Adaptive Recognition of On-line Connected-cursive Script for use in Pen-based Notebook Computers *

Lambert Schomaker,
Hans-Leo Teulings,
Gerben Abbink,
Eric Helsper

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

In this demonstration a number of programs are shown, running on a Sparcstation: *PenBlock, VHS*, and *LabelWord*.

1 PenBlock$^{(Schomaker)}$

The *PenBlock* program is a simple application example, featuring simple block-print and gesture recognition. This Pen-based User Interface (PUI) actually serves as the front end to a Cursive Script Recognition (CSR) server *Cursive* which is running in the background. It is a typical demo program, not designed to match or improve the PenWindows or Penpoint PUI's, but to have an experimentation platform for the development of the cursive recognition system.

Penblock displays some basic functions in pen-based interfacing that are relevant to the editing of a screen-full of text. This text is entered in the form of block print, virtual keyboarding, or cursive words. Commands are executed on the basis of gestures or button clicking with the pen. The character and gesture recognition is based on a simple template matching scheme (nearest neighbour) and tuned to the writer on the basis of the Kohonen LVQ approach. Both Gestures and Handprint are assumed to be single-pendown traces. Thus, writing a capital E with penups is not handled in the current version. If a cursive word is written in a window of the working surface, it is sent to the Cursive Script Recognition Server (CSR). PenBlock waits until a recognized result file appears. The most likely word is automatically entered in the text. If it is wrong, the user may enter a gesture to bring up a pop-up menu with a list of possible other words. The screen is divided into three parts, from top to bottom: Text, Cursive, Menu. Occasionally dialog boxes appear, where a single click is expected from the user. For instance, if a cursive word is not recognized correctly at all, the user may enter the correct word in block print or by virtual keyboard clicking and train *Cursive* through entering a gesture. Penblock runs on VAX/VMS with the VWS window system, on SUN Sparc/X11 and on Ultrix/X11.

2 Cursive recognition: VHS
$(Schomaker\&Teulings)$

The program *VHS* was designed within Esprit Project 5204 "Papyrus" (Teulings et al., 1990; Schomaker & Teulings, 1990; Schomaker, 1993). The development of this system is based on knowledge of the human-movement aspects of handwriting. The signal processing, segmentation, and normalization modules are based on empirical findings collected within the Nijmegen handwriting research group (Thomassen et al., 1984; Kao et al., 1986; Plamondon et al. 1989), also within an Esprit project (P419/IMU).

The recognizer is designed as a monolithic Cursive Script Recognition (CSR) server, accept-

*Demonstrations presented at the IWFHR III, CEDAR, SUNY Buffalo, USA, May 25-27, 1993.

ing packets with raw XYZ coordinate time functions of handwritten words, and producing output packets containing an ASCII list of words with recognition quality values, sorted in descending order. In a practical situation, the user will see the graphical user interface of his own computer, e.g., PenWindows on a PC, or a pen interface running under the X Window System, and will not be aware of the fact that some of the necessary computations are performed on a special board or on a different computer. The functional distinction between the user interface and recognition functions is a necessity, given the complexity of the system as a whole. Through a data link, the CSR server performs the computations needed for cursive script recognition. This frees the Pen User-Interface (PUI) computer, which is mostly already heavily loaded with I/O interrupts and cpu- and memory-intensive work, from the demanding cursive-script recognition task.

Table 1 gives an overview on the basic steps in cursive script recognition system developed at the NICI.

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:

Stages 6-8 are typical for the current system. The feature vector quantization (6) is performed by using a Kohonen's self-organizing feature map for single velocity-based strokes. This network is trained off-line. The Kohonen cells are labeled with a stroke code during training. In the letter hypothesization (7), sequences of stroke codes are matched to create a solution space of letters. In word hypothesization (8) the letter graph is matched with words in an existing lexicon. For word existence testing, simple binary search is used. For large dictionaries, hashcoding is used for further speed-up. This approach allows for the use of plain ASCII word lists which are easily extensible by the user. Word recognition is organized in a stepwise approach. First, a Personal Lexicon is accessed.

Table 1. Basic steps in the VHS cursive script recognition server

0	(*PenBlock* Collecting pen-tip displacement signals)
1	*VHS* Receiving a raw data packet with xy coordinates
2	Low-pass filtering
3	Differentiation to obtain velocity
4	Geometric normalization (slant, size)
5	Segmentation into velocity-based strokes, white spaces, dots and t-bar crossings
6	Stroke feature vector extraction 9 angular, 3 Cartesian, 2 structural features
7	Stroke classification: Kohonen SOM, $O[n]$
8	Letter classification: Stroke transition network $O[n^2]$
9	Word classification: recursive tree traversal, binary word search $O[x^n]$
10	Sending a list-of-words packet

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

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

This lexicon is organized on the basis of global word contour features. If a word is not found in this lexicon with enough confidence, a large General Lexicon is used, such as the 25k word "Berkeley" list found in many Unix systems. Recognition rates vary from 50-90% first word correct recognition, depending on the amount of training and on the handwriting style used. The system is optimized for completely connected cursive lowercase words. A small number of cursive capitals have been presented to the system in training.

3 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. Sampling frequency was 125 Hz, resolution 0.02 mm/bit. 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.

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 in the test. Words were recorded using a Wacom HD-648A LCD-integrated digitizing tablet, using a Pen Windows data collection application. Sampling frequency was 50-80 Hz, tablet resolution was 0.05 mm/bit. The first half of a set was used for training, the second half was used as test set.

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 [a]. 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.

[a] With special thanks to Olivetti, Naples, and Captec, Dublin, who kindly provided the data within the framework of Esprit project P5204 Papyrus.

4 Results

Table 2 shows untrained and trained recognition results *. Looking at the Topword recognized column in Table 2, 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 upto 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. With respect to lexicon size, the following observations were done. Going from 3k words to 50k words, there can be a 10% decrease in recognition for the Topword correct, whereas the decrease is much smaller ($< 5\%$) for the Top-5 and Top-10 recognition rates. 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%).

* Summary of results to be presented at the ICOHD'93, Paris

Table 3 gives the **Untrained** results from an independent group of Dutch writers, writing mainly **mixed cursive** and **handprint** words in lowercase. Data acquisition was done on a Wacom HD-648A LCD-integrated digitizing tablet, except for writer *nl3*, who wrote on a Wacom PLV100. Sampling frequency was 100 Hz, resolution 0.05 mm/bit. The difference between the Top-10 results and the Topword results show the potential for autonomous (Passive) training. Since the lexical matching is based on "hard" matches, adapting the probabilities of the underlying allographs will yield higher Topword recognition without requiring user labeling at the letter level: Only the correct ASCII representation of the word must be presented.

Writer	Topword	Top-5	Top-10	Nwords	Nlex
nl1	44	72	74	50	5k
nl2	44	60	62	50	5k
nl3	40	48	48	50	5k
nl4	46	62	64	50	5k
nl5	50	60	64	50	5k

Table 3. Recognition rate in % words, for **Untrained** Dutch (nl) writers, mainly **mixed cursive** and **handprint**. Words were written in lowercase. All individual letters must be classified correctly by the system, for a match to occur.

On average, these recognition results are higher than in the comparable **Untrained** condition of the international test in Table 2 (45% vs 29%). It is not certain whether this result is due to international style differences or to the more reliable and slightly higher sampling rate of 100 Hz in the Dutch test set. Future work is directed at cursive uppercase allographs, at improved lexical postprocessing and at identifying writers in order to prevent an inefficient use of non-relevant allographs.

5 LabelWord$^{(Helsper)}$

The *LabelWord* program is invoked through the PUI training gesture *"ok"* in *PenBlock* if the labeling of individual letters in a word is required. The program displays the current word, together with the stroke segmentation according to the CSR. The user enlarges or reduces a piece of cursive handwriting until it has a number of strokes exactly corresponding to a legible letter. Most letters can be separated into a "kernel" and into variants with a number of initial or final ligatures. After training, the correct Kohonen cells are labeled with the stroke codes, and an entry in the word lexicon is generated.

6 References

Hollerbach, J.M. (1981). An oscillation theory of handwriting. *Biological Cybernetics, 39*, 139-156.

Kao, H.S.R., Van Galen, G.P., & Hoosain, R. (Eds.). (1986). *Graphonomics: Contemporary Research in Handwriting.* Amsterdam: North Holland.

Plamondon, R., Suen, C.Y. & Simner M.L. (Eds.). (1989). *Computer Recognition and Human Production of Handwriting* Singapore: World Scientific.

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. (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., 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., Keuss, P.J.G., & Van Galen, G.P. (Eds.). (1984). *Motor aspects of handwriting: Approaches to movement in graphic behavior.* Amsterdam: North-Holland.