Paper prepared for the 6th International Conference on Handwriting and Drawing (ICOHD'93), (IGS), 5-7 July 1993, Paris, France.

# Adaptive recognition of online, cursive handwriting

L.R.B. Schomaker, Eric H.Helsper, H.L.Teulings, G.H.Abbink

Nijmegen Institute for Cognition and Information (NICI)
P.O. Box 9104, NL-6500 HE, Nijmegen, The Netherlands
Tel: +31-80-616029; Fax: +31-80-615938
E-mail: Schomaker@nici.kun.nl

## INTRODUCTION

In earlier studies, a stroke-oriented recognizer (VHS) of on-line cursive handwriting is reported [Thomassen et al., 1988; Schomaker & Teulings, 1990; Teulings et al., 1990; Schomaker & Teulings, 1992; Schomaker, 1993]. This system uses a movement-theoretical segmentation into strokes as the starting point of the recognition process. The pen-tip trajectory of a written word is low-pass filtered, and geometrically normalized with respect to size and slant. The absolute velocity of the pen-tip displacement is calculated, and the signal is segmented in strokes, each stroke being the trajectory between two robust minima in the absolute velocity [Teulings et al., 1987]. Strokes are characterized by feature vectors that are clustered using a Kohonen Self-Organizing Map as a feature quantizer. In the current system, as opposed to earlier versions, a number of typical problems in connected-cursive and mixed-cursive script recognition are dealt with, such as t-bar crossing, dotting of i's and j's, and hesitations. Processing stages in the on-line cursive recognizer VHS:

- preprocessing: Low-pass filtering, differentiation
- segmentation: Find velocity-based strokes, white spaces, dots and t-bar crossings
- normalization: Slant, Size
- feature extraction: 9 angular, 3 Cartesian, 2 structural features
- stroke classification: Kohonen net, $O[n]$
- letter classification: Stroke transition net $O[n^2]$
- word classification: recursive tree traversal with binary word search $O[x^n]$

Here, $O$ is the computational complexity order, $n$ is the number of strokes, and $x$ is the average number of active letter hypotheses per stroke.

In this study, the problem is addressed how a handwriting recognition system, trained on the handwriting of a limited number of writers, can be used as a starting point ("bootstrap") for adaptation to a new writer. In an earlier study, it appeared that about 500 words per writer were necessary, if training started with a blank system. This is unacceptable, since labeling requires about 1-2 hours of work per 100 written words. Methods for automatizing the labeling task have been proposed, but are of limited use, since manual interaction is still required [Teulings & Schomaker, 1992]. Apart from the training issues, there is the problem of lexicon size. In this study we will vary dictionary size in the recognition of a number of good test sets to investigate its effect on the recognition performance. Furthermore, computation time is assessed as a function of the number of letter hypotheses for given input words.

## METHOD

**Phase 1: Training the "bootstrap" system.** Isolated words from 17 Dutch writers, (age 18-35) were collected. On average, each writer produced 219 cursive or mixed-cursive isolated words from a printed list, writing on a Calcomp 2500 digitizing tablet with an inking ballpoint pen. Stroke feature vectors were calculated from all words and a 20x20 Kohonen self-organizing map was trained in order to have a list of prototypical strokes, describing the ensemble of strokes in the training set with a minimized rms error. This network is considered to be a good estimate of all possible stroke shapes in the target writer population and is not updated for new writers, in this study. The allographs were manually labeled and stroke interpretations were added to the Kohonen cells, yielding a transition network of possible cell-to-cell connections. Allographs were only labeled if they were clearly legible in isolation from the word context. Of the total of 3731 words, there were 2827 words from which allograph labels were actually used. Table 1 gives an overview of the training set.

| Writer | Age | Hand | Male/Female | #Words | #Words Labeled |
|---|---|---|---|---|---|
| 01 | 27 | L | M | 410 | 409 |
| 02 | 30 | R | M | 405 | 405 |
| 03 | 39 | R | M | 577 | 576 |
| 04 | 35 | R | M | 671 | 671 |
| 05 | ? | ? | F | 46 | 46 |
| 06 | ? | ? | M | 82 | 82 |
| 07 | 22 | R | F | 140 | 137 |
| 08 | 19 | R | F | 140 | 100 |
| 09 | 19 | R | F | 140 | 69 |
| 10 | 29 | R | F | 140 | 63 |
| 11 | 20 | R | F | 140 | 41 |
| 12 | 20 | R | F | 140 | 95 |
| 13 | 18 | R | F | 140 | 39 |
| 14 | 19 | R | F | 140 | 41 |
| 15 | 18 | R | F | 140 | 32 |
| 16 | 26 | R | M | 140 | 5 |
| 17 | 18 | R | F | 140 | 16 |
| Total | | | | 3731 | 2827 |

Table 1: Training set for the "bootstrap" system.

**Phase 2: Testing the adaptivity of the system.** The adaptive training of the recognition system on the handwriting of a new writer consists of three stages: I. Allograph Probability Adaptation, II. Allograph Labeling, and III. Final Allograph Probability Adaptation. In stage I, the "bootstrap" system, consisting of the 17-writer Kohonen net and Transition net $U_i$ ("user independent"), is used to recognize a new writer's set for the first time. If a word is found in the Top-20 list of output words from the recognizer, the probability of matching allographs in the transition net is incremented in small steps until either the target word is at the top of the output list of words, or until a maximum number of iterations is reached. This operation yields a transition network $U_i'$ (modified user independent). Stage II, Allograph Labeling, is a manual process using a graphical pen-driven interface. A list of rules/criteria is used to obtain consistent labeling. Only allographs in un-recognized words are labeled. Completely idiosyncratic shapes were not labeled. Stage III is the same as stage I, with the difference that the starting point is the transition network $U_i'$, now also using the newly labeled allographs for this writer. The output of the third stage is the user-dependent stroke transition network $U_d$, on which the test sets were tested. Effectively, this network contains adjusted probabilities for individual stroke interpretations, as well as new stroke interpretations, typical for the new writer. There were 11 writers (Italian and Irish) in the first test. Words were recorded using a Wacom HD-648A LCD-integrated digitizing tablet, using a Pen Windows data collection application. The first half of a set was used for training, the second half was used as test set. A second test was performed with 5 Dutch writers (students, age 19-21, 3F/2M, 4 righthanded, 1 lefthanded), with predominantly **mixed cursive** and **handprint** writing styles. Each writer wrote 100 words, of which 50 words were used as training set and the other 50 words as test set. The training procedure was the same as in the first test.

## RESULTS & DISCUSSION

### Word recognition rate as a function of training

Table 2 shows the untrained and trained recognition results. Looking at the "Topword recognized" column, roughly four types of writers can be identified. The table is sorted from high to low initial recognition rate. There is a group of "good" writers (it1,ir1,ir2), starting at 40% and up before training, ending at rates of up to 83% after training. Then there is a group of writers (irb2,ir3,irb1,ir4) that start with mediocre initial recognition rates of 25-41%, ending with modest rates of 50-57% recognition, but with a promise for improvement through additional training. This can be inferred from the Top-10 column, where 63-80% may be obtained. The third group (it2,it3) starts with low rates (14-17%) which is elevated to acceptable levels (70-77%) through training. The fourth group (ir5,it4) consists of writers with a very low initial recognition rate (2-14%) that can be increased to (38-58%) but with little hope for improvement through training as evidenced from the Top-10 column. Independent human readers classified ir5 and it4 as very sloppy handwriting, with idiosyncratic allographs (writer it4 wrote <*p*> with the shape of a <*j*>). This illustrates the problem how to decide if a shape should be labeled.

| Writer | Topword | Top-5 | Top-10 | Nwords | Nlabeled | Nlexicon |
|---|---|---|---|---|---|---|
| it1 | 66/83 | 84/90 | 85/90 | 180 | 18 | 7k |
| ir1 | 44/69 | 67/83 | 72/86 | 120 | 39 | 6k |
| ir2 | 42/60 | 58/73 | 59/78 | 113 | 52 | 6k |
| irb2 | 41/55 | 53/65 | 53/67 | 51 | 53 | 6k |
| ir3 | 38/55 | 55/76 | 59/80 | 113 | 37 | 6k |
| irb1 | 37/57 | 47/63 | 47/63 | 49 | 49 | 6k |
| ir4 | 25/50 | 43/70 | 47/72 | 129 | 71 | 6k |
| it2 | 17/77 | 23/88 | 25/89 | 180 | 65 | 7k |
| ir5 | 14/38 | 26/50 | 26/53 | 136 | 49 | 6k |
| it3 | 14/70 | 21/81 | 21/82 | 180 | 56 | 7k |
| it4 | 2/58 | 3/58 | 3/58 | 180 | 64 | 7k |

Table 2. Recognition rate in % words, **before/after** training. Writer: codes it.. are Italian writers, ir.. are Irish writers The recognizer output is a list of words which is sorted in descending order of match quality. Topword: % of correct words at the top of the output list. Top-5: % of correct words found in the topmost 5 words of the recognizer output list, Top-10: % of correct words found in the topmost 10 words. Nwords: number of words in the test set. Nlabeled: number of manually labeled words in the training set, Nlexicon: number of words in the lexicon used in recognition. With special thanks to Olivetti, Naples, and Captec, Dublin, who kindly provided the handwriting data within the framework of Esprit project P5204 Papyrus.

The same procedure was applied to sets from five Dutch writers (students, age 19-21, 3F/2M, 4 righthanded, 1 lefthanded), with predominantly **mixed cursive** and **handprint** writing styles. Each writer wrote 100 words, of which 50 words were used as training set and the other 50 as test set. Table 3 gives the results.

| Writer | Topword | Top-5 | Top-10 | Nwords | Nlabeled | Nlexicon |
|---|---|---|---|---|---|---|
| nl5 | 50/68 | 60/82 | 64/82 | 50 | 43 | 5k |
| nl2 | 44/66 | 60/82 | 62/82 | 50 | 49 | 5k |
| nl1 | 44/70 | 72/82 | 74/82 | 50 | 38 | 5k |
| nl4 | 46/66 | 62/72 | 64/74 | 50 | 37 | 5k |
| nl3 | 40/68 | 48/78 | 48/80 | 50 | 41 | 5k |

Table 3. Recognition rate in % words, **before/after** training, for five Dutch writers.

The recognition rates in this second test are comparable to the results of writers ir1, ir2, irb2, and ir3 in the first test.

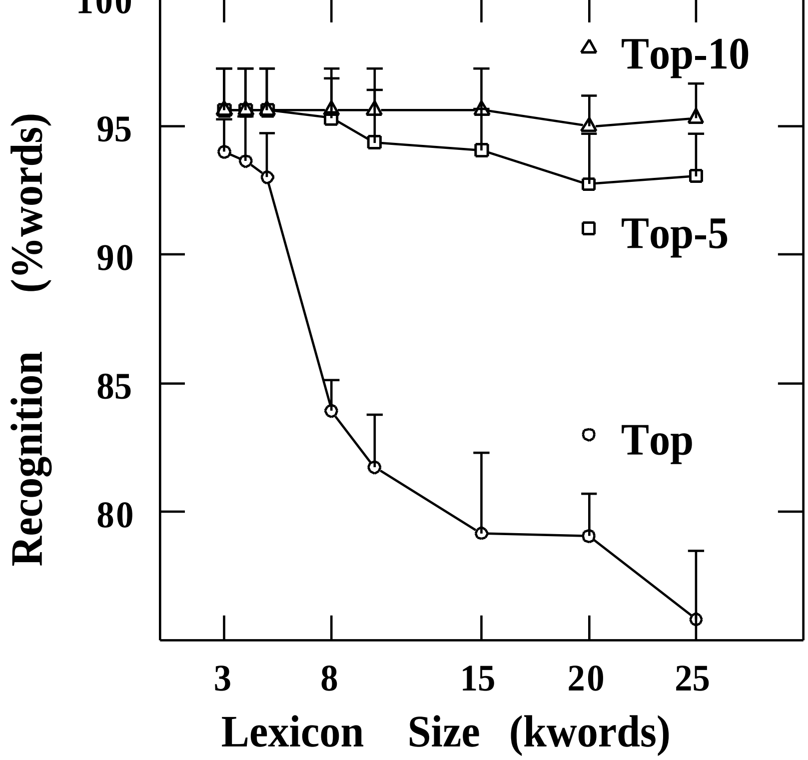


Figure 1. Average effect of lexicon size on the word recognition rate (N=3 sets, 100 words each). Vertical bars display the standard deviation of the recognition rate.

**Word recognition rate as a function of lexicon size**

Three "good" testsets of 100 English words from a single writer were used in a test on the effect of lexicon size. The size of the training set from this writer was over 600 words. As can be seen from Figure 1, going from 3k words to 25k words, there can be a 15% decrease in recognition for the Topword correct, whereas the decrease is much smaller ( $< 3\%$) for the Top-5 and Top-10 recognition rates. The computation time per word, expressed in SUN Sparcstation2 cpu seconds increased from 2s on a 3k lexicon, to 4s on a 25k lexicon. Repeating this test on a single test set of Dutch words, going from 3k words to 50k words gave a 10% decrease in the Topword recognition rate, and a $< 3\%$ decrease for both Top-5 and Top-10 recognition rates.

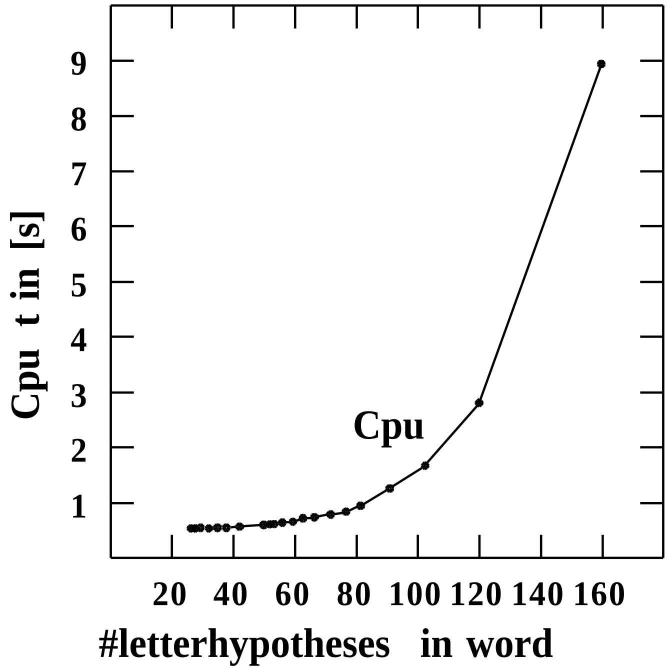


Figure 2. Effect of the number of letter hypotheses per word on computation time. The Y-axis displays the Sparcstation2 CPU time in [s], The X-axis displays the number of letter hypotheses per word, varied by changing a probability threshold on stroke name usage.

**Computation time as a function of the number of letter hypotheses.**

Figure 2 shows the effect of changing the number of letter hypotheses. In the current system this can be done by varying a threshold on the probability of a give stroke name, given a best-matching Kohonen cell. A single test set (100 words) was processed a number of times with a number of threshold values and the average CPU time per word was calculated. It can be seen that the computation time rises exponentially with increasing number of letter hypotheses, as could be expected, given the tree-based word search algorithm. Beyond 100 letter hypotheses per word, computation time rises steeply.

**Word recognition rate as a function of lexical postprocessing**

It should be noted that in this study, each individual allograph must be classified correctly by the letter classification stage. Applying fuzzy matching in case of rejected words yields an improvement. Fuzzy matching was done by simply counting the number of correct allographs in letter zones for each word in the lexicon. It was found that improvements for a good writer are only marginal (it1, top-5: 90 $\rightarrow$ 91%), whereas sloppy writing may benefit substantially from this computationally expensive matching method (it4, top-5: 58 $\rightarrow$ 84%).

## Conclusions

A training set based on a limited number of writers could be used to obtain reasonable recognition rates after individual additional training of a new writer's handwriting on the basis of $\approx$ 100 words. However, the manual labeling is a tedious process, and an intelligent user interface must be developed to alleviate this problem by giving reasonable "hints" about possible letter segmentations. Letter shapes appear to vary considerably among writers. By combining the allograph information from several writers in a single system, the number of letter hypotheses per word increases. As a consequence of the tree-based word search in letter hypothesis space, the computation time rises steeply. A possible method to avoid this problem is the a priori identification of writing style (like in some Multifont OCR recognizers of machine print). The results are encouraging, but practical applications require $> 95\%$ Topword recognition. Fuzzy lexical postprocessing may improve the recognition rate, but an elegant, scalable, lexical post-processing algorithm which can handle local letter ambiguity is still lacking. Future work will be directed at combining two different approaches is one single system: Stroke-based and Character-based cursive recognizers.

## REFERENCES


SCHOMAKER, L.R.B. (1993). Using Stroke- or Character-based Self-organizing Maps in the Recognition of On-line, Connected Cursive Script. *Pattern Recognition*, 26(3), 443-450.

SCHOMAKER, L.R.B., & TEULINGS, H.-L. (1992). Stroke- versus Character-based Recognition of On-line, Connected Cursive Script. In J.-C. Simon & S. Impedovo (Ed.), *From Pixels to Features III* (pp. 313-325), Amsterdam: North-Holland.

SCHOMAKER, L.R.B., & TEULINGS, H.-L. (1990). A Handwriting Recognition System based on the Properties and Architectures of the Human Motor System. *Proceedings of the International Workshop on Frontiers in Handwriting Recognition (IWFHR).* (pp. 195-211). Montreal: CENPARMI Concordia.

TEULINGS, H.-L., & SCHOMAKER, L.R.B. (1992). Un-supervised learning of prototype allographs in cursive script recognition using invariant handwriting features. In J.-C. Simon & S. Impedovo (Ed.), *From Pixels to Features III* (pp. 61-73), Amsterdam: North-Holland.

TEULINGS, H.-L., SCHOMAKER, L.R.B., MORASSO, P., & THOMASSEN, A.J.W.M. (1987). Handwriting-analysis system. In R. Plamondon, C.Y. Suen, J.-G. Deschênes, & G. Poulin (Eds.), *Proceedings of the Third International Symposium on Handwriting and Computer Applications* (pp. 181-183). Montreal: Ecole Polytechnique.

TEULINGS, H.L., SCHOMAKER, L.R.B., GERRITSEN, J., DREXLER, H., & ALBERS, M. (1990). An on-line handwriting-recognition system based on unreliable modules. In R. Plamondon, & G. Leedham (Eds.), *Computer Processing of Handwriting* (pp. 167-185). Singapore: World Scientific.

THOMASSEN, A.J.W.M., TEULINGS, H.-L., & SCHOMAKER, L.R.B. (1988). Real-time processing of cursive writing and sketched graphics. In G.C. van der Veer & G. Mulder (Eds.), *Human-Computer Interaction: Psychonomic Aspects* (pp. 334-352). New York: Springer.