Document Retrieval in Pen-Based Media Data

Sascha Schimke and Claus Vielhauer
Otto-von-Guericke University of Magdeburg, Germany
{sascha.schimke|claus.vielhauer}@iti.cs.uni-magdeburg.de

Abstract

The rise of devices with pen as the primary input option (personal digital assistants and Tablet PCs) or specially equipped ink pens for capturing scripture on paper will produce large amounts of handwritten or hand drawn documents. In this paper we describe a prototype implementation of a search system for this kind of pen-based documents without involving character recognition. Although in the last years handwriting recognition systems experienced large performance improvements, they lack in situations of drawing recognition and for people, having a very unclear script style. Our algorithm for handwriting retrieval achieves in database tests with 59 documents from twelve users a performance of about 81% recall and precision rate.

1. Introduction

Even in the age of omnipresent computers, the handwriting as a way for communication and interaction doesn't lose ground. An indicator for this is the increasing number of pen oriented devices like Tablet PCs, personal digital assistents (PDA) or special electronic pens, which are able to digitize the pen position data while writing movement. With all these devices it is possible to write and store handwritten inputs. The common method for handling these handwritten inputs is the recognition, i.e. the automatic translation into ASCII. This proceeding is well researched and the current recognizer algorithms work stable in many situations. But in the case of individuals with an unclear writing style, the recognition often is very erroneous. Another case, where a textual recognition is not adequate, is, when no text at all is entered, namely when the user sketches for example diagramms. In these situations it is the better solution to store the pen-input without processing and handle it as "virtual ink" instead as textual content.

Unlike in off-line handwriting, where an image of the written data is the basis of all processing steps, we focus on on-line handwriting data, which are acquired by sampling the position and pressure of the pen tip while the activity of writing. So the basis for all further processing steps are time discrete signals ($x(t)$, $y(t)$, $p(t)$).

To manage large amounts of such pen-originated documents, it is necessary to have a search functionality. This insight is not new. In 1994, Lopresti and Tomkins presented a first idea for searching in handwritten documents [1]. Their ScriptSearch algorithm works as follows: the sequence of handwriting sampling points is segmented in strokes. For the latter 13-dimensional feature vectors are extracted (using Rubine's algorithm [2]) and quantized into 64 clusters. The result is a sequence of clusters, which represents the former sequence of strokes. In order to search for a phrase within a document, both, the phrase and the document are translated in such sequences. The actual search is performed by approximated string matching. The best achieved results are 87% precision for a recall rate of 80% or 68% precision for 90% recall rate.

Another approach was presented in 1995 [3], where the search process was performed on a word-word basis, instead of word-text. This means, that a search word is compared separately with each word from the document. The implication of this procedure is, that if the word segmentation fails, the whole searching process fails too. The purpose of the system in [3] was a pen based query in personal address or phone books. So the word segmentation is suitable. The authors don't report any evaluation results in the form of error rates.

In [4] an approach for retrieval of handwritten documents is described, which bases on wordwise comparison using dynamic time warping, which is done after a quality enhancing pre-processing. The features, which were used for comparison are the y-coordinate, the direction and the curvature of the ink trace in the sampling points. The precision of this approach is given as 92.3% at a recall rate of 90%. Because of different test databases and different evaluation proceedings, these existing approaches and out new one are not easily comparable.

0-7695-2625-X/06 $20.00 © 2006 IEEE

Our novel retrieval system will be explained in the following two sections. We try to avoid two disadvantages of the three approaches above: a) the need for segmentation of the documents in words or other basis entities and b) the requirement of complex and complicated algorithms for feature extraction. But thereby we adapt some techniques of the existing approaches above.

2. Features

As mentioned in section 1, the basis of on-line handwritten documents are sampled signals, i.e. sequences of pen tip positions. Sampling rate and the position resolution is dependent on the used acquisition device. For retrieval in handwritten documents, we utilize a technique called approximate string searching, which realizes a fuzzy substring search (see section 3).

For using this technique, it is necessary to extract string-like features, which represent the ink traces of the writing process. Figure 1 shows the basic idea of our approach. The handwritten input (dashed line) is superimposed with a square grid. For each sampling point (gray circles) the next grid node is determined. Considering the eight neighbors of each grid node, it is possible to code the original ink shape by using eight symbols, which stand for the eight possible directions (solid line). This idea was first presented by H. Freeman in 1974, who used this method for efficient coding of line drawings [5].

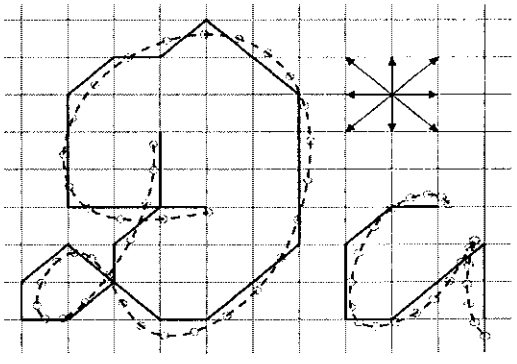

Figure 1. Quantization of ink traces for feature extraction.

To code a gap between segments, a ninth symbol can be used. This direction and gap coding is a kind of spatial quantization.

The resulting code sequences are dependent on the size of the quantization square, i.e. the grid width. The larger this grid width is, the more imprecise is the coded sequence. The smaller this grid width is, the more accurate and longer is the coded sequence. In section 5 we test different grid widths.

Using grid width, which is too small, could lead to a coding of not only the user intended handwriting data but even of the involuntary noise of the hand movement, which would influence the retrieval performance in a negative way.

Instead of using a square grid for quantization of ink traces, it is possible choose trigonal or hexagonal grids. An advantage of these over square grids is, that for square grids the eight neighbors of a node have different distances (grid width w_g and $2^{1/2} \cdot w_g$). A disadvantage is the smaller number of node neighbors; six for trigonal grids and only three for hexagonal ones. In this paper we restrict our self on square grids.

Due to the characteristic of the direction based features, two sequences of the same word, written by two different persons will be on many cases not similar, because the stroke ordering and the writing style is different between. So, a simultaneous retrieval in documents of different persons is hardly possible with good results. That's why our search system is intended for an individual usage in one's own documents.

3. Approximate String Searching

To search for a short string q within a longer string d, we use the approximate string searching method. The idea is to find those substrings of d, which have an edit-distance [6] to q smaller than a threshold τ. The approximate string searching problem can be solved using the following equation:

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

In this formula, D is a matrix of size $(m+1)\times(n+1)$, where m and n are the lengths of q and d, respectively. The function $\delta(i, j)$ is defined:

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

where with $0 \le i \le m$ and $0 \le j \le n$ and $q[i]$ or $d[j]$ are the i^{th} or j^{th} symbol in the strings q or d, respectively.

Those positions $d[j]$, where $D(m, j)$ is smaller than τ, are the end points of substrings of d, which are similar to q. To consider the influence of string length m on the value of $D(m, j)$, we make a normalization by dividing through m before comparing with the threshold τ.

It is obvious, that the computational complexity of the approximate string searching, using the explained method, is $O(m \cdot n)$.

For searching in handwritten documents, our system first extracts direction feature strings from query word or phrase and from all documents. Then for each document string, the substring searching for the word string is performed, as described above. As a result, all matching documents are displayed with highlighted occurences of the search word. See figure 2 for an example of a handwritten document with four search results highlighted.

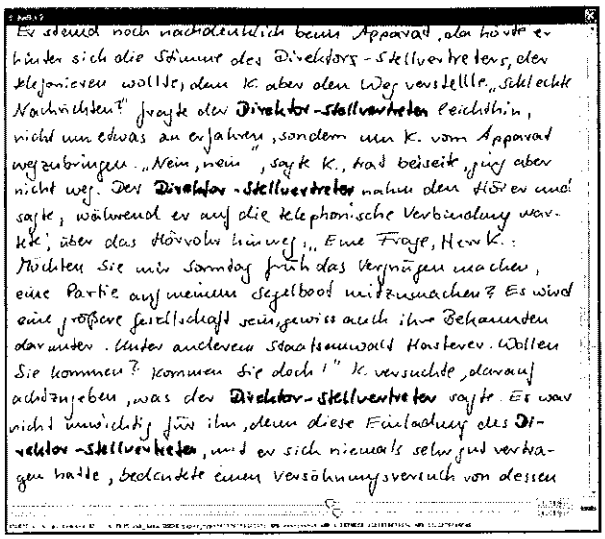

Figure 2. Handwritten text and four highlighted matches.

4. Data Collection

For ongoing testing our system, we are still collecting handwritten and handdrawn documents. We ask different people to write at least one page of text of their own choice. They are free draw diagrams or sketches additinally. Until now our database consists of 70 documents from seven individuals. For acquisition we used the *Logitech ioPen* [9], but we plan to broaden our focus to more and different pen devices, e.g. TabletPCs and *Pegasus PC Notes Taker*, which works on basis of ultrasonic measuring of pen position [10].

The *ioPen* device is a ballpoint pen, which is equipped with an optical sensor next to the pen tip (see figure 3). When writing on special paper, the optical sensor and the built-in electronic are able to determine the actual position on the sheet of paper as well as the page number. This position information is coded in fine dot pattern (see [8]) at the special paper. Figure 4 shows the structure of this pattern; it consists of small dots having a diameter of 0.1mm and an average distance of 0.3mm. The position is coded in the displacement of these dots. Using an array of 6×6 dots, the pen determines its position.

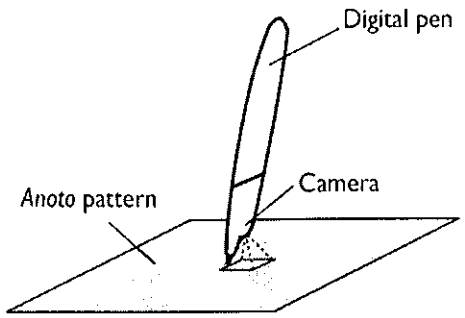

Figure 3. Sketch of a digital pen for optical position determination [8].

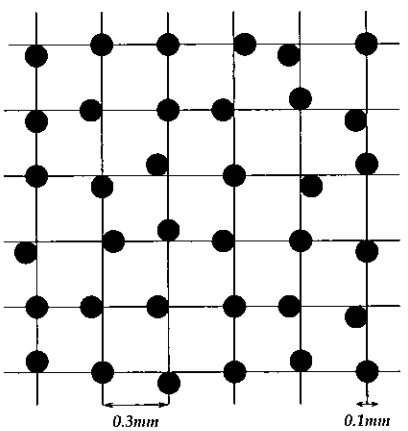

Figure 4. Example of dot pattern for coding paper position.

The sample point coordinates, as the come from the device driver, are given at 680 units per inch both horizontally and vertically. The sampling rate is variable and can reach up to 50Hz.

Using this *ioPen* digitizer device, we acquired 59 documents (a document corresponds to one A4 sheet of paper) consisting of 10,990 words and 65 icons or sketches from twelve persons. The persons were asked to write free text or transcribe articles from books or journals. Each person wrote up to twelve documents. Document languages are English and German. In Figure 5 six short excerpts from handwritten documents from different persons can be seen. [1]

[1] The public part of our database is available under http://wwwiti.cs.uni-magdeburg.de/~sschimke/hwdb/

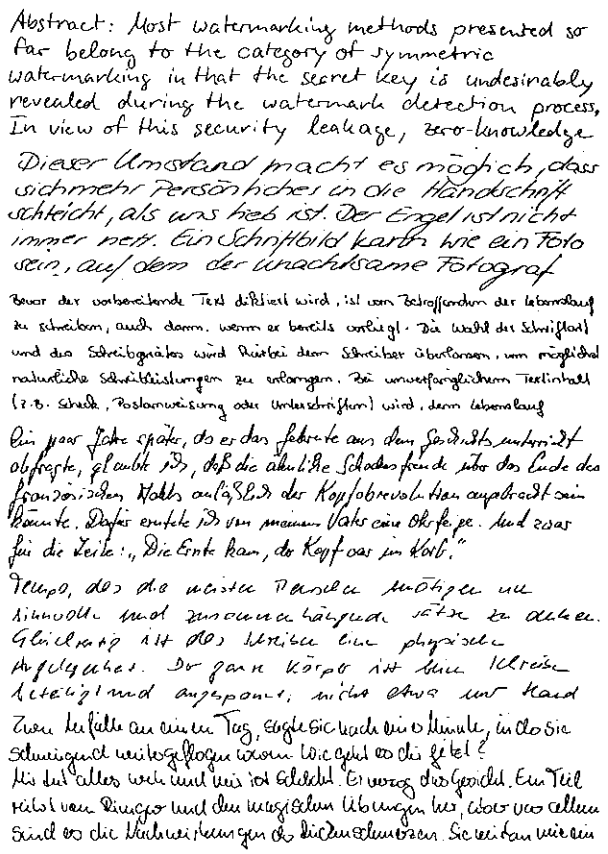

Figure 5. Exemplary excerpt of our handwriting database.

5. Evaluation and Experimental Results

In order to evaluate the searching performance, in our ben-based documents (see section 4) a set of 271 search queries (words, short phrases and symbols) as well as all their respective repetitions in the documents (i.e. the positions of expected correct matches) were tagged manually. Then the system was tested with different setting for grid width and threshold. The grid widths are given in points (pt), i.e. as integral multiples of the basis unit of pen coordinates, coming from the sampling device (see section 4). For each parameter setting the number of correct matches, mismatches and missed instances of the search queries were counted in order to obtain recall and precision rates. Because of the user dependent nature of our system we didn't perform queries to the complete database but only to all single users' documents.

Using the mentioned measures *number of correct matches*, *mismatches* and *missing matches* we calculate the retrieval measures *recall rate*, *precision* and F_1-*Measure* [7]:

$$precision = \frac{matches}{matches + mismatches}$$

$$recall = \frac{matches}{matches + missings}$$

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

A graphical representation of these rates is shown in Figure 6. It can be seen, that there is a correlation between retrieval performance and the size of quantization grid. Using a fixed threshold for all persons, we got results of 81% precision at a recall rate of 81%. As can be seen, the performance is so much the better the smaller is the grid width. First continuative tests show, that an individual threshold for each user can enhance the precision and recall rate. But until now there are not enough test data for a robust substantiation of this fact.

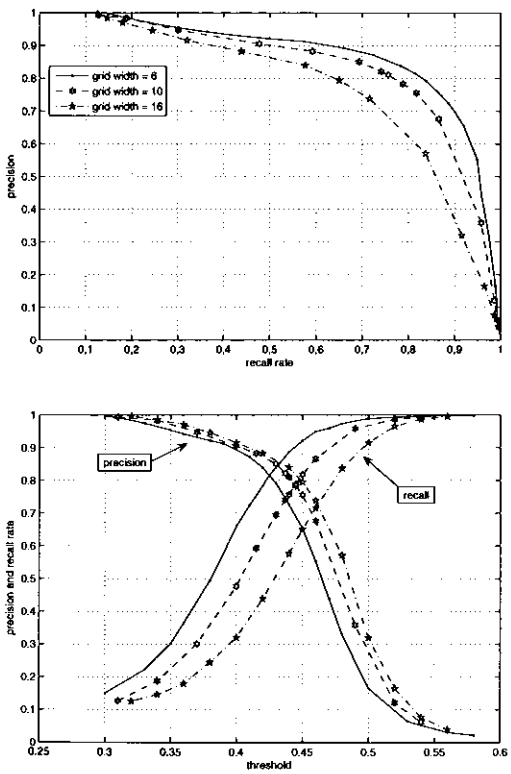

Figure 6. Recall rates and precision as ROC curve and as function of threshold.

The tests were performed on a notebook computer, equipped with a 1.6GHz Pentium M processor, 512MB RAM; the system is WindowsXP and the implementation of our retrieval system uses Java JRE 1.5.

Table 1 lists the precision, recall rate, F_1-Measure and the average search time (in ms) per document for three different grid widths. It is visible, that for better

retrieval performance a higher search time per document is needed. This is a result of the $O(m \cdot n)$ complexity of the used search algorithm (see section 3).

grid width	precision	recall	F_1	ms/document
6	0.815	0.816	0.815	1,555
10	0.783	0.788	0.785	572
16	0.738	0.716	0.727	285

Table 1. Recall rates and precision as ROC curve and as function of threshold.

6. Conclusion and Future Work

In this paper we presented a novel algorithm for the retrieval in handwritten documents. Initial performance evaluations showed it's general possibility for this task. Our future work will include further testing with an increasing document database to evaluate the power of user dependent parameter optimization. At the algorithmic front we will analyse other methods of quantization, e.g. trigonal or hexagonal quantization instead of the square one, as described in section 2. Furthermore we try will try to reduce to needed retrieval time by searching for adequate indexing strategies.

Acknowledgements

This work has been partly supported by the EU Network of Excellence SIMILAR (FP6–507609). The contents of this publication are the sole responsibility of the authors and can in no way be taken to reflect the views of the European Union.

We would like to thank the colleagues and students of the Research Group Multimedia and Security for their assistance while collection handwriting data.

References

[1] Daniel Lopresti, Andrew Tomkins, "On the Searchability of Electronic Ink", *Proc. of International Workshop on Frontiers in Handwriting Recognition*, 1994, pp. 156-165.

[2] Dean Rubine, *The Automatic Recognition of Gestures*, PhD thesis, Carnegie Mellon University, 1991.

[3] David Frohlich, Richard Hull, "The usability of scribble matching", *Proc. of CHI'96*, ACM, 1996, pp. 189-190.

[4] Anil K. Jain, Anoop M. Namboodiri, "Indexing and Retrieval of On-line Handwritten Documents", *Proc. of the 7th International Conference on Document Analysis and Recognition*, IEEE, 2003, pp. 655-659.

[5] Herbert Freeman, "Computer Processing of Line-Drawing Images", *Computer Surveys*, Vol. 6, No. 1, ACM, 1974, pp. 57-97.

[6] Dan Gusfield, *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, 1997.

[7] Yiming Yang, Xin Liu, "A re-examination of text categorization methods", *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 42-49.

[8] Anoto Group AB, http://www.anotofunctionality.com/.

[9] Logitech Inc., http://www.logitech.com/.

[10] Pegasus Technologies Ltd., http://www.pegatech.com/.