

# Stiftbasierte Suche in digitalen handschriftlichen Dokumenten

Sascha Schimke<sup>1</sup>, Claus Vielhauer<sup>1</sup>

<sup>1</sup>AG Multimedia and Security, Otto-von-Guericke Universität Magdeburg  
sschimke@iti.cs.uni-magdeburg.de

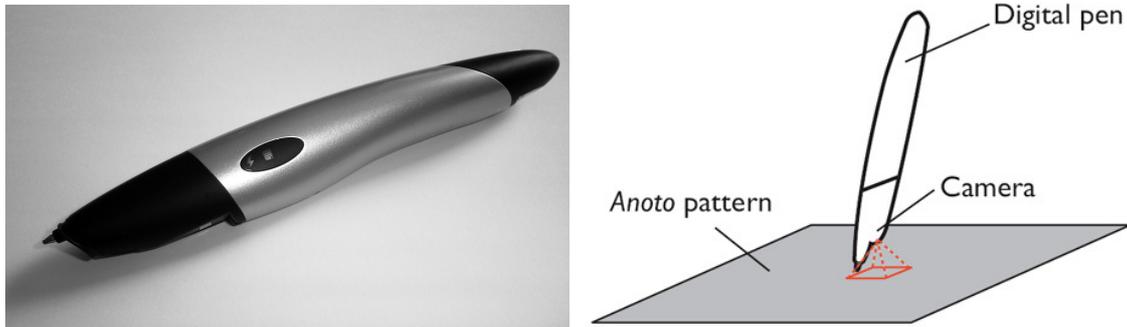
## Zusammenfassung

In den letzten Jahren ist eine zunehmende Verbreitung von stift-basierten mobilen Geräten zu beobachten, angefangen von klassischen PDAs, über Mobiltelefone mit Stifteingabemöglichkeit bis hin zu TabletPCs. Darüber hinaus kann erwartet werden, dass der Markt für spezielle autarke Stifte wächst, die in der Lage sind, die Schreibbewegungen zu erfassen und zu speichern [Anot05]. Durch die Verwendung all dieser Geräte ist zukünftig ein vermehrtes Anfallen von digitalen handschriftlichen Notizen sowie kürzeren und längeren Handschriftdokumenten zu erwarten. Um Archive solcher handschriftlichen Dokumente bequem verwalten zu können, ist neben anderen Voraussetzungen die Möglichkeit der automatischen Suche in diesen Dokumenten unabdingbar. In diesem Beitrag soll ein einfaches Verfahren beschrieben werden, welches eine solche Suche realisiert, ohne sich auf die klassische Schrifterkennung zu stützen. Obgleich mit aktuellen Handschrifterkennungsverfahren eine hohe Erkennungsgenauigkeit erreicht werden kann, scheitern diese Verfahren noch an schwer zu entziffernden, unsauberen Schriften, insbesondere auch an Schriften aus Kulturkreisen mit bislang nicht unterstützten Schriftsystemen, sowie an der Erkennung von nicht-schriftlichen Eingaben, wie beispielsweise Symbolen oder Skizzen. Das hier vorgestellte Verfahren zur Lösung dieser Art von Problemen wird im Laufe des Beitrags im Detail erläutert und schließlich mittels entsprechender Erkennungsraten aus dem Gebiet des *Document Retrieval* bewertet.

## 1 Einleitung

Nach einer langjährigen Vormachtstellung der Tastatur als primärer Eingabemodalität für technische Geräte, ist in jüngster Zeit eine zunehmende Verbreitung von stift-basierten Geräten, und damit ein neues Aufleben der Handschrift, im technischen Umfeld zu beobachten. Angefangen von den klassischen PDAs seit Beginn der 1990er Jahre, erlangt der TabletPC aktuell eine gewisse Bedeutung. Bei TabletPCs handelt es sich zumeist um Notebookcomputer, deren Benutzer mittels spezieller Stifte direkt auf dem Bildschirm schreiben oder zeichnen können, wobei Sensoren im Bildschirm und/oder im Stift die Schreibbewegungen digital erfassen und an das Betriebssystem weitergeben. Neben solchen Geräten, bei denen lediglich die Tastatur durch den Stift ersetzt bzw. um diesen ergänzt wurde, gibt es auch spezielle Geräte, die nur aus einem speziellen Stift bestehen, welcher in der Lage ist, die Stiftbewegungen während des Schreibvorgangs autark zu erfassen und zu speichern sowie später diese gespeicherten Schriftdaten an einen Computer zu übertragen [Anot05] (siehe Abbildung 1). Mit der fortschreitenden Verfügbarkeit dieser Art von Geräten ist zu erwarten, dass mehr und mehr Nutzer zumindest in bestimmten Situationen den Stift, an Stelle der Tastatur, für das Verfassen von Notizen oder anderen Dokumenten verwenden werden. Die dadurch anfallenden digitalen

handschriftlichen Dokumente müssen ähnlich klassischen „ASCII-Dokumenten“ verwaltbar sein, um effizient mit ihnen arbeiten zu können. An vorderster Stelle steht dabei die Notwendigkeit einer Suchfunktionalität, also die Möglichkeit des schnellen und verlässlichen automatisierten Auffindens von Handschriftdokumenten mit bestimmten Inhalten.



**Abb. 1:** Beispiel eines autarken Schriftdatenerfassungsgerätes sowie seine Funktionsweise [Anot05]

Der intuitive Ansatz, solch eine Suchfunktion für Handschriftdokumente umzusetzen, ist, die Dokumente zuerst mittels Handschrifterkennung in Text umzuwandeln und dann diese Textdaten zu durchsuchen [PeRZ02][OKON04]. Dieses Vorgehen hat den großen Vorteil, dass es bereits sehr robuste Verfahren zur Handschrifterkennung gibt – ein solches Suchsystem wäre also mit wenig Aufwand implementierbar. Die wichtigsten Nachteile dieses Ansatzes sind jedoch a) die noch immer schlechte Erkennungsqualität bei Personen mit unsauberer Handschrift, b) die fehlende oder bislang nur unzureichende Unterstützung von bestimmten Sprachen und Schriftsystemen c) die Unmöglichkeit, nach Stifteingaben zu suchen, die keinen Text darstellen, also beispielsweise Diagramme, Skizzen und dergleichen mehr.

Ein weiterer sinnvoller Ansatz ist es deswegen, eine Suchanfrage in Form einer Stifteingabe (Suchwort oder Suchsymbol) zu stellen, indem entweder ein einzelnes Vorkommen des Suchwortes in einem der Dokumente markiert wird oder indem das Suchwort erneut geschrieben wird.

In der Literatur finden sich einige wenige Verfahren, die eine solche Suche ohne Texterkennung realisieren. Zumeist basieren sie auf einer Segmentierung der Dokumente in einzelne Worte oder noch kleinere Einheiten und einem anschließenden Vergleich der Sucheingaben mit den Worten der Dokumente [LoTo94] [FrHu96] [JaNa03]. Dabei ist das korrekte Auffinden von gesuchten Begriffen abhängig von der Korrektheit der Segmentierung der Dokumente. Das in diesem Beitrag vorgestellte Verfahren verzichtet auf eine solche Segmentierung als Vorverarbeitung und versucht dadurch, eine mögliche Fehlerquelle zu beseitigen.

## 2 Rohdaten und Merkmalsextraktion

In der gängigen Klassifikation, wie sie z.B. auch in der Handschriftbasierten Biometrie [LeP194] vertreten wird, werden die Ausgangsdaten für die Verarbeitung von Handschrift in digitaler Form prinzipiell in zwei Arten unterteilt – Off-line- und On-line-Daten. Um **Off-line-Daten** handelt es sich, wenn das Schriftbild **im Anschluss an den Schreibvorgang** beispielsweise mit Hilfe eines Scanners digitalisiert wird. Diese Daten liegen zumeist in zweidimensionaler Form als Bilddatei vor. Suchverfahren für solche Off-line-Daten wurden bei-

spielsweise in [GoXu04], [MaHR96] und [SrHS04] beschrieben. Von **On-line-Daten** ist die Rede, wenn die Informationen über die Stiftbewegung **während des Schreibvorgangs** erfasst werden. On-line-Daten sind also zeitdiskret abgetastete Signale, die in den meisten Fällen die x-y-Position sowie den Stiftdruck während des Schreibvorgangs repräsentieren. In seltenen Fällen kann zusätzlich oder alternativ auch die Stiftgeschwindigkeit oder Stiftbeschleunigung erfasst werden.

Für die Suche in handschriftlichen Dokumenten, welche mittels TabletPCs oder der anderen in Abschnitt 1 erwähnten Geräte erfasst werden, ist die Analyse von On-line-Daten relevant, da bei diesen Geräten eine Datenaufzeichnung zum Zeitpunkt der Schrifterzeugung erfolgt. Für unsere Zwecke liegen die Daten als zeitdiskrete Signale  $x_t$ ,  $y_t$ ,  $p_t$  vor, wobei  $x_t$  und  $y_t$  die Stiftpositionen in hoher Auflösung sind (siehe Abschnitt 4) und  $p_t$  den Druck in binärer Form repräsentiert (Stift berührt/berührt nicht die Schreiboberfläche).

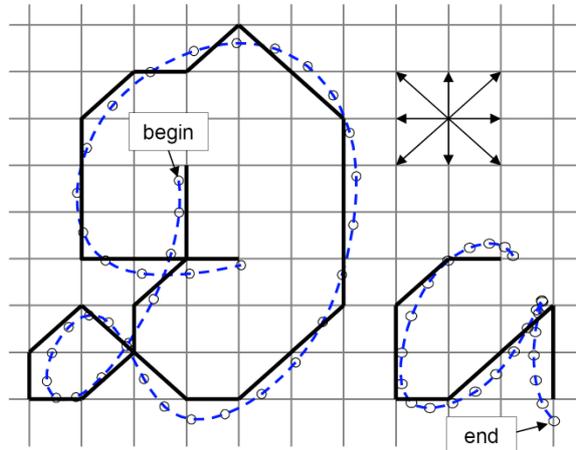
Die Grundidee der hier vorgestellten Suche in Handschriftdokumenten besteht darin, diese Dokumente (bzw. die entsprechenden Signale) in eine Stringform zu überführen (also eine Sequenz von Symbolen) und auf diesen Strings eine unscharfe Teilstringsuche nach dem entsprechenden Suchwort oder -symbol durchzuführen. Ein ähnliches Verfahren der unscharfen Stringvergleiche wurde durch die Autoren dieses Beitrags bereits erfolgreich für die biometrische Unterschriftenvergleiche eingesetzt [ScVD04].

Die Wahl geeigneter Strings zur Repräsentation der Ausgangssignale ist entscheidend für die Qualität der Suchfunktion. Zwei ähnlich aussehende handschriftliche Worte sollen möglichst ähnliche Strings besitzen, während unähnliche Handschrifteingaben auch unähnliche Strings haben sollen. Drei mögliche Merkmale und daraus abgeleitete Symbole bzw. Stringtypen sollen im Folgenden erläutert werden.

## 2.1 Gitterbasierte Richtungsmerkmale

Die Idee der gitterbasierten Richtungsmerkmale ist es, die handschriftliche Eingabe mit einem beispielsweise quadratischen Gitter zu überlagern und die geometrische Form der Eingabe anhand der Gitterknoten zu quantisieren. Da jeder Gitterknoten acht Nachbarknoten besitzt, kann die geometrische Form der Stifteingabe dann als Abfolge von Symbolen aus der Menge  $\{\uparrow \rightarrow \downarrow \leftarrow \nwarrow \nearrow\}$  dargestellt werden, wobei die Elemente dieser Menge für die Nachbarschaftsbeziehungen der Gitterknoten stehen. Eine Lücke im Verlauf der Stiftbewegung kann durch ein weiteres Symbol ausgedrückt werden. Abbildung 2 illustriert diese Quantisierung – die gestrichelte Linie entspricht der ursprünglichen Stifteingabe; die dickere durchgezogene Linie steht für die quantisierte Stiftbewegung. In diesem Beispiel würde die Sequenz folgendermaßen kodiert werden:  $\downarrow \downarrow \downarrow \downarrow \leftarrow \nearrow \rightarrow \rightarrow \nearrow \nearrow \uparrow \uparrow \uparrow \nwarrow \nwarrow \nwarrow \nwarrow \leftarrow \leftarrow \downarrow \downarrow \downarrow \rightarrow \rightarrow \rightarrow \leftarrow \leftarrow \downarrow \downarrow \rightarrow \nearrow \downarrow$ .

Es ist offensichtlich, dass die Gitterweite einen direkten Zusammenhang zur Länge des resultierenden Merkmalsstrings sowie zur Genauigkeit der Quantisierung hat. Je kleiner der Abstand zwischen benachbarten Gitterknoten, umso höher ist die Annäherung der quantisierten Daten an die Ausgangsform der Handschrifteingabe. Dieser Ansatz entspricht der Kodierung von Linienzeichnungen nach H. Freeman [Free74].



**Abb. 2:** Quantisierung mit Hilfe eines quadratischen Gitters

Neben quadratischen eignen sich auch Gitter, bestehend aus gleichseitigen Dreiecken oder Sechsecken. Die Anzahl von Nachbarknoten wäre in diesen Fällen sechs beziehungsweise drei. Ein Vorteil des gitterbasierten Ansatzes zur Strichrichtungsermittlung besteht darin, dass der Rechenaufwand für diese Merkmalsbestimmung vergleichsweise gering ist. Ein Nachteil ist die strenge Begrenzung der Anzahl möglicher Richtungen aufgrund der Nachbarschaften innerhalb geometrischer Gitter.

## 2.2 Gitterlose Richtungsmerkmale

Ein Nachteil der gitterbasierten Richtungsmerkmale, die feste Anzahl möglicher Richtungen, lässt sich unter Inkaufnahme von Rechenaufwand umgehen. Die Idee dabei ist, die Hand-schrifteingaben mittels kubischer Spline-Interpolation [Boor78] so umzuparametrisieren (Resampling), dass die ursprüngliche Kurve der Stifteingabe durch eine Folge von äquidistanten Punkten ausgedrückt wird. Diese äquidistante Umparametrisierung ist sinnvoll, da die Erfassung der Stiftposition zum Schreibzeitpunkt bei den meisten Geräten mehr oder minder mit fester Abtastrate erfolgt, also bei einer schnellen Stifteingabe weniger Abtastwerte anfallen als bei langsameren Eingaben.

Für je zwei aufeinander folgende äquidistanten Punkte  $(x_t, y_t)$  und  $(x_{t+1}, y_{t+1})$  lässt sich der Winkel  $\alpha_t$  der Verbindung bestimmen (siehe Abbildung 3). Die Anzahl unterschiedlicher Richtungen (d.h. der Grad der Quantisierung des Winkels) lässt sich hier (im Gegensatz zum Ansatz aus 2.1) frei wählen. Ein Nachteil ist jedoch die algorithmische Komplexität der vorangehenden Spline-Interpolation.

Ähnlich wie in Abschnitt 2.1 wird auch bei der gitterlosen Ermittlung von Richtungsmerkmalen die geometrische Form der ursprünglichen Stifteingabe als Sequenz von Symbolen ausgedrückt, die jeweils für die Ausprägungen der möglichen Richtungen stehen. Neben der Anzahl möglicher Richtungen ist der Abstand der durch die Interpolation ermittelten Punkte ein möglicher Freiheitsgrad, der Einfluss auf die Länge der Merkmalssequenzen und auf die Genauigkeit der Quantisierung hat.

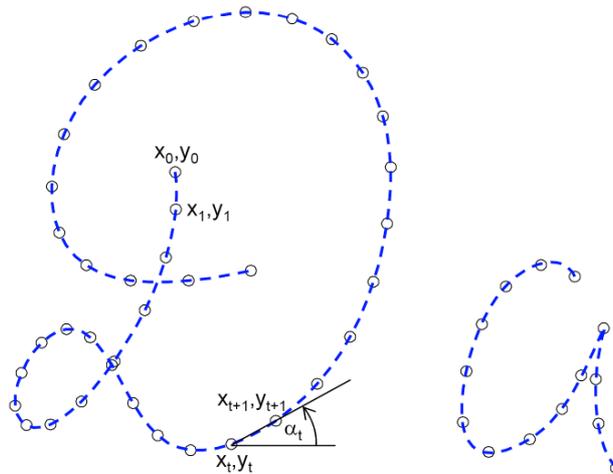


Abb. 3 Lokale Schreibrichtung zwischen zwei Abtastpunkten

## 2.3 Linienkrümmung als Merkmal

Ähnlich der lokalen Richtung der Stiftbewegung, wie sie im Abschnitt 2.2 beschrieben wird, lässt sich die Linienkrümmung (Kurvatur) bestimmen. Genau wie im vorigen Abschnitt wird mittels kubischer Spline-Interpolation eine äquidistante Neuabtastung der Schriftkurve vorgenommen. Statt jedoch für zwei aufeinander folgende Abtastpunkte die Linienrichtung zu ermitteln, wird nun für drei aufeinander folgende Punkte die Krümmung bestimmt. Zwei Möglichkeiten bieten sich dafür an. Zum einen ist dies die Änderung der Linienrichtungen, also eine simple Erweiterung des Verfahrens aus Abschnitt 2.2. Zum anderen ist die Bestimmung des Kurvenradius für die drei Punkte möglich; zu je drei verschiedenen Punkten existiert genau ein vierter Punkt, der Kurvenmittelpunkt, zu dem alle drei Punkte denselben Abstand haben. Die einzige Ausnahme ist, wenn alle drei Punkte auf einer Linie liegen. In diesem Fall wird der Kurvenradius als unendlich groß angenommen.

Sowohl die Winkeländerungen als auch der Kurvenradius kann dazu dienen, die geometrische Form einer Stiftbewegung auszudrücken. Um beschreibende Sequenzen von solchen Merkmalen, vergleichbar denen in den Abschnitten 2.1 und 2.2, zu ermitteln, ist es lediglich nötig, bestimmte Teilbereiche der möglichen auftretenden Werte für die Winkeländerung oder den Kurvenradius zusammenzufassen, d.h. zu quantisieren.

## 3 Suche mittels Approximate String Matching

Die Grundlage der Suche ist das so genannte *Approximate String Matching* [Gusf97]. Es basiert auf einem Maß für den Abstand zwischen zwei Zeichenketten – Editier- oder Levenshteinabstand [Leve65]. Dieser Abstand zweier Zeichenketten  $a$  und  $b$  ist dadurch definiert, dass er die minimal nötige Anzahl an Operationen zählt, um die Zeichenkette  $a$  in  $b$  zu überführen. Erlaubte Operationen sind dabei das Löschen, Einfügen und Ersetzen jeweils einzelner Zeichen. Beim Approximate String Matching wird das Ziel verfolgt, innerhalb eines längeren Strings  $d$  (*document*) die Positionen aller Teilstrings zu finden, deren Editierabstand zu einem kürzeren String  $q$  (*query*) kleiner als ein bestimmter Schwellwert ist. Für die Suche nach handschriftlichen Worten in Handschriftdokumenten wird die Merkmalssequenz  $q$  des Suchwortes mittels Approximate String Matching innerhalb der Merkmalssequenz  $d$  des Doku-

mentes gesucht.  $d$  und  $q$  sind hierbei definiert als Ketten von Symbolen eines endlichen Alphabetes  $\Sigma$  –  $d, q \in \Sigma^*$ .

Die Bestimmung der Positionen dieser Teilstrings wird klassischerweise mit folgender Formel mittels dynamischer Programmierung realisiert:

$$D(i, j) = \begin{cases} 0 & \text{if } i = 0, \\ D(i-1, 0) + 1 & \text{if } i > 0 \text{ and } j = 0, \\ \min \left\{ \begin{array}{l} D(i, j-1) + 1 \\ D(i-1, j) + 1 \\ D(i-1, j-1) + \delta(i, j) \end{array} \right\} & \text{else.} \end{cases}$$

$$\delta(i, j) = \begin{cases} 0 & \text{if } q[i] = d[j], \\ 1 & \text{else.} \end{cases}$$

Hierbei sind  $i$  und  $j$  Indexpositionen innerhalb der Sequenzen  $d$  und  $q$ . Für Merkmalssequenzen  $q$  der Länge  $m$  und Sequenzen  $d$  der Länge  $n$  hat die Matrix  $D$  die Größe  $(m+1) \times (n+1)$ . In der untersten Zeile dieser Matrix befinden sich nach Berechnung durch diese Formel die Werte der Editierabstände zwischen  $q$  und den Teilstrings von  $d$ . Um die Suchergebnisse zu erhalten, bestimmt man jene Werte  $D(m, j)$ , die kleiner als ein bestimmter Schwellwert sind. Wird dieser Schwellwert zu klein gewählt, werden unter Umständen einige Stellen fälschlicherweise nicht gefunden. Im umgekehrten Fall, wenn der Schwellwert zu groß gewählt wird, können Fehlerhafte Ergebnisse auftreten. Es ist leicht ersichtlich, dass die Berechnung der kompletten Matrix  $D$  eine asymptotische Komplexität von  $O(m \cdot n)$  hat, das heißt, mit Verdopplung der Länge der Merkmalssequenzen  $d$  und  $q$  vervierfacht sich der Zeitaufwand.

## 4 Testumgebung

Die Datenerfassung für unsere Testumgebung erfolgte mit einem Logitech ioPen. Dieser Stift verfügt über einen optischen Sensor nahe der Stiftspitze, mit dem die Schreibbewegung erfasst wird (siehe Abbildung 1). Dazu ist es nötig, dass auf Papier geschrieben wird, auf welchem sich ein sehr feines Punktmuster befindet, in dem die jeweilige Position innerhalb einer Seite kodiert ist [Anot05]. Der Stift ist in der Lage, die Schreibdaten von etwa 30 Seiten DIN-A4 zu speichern. Nach Übertragung dieser Daten auf den Computer liegen die x-y-Positionsdaten in Einheiten von etwa 710 lpi (engl. *lines per inch*) vor. Die Abtastrate schwankt zwischen 20 und 50 Hz.

Aktuell neun Personen haben mit diesem Stift bisher etwa 100 Seiten handschriftlicher Aufzeichnungen angefertigt. Die Anzahl geschriebener Seiten pro Person variierte zwischen einer und 20 Seiten. Es wurde dabei darauf geachtet, dass sich in den geschriebenen Texten Worte und Symbole befinden, die mehr als ein Auftreten haben, sodass eine Suche nach diesen Wiederholungen möglich ist. Die Positionen dieser Worte wurden manuell bestimmt und gespeichert, um eine automatische Suche und Bewertung der Suchergebnisse zu ermöglichen. Diese manuelle Bestimmung relevanter Treffer für die Suchanfragen wurde mit einem von den Autoren entwickelten Programm (siehe Abbildung 4) vorgenommen. Es gestattet die Sample-point-genaue Bestimmung von Wortanfang- und -endepositionen.

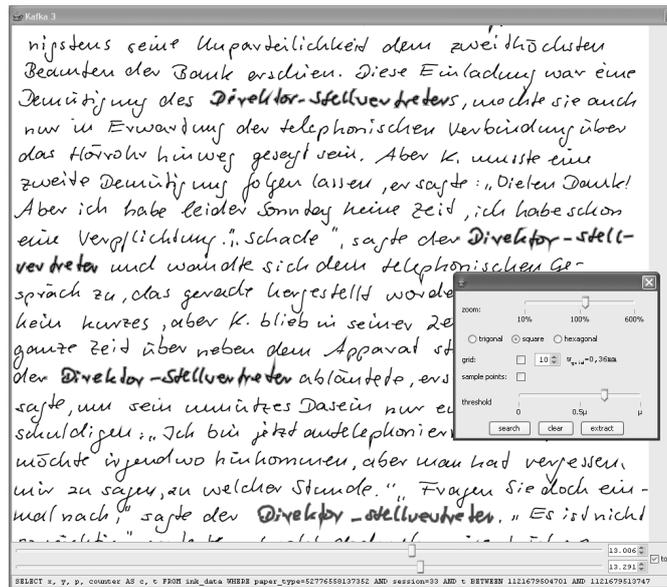


Abb. 4 Screenshot der Benutzeroberfläche zur Suche innerhalb eines einzelnen Dokumentes; die Ergebnisse einer Suche nach dem Wort „Direktor-Stellvertreter“ sind hervorgehoben

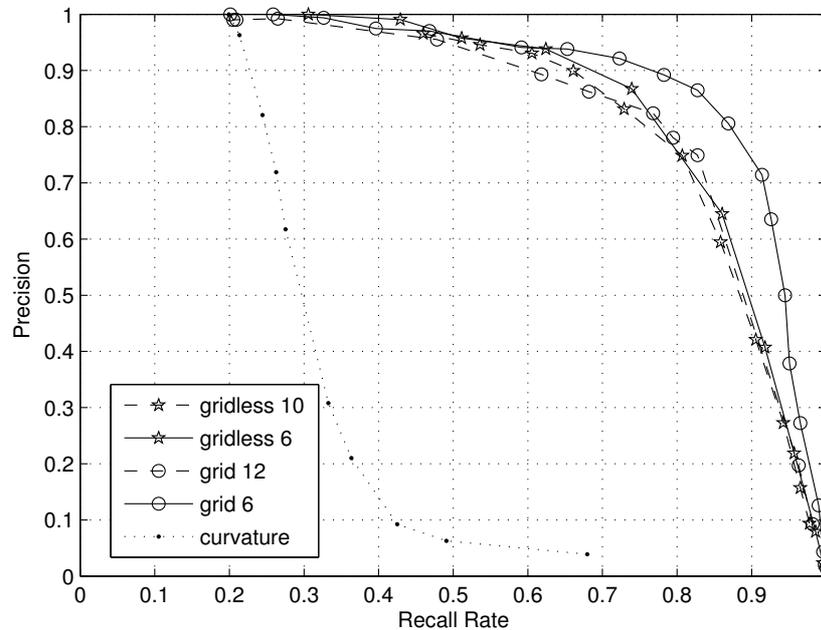
## 5 Testergebnisse

Die Qualität des hier vorgestellten Verfahrens zur Suche in handschriftlichen Dokumenten wird anhand der im Gebiet des Information Retrieval üblichen Qualitätsmaße *Precision* und *Recall Rate* ermittelt. Unter *Precision* wird der Anteil der korrekten Suchergebnisse im Verhältnis zur Gesamtzahl der Suchergebnisse verstanden. Die *Recall Rate* beschreibt die Anzahl der korrekten Suchergebnisse im Verhältnis der erwarteten korrekten Treffer. Im Idealfall sind beide Werte 1 bzw. 100%, das heißt es werden alle relevanten Dokumente gefunden und es gibt keine weiteren (also falschen) Treffer. Aus dem Gebiet der biometrischen Benutzerauthentifikation sind parallel zu diesen Qualitätsmaßen die Falsch-Positiv- und die Falsch-Negativ-Erkennung (bzw. *false acceptance rate/error type I* und *false rejection rate/error type II*) bekannt, also Fehler- anstelle von Erfolgsraten. Als Wert zur vereinfachten Beurteilung der Fehler von biometrischen Systemen gilt die *EER (equal error rate)* – jene Schwellwert-einstellung der Klassifikationsverfahrens, bei der beide Fehlertypen gleich groß sind ( $FAR = FRR$ ). Ein ähnlicher Wert aus dem Gebiet des Information Retrieval ist das  $F_1$ -Measure – das harmonische Mittel aus *Precision* und *Recall Rate* [YaLi99]:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Im Diagramm in Abbildung 4 sind die Erfolgsraten *Precision* und *Recall* für die Suche mittels der drei Merkmalstypen aus den Abschnitten 2.1 bis 2.3 zu sehen. Der Merkmalstyp „grid“ (gitterbasierte Kodierung) wurde im Abschnitt 2.1 näher ausgeführt. Im Diagramm sind die Ergebnisse für Gitterweiten von 6 und 12 Punkten (in Einheiten von 710 lpi) mit Sternen markiert. Für die gitterlosen Merkmale „gridless“ wurden die Rohdaten mittels Resampling (siehe Abschnitt 2.2) so umgerechnet, dass die Abtastpunkte einen Abstand von 6 bzw. 10 Punkten haben. Die Richtungen wurden so quantisiert, dass sich 12 Ausprägungen ergaben. Im Diagramm sind die Ergebnisse für den gitterlosen Merkmalstyp mit Sternen versehen. Für

die Kurvaturmerkmale aus Abschnitt 2.3 wurde ein Resampling auf einen Punktabstand von 25 Einheiten verwendet. Die Anzahl unterschiedlicher Richtungen beträgt 16. Die Ergebnis-kurve im Diagramm für die Kurvatur ist als punktierte Linie dargestellt.



**Abb. 5** Erfolgsraten der Handschriftsuche für verschiedene Merkmalstypen als ROC-Kurven

Wie in Abbildung 5 zu erkennen ist, sind die *Precision* und die *Recall Rate* für Kurvaturmerkmale sehr gering im Vergleich zu den anderen Merkmalstypen. Deutlich erkennbar ist dies auch am  $F_1$ -Measure in Tabelle 1. Es ist weiterhin ersichtlich, dass die Suche mithilfe der richtungsbasierten Merkmale mit einem  $F_1$ -Measure zwischen 0,77 und 0,84 eine erheblich bessere Performance hat. Hierbei sticht die gitterbasierte Richtungskodierung mit einer möglichst kleinen Gitterweite mit  $F_1=0,84$  besonders hervor.

**Tab. 1:** Erfolgsraten der Handschriftsuche für verschiedene Merkmalstypen

Merkmal	Precision	Recall Rate	$F_1$ -Measure
<i>gridless 6</i>	0.86	0.74	0.795
<i>gridless 10</i>	0.83	0.73	0.777
<i>grid 6</i>	0.86	0.83	0.844
<i>grid 12</i>	0.78	0.79	0.785
<i>curvature</i>	0.31	0.33	0.32

## 6 Zusammenfassung und Ausblick

Es wurde ein neues Verfahren vorgestellt, das es gestattet, innerhalb von on-line erfassten, digitalen handschriftlichen Dokumenten zu suchen. Dabei wurde auf eine Texterkennung verzichtet und stattdessen eine Wiedererkennung anhand der graphischen Form realisiert. Hierfür kam das Approximate-String-Matching zum Einsatz – eine Methode zur „unscharfen“ Suche

in Zeichenketten, die auch in der Bioinformatik zur Suche nach Teilen von DNA-Sequenzen Verwendung findet.

Dieses Matching-Verfahren wurde mithilfe von Testdokumenten von neun verschiedenen Personen für drei verschiedene Merkmalstypen evaluiert und anhand der Qualitätsmaße *Precision*, *Recall Rate* und *F<sub>1</sub>-Measure* bewertet. Im besten Fall konnten mithilfe gitterbasierter Merkmale (siehe Abschnitt 2.1) eine *Precision* von 86% bei einer *Recall Rate* von 83% erzielt werden.

Gegenstand zukünftiger Untersuchungen wird der Einfluss der einzelnen Parameter der verschiedenen Merkmalstypen (bspw. Gitterweite oder Grad der Richtungsquantisierung) auf die Suchqualität sein. Zusätzlich soll die Datenbank der handschriftlichen Dokumente kontinuierlich vergrößert werden, um aussagekräftigere Resultate zu erhalten.

## Danksagung

Diese Veröffentlichung entstand unter Mithilfe des EU Network of Excellence SIMILAR (FP6–507609). Der Inhalt dieser Veröffentlichung steht in alleiniger Verantwortung der Autoren und widerspiegelt somit in keiner Weise die Meinung der Europäischen Union. Weiter bedanken wir uns an dieser Stelle bei allen Mitarbeitern und Praktikanten der Arbeitsgruppe Multimedia and Security für die Mitarbeit beim Sammeln der Testdaten.

## Literatur

- [Anot05] Anoto Group AB, <http://www.anotofunctionality.com/>, 2005.
- [Boor78] C. de Boor: A Practical Guide to Splines, Springer-Verlag (1978).
- [Free74] H. Freeman: Computer Processing of Line-Drawing Images, In: Computer Surveys, ACM (1974), 57-97.
- [FrHu96] D. Frohlich, R. Hull: The usability of scribble matching, In: Proceedings of CHI'96, ACM (1996) 189-190.
- [GoXu04] V. Govindaraju, H. Xue: Fast Handwriting Recognition for Indexing Historical Documents, In: Proceedings of the 1<sup>st</sup> International Workshop on Document Image analysis for Libraries (DIAL'04), IEEE (2004), 314-320.
- [Gusf97] D. Gusfield: Algorithms on Strings, Trees, and Sequences, Cambridge University Press (1997).
- [JaNa03] A. K. Jain, A. M. Namboodiri: Indexing and Retrieval of On-line Handwritten Documents, In: Proceedings of the 7<sup>th</sup> International Conference on Document Analysis and Recognition, IEEE (2003) 655-659.
- [LeP194] F. Leclerc, R. Plamondon: Automatic Signature Verification: The State of the Art – 1989-1993, In: International Journal of Pattern Recognition and Artificial Intelligence (Vol. 8, No. 3, 1994), 643-660.
- [Leve66] V. I. Levenshtein: Binary codes capable of correcting deletions, insertions and reversals, In: Soviet Physics Doklady (Vol. 10, No. 8, 1966), 707-710.

- [LoTo94] D. Lopresti, A. Tomkins: On the Searchability of electronic Ink, In: Proceedings of International Workshop on Frontiers in Handwriting Recognition (1994), 156-165.
- [MaHR96] R. Manmatha, C. Han, E. M. Riseman: Word Spotting: A New Approach to Indexing Handwriting, In: Computer Vision and Pattern Recognition, IEEE (1996), 631-637.
- [OKON04] H. Oda, A. Kitadai, M. Onuma, M. Nakagawa: A search method for on-line handwritten text employing writing-box-free handwriting recognition, In: 9<sup>th</sup> International Workshop on Frontiers in Handwriting Recognition (IWFHR'04), IEEE (2004), 545-550.
- [PeRZ02] M. P. Perrone, G. F. Russel, A. Ziq: Machine learning in a multimedia document retrieval framework, In: IBM Systems Journal (Vol. 41, No. 3), IBM (2002), 494-503.
- [ScVD04] S. Schimke, C. Vielhauer, J. Dittmann: Using Adapted Levenshtein Distance for On-line Signature Authentication, In: Proceedings of the 17th International Conference on Pattern Recognition, IEEE (2004), 931-934.
- [SrHS04] S. Srihari, C. Huang, H. Srinivasan: A Search Engine for Handwritten Documents, In: Document Recognition and Retrieval XII, Society of Photo Instrumentation Engineers (2005), 66-75.
- [YaLi99] Y. Yang, X. Liu: A re-examination of text categorization methods, In: Proceedings of the 22<sup>nd</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM (1999), 42-49.