

BMFT-Verbundprojekt: HYBRID
Teilvorhaben: GEHA

Förderkennzeichen: 413-4001 01 IB 405 B/5

Abschlußbericht

Dipl.-Inf. Stefan Jäger

Daimler-Benz AG

Forschung und Technik

Postfach 2360

89013 Ulm

Telefon: (0731) 505-2345

E-Mail: jaeger@dbag.ulm.DaimlerBenz.COM

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Forschung und Technologie unter dem Förderkennzeichen 01 IB 405 B5 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Inhaltsverzeichnis

1	Aufgabenstellung	3
2	Voraussetzungen	4
3	Planung und Ablauf	5
4	Wissenschaftlicher und technischer Stand bei Projektbeginn	5
4.1	Benutzte Hilfsmittel und Verfahren	5
4.2	Literatur	6
5	Zusammenarbeit	6
6	Erzielte Ergebnisse	6
6.1	1. Arbeitsphase	6
6.2	2. Arbeitsphase	7
6.3	3. Arbeitsphase	9
6.3.1	Einleitung	9
6.3.2	Abbildung auf ein genetisches Optimierungsproblem (TSP)	10
6.3.2.1	Einlesen des Buchstabens	10
6.3.2.2	Skelettierung	10
6.3.2.3	Erzeugung des Line-Graphen	10
6.3.2.4	Zuweisung von Gewichten (Fitnessberechnung)	10
6.3.2.5	Vervollständigung des Line-Graphen	13
6.3.2.6	Einführung eines zusätzlichen Knotens	15
6.3.2.7	Genetische Optimierung	16
6.3.2.8	Ableiten der Kantenfolge	16
6.3.3	Verkleinerung des Suchraumes der genetischen Optimierung	16
6.3.4	Lösung des TSP-Problems	18
6.3.4.1	Verfahren zur Lösung	18
6.3.4.2	Mathematischer Rahmen	18
6.4	Resultate	19
6.4.1	Rekonstruierte Schriftzüge	19
6.4.2	Klassifikationsergebnisse	22
6.5	Weitere Arbeiten	22
7	Voraussichtlicher Nutzen	22
8	Fortschritte von dritter Seite	24
8.1	Verwandte Arbeiten	24
8.2	Verwandte Forschungsgebiete	25
9	Veröffentlichungen und Schutzrechtsanmeldungen	26

1 Aufgabenstellung

Das geförderte Projekt “GeHa” ist ein Teilprojekt des geförderten Verbundprojektes “Hybrid”. Hybrid beschäftigt sich mit der Anwendung paralleler genetischer Algorithmen in der kombinatorischen Optimierung und ist selbst ein Teilprojekt im größeren Fördergebiet “Bioinformatik”. Das Ziel von GeHa ist die Anwendung genetischer Algorithmen zur Erkennung gebundener Handschrift. Der Schwerpunkt von GeHa liegt in der Rekonstruktion des Schriftzuges, die den weitaus größten Anteil am gesamten Arbeitspaket von GeHa hat.

In der Handschrifterkennung wird zwischen Off-line- und On-line-Erkennung unterschieden:

- Die On-line-Erkennung ist dadurch gekennzeichnet, daß sie zur Erkennung von handgeschriebenen Wörtern dynamische Information einsetzen kann. Dynamische Information besteht hierbei z.B. aus der zeitlichen Reihenfolge der geschriebenen Striche, der Geschwindigkeit und Druckstärke während des Schreibens. Zusätzlich kann Information über den Zustand des Schreibgerätes während des Schreibens abgefragt werden, z.B. ob und wann das Schreibgerät Berührung mit dem Papier hat. Zur Erfassung der dynamischen Information wird in der Regel spezielle Hardware benötigt. Dies beschränkt die Anwendungsmöglichkeiten der On-line-Erkennung. Eine in jüngster Zeit häufige Anwendungsmöglichkeit für On-line-Erkennung sind Personal Digital Assistants (PDAs), die eine Art elektronisches Notizbuch darstellen. Neuere Produkte sind z.B. der “Newton” von Apple oder der “Pilot” von US Robotics.
- In der Off-line-Erkennung ist a-priori keine dynamische Information verfügbar. Die Eingabe für einen Off-line-Handschrifterkennung besteht daher nur aus dem statischen Bild des geschriebenen Wortes. Ein typisches Beispiel für die Off-line-Erkennung von Handschrift ist die Postautomatisierung. Hier werden Adressen von Briefen gescannt und diese mit Hilfe der Off-line-Erkennung automatisch an den richtigen Zielort weitergeleitet. Dies vermindert den Aufwand für das mühselige Lesen von Anschriften durch Menschen. So war die mögliche Leistungssteigerung des Erkennungssystems im Anschriftenleser der AEG-Electrocom die Hauptmotivation für die Beantragung des geförderten Projekts [CGM93a, CGM93b, KCGM93, CGKM94, CGKM93].

Aufgrund der Tatsache, daß die On-line-Erkennung dynamische Information zur Erkennung einsetzen kann, sind ihre Erkennungsraten höher als die der Off-line-Erkennung. Eine naheliegende Überlegung ist daher, die dynamische Information aus einem Off-line-Bild zu extrahieren und damit die Off-line-Erkennungsraten zu verbessern. Dies soll gerade durch die Rekonstruktion des Schriftzuges,

d.h. durch das geförderte Projekt, erreicht werden. Zur Erleichterung der Aufgabe wird die dynamische Information innerhalb dieses Projektes gleichgesetzt mit Information über die zeitliche Reihenfolge von Bestandteilen des Wortes. Geschwindigkeit und Druckstärke soll also nicht extrahiert werden.

Zur Rekonstruktion des Schriftzuges mit Hilfe von genetischen Algorithmen muß zunächst eine geeignete Problemrepräsentation und ein problemspezifisches Optimierungskriterium gefunden werden. Die Wahl der Problemrepräsentation wird hierbei praktisch durch die Skelettierung des Wortes vorgegeben, die einen wichtigen Vorverarbeitungsschritt im Erkennungsprozeß darstellt. Das Skelett von einem Wort kann im mathematischen Sinne als Graph mit Knoten und Kanten aufgefaßt werden. Die Knoten repräsentieren die Kreuzungspunkte im Schriftzug, und die Kanten repräsentieren die eigentlichen Schreiblinien. Diese Repräsentation bietet eine mathematische Grundlage für die Rekonstruktion des Schriftzuges und erlaubt eine Formalisierung der Aufgabe. Einige der existierenden Ansätze haben Probleme mit dieser Formalisierung (z.B. [DR92, DR95]). Im Rahmen des Projekts war im wesentlichen das Optimierungskriterium zu bestimmen. Die eigentliche Optimierung sollte dann mit genetischen Algorithmen geschehen.

Neben der Rekonstruktion des Schriftzuges wurden im Projektantrag von GeHa auch zwei kleinere Teilaufgaben angegeben, die vom Daimler-Benz Forschungszentrum Ulm bearbeitet werden sollten. Die erste Aufgabe betraf die Auswahl geeigneter Merkmale durch einen genetischen Algorithmus. In der zweiten Aufgabe sollten optimale Diskriminanzfunktionen bestimmt werden, indem die Koeffizienten für einen Polynomklassifikator mit Hilfe genetischer Algorithmen bestimmt werden.

2 Voraussetzungen

Der Arbeitsumfang des geförderten Projektes wurde auf drei Jahre ausgelegt. Im Forschungszentrum Ulm konnte auf Methoden und Algorithmen des Handschrifterkenners im Erkennungssystem der AEG-Electrocom zurückgegriffen werden. Handgeschriebenes Datenmaterial von amerikanischen Städtenamen wurde von der AEG-Electrocom bereitgestellt. Aufgrund seiner Erfahrungen auf dem Gebiet der genetischen Algorithmen war Herr Mühlenbein von der GMD und seine Arbeitsgruppe der fachliche Ansprechpartner während der Durchführung des Projektes.

3 Planung und Ablauf

Im wesentlichen stimmte die Planung des Projektes mit der Durchführung überein. Der genaue zeitliche Ablauf des Projektes kann auch den einzelnen Zwischenberichten entnommen werden. Abgesehen von einem verspäteten Start des Projektes konnten alle wichtigen Meilensteine eingehalten werden. Allerdings wurden die beiden kleineren Arbeitspakete der Rekonstruktion des Schriftzuges untergeordnet. Die Extraktion der Merkmale wurde im Zusammenhang mit der Schriftzugrekonstruktion bearbeitet. Das Arbeitspaket zur Bestimmung von Diskriminanzfunktionen wurde zugunsten der aussichtsreichen Schriftzugrekonstruktion aufgegeben.

Die zeitliche Abfolge der erzielten fachlichen Resultate kann in einem folgenden Abschnitt, welcher die fachlichen Ergebnisse enthält, nachgelesen werden.

4 Wissenschaftlicher und technischer Stand bei Projektbeginn

4.1 Benutzte Hilfsmittel und Verfahren

Zur Durchführung des Projektes wurde sehr stark auf vorhandene Programmbibliotheken zurückgegriffen. Die meisten der verwendeten Prozeduren wurden im Daimler-Benz Forschungszentrum Ulm entwickelt und bilden einen wesentlichen Bestandteil der Vorverarbeitung. So wurden auch fast alle Verarbeitungsschritte der Vorverarbeitung in das entwickelte Programmpaket übernommen, z.B.:

- Filterung,
- Konturanalyse,
- Skelettierung.

Eine Ausnahme bildet die Schernormierung, die nicht verwendet wurde, um eine unnötige Verzerrung der Winkel zu verhindern. Als weiteres Hilfsmittel wurde ein Programmpaket von R.J.Craig von AT&T Bell Labs verwendet, das zur Lösung des Traveling Salesman Problems benutzt wurde [Cra]. Alle entwickelten Algorithmen wurden in der Programmiersprache "C" implementiert. Visualisierungen und Oberflächen wurden unter dem Betriebssystem NextStep erstellt, das sich als sehr leistungsfähig erwiesen hat. Zur Aufstellung von Regeln, die ein Ablaufen eines skelettierten Wortes ermöglichen, wurde das Programmpaket "C4.5" von Herrn Quinlan verwendet [Qui93]. Es handelt sich hierbei um ein symbolisches Lernverfahren, mit dem eine Regelbasis erstellt werden kann.

4.2 Literatur

Zur Durchführung des Projektes wurde auf eine große Anzahl von Literaturstellen zurückgegriffen. Diese werden in diesem Bericht an den entsprechenden Stellen erwähnt und anschließend am Ende des Berichts zusammenfassend aufgelistet.

5 Zusammenarbeit

Die Universität Karlsruhe wurde mit einem wichtigen Beitrag in das Projekt eingebunden. So wurden die rekonstruierten On-line-Daten in Karlsruhe als Eingabe für einen On-line-Erkennen benutzt und mehrere erfolgversprechende Testläufe durchgeführt (Stefan Manke, Arbeitsgruppe von Prof. Waibel). Daneben wurde eine Diplomarbeit von der Universität Karlsruhe am Forschungszentrum Ulm betreut.

6 Erzielte Ergebnisse

In diesem Abschnitt werden die erzielten Ergebnisse chronologisch dargestellt und bewertet. Im Verlauf des Projektes wurden verschiedene Ansätze entwickelt und zum Teil wieder verworfen. Durch die Erfahrungen mit diesen verschiedenen Ansätzen konnte jedoch eine elegante Methode entwickelt werden, die es erlaubt, mit Hilfe eines einfachen Optimierungskriteriums den Schriftzug zu rekonstruieren. Wir beginnen mit der Beschreibung der ersten Arbeitsphase.

6.1 1. Arbeitsphase

Das Ziel der ersten Arbeitsphase war, die Rekonstruktion des Schriftzuges durch einen Satz von Regeln zu steuern. Diese Regeln sollten durch einen genetischen Algorithmus generiert und verbessert werden. Dies hätte eine einmalige Erstellung und Optimierung der Regelbasis bedeutet. Die Regeln beschrieben eine lokale Umgebung um den aktuellen Knoten. Der aktuelle Knoten war dabei der Knoten, der durch das Ablaufen des Skeletts, unter Anwendung der Regeln, bisher erreicht worden war. Für die Synthese der Regeln wurden verschiedene Bausteine bereitgestellt. Diese Bausteine sollten helfen, die gegenwärtige Situation in der lokalen Umgebung des aktuellen Knotens zu beschreiben. Die folgenden Bausteine wurden zum Aufbau der Regelbasis verwendet:

- der Grad eines Knotens,
- die Länge einer Kante,
- das Länge-Breite-Verhältnis einer Kante,
- die Krümmung einer Kante,

- die Anzahl der bisherigen Durchläufe durch eine Kante,
- die Richtungen der bisherigen Durchläufe,
- der momentan erreichte Knoten,
- die Position der Kanten an den jeweiligen Knoten.

Als Regeln wurden einfache Produktionsregeln verwendet, die mit dem symbolischen Lernverfahren "C4.5" erzeugt wurden [Qui93]. Der Bedingungsteil der Regeln bestand dabei jeweils aus einer Konjunktion von Bedingungen. Der Aktionsteil der Regeln gab an, mit welcher Kante am aktuellen Knoten der bisherige Schriftzug fortzusetzen ist. Zur Identifizierung der Kanten am aktuellen Knoten wurden die Kanten am aktuellen Knoten in einer festen Ordnung aufgezählt. Zur Erstellung der Regelbasis wurde das System mit echten On-line-Daten trainiert, die uns freundlicherweise von der Uni Karlsruhe (Prof. Waibel) zur Verfügung gestellt wurden. Die On-line-Daten ermöglichten die Erstellung einer Regelbasis, die sich an bekannten Schriftzügen orientiert. Dazu wurden die On-line-Daten zuerst in Off-line-Daten umgewandelt, indem entlang des Schriftzuges die Pixel in einem Binärbild schwarz gefärbt wurden. Dann wurde auf den erzeugten Off-line-Bildern die Skelettierung durchgeführt. Da auf den anschließend vorliegenden Skeletten die zeitliche Information in Gestalt der On-line-Daten vorliegt, ist die Reihenfolge der Kanten des Skeletts im wesentlichen bekannt. Diese bekannten Schriftzüge wurden zur Initialisierung der Regelbasis verwendet. Leider zeigte es sich, daß die erzeugte Regelbasis sehr groß und somit nur schwer handhabbar ist. Eine genetische Optimierung hatte somit nur bedingte Aussichten auf Erfolg und wurde daher nicht durchgeführt. Auch wurde erkannt, daß zur Formalisierung des Problems ein strukturelles Grundgerüst notwendig ist, das die Anwendung von Regeln unterstützt. Diese Erkenntnis beendete die 1. Arbeitsphase und gab Anlaß zu einer Erweiterung in der 2. Arbeitsphase.

6.2 2. Arbeitsphase

In der 2. Arbeitsphase wurde damit begonnen, die Rekonstruktion des Schriftzuges als graphentheoretisches Problem aufzufassen. Das Skelett eines Wortes ist in den meisten Fällen identisch mit der Mittellinie der Schreiblinie und kann durch einen Graphen mit Knoten und Kanten mathematisch repräsentiert werden. Damit kann die Schriftzugrekonstruktion als Pfadsuche formuliert werden. Es muß der Pfad durch das Wortskelett gesucht werden, der jede Kante mindestens einmal durchläuft und in gewisser Weise optimal ist. In der zweiten Arbeitsphase wurde die Pfadsuche als Längenminimierung definiert. Es wurde der Pfad durch das Skelett gesucht, der jede Kante mindestens einmal durchläuft und minimale Länge hat. Die Länge eines Pfades ist definiert als die Summe der Länge seiner Kanten, die er durchläuft. Diese Formulierung des Problems ist identisch mit

dem Chinese Postman Problem in der Graphentheorie [GM84, BM76]. In diesem Problem muß ein Postbote jede Straße einer Stadt (eines Graphen) mindestens einmal ablaufen. Natürlich will er dabei so wenig Zeit wie möglich verbrauchen, d.h. eine möglichst kurze Strecke zurücklegen. Mathematisch läßt sich dieses Problem lösen, indem zuerst ein Minimum Perfect Matching ([Chr77]) zwischen allen ungeraden Knoten eines Graphen bestimmt wird und anschließend die Kanten entlang der kürzesten Pfade, welche die gepaarten Knoten verbinden, verdoppelt werden [Neu88, EJ73]. Der gesuchte Pfad läßt sich dann einfach mit einem Algorithmus von Euler finden [Neu88, GM84]. Der Algorithmus von Euler hat eine polynomiale Komplexität und enthält einige Parameter, die nicht fest vorgegeben sind. Das heißt, daß es mehrere Pfade gibt, die jede Kante mindestens einmal durchlaufen und minimale Länge besitzen. Durch die Anwendung der in Arbeitsschritt 1 hergeleiteten Regeln läßt sich der regelkonforme Schriftzug mit minimaler Länge herausfinden.

Obwohl dieser Ansatz zu einem, im Hinblick auf die Strichlänge, optimalen Ablauf des Wortskeletts führte, tauchten doch einige Kritikpunkte auf. Erstens mußte immer noch eine komplizierte Regelbasis optimiert werden. Zweitens lassen sich durch die Minimierung der Länge nicht alle Schriftzüge rekonstruieren. Dies liegt daran, daß das Verfahren zur Lösung des Chinese Postman Problems garantiert, daß jede Kante des Skeletts höchstens zweimal abgelaufen wird. Ein typisches Beispiel für einen Schriftzug, der sich nicht mit der Längenminimierung rekonstruieren läßt, ist das Skelett des Buchstabens “k” in Abbildung 1. In diesem

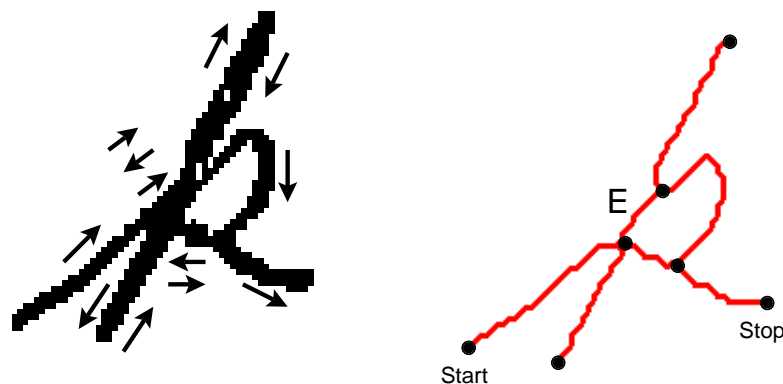


Abbildung 1: Eine dreifach durchlaufene Kante.

Beispiel muß die Kante “E” dreimal durchlaufen werden. Der korrekte Schriftzug ist durch Pfeile im linken Teil der Abbildung angedeutet.

Auf der Suche nach einer graphentheoretischen Lösung, die prinzipiell das Ablaufen von allen Schriftzügen ermöglichte, wurde schließlich im dritten Ar-

beitsschritt die Minimierung der Krümmung verwirklicht. Die Minimierung der Krümmung erlaubt die Rekonstruktion des Schriftzuges mit Hilfe eines einfachen Optimierungskriteriums.

6.3 3. Arbeitsphase

6.3.1 Einleitung

Die dritte Arbeitsphase soll nun ausführlicher dargestellt werden. In ihr wurden die wesentlichen Ergebnisse des geförderten Projekts erarbeitet. Sie stellt ein einfaches Optimierungskriterium bereit, das sich direkt zur Schriftzugrekonstruktion verwenden läßt. In Anlehnung an den zweiten Arbeitsschritt, wird das Problem der Schriftzugrekonstruktion auf ein graphentheoretisches Problem abgebildet, das sich für die genetische Optimierung besonders gut eignet. Es handelt sich hierbei um das Traveling Salesman Problem (TSP), das ein sehr bekanntes Problem in der Informatik ist. Das Traveling Salesman Problem beschreibt die Situation eines Handlungsreisenden, der bestimmte Kunden in verschiedenen Städten besuchen muß und wieder zu seinem Ausgangspunkt zurückkehren will. Er möchte jede Stadt nur einmal besuchen und seine insgesamt zurückgelegte Wegstrecke (oder seine Reisekosten) sollte so kurz (oder so niedrig) wie möglich sein.

Die folgenden fachlichen Ausführungen befassen sich mit der Abbildung des Rekonstruktionsproblems auf das Traveling Salesman Problem. Diese Abbildung hat zwei wichtige Vorteile:

- Erstens ist das TSP sehr gut für die genetische Optimierung geeignet [Gol89].
- Zweitens ist das Gütekriterium (Fitnessfunktion) beim TSP einfach und klar definiert. Die Kosten sind die Längen der Rundreisen. Kürzere Rundreisen besitzen eine höhere Fitness.

Ein Ergebnis der bisher durchgeführten Arbeiten ist die Tatsache, daß die Krümmung bzw. die Krümmungsabweichung eine sehr wichtige Rolle bei der Rekonstruktion des Schriftzuges spielt. Stehen an einem Knoten des Wortskeletts mehrere Folgekanten zur Auswahl, so setzt der menschliche Schreiber sehr häufig mit der Kante fort, die ihm die Fortsetzung des Schreibzuges mit einer möglichst geringen Abweichung von der bisherigen Richtung erlaubt. Um der Wichtigkeit der Krümmung Rechnung zu tragen, wird sie als Fitness für eine genetische Optimierung herangezogen. Es wird angenommen, daß der krümmungsminimale Pfad durch das Wortskelett identisch mit der menschlichen Schreibweise ist. Technisch läßt sich die Minimierung der Krümmung durch eine Abbildung auf das TSP lösen. Im folgenden sollen an einem Beispiel die für die Abbildung notwendigen Arbeitsschritte aufgelistet und erklärt werden. Aus Gründen der Übersichtlichkeit wird als Beispiel kein Wort, sondern der Buchstabe "B" benutzt.

6.3.2 Abbildung auf ein genetisches Optimierungsproblem (TSP)

6.3.2.1 Einlesen des Buchstabens Im ersten Verarbeitungsschritt wird der Buchstabe (das Wort) mit Hilfe eines Scanners eingelesen. Das Ergebnis ist ein Binärbild, das aus weißen und schwarzen Pixeln besteht. Abbildung 2 zeigt ein eingelesenes “B”.



Abbildung 2: Das Binärbild des Buchstabens “B”.

6.3.2.2 Skelettierung Der nächste Schritt ist die Skelettierung des Binärbildes. Der Algorithmus zur Skelettierung versucht die mittlere Schreiblinie zu finden. Das Ergebnis der Skelettierung ist das Wortskelett [SW94]. Das Wortskelett ist ein Graph mit Knoten und Kanten. In Abbildung 3 ist dieser Graph gezeigt. Die Knoten des Skeletts sind als schwarze Punkte eingezeichnet. Kanten sind mit den Buchstaben *A* bis *H* gekennzeichnet.

6.3.2.3 Erzeugung des Line-Graphen Anschließend wird der ‘Line-Graph’ aus dem Wortskelett erzeugt [GM84]. Im Line-Graph ist jede Kante des Wortskeletts durch einen Knoten symbolisiert. Sind zwei Kanten im Wortskelett angrenzend, so werden die entsprechenden Knoten im Line-Graphen verbunden. In der Abbildung 4 ist das Wortskelett zusammen mit dem daraus resultierenden Line-Graphen abgebildet. Die Abbildung 5 zeigt nur den Line-Graph. Der Line-Graph bietet eine Möglichkeit, die Kosten, die beim Übergang zwischen Kanten entstehen, formal zu repräsentieren. Das Wortskelett ist dazu ungeeignet, da hier nur den Kanten Kosten zugeordnet werden können und Übergänge zwischen Kanten unberücksichtigt bleiben.

6.3.2.4 Zuweisung von Gewichten (Fitnessberechnung) Im jetzt folgenden Schritt werden die Gewichte der Kanten des Line-Graphen berechnet.

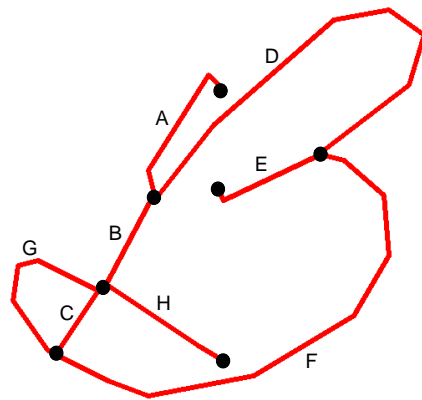


Abbildung 3: Das Skelett des Buchstabens "B".

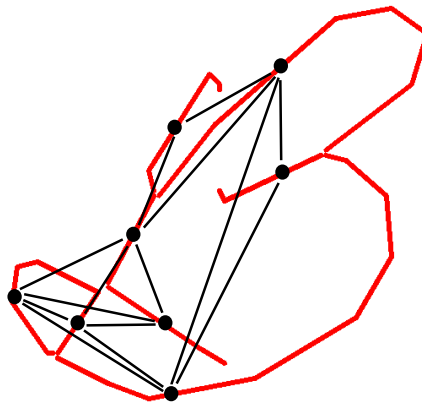


Abbildung 4: Das Wortskelett und der Line-Graph.

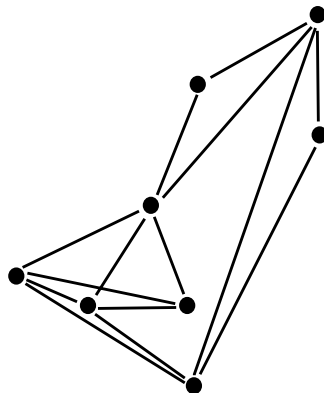


Abbildung 5: Der abgeleitete Line-Graph.

Sie bilden die Grundlage für spätere Berechnungen der Fitness. In Abbildung 7 werden die Kosten im Line-Graphen mit c_i bezeichnet. Die Kosten zwischen zwei angrenzenden Kanten im Wortskelett sind die Winkel zwischen den Kanten. Je größer die Winkelabweichung zwischen den Kanten, desto größer die Kosten. Die Berechnung der Winkel ist in Abbildung 6 verdeutlicht. Der Mittelpunkt des

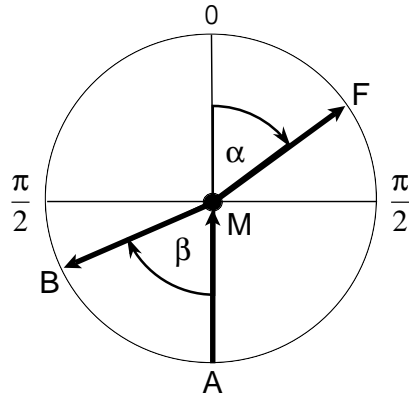


Abbildung 6: Die Berechnung der Winkel.

Kreises in Abbildung 6 soll einen Knoten des Skeletts repräsentieren. Der Pfeil von A nach M symbolisiert das Ende einer Kante des Knotens. Die Richtung des Pfeils beschreibt hierbei die Schreibrichtung. Angenommen der Schreiber hat mit dem bisherigen Schreibverlauf den Mittelpunkt des Kreises in Abbildung 6 erreicht, d.h. das Schreibgerät ist momentan auf dem Mittelpunkt M des Kreises positioniert. Wird nun der Schriftzug auf einer anderen Kante in Richtung des Punktes F fortgesetzt, so gibt der Winkel α die Kosten dieser Schreibbewegung an. Wird der Schriftzug dagegen in Richtung des Punktes B fortgesetzt, so gibt der Winkel β die Kosten an. Der Winkel α beschreibt die Kosten einer Vorwärtsbewegung und der Winkel β beschreibt die Kosten einer Rückwärtsbewegung. Sämtliche Winkel werden in dem Intervall $[0, \frac{\pi}{2}]$ gemessen. Gemäß dieser Winkeldefinition wird einer geraden Vorwärtsbewegung ebenso wie einem direktem Zurücklaufen in derselben Schreibspur der Winkel 0, d.h. die Kosten Null, zugeordnet. Der Grund dafür ist, daß diese Schreibbewegungen einfach zu schreiben sind und häufig in der Handschrift auftauchen. Demnach werden rechten Winkeln die höchsten Kosten von $\frac{\pi}{2}$ zugeordnet. Der Radius des Kreises in Abbildung 6 wird durch die Strichdicke bestimmt. Bei einer größeren Strichdicke wird auch der Radius entsprechend größer gewählt. Dadurch wird eine größere Umgebung um einen Knoten zur Winkelberechnung herangezogen. Damit soll die Winkelberechnung zuverlässiger gemacht werden. Zusätzlich werden die Kanten des Skeletts durch Geradenstücke approximiert [WD83]. Dies wird gemacht, um sich von lokalen Richtungsänderungen an einem Knoten unabhängiger zu machen und mehr

globale Information in die Winkelberechnung mit einzubeziehen.

Die berechneten Kosten werden den Kanten des Line-Graphen zugeordnet. Man beachte, daß den Knoten in Abbildung 7 Kanten im Wortskelett entsprechen (Abbildung 3). Kosten zwischen Kantenübergängen in Abbildung 3 können daher den Kanten des Line-Graphen zugeordnet werden. Die Winkelberechnung

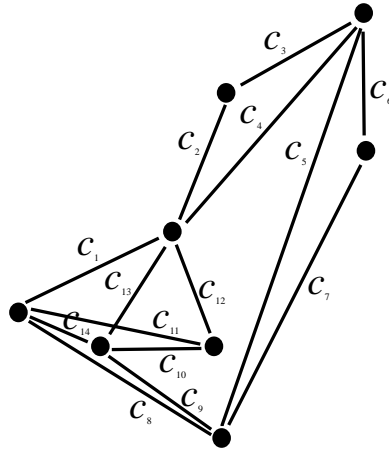


Abbildung 7: Der Line-Graph mit Kosten.

zwischen einer einfachen Kante und einer Schleife (einer Kante mit denselben Anfangs- und Endpunkten) wird folgendermaßen berechnet: Die einfache Kante bildet mit den beiden Enden der Schleife zwei Winkel. Der kleinere der beiden Winkel wird als Winkel zwischen der Kante und der Schleife verwendet. Die Winkelberechnung selbst geschieht analog zur Abbildung 6.

Beispiel: In Abbildung 8 wird der Winkel zwischen der Kante A und der Schleife B entlang des Weges berechnet, der durch den Pfeil mit der Nummer 2 gekennzeichnet ist. Der Winkel beim Übergang von Kante A nach Schleife B entlang des Pfeils 2 ist geringer als der Winkel des Übergangs von A nach B entlang des Pfeils 1.

6.3.2.5 Vervollständigung des Line-Graphen Im fünften Schritt wird der bisher hergeleitete Line-Graph vervollständigt, d.h. er wird zu einem vollständigen Graphen erweitert. Jedes Paar von Knoten, das bisher noch nicht durch eine Kante verbunden war, wird jetzt durch eine Kante verbunden. Abbildung 9 zeigt den erweiterten Graphen. Den neuen Kanten werden die Kosten der kürzesten Wege zwischen den Knoten, die sie verbinden, zugeordnet. Beispiel: Die kürzeste Verbindung zwischen Knoten C und D aus Abbildung 9 geht direkt über den Knoten B . Sie ist definiert durch die Knotenfolge C - B - D . Ein Blick zu Abbildung 7 zeigt, daß dieser Verbindung die Kosten $c_{13} + c_4$ zugeordnet werden müssen. Diese Kosten entstehen, wenn man im Skelett mit minimaler Krümmung von

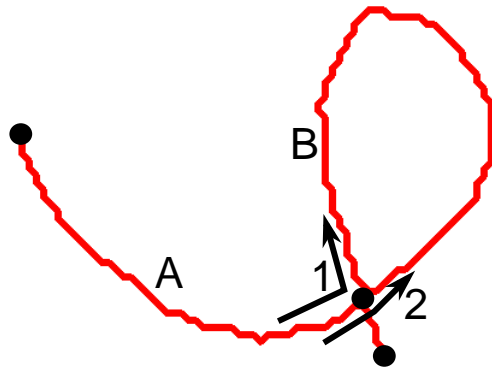


Abbildung 8: Winkel zwischen einer einfachen Kante und einer Schleife.

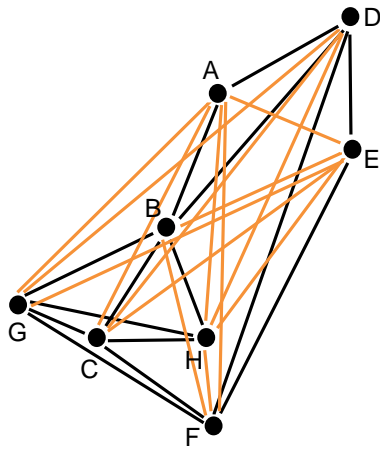


Abbildung 9: Der vervollständigte Line-Graph.

Kante C zur Kante D laufen will. Der kürzeste Weg geht über die Kante B . Kosten entstehen beim Übergang von Kante C zur Kante B und beim Übergang von Kante B zur Kante C . Diese Kosten sind identisch mit $c_{13} + c_4$.

Der Grund für die Vervollständigung des Graphen liegt in den folgenden Überlegungen: Jede Kante des Buchstabens “B” muß einmal geschrieben worden sein. Kombiniert man diese einfache Tatsache mit der Forderung nach minimaler Krümmung, so gelangt man zu dem Begriff des Hamilton-Pfads. Ein Hamilton-Pfad durch einen Graphen ist ein Pfad, der jeden Knoten genau einmal durchläuft. Gesucht ist hier also der kürzeste Hamilton-Pfad im Line-Graphen, d.h. der Pfad im Wortskelett, der jede Kante genau einmal besucht und minimale Krümmung besitzt. Nicht in jedem Graphen existiert ein solcher Hamilton-Pfad. In einigen Graphen müssen Knoten mehrfach besucht werden, um alle Knoten abzulaufen. Dies ist in Übereinstimmung mit der Tatsache, daß der Mensch beim Schreiben einige Striche (Kanten im Wortskelett bzw. Knoten im Line-Graphen) mehrfach abläuft. Das Problem besteht also darin, einen Pfad mit minimaler Krümmung zu finden, der jeden Knoten **mindestens einmal** abläuft. Solch ein Pfad wird in der Literatur auch als Pre-Hamilton-Pfad bezeichnet [GM84]. Das Problem des Auffindens eines Pre-Hamilton-Pfads in einem Graphen G läßt sich auf die Suche nach einem Hamilton-Pfad in einem vollständigen Graphen G' reduzieren. An den Kanten des vollständigen Graphen G' stehen die Kosten der kürzesten Wege zwischen den jeweiligen Knoten [GM84]. Das erklärt die Idee der Vervollständigung des Line-Graphen.

6.3.2.6 Einführung eines zusätzlichen Knotens Im nächsten Schritt wird ein zusätzlicher Knoten, der sog. Pseudo-Knoten, eingeführt. Dieser Knoten wird mit allen anderen Knoten des in Schritt 5 vervollständigten Line-Graphen verbunden. Die Kosten der neu eingefügten Kanten werden auf eine geeignete positive Konstante gesetzt [GM84]. Abbildung 10 zeigt den vervollständigten Line-Graphen zusammen mit dem Pseudo-Knoten P . Die neu eingeführten Kanten sind gestrichelt gezeichnet. Der vervollständigte und erweiterte Line-Graph bildet den formalen Rahmen für die genetische Optimierung. Der Zweck des Pseudo-Knotens P ist die Bestimmung eines Anfangs- und eines Endpunktes. Es wird dasjenige Paar von Anfangs- und Endpunkten gesucht, das die minimale Krümmung unter allen möglichen Paaren aufweist. Die Annahme ist, daß der Mensch den Buchstaben/das Wort auf krümmungsminimale Weise geschrieben hat. Die Suche nach dem krümmungsminimalen Pfad im Wortskelett entspricht der Suche nach dem kürzesten Hamilton-Pfad im vollständigen Line-Graphen. Diese Suche ist gleichbedeutend mit der Suche nach dem kürzesten **Hamilton-Zyklus** auf dem um den Knoten P erweiterten, vollständigen Line-Graphen [GM84]. Die Suche nach einem kostenminimalen Hamilton-Zyklus auf einem vollständigen Graphen wird auch als Traveling Salesman Problem (TSP) bezeichnet.

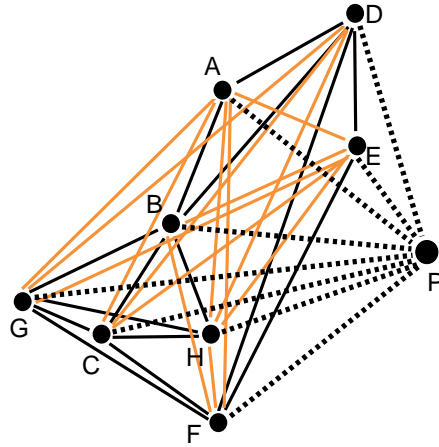


Abbildung 10: Der vollständige Line-Graph mit Pseudo-Knoten P .

6.3.2.7 Genetische Optimierung Auf dem bisher hergeleiteten Graphen wird das TSP gelöst. Es besteht darin, eine Rundreise durch alle Knoten zu finden, die jeden Knoten **genau einmal** besucht und minimale Kosten besitzt. Die Kosten einer Rundreise sind definiert als die Summe der Kosten aller Kanten dieser Rundreise. Das Lösen des TSP-Problems auf dem in Abbildung 10 dargestellten Graphen ergibt folgende Rundreise: P-A-B-C-D-E-F-G-H-P. Die genetische Fitness dieses Zyklus ist die Summe aller Winkelabweichungen entlang der Kantenfolge im Skelett.

6.3.2.8 Ableiten der Kantenfolge Die gefundene Rundreise läßt sich unmittelbar in eine Kantenfolge im Wortskelett in Abbildung 3 übertragen. Die Kantenfolge startet mit einer Kante, die von einem benachbarten Knoten von P repräsentiert wird. Also entweder die Kante A oder die Kante H . Lassen wir die Folge mit der Kante A anfangen, folgt gemäß der berechneten Rundreise die Kante B und anschließend die Kante C . Die nächste Kante — die Kante D — läßt sich nicht auf direktem Weg von der Kante C aus erreichen. Es muß dem kürzesten Pfad zwischen der Kante C und der Kante D gefolgt werden, der über die Kante B läuft. Nach dem Erreichen der Kante D ist der Weg wie folgt fortzusetzen: D-E-F-G-H. Der Endknoten P der TSP-Reise wird nicht weiter berücksichtigt. Die vollständige Kantenfolge sieht also folgendermaßen aus: A-B-C-B-D-E-F-G-H. Sie entspricht genau der Art und Weise, in welcher der menschliche Schreiber den Buchstaben geschrieben hat.

6.3.3 Verkleinerung des Suchraumes der genetischen Optimierung

Ein Mechanismus zum Zerschneiden eines Wortes ist zur Anwendung des beschriebenen Verfahrens prinzipiell nicht nötig. Das Problem des Traveling Salesman ist in der Informatik sehr bekannt und gehört in die Klasse der schwierigen

(NP-vollständigen) Probleme. Mit ansteigender Knotenzahl wird es immer aufwendiger, die korrekte Lösung zu finden. Aus praktischen Erwägungen kann es daher notwendig sein, den Suchraum für die genetische Optimierung zu verkleinern, um die Suche zu beschleunigen. In Abbildung 11 ist gezeigt, wie das Beispielwort “Duxbury” in einzelne TSP-Teilprobleme zerlegt werden kann, die effizienter gelöst werden können als das komplette TSP-Problem auf dem gesamten Wort. Das “D” am Anfang des Wortes wird als eigenständige Zusammenhangs-



Abbildung 11: Zerlegung des Wortes “Duxbury” in Teilprobleme.

komponente gelöst. Dies geschieht ähnlich zu dem vorher beschriebenen “B”. Der Rest des Wortes wird in kleinere Teilprobleme zerlegt. Wird der Anfangspunkt und der Endpunkt wie in Abbildung 11 festgelegt, so kann das Problem in die einzelnen unabhängigen Teilprobleme zerlegt werden, die in Abbildung 11 durch senkrechte Striche abgetrennt sind. Jeder dieser senkrechten Striche geht durch eine trennende Kante. Auf jedem dieser Teilprobleme läßt sich das TSP-Problem unabhängig lösen. Nur der Start- und der Endpunkt ist bei diesen Teilproblemen vorgegeben, was beim Lösen des TSP-Problems durch entsprechende Wahl der Gewichte der Kanten des Pseudo-Knotens berücksichtigt werden kann [GM84]. Trennende Kanten werden nur einmal von links nach rechts durchlaufen. Der vollständige Weg durch das Wort wird aus den Teillösungen zusammengesetzt. Berechnet man die Länge aller Kanten und anschließend die kürzesten Wege zwischen den Knoten, so läßt sich folgendes festhalten: Für jede trennende Kante gilt, daß einer ihrer Knoten näher zum Anfangspunkt ist, während der andere Knoten näher zum Endpunkt ist. Ferner befindet sich auf beiden Seiten einer trennenden Kante dieselbe Kontur [MO90, Kwo88]. In Abbildung 11 sind die Konturen von 1 bis 6 durchnummeriert. Alle trennenden Kanten in Bild 11 haben auf beiden Seiten die Kontur mit der Nummer 1 (äußere Kontur).

6.3.4 Lösung des TSP-Problems

6.3.4.1 Verfahren zur Lösung Obwohl sich das TSP-Problem relativ einfach beschreiben läßt, ist es jedoch bei einer großen Anzahl von Städten nicht effizient auf einem Rechner zu lösen. Es gehört zu der bekannten Klasse von NP-vollständigen Problemen, die nicht mit polynomialem Zeitaufwand zu lösen sind [CLR90]. In der Postautomatisierung sind aufgrund des hohen Postaufkommens starke zeitliche Restriktionen vorgegeben. Es ist daher sehr wichtig, die Schriftzugrekonstruktion so schnell wie möglich durchzuführen.

Ein wichtiger Schritt zur Beschleunigung des Problems ist daher die Zerlegung eines Wortes in Teilprobleme. Diese Teilprobleme sind kleiner als das ursprüngliche Problem und können unabhängig voneinander und evtl. auch parallel gelöst werden.

Eine weitere Beschleunigung des vorgestellten Verfahrens kann durch die Anwendung von Heuristiken erreicht werden. Heuristiken müssen nicht notwendigerweise die optimale Lösung finden. Für Probleme mit einer geringen Anzahl von Knoten, wie das für unsere Problemstellung der Fall ist, berechnen sie jedoch häufig die optimale Lösung [M96].

Wir verwenden für die Lösung des TSP-Problems ein einfaches “Branch and Bound Verfahren” und benutzen einfache Heuristiken, wenn die Anzahl der Knoten eine bestimmte Anzahl überschreitet [Cra]. Eine einfache Heuristik, die eingesetzt wurde, ist z.B. der “farthest insertion algorithm” [Cra]. Es hat sich sogar herausgestellt, daß diese Verfahren für unsere Problemgrößen ausreichend sind und daß wir nicht notwendigerweise auf genetische Algorithmen zurückgreifen müssen.

6.3.4.2 Mathematischer Rahmen Interessanterweise lassen sich die meisten Probleme, auf die wir in den Arbeitsschritten eins bis drei gestoßen sind, mit einem einfachen mathematischen Formalismus beschreiben. Sowohl die Länge des kürzesten Pfades als auch alle möglichen Hamilton-Pfade in einem Graphen lassen sich durch eine einfache Multiplikation von Matrizen berechnen. Es läßt sich auch der Viterbi Algorithmus mit diesem Modell realisieren. Der Viterbi Algorithmus bestimmt den wahrscheinlichsten Pfad in einem hidden Markov Modell, das ein wesentlicher Bestandteil des Handschrifterkenners ist. Unabhängig von unserer Problemstellung lassen sich noch viele andere graphentheoretische Fragestellungen mit demselben mathematischen Formalismus berechnen. Bei diesem Formalismus handelt es sich um eine algebraische Struktur, die auch Dioid genannt wird. Sie wird durch zwei Operatoren definiert, die bestimmten Gesetzen (wie z.B. Assoziativität oder Kommutativität) genügen müssen. Diese beiden Operatoren sind vergleichbar mit den bekannten arithmetischen Operatoren $+$ und $*$

aus der Mathematik. Allerdings sind sie allgemeiner definiert und können auf verschiedenartige Objekte angewendet werden. In Abhängigkeit von der zu lösenden Aufgabe werden die Operatoren und die Werte einer Matrix A unterschiedlich initialisiert. Anschließend wird die Matrix A mit sich selbst multipliziert, bis ein Fixpunkt erreicht ist. Der Fixpunkt enthält die gesuchte Lösung. Nähere Erläuterungen zu dieser Vorgehensweise finden sich in [GM84] und [BCOQ92].

Durch den beschriebenen Mechanismus läßt sich die Schriftzugrekonstruktion auf eine rekursive Fixpunktsuche abbilden, die sich im wesentlichen durch eine Multiplikation von Matrizen realisieren läßt.

6.4 Resultate

6.4.1 Rekonstruierte Schriftzüge

In diesem Unterabschnitt werden einige erfolgreich rekonstruierte Schriftzüge gezeigt. Neben dem schon in einem vorhergehenden Abschnitt gezeigten “k”, dessen Schriftzug sich erfolgreich rekonstruieren läßt, sei hier auch der rekonstruierte Schriftzug des Buchstabens “B” aus Abbildung 2 gezeigt. Abbildung 12 zeigt das Binärbild und das approximierte Skelett von “B”. Das approximierte Skelett wird zur Berechnung der Winkel verwendet. Der Anfang und das Ende des Schriftzuges sind im Skelett gekennzeichnet. Die kleinen Pfeile im Binärbild verdeutlichen

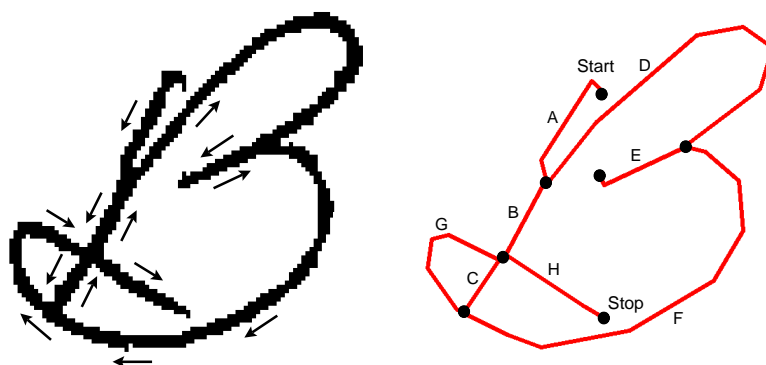


Abbildung 12: Rekonstruierter Schriftzug des Buchstabens “B”.

den ursprünglichen Schriftzug, welcher der Kantenfolge $A, B, C, B, D, E, F, G, H$ im Skelett entspricht. Da das approximierte Skelett dem nicht approximierten Skelett ähnelt, werden wir in den weiteren Beispielen nur noch das nicht approximierte Skelett zeigen.

Ein anderes Beispiel ist in Abbildung 13 gezeigt. Hier wird der Schriftzug des Buchstabens “D” rekonstruiert. Die grobe Struktur des Buchstabens wird u.a. durch einen Knoten mit dem Grad vier und einen Knoten mit dem Grad fünf im

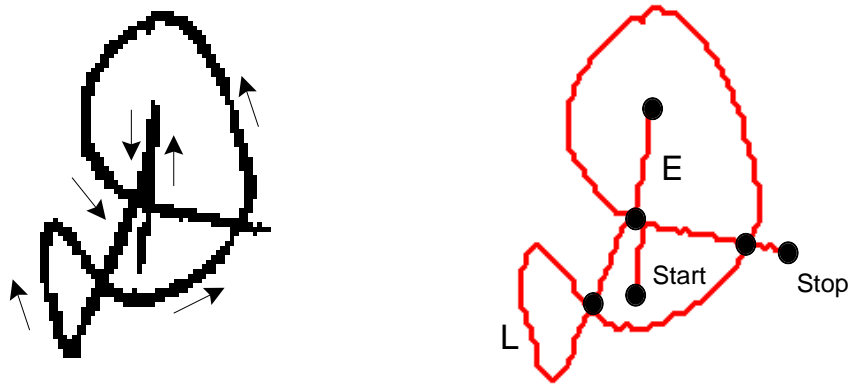


Abbildung 13: Rekonstruierter Schriftzug des Buchstabens “D”.

Skelett repräsentiert. Die Schleife L wird richtig durchlaufen, und die Kante E wird korrekt zurückverfolgt.

Der Schriftzug des Buchstabens “o” wird in Abbildung 14 richtig rekonstruiert.

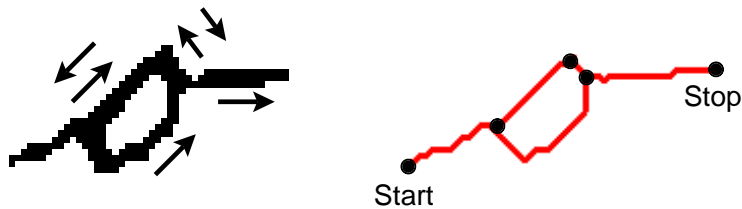


Abbildung 14: Rekonstruierter Schriftzug des Buchstabens “o”.

Ein weiteres Beispiel ist in Abbildung 15 gezeigt. Hier wird wieder die Schleife und die Kanten e_1, e_2 richtig durchlaufen.

Ähnlich zu dem vorhergehenden Beispiel wird auch in Abbildung 16 die Kantenfolge e_1, e_2 korrekt durchlaufen.

Probleme bei der Schriftzugrekonstruktion treten auf, wenn der Stift abgesetzt wird und damit die kontinuierliche Schreiblinie nicht auf dem Papier, sondern in der Luft fortgesetzt wird. T-Striche sind hier ein typisches Beispiel. Auch können Fehler auftreten, wenn der Anfangs- oder der Endpunkt des Schriftzuges nicht richtig bestimmt wurde.

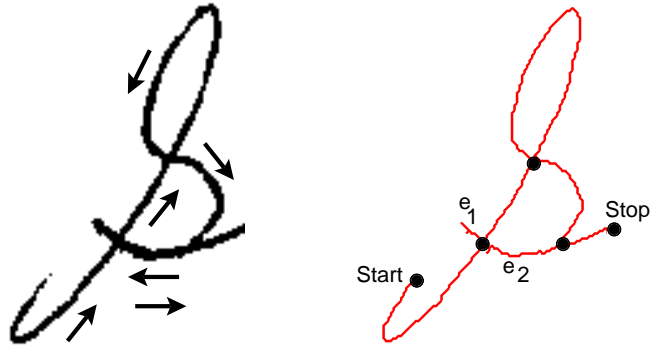


Abbildung 15: Rekonstruierter Schriftzug des Buchstabens "G".

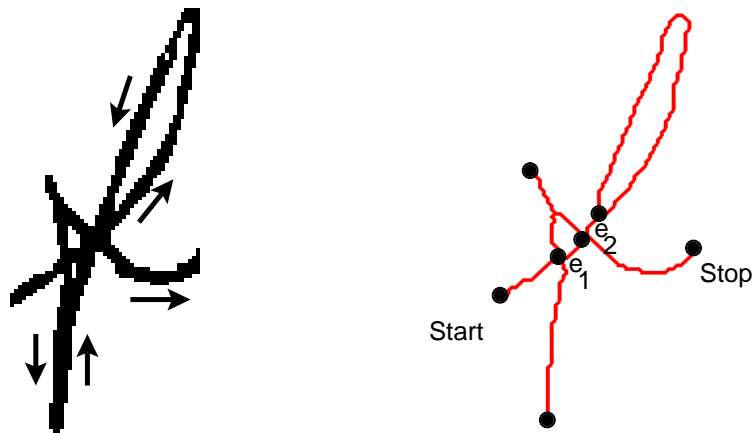


Abbildung 16: Rekonstruierter Schriftzug des Buchstabens "f".

6.4.2 Klassifikationsergebnisse

Das entwickelte Verfahren wurde benutzt, um aus Off-line-Bildern Schriftzüge zu rekonstruieren und diese dann als On-line-Daten abzuspeichern. Dazu wurde das Unipen-Format verwendet, das ein einfaches Format beschreibt, mit dem On-line-Erkennen ihre Daten verwalten und abspeichern können [GSP⁺93]. Mit den rekonstruierten On-line-Daten wurde ein On-line-Erkennen trainiert und getestet. Für diese Experimente konnten wir freundlicherweise den Erkennen der Universität Karlsruhe verwenden (Prof. Waibel). Obwohl die Erkennungsraten bisher niedriger sind als die entsprechenden Ergebnisse des reinen Off-line-Erkenners, zeigte sich doch, daß dieser Ansatz weiterverfolgt werden sollte. Insbesondere da der Karlsruher Erkennen die Hälfte der Wörter richtig erkannte, die der Off-line-Erkennen nicht richtig erkennen konnte. Somit könnte eine Kombination der beiden Erkennen zu einer Erhöhung der Erkennungsraten führen. Vor allem das Training des Karlsruher Erkenners ist noch nicht optimal an die rekonstruierten On-line-Daten angepaßt und kann noch in einigen Punkten verbessert werden.

6.5 Weitere Arbeiten

Um einen optischen Eindruck von der Leistungsfähigkeit der entwickelten Methoden und Algorithmen zu erhalten, wurde eine Oberfläche entwickelt, die es erlaubt, die rekonstruierten Schriftzüge zu visualisieren. Es lassen sich Testläufe mit den unterschiedlichen, implementierten Algorithmen durchführen. Zusätzlich lassen sich verschiedene Parameter für die Testläufe einstellen. Handgeschriebene Wörter lassen sich in mehreren Verarbeitungsstufen anzeigen. Es lassen sich z.B. das originale Binärbild, das gefilterte Bild, die Konturen im Bild, das Skelett und das approximierte Skelett anzeigen. Einen ungefähren Eindruck von der entwickelten Oberfläche gibt Abbildung 17.

7 Voraussichtlicher Nutzen

Dieser Abschnitt beschreibt die wesentlichen wissenschaftlichen Ergebnisse und den Nutzen, der sich aus ihnen ergibt.

- Im Rahmen des Projekts wurde gezeigt, daß sich die Schriftzugrekonstruktion als Optimierungsproblem lösen läßt. Im Gegensatz zu den bekannten Überlegungen und Ansätzen wurde gezeigt, daß eine Minimierung der Krümmung zur Rekonstruktion des Schriftzuges ausreicht. Ferner wurde gezeigt, daß sich einige Schriftzüge nicht mit der Minimierung der Länge, sondern nur mit der Krümmungsminimierung rekonstruieren lassen. Es handelt sich bei der Schriftzugrekonstruktion keineswegs um ein unlösbares Problem. Was die Schriftzugrekonstruktion vielmehr schwierig macht, ist die Berechnung von exakter Information, wie z.B. Information über Win-

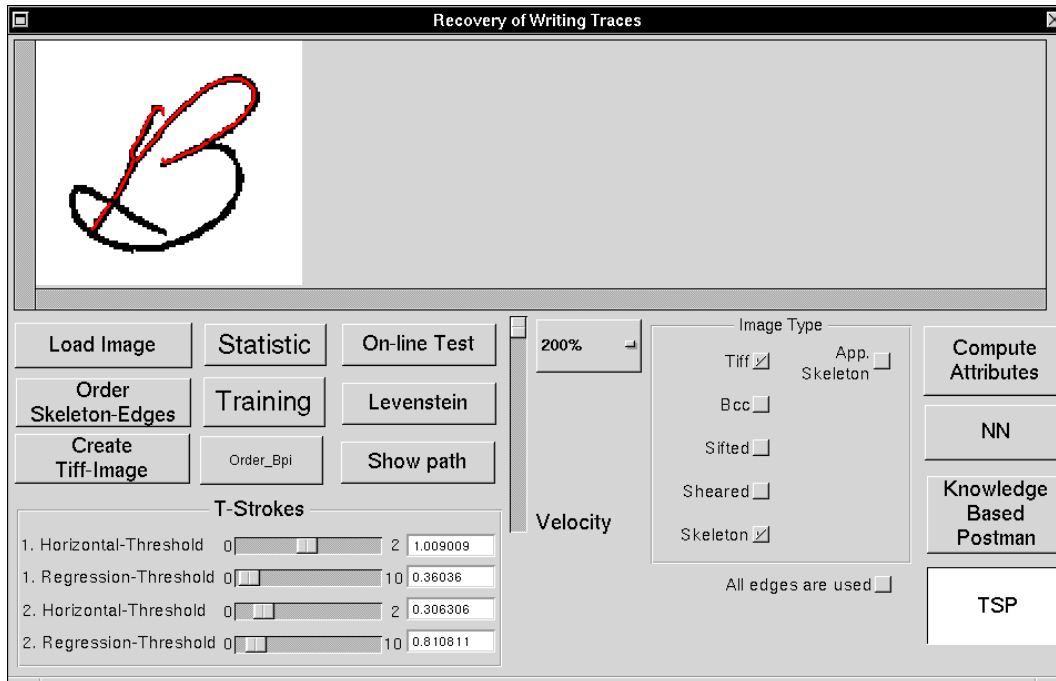


Abbildung 17: Benutzeroberfläche

kel aus dem Binärbild. Diese Probleme sollten in Zukunft durch geeignete Algorithmen beseitigt oder zumindest abgeschwächt werden. Die Wahl der richtigen Auflösung während der digitalen Abtastung ist ein wichtiger Parameter. Die Verwendung des Grauwertbildes als Repräsentation des geschriebenen Wortes kann hier zu Hilfe genommen werden. Sie erleichtert die Berechnung von zuverlässiger Information.

- Durch die relativ hohen On-line-Erkennungsergebnisse wurde deutlich, daß sich die Schriftzugrekonstruktion sinnvoll zur Erkennung von Handschrift einsetzen läßt. Insbesondere die Kombination mit einem existierenden Offline-Erkennen erscheint nun besonders aussichtsreich. Dies ist ein wertvolles praktisches Ergebnis und eine Aufforderung für zukünftige Arbeiten.
- Das zweidimensionale Bild eines handgeschriebenen Wortes stellt die Eingabe für den komplexen Prozeß “Lesen” dar. Es dürfte zweifelsfrei feststehen, daß es sich beim Lesen um einen sehr komplexen kognitiven Prozeß handelt. Leider fehlt bisher eine einheitliche Theorie der Kognition. Daher dürfte dem Ergebnis, daß der Schriftzug eines geschriebenen Wortes durch eine rekursive Fixpunktsuche rekonstruiert werden kann, einige Bedeutung zukommen. Darüber hinaus ist diese Feststellung in Übereinstimmung mit einigen wichtigen kognitiven Überlegungen, wie sie z.B. in [vF96] gemacht werden. Die Auswirkungen dieses theoretischen Ergebnisses sind noch nicht abzusehen.

- Die Ergebnisse des Projekts haben deutlich gezeigt, daß die wesentlichen Bestandteile der Rekonstruktion des Schriftzuges und der eigentlichen Erkennung nicht mehr sind als eine Multiplikation von Matrizen. Dies eröffnet den Weg zu neuen Denkweisen und ermöglicht Implementierungen in spezieller Hardware.
- Im Rahmen des Projekts wurde ein Algorithmus entwickelt, der sich vollständig mit diskreter Mathematik realisieren läßt. Die Eingabe des Algorithmus, seine Verarbeitung und seine Ausgabe sind diskret. Der Schriftzug wird nicht durch kontinuierliche Funktionen beschrieben. Der Algorithmus paßt sich daher problemlos in die Welt der Digitalrechner ein.
- Zur Beschleunigung des entwickelten Algorithmus mußte ein Wort in Teilprobleme zerlegt werden. Dazu wurde ein effizientes Verfahren entwickelt. Das Verfahren läßt sich zwar nicht unmittelbar zur Segmentierung von Buchstaben benutzen, da es auch innerhalb von Buchstaben segmentieren kann, es ist jedoch sehr wahrscheinlich, daß es auch in anderen Anwendungen sinnvoll eingesetzt werden kann.

8 Fortschritte von dritter Seite

8.1 Verwandte Arbeiten

Während der Durchführung des Projekts wurden nur sehr wenige Arbeiten von dritter Seite bekannt. Es ist auch fraglich, ob in diesen Fällen von Fortschritten gesprochen werden kann. Zwei Arbeiten, die in dem Zeitraum des Projekts bekannt wurden, sind [BAK⁺97] und [LVG97]. In [BAK⁺97] wird die Rekonstruktion des Schriftzuges als ein Suchproblem definiert. Ähnlich zu der im geförderten Projekt entwickelten Problemrepräsentation wird eine graphentheoretische Repräsentation des Problems gewählt. Die Suche nach dem korrekten Schriftzug wird durch eine einfache Best-First Suche durchgeführt. Zur Bestimmung des Gütekriteriums eines Schriftzuges werden sowohl lokale als auch globale Kriterien herangezogen. Globale Kriterien sind:

- die allgemeine Schreibrichtung von links nach rechts,
- die Pfadlänge des rekonstruierten Schreibzuges.

Als lokale Kriterien werden die folgenden Merkmale verwendet:

- die lokale Krümmung an einem Kreuzungspunkt,
- der Durchlaufsinne einer Schleife.

Die lokale Krümmung an einem Kreuzungspunkt wurde schon in [BCCM93], einem schon vor Projektbeginn existierenden Ansatz, verwendet. Um den Durchlaufsinne einer Schleife zu bestimmen, wurde eine einfache Heuristik verwendet. Schleifen, die im oberen Teil des Wortes liegen, z.B. “e”, “l”, werden gegen den Uhrzeigersinn durchlaufen. Schleifen, die im unteren Teil des Wortes liegen, z.B. “g”, “j”, werden im Uhrzeigersinn durchlaufen. Obwohl diese Heuristik in sehr vielen Fällen den richtigen Umlaufsinne einer Schleife angibt, gibt es jedoch einige Ausnahmen. Die Leistungsfähigkeit des in [BAK⁺97] beschriebenen Ansatzes wurde ebenfalls mit Hilfe eines echten On-line-Erkenner getestet, d.h. die rekonstruierten Schriftzüge wurden als Eingabe für einen On-line-Erkennner verwendet. Da das Verfahren mit einfachen Daten von kooperativen Schreibern getestet wurde, sind auch recht hohe Erkennungsraten erzielt worden. Allerdings ist dieser Ansatz nicht mit real-world Daten getestet worden. Ein weiterer Nachteil ist, daß sich einige Schriftzüge mit der vorgestellten Methode nicht rekonstruieren lassen. So lassen sich Schriftzüge mit dreifach durchlaufenen Kanten bisher nicht erfolgreich rekonstruieren. Auch erscheint der im Rahmen des geförderten Projekts entstandene Algorithmus einfacher. Insbesondere wird nur eine Variable, nämlich die Krümmung, zur Optimierung herangezogen. Dies erleichtert nicht unwesentlich die Suche nach der optimalen Lösung.

Der zweite Ansatz [LVG97] versucht den Schriftzug durch mathematische Funktionen zu beschreiben. Hierzu werden Kalman Filter verwendet. Eine wichtige Voraussetzung ist, daß sich Schreibzüge durch eine Sequenz von Kreisbögen beschreiben lassen. Obwohl [LVG97] einen interessanten theoretischen Ansatz darstellt, muß die praktische Anwendbarkeit erst noch bewiesen werden.

8.2 Verwandte Forschungsgebiete

Eine interessante Beobachtung, die jedoch erst gegen Ende der Laufzeit des Projektes gemacht wurde, ist, daß die Rekonstruktion des Schriftzuges nicht nur für die Erkennung von Wörtern wichtig ist, sondern auch in der Forensik an Bedeutung gewinnt. Dies konnte man auf den folgenden Konferenzen beobachten:

- Eighth Biennial Conference of the International Graphonomics Society (Genua)
- 3. Internationaler Kongreß der Gesellschaft für Forensische Schriftuntersuchung (GFS/Luzern),

Einige Themen, mit denen sich die Forensik intensiv beschäftigt, sind Scheckbetrug und das Fälschen von Unterschriften. Hier wird deutlich, daß die Untersuchung von Handschrift ein wichtiges Gebiet innerhalb der Forensik ist. Eine Aufgabe in der Forensik ist das Überprüfen von Unterschriften, d.h. die Beantwortung der Frage, ob eine Unterschrift gefälscht ist oder nicht. Eine weitere Aufgabe

in der Forensik ist das Heraussuchen von Unterschriften, die zu einer vermutlich gefälschten Unterschrift ähnlich sind. Insbesondere bei dieser Aufgabe ist die Rekonstruktion des Schriftzuges von Bedeutung, da mit Hilfe des rekonstruierten Schriftzuges Merkmale für eine Unterschrift berechnet werden können. Anhand dieser Merkmale können dann ähnliche Unterschriften aus einer Datenbank herausgesucht werden.

Ein wichtiger Unterschied zwischen der Forensik und der Handschrifterkennung liegt darin, daß man in der Forensik versucht, schreiberabhängige Eigenschaften der Handschrift zu erhalten und zu extrahieren. Die Handschrifterkennung hingegen versucht gerade diese Individualitäten zu normalisieren, um den Erkennungsprozeß nicht durch unnötige Varianten der Eingabe zu erschweren. Dennoch sind sowohl die Forscher in der Forensik als auch die Forscher in der Handschrifterkennung an einer Rekonstruktion des Schriftzuges interessiert.

Im Gegensatz zu den einfachen Binärbildern, die im geförderten Projekt die Eingabe für den entwickelten Algorithmus darstellen, sind in der Forensik die meisten Unterschriften als Grauwertbilder gespeichert. Grauwertbilder tragen deutlich mehr Information als Binärbilder und können dazu benutzt werden, Informationen über Winkel und Druckstärken zuverlässiger zu berechnen. Auch werden in der Forensik leistungsfähigere optische Geräte eingesetzt, wie z.B. Mikroskope. Ein großes Problem in der Forensik stellt allerdings die Trennung von Hintergrund und Unterschrift dar (z.B. auf Schecks).

9 Veröffentlichungen und Schutzrechtsanmeldungen

Im Verlauf des Projekts wurden vom Autor die folgenden Arbeiten veröffentlicht: [Jae96, Jae97a, Jae97b]. Sie beschreiben im wesentlichen den während des Projekts entwickelten Algorithmus zur Schriftzugrekonstruktion. Eine ausführliche Darstellung des Algorithmus wird demnächst als Dissertation veröffentlicht. Die Dissertation wurde im Rahmen des geförderten Projekts durchgeführt und somit wesentlich vom BMBF unterstützt. Sie enthält neben dem theoretischen Grundgerüst der entwickelten Algorithmen auch die notwendigen Komplexitätsbetrachtungen und die Analyse der Leistungsfähigkeit der einzelnen Algorithmen. Der entwickelte Algorithmus zur Schriftzugrekonstruktion (3. Arbeitsphase) wurde beim deutschen Patentamt zur Patentierung angemeldet.

Literatur

- [BAK⁺97] H. Bunke, R. Ammann, G. Kaufmann, T. M. Ha, M. Schenkel, R. Seiler, and F. Eggimann. Recovery of Temporal Information of Cursively Handwritten Words for On-line Recognition. In *Fourth International Conference on Document Analysis and Recognition (ICDAR)*, pages 931–935, 1997.
- [BCCM93] G. Boccignone, A. Chianese, L. P. Cordella, and A. Marcelli. Recovering dynamic information from static handwriting. *Pattern Recognition*, 26(3):409–418, 1993.
- [BCOQ92] F. L. Baccelli, G. Cohen, G. J. Olsder, and J. Quadrat. *Synchronization and Linearity*. Wiley, 1992.
- [BM76] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. Elsevier Science Publishing Co., Inc., 1976.
- [CGKM93] T. Caesar, J. M. Gloger, A. Kaltenmeier, and E. Mandler. Recognition of handwritten word images by statistical methods. In *3rd International Workshop of Frontiers in Handwriting Recognition*, pages 409–416, Buffalo, USA, 1993.
- [CGKM94] T. Caesar, J. M. Gloger, A. Kaltenmeier, and E. Mandler. Handwritten word recognition using statistics. In *1st IEE Workshop Handwriting Analysis and Recognition: A European Perspective*, Brussels, 1994.
- [CGM93a] T. Caesar, J. M. Gloger, and E. Mandler. Design of a system for off-line recognition of handwritten word images. In *Jet Poste 93, Proceedings of the First European Conference Dedicated to Postal Technologies*, pages 156–161, Nantes, France, 1993. Service de Recherche Technique de la Poste.
- [CGM93b] T. Caesar, J. M. Gloger, and E. Mandler. Preprocessing and feature extraction for a handwriting recognition system. In *2nd IAPR Int. Conference on Document Analysis and Recognition*, pages 408–411, Tsukuba, Japan, 1993.
- [Chr77] N. Christofides. *Graph Theory: An Algorithmic Approach*. Academic Press, New York, 1977.
- [CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.
- [Cra] R.J. Craig. AT&T Bell Labs, 1200 E. Warrenville Rd, Naperville, IL. 60566-7045.

- [DR92] D. S. Doermann and A. Rosenfeld. Recovery of temporal information from static images of handwriting. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 162–168, 1992.
- [DR95] D. S. Doermann and A. Rosenfeld. Recovery of temporal information from static images of handwriting. *International Journal of Computer Vision*, 15:143–164, 1995.
- [EJ73] J. Edmonds and E. L. Johnson. Matching, Euler Tours and the Chinese Postman. *Mathematical Programming*, (5):88–124, 1973.
- [GM84] M. Gondran and M. Minoux. *Graphs and Algorithms*. Wiley, 1984.
- [Gol89] D. E. Goldberg. *Genetic Algorithms in Search Optimization & Machine Learning*. Addison-Wesley, 1989.
- [GSP+93] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN project of on-line data exchange and recognizer benchmarks. 1993.
- [Jae96] S. Jaeger. Recovering Writing Traces in Off-line Handwriting Recognition: Using a Global Optimization Technique. In *Proc. of the 13th International Conference on Pattern Recognition*, pages C:150–154, 1996.
- [Jae97a] S. Jaeger. Motor Control, Curvature Minimization, and the Traveling Salesman. In *Eighth Biennial Conference of the International Graphonomics Society (IGS)*, pages 3–4, 1997.
- [Jae97b] S. Jaeger. A Psychomotor Method for Tracking Handwriting. In *Fourth International Conference on Document Analysis and Recognition (ICDAR)*, pages 528–531, 1997.
- [KCGM93] A. Kaltenmeier, T. Caesar, J. M. Gloger, and E. Mandler. Sophisticated topology of hidden markov models for cursive script recognition. In *2nd IAPR Int. Conference on Document Analysis and Recognition*, pages 139–142, Tsukuba, Japan, 1993.
- [Kwo88] P. Kwok. A thinning algorithm by contour generation. In *Communications of the ACM*, number 11, pages 1314–1324, 1988.
- [LVG97] P. M. Lallican and C. Viard-Gaudin. A Kalman Approach for Stroke Order Recovering from Off-line Handwriting. In *Fourth International Conference on Document Analysis and Recognition (ICDAR)*, pages 519–522, 1997.

- [M96] H. Mühlenbein. 1996.
- [MO90] E. Mandler and M. F. Oberländer. A single pass algorithm for fast contour coding of binary images (in german). In *Proc. of the 12th DAGM-Symposium*, pages 248–255, 1990.
- [Neu88] K. Neumann. Das Briefträgerproblem in Graphen, Digraphen und gemischten Graphen. Technical Report WIOR-329, Institut für Wirtschaftstheorie und Operations Research - Universität Karlsruhe, März 1988.
- [Qui93] J. R. Quinlan. *C4.5 Programs For Machine Learning*. Morgan Kaufmann, 1993.
- [SW94] C. Y. Suen and P. S. P. Wang, editors. *Thinning Methodologies For Pattern Recognition*. World Scientific, 1994.
- [vF96] H. v. Foerster. *Wissen und Gewissen (german translation)*. suhrkamp taschenbuch, 1996.
- [WD83] K. Wall and P.E. Danielsson. A new method for polygonal approximation of digitized curves. In *Proc. of the 3rd Scandinavian Conf. on Image Analysis*, 1983.