



GMD –
Forschungszentrum
Informationstechnik
GmbH

Liliane Peters, Ashutosh Malaviya,
Franjo Ivančić, Bertin Klein

Erkennungssysteme und Dokumentanalyse

READ, Beitrag zum Abschlußbericht

© GMD 1999

GMD –
Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
<http://www.gmd.de>

In der Reihe GMD Report werden Forschungs- und Entwicklungsergebnisse aus der GMD zum wissenschaftlichen, nicht-kommerziellen Gebrauch veröffentlicht. Jegliche Inhaltsänderung des Dokuments sowie die entgeltliche Weitergabe sind verboten.

The purpose of the GMD Report is the dissemination of research work for scientific non-commercial use. The commercial distribution of this document is prohibited, as is any modification of its content.

Anschriften der Verfasser/Addresses of the authors:

Dr. Liliane Peters
Dr. Ashutosh Malaviya
Franjo Ivančić
Institut für Autonome intelligente Systeme
GMD – Forschungszentrum Informationstechnik GmbH
D-53754 Sankt Augustin

Bertin Klein
Institut für Integrierte Publikations- und Informationssysteme
GMD – Forschungszentrum Informationstechnik GmbH
Dolivostraße 15
D-64293 Darmstadt

E-mail: Liliane.Peters@gmd.de

ISSN 1435-2702

Kurzfassung

Papiergebundene Information spielt bei der Kommunikation von Menschen und Organisationen trotz der weitverbreiteten elektronischen Medien weiterhin eine zentrale Rolle. Ziel der Arbeiten im READ Verbundprojekt war die Verbesserung und Weiterentwicklung von Erkennungssystemen, die den Bruch im Informationsfluß zwischen Papierdokumenten und elektronischer Welt überwinden. Die Bearbeitung von Dokumenten soll mit den Ergebnissen des READ Projekts weitgehend automatisch möglich werden.

Im Projekt READ wurde eine neue Methode und darauf aufbauend Werkzeuge zur Analyse und Test zur Erkennung von handgeschriebenen Wörtern entwickelt. Durch die Wahl der Fuzzy Logik Methoden konnte die Lösung von Handschrifterkennungsproblemen, die sich durch eine Vielzahl komplexer Abweichungen auszeichnet, effizient gelöst werden.

Es wurde eine Verbesserung der Entwicklungsmethodik von Systemen zur Dokumentanalyse erreicht durch die Entwicklung eines Formalismus, die Zusammenführung existierender Verfahren ermöglicht hat. Dieser bildet den Rahmen des Systemdesigns und wurde erfolgreich implementiert und getestet.

Schlagwörter

Handschrifterkennung, Fuzzy Logik, Regelbasis, Lernverfahren, stark strukturierte Information, Parser.

Abstract

Although electronic media is widely used within any organisation, paperwork is still a very important medium for information communication. The central goal of the READ project was the continuous improvement and design of new recognition systems, that would cover the gap between the electronic available information and the paper based one. The automatic processing of documents should be possible with the results of this project.

In the research and development project READ a new method and tools for an adaptive handwriting recognition system were developed. Fuzzy clustering techniques supplement the fuzzy syntactic methods to extract the expert information automatically to create the required rule-base. By choosing a fuzzy paradigm, we were able to find an efficient solution for the classification of handwritten data.

The restricted similarities of the existent proprietary document analysis systems were extracted. These similarities create the framework of the proposed design methodology for a generic document analysis system. The design methodology was supported by the development and test of a formalism to generalise the approach. The new design paradigm was tested successfully.

Keywords

Handwriting Recognition, Fuzzy Logic, Rule Base, Learning, context dependent Information, Parser

Das diesem Report zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie unter dem Förderkennzeichen 01 IN 503E/8 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhaltsverzeichnis

0. Einleitung / Zusammenfassung.....	1
1. Aufgabenstellung und Ausgangsbedingungen des Vorhabens.....	3
1.1 Aufgabenstellung	3
1.2 Voraussetzungen des Vorhabens	4
1.3 Wissenschaftlich-technischer Stand zu Beginn des Vorhabens.....	6
1.4 Zusammenarbeit mit anderen Stellen.....	8
2. Planung und Ablauf des Vorhabens.....	9
3. Ergebnisse.....	10
3.1 Wissenschaftlich-technische Ergebnisse	10
3.2 Nutzen und Verwertbarkeit der Ergebnisse	28
3.3 Fortschritt auf diesem Gebiet bei Dritten	29
Literaturhinweis.....	30
Anhang 1: Veröffentlichungen & Patente	33
Publikationen	33
Diplomarbeiten und Dissertationen	34
Ausstellungen und Präsentationen.....	34
Patente	34

0. Einleitung / Zusammenfassung

Papiergebundene Information spielt bei der Kommunikation von Menschen und Organisationen trotz der weitverbreiteten elektronischen Medien weiterhin eine zentrale Rolle. Ziel der Arbeiten im READ Verbundprojekt war die Verbesserung und Weiterentwicklung von Erkennungssystemen, die den Bruch im Informationsfluß zwischen Papierdokumenten und elektronischer Welt überwinden. Die Bearbeitung von Dokumenten soll mit den Ergebnissen des READ Projekts weitgehend automatisch möglich werden.

Forschungsthemen

Zur Lösung dieser anspruchsvollen Aufgaben sind im Projekt READ eine Reihe von Technologien zu unterschiedlichen Stufen des Erkennungsprozesses entwickelt worden. Die Arbeiten folgten dabei der natürlichen Verarbeitungsreihenfolge Erfassen, Objektbilden, Erkennen, Interpretieren, Optimieren und Anwenden. Im einzelnen konzentrierten sich die Arbeiten auf folgende Schwerpunkte:

Für die Erfassung papiergebundener Informationen wurde eine hochauflösende Farbkamera mit exakter Farbwiedergabe und hohem Durchsatz sowie Algorithmen zur Bildkorrektur und Objektrepräsentation entwickelt.

Methoden zur Isolierung und Erkennung von bedeutungstragenden Objekten sowie zur robusten Erkennung handschriftlicher Dokumente sind implementiert worden: Erkennung gebundener Handschrift im westlichen (alphabetischen) Stil sowie handgeschriebener chinesischer Zeichen als am weitesten verbreitete fernöstliche Schrift.

Hier wurden die für eine Aufgabe relevanten Informationen aus einem Text extrahiert, um ihn einem Thema zuzuordnen, automatisch indizieren und als Vorgang bearbeiten zu können. Einen gesonderten Schwerpunkt bildeten hier Verfahren zur automatischen Wissensakquisition, um das für die Künstliche Intelligenz typische Problem des Wissenserwerbs zu lösen.

Die neuen Techniken zur Dokumentanalyse erfordern auch neue Methoden zur Einschätzung der Leistungsfähigkeit und zur flexiblen Konfiguration der am Analyseprozeß beteiligten Komponenten.

Anwendungen

Die in READ erzielten wissenschaftlichen Ergebnisse ermöglichen industrielle Anwendungen von beträchtlichem Potential.

Farbscannen beschränkte sich bisher auf Desk Top Publishing (DTP) - Anwendungen, bei denen es genügte, wenige Bilder pro Minute meist einseitiger Vorlagen auszugeben. Der hohe Durchsatz der in READ entwickelten Kamera erlaubt erstmals Farbe auch in industriellen Produktionsumgebungen einzusetzen, wobei durch die hohe Bildqualität und eine problemangepaßte Datenreduktion eine äußerst effiziente Bildgewinnung möglich wird.

Im Bereich der Postautomatisierung ermöglicht die Erkennung von chinesischen Schriftzeichen den Zugang zu den asiatischen Märkten. Mit der Beherrschung der Erkennung von Handschriften und von komplexen Dokumentlayouts gewinnt der kommerzielle Markt für die Erkennungstechnik zunehmend an Bedeutung, z.B. für Kommerzielle Dienstleister, Vorsortier-Büros, Inhouse-Mail-Verteilung.

Die zukunftssträchigsten Anwendungen liegen im Bereich der Dokument Management Systeme, die Dokumente automatisch klassifizieren, indizieren und relevante Informationen daraus extrahieren; die Bearbeitung der Dokumente im Workflow wird somit wesentlich effizienter gestaltet.

Team und Schungsumfeld

For- Das Forschungsprojekt READ wurde in einem Verbund durchgeführt, der gezielt industrielle Kompetenz und Innovationspotential aus Wissenschaft und Forschung bündelt. Zehn Partner aus Industrie, Großforschung und Universitäts- Bereich arbeiteten für die Erreichung der hochgesteckten Ziele erfolgreich zusammen .

Das Projekt wurde vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) im Rahmen des Fördergebiets "Intelligente Systeme" mit insgesamt 9,3 Millionen DM unterstützt. Das Projekt startete im August 1995 und endete im Juli 1998.

Schlußbericht Der vorliegende Schlußbericht beschreibt die Forschungsergebnisse von GMD IPSI und GMD SET im Rahmen des Projekts READ. Er ist folgendermaßen gegliedert:

Kapitel 1: Hier werden die Aufgabenstellungen und Voraussetzungen, unter denen die Arbeiten durchgeführt wurden, geschildert. Dazu gehören der Stand der Technik, der bekannt war oder zusätzlich zu erschließen war, und die wissenschaftlichen, technischen, personellen und sonstigen Rahmenbedingungen, die die Arbeit in READ sinnvoll ermöglichten.

Kapitel 2: Dieses Kapitel beschreibt die globale Planung der Forschungsarbeiten in READ und den detaillierten Ablauf der Arbeiten des Projektpartners GMD.

Kapitel 3: Hier werden die erzielten Forschungsergebnisse des Projektpartners GMD ausführlich beschrieben und deren Nutzen und Verwertbarkeit diskutiert.

Anhang: Der Anhang zum Schlußbericht gibt den Überblick über die Veröffentlichungen und Patente, die im Zusammenhang mit READ erstanden sind.

* Die Partner im Projekt READ sind: CGK Computer Gesellschaft Konstanz mbH, Daimler-Benz AG, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Forschungszentrum für Informationstechnik GmbH (GMD), Graphikon GmbH, Siemens AG, Siemens ElectroCom GmbH & Co., Technische Universität Braunschweig, Universität Koblenz-Landau und Otto-von-Guericke-Universität Magdeburg.

1. Aufgabenstellung und Ausgangsbedingungen des Vorhabens

1.1 Aufgabenstellung

AP 2150***Lernfähige Algorithmen***

Dieses Arbeitspaket ist ein Teil des AP 2100 - Objektbezogene Vorverarbeitung. Es sollen die vorhandenen Methoden auf dem Gebiet der gesteuerten Regelbasis Generierung untersucht werden, und durch eine geschickte Kombination von Methoden soll ein Verfahren zur automatisierten Regelbasisgenerierung entwickelt werden, das ein lokales Anpassen für Vor-Ort-Lernen ermöglicht. Damit soll der Lernprozeß deutlich verkürzt werden und die Anpassung der Klassifikation Regelbasis auch auf nicht trainierte Umgebungssituationen reagieren.

AP 2210***Anwendung von Fuzzy Techniken***

Dieses Arbeitspaket ist ein Teil des AP 2200 - Erkennung gebundener Handschrift. Im Rahmen des AP 2200 sollen unterschiedliche Methoden zur Erkennung gebundener Handschriften entwickelt, getestet und in einem Werkzeugkasten (Modular) zur Verfügung gestellt werden. Im AP 2210 soll eine Methodik basierend auf Fuzzy Logik entwickelt werden, die für unterschiedliche Zwecke (Anwendungen), z.B. Adressenleser, Scheckleser, etc. eingesetzt wird. Zusätzlich dazu soll das entwickelte Verfahren eine Vor-Ort- Lernkomponente anbieten, anhand der die Landspezifika angepaßt wird. Eine Fehlertoleranz Komponente soll leichte Fehler, wie z.B. Orthographie Fehler tolerieren.

AP2250***Kombination von Ergebnissen***

Anhand der erarbeiteten Ergebnisse in den AP2210 –AP2240 soll eine Kombinationsstrategie entworfen werden, die nach dem Baukasten Prinzip die Auswahl einer geeigneten Methode für den jeweiligen Datensatz (Einsatzort) ermöglicht. Dadurch soll eine erhöhte Leistungssteigerung der Erkennungssysteme erreicht werden.

AP3200***Extraktion stark strukturierter Information***

Dieses Arbeitspaket gliedert sich in zwei Bereiche. Im ersten Arbeitsabschnitt geht es um den 'Entwurf von Beschreibungsformalismen', während im zweiten Teil das 'Modellernen' im Vordergrund steht. Die Aufgabe bestand darin, einen Beschreibungsformalismus zu entwerfen, mit dem sowohl einzelne Dokumente, wie auch Dokumentklassen einheitlich repräsentiert werden können. Darüber hinaus sollte es möglich sein, Repräsentationen von Dokumentklassen weitestgehend automatisch aus Mengen von gelabelten [TC1] Dokumenten abzuleiten.

AP4200**Benchmarking**

Im Rahmen dieses Arbeitspaketes sollen Methoden zur Leistungsmessung der einzelnen Anwendungssysteme durchgeführt werden. GMD SET sollte sich hauptsächlich auf Methoden zur Leistungsmessung gebundener Handschrift konzentrieren, um diese Messungen zur Kombination von Klassifikatoren einzusetzen. Ein zweites Ziel ist die Verbesserung der einzelnen Module eines Klassifikators anhand der Fehler-Ursachen-Forschung einzelner Klassifikatoren.

AP5300**Dokumentlesen**

Dieses Arbeitspaket hatte die Aufgabe die entwickelten Ansätze prototypisch zu implementieren. Dafür mußte einerseits vorhandene Software verknüpft werden, andererseits Softwaremodule speziell zur Demonstration entwickelter Konzepte neu erstellt werden.

1.2 Voraussetzungen des Vorhabens

AP2150**Lernfähige Algorithmen****AP2210****Anwendung von Fuzzy Techniken**

Das automatisierte Briefsortieren ist ein wesentlicher Aspekt moderner Posteinrichtungen. Während die mit Schreibmaschine geschriebene Post schon automatisiert verteilt werden kann, ist es für gebundene Handschrift eher die Ausnahme. Auch diese Schrift soll mit der gleichen Geschwindigkeit, 50 ms pro Schriftstück, gelesen und sortiert werden.

Da die Klassifizierung oder Erkennung der Schrift nur ein Schritt in der Bearbeitungskette ist, wird als Eingabe für unser Vorhaben nicht die gescannte Bild Information eines Briefes genommen, sondern nur ein Ausschnitt - „region of interest„ - der in dem Arbeitspaket AP2100 extrahiert (vom Hintergrund, Stempel, etc. getrennt) und in das SEML Format umgewandelt wurde. Als Trainingsdaten wurden Standarddaten wie NIST und CEDAR-Datenbasen sowie Siemens Electrocom Daten zur Verfügung gestellt.

Ein global einsetzbares Handschrifterkennungssystem soll in der Lage sein, handgeschriebene Texte von verschiedenen Schreibern mit unterschiedlichen Spezifika (wie Alter, Nationalität, Ausbildung, Geschlecht, Beruf) mit gleichem Erkennungsgrad zu klassifizieren und zu interpretieren. Die meisten Probleme der existierenden Handschrifterkennungssysteme liegen in ihrer hohen Empfindlichkeit gegenüber der Varianz menschlicher Handschrift und ihre schlechte Rekonfigurierbarkeit in neue Umgebungen; wünschenswert wäre jedoch mehr Robustheit und eine schnelle lokale Anpassung.

Das menschliche Sehsystem ist sehr komplex und funktioniert sehr gut bei unterschiedlichen Bedingungen. Die Anpassung des Sehsystems an unterschiedliche Sehbedingungen wird anhand einer Reihe von Kompromissen erreicht. Die wesentlichen Eigenschaften sind: jeweilige lokale Anpassung der bekannten Merkmale an die lokale Umgebung; Information, die nicht aussagekräftig oder unvollständig ist, wird teilweise ignoriert [ORPE94]. Angelehnt an diese Erkenntnis-

se über biologische Systeme soll mit Hilfe der Fuzzy Logik ein mehrschichtiges Erkennungssystem entwickelt werden. Die Komplexität der einzelnen Schichten ist unterschiedlich.

AP2250***Kombination von Ergebnissen*****AP4200*****Benchmarking***

Auf dem Gebiet der Mustererkennung ist bekannt, daß die Multi-Klassifikatoren immer sehr gute Ergebnisse bringen, wenn jeder einzelne Klassifikator gute Ergebnisse bringt. Diese Regel ist nur dann gültig, wenn die Güte der einzelnen Klassifikatoren sehr hoch ist. Bei großen Güteunterschieden zwischen den Klassifikatoren führt ihre Kombination zu sehr schlechten Ergebnissen. Das bedeutet, daß die richtige Auswertung von unterschiedlichen Klassifikatoren eine wichtige Voraussetzung für eine sinnvolle Kombination der Klassifikatoren ist.

Die Güte des Benchmarks ist abhängig vom Gültigkeitsbereich der Aussage. Bei einem breiten Spektrum von Datensätzen können die Ergebnisse des Benchmarks fein differenzierbare Ergebnisse aufzeigen.

Genau wie im Fall eines Klassifikators sollen auch die Daten für die Benchmarks anhand ihrer Herkunft oder Fehlerquelle in Klassen eingeteilt werden. Die Beschreibung dieser Klassen geschieht mit Hilfe von Merkmalen, z. B. enthält die Klasse "Herkunft" solche Merkmale wie Land, Sprachräume, Ausbildung, Alter. Dadurch wandelt sich die Auswertung (Evaluierung) von Klassifikatoren aus einem passiven in einen aktiven Ansatz. Nicht nur die Schwächen oder Stärken der einzelnen Klassifikatoren gegenüber unterschiedlichen Datentypen können festgestellt werden, sondern es kann die Güte der einzelnen Bearbeitungsstufen in der Erkennungskette gemessen an den Merkmalen bestimmt werden. Diese Güte kann in eine apriori Wahrscheinlichkeit umgewandelt werden.

Die Güte der einzelnen Erkennungsstufen wird nicht nur durch die ausgewählten Merkmale der Datenklassifikation bestimmt, sondern auch durch die gewählten *Meßpunkte*. Die berechnete apriori Wahrscheinlichkeit der Erkennungsrate einzelner Klassifikatoren wird von der *Datenklassifikation* und der *Bewertungsfunktion* bestimmt. Eine Voraussetzung für die zügige Bewertung der Daten ist die Vollständigkeit der *Ground Truth Table* für die Trainingsdaten.

Es ist Aufgabe der Methodik, die Meßpunkte und Bewertungsfunktionen zu bestimmen und die Randbedingungen für gute Trainingsdaten und die Ground Truth Table aufzuzeigen.

AP3200***Extraktion stark strukturierter Information***

Typischerweise läßt sich in Dokumenten zwischen den Textbestandteilen unterscheiden, die eine starke geometrische Struktur und einfache syntaktische Konstrukte aufweisen und den frei formulierten Textabschnitten, die den Syntaxregeln der jeweiligen Sprache unterliegen. Im ersten Fall spricht man von stark strukturierter Information. Stark strukturierte Information zeichnet sich dadurch aus, daß

- ein enger Bezug zwischen geometrischer Position und Bedeutung für die Informationseinheiten vorliegt
- einfache, stereotype syntaktische Strukturen vorliegen
- sie nur eine begrenzte Semantik enthalten.

Solche stark strukturierten Informationseinheiten stellen zum Beispiel

- auf Geschäftsbriefen der Absender, der Adressat und das Datum
- auf technischen Publikationen der Titel und der Autor oder
- auf Zahlungsüberweisungen Konto- und Betragsfeld dar.

Die Extraktion stark strukturierter Information stellt die logisch konsequente Weiterführung der Formularerkennung dar. In der heute im Einsatz befindlichen Formularerkennung, bzw. Formularlesetechnik, leitet sich die Bedeutung der aus dem analysierten Formular ermittelten Daten direkt aus seiner geometrischen Position ab. Diese Vorgehensweise ist darauf angewiesen, daß die zu analysierenden Formulare und der Abtastprozeß sehr stabil die angenommenen geometrischen Bedingungen erfüllen. Diese Vorgehensweise scheitert, wenn die Erfassung der Information aus einer Vielzahl im geometrischen Aufbau sehr heterogenen Dokumenten vorgenommen werden soll. Hier sind neue Wege zu beschreiten, um die in den Dokumenten enthaltene Information, automatisch extrahieren zu können.

AP5300

Dokumentlesen

Weltweit waren Forschungsprototypen und handgemachte, spezialisierte Applikationen bekannt. Allerdings fehlte vor allem eine umfassendere Betrachtung der Problemstellungen in dem Gebiet, und man kann die Systeme daher alle nicht als sehr viel mehr betrachten, als OCR Anwendungen mit mehr oder weniger aufwendiger handgemachter Nachbearbeitung. Auf dem Gebiet der Parsergenerierung gibt es seit Ende der 60er Jahre Ansätze und bis heute Verbesserungen und Neuansätze. Das mag verdeutlichen, daß die wahren Probleme noch nicht gelöst werden konnten. Wesentlich für das Dokumentlesen ist aber vor allem, daß die Parsergenerierungstechnik in ihrer konventionellen Form nicht ausreichen kann und daher erweitert werden muß.

1.3 Wissenschaftlich-technischer Stand zu Beginn des Vorhabens

AP2150

Lernfähige Algorithmen

AP2210

Anwendung von Fuzzy Techniken

AP2250

Kombination von Ergebnissen

Erkennungssysteme zur Erkennung gebundener Handschrift verfolgen zwei Klassifikationsstrategien. Der erster Ansatz verfolgt die Erkennung des Wortes als Ganzes (holistischer Ansatz). Der vielversprechendste Ansatz ist zu diesem Zeitpunkt der Hidden Markov Model Ansatz, wenn auch noch weit von einer befriedigenden Lösung entfernt. Klassifikationsmethoden, anlehnend an Fuzzy Logik, ha-

ben zu diesem Zeitpunkt nicht befriedigend die automatisierte Regelgenerierung beantwortet.

Der zweite Ansatz besteht in einer Rückführung der Worterkennung auf Zeichenerkennung (zeichenbasierter Ansatz). Lösungsansätze tragen hier stark improvisierenden Charakter. Es liegen schon die ersten Ansätze vor, die auf modifizierten kontextfreien Grammatiken und Parsingstrategien beruhen, und die die Tendenz erkennen lassen, Parsingtechniken mit Eigenschaften wissensbasierter Systeme zu verbinden.

M.K. Brown: Preprocessing Techniques for Cursive Script Word Recognition, *Pattern Recognition Journal*, pp. 447-457, (16) 1983.

K.-P. Chan: Application of guarded fuzzy-attributed context free grammar to syntactic pattern recognition, *Int. Journal of Pattern Recognition and Artificial Intelligence*, 1992, 6, 5, pp. 777-797.

J. Zhou, T. Pavlidis: Discrimination of characters by a multi-stage recognition process, *Pattern Recognition Journal*, Vol. 27, 11, pp. 1539-1549, 1994.

A. Malaviya, L. Peters, M. Theißinger: FOHDEL – A new fuzzy language for online handwriting recognition, FUZZ-IEEE'94, 1994, Orlando, Florida, USA.

AP3200

Extraktion stark strukturierter Information

Zur Informationsextraktion aus strukturierten Texten werden typische KI-Techniken zum Bildverstehen eingesetzt, aber auch eher klassische Techniken aus dem Compilerbau, die man auch als syntaktische Klassifikatoren bezeichnen kann. Typische Systeme die KI Techniken verwenden, besitzen als Komponenten eine Wissensrepräsentationssprache, wie semantische Netze oder Frames (s. Bayer, Kreich, Yashiro), einen speziellen Inferenzmechanismus und einen Kontrollalgorithmus. Mit der Repräsentationssprache werden sowohl geometrische als auch inhaltliche Eigenschaften der zu erkennenden Strukturen modelliert. Zur Inferenz und Kontrolle werden logisch orientierte Verfahren eingesetzt (Kreich), die mit heuristischen Suchverfahren verbunden werden können (Bayer). Ein Blackboardmodell als grundlegende Systemarchitektur wird u.a. in den Arbeiten von Lam verwendet. Eher orientiert an Konzepten des Compilerbaus für Programmiersprachen sind die Arbeiten von Lorie. Hier liefert ein OCR System die terminalen Symbole, eventuell mit alternativen Erkennungsergebnissen. Eine kontextfreie Grammatik definiert die zu erkennenden Strukturen, nicht aber deren geometrische Eigenschaften, und ein Parser zerlegt den Eingabestrom in die zu extrahierenden Einheiten.

Dengel et.al.: A. Dengel, R. Bleisinger, F. Fein, R. Hoch, F. Hönes, M. Malburg: Officemaid -- A System for Office Mail Analysis, Interpretation and Delivery, in: *Proceedings of the 1st Workshop on Document Analysis Systems*, Kaiserslautern, 1994, pp. 253-275

Lorie: Raymond A. Lorie: A System for Exploiting Syntactic and Semantic Knowledge in Automatic Recognition, *Proceedings of the 1st Workshop on Document Analysis Systems*, Kaiserslautern, 1994, pp. 277-294.

Lam: Stephen W. Lam: An Adaptive Approach to Document Classification And Understanding, in: Proceedings of the 1st Workshop on Document Analysis Systems, Kaiserslautern, 1994, pp. 231-251.

Bayer: T.A. Bayer: Understanding Structured Text Documents By a Model Based Document Analysis System, in: Proceedings of the 2nd International Conference on Document Analysis and Recognition, Tokyo, 1993, pp. 448 – 453.

Kreich et.al.: J. Kreich, A. Luhn. G. Maderlechner: An Experimental Environment for Model based Document Analysis, in: Proceedings of the 1st International Conference on Document Analysis and Recognition, Saint-Malo, 1992, pp. 50 -58.

Yashiro et.al.: Yashiro, H., Murakami, T., Shima, Y., Nakano, Y., Fujisawa, H.: A New Method of Document Structure Extraction Using Generic Layout Knowledge, in Proceedings of the International Workshop of Industrial Applications of Machine Intelligence and Vision (MIV-89), Tokyo, 1989, pp. 282-287,

AP4200

Benchmarking

In dem Forschungsbereich „Benchmarking“ gibt es einige Ansätze, die schwerpunktmäßig die Einzelzeichenerkennungsebene betrachten. Ziel der laufenden Aktivitäten auf diesem Gebiet ist, den Vergleich unterschiedlicher Klassifikatoren im Rahmen von Wettbewerben zu ermöglichen. Das führt dazu, daß es keine Methoden zur Leistungsmessung gibt, die eine Bewertung einzelner Module einer Erkennungskette ermöglichen und somit die Schwachpunkte einzelner Schritte erkennen und bei der Behebung dieser unterstützen.

Folgende Vorhaben sind in diesem Zusammenhang von Interesse: ISRI'93: 1993 Annual Report, Univ. of Nevada, Las Vegas, 1993. ISRI'94, 1994 Annual Report, University of Nevada, Las Vegas, 1994.

AP5300

Dokumentlesen

Alle drei an diesem Paket beteiligten Partner (Siemens, München, DFKI und GMD IPSI) verfügten zu Beginn des Projektes über prototypische, funktionsfähige Softwaremodule. Während der Projektlaufzeit wurden neuere Versionen oder ganz neue Module entwickelt.

Kreich et.al.: J. Kreich, A. Luhn, G. Maderlechner: An Experimental Environment for Model based Document Analysis, in: Proceedings of the 1st International Conference on Document Analysis and Recognition, Saint-Malo, 1992, pp. 50 - 58

Klein, B., Fankhauser P., Error tolerant Document Structure Analysis, in: International Journal on Digital Libraries, 1(4), 1997.

1.4 Zusammenarbeit mit anderen Stellen

Siemens ZT (vorher ZFE), Daimler Benz Forschungszentrum Ulm, DFKI , TU Braunschweig, Universität Koblenz-Landau

2. Planung und Ablauf des Vorhabens

AP2150**Lernfähige Algorithmen****AP2210****Anwendung von Fuzzy Techniken****AP2250****Kombination von Ergebnissen****AP4200****Benchmarking**

Die vier Arbeitspakete sind miteinander verbunden über den gewählten Fuzzy Ansatz. Anhand der Spezifikation, die in AP2150 entwickelt wurde, konnte in AP 2210 ein Verfahren entwickelt werden, das diese Spezifikation einhält. Aus diesem Grund wurden verschiedene Lernmethoden in parallel zu dem entwickelten Klassifikationsverfahren untersucht und getestet. Während im ersten Drittel des Projektes der Schwerpunkt bei der Entwicklung des Klassifikators lag, wobei zuerst mit einzelnen Zeichen angefangen wurde, liegt im zweiten Drittel des Projektes der Schwerpunkt auf der Leistungsmessung und der Entwicklung einer Methodik zur Fehlerursachenforschung.

Aus den Erkenntnissen und den Ergebnissen der ersten zwei Jahre wurde im dritten Projektjahr der ursprüngliche Fuzzy Klassifikator für die Erkennung von Zeichen auf Wörtern erweitert. Gleichzeitig wurde die ausgearbeitete Leistungsmethode eingesetzt zur Entwicklung eines Ansatzes, das die Kombination unterschiedlicher Klassifikatoren anhand von Leistungsmessung ermöglicht.

Die entwickelten Methoden zur Klassifikation und Leistungsmessung, und der aufgebaute Demonstrator zur Worterkennung wurde im Rahmen der Abschlußpräsentation vorgetragen und demonstriert.

AP3200**Extraktion stark strukturierter Information****AP5300****Dokumentlesen**

Die Extraktion stark strukturierter Information wurde von GMD IPSI während der gesamten Projektlaufzeit kontinuierlich bearbeitet, da man das Arbeitspaket Dokumentlesen als die Umsetzung der vorherigen Arbeiten ansehen kann. Im ersten Drittel wurde ein Formalismus zur Beschreibung der stark strukturierten Information entwickelt. Dann wurden Methoden entwickelt, um auf Basis dieses Formalismus Software zu designen, und mit diesen Methoden wurde ein Softwareprototyp geschrieben. Im Arbeitspaket Dokumentlesen wurde die Funktionsfähigkeit der Software demonstriert.

3. Ergebnisse

3.1 Wissenschaftlich-technische Ergebnisse

AP2150 **Lernfähige Algorithmen**

AP2210 **Anwendung von Fuzzy Techniken**

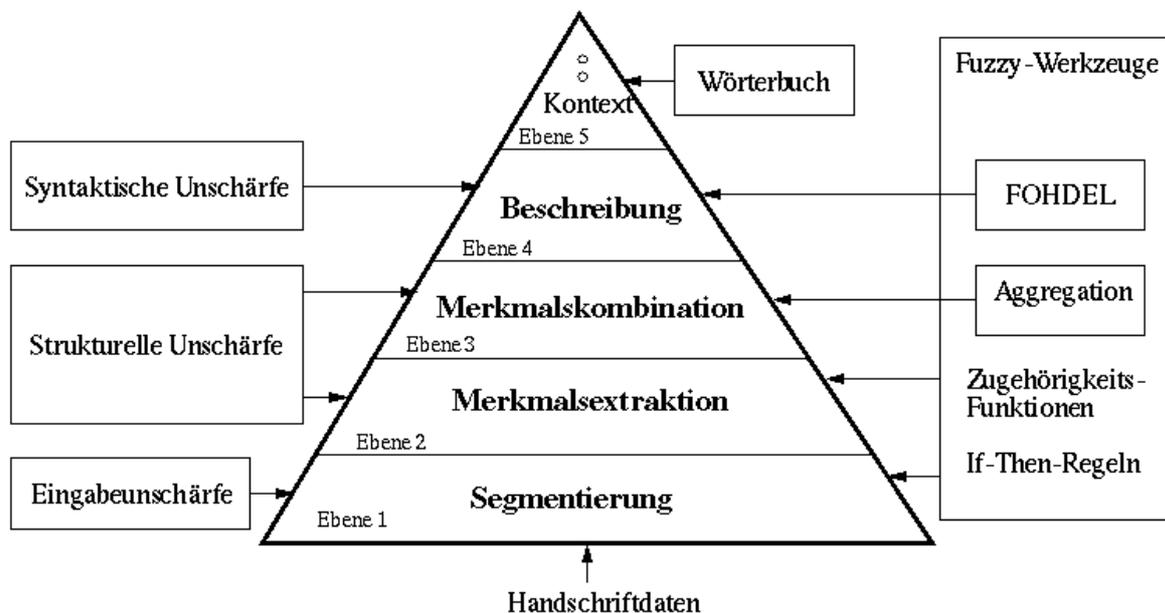


Abbildung 1: Konzept des Fuzzy-Zeichenerkennungssystems

In der vorgeschlagenen Methode haben wir die Aufgabe der Handschrifterkennung oder Klassifizierung in mehrere Ebenen eingeteilt (s. Abbildung 1), wobei jede Ebene im eigenen linguistischen Wertraum [MAPE95] dargestellt ist. Diese Aufteilung gibt die zusätzliche Möglichkeit, den Datenaustausch ohne Verlust zwischen den einzelnen Ebenen durchzuführen, weil der Ausgangswertebereich einer Ebene zugleich der Eingangswertebereich der nächsten Ebene ist. Auch die Rückverfolgung der Daten wird damit transparent, und die Lokalisierung von Fehlern wird leichter gemacht. Dieser mehrschichtige Ansatz wurde ursprünglich für unser Online Handschrifterkennungssystem FOHRES entwickelt [MAPE97-2]. Im Rahmen des READ Projektes wurde der Ansatz erweitert und den Gegebenheiten der Off-Line Handschrifterkennung angepaßt. Der Schwerpunkt der Arbeiten hat sich auf die Entwicklungen der automatisierten Regelgenerierung von Fuzzy Regelbasen verlagert.

Der Fuzzy Klassifikator ist ein regelbasierter Ansatz, der die Zugehörigkeit einzelner Merkmale eines Zeichens oder Wortes zu den vordefinierten Regeln feststellt. Das Zeichen oder Wort, dessen Regeln am besten auf das unbekannte Zeichen oder Wort paßt, wird mit der entsprechenden Wahrscheinlichkeit als „erkannt“, ausgegeben. Wenn im Falle der Zeichen die Regeln – außer für 0 (Null) und O (Buchstabe) – ausreichend sind, muß bei Wörtern ein Wörterbuch den Klassifikator unterstützen. Hier wird aus der Schrift versucht zu erkennen, welche der im Wörterbuch beschriebenen Wörter erkannt werden. Die Beschreibung wird anhand von Merkmalen festgehalten, die somit die Anzahl der möglichen Merkmalskombinationen mit jedem Erkennungsschritt beschränken, bis ein Wort erkannt wird. Je schlechter die Erkennungsmöglichkeit ist, desto größer ist die Anzahl der gezeigten Möglichkeiten, um aus zusätzlicher oder redundanter Information (z.B. Postleitzahl) die Erkennungsrate zu erhöhen [MALP96]. Im folgenden wird der Ansatz von GMD-SET näher erläutert.

Konzeptuell existieren zwei verschiedene Ansätze zur Worterkennung: Der erste Ansatz basiert auf einem holistischen Erkennungsansatz. Dies bedeutet, daß ein Wort als ein Muster angesehen wird, das als Ganzes erkannt werden soll. Der zweite Ansatz ist ein zeichenbasierter Erkennungsansatz, d.h. man liest das Wort durch Erkennen einzelner Zeichen des Wortes. Prinzipiell benutzt man in einem regelbasiertem Ansatz eine Regelbasis, die für isolierte Einzelzeichen geschrieben bzw. generiert wurde. Bei der Erkennung des Wortes muß das gebunden geschriebene Wort zunächst in einzelne Buchstaben segmentiert werden.

Zeichensegmentierung

Wie in sehr vielen Anwendungen der Bilderkennung ist die Effizienz von anfänglichen Low-Level-Bearbeitungsschritten oft kritisch für die endgültigen Erkennungsergebnisse. Bei der Erkennung von gebundener Handschrift mittels zeichenweiser Worterkennung kann die anfängliche Segmentierung des Wortes in Buchstaben den Unterschied zwischen sehr guten und sehr schlechten Ergebnissen ausmachen [OGKA97].

Das Ziel der Zeichensegmentierung ist die Erkennung von einzelnen Buchstaben in einem Wort. Ein sehr häufig benutzter Ansatz bei der Zeichensegmentierung, der aus dem Gebiet der Erkennung gedruckter Schrift stammt, ist die Berechnung eines vertikalen Projektionsprofils (siehe Abbildung 2). Das vertikale Projektionsprofil für ein Bild eines Schriftzuges ist ein Histogramm der Anzahl der schwarzen Pixel, die vertikal akkumuliert werden beim horizontalen Durchlauf über den Schriftzug. Dieses Profil wird niedrige Werte zwischen Wörtern, höhere Werte innerhalb Wörtern und sehr hohe Werte bei Buchstaben besitzen, die entweder nach oben oder unten herausragen.

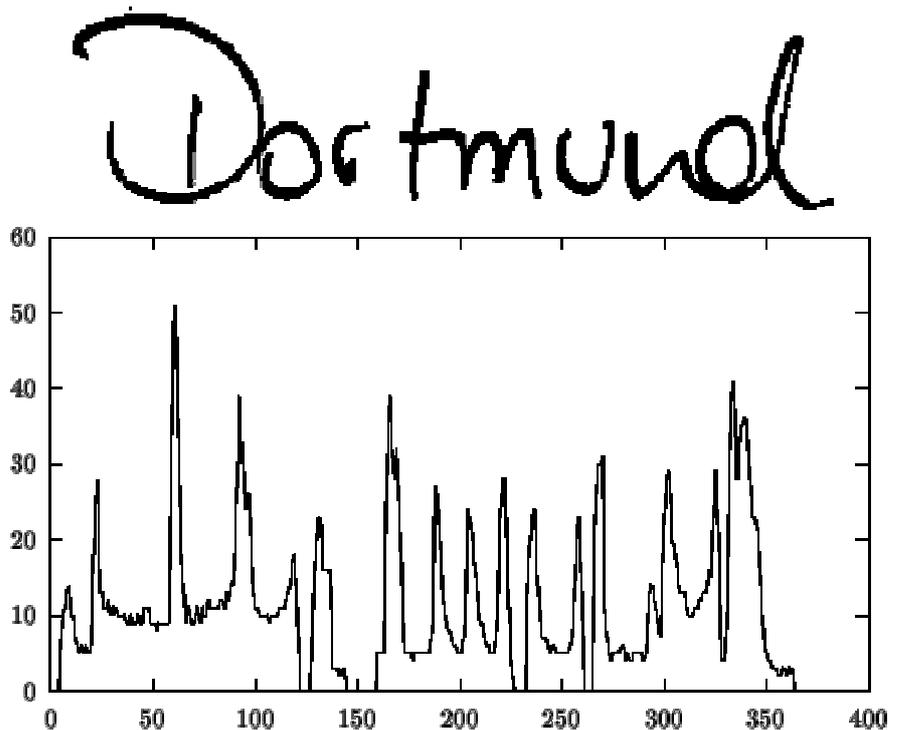


Abbildung 2: Berechnung des vertikalen Projektionsprofils zur Zeichensegmentierung: Oben ist das Eingabebild gegeben, für das das untere vertikale Projektionsprofil erzeugt wurde.

Probleme bei der Zeichensegmentierung ergeben sich durch einige Eigenschaften handgeschriebener Dokumente. Die folgenden Charakteristiken von Handschrift verkomplizieren die Segmentierung:

- *Schräggeschriebene Schrift*: Jeder Mensch besitzt einen individuellen Schreibstil. Daher können viele verschiedene Neigungswinkel bezogen auf eine vertikale Bezugslinie in handgeschriebenen Worten auftauchen. Manchmal tritt sogar der Fall auf, daß der Neigungswinkel innerhalb eines Wortes variiert. Dies erschwert die Zeichensegmentierung mittels der vertikalen Projektionsmethode erheblich.
- *Nichthorizontale Schrift*: Einige Schriftzüge haben die Eigenschaft, daß die Schrift gegenüber einer horizontalen Bezugslinie einen Neigungswinkel aufweist. Besonders oft tritt dieser Fall ein, wenn die Grundlage der Schrift keine horizontalen Linien aufweist, wie es oft bei Briefumschlägen der Fall ist.
- *Unterstreichungen*: Es ist eine weitverbreitete Praxis, daß Schreiber verschiedene Wörter unterstrichen, um ihre Wichtigkeit hervorzuheben.
- *Verbindungsformat*: Durch die unterschiedlichen Weisen zu schreiben, hat jeder Mensch eine eigene Art, einzelne Buchstaben zu schreiben, aber auch eine individuelle Art, diese zu verbinden. Wegen dieses Verbindungsformats kann es passieren, daß ein o als a erkannt wird. An dieser Stelle ist eine gute Segmentierung erforderlich, da eine Fehlsegmentierung zur Ablehnung des Wortes bei der Erkennung führt.

- *Gebrochene Zeichen*: Einige Zeichen werden oftmals als nichtzusammenhängend geschrieben. Zum Beispiel ist die zweite horizontale Linie des Buchstabens H oft nicht zusammenhängend mit dem Rest des Buchstabens geschrieben. Ähnliches kann auch bei den Buchstaben T, R, B usw. auftreten. Dies kann zu Fehlsegmentierungen führen, da die betreffenden Buchstaben in mehrere unterteilt werden.
- *Überlappende Zeichen*: In anderen Fällen überlappen sich einige Zeichen, so daß die Trennung der Buchstaben wiederum erschwert wird.
- *Hintergrund*: Die Schrift ist manchmal auf nichteinheitlichen Hintergrund geschrieben. Bei der notwendigen Entfernung des Hintergrundes durch spezielle Filterungsmethoden kann es dabei zu Fehlern kommen, die zusätzliche Punkte zur Schrift hinzufügen.

Auf Grund dieser Eigenschaften handgeschriebener Dokumente werden bei der hier beschriebenen Applikation vor der vertikalen Projektionsmethode weitere Vorbereitungsschritte eingeführt:

Zunächst wird die Schrift vom Hintergrund getrennt. Nach diesem sogenannten Binärisierungsschritt wird eine Glättung vorgenommen, um Fehler der Binärisierung aufzuheben. Als nächstes wird die vertikale Neigung der Schrift korrigiert. Dabei existieren Ansätze, die auf Projektionsmethoden [GUSU94] oder Chain-Code-Methoden [LESR93] basieren. Unsere Methode berechnet die obere und untere Kontur der Schrift und benutzt diese, um die Neigung der Schrift zu bestimmen, und das Schriftbild dementsprechend zu drehen. Ein Beispiel der Berechnung der Kontur der Schrift ist in Abbildung 3 gegeben. Näheres zu unserem Ansatz kann in [PEGI99] gefunden werden.

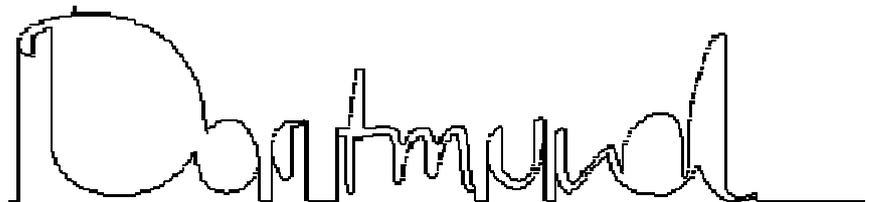


Abbildung 3: Die obere und untere Kontur eines Schriftzuges.

Nachdem die vertikale Neigung der Schrift korrigiert wurde, kann das Bild immernoch einen Winkel bezüglich einer horizontalen Bezugslinie einschließen. Da wir einen zeichenbasierten Ansatz verfolgen, und unsere Regelbasis, wie später noch erklärt wird, jedoch aus Zeichen antrainiert wurde, die horizontal geschrieben wurden, sollte die Grundlinie in dem zu erkennenden Schriftzug ebenfalls horizontal sein. Ein verbreiteter Ansatz zur Bestimmung der Grundlinie basiert auf Senior [SENI94]. Unser Ansatz benutzt die untere Kontur der Schrift, um die Grundlinie zu bestimmen. Näheres dazu kann in [PEGI99] nachgelesen werden.

Nachdem das Bild bezüglich der vertikalen Neigung gedreht wurde, die Grundlinie bestimmt wurde, und dementsprechend das Bild horizontal gedreht wurde, wird nun die Segmentierung vorgenommen. Dabei benutzen wir neben der vertikalen Projektionsmethode

wiederum die obere und untere Kontur der Schrift, um den Schriftzug in einzelne Zeichen zu segmentieren [PEGI99]. Die gefundenen Zeichen sind nun horizontal geschrieben, so daß nun die generierte Regelbasis gut zur Erkennung der Zeichen geeignet ist.

Automatische Regelbasis-Generierung in FOHDEL

Der in unserer Arbeitsgruppe verfolgte Ansatz basiert auf einem regelbasierten System, welches weiterhin die Vorzüge der Fuzzy-Logik zur Mustererkennung benutzt. Regelbasierte Systeme klassifizieren unbekannte Muster auf der Grundlage von einigen, in einer Beschreibungssprache formulierten, Regeln, die in der Regelbasis des Systems zusammengefaßt sind. Wir haben dazu die in unserem Institut entwickelte Fuzzy-Beschreibungssprache *FOHDEL*[†]. In FOHDEL werden Fuzzy-Merkmale, linguistische Terme, Fuzzy-Modifizierer und Fuzzy-Operatoren im grammatikalischen Inferenzprozeß benutzt [MALA96][MAPT94]. Durch FOHDEL wird ein Rahmen hauptsächlich zur Beschreibung handgeschriebener Symbole bereitgestellt, welcher zur Klassifizierung benutzt werden kann. Eine Beispielsregel, die in FOHDEL formuliert ist, ist in Abbildung 4 dargestellt.

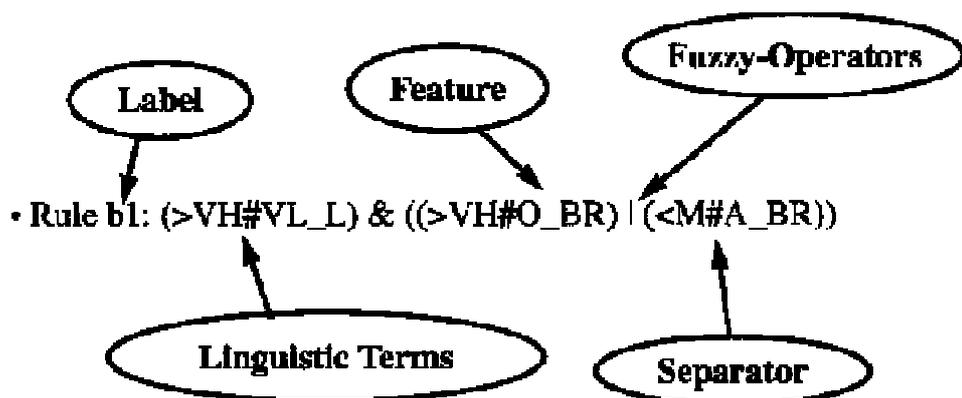


Abbildung 4: Eine Beispielsregel formuliert in FOHDEL

Oft sind die Regeln eines regelbasierten Systems, das auf Fuzzy-Logik basiert, aufgebaut auf dem Wissen eines menschlichen Experten. Die vom Experten aufgestellten Regeln werden dann meist mittels der trial-and-error-Methode verfeinert. Je komplexer jedoch ein System ist, desto schwieriger ist die genaue Vorhersage, wie das System auf die Änderungen reagieren wird. Weiterhin handelt es sich beim Entwurf komplexer Systeme um einen äußerst langwierigen Prozeß, der zudem nicht eine optimale Lösung garantiert. Deshalb ist es ein großes Anliegen vieler Wissenschaftler auf diesem Gebiet, einen Ansatz zu finden, der eine automatisierte Regelgenerierung ermöglicht. Diese Zielsetzung ist bis heute immer noch nicht befriedigend gelöst [LEJA96].

[†] Fuzzy On-Line Handwriting Description Language.

Der folgende Lernalgorithmus stellt einen neuartigen Ansatz zur Lösung dieser Problematik dar. Die meisten bisherigen Algorithmen basieren auf separatistischen Ansätzen, die Muster klassifizieren, indem sie die Unterschiede der verschiedenen Musterklassen hervorheben. Der hier präsentierte Ansatz versucht Muster zu klassifizieren, indem er sich auf die charakteristischen Besonderheiten der jeweiligen Musterklasse bezieht. Die automatische Regelgenerierung basiert auf einigen Trainingsdaten, die durch Fuzzy-Clusteranalysen bearbeitet werden. Aufgrund der hohen Dimensionsanzahl der Eingabedaten wird eine mehrphasige Clusteranalyse eingeführt [IVMP98].

Der Lernalgorithmus, der in zwei Teile aufgeteilt ist, ist in Abbildung 5 zusammengefaßt. Der erste Teil generiert eine anfängliche Regelbasis, wobei zunächst Regeln für jede einzelne Musterklasse individuell erzeugt werden (Schritte 1 - 4). Jede Regel besteht aus einigen Merkmalen, die die Musterklasse beschreiben. Es werden linguistische Terme zu den Merkmalen assoziiert, um deren Qualität in der betrachteten Musterklasse zu präzisieren. Im zweiten Teil des Algorithmus (Schritt 5) werden die generierten Regeln nachgeprüft, um mögliche Überlappungen der Regeln, sogenannte Regelkonflikte, aufzufinden. Die überlappenden Regeln werden verändert, genauer gesagt erweitert, bis eine gewisse Unterscheidbarkeit erreicht ist oder aber die Merkmale nicht ausreichend sind, um die Musterklassen voneinander zu trennen. In diesem Fall ist es notwendig, daß der Benutzer weitere Merkmale hinzufügt oder aber mehr Kontextinformationen zur Verfügung hat. Der Lernalgorithmus schreibt als Ausgabe direkt eine Regelbasis, die in der Sprache FOHDEL formuliert ist.

Eingabe:	<ul style="list-style-type: none"> • eine Menge A der zu erkennenden Muster • Menge von Trainingsdaten Γ mit $\forall \gamma \in A : X(\gamma) \in \Gamma$ • $\forall \gamma \in A$ p-dimensionale Fuzzy-Merkmalvektoren, d.h. $X(\gamma) \subseteq [0, 1]^p$
Ausgabe:	Fuzzy-Regelbasis formuliert in FOHDEL
Methode:	<pre> $\forall \gamma \in A$ <u>do</u> (1) Gruppierung der Trainingsdaten \forall Gruppen <u>do</u> (2) Einschränkung der Merkmalsmenge (3) Multiphasen-Clusteranalyse (4) Regelformulierung in FOHDEL <u>od</u> <u>od</u> (5) Nachprüfung der Regeln </pre>

Abbildung 5: Algorithmus zur automatischen Regelbasengenerierung

Zeichenklassifikation

Zunächst einmal betrachte man die Klassifikation eines Zeichens. Dabei benutzt man die zuvor generierte Regelbasis, wie es im vorherigen Abschnitt beschrieben wurde. Die Erkennung ist in mehrere Schritte unterteilt. Nachdem in einem ersten Bildvorverarbei-

tungsschritt einige Operationen wie z.B. Thinning durchgeführt werden, wird das Zeichen in einzelne Segmente unterteilt. Diese beiden Schritte werden als Preprocessing und Segmentierung bezeichnet, wobei der implementierte Ansatz zur Segmentierung auf der Arbeit von Watkins [WATK97] basiert. Die einzelnen Segmente werden dann an einen Merkmals-Extrahierer übergeben. Für jedes Segment wird ein Fuzzy-Merkmalsvektor berechnet, d.h. es wird für jedes Merkmal ein Fuzzy-Zugehörigkeitswert für jedes einzelne Segment berechnet. Diese Vektoren werden zusammengefaßt zu einem einzigen Vektor x . Nach der Merkmals-Extraktion hat man also zu dem Eingangsbild einen Merkmalsvektor berechnet.

Zur Klassifikation des Eingabebildes wird der berechnete Merkmalsvektor x betrachtet. Dieser wird an den Klassifizierer übergeben. Mit Hilfe der zum System gehörenden Regelbasis wird zu jeder Fuzzy-Regel der Regelbasis eine Fuzzy-Zugehörigkeit des Bildes, genauer gesagt des Merkmalsvektors, berechnet. Das korrespondierende Zeichen der Fuzzy-Regel, deren Fuzzy-Zugehörigkeit maximal ist, wird als erkanntes Zeichen durch den Zeichenklassifizierer zurückgegeben. Sollten jedoch die Zugehörigkeiten aller Regeln einen gewissen Threshold-Wert nicht übersteigen, so erkennt der Klassifizierer kein Zeichen. Ein Beispiel eines gesamten Durchlaufs der Klassifikation eines Zeichens durch einen Zeichenklassifikator ist in Abbildung 6 dargestellt.

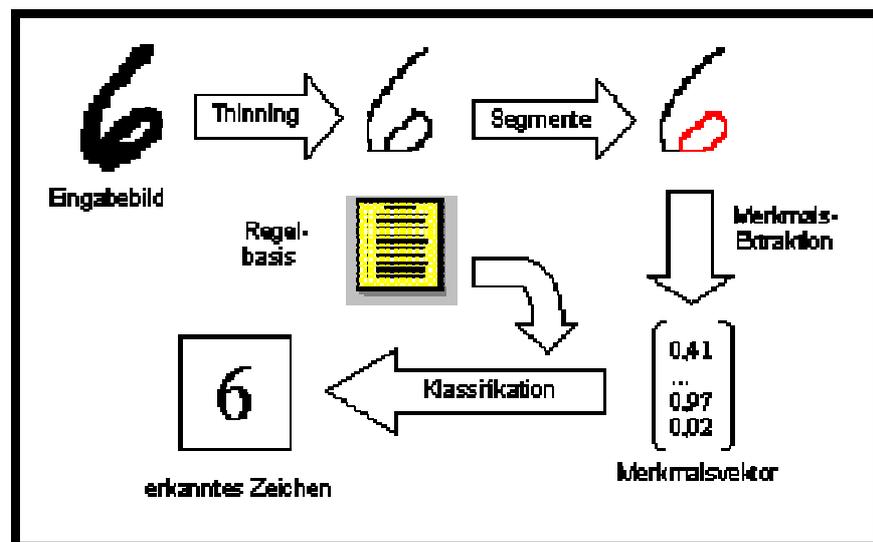


Abbildung 6: Die verschiedenen Schritte des Zeichenklassifikators

Nachdem in den Vorverarbeitungsschritten die Zeichen des Wortes in dem Schriftzug erkannt, und dementsprechend segmentiert wurden, kann nun der Zeichenklassifizierer, der durch die zuvor beschriebene Regelbasis-Generierung definiert ist, eingesetzt werden. Neben möglichen Segmentierungsfehlern können nun noch Fehler in der Erkennung durch den Klassifizierer dazukommen. Die Studien in [KKMS97] zeigen, daß heutige Zeichenklassifikations-Systeme vergleichbar gute Ergebnisse wie der Mensch erzielen. Dennoch können heutige Worterkennungssysteme bei Weitem nicht die Leistung des Menschen erreichen. Dies zeigt, daß der Mensch weitere Informationsquellen ausschöpft, um die benötigte Information

zu erhalten. Daher ist die Hinzunahme von Kontextinformationen notwendig, um eine komperative Erkennungsleistung zu erreichen.

Einsatz eines Wörterbuches und Spitz-Kodierung

Handschrifterkennungssysteme benutzen verschiedene Wissens Ebenen, um die Erkennungsergebnisse zu steigern. Die inhärente Undeutlichkeit von Handschrift beschreibt die Notwendigkeit für derartiges Wissen für das System auf der Ebene von Wörtern, Syntax, Sätzen und Semantik [MIBP99]. Diese Wissensbasen werden beim textabhängigen Nachbearbeiten der Ergebnisse des Klassifizierers eingesetzt.

Zunächst präsentiert der Klassifizierer für jeden Buchstaben eine Menge von möglichen Zeichen, zusammen mit den berechneten Fuzzy-Zugehörigkeitswerten. Die Benutzung eines Wörterbuchs als semantische Informationsquelle kann sehr hilfreich bei der Erkennung von Handschrift sein. Dabei werden alle legalen Zeichenketten als Eintrag in das Wörterbuch aufgenommen. Dieses Wörterbuch kann als Expertenwissen auf der Wortebene angesehen werden. Auf der nächsten Ebene kann Kontextwissen auf Satzebene benutzt werden, um zu bestimmen, welches Wort in einem bestimmten Kontext besser geeignet ist als ein anderes. Eine Wissensbasis auf der Syntaxebene kann bestimmen, welches Wort grammatikalisch gesehen besser zu dem umgebenden Text paßt, während auf der semantischen Ebene die Bedeutung von Phrasen eine Rolle spielt [MIBP99].

Der meistverbreitete Gebrauch von Kontextinformation ist auf der Wortebene. Dabei wird die Gültigkeit eines Buchstabens im Hinblick auf die ihn umgebenden Buchstaben bestimmt. Die Zugehörigkeitswerte, die das System mit den klassifizierten Zeichen und ihren Alternativen zurückgibt, werden dann benutzt, um Zugehörigkeitswerte der resultierenden Kandidaten zu berechnen. Eine eingehendere Diskussion weiterer Methoden und Strategien zur Benutzung von Kontextinformationen bei der Worterkennung wird in [LEBA94] gegeben.

Ein Problem bei der Benutzung von Wörterbüchern, um die Erkennungsrate zu steigern, ist meist die große Menge an legalen Zeichenketten. Neben der Möglichkeit, daß Wörterbuch entsprechend des zu lesenden Textes auszuwählen, und somit die Erkennungsleistung zu steigern (siehe dazu [SPIT97]), benutzen wir eine sogenannte Spitz-Kodierung zur Verbesserung der Erkennung. Betrachten wir den Fall des Adreßlesens. Wir wollen nun den Stadtnamen lesen. Daher besteht das Wörterbuch nun aus allen möglichen Städten, zum Beispiel allen Städten der USA.

Obwohl dieses Wörterbuch wenige legale Einträge relativ zu allen Einträgen besitzt, so treten immernoch zu viele Mehrdeutigkeiten auf. Dies wird dadurch begünstigt daß die Segmentierung des gebundenen Textes eine recht schwierige Aufgabe ist. Daher wurde das Prinzip der Word Shape Token von Spitz, das er in [SPIT95] [SPIT97] als Hilfe zur Erkennung von gedruckter Schrift vorgestellt hat, erweitert und auf den Fall von gebundener Schrift verallgemeinert. Mit Rücksicht auf den Ideengeber unserer verallgemeinerten Form der Kodierung nennen wir sie Spitz-Kodierung.

Bei unserer Spitz-Kodierung wird die Menge der Kleinbuchstaben in vier Untermengen eingeteilt. Dabei unterscheiden wir die Buchstaben nach ihrer Form. Das kleine f bildet eine Klasse für sich, da es das einzige kleine Zeichen ist, das unter die Grundlinie der Schrift reicht und gleichzeitig über die Mittellinie bis zur Oberlinie der Schrift ragt. Zur Verdeutlichung der Begriffe Grundlinie, Mittellinie und Oberlinie der Schrift betrachte man Abbildung 7. Die zweite Klasse von Buchstaben umfaßt die Buchstaben g, j, p, q und y, die allesamt unter die Grundlinie ragen. Die dritte Klasse besteht aus den Buchstaben b, d, h, k, l und t, die über die Mittellinie hinaus geschrieben werden. Die restlichen Kleinbuchstaben werden in die vierte Klasse eingeteilt. Diese Buchstaben werden zwischen Grundlinie und Mittellinie geschrieben. Die Aufteilung der Kleinbuchstaben ist in Tabelle 1 zusammengefaßt. Um Wörter zu kodieren, werden Großbuchstaben zu der Klasse der über die Mittellinie ragenden Kleinbuchstaben hinzugefügt. Beispielskodierungen sind in Tabelle 2 gegeben.

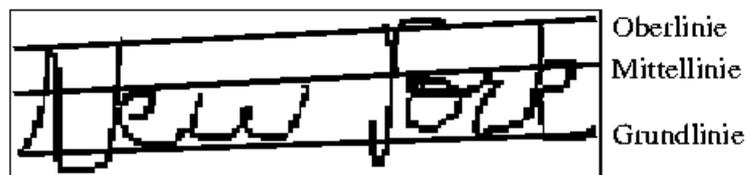


Abbildung 7: Grundlinie, Mittellinie und Oberlinie eines Schriftzuges

Buchstabe	f	g j p q y	b d h k l t	a c e i m n o r s u v w x z
Spitz-Kodierung	f	G	X	a

Tabelle 1: Spitz-Kodierung von Kleinbuchstaben

Wort	Hadley	Amherst	Northampton
Spitz-Kodierung	XaXXag	XaXaaaX	XaaXXaagXaa

Tabelle 2: Beispiele der Spitz-Kodierung für Stadtnamen

Mit Hilfe der Spitz-Kodierung läßt sich, wie bereits erwähnt, das Wörterbuch verkleinern. Dabei betrachtet man auf Low-Level-Ebene umgebende Boxen der Buchstaben. Nach den zuvor beschriebenen Drehungen des Bildes und der Korrektur der Grundlinie der Schrift, wird die Schrift in Zeichen segmentiert. Zu jedem gefundenen Zeichen kann man dann eine umgebende Box bestimmen. Wir bestimmen die Spitz-Kodierung des Zeichens durch einfaches Betrachten der umgebenden Box des Zeichens relativ zu den umgebenden Boxen der Nachbar-Zeichen und des ganzen Wortes.

Auf diese Art und Weise berechnen wir eine Spitz-Kodierung des zu erkennenden Schriftzuges. Diese Berechnung erfolgt auf Low-Level-Ebene, ist dadurch schnell und vermindert dennoch den

Lösungsraum effektiv. Das Wörterbuch wird derart vermindert, daß nur noch Wörter enthalten sind, die einerseits legale Zeichenketten darstellen, und andererseits die zugehörige Spitz-Kodierung des Wortes der berechneten Spitz-Kodierung des Schriftzuges ähnelt. Man darf dennoch nicht zu viele Wörter aus dem Wörterbuch entfernen, da die Betrachtung der Spitz-Kodierung auf Low-Level-Ebene durchgeführt wird, und man nicht von einer fehlerfreien Segmentierung ausgehen kann. Ein Beispiel dieser Vorverarbeitungsschritte ist in Abbildung 8 gegeben.

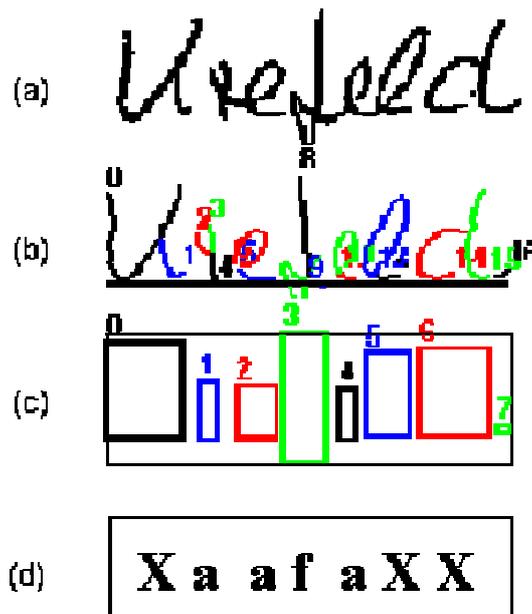


Abbildung 8: Die Vorverarbeitungsschritte im Worterkennungsprozeß: (a) das Ausgangsbild, (b) das verdünnte, gedrehte und der Grundlinie entsprechend korrigierte Bild, (c) die umgebenden Boxen der gefundenen Zeichen und (d) die berechnete Spitz-Kodierung des Schriftzuges.

Nun werden die gefundenen Zeichen entsprechend des zeichenbasierten Worterkennungsansatzes einzeln an einen Zeichenklassifizierer übergeben, der für jedes Zeichen eine Anzahl von Möglichkeiten mit samt den Fuzzy-Zugehörigkeitswerten zurückliefert. Dabei wird die berechnete Spitz-Kodierung für das segmentierte Zeichen auch an den Klassifizierer übergeben, da dadurch weniger Möglichkeiten zur Erkennung gegeben sind. Einerseits erhöht dies die Erkennungsleistung des Systems, da weniger Fehler auftreten können, andererseits beschleunigt dies auch das System. Desweiteren wird das Wörterbuch entsprechend der aus dem Schriftzug berechneten Spitz-Kodierung verkleinert. In einem abschließenden Schritt werden dann die erkannten Zeichen und ihre Zugehörigkeitswerte kombiniert, und mit dem Wörterbuch verglichen. Auf diese Art und Weise wird die Kombination der Erkennung mit der semantischen Information des Wörterbuchs bewerkstelligt, die als Ergebnis dann eine Lösung und einen Zuverlässigkeitswert liefert. Als Beispiel betrachte man die Tabelle 3, die zur Abbildung 8 durch das System berechnet wurde. Die grafische Oberfläche des

Worterkennungssystem, das hier beschrieben wurde, ist in Abbildung 9 gegeben.

Zeichen Nr	Erste Wahl		Zweite Wahl		Erkanntes Wort: Krefeld 85 %
	c	z	c	z	
1	K	1.00	U	0.50	
2	r	0.54	v	0.50	
3	e	0.99	o	0.43	
4	f	0.99			
5	e	0.74	i	0.22	
6	l	0.93	k	0.27	
7	d	0.79			

Tabelle 3: Die erkannten Zeichen c mit ihren Zugehörigkeitswerten z, und das erkannte Wort.

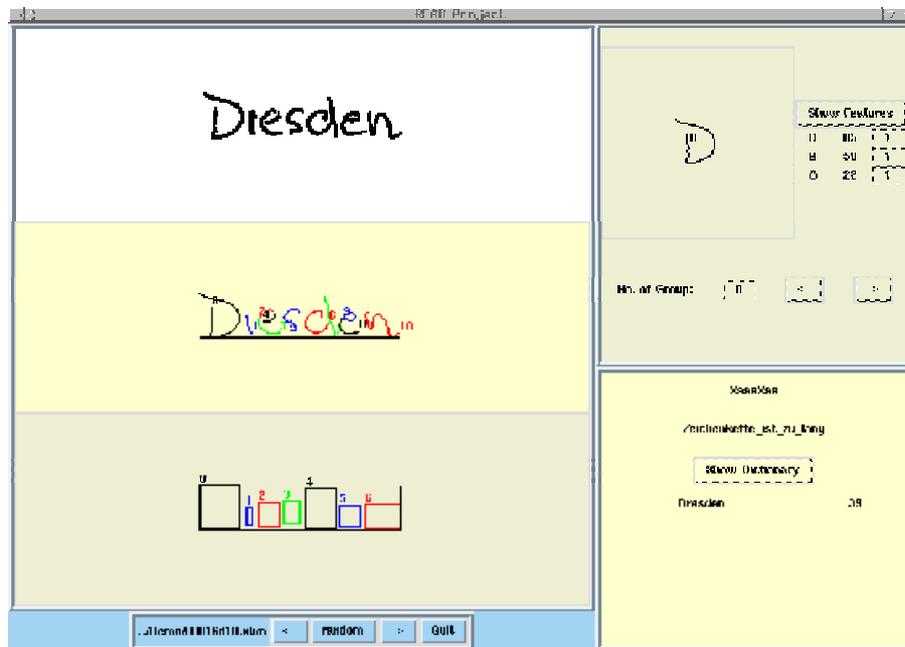


Abbildung 9: Die graphische Oberfläche des Worterkennungssystems des READ-Projektes von GMD-SET.

AP2250***Kombination von Ergebnissen*****AP4200*****Benchmarking***

Für die Lösung von Mustererkennungsaufgaben werden schon über viele Jahre hinweg im Hause Siemens Electrocom Klassifikationsmethoden verwendet [BUWA97, DOIM97, MILE97]. Je nach Aufgabenstellung kommen unterschiedlichste Methoden zum Einsatz.

Die Güte des Klassifikators ist abhängig vom Gültigkeitsbereich der Aussage. Die Ergebnisdaten für die Klassifikatoren anhand ihrer Herkunft oder Fehlerquelle kann in Klassen eingeteilt werden. Die Beschreibung dieser Klassen geschieht mit Hilfe von Merkmalen, z. B. enthält die Klasse "Herkunft" solche Merkmale wie Land, Sprachräume, Ausbildung, Alter. Dadurch wandelt sich die Auswertung von Klassifikatoren aus einem passiven in einen aktiven Ansatz. Nicht nur die Schwächen oder Stärken der einzelnen Klassifikatoren gegenüber unterschiedlichen Datentypen können festgestellt werden, sondern es kann die Güte der einzelnen Bearbeitungsstufen in der Erkennungskette, gemessen an den Merkmalen, bestimmt werden. Diese Güte kann in eine apriori Wahrscheinlichkeit umgewandelt werden.

Die Güte der einzelnen Erkennungsstufen wird nicht nur durch die ausgewählten Merkmale der Datenklassifikation bestimmt, sondern auch durch die gewählten *Meßpunkte*. Die berechnete apriori Wahrscheinlichkeit der Erkennungsrate einzelner Klassifikatoren wird von der *Datenklassifikation* und der *Bewertungsfunktion* bestimmt. Eine Voraussetzung für die zügige Bewertung der Daten ist die Vollständigkeit der *Ground Truth Table* für die Trainingsdaten .

Die Störungs- oder Fehlerquellen können in drei große Klassen eingeteilt werden:

- *Hintergrundstörungen*, die durch Stempel, Hintergrundmotiv, Farbe etc. verursacht werden.
- *Segmentierungsstörungen*, die durch verklebte Schrift, übereinander geschriebene, verwischte oder durchgestrichene Schrift verursacht werden.
- *Wortklassifikationsstörungen*, die durch falsche Wörter, orthographische Fehler etc. verursacht werden.

Anhand solcher Merkmale kann jedem Klassifikator ein apriori Gütewert anhand von Testdaten zugewiesen werden. Die Güte des Klassifikators K ist nicht ein gemittelter Wert über alle Fehlerklassen, sondern ein Vektor, der der jeweiligen Güte des Klassifikators für die vordefinierten Fehlerklassen entspricht. Wenn N die Anzahl der Klassifikatoren ist und jeder unterschiedliche oder teilweise unterschiedliche Ansatz die Implementierung dieser Eigenschaften verfolgt, kann eine Auswertung der Ergebnisse auch unter der Güte einzelner Eigenschaften betrachtet werden. Das bedeutet, daß das Kombinationsverfahren auf eine Matrix von $N \times M$ aufsetzt, wobei N der Anzahl der möglichen Klassifikatoren entspricht und M der Anzahl der definierten Fehlerklassen.

Zuerst werden die Testdaten in Fehlerklassen klassifiziert und erst in der zweiten Stufe wird die Güte der einzelnen Klassifikatoren zu den Fehlerklassen bestimmt. Anhand dieser Werte kann ein Vor-Ort Anpassen stattfinden. Das bedeutet, es wird der Klassifikator oder die Module eines Klassifikators gewählt, die am besten für die gefundenen Fehlerklassen geeignet sind. Dies wird erreicht, in dem für eine gegebene Stichprobe prozentual ermittelt wird, welche Fehlerklassen im Datensatz enthalten sind. Anhand dieser Werte kann die Bedeutung der einzelnen Gütewerte eines Klassifikators für die Stichprobe ermittelt werden.

$$G_j = \sum w_{ij} \cdot g_{ij}$$

wobei G_j die Güte des Klassifikators K_j für die ausgewählte Stichprobe ist, w_{ij} die Wichtung der Fehlerklassen in der Stichprobe und g_{ij} die vorberechnete Güte des Klassifikators K_j zur Fehlerklasse i ist.

Es wird der Klassifikator mit der besten Güte für die gegebene Stichprobe ausgewählt.

AP3200

Extraktion stark strukturierter Information

AP5300

Dokumentlesen

Wie vorher bereits gesagt, stellen die Arbeiten in AP 5300 eine inhaltliche Fortsetzung des AP 3200 dar. Daher wird im weiteren auf die Unterscheidung verzichtet.

Es wurde eine Verbesserung der Entwicklungsmethodik von Systemen zur Dokumentanalyse erreicht sowie prototypische Demonstratoren vorgelegt. Die wichtigsten Ergebnisse des Arbeitspaketes AP3200 sind in Deliverable D322 angeführt. Die begrenzten proprietären Systeme, die vor Beginn von Read bekannt waren, haben Gemeinsamkeiten, die zusammengefaßt und sozusagen kultiviert werden konnten.

Ein wesentlicher Informationsträger in Dokumenten ist ihre logische Struktur. Die Verbreitung und Verwendung von SGML und XML ist ein starkes Indiz dafür. Dokumente, die nur für den Menschen bestimmt sind, auf Papier oder in elektronischer Form, verdeutlichen ihre logische Struktur mit Hilfe von Stilelementen, wie der Auftrennung in Absätze, Einrückungen und Kapitelnumerierungen. (Diese Erscheinungsform logischer Struktur wird hier als starke Struktur bezeichnet. Die von ihr repräsentierte Information wird als stark strukturierte Information bezeichnet.) Zur automatisierten, höherwertigen Weiterverarbeitung muß die logische Struktur explizit sein, das heißt sie muß mit sogenanntem Markup formal eindeutig markiert sein. Das Potential automatisierter, höherwertiger Weiterverarbeitung ist erheblich und daher ist die Gewinnung von expliziter Struktur (Markup) aus der impliziten Struktur (starke Struktur) das Ziel.

Zur Gewinnung der expliziten aus der impliziten Struktur sind dreierlei Kenntnisse nötig:

- die Konzepte der starken Struktur (vorher),
- die Konzepte der expliziten Struktur (nachher) und
- das Vorgehen erstere in zweitere zu überführen.

Ein Beispiel ist die Kenntnis des Konzeptes Hauptkapitelüberschrift (der expliziten Struktur), das in Dokumenten jeweils eindeutig daran zu erkennen ist, daß es jeweils aus einer einzelnen Zeile besteht, beginnend mit einer einstelligen Ziffer mit Punkt.

Am Ende des Arbeitspaketes steht eine Software, die Konzepte von starker in explizite Struktur überführt. Einer solchen Software müssen die angegebenen Kenntnisse zugänglich gemacht werden. Dafür muß man zuerst einen Formalismus entwickeln, eine (formale) Sprache, die derartige Konzepte ausdrücken kann.

Es muß dann diese Software entwickelt werden, die den Formalismus und die enthaltenen Konzepte verwendet. Das Hauptinteresse besteht hier nicht in der erhaltenen Software. Es werden noch viele Mannjahre in Weiterentwicklungen und Varianten solcher Software gesteckt werden. Das Hauptaugenmerk liegt auf der Art und Weise, wie aus dem Formalismus die Software entworfen wird, eben die Entwicklungsmethodik von Systemen zur Dokumentanalyse.

Beschreibungsformalismus

Für die Beschreibung der Merkmale starker Struktur, expliziter Struktur und der Umwandlung ineinander waren nur Insel-lösungen bekannt, die jeweils einige wenige Aspekte berücksichtigten. Es war klar, daß bessere als diese Ansätze möglich waren, indem man einfach in einem Formalismus mehrere Aspekte integrierte. Es stellte sich heraus, daß im Prinzip eine Integration aller bis dato angedachten Aspekte möglich ist.

Der Formalismus verändert dadurch aber seine Bedeutung für die konkrete Umsetzung in der angestrebten Software. Durch die Kombination der verschiedenen Aspekte entsteht ein (viel) ausdrucks-mächtigerer Formalismus. Die höhere Ausdrucksmächtigkeit erhöht den Aufwand für die Verarbeitung (in diesem Falle ganz erheblich), so daß eine direkte Umsetzung wie vorher gar nicht mehr ohne weiteres möglich ist.

Das verdeutlicht sich am eindeutigsten am Beispiel. Ein bekannter Formalismus ist die Grammatik. Mit einer Grammatik kann man z.B. die Syntax von Programmiersprachen festlegen. Danach kann man mit ihr auch unmittelbar Software generieren, die Programme in der festgelegten Sprache auf ihre Korrektheit prüfen. Die in der Grammatik festgelegten Kenntnisse sind also unmittelbar für die Verarbeitung geeignet. Dafür wird entsprechend der Struktur der Grammatik Symbol für Symbol vorgegangen. Grammatiken werden auch in dem System von GMD IPSI als Formalismus verwendet.

Ein wesentliches Moment mancher anderer Formalismen sind die Layoutattribute. Sie können an verschiedenen Stellen geometrische, d.h. zweidimensionale Informationen enthalten. Jetzt soll beides integriert werden. Wenn man eine Grammatikrepräsentation hat und die Symbole der Grammatik mit Layoutattributen erweitert, dann überlagert sich der für die Verarbeitung wesentliche Grammatik-Struktur eine weitere Struktur (die sogar alleine schon sehr kompliziert, weil zweidimensional, ist) von Verknüpfungen zwischen den Symbolen. Die konventionelle Verarbeitungsstrategie ist daher nicht mehr anwendbar.

Ontologien

Dieses Phänomen ist nicht spezifisch für die Dokumentanalyse und daher hat sich die Wissenschaft seiner in letzter Zeit angenommen. Nach der dort entwickelten Terminologie ist der angestrebte Formalismus eine Ontologie. Das Forschungsgebiet der Ontologien beschäftigt sich mit Formalismen, die Wissen repräsentieren, und die -- meistens wegen grosser Ausdrucksmächtigkeit-- nicht mehr allgemein mit den Methoden der konventionellen Wissensrepräsentation behandelt werden können.

Man könnte vereinfachend folgendermaßen sagen: in der konventionellen Wissensrepräsentation bestimmt die Wissensmethodik die erzielbare Funktionalität. Zugunsten der Methodik muß fast immer darauf verzichtet werden, das verfügbare Wissen vollständig zu verwenden. Die ontologische Sichtweise entspricht der umgekehrten Vorgehensweise, die bereits oben als die erwünschte formuliert wurde (nämlich die Integration der bekannten Insellösungen). Das verfügbare Wissen soll zusammengestellt werden und die Methodik maßgeschneidert dazu entwickelt werden. Die Methodik ist dann kein alleinstehender Algorithmus im konventionellen Sinne. Statt dessen werden Stück für Stück Algorithmen entwickelt und der Software hinzugefügt, die jeweils weitere Wissensanteile verarbeiten können.

So kann man z.B. an ein einfaches Modul denken, welches das Datum erkennen kann. Von vornherein wird aber berücksichtigt, daß es jederzeit nötig werden kann, dieses Modul zu erweitern. Wenn man dann das Wissen über römische Zahlen in die Ontologie hinzufügt, kann das Modul ergänzt werden und römische Monatsangaben in Datum erkannt werden. Diese Vorgehensweise ist in der konventionellen Wissensrepräsentation in dieser Art unmöglich.

Anhand dieses Szenario sind noch einige Anmerkungen möglich. Es stellt sich heraus, daß der Aufwand Modularerweiterungen der von vornherein vorzusehen, nicht so groß ist, wie befürchtet. Im wesentlichen muß nur eine Stelle festgelegt werden, an der zur Analyse des Monats zusätzlich in das Ergänzungsmodul gesprungen wird. Diese Festlegung einer Stelle ist genauer betrachtet der Vorteil gegenüber der konventionellen Wissensverarbeitung. Dort muß nämlich ein zusätzliches Wissensatom prinzipiell in jedem einzelnen Verarbeitungsschritt wieder und wieder berücksichtigt werden. Ein weiterer Vorteil dieses Ansatzes ist, daß die Zwischenergebnisse, bevor in ein Erweiterungsmodul gesprungen wird und die danach, leicht zugänglich sind und daher die Auswirkungen der Erweiterung relativ leicht überprüfbar sind. Daraus ergibt sich noch eine zusätzliche Erweiterungsmöglichkeit, nämlich Meta-Module, die kritische Eigenschaften von Zwischenergebnissen überprüfen.

Initiale Anforderungen an die Beschreibungssprache

1. Meistens wird die von einem Segmentierer gelieferte Blockstruktur gescannter Dokumente als die Layoutstruktur angesehen. Auf deren Granularität wird dann die logische Struktur bestimmt. Dafür werden einzelne, je nach System unterschiedliche Layout-Merkmale der Segmente verwendet, wie z.B. Eigenschaften des Font.
2. Der Read vorhergehende Ansatz der GMD erlaubte auch die Verwendung kleinerer Strukturen. Z.B. kann eine Titelzeile nicht nur

aufgrund des sie umfassenden Segmentes erkannt werden, sondern auch an der Ziffer am Anfang der Zeile. Das erhöht die Zuverlässigkeit der Ergebnisse und macht in manchen Fällen die Analyse überhaupt erst möglich. Diese Eigenschaft soll erhalten bleiben.

3. Allerdings wird die Analyse der Layout-Merkmale nicht ausgeklammert, denn sie kann in vielen Fällen viel Information liefern. Die Kombination von so verschiedenen Merkmalen ist schwierig.

4. Wenn man sich an Standards zur Repräsentation logisch strukturierter Dokumente orientiert, kann man auf eine Infrastruktur zurückgreifen, die es stark vereinfachen sollte, Wissensbasen anzulegen. Zum einen kann man sich an vorhandenen Dokumenten orientieren, zum anderen sind theoretische Kenntnisse über die Dokumente übertragbar (Komplexitäten etc.).

5. Für die meisten analysierten Dokumente muß davon ausgegangen werden, daß sie Fehler aus den vorhergehenden Prozessen enthalten. Daher müssen die Interpretationsmethoden fehlertolerant sein. Darüber hinaus soll es aber auch möglich sein, bestimmte Fehler konkret in der Modellierung vorzusehen und in die Modellierung mit einzubeziehen. Auch dieser Ansatz wird es erlauben, Dokumente zu interpretieren, die ansonsten zurückgewiesen werden müßten.

Ergebnisse der Betrachtungen der Formalismen der Partner

Den vorhergehenden Anforderungen kann man entsprechen, wenn man die Formalismen der vorhandenen Systeme integriert. Dazu kommt die Integration der erwähnten Standards: Zu Beginn von Read handelte es sich dabei ganz eindeutig (ausschließlich!) um SGML, das aber nunmehr durch den Nachfolger XML ersetzt worden ist.

Die Formalismen der Partner wurden untersucht, ob sie zu integrieren sind. Dabei stellte sich heraus, daß große Ähnlichkeiten zwar nicht auf unteren Ebenen der Funktionsweise, aber trotzdem grundsätzlich ähnliche Strukturen erkennbar waren.

1) Bei der Dokumentrepräsentation besteht eine grundsätzliche Einteilung in Begriffe des Dokument Layout und des Dokument Inhaltes.

2) Hinter dem Begriff des Dokument Inhaltes stehen Assoziationen mit weiteren Begriffen wie Dokument Logik, logische Struktur, Bedeutung, Interpretationen und weitere. Die Abstraktion der Informationsart in der Beschreibungssprache ermöglicht es mit diesen Begriffen zu arbeiten ohne sie für die Partner verbindlich zu definieren.

3) Der Begriff des Elementes bei der logischen Struktur wird ein Pendant in der Layout Struktur haben. Der vorerst gewählte Arbeitsbegriff ist der der (bounding) Box. Eine Box B ist charakterisiert durch eine Position und eine Ausdehnung. Diese beiden Eigenschaften müssen nicht absolut, also in einem Koordinationsystem, beschrieben oder beschreibbar sein, sondern sind in der allgemeinsten Form erlaubt.

4) Eine Box B kann weitere Eigenschaften haben, die sich in Attributen ausdrücken.

5) Zwischen Boxen B1 und B2 können Relationen angegeben werden. Jede Relation kann eine Bewertung zugeordnet haben. Auf Grundlage des Relationsbegriffes lassen sich beliebige Strukturen aufbauen; einen Sonderfall stellen dabei z.B. hierarchische Struktu-

ren wie die Struktur in SGML dar, denen die "enthält" oder "part-of" Relation zwischen den logischen Elemente zugrunde liegt.

6) Über mindestens einen Typ von Relation können (Layout) Boxen mit (Inhalt/Logik) Elementen verknüpft werden.

Mit einer formalen Sprache, die auf diesen Charakteristika aufbaut, können Aussagen formuliert werden, so daß sie in die spezielleren Formalismen der Systeme aufgeteilt und übersetzt werden können. Z.B. könnte man Aussagen über einen Typ von Paragraphen machen, wie Fonttypen, Pattern und seine Hintergrundfarbe. Diese Aussage kann man in drei Aussagen auftrennen, drei unterschiedliche Systeme wählen, die Fonts erkennen, Pattern matchen (so wie das System von GMD IPSI) und Farben verarbeiten können und sie in deren Formalismen übersetzen. Aus einer einheitlichen Wissensbasis kann man so drei sich ergänzende Systeme ansteuern. Damit ist eine erste Stufe der Integration erreicht, die trivial nachvollziehbar eine größtmögliche Kontinuität für die Arbeiten bei den Partnern bedeutet.

Für die Berücksichtigung der Fehlertoleranz gab es schon Ansätze in dem System von GMD IPSI. Dort war die Einführung der nötigen Vorkehrungen auf Grundlage des Formalismus möglich. Im Prinzip ist es möglich, das Hinzufügen von Fehlertoleranz einfach auf einen weiteren Integrationsschritt zurückzuführen.

Entwickelte Software Module

DS1 Parser: Mit der Fertigstellung des Formalismus, der kurz DS1 bezeichnet wird, wurde ein Parser zur Verfügung gestellt, mit dem Aussagen in DS1 geparkt und verarbeitet werden können. Dabei können Aussagen in DS1 nicht für die Analyse verwendet werden. Der Parser ist vielmehr Grundlage für Übersetzungswerkzeuge, mit denen man DS1-Wissen in den Formalismus eines Partnersystems übertragen kann, das man dort verwenden möchte. Dieses Modul wurde mit dem Unix-Standard Tool und der frei verfügbaren Library libg++ programmiert.

DSD-to-DS1 Compiler: Für Spezifikationen im DSD Formalismus des GMD IPSI Systems wurde ein Cross Compiler geschrieben, so daß die Wissensspezifikationen des Systems weiter verwendet werden können, indem sie nach DS1 übersetzt werden. Damit wurde implizit auch nachgewiesen, daß DS1 tatsächlich, wie behauptet, alle im DSD Formalismus möglichen Aussagen darstellen kann.

VRML Visualisierer: Es wurde ein Visualisierer geschrieben, der Grammatiken in das grafische Format von VRML Dateien transformiert. In dieser Form können sie in einen Standard VRML Viewer geladen und angesehen werden. Ein Visualisierer für Grammatiken ist interessant, weil man aus DS1 in unterschiedlicher Art Auszüge gewinnen kann, die die Mächtigkeit von Grammatiken haben. Z.B. die Transformation von DS1 in den DSD Formalismus entspricht einem solchen Auszug.

PerlDream Module:

In der Programmiersprache Perl wurde ein Modulsystem programmiert, das vielerlei Funktionalität in sich vereinigt. Dieses Framework spiegelt die Vorgehensweise zur Integration der Formalismen wider.

Es ermöglicht die weiter oben beschriebenen schrittweisen bzw. modulweisen Erweiterungen.

Es stellt eine graphische Oberfläche zur Verfügung. Dort kann man Input Dokumente, aus DS1 extrahierte Grammatiken und erkannte strukturierte Dokumente anzeigen lassen. Das heißt, einzelne Verarbeitungsschritte werden Stück für Stück sichtbar. Die Bedienung wird dadurch erheblich unterstützt.

Ein Parser arbeitet entsprechend der angezeigten Grammatik die Eingabe ab. Dabei erlaubt die Programmstruktur unter Ausnutzung der Programmiersprache Perl, den leichten Zugriff auf die Vorgehensweise. Im Prinzip ist es sofort möglich, die anfangs beschriebenen Erweiterungen einzubauen. Prototypisch ist das auch schon passiert.

Genau so einfach ist es, die zugrunde liegende Parsing Strategie zu ändern, also vom standardmäßigen "Top-Down" z.B. zu "Bottom-Up" zu schalten. Das ist aber noch nicht in Gänze demonstriert worden, nur in kleinen Vorstufen.

Es wurde für Perl auch ein Yacc Parser geschrieben, der das XDOC Format von Xerox einlesen und für graphische Textfenster der Perl-Oberfläche aufbereiten kann. XDOC ist ein hochwertiges Ausgabeformat, das von guter OCR Software ausgegeben werden kann.

Reale Daten

Auf einem Sample von 123 gescannten Geschäftsbriefen wurde nachvollzogen, auf welche Art und Weise man mit Pattern matches das Briefdatum am besten bestimmen kann.

Es stand dabei im Vordergrund Vorgehensweisen zu finden, die einleuchtend sinnvoll sind, und die mit dem Framework in Perl gut programmierbar, mit konventionellen Parsertools schlecht oder gar nicht programmierbar waren.

Die OCR Voraufbereitung der Tiff Dateien geschah mit Scanworx. Als Arbeitsformat wurde „ASCII spaces“ gewählt (weil das für Pattern Matches ausreicht). Mit Unix-grep und kleinen Perlscripten wurde dann versucht, charakteristische Pattern herauszuarbeiten, beispielsweise „Sehr geehrte“, „Strasse“, etc.. Verwechslungen zu vermeiden ist dabei kein triviales Programm, insbesondere da Briefe in seltenen Fällen gar kein Datum enthalten, häufiger mehr als ein Datum. Auf diese Weise stößt man als ein Beispiel auf das menschliche Vorgehen, Gegenproben zu machen. Diese Vorgehensweise ist für die Parserprogrammierung sehr interessant und bisher nirgends vorgesehen. PerlDream hat diese Funktion jetzt fest implementiert.

3.2 Nutzen und Verwertbarkeit der Ergebnisse

AP2150 *Lernfähige Algorithmen***AP2210 *Anwendung von Fuzzy Techniken*****AP2250 *Kombination von Ergebnissen*****AP4200 *Benchmarking***

Die vorgeschlagene Methodologie zur Klassifizierung von Zeichen, Wörtern oder Bildern kann sowohl für „Black-Box“ Klassifikatoren – wo die Methodik der Erkennung nicht genau oder sehr verschieden voneinander ist - eingesetzt werden, als auch für die Optimierung Vor-Ort eines einzelnen Klassifikators.

Im Falle mehrerer Klassifikatoren erfolgt der Einsatz wie beschrieben im AP 2250 Kombination von Ergebnissen. Im Falle eines Vor-Ort-Lernens eines Klassifikators besteht der Vorteil der Methodik durch die getrennte Entscheidung für die jeweilige Fehlerklasse gegenüber einem Fehlerreduktionsansatzes, in dem die Gesamtgüte eines Systems erhöht wird ohne genau die Ursache – in unserem Fall die Stichprobe – zu analysieren. So zum Beispiel kann die Methode eingesetzt werden um zwischen verschiedenen Fuzzy Regelbasen für verschiedene Datentypen (Fehlerklassen) zu alternieren. Einsatzort wäre der Fuzzy Clustering Klassifikator, vorgeschlagen im AP 2210/AP2150. Ein anderes Beispiel ist die Optimierung der Vor-Ort-Anpassung der Siemens ElectroCom Ansätze. So zum Beispiel kann man in der vorgeschlagenen direkten Methode schneller zu einem Gesamtergebnis kommen, wenn schon apriori die verschiedenen Momentmatrizen für die einzelnen Fehlerklassen bekannt sind. Durch die Ermittlung der Zugehörigkeit der Stichproben zu den Fehlerklassen kann schon in der ersten Iterationsstufe ein „grob“ angepaßter Klassifikator entstehen. Eventuelle Abweichungen werden über die Gesamtgüte des Klassifikators als Erkennungsrate „fein“ (Iterations-schritte) angepaßt.

Das Verfahren wurde als Patent eingereicht.

AP3200 *Extraktion stark strukturierter Information***AP5300 *Dokumentlesen***

Die Ergebnisse sind auf drei Ebenen verwertbar. Zum einen können direkt mit der erstellten Software im entsprechenden Zielbereich Analysatoren zusammengestellt und konfiguriert werden. Zum zweiten können die Projektpartner, andere Inhaber von ähnlicher Dokument Analyse Software oder auch Programmierer neuer Software „from scratch“ die Richtlinien zur Strukturierung solcher Module für ihr Design verwenden. Das erleichtert das Design und vereinfacht gleichzeitig spätere Änderungen an der Software sowie die tägliche Arbeit mit ihr. Zum dritten kann die Vorgehensweise auf andere nahe

verwandte Gebiete übertragen werden. Allerdings sind sie z.B. auf dem Gebiet der Extraktion schwach strukturierter Information, das außerhalb der Dokumentanalyse als Computerlinguistik bezeichnet wird, schon länger bekannt. Die Übertragung sollte also in Richtung der niederen Verarbeitungsschichten gehen. Die Verwertbarkeit der Ergebnisse ist daher tatsächlich sehr weitreichend.

Bei GMD IPSI zielt die erstellte Framework Applikation in Richtung Anwendungen auf den Gebieten Digital Libraries, E-Commerce und intelligente Webtechnologien. Dort sind die zu analysierenden Dokumente oft nicht ursprünglich auf Papier, haben aber viele verwandte Eigenschaften. Im allgemeinen erhofft sich GMD IPSI eine stärkere Konkurrenzfähigkeit mit den Entwicklungen, die im nächsten Kapitel beschrieben sind.

3.3 Fortschritt auf diesem Gebiet bei Dritten

AP2150 ***Lernfähige Algorithmen***

AP2210 ***Anwendung von Fuzzy Techniken***

AP2250 ***Kombination von Ergebnissen***

AP4200 ***Benchmarking***

Für die Problematik der Schrifterkennung werden verschiedene Klassifikatoren verwendet, die einzelne Schritte der Erkennung mit unterschiedlicher Güte bewältigen. In diesem Zusammenhang können verschiedene Methoden benutzt werden, so z. B. der Fuzzy Logik Ansatz von Gader et al [GMK96], der die Fuzzy Integrationsmethode benutzt, um die Eigenschaften der Klassifikatoren zu erfassen und anhand dessen später diese zu kombinieren. Die verwendete Choquet Integrals und Sugeno Integrals waren erheblich besser als Neuronale Netze. In [LUYA97] haben Lu und Yamaoka eine Fuzzy Reasoning basierte Kombinationsmethodik bevorzugt, indem sie die Gütemerkmale mit possibilistischem Ranking an verschiedene Kategorien zugewiesen haben. Andere Kombinationsmethoden basieren auf ein stochastisches lernbasiertes Voting Verfahren. Es werden z. B. stochastische Kombinationen oder auf Neuronalen Netzen basierende Kombinationen vorgeschlagen [SCHU96]. Die Güte der einzelnen Klassifikatoren – bezogen auf die Erkennungsrate – wird in die Bewertungsfunktion mit einbezogen, und anhand von Wichtungen wird die beste Lösung gefunden.

AP3200 ***Extraktion stark strukturierter Information***

AP5300 ***Dokumentlesen***

Auf dem ureigenen beschriebenen Gebiet sind keine Fortschritte bei Dritten bekannt geworden, außer bei den Projektpartnern. Es gibt aber einige Entwicklungen, die man hier aufführen kann.

Auf dem Gebiet der Ontologies und dem Spezialgebiet der "Problem Solving Methods" (PSM) wurde, vor allem im Rahmen des Esprit-

Projektes CommonKADS, die Designmethodologie zur Entwicklung wissensbasierter Systeme stark vorangetrieben. Viele der dortigen Arbeiten sollten sich auf die Dokument Analyse übertragen lassen, wenn man sie nur als eine spezielle wissensbasierte Anwendung ansieht.

Das Gebiet der semi-strukturierten Daten beschäftigt sich mit Daten, die formal etwas schwächer oder unzuverlässiger sind als z.B. Daten in einer Datenbank. Eine typische Erscheinungsform sind Form-Daten auf Web-pages. Im allgemeinen ist XML weit für die Repräsentation solcher Daten verbreitet. Dieses Gebiet entwickelt sich stark, da das Internet die lokale Datenbank immer mehr als Informationsquelle verdrängt und diese Entwicklung bessere Verarbeitungsmodelle für Websuchmaschinen und andere intelligente Agenten-Module benötigt.

Weltweit sind starke Aktivitäten der großen Bibliotheken zu sehen, die in sehr großen Anstrengungen die Vorzüge der neuen Technologien zu nutzen suchen. Der unmittelbar naheliegendste Schritt, das vorhandene Material elektronisch ins Netz zu stellen, bedarf dokumentanalytischer Software. Die Entwicklung ist dort allerdings noch nicht sehr fortgeschritten.

Im Bereich der Computerlinguistik hat seit Beginn dieses Jahrzehntes eine starke Weiterentwicklung der Wissensspeichertechnik und der Parsertechnologie stattgefunden. Die Bedingungen in der Dokumentanalyse sind leicht variiert, so daß die Übertragung nicht trivial ist. Wie oben erwähnt, haben aber die Linguisten früher als die Dokumentanalyse gelernt Ontologies zu verwenden. Vielleicht gibt es noch anderes zu lernen.

Literaturhinweis

[BUWA97] H. Bunke, P.S.P. Wang: "Character Recognition and Document Image", World Scientific, London, 1997.

[DOIM97] A. C. Downton, S. Impedovo: "Progress in Handwriting Recognition", World Scientific, London, 1997.

[GMK96] P.D. Gader, M.A. Mohamed und J.M. Keller, "Fusion of Handwriting Word Classifiers," Pattern Recognition Letters 17, pp. 577-584, 1996.

[GUSU94] D. Guillevic und C.Y. Suen, „Cursive Script Recognition: A Sentence Long Recognition Scheme“, Proceedings of 4th IWFHR, 1994.

[IVMP98] F. Ivancic, A. Malaviya und L. Peters, „An Automatic Rule Base Generation Method for Fuzzy Pattern Recognition with Multi-phased Clustering“, in Proceedings of 2nd International Conference on Knowledge-Based Intelligent Electronic Systems, Vol. 3, pp 66-75, Adelaide, 1998.

[KKMS97] F. Kimura, N. Kayahara, Y. Miyake und M. Shridhar, „Machine and Human Recognition of Segmented Characters from Handwritten Words“, in Proceedings of 4th International Conference on Document Analysis and Recognition, Ulm, 1997.

- [LEBA94] E. Lecolinet und O. Baret, „Cursive Word Recognition: Methods and Strategies“, in *Fundamentals in Handwriting Recognition*, S. Impedovo (ed.), NATO ASI Series F: Computer and System Sciences, Vol. 124, Springer-Verlag.
- [LESR93] D. Lee und S.N. Srihari, „Handprinted Digit Recognition: A Comparison of Algorithms“, *Proceedings of 3rd IWFHR*, 1993.
- [LEJA96] C. Leja, „Werkzeuge für die automatische Generierung von FOHDEL Regelbasen“, GMD-Studien Nr. 292, Sankt Augustin, 1996.
- [LUYA97] Yi Lu und Fumiaki Yamaoka, „*Fuzzy Integration of Classification Results*“, *Pattern Recognition*, pp.1877-1891, 1997.
- [MALA96] A. Malaviya, „On-line Handwriting Recognition with a Fuzzy Feature Description Language“, GMD-Bericht Nr. 271, 1996.
- [MIBP99] A. Malaviya, F. Ivancic, J. Balasubramaniam und L. Peters, „Off-Line Handwriting Recognition with Context Dependent Fuzzy Rules“, in *Knowledge-Based Intelligent Techniques in Character Recognition*, BN. Lazzerine und L.C. Jain (eds.), CRC-Press, 1999.
- [MALP96] A. Malaviya, C. Leja und L. Peters, „Multi-Script Handwriting Recognition with FOHDEL“, *Proc. of NAFIPS'96*, IEEE Press, pp. 147-151, Berkeley, 1996.
- [MAPE95] A. Malaviya und L. Peters, „Extracting Meaningful Handwriting Features with Fuzzy Aggregation Method“, *3rd ICDAR*, pp. 841-844, Montreal, 1995.
- [MAPE97-2] A. Malaviya und L. Peters, „Multi-Layered Handwriting Recognition Approach“, Accepted for Publication in *Fuzzy Sets and Systems*, 1997.
- [MAPT94] A. Malaviya, L. Peters und M. Theissinger, „FOHDEL: A New Fuzzy Language for On-Line Handwriting Recognition“, *FUZZ_IEEE*, Seiten 624-629, Orlando, 1994.
- [MILE97] U. Miletzki: „*Character Recognition in Practice Today and Tomorrow*“, In: „*Proceedings of Fourth International Conference on Document Analysis and Recognition*“, Computer Society, Brussels, 1997, pp. 902-907.
- [OGKA97] L. O’Gorman und R. Kasturi, „Document Image Analysis“, IEEE Computer Society Executive Briefing, Los Alamitos, 1997.
- [ORPE94] M.W. Oram und D.I. Perret, „Modelling Visual Recognition from Neuro-Biological Constraints“, *Neural Networks*, Vol. 7, No. 6-7, pp. 945-972, 1994.
- [PEGI99] L. Peters, J. Ghoshal und F. Ivancic, „Character Segmentation for Cursive Script Recognition“, *Pattern Recognition Journal*, 1999 (submitted) .
- [SCHU96] J. Schürmann, „*Pattern Classification – a Unified View of Statistical and Neural Approaches*“, John Wiley Sons, N.Y.,1996.
- [SENI94] A.W. Senior, „Off-line cursive handwriting recognition using recurrent neural networks“, Ph.D. Thesis, Eng. Dept., Cambridge University, 1994.
- [SPIT95] A.L. Spitz, „An OCR based on character shape codes and lexical information“, *Proceedings of 3rd ICDAR*, Montreal, 1995.

[SPIT97] A.L. Spitz, „Moby Dick meets GEORC: Lexical Considerations in Word Recognition“, Proceedings of 4th ICDAR, Ulm, 1997.

Anhang 1: Veröffentlichungen & Patente

Publikationen

- [1] Bertin Klein and Andreas Abecker: **Distributed Knowledge-based Parsing for Document Analysis and Understanding**, IEEE ADL 1999 (submitted).
- [2] A. Malaviya, L. Peters: **Handwriting Recognition with Fuzzy Linguistic Rules**, EUFIT'95, MIT Aachen, 28.-31. August 1995.
- [3] A. Malaviya, L. Peters: **Extracting Meaningful Handwriting Features with Fuzzy Aggregation Method**, ICDAR'95, 3rd International Conference on Document Analysis and Recognition, Montréal, August 14-16, 1995.
- [4] L. Peters, A. Malaviya: **Fuzzy Rule Based Handwriting Recognition**, KI95, Bielefeld, 11.9-15.9.95.
- [5] A. Malaviya, Ch. Leja, L. Peters: **Multi Script Handwriting Recognition with FOHDEL**, Proc. of 1996 Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS'96), June 19-22, 1996, Berkeley, CA., IEEE Press, pp. 147-151.
- [6] A. Malaviya, Ch. Leja, L. Peters: **A hybrid approach of automatic rule generation for handwriting recognition**, Pre-proceedings of the Fifth International Workshop on Handwriting Recognition (IWFHR5), Essex, 1996, pp. 405-408.
- [7] A. Malaviya, L. Peters: **Fuzzy Feature Description of Handwriting Patterns**, Pattern Recognition Journal, Volume 30, Number 10, 1997, pp 1591-1604, 1997.
- [8] A. Malaviya, Ch. Leja, L. Peters: **A Hybrid Approach of Automatic Fuzzy Rule Generation For Handwriting Recognition**, Progress in Handwriting Recognition, World Scientific Publishing, Singapore, pp. 413 - 418, 1997.
- [9] L. Peters, A. Malaviya: **A fuzzy statistical rule generation method for handwriting recognition**, Expert Systems, February 1998, Vol. 15., No. 1, pp. 48-56.
- [10] F. Ivancic, A. Malaviya, L. Peters: **An Automatic Rule Base Generation Method for Fuzzy Pattern Recognition with Multi-phased Clustering**, Proc. of the International Conference on Knowledge-Based Intelligent Electronic Systems“ (KES98), Adelaide, 18.04-25.05.98.
- [11] A. Malaviya, L. Peters: **Multi-layered handwriting recognition approach**, Journal Fuzzy Sets and Systems, (accepted for publication).

- [12] A. Malaviya, L. Peters: **Fuzzy Handwriting Description Language**, Pattern Recognition Journal, (accepted for publication).
- [13] A. Malaviya, F. Ivancic, J. Balasubramaniam, L. Peters: **Off-line Handwriting Recognition with Context Dependent Fuzzy Rules**, in Knowledge-Based Intelligent Techniques in Character Recognition, Eds. B. Lazzerini, L.C. Jain, CRC Press (to appear).
- [14] L. Peters, J. Ghoshal, F. Ivancic: **Character Segmentation for Off-line Cursive Skript Recognition**, Pattern Recognition Journal (submitted for publication).

Diplomarbeiten und Dissertationen

- [15] Ashutosh Malaviya: On-line Handwriting Recognition with a Fuzzy Feature Description Language, Dissertation, TU Berlin, 1996.
- [16] Franjo Ivancic: Lernen von Regelmengen zur Fuzzy-Mustererkennung durch mehrphasige Clusteranalyse, Diplomarbeit, Universität Bonn, 1999.

Ausstellungen und Präsentationen

- Innovationsmesse Leipzig 1997
- Schloßtag St. Augustin 1997
- Tag der offenen Türen, St. Augustin, November 1997

Patente

- [1] A. Malaviya, L. Peters, F. Ivancic: **Verfahren zur automatisierten Regelgenerierung für die Klassifizierung von Bilddaten** (submitted April 1998),