Recognizing Sequences of Letters in Mixed-Script Handwriting

Caroline Barrière and Réjean Plamondon

Laboratoire Scribens
Département de Génie Electrique et Génie Informatique
École Polytechnique, Montréal, CANADA, H3C 3A7

Abstract

A new approach to mixed-script recognition is explored. Words are inspected through a fixed size window and sequences of letters are identified by slowly moving the window on the word. From the sequences of letters found, the word length can be estimated, and letter candidates for each position within the word can be proposed. Six writers participated in an experiment. Each wrote a training data set of 250 words and a test data set of 275 words. The recognition of each sequence of letters seen through a window is performed by a correlation with the patterns from the training set kept in memory. Results are presented on estimation of word length, and on letter recognition rates.

1. Introduction

Handwriting recognition is a very broad field in which different approaches are followed to solve various types of problems. Some applications need an optical approach, like recognition of addresses on envelopes, or amounts of money on bank checks. Others, like those based on an electronic tablet for data or text entry, need an on-line solution. This paper focuses on this second type of problems.

An electronic pen-pad is actually in development at Laboratoire Scribens (Plamondon 1991d). Different aspects are being considered: editing (Guerfali 1991), capital letter recognition (Nouboud 1991), writer-independent cursive script recognition (Parizeau and Plamondon 1991) and writer-dependent mixed-script recognition (Barrière 1991).

A person taking notes on an electronic pen-pad will want to write in a natural style. Many people don't write pure cursive script or pure discrete letters, they use a mixture of both. This paper describes part of the mixed-script recognition system, which globally aims at recognizing natural mixed-script handwriting in an on-line writer-dependent environment. This system will realize all the following steps: acquisition of a word, segmentation of the word in strokes, recognition of sequences of letters in the word with a correlation method, estimation of the word length, proposition of letter candidates for each position within the word, generation of pseudo-words, dictionary validation of the pseudo-words.

The work actually completed and tested is presented here in detail, and the other topics in development are briefly introduced. Therefore, the following sections explain the different steps leading to the proposition of letter candidates for each position within a word, using the correlation method for recognition. A testing protocol is described and results are reported in the last section.

2. Acquisition

The acquisition of words is done on an electromagnetic digitizer (Penpad 310, Pencept Corporation). The writing surface has an area of 11 inches x 11 inches (27.94 cm x 27.94 cm), a thickness of 0.5 inch (1.27 cm) and a maximum resolution of 1000 points per inch (394 points/cm). When the pen is within 0.5 inch (1.27 cm) of the tablet, coordinates of the pen tip are sent to the host computer at a frequency of 100 points per second. Information on the pen status (up or down) is also sent. The data transmission is done via a RS-232C link at 9600 bauds.

The xy-coordinates of the pen tip and the pen status for the total word acquisition period are kept in a file. A low-pass filter with a 10 Hz cut-off frequency is used to reduce noise. Studies on the hand muscular constraints have shown that the peak in human writing frequency is around 5 Hz, and that the whole spectrum rarely exceeds 10 Hz (Teulings and Maarse 1984). From those results, it has been assumed that a 10 Hz filtering would reduce the noise introduced by the acquisition system without losing important information on the signal.

3. Segmentation

Handprinted letters are always composed of one or more components. A component is defined as the tracing between a pen status of down to up (Plamondon and Maarse 1989). However, in cursive or mixed script, components can't be considered as the basic writing elements composing letters. A component sometimes covers one letter, two-an-a-half letters, or even the entire word. Other points have to be inserted to segment the components and find the letters.

According to a recent handwriting segmentation model (Plamondon 1991a), each of these components is made up of strings that are portions of the trace between two angular discontinuities or between the beginning (or the end) of a component and the next (or previous) angular discontinuity. Each string is made up of a combination of curvilinear and angular strokes, that is curvilinear and angular displacements characterized by log-normal velocity profiles. In this general context, a curvilinear (or angular) stroke is thus defined as a portion of the pentip trajectory that corresponds to the curvilinear (or angular) displacement resulting from the production of a support bounded log-normal velocity profile, produced by a specific generator as a response to a specific impulse function fed into it. (Plamondon 1991b, 1991c)

One major conclusion arising from this model is that strokes must be superimposed to generate fluent handwriting. This is in accordance with a basic psychophysical phenomenon often reported in motor control: the handwriting generation process, like many other types of movements, is not exclusively sequential but very often advanced preparation of the forthcoming stroke is done in parallel with the execution of the actual stroke. Such an observation is reflected by the necessity of superimposing partially a few log-normal profiles to regenerate the velocity patterns of a handwritten word. In other words, the basic curvilinear and angular strokes are hidden in the signal. The only way to extract them perfectly is to perform an analysis by synthesis experiment, with the help of the proper impulse response for each stroke.

This segmentation framework is very general, in a sense that the model can be used to interpret and analyze different segmentation methods proposed previously by many research groups. Indeed, most of these methods can be viewed as using approximate operators that estimate roughly the location of the hidden strokes. For example, Enrich and Koehler (1975) use the minima of the y-coordinate curve, focusing on non-overlapping curvilinear strokes. Kurtzbert and Tappert (1985) segment in the minima of the writing density projected on the x-axis, roughly estimating angular strokes. Schomaker, Thomassen and Teulings (1989) use the zero-crossings of the y-velocity curve and x-velocity curve, estimating respectively curvilinear and angular strokes. None of these approaches take overlapping into consideration.

A few of these approximate segmentation operators have been studied here: the minima of the absolute velocity curve and the minima of the angular velocity curve (Plamondon 1989), the minima of the y-coordinate curve (Enrich and Koehler 1975), and the extrema of the y-coordinate curve (same as the zero-crossings in the y-velocity curve used in Schomaker, Thomassen, Teulings 1989).

The stability of segmentation will influence the recognition process. One criterion for evaluating this stability is to count the number of segment combinations that arise from segmenting a set of letters with a specific segmentation operator (see Barrière 1991). According to the criterion used, one can see that the operator finding the extrema of the y-coordinate curve has a better stability. This method was chosen to segment each word for the rest of this study. In the context of the segmentation framework proposed by Plamondon (1991a), this choice is equivalent to focusing the analysis on an operator that roughly estimates the position of curvilinear strokes, only without directly taking into account the strokes superimposition.

4. Segment characterization

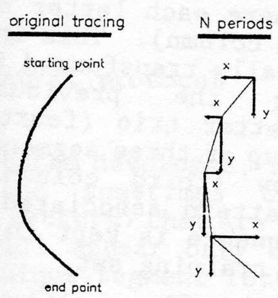

Figure 1. Segment characterization.

Once the segments are obtained, they are represented by a fixed number of elements. The total writing time T of a segment is divided into N equal time periods. A segment is characterized by the N xy-displacements of the tracing done in a time T/N. Figure 1 shows the representation used, with N=4.

This description of segments was inspired by the work of Morasso (1989). This author uses a back-propagation neural network (Rumelhart, Hinton and Williams 1986) for digram (2 successive letters) recognition in cursive script.

These networks only work with pattern of fixed size as input. Since we plan, in a near future, to compare neural networks to the correlation method described here for letter recognition, it was decided to use a similar representation for the segments.

5. Moving Window

A word is thus segmented at the extrema of the y-coordinate curve, with each segment characterized by 4 xy-displacements. The next step consists of designing a recognition model that transforms the stream of segments into a succession of letters. The segment-letter relation is not obvious. Different letters are composed of different numbers of segments. For example, an "a" can be formed of 3 segments and an "m" of 6 segments. Therefore, a window used to look at letters in a word must have an adjustable size. On one hand, it is impossible to know the window size without recognizing the letter in it. On the other hand, to recognize an "a" or an "m", it would be useful to know in advance how many segments to join together for adjusting the size of the window before making an attempt to identify the letter in it. So, to divide a word into characters, the characters should be recognized, and to recognize the characters, they should be well surrounded!

A way to consider this chicken and egg problem from a different angle is to look at a word through a window composed of a fixed number F of segments. The window scans the word, moving one segment at a time, and the letters partially or entirely covered by the window are identified at each step.

For a window with F=3, the letters partially covered by the segments 1-2-3 are identified, and then the ones covered by the segments 2-3-4, 3-4-5, and similarly to the end of the word. This overlapping process can be viewed as an operational way to take into account the superimposition of strokes as described in the model proposed by Plamondon (1991a). A character is composed of a minimum of one segment, and therefore, a window of F segments can cover a maximum of F characters.

Humans are used to seeing letters in sequences. In any language, some sequences are more frequent than others. Looking at the previous and following letters generally augments the recognition of the present letter. The system takes this context effect into account as follows: a letter is identified at the same time as the previous letter is

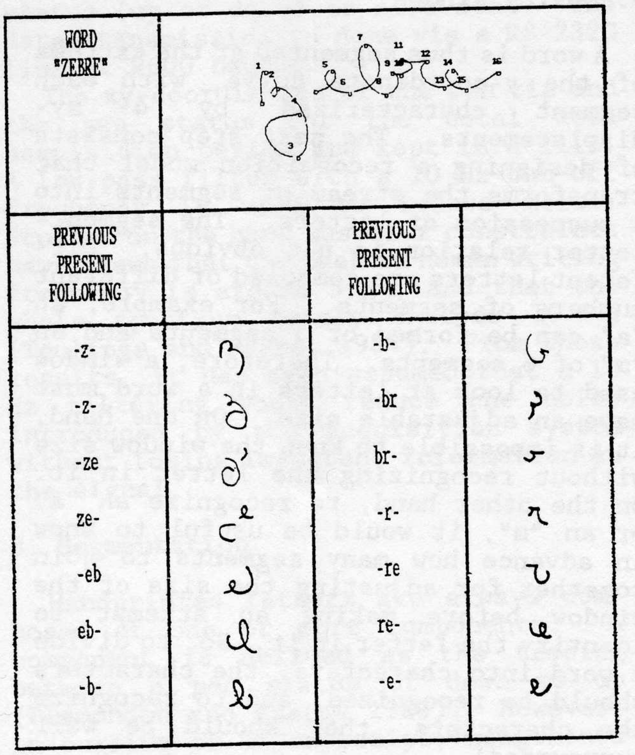

Figure 2. 3-letter sequences identified by a human.

confirmed and the following letter is predicted. A window of 3 segments is appropriate to describe that process of confirmation, identification and prediction, while moving on the word. This is also in accordance with the recent results of Schomaker (1991) where it has been found that multiple stroke-based script recognition is better than character-based recognition provided that three strokes are considered at a time.

Figure 2 shows the 3-letter sequences identified by a human for the word "ZEBRE". The most covered letter is always the present letter. If the two last segments of the "z" and first of the "e" are covered, the previous letter is non-existent "-", the present letter is the "z" and the following letter is the "e". If the window covers the first two segments of the "b" and the last of the "e", than the "b" is the present letter, the "e" is the previous one, and the following one is non-existent "-".

6. Training data

Table 1. Beginning and end of letters.

WORD OUTPUT ON SCREEN	BEGINNING AND END OF LETTERS ENTERED BY HUMAN TUTOR	3-SEGMENT PATTERN	3-LETTER SEQUENCE ASSOCIATED
	u 1 4	1-2-3	-u-
	l 5 6	2-3-4	-u-
	c 7 8	3-4-5	-ul
	e 9 10	4-5-6	ul-
	r 11 12	6-7-8	-l-
	e 13 14	7-8-9	-lc
	u 15 18	...	...
	x 19 22		

As seen in Table 1, each segmented word is output on the screen (first column) and a human tutor has to determine where each letter begins and ends (second column). This information is automatically transformed in classes representing the previous-present-following letter trio (fourth column) for each group of three segments covered by a window (third column). Each 3-segment pattern associated with its 3-letter sequence is kept in memory as part of the training set.

7. Recognition process

A correlation method is used, as a similarity measure, to classify a new pattern. The correlation is done with all the patterns in the training set, and the class associated to the pattern with highest correlation wins. If P_J is one of the J patterns kept in memory, and T_k one of the K test patterns, the correlation at a fixed point is given by

$$R(0)_{T_k * P_j} = const \sum_{i=1}^{24} T_k(i) P_j(i)$$

if the patterns are normalized with respect to their total length, and the constant is fixed to 1, this correlation will give a number between -1 and +1 that can be used to find the winning class. Moreover, the pattern normalization involves that the similarity measure becomes proportional to the squared distance measure.

The patterns to classify have been previously defined has 3 consecutive segments, each segment being represented by 4 xy-displacements. Then, all K patterns T_k from the test set and J patterns P_j from the training set are vectors containing 24 elements.

Each element i of a normalized pattern TN_k or PN_j is obtained by the following equation:

$$PN_j(i) = \frac{P_j(i)}{\sqrt{\sum_{i=1}^{24} P_j(i)^2}}$$

All J 3-segment training patterns are kept normalized in memory, and K 3-segment test patterns are normalized before being compared to the training patterns.

8. Experimental protocol

In the system developed, it is mostly digrams, and sometimes trigrams (3 successive letters) that are recognized. Trigrams will appear only if a window covers a unique-segment letter and two letter transitions. The recognition of a test pattern depends on the patterns learned. It is then important that the training set contains models of everything that the system will have to recognize in proportions similar to what it will be exposed. For example, learning many variations of "zy" is not appropriate when the system may have to recognize this pattern only once compared to a thousand of "ti" appearing in different contexts.

A non-optimal polynomial selection algorithm has been developed to choose a subset of words that is representative of the proportions of digrams in the set of words to recognize. This set of words, for the experiment, has been defined as the French dictionary "Larousse de Poche" (Librairie Larousse 1990) containing approximately 30000 common nouns. The real problem is in fact to find a combination of X (X = 100, 200, ...) words in 30000 words respecting C criteria. It is uncertain that a solution exists, even by trying all the possibilities.

The steps for generating a subset are stated hereafter:

1) Reduce to N/1, the ratio in the real set of the most frequent digram to the least frequent (7715/1), and keep the new proportions in a table.
2) Browse through the dictionary in an order that respects the word length proportions, and for each word length, in a random order based on the first letter in the word.
3) Keep a word as part of the subset, only if all of its digrams are available (quantity > 0) in the table.
4) Decrement by 1 in the table all the digrams quantity of a chosen word.

The algorithm was run two times, with reduced proportions fixed to 100/1. It generated a training set of 250 words and a test set of 275 words.

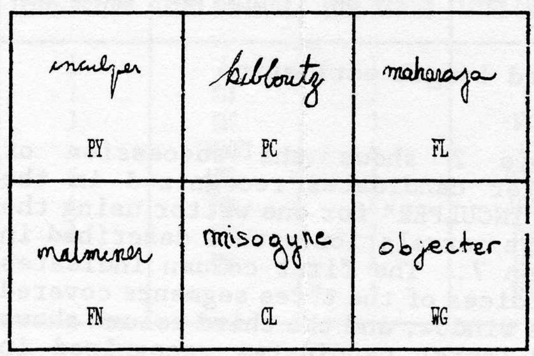

Figure 3. Writing examples.

Six writers from the Scribens laboratory (Ecole Polytechnique) participated in the experiment. The training set has been divided into 5 training subsets of 50 words each, and the test set into 5 test subsets of 55 words each. Each writer enters at most one subset a day, which takes 10 to 20 minutes to write. The entire acquisition process has been done over a month. This long period favors fluctuations in the persons writing, giving more realistic training and test sets.

Participants have been asked to write in a natural way, with a normal size and speed, using lower-case letters. The only constraint imposed was that no component can be added to a letter after another letter has been started. The order of the strokes is very important here to be able to analyze the succession of segments composing a word. Therefore, the points on "i" and bars on "t" had to be put right after the writing of the letter, or not put at all. Figure 3 shows typical writing examples of the 6 participants.

All words have to go through the steps of acquisition, segmentation, segment characterization. Section 6 described the following steps for the training patterns to keep them in memory. The next steps for the 275 test words are the recognition of sequences of letters through a moving window by a correlation method, the estimation of word length using the succession of 3-letter sequences found in the word and the proposition of letter candidates for each position within the word. The following sections describe the two last steps.

9. Word length estimation

Table 2 shows the succession of 3-letter candidates recognized in the word "INCULPER" for one writer using the highest correlation method described in section 7. The first column indicates the indices of the three segments covered by the window, and the third column shows the 3-letter candidates recognized in the 3-segment pattern. More than one candidate is given when different patterns in memory have the same correlation with the test pattern (within 5%).

As one can see, a post-processing is necessary for the reconstruction of a word from the succession of sequences recognized. The first step consists of finding where the letters are in the word, and consequently evaluating the length of the word. This is possible by following the basic considerations that inspired the initial design of the system. A letter should be announced as the following letter at a sequence Q, stabilize as the present letter from sequence Q+1 to sequence Q+R (with R being the number of segments in the letter) and then be confirmed as the previous letter at sequence Q+R+1.

Table 2. Sequencing of letters.

SEGMENTS COVERED BY THE WINDOW	DESIRED PATTERN	3-LETTER CANDIDATES	POSITION OF LETTERS	SEQUENCING OF LETTERS
01-02-03	-i-	-i- -i-	-*-	-1-
02-03-04	-in	-ib	-**	-12
03-04-05	in-	-im in- in-	i*i	-2-
04-05-06	-n-	-n-	-*-	-2-
05-06-07	-nc	-nc	-**	-23
06-07-08	nc-	nc-	**-	23-
07-08-09	-cu	-a- -a- -ca	-*i	-3-
08-09-10	-u-	-u-	-*-	-3-
09-10-11	-u-	ad-	**-	34-
10-11-12	-ul	-e- -us	-*i	-4-
11-12-13	ul-	ul- ar- ol- fl- tl-	**-	45-
12-13-14	-l-	-lv	-**	-56
13-14-15	-lp	ub-	**-	56-
14-15-16	lp-	ep-	**-	56-
15-16-17	-p-	-p- -p- -p-	-*-	-6-
16-17-18	-p-	-p-	-*-	-6-
17-18-19	-p-	-p- -p- -b-	-*-	-6-
18-19-20	-pe	-pe -pa	-**	-67
19-20-21	pe-	pa-	**-	67-
20-21-22	-e-	-e- -e- -e- -e- -e-	-*-	-7-
21-22-23	-e-	-e- -e- -e-	-*-	-7-
22-23-24	-er	-er -er -em -en	-**	-78
23-24-25	er-	er- er-	**-	78-
24-25-26	-r-	-r- -r-	-*-	-8-

The second column of Table 2 shows the perfect recognition for the word "INCULPER", where each letter can be seen travelling from the following column to the preceding column.

For each sequence, for each column, the proportion P of 3-letter candidates containing a letter in that column is evaluated. A threshold [T1] is fixed on P above which a letter is considered certain in that column. A second threshold [T2] is fixed on P above which a letter is considered possible in that column. Table 2 shows an example for the word "INCULPER" where the thresholds [T1,T2] are fixed at [1.00, 0.33]. The fourth column of Table 3 shows where the letters can possibly or certainly be ("-" : non-existent letter, "*" : certain letter, "i" : possible letter). The fifth column indicates the index of the letter found at each position.

Let us take the pattern formed from segments 7-8-9 in Table 2 as an example. None of the three candidates has a letter in the first column (-,-,-) , so at the corresponding position, a non-existent letter "-" is put. The three candidates have a letter in the second column (a,a,c), the proportion is than 1.00 and a certain letter "*" is put at the corresponding position. Only one candidate has a letter (-,-,a) in the third column, the proportion is than 0.33, and a possible letter "i" is put at the corresponding position.

The second step consists of determining where each new letter starts. The following rules have been applied to generate the fifth column of Table 3 from the fourth column.

1. For the first sequence, each "*" corresponds to a new letter.
2. A "*" in the third column corresponds to a new letter if no "*" appears in the third column of the previous sequence.
3. A "*" in the second column corresponds to a new letter if a "*" is in the first column and no "*" is in the first or third column of the previous sequence.

As one can see, the sequencing of letters showed in the fifth column of Table 2 gives a word length estimation at the same time.

10. Proposition of Letter candidates

Now that the sequencing of letters has been found, the enumeration of letter candidates for each position within the word is possible. For each new position, a list of the letters present at that position is kept. The letters in the list are sorted in descending order of their frequency of occurrences at each position. A letter can be present in many candidates for one sequence, and also in many successive sequences. A score is given to each letter candidate, by adding the proportions from each sequence where it is present and by multiplying that number by the number of sequences where it appears. Using the results from Table 2 in the previous section, for the word "INCULPER", the calculated scores are shown in Table 3.

Table 3. Scores for letter candidates.

LETTER POSITION	LETTER CANDIDATES	PROPORTION BY SEQUENCE	TOTAL OF PROPORTIONS	TOTAL SCORE
1	I	1- 1.00 2- 1.00	2.00	4.00
2	N	3- 0.67 4- 1.00 5- 1.00 6- 1.00	3.67	14.68
	B	2- 1.00	1.00	1.00
	I	3- 0.33	0.33	0.33
...				

Table 4. Examples of configurations kept.

CONFIGURATION 1 WORD LENGTH = 8		CONFIGURATION 2 WORD LENGTH = 9	
LETTER POSITION	LETTER CANDIDATES	LETTER POSITION	LETTER CANDIDATES
1	I	1	I
2	NIB	2	N
3	CAU	3	BIC
4	UDEAOFT	4	AU
5	LUER	5	DEFOT
6	PBV	6	LUER
7	EAC	7	PBV
8	RMN	8	EAC
		9	RMN

The highest letter score at each position are summed up and the total is divided up by the number of letters in the word, to give the total score of a word. The configuration of the ten highest scoring word candidates are kept in memory. Then the process (estimation of the number of letters, positioning of letters in the word, and generation of letter candidates for each position within the word) is repeated many times, changing one at a time each probable letter to a certain letter (changing an "i" to a "*" in Table 2). Each change is kept if it gives a better word score than the smallest of the ten kept. The process is then entirely repeated by modifying the thresholds [X1,Y1] and the same rule is applied to keep any new configuration in the table. Table 4 shows examples of configurations kept.

11. Results

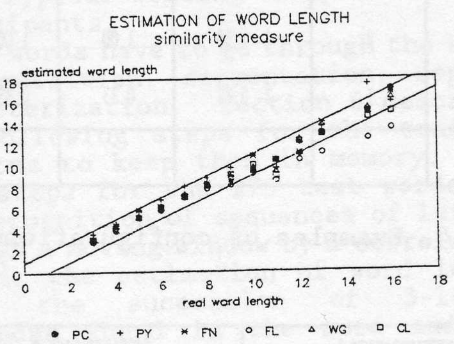

Figure 4. Estimated word lengths.

Table 5. Dictionary search limits.

ESTIMATED WORD LENGTH (WL)	WL < 10	WL < 10
DICTIONARY SEARCH LIMITS	WL-2 TO WL+2	WL-2 TO WL+5

The transformation process applied to the test words includes the following steps: acquisition, segmentation, segment characterization, recognition of 3-letter sequences seen through a moving window by a correlation with training patterns kept in memory, word length estimation by letter positioning and proposition of letter candidates for each position. Results are presented here on the two last transformation steps.

Figure 4 shows the estimated word lengths, for each of the 6 writers [WG, CL, PY, PC, FL, FN], versus the real word lengths. Estimation error is near one letter for words of length less than 10. The estimator is somewhat less stable for longer words, and often underestimates the number of letters. This estimation of the word length will certainly be useful to limit a subsequent dictionary search, as seen in Table 5.

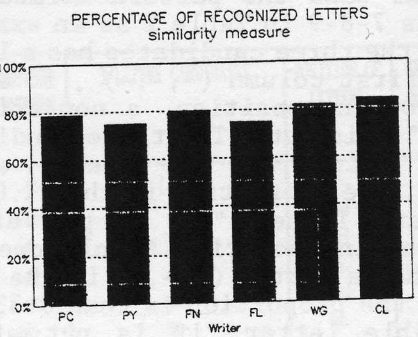

Figure 5. Percentage of recognized letters.

Table 6. Global statistics on letters.

% RECOGNIZED LETTERS	% DELETIONS	% SUBSTITUTIONS	% INSERTIONS
79.0%	10.0%	11.0%	5.5%

Figure 5 shows the percentage of recognized letters, which varies between 75% and 84% depending on the writer.

A letter at position N in a test word is considered recognized if it appears in the letter candidates for position N-1, N or N+1. If the letter is found at position N-1, and the previous letter was not found at position N-1, a deletion is added and letter candidates for position N-1 become candidates for position N. If the letter appears at position N+1, an insertion is added, and letter candidates for position N+1 become candidates for position N. If the letter does not appear at position N, and the next letter appears at position N+1, than a substitution is added. Table 6 shows percentages averaged over the 6 writers for recognized letters, letter insertions, deletions and substitutions.

12. Conclusion

A writer-dependent system for recognizing letters in mixed-script handwriting has been presented. The system performs word acquisition, word segmentation, segment representation and recognition of 3-letter sequences from 3-segment patterns seen through a moving window. Recognition is done taking the highest correlation with the patterns from a training set. The system also realizes an estimation of word length, and proposition of letter candidates for each position within the word. The estimated word length is quite accurate, and would be very useful to limit a subsequent dictionary search. To be consistent, the dictionary used for the search should be the same as the one used to generate the data and test sets. The average recognition rate for letters is 79%. The misrecognized letters could be corrected by a dictionary search to achieve better recognition rates at the word level. This process is actually under study.

13. References

BARRIERE, C., (1991) "Exploration d'une approche par réseaux neuronaux pour la reconnaissance de symboles manuscrits", M.Sc.A. Thesis, École Polytechnique de Montréal, December 1991

EHRICH, R.W. And KOEHLER, K.J., (1975) "Experiments in the Contextual Recognition of Cursive Script", IEEE Trans. on Computers, vol. 24 #2, Feb. 1975, pp. 182-194

GUERFALI, W. and PLAMONDON, R., (1991) "An Editing Tool for Handwritten Text", Proc. 5th IGS Conference, Tempe, Arizona, pp. 151-153

KURTZBERG, J.M. and TAPPERT, C.C., (1985) "Handwriting Recognition by Substroke Segmentation and Composition", IBM Tech. Disclosure Bull., vol. 27 #9, Feb. 1985, pp. 5167-5170

LIBRAIRIE LAROUSSE, (1990) "Larousse de Poche, dictionnaire des noms communs et des noms propres", ed. Les Editions Françaises Inc

MORASSO, P., (1989) "Neural Models of Cursive Script Handwriting", Proc. Int. Joint Conf. on Neural Network, vol. II, pp. 539-542

NOUBOUD, F. and PLAMONDON, R., (1991) "A structural Approach to On-line Character Recognition: System Design and Applications", Int. Journal of Pattern Recognition and Artificial Intelligence, vol. 5, no. 1-2

PARIZEAU, M. and PLAMONDON, R. (1992) "A Handwriting Model for Syntactic Recognition of Cursive Script", accepted 11th Int. Conf. Pattern Recognition, The Hague, June 1992

PLAMONDON, R., (1989) "A Handwriting Model Based on Differential Geometry", Computer and Human Applications of Handwriting, R. Plamondon, C.Y. Suen et M.L. Simmer eds., World Scientific Publishing, Singapore, pp. 179-192

PLAMONDON, R. And MAARSE, F.S., (1989) "An Evaluation of Motor Models of Handwriting", Proc. IEEE Trans. on Systems, Man and Cybernetics, vol. 19, no. 5, pp. 1060-1072

PLAMONDON, R., (1991a) "A Model Based Segmentation Framework for Computer Processing of Handwriting", accepted 11th Int. Conf. Pattern Recognition, The Hague, June 1992

PLAMONDON, R., (1991b) "A theory of Rapid Movements", Tutorials in Motor Behavior II, G.E. Stelmach, J. Requin (Eds), North Holland, pp. 283-295

PLAMONDON, R., (1991c) "What Does Velocity Information Reveal About Handwriting Generation", Proc. 5th Handwriting Conference of IGS, pp. 9-10

PLAMONDON, R., (1991d) "Step Toward the Production of an Electronic Pen-Pad", Proc. of ICDAR 91, pp. 361-371

RUMELHART, D.E. and HINTON, G.E. and WILLIAMS, R.J., (1986) "Learning Internal Representations by Error Propagation", Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1., D.E. Rumelhart and J.L. McClelland (Eds), Cambridge, MIT Press, pp. 318-362

SCHOMAKER, L.R.B. and THOMASSEN, A.J.W.M. and TEULINGS, H.-L., (1989) "A Computational Model of Cursive Handwriting", Computer Recognition and Human Production of Handwriting, Eds. R. Plamondon, C.Y. Suen and M.L. Simmer, World Scientific Publ. Co., pp. 153-177

SCHOMAKER, L.R.B., (1991) "Stroke- versus Character-Based Recognition of On-line Connected Cursive Script", Proc. 5th Handwriting Conference, pp. 213-217

TEULINGS, H.L. and MAARSE, F.J., (1984) "Digital Recording and Processing of Handwriting Movements", Human Movement Science, vol. 3, pp. 193-217