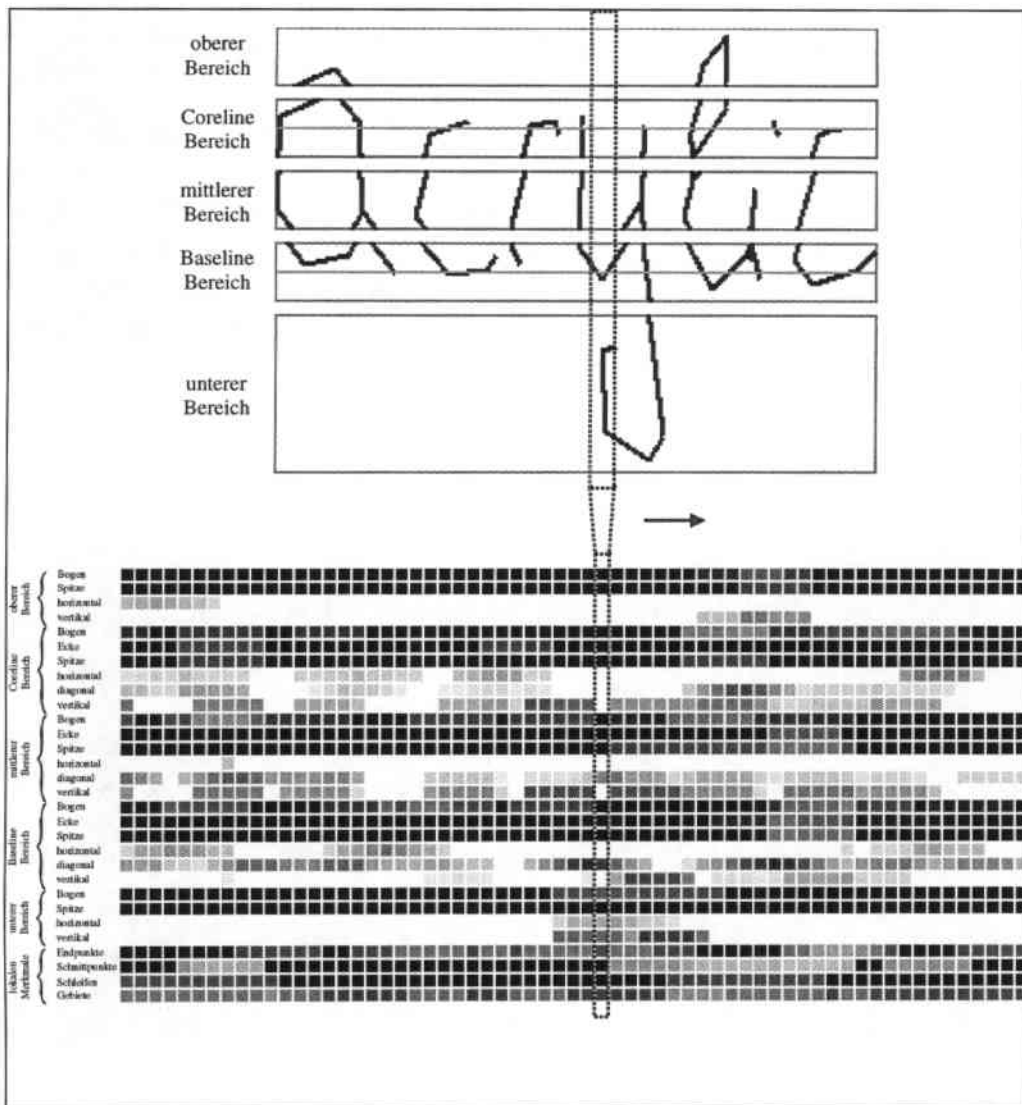


Abb. 19: Merkmalsatz 2

## 5 Experimente

Als Erkennen für die Offline-Handschriften wurde der NPen<sup>++</sup>-Erkennung [MFW95] benutzt. Diese Komponente integriert Segmentierung und Erkennung von Wörtern in einer einzigen Netzwerk-Architektur. Das Multi-State Time Delay Neural Network (MS-TDNN) [HaWa92] wurde eigentlich für die Erkennung von kontinuierlich gesprochener Sprache entworfen und wird auch erfolgreich für Online Wort Erkennung eingesetzt [MFW95].

### 5.1 Rahmenbedingungen

#### 5.1.1 Trainings- und Testdaten

Die zum Training und zum Testen der Netzwerke benutzten Daten stammen vom MIT. Insgesamt 120 Schreiber mußten jeweils ca. 53 Wörter schreiben. Der Schreibstil ist fast ausschließlich Druckschrift. In einigen Fällen werden jedoch zusammenhängende Buchstaben benutzt und somit ist ein geringer Anteil der Datenbasis als Mischung aus Druckschrift und Schreibschrift zu klassifizieren. Alle Wörter sind markiert, d.h., daß die Anfänge eines neuen Buchstaben innerhalb des Wortes markiert sind.

Die Datenbasis wurde in zwei disjunkte Teilmengen geteilt, so daß die Trainingsdaten aus ca. 5300 Wörtern von 100 Schreibern und die Testdaten aus ca. 1000 Wörtern von 20 Schreibern stammen. Obwohl die Datenbasis relativ klein ist, reicht sie aus, um tendenzielle Aussagen zu machen.

Diese Daten wurden ursprünglich zum Zwecke der Online-Erkennung vom MIT veröffentlicht und liegen deshalb auch in einer Online-Repräsentation vor. Daher mußten die Daten zuerst in eine Bitmap Darstellung konvertiert werden. Die Online-Daten wurden jeweils in Bitmap der Höhe 100 Pixel konvertiert, was ca. 100 dpi entspricht und auch für Offline-Anwendungen realistisch ist. Dabei ist die Strichstärke frei wählbar. Für anfängliche Testzwecke wurde eine Strichstärke von 1 gewählt und beibehalten. Dies ist jedoch keine gravierende Vereinfachung des Versuchs. Insbesondere war bei der Konvertierung und der Vorverarbeitung darauf zu achten, daß die Marken der Wörter korrekt übernommen werden.

#### 5.1.2 Der Erkennung

Wie oben erwähnt wurde ein MS-TDNN zur Erkennung benutzt. Jeder Buchstabe wird durch drei States modelliert. Dies entspricht dem Anfang, der Mitte und dem Ende eines Buchstaben. Die Architektur besteht aus vier Schichten: einer Eingabeschicht mit 24 (Merkmalssatz 1) oder 30 (Merkmalssatz 2) Neuronen, zwei versteckte Schichten mit 100 und 120 Neuronen und einer Ausgabeschicht mit 156 Neuronen, was den definierten Buchstabenbereich multipliziert mit drei

States pro Buchstabe entspricht. Die Eingabeschicht hat 3 Verzögerungen, die 1. Versteckte Schicht hat ebenfalls 3 Verzögerungen und die 2. Versteckte Schicht hat 1 Verzögerung.

Das Training verläuft in 3 Schritten: Im ersten Schritt wird der Erkenner 4 Iterationen auf der State Ebene trainiert. Die jeweiligen States werden durch einen Viterbi-Algorithmus vorgegeben. Im zweiten Schritt werden nur noch die Marken der Buchstabenanfänge vorgegeben. Der Erkenner muß die States eigenständig für 6 Iterationen klassifizieren. Danach muß der Erkenner selbständig in der Lage sein, ohne Marken die States eines Wortes zu klassifizieren.

Mit Hilfe der ausgegebenen States muß nun in einem Standard-Unix Wörterbuch der Größe 50.000 ein passendes Wort gefunden werden. Dies geschieht mit einem Pruning Suchalgorithmus. Das Wörterbuch umfaßt sowohl Wörter mit Klein- als auch Großbuchstaben.

## 5.2 Vorversuch

Diese Kennzahlen stützen sich aus einer Untersuchung während meiner HIWI-Tätigkeit. Dabei wurde dieselbe Datenbasis benutzt. Jedoch wurden die Daten mit der NPen<sup>++</sup>-Vorverarbeitung aufbereitet und dann erst in ein Bitmap der Höhe 20 konvertiert. Dies entspricht also annähernd dem Merkmalssatz 1 ohne die lokal-statistischen Merkmale. Es wurden mehrere Netzwerk-Architekturen und Modellierungen untersucht, um die Erkennungsleistung zu steigern. Dabei erwies sich die oben beschriebene Architektur und Modellierung als am erfolgreichsten.

Unter den oben beschriebenen Voraussetzungen erhält man eine Erkennungsrate von 88,75%. Diese Erkennungsleistung ist durchaus als eine obere Schranke für die folgenden Versuche zu interpretieren. Dadurch daß in der Online-Vorverarbeitung nur mit Punktfolgen gearbeitet wird, können wichtige Vorverarbeitungsschritte besser umgesetzt werden als im Offline-Bereich. Auch die Tatsache, daß das Netzwerk gezielt auf diesen Merkmalssatz kalibriert wurde, läßt annehmen, daß andere Merkmalssätze auf diesem Netzwerk nicht dieselbe Leistung erzielen.

## 5.3 Versuche

	Erkennungsrate bei Wörterbuch 50k	Erkennungsrate bei Wörterbuch 25k
Vorversuch	88.7%	---
Merkmalssatz 1	88.2%	94.1%
Merkmalssatz 2	85.1%	92,4%

**Tab. 1: Ergebnisse**

Beim Merkmalssatz 1 wurde eine Bitmaphöhe von 20 gewählt, die sich schon bei dem oben beschriebenen Vorversuch bewährt hat. Mit den zusätzlichen 4 lokal-statistischen Merkmalen erhält man dann einen Merkmalsvektor der Dimension 24. Das Ergebnis von einer Erkennungsrate von 88.27% ist im Vergleich zum Vorversuch akzeptabel.

Beim Merkmalsatz 2 gibt es mehrere Parameter, deren Einstellung sehr sorgfältig vorgenommen werden muß. Nach einigen Vorabversuchen haben sich folgende Einstellung am besten geeignet: eine Fensterbreite von 5 Pixeln, eine Fensterschrittweite von 3 Pixel und ein Überlappungsanteil der Ausschnitte von 5%. Die Dimension des Merkmalsvektors beträgt konstant 30. Das Ergebnis von 83.74% Erkennungsleistung hinkt dem von Merkmalsatz 1 deutlich hinterher. Zu begründen ist dies vor allem durch einen sehr großen Merkmalsatz im Vergleich zum Vorversuch, deshalb sollte das Netzwerk speziell auf diesen Merkmalsatz kalibriert werden. Erwähnenswert ist auch, daß der Merkmalsatz 2 mehr von der Basislinienerkennung abhängt als der Merkmalsatz 1. So können hier schon geringe Fehlbestimmungen der Basislinien gravierende Änderungen der Merkmale bewirken.

Zusätzlich wurden die Versuche noch mit einem Wörterbuch mit 25.000 Einträgen wiederholt. Hierzu wurden zufällig Wörter aus dem ursprünglichen 50k Wörterbuch entfernt, unter der Bedingung, daß die Wörter aus den Testdaten im Wörterbuch erhalten bleiben. Diese Ergebnisse findet man in Tab. 1.

## 5.4 Bewertung und Ausblick

Obwohl Merkmalsatz 2 eine schlechtere Erkennungsrate hat als Merkmalsatz 1, werden für die Weiterentwicklung des Systems die geometrisch-statistischen Merkmale eine entscheidende Rolle spielen. Die topologischen Merkmale sind nur ein Abbild der Vorverarbeitung, der Erkennen muß interessante Zusammenhänge selbst erlernen. Mit der Verbesserung der Vorverarbeitung werden sich dann nur unwesentliche Steigerungen der Erkennungsleistung abzeichnen. Die geometrisch-statistischen Merkmale hingegen haben einen analytischen Charakter und fassen komplexere Informationen wie z.B. die Verbindung zwischen zwei Kanten in einem Merkmal zusammen. Eine genauere Merkmalsanalyse würde wahrscheinlich bei gleichbleibender Vorverarbeitung bessere Resultate hervorbringen. Ein wichtiger Punkt der gesamten Merkmalsgenerierung bleibt die Basislinienerkennung. Neue interessante Methoden hierzu wurden bereits veröffentlicht (siehe [SSE96]). Eine Verbesserung dieses Vorverarbeitungsschrittes würde sich sehr positiv auf die Erkennungsrate des Systems auswirken.

Um die Ergebnisse aus 5.3 zu bewerten, werden diese mit den Resultaten schon bestehender Systeme verglichen. Ein direkter Vergleich von Ergebnissen in der Literatur ist aber oft nicht möglich, da die Trainings- und Testbedingungen für die Erkennen stark abweichen. Im folgenden wird jedoch ein Vergleich mit [SSE96] durchgeführt, um die Qualität der Ergebnisse zu bewerten. Das Schweizer Institut für Technologie in Zürich benutzt eine ähnliche Vorverarbeitung wie die hier beschriebene und einen MS-TDNN Erkennen. Für Wörter in Druckschrift erzielte das Institut eine Erkennungsrate von 94,7% und für Wörter in Schreibschrift eine Erkennungsrate von 59,2% unter Verwendung eines Wörterbuchs mit 25.000 Einträgen. Da für die Druckschrift nur Großbuchstaben verwendet werden durfte, ist anzunehmen, daß damit der Schreibstil der Schreiber stark beeinträchtigt wurde. Dies resultiert in einer verhältnismäßig ordentlichen und nicht mehr natürlichen Druckschrift. Diese Annahme wird durch die große Kluft zwischen der Erkennungsrate von Druckschrift und Schreibschrift gestützt. Zusätzlich wurden nur 606 Wörter zum Testen des Erkenners benutzt, bei einer Trainingsdatenbank von ca. 25.000 Wörtern bei beiden Versuchen. Das hier vorgestellte System benutzte zum Training und Testen nur Wörter, die von Schreibern geschrieben wurden, die natürliche Druckschrift verwenden,

deshalb sind sowohl Groß- als auch Kleinbuchstaben in der Datenbasis enthalten. Aus diesem Grund ist es auch anzunehmen, daß dieses System nicht extrem schlechter auf Schreibschrift reagieren wird, da die Verbindung zwischen natürlicher Druckschrift und natürlicher Schreibschrift oftmals verschwindend gering ist.. Somit kann man trotz geringeren Erkennungsrate im Bereich der Druckschrift als in [SSE96], gestützt durch die unterschiedlichen Ausgangsbedingungen, behaupten daß dieser System unter gleichen Testbedingungen eine bessere Erkennungsrate erzielen würde.

## 6 Zusammenfassung

In dieser Studienarbeit wurde ein Handschriftenerkennungs-System präsentiert, das Offline-Einzelwörter schreiberunabhängig mit Erkennungsraten von über 94% erkennen kann. Die Vorverarbeitung und Merkmalsgenerierung wurde in das schon bestehende Online-System NPen<sup>++</sup> integriert und arbeitet davon unabhängig.

Für praktische Einsätze ist dieses System noch nicht geeignet, da die Erkennungsdauer bei einem 50k Wörterbuch von ca. 4,5 Sekunden für Merkmalsatz 1 und ca. 4,2 Sekunden für Merkmalsatz 2 noch zu langsam ist. Dies liegt vor allem an der Größe des Netzes und an der Tatsache, daß die Erkennung noch nicht auf Zeitaspekte optimiert wurde. Die ebenfalls noch nicht auf Zeit optimierte Vorverarbeitung hingegen benötigt nur ca. 0,4 Sekunden für Merkmalsatz 1 und 0,5 Sekunden für Merkmalsatz 2. Diese Messungen wurden auf einer DEC ALPHA AXP 3000/600 durchgeführt.

Eine Erweiterung dieses Systems auf Erkennung von sowohl Druck- als auch Schreibschrift sollte nur mit geringer Verschlechterung der Erkennungsleistung möglich sein, da durch die Abstandnormalisierung die Buchstaben bereits sehr dicht aneinander gereiht sind. Auch die Tatsache, daß die Schreiber in der Datenbasis natürliche Druckschrift benutzen und somit der Schritt zur Schreibschrift nicht mehr groß ist, unterstützt diese Vermutung.

Die Vorversuche während meiner HIWI-Tätigkeit könnten dazu benutzt werden, um die Erkennungsleistung der Online-Erkennung von NPen<sup>++</sup> zu steigern. Da in diesem Versuch das vorverarbeitete Online-Handschrift in eine Bitmap konvertiert wurde, wäre es möglich mit einem zusätzlichen Erkennen für die Offline-Handschrift, die Ergebnisse des Online-Erkenner zu validieren.



## 7 Literatur

- [BeCh94] Bengio und Y. Le Chun. „Word Normalization for On-Line Handwritten Word Recognition“, Proceedings of the ICPR-94, Jerusalem, October 1994
- [CGM93] T. Caesar, J.M. Gloger, E. Mandler. „Preprocessing and Feature Extraction for a Handwriting Recognition System“, Proceedings of the 2. Int. Conf. on Document Analysis and Recognition, Japan, October 1993
- [BCGM95] A. Braun, T. Caesar, J.M. Gloger, E. Mandler. „Preprocessing Raw Binary Images by Means of Contours“, Proceedings of the 3. Int. Conf. on Document Analysis and Recognition, Montreal, August 1995
- [BaBr82] D.H. Ballard, C.M. Brown, „Computer Vision“, Prentice-Hall, Inc., Englewood Cliff, New Jersey, 1982
- [Glog92] J. M. Gloger, „Use of the Hough transform to separate merged text / graphics in forms“, 11<sup>th</sup> Int. Conf. on Pattern Recognition, The Hague, The Netherlands, August 1992
- [WaDa83] K. Wall, P.-E. Danielsson: „A New Method for Polygonal Approximation of Digitized Curves“, Proc. of the 3<sup>rd</sup> Scandinavian Conf. on Image Analysis, 1983
- [CoCo82] C.H. Cox, P. Coueignoux, B. Blesser, M. Eden, „Skeletons: A Link Between Theoretical and Physical Letter Descriptions“, Pattern Recognition, Vol. 15, No. 1, 1982, pp 1122
- [Kwok88] P.C.K. Kwok, „A Thining Algorithm by Contour Generation“, Communication of the ACM, Vol. 31, No. 11, pp 1314-1324, 1988
- [Jäger96] Stefan Jäger, „Recovering Writing Traces in Off-Line Handwriting Recognition: Using a Global Optimization Technique“, Proc. of the ICPR'96, 1996
- [HaWa92] P. Haffner und A. Waibel, „Multi-State Time Delay Neural Networks for Continuous Speech Recognition“, Advances in Neural Information Processing Systems (NIPS-4), Morgan Kaufman, 1992
- [MFW95] S. Manke, M. Finke, A. Waibel, „NPen<sup>++</sup>: A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System“, Proc. of the Int. Conf. on Document Analysis and Recognition, Montreal, August 1995
- [SSE96] R. Seiler, M. Schenkel, F. Eggimann, „Cursive Handwriting: Off-line versus On-line Recognition“

**Studienarbeit**

# **Methoden der Merkmalsgenerierung bei Offline-Handschriften**

**12.05.1997**

**Thomas Burkhardt**

---

## **Übersicht des Offline-Erkennungssystem**

---



# Online versus Offline

---

- Eingabe
  - Online: Eingabe-Tablett
  - Offline: Scanner
- Speicherung
  - Online: Koordinatensequenz
  - Offline: Bitmap
- Anwendungen
  - Online: Mensch-Maschine Dialog
  - Offline: z.B. Adressen Erkennung
- Erkennungsleistung
  - Online-Erkennung der Offline-Erkennung überlegen
- Problematik der Offline Erkennung
  - dynamische Schreibinformationen nicht vorhanden

# Einleitung

---

- Aufgabe:
  - Entwicklung eines Offline-Einzelwort Erkenners, der als Eingabe Online-Daten erhält
  - insbesondere die Entwicklung der
    - Offline-Vorverarbeitung
    - Merkmalsgenerierung
- Ziel:
  - Verbesserung des Online-Erkennungssystems NPen<sup>++</sup> durch einen Offline-Erkenner

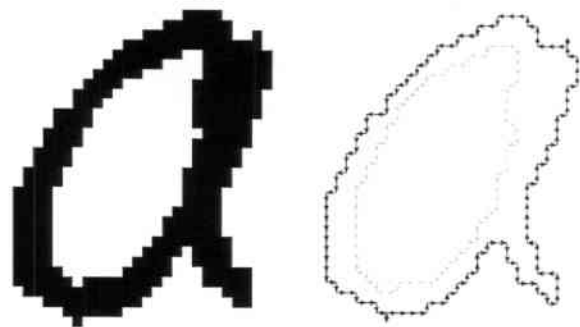
# Übersicht Vorverarbeitung

---

Online-Daten

acrylic

Konvertierung in Bitmap



Erstellung der BCCs

Basisliniennormierung

acrylic

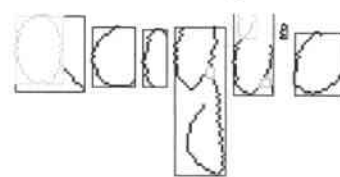
Scherwinkelnormierung

acrylic

Skelettierung

acrylic

Abstandnormalisierung



Approximation

acrylic

# Konvertierung

---

Konvertierung der zur Verfügung stehenden Online-Daten in ein Bitmap

- Strichstärke und Höhe der Bitmap sind frei wählbar
- Breite der resultierenden Bitmap wird durch Skalierungsfaktor von Bitmaphöhe und Online-Handschriften Höhe bestimmt

acrylic acrylic

# Basisliniennormierung

---

## Normierung der Lage einer Handschrift

- Bestimmung der lokalen Extrema anhand der BCCs
- Erweiterung des Algorithmus von Bengio und Le ~~Chun~~ ermittelt die Basislinien (obere Basislinie, Coreline, Baseline, untere Basislinie) anhand der lokalen Extrema einer Handschrift.
- Normierung der Handschrift durch Rotation oder Zerrung



# Scherwinkelnormierung I

---

Beseitigung der Schreibneigung einer  
Handschrift

- Approximation der Handschrift durch  
Liniensegmente, um statistische Erhebung  
der Neigung durchzuführen



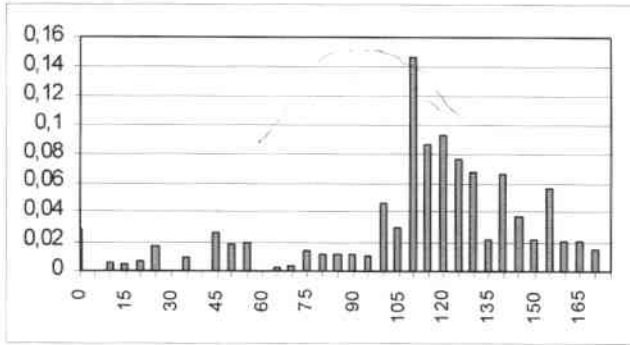
The image shows the word "acrylic" written in a cursive, handwritten style. Each letter is overlaid with a series of black line segments that approximate its shape. This visualizes the process of digitizing handwriting by breaking it down into straight line segments for statistical analysis.



# Scherwinkelnormierung II

---

- Eintragung der Kanten, gewichtet durch ihre Länge, in ein Histogramm von  $0^\circ$ - $180^\circ$



- Ermittlung des Scherwinkels durch verschiedene Strategien möglich:
  - straight maximum
  - soft maximum
  - low rise maximum
  - weighted maximum

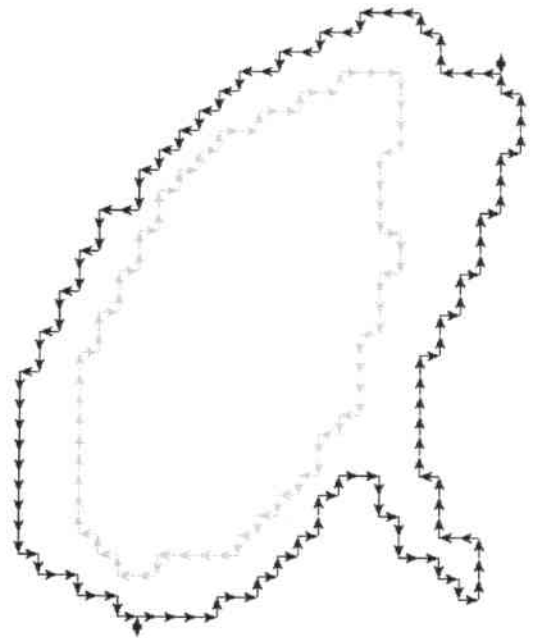
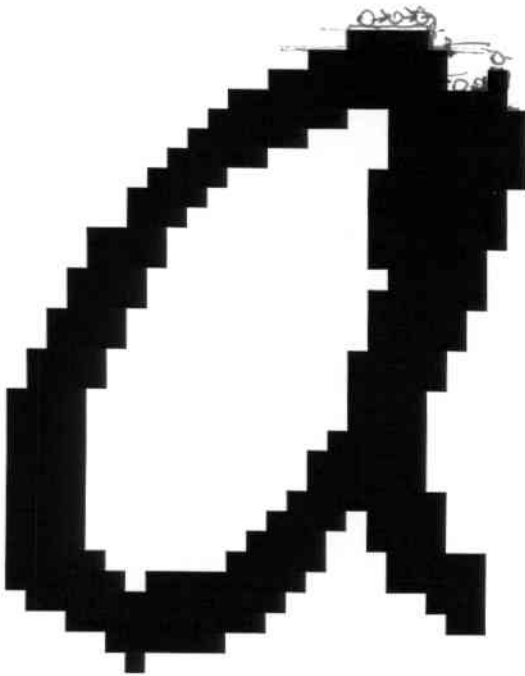


# Binary Connected Components (BCCs)

---

Effektiver Datentyp zur Speicherung von Handschriften die als Bitmap vorliegen

- Jede BCC repräsentiert einen geschlossenen Konturverlauf als zweifach zusammenhängenden Graphen
- Die Menge der Pixel zwischen einer äußeren und inneren Kontur nennt man Zusammenhangsgebiet.

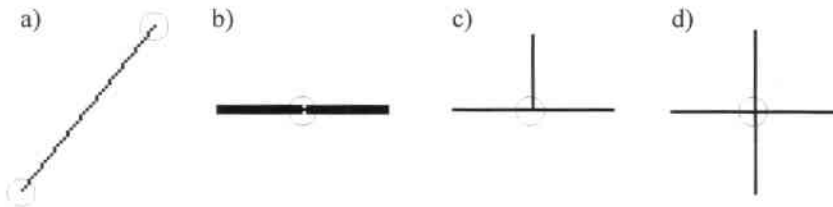


# Skelettierung

---

Normierung der Schriftstärke auf einen Pixel

- Iterativer Algorithmus von Kwok:
  - klassifiziert Pixel nach Endpunkt (a), Schnittpunkt (b)-(d) und nicht-sicheren Pixel



- erstellt in jeder Iteration eine neue Kontur, so daß die Handschrift „dünner“ wird
- Resultierendes Skelett besteht nur aus Schnitt- und Endpunkten, die interessante Merkmale darstellen



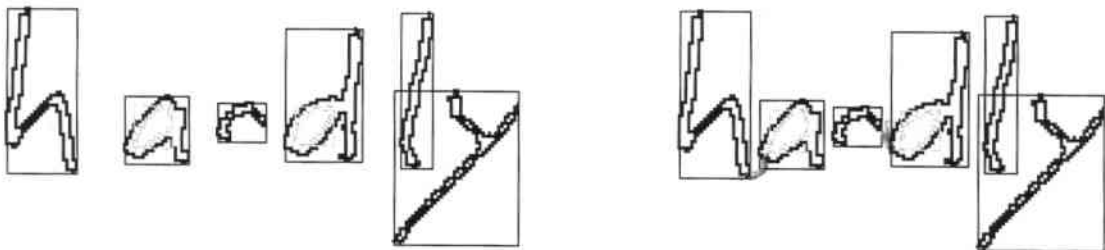
# Abstandnormalisierung

---

Abstände zwischen isolierten Buchstaben normieren

- Reduktion der Abstandnormalisierung zwischen Buchstaben auf Zusammenhangsgebiete
  - Entscheidung anhand Bounding Box einer BCC, ob Zusammenhangsgebiete benachbart.
  - Verschachtelte Zusammenhangsgebiete können nicht normiert werden

⇒ Unterschied zwischen Schreib- und Druckschrift wird geringer



# Approximation

---

## Approximation der skelettierten Handschrift durch Liniensegmente

- Eigene Erweiterung des Algorithmus von Wall und Danielsson approximiert die Handschrift unter Einhaltung wichtiger Merkmale:

Schnitt- und Endpunkte sind Endpunkte eines Liniensegments

⇒ Keine Zerstörung des Skeletts

⇒ Geometrische Darstellung eines Skeletts durch Liniensegmente

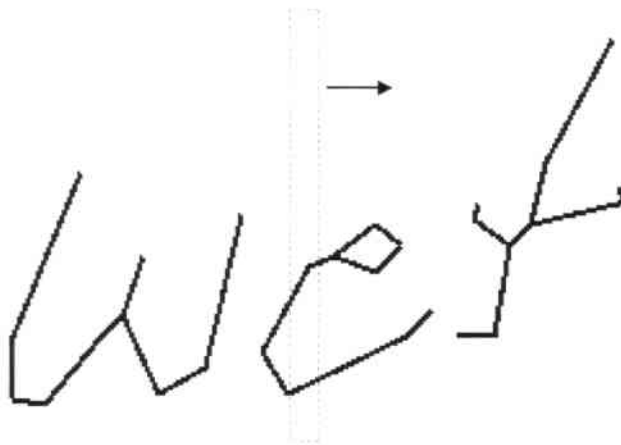


# Merkmalsgenerierung

---

Generierung von Merkmalsvektoren abhängig von Breite der Handschrift zur Anwendung auf einem MS-TDNN

- Prinzip:
  - Ausgangspunkt: approximierte Handschrift
  - Generierung der Merkmale aus einem fest definierten Fenster
  - Generierung von zusätzlichen lokal-statistischen Merkmalen: Endpunkte, Schnittpunkte, usw.
  - Bewegung des Ausschnitts entlang der x-Achse

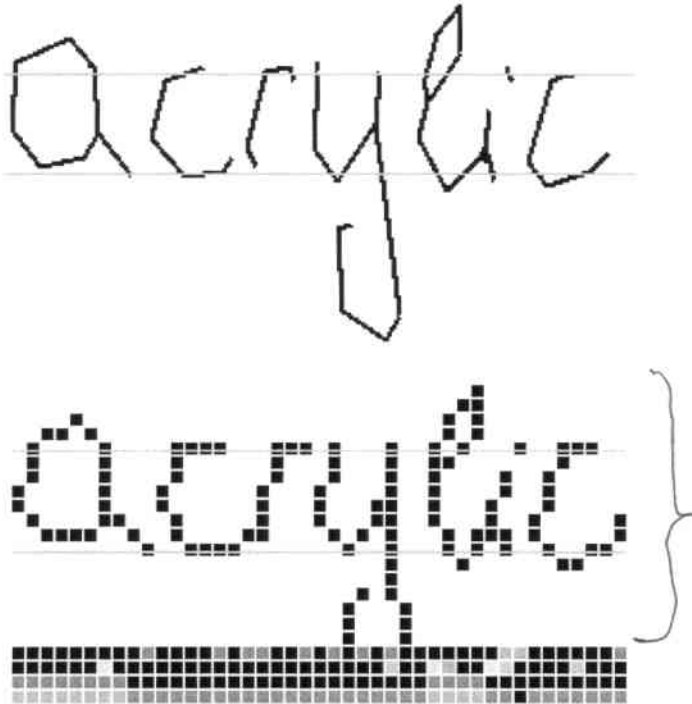


# Topologische Merkmale

---

Bitmaprepräsentation der Handschrift stellt Topologie dar

- Konvertierung der Handschrift in Bitmap fester Höhe (z.B. 20 Pixel)
- Dabei wird die Handschrift so skaliert, daß die Basislinien in der Bitmap fest sind
- Jede Spalte der Bitmap repräsentiert zusammen mit den lokal-statistischen Merkmalen den Merkmalsvektor



# Geometrisch-statistische Merkmale

---

Analyse der Kanten der approximierten  
Handschrift nach Lage und Verbindung

- Ausschnitt mit Höhe der Handschrift enthält zu viel Kanten
- ⇒ Einteilung der Handschrift in 5 Gebiete
- Klassifikation in jedem Gebiet nach
    - Lage jeder Kante (horizontal, vertikal, diagonal)
    - Verbindung zweier Kanten (Bogen, Ecke, Spitze)
  - Anteile werden in jedem Gebiet bestimmt
- ⇒ Geometrisch-statistische Merkmale



# Experimente

---

- Trainings- und Testdaten
  - Natürliche Druckschrift mit Groß- und Kleinbuchstaben von 120 Schreibern
  - Disjunkte Trainings- (5300 Wörter von 100 Schreibern) und Testmengen (1000 Wörter von 20 Schreibern)
- Erkenner: MS-TDNN
- Wörterbuch
  - Standard Unix Wörterbuch mit 50000 Einträgen
  - Halbiertes Unix Wörterbuch mit 25000 Einträgen
- Erkennungsraten

	Wörterbuch 50k	Wörterbuch 25k
Topologisch	88.3%	94.1%
Geometrisch	85.2%	92,4%
Online	91,1%	-----

# Zusammenfassung

---

- Merkmalsgenerierung stark von Basislinien abhängig
- Weitere Entwicklung der topologischen Merkmale aussichtsreich
- Erkennungsleistung für weitere Arbeiten vielversprechend
  
- Ausblick:
  - Experimente mit Druck- und Schreibschrift
  - Einfache Umstellung auf pure Offline-Erkennung
  - Erweiterung der Vorverarbeitung