

Kapitel 8

Zusammenfassung

In den vorangegangenen Kapiteln wurden verschiedene Aspekte zur Online-Handschrifterkennung beleuchtet, mit denen die gesamte Verarbeitungskette von der Vorverarbeitung, Merkmalsextraktion, Modellierung bis hin zur Erkennung erfasst wird. Neben der reinen Texterkennung konnten weiterhin Ansätze für die Erkennung handgeschriebener Formeln präsentiert werden, die zusammen mit diversen Editierfunktionen in einen interaktiven Demonstrator integriert werden konnten.

Den Schwerpunkt in Kapitel 2 bildete die Vorverarbeitung der digitalisierten Stiftrajektorie mit der Neuabtastung und der Normalisierung. Insbesondere konnten hier Histogrammverfahren für die Online-Handschrifterkennung adaptiert werden, mit denen sich sowohl die Zeilenneigung, wie auch die Schriftneigung automatisch korrigieren lassen. Durch die Minimierung eines Entropiemaßes konnte speziell für die als problematisch geltende Schriftneigungskorrektur eine Alternative zu den bisher verwendeten Richtungshistogrammen erarbeitet werden. Die iterative Bereichsdetektion für die Schriftgrößenskalierung komplementiert die für eine vollständige Normalisierung notwendigen Funktionen.

Kapitel 3 stellt eine Auswahl der für die Online-Handschrifterkennung untersuchten Merkmale dar. Hierbei wurden insbesondere zwei Konzepte zur Merkmalsentnahme intensiver untersucht. Dies sind zum einen die Trajektorienmerkmale und zum anderen die Bitmap-Merkmale. Beide Ansätze machen von der dynamischen Information der Stiftrajektorie Gebrauch, wobei die Bitmap-Merkmale in zeitlich globaler Weise alle für einen geometrisch begrenzten Bereich relevanten Ereignisse berücksichtigen. Insgesamt wurden zahlreiche Trajektorien- und Bitmap-Merkmale untersucht, von denen eine Auswahl von 5 Trajektorien- und 3 Bitmap-Merkmalen näher präsentiert wurde. Über die Betrachtung von Einzelmerkmalen hinaus wurde eine optimale Merkmalskombination gefunden. Aufgrund der erzielten Erkennungsleistungen sind dabei die Trajektorienapproximation durch Bézierkurven in Kombination mit der DCT-Kodierung der gleitenden Bitmap besonders hervorzuheben. Beide Verfahren waren aus der Literatur bisher nicht bekannt. Speziell mit

der Singulärwertzerlegung konnte für die Trajektorienapproximation durch Bézierkurven ein sehr effizienter und genauer Algorithmus gefunden werden.

Unter der Berücksichtigung der dynamischen Eigenschaften der online abgetasteten Handschrift bietet ein Hidden Markov Modell basierter Ansatz besondere Vorteile. In Kapitel 4 wurden erstmals für die Online-Handschrifterkennung verschiedene HMM-basierte Modellierungsverfahren eingehender untersucht. Dazu wurden diskrete, kontinuierliche und hybride Ansätze anhand schreiberabhängiger Systeme mit sehr großem Vokabular miteinander verglichen. Als überraschendes Ergebnis konnte der erstmalig für die Handschrifterkennung eingesetzte hybride Ansatz als beste Modellierungsform identifiziert werden. Bei dieser hybriden Modellierungsform werden neuronale Vektorquantisierer für die Merkmalsquantisierung eingesetzt. Als Bewertungsfunktion für das Training der VQ wird die Transinformation maximiert, die zwischen Sender (Schreiber) und Empfänger (Erkenner) über die Verarbeitungskette übertragen wird. Durch eine Verallgemeinerung dieses Ansatzes ist es möglich, mehrere VQ anhand der Verbund-Transinformation simultan zu optimieren. Die Verbund-Transinformation konnte darüber hinaus auch für die Vorauswahl von Merkmalskombinationen genutzt werden, da hierbei die Korrelation der Einzelmerkmale einfließt.

Einen weiteren Modellierungsaspekt stellt die Einbeziehung von Kontextinformation dar (Kapitel 5), die im Bereich der Handschrifterkennung bisher weitgehend unberücksichtigt blieb. Im Rahmen dieser Arbeit wurden zum ersten Mal umfassende Untersuchungen zur Nutzung von Kontextinformation für die automatische Handschrifterkennung durchgeführt. Der Vorteil kontextbezogener Graphemmodelle liegt klar in der höheren Auflösung der Modelle. Kontraproduktiv wirkt sich auf die Trainierbarkeit der Modelle hingegen die z. T. dramatisch ansteigende Parameteranzahl aus. Im Gegenzug dazu sind diverse Clustering-Verfahren für HMM bekannt, mit denen sich die Parameteranzahl durch Verknüpfungen reduzieren läßt. Zur Adaption dieser Verfahren auf die Handschrifterkennung war es in erster Linie erforderlich Wissen über Kontexteinflüsse zu generieren. Dazu wurden in Kapitel 5 verschiedene leistungsfähige Ansätze entwickelt, deren Eignung anhand schreiberabhängiger Systeme mit Lexika von bis zu 200000 Wörtern getestet wurden.

Ein Erkennungssystem, bei dem die Eingabe verbundener Handschrift zugelassen wird, ist im allgemeinen auf ein vorgegebenes Lexikon angewiesen. Zur Abschwächung dieser Beschränkungen wurden die Forschungsarbeiten von den Bestrebungen begleitet eine maximale Lexikongröße zu realisieren. Wie in Kapitel 6 gezeigt wird, kann alternativ zu einem vom Umfang begrenzten, fest vorgegebenen Lexikon auch eine statistische Beschreibung von Buchstabenabhängigkeiten herangezogen werden. Der Vorteil ist, dass eine deterministische Definition gültiger Buchstabensequenzen mittels eines Lexikons damit überflüssig wird. Die durchgeführten Experimente belegen, dass sich eine brauchbare Genauigkeit jedoch erst mit einem relativ großen Kontext von mehr als 6 vorausgehenden

Zeichen einstellt. Solche Kontexttiefen sind bisher nicht berücksichtigt worden, da sie mit gebräuchlichen Dekodierverfahren nicht mehr beherrschbar sind. Als sinnvolle Alternative hat sich hier die Verwendung eines Stack-Decoders erwiesen.

Wird in einem System die Text- oder Befehlseingabe über einen Stift vorgesehen, erschließen sich weitere interessante Szenarien, wie die handschriftliche Eingabe von Formeln zur Dokumenterstellung oder auch für handschriftliche Taschenrechner. Methoden für die Realisierung dieses Szenarios wurden in Kapitel 7 entwickelt. Das Erkennungsergebnis wird dabei in einer \LaTeX -Struktur ausgegeben. Darüber hinaus wurden erstmals Konzepte für eine automatische Erkennung und anschließende Ausführung handschriftlicher Korrekturbefehle für Formeln präsentiert.

Demonstrator

Die Evaluierung verschiedener Methoden anhand konkreter Erkennungsergebnisse geschieht in der Regel in Form einer Stapelverarbeitung, bei der eine möglichst aussagefähige Menge von Schriftproben verarbeitet, und die Erkennerausgaben abschließend mit den 'Solltexten' verglichen werden. Für die Präsentation der verschiedenen Erkennungssysteme hingegen wurden die entwickelten Methoden und Software-Tools in einen Demonstrator integriert. Abb. 8.1 zeigt die für X-Windows entwickelte Oberfläche des Systems [Kel95, Mes99].

Die über die serielle Schnittstelle von einem Digitalisiertablett abgetasteten Handschrift-daten werden während deren Erzeugung bereits im Canvas visualisiert. Nach einem automatischen oder auch manuellen Start der Erkennung werden die Rohdaten über eine Pipeline an ein weiteres Programm im Hintergrund gesendet, welches die eigentliche Erkennung ausführt. Aus Flexibilitätsgründen ist es natürlich auch möglich den Ablauf der Erkennung im Hintergrund in Form von Skripten zusammenzufassen. Die Ausgaben des jeweiligen Erkennungssystems werden an die Oberfläche zurückgesendet. Statusmeldungen, sowie der erkannte Text können somit durch die Oberfläche angezeigt werden (linkes unteres, bzw. linkes oberes Fenster des 'NeuroGraphen' in Abb. 8.1).

Im Falle der (dargestellten) Formelerkennung wird das Erkennungsergebnis nicht wie bei der Texterkennung im integrierten linken oberen Fenster des NeuroGraphen dargestellt. Zur Visualisierung der erkannten Struktur wird die \LaTeX -Ausgabe des Erkenners zunächst kompiliert und über einen DVI-Viewer extern dargestellt. Die Korrekturfunktionen für die Formelerkennung folgen dem gleichen Prinzip. Dazu werden die manuellen Korrekturen mittels eines Stiftes auf dem Canvas vorgenommen. Nach erneuter Erkennung und nach Ausführung der Korrekturen wird das externe Ausgabefenster, sowie bei Bedarf das korrigierte Schriftbild aktualisiert.

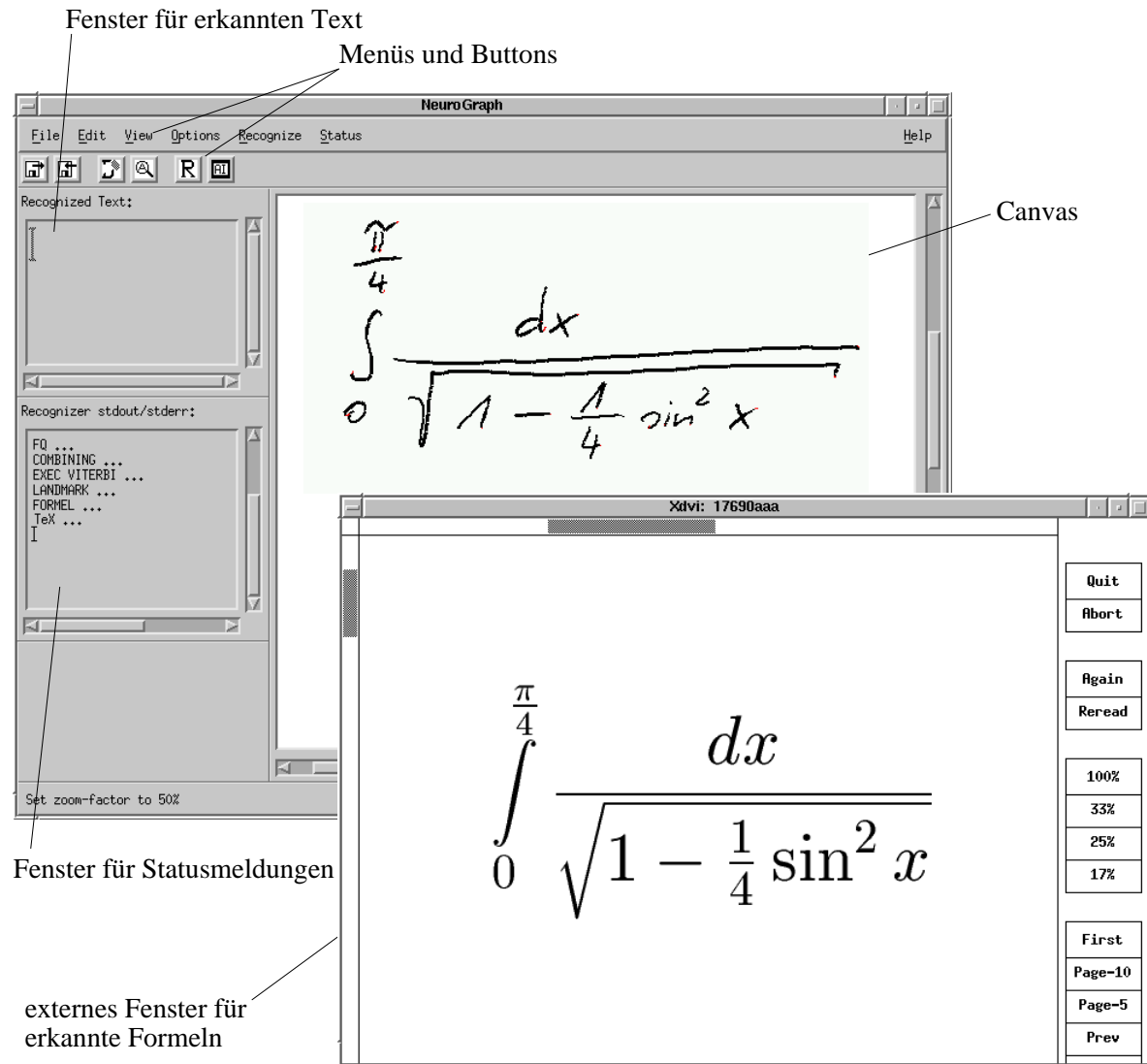


Abbildung 8.1: Oberfläche des Demonstrators