Reverse handwriting: from ink to word

Angelo Marcelli
DIIIE – University of Salerno, Via Ponte don Melillo, 84084 Fisciano (SA), ITALY
amarcelli@unisa.it; http://nclab.diiie.unisa.it

Abstract: *We present a principled approach for cursive handwriting recognition which builds upon handwriting generation models. According to such models handwriting is a learned complex motoric task which is accomplished by sequencing simpler movement called stroke. As learning proceeds in human, so does fluency, which results in producing similar sequence of strokes in correspondence of the same sequence of letters. Such invariants represents therefore the basic drawing units to which an interpretation can be associated. Recognition is then achieved by detecting the invariants used to produce the word to be recognized, associating to them their interpretations, and eventually concatenating the interpretations along the ink of the word. Experiments on on-line data of the current implementation are reported and discussed to show the effectiveness of the method.*

Keywords**:** handwriting recognition, handwriting generation, saliency, sequence matching.

1. INTRODUCTION

Studies on handwriting generation have shown that handwriting is produced through a perception/action cycle involving attentive vision, learning and movement [1,2]. The complex movements needed to generate handwriting result from concatenation of elementary movements, each aiming at a time-varying spatial target. Fluency, then, emerges when proper time superimposition of successive elementary movements is achieved through learning. Time superimposition of strokes result in anticipatory effect in the actual drawing, so that group of strokes with whom the writer is familiar with, we called *invariants*, are "embedded" into a single sequence, which is drawn without any feedback, as in case of "elementary" writing movements. Then, complex handwriting, such as character or cursive words results from spatial superimposition of the corresponding invariants. Invariants are produced by learning a complex motor task, therefore different individuals develop different invariants, even when the same models are used during the learning.

According to those findings, we have proposed to achieve cursive recognition by detecting the invariants embedded in the handwriting and then associating the invariant's interpretations to the unknown word. We assume that a set of words is available in electronic ink, that each word has been segmented into strokes, and that each stroke has been labeled with the character it belongs to in that word. Accordingly, each word of this set, called *reference set*, is associated with a string of as many symbols as the number of strokes presumably used to produce the handwriting. Since the invariants may correspond to part of a character, or a sequence of them, the same invariants may be embedded into different words, thus to a different string. Thus, many partial interpretations may be produced and the final interpretation is achieved by ranking the strings corresponding to the invariants extracted from the word, sorting them along the writing direction into a directed graph, where node correspond to the invariants and arcs to possible connections between successive invariants, and eventually searching for the best paths in the graph.

In the following, Section 2 illustrates the stroke segmentation, Section 3 the algorithm to perform ink matching and interpretation, and Section 4 describes how the interpretations are combined to provide the word corresponding to the unknown ink. Section 5 reports the result of experiments on the Unipen on-line dataset, while discussion of the results and concluding remarks are left to Section 6.

2. FROM INK TO STROKES

As mentioned in the Introduction, the first step of our approach is devoted to extract from the ink the strokes, i.e. the elementary movement used by the writer to produce the cursive. It has been shown that such elementary movements correspond to elementary shape [3, 4], and straight segments and arcs of circle have been widely used as the set of basic writing units, while the process of decomposing the input trace into a set of suitable shape primitives has been reformulated as a curve fitting problem, where segments and arcs of circle are the primitives to fit within the original curve [5]. This approach is very appealing because, by using the arclength representation of the points corresponding to the input trace, the curve fitting problem can be reduced to that of approximating a set of points by means of straight segments [6]. Many curve fitting strategies have been proposed in the literature, which have proved their effectiveness in a number of applications. Unfortunately, in case of on-line handwriting, these strategies do not allow to obtain satisfactory results. In fact, independently on the algorithm used to perform the fitting, the attainable decompositions generally exhibit a very large variability. This behaviour can be explained considering that the points are collected by the input device with uniform time sampling: this implies that changes in the writing speed, either due to noise or exhibited in correspondence of curvature variations, produce changes in the density of the points and local distortion along the line. This effect is particularly undesirable because curvature variations typically occur in regions where two successive strokes interact. As a consequence, the obtained decompositions result extremely sensitive to non significant shape variations of the trace that frequently occur in proximity of those interacting regions.

To tackle this problem, we have proposed a decomposition method which exploits the concept of saliency used to model attentive vision in primate visual system [7] and that proved to be much more invariant with respect to non-significant shape variations or changes in the writing speed. The method is based upon a multi-scale representation of the original curve, obtained through a frequency analysis of the discrete-time sequences *x(n)* and *y(n)*, representing the (x,y) coordinates of the points collected by the input device. To this aim, we compute the Discrete Fourier Transform *X(n)* and *Y(n)* of the sequences *x(n)* and *y(n)*, and then apply the Inverse Discrete Fourier Transform to the first *T* elements of the sequences *X(n)* and *Y(n)*: the smaller the value of *T*, the coarser the approximations of the curve, while the opposite is true for values of *T* close to *n*. In practice, at each scale, we obtain a smoothed version of the original curve containing a smaller number of points. This multi-scale representation is then used to build a saliency map to highlight the so called "focus of attention", i.e. the regions of the image representing salient information for the application at hand. In our case, those "focus of attention" are the points of the original curve in which significant curvature variations are recorded at different scales, and therefore represent the desired decomposition points [8]. Fig 1c) show the results of the segmentation on the word of fig. 1b). The final description of the handwriting shape is given in terms of a sequence of strings, each encoding the curvature changes relative to each stroke. To this aim, the actual values of the curvature are quantized into 16 intervals and each interval encoded by one of the letter of the subset [A-P] in such a way that the letter A corresponds to the first interval (from 0 to $2\pi/16$), the letter B to the second one (from $2\pi/16$ to $2*2\pi/16$) and so on. By this encoding, the shape of the word is described by a string of labels that encodes the local curvature of the selected smoothed representation of the original signal .

3. FROM STROKES TO INVARIANTS

As mentioned in the Introduction, generation models suggest that handwriting is composed of strokes that are drawn one after the other, and that the fluency emerges from the time superimposition of strokes. In other words, as the writer becomes familiar with a given word, he knows how long it takes to draw a stroke and where it will finish, so that the next stroke can be initiated before the current one is completed. As a consequence, group of strokes with whom the writer is familiar with are "embedded" into a single sequence, which is drawn without any feedback, as they were "elementary" writing movements. Thus, words are obtained by concatenating such handwriting *invariants*. Accordingly, cursive handwriting recognition can be achieved by detecting those invariants in a set of reference words, associating to each of them an *interpretation* (in terms of the ASCII code corresponding to the characters the invariant is meant to encode), matching the unknown words with the the reference words to extract the invariants and eventually concatenating the invariants ASCII code.

In order to extract the invariants, we aim at further exploiting saliency-based method in order to find similar pieces of ink between two cursive words. The rationale behind this choice is similar to the one behind our approach to segmentation: by evaluating the similarity at different scales and then combining this information across the scales, we expect that sequence of strokes that are "globally" more similar than other to stand out in the saliency map. The "global" nature of the saliency guarantees that its map provides more reliable estimation of ink similarity with respect to that provided by "local" criteria, as it is usually proposed in the literature. To implement such an approach we need to define a scale space, find a similarity measure to be adopted at each scale, compute the saliency map, and eventually select the matching pieces of ink. As with regards to the scale space, we adopt the number of strokes in the sequences whose similarity is being measured. Such a number will be referred in the following as the *length* of the sequence. Accordingly, the number of scales corresponds to the length *K* of the longest common sequence of strokes. To compute the value of *K*, some constraints derived from the different nature of the strokes are applied. Basically, those constraints are used to exclude from the longest common sequence strokes that may have similar shape but different semantic values. For instance, the shape of both the character "e" and the character "l" can be segmented in two strokes whose shapes are very similar, except for the size. An algorithm using information from the ink layout (zoning), shape and relative position of the strokes labels the strokes as ascender, descender, upper central, lower central and central. The labels of each pair of strokes are compared to check their compatibility, so that pairs of incompatible strokes are not included in any sequence. Thus, *K* represents the length of the longest common sequence of compatible strokes. Successive scales are obtained considering sequences made of *K-1*, *K-2*,…, *2* strokes. At the end of this stage, thus, we obtain *K-1* similarity maps, each of which contains as many elements as the number of longest common sequences of compatible strokes of the given length that can be extracted from the original ink. As similarity measure, we adopt a new string edit distance called Weighted Edit Distance (WED). WED is based on the concept of string stretching: it does not introduce or delete any symbol in the strings to compare, but simply extends, or stretches, the shortest string in such a way that each symbol of this string is compared with one or more symbols of the other, depending on the ratio *R* between the lengths of the two strings. The edit distance is then computed by summing the cost of substitution of each compared pair of symbols, weighted by a coefficient whose value depends on both the position of the symbols in the two strings, and the value of *R*. After the ink similarity is evaluated at each scale by computing the WED on the corresponding strings of symbols, we compute the saliency as it follows. At each scale *K*, the best matching pair of sequence, i.e. the pair whose WED is the highest, is selected and the strokes belonging to the corresponding sequences are assigned a saliency value $S_k = WED_K / K$. The saliency value S_{ij} for each pair of strokes is then obtained by adding up all the saliency values S_k for that pair of strokes for k= *2, 3, ..K*. Thus, the saliency map *S* for a pair of inks made of *N* and *M* strokes, respectively, assumes the shape of an *N*x*M* array, whose elements are either 0, in case of incompatible strokes, or S_{ij}. The saliency map is then thresholded, and the diagonal sequences of values S_{ij} greater than the threshold constitute the matching pieces of ink [9].

In order to associate to each invariants its interpretation, we use a set of words from a *reference* set.

Each reference word is segmented into strokes as described in Section 2, and each stroke is labeled with the character it belongs to in that word, so that it is associated with a string of as many symbols as the number of strokes the word is presumably made of. This is achieved by estimating the distribution of the number of strokes for each character class from a training set of words for whom the labels to be associated to each of its stroke have been manually entered. With such a piece of information we estimate the actual number of strokes for each character of the reference word by solving a MAP problem. Fig 1a) shows segmented and labeled samples from the reference set.

During the matching between the unknown word and each reference words, every time a match is found the labels associated to the matching strokes of the reference are assigned to the matching strokes of the unknown, and the *score S* of that sequence is incremented by one. Since each unknown is matched with all the references, it may happen that the same sequence of strokes of the unknown receives different labels, as well as that different sequences of strokes receive the same labels. This is not unusual, in that, due to shape variability in handwriting, pieces of inks similar to the one of the unknown may be associated to different labels, but also that different pieces of ink in different references are associated to the same labels. For instance, an open loop can be part of an a in one of the reference, correspond to an o in another one, or found in the group ce in a third one. When sets of sequences of strokes with the same label which partially overlap are found, we merge them and retain only the highest scoring one. For each overlapping sequence, we add to the score *S* of the highest ranking one(s) a merging factor *M* given by:

$$M = S*L/(b1-e2)$$

where *L* is the length of the overlapping part, and *b1* and *e2* represent the position of the first and last strokes of the overlapping pair, respectively. The rationale behind is that less frequent sequences are either associated with less frequent shapes or found because of small variations in the labeling of the segments endings, due to anticipatory effects or the presence of ligatures, which may results in a displacement of the string of symbols corresponding to a (group of) character. In both cases, overlapping sequences have a common part and differ only at the endings, thus only contribute to improve the occurrence of the common part in the most frequent ones. The set of partially overlapping sequences of symbols obtained at the end of the merging, shown in fig. 1d) represents the data from which the desired interpretation of the word is computed, as it is described in the next Section.

4. FROM INVARIANTS TO WORDS

Given the set of partially overlapping interpretations provided by the previous step, reading the word becomes finding the sequence of invariants that goes from (possibly) the beginning of the ink to (possibly) the end, under the constraint that the positions of the invariants along the ink is fixed.

While ordering the invariants along the ink is trivial in case of adjacent invariants, i.e. a pair of successive invariants with the ending of the leftmost immediately followed by the beginning of the rightmost, this decision may be troublesome in presence of overlapping invariants, in that this means that we have found different invariants associated to the same (group of) characters. In such a case, all of them are retained and considered either as successive or alternative to each other depending on which choice leads to better interpretations, as it will be explained soon. In case overlapping invariants are associated to different strings of symbols, this means that we have found the same invariant associated to different characters, as expected because of the variability of handwriting shape, as illustrated earlier. Therefore, they should not be considered as successive, but rather as alternative interpretation for the corresponding ink. It is often the case, however, that two invariants overlap at one of their extreme for just one symbol. Such a case was also expected, in that when two characters are written one after the other, the end of the first and/or the beginning of the last may be modified or even merged into a single stroke because of the anticipatory effect mentioned earlier, or because an extra stroke is added to avoid the much more expensive (for human movement point of view) sequence of perception/action cycles needed to lift the pen from the paper, compute the next target and perform a ballistic movement towards it. In both cases, the symbols corresponding to the first and/or the last stroke of a sequence are intrinsically less reliable. So, when these cases happen we consider the two overlapping invariants as successive.

Given the above criteria for ordering the invariants along the ink, however, there may be many possible solutions to the problem, each of which corresponds to an interpretation. To order them, we consider that, at least intuitively, the best solution is the one that simultaneously minimize the gaps and overlaps among invariants and maximize the rank of the solution. Under these assumptions, building the interpretation can be reformulated as an optimization problem. To tackle it, we map our sequencing problem to a graph, representing each invariant as a node of the graph, and possible connections between pair of successive nodes as arcs (fig. 1e). Adopting such a representation, the optimization problem can be reformulated as the well-know shortest path problem in graph theory by defining a function that maps the arc to a cost function and then searching for the path corresponding to the minimum cost by adopting any of the algorithm proposed in the literature, such as the Dijkstra's algorithm and its variants [11]. Our cost function is obtained by assigning to the node i the weight

$$Ni = Max - Si,$$

where *Si* is the score assigned to the node *i* as explained before and *Max* the largest *Si*, and to the arc going from node *i* to node *j* the cost

$$Aij = 0 \qquad \text{for adjacent nodes;}$$

$$Aij = L*P \qquad \text{for gap of length } L;$$

$$Aij = min(Si,Sj)*L/Lk \qquad \text{for overlap of length } L;$$

where *P* is the gap penalty, and *Lk* denotes the length of

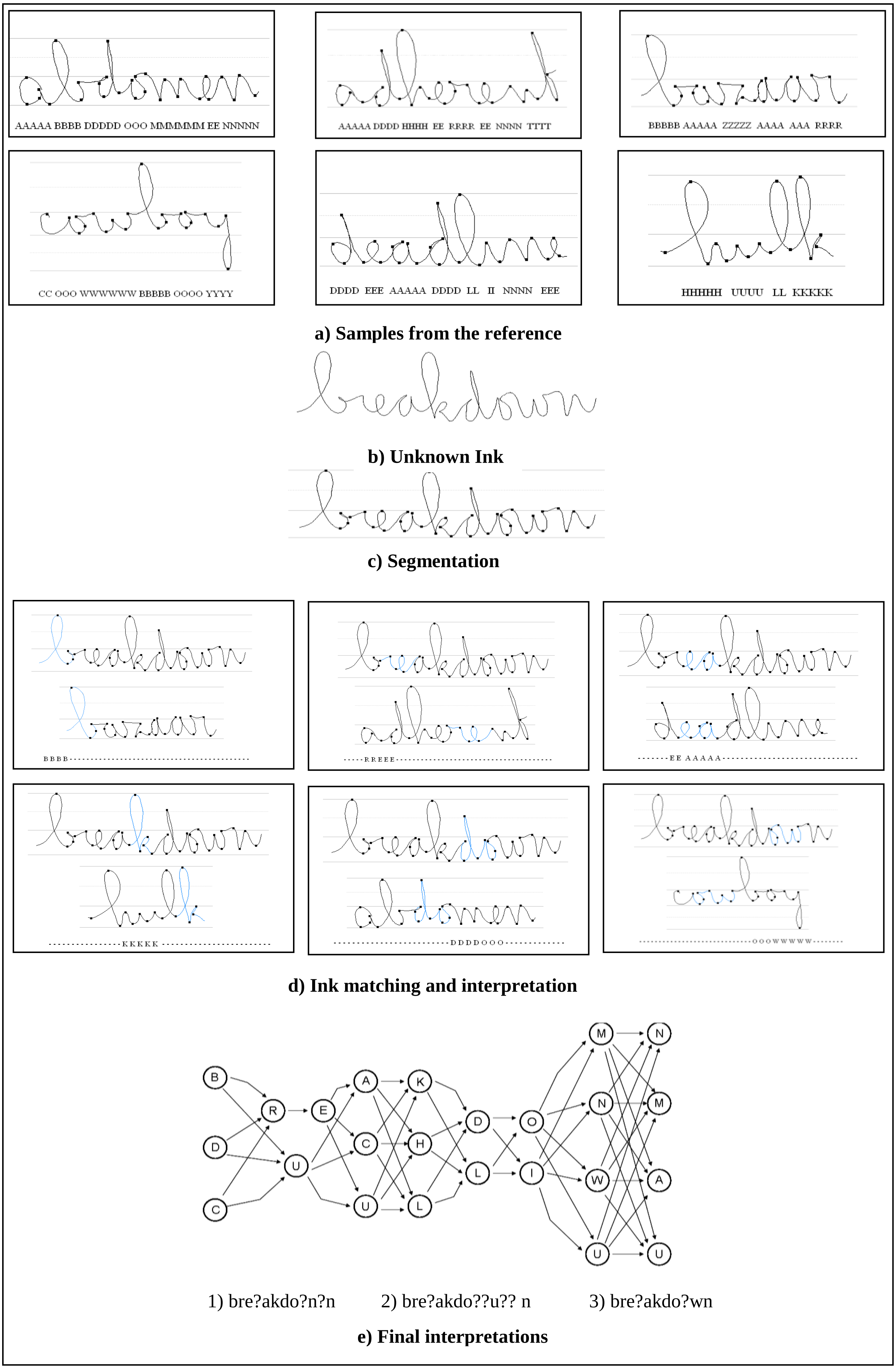

Figure 1: The method at a glance.

the lowest ranking sequence of the pair. Thus the cost *Cij* of the path going from node *i* to node *j* is:

$$Cij=Ni+Aij+Nj$$

The desired interpretation for the unknown, thus, corresponds to the shortest path of the directed graph representing the adjacency between the matching sequences. In case there are overlapping sequences at the beginning/end of the word, each of them needs to be considered as starting/ending node. Fig. 1e shows the top 3 interpretation for the unknowm word of fig 1b), the symbols ? indicate unmatched strokes in the unknown.

5. EXPERIMENTAL RESULTS

The performance of the proposed method has been evaluated on the Unipen data. In particular, we have used the benchmark ICROW_03 that was proposed at ICDAR 2003 [12]. We have selected a data set of 210 words, of which 106 were used as reference set and the remaining 104 as test set. The words of the reference setwere processed as described in Section 3 for associating to each of their strokes the character labels.

During the experiment, each word of the test set was matched against the whole reference set to find the matches, merge and score them. Then, the sequences provided by the ink matching were organized into a graph and the shortest path algorithm provided a ranked list of interpretations. To provide a quantitative evaluation of our method, we have considered that a word has been correctly recognized even when some of its strokes do not receive any label, but all the characters are correctly recognized and located. This assumption was introduced to deal with extra strokes associated to ligatures, as discussed in Section 3. Under this assumption, we found that 12 out of the 104 words of the test set (11,58%) were not correctly recognized. For the remaining 92 words, in 8 cases (7,69%) the best interpretation was the correct one, in 13 cases the correct interpretation was either the second or the third one, for 23 it was the 4th or the 5th, for 36 words it ranked between the 6th and the 10th and eventually, for 8 words it ranked below the 10th position. Table I reports the recognition rates that follows from those numbers.

Table I. Recognition rates

Top 1	Top 3	Top 5	Top 10	Below	Errors
7,69	20,19	42,31	76,92	92,39	7.69

6. CONCLUSION

We have surveyed a research project aimed at designing a method for cursive handwriting recognition which follows from studies on handwriting movement. The approach proceeds by extracting from the ink the invariants, i.e. those pieces of ink that are usually associated when drawing a character or a sequence of them, and then concatenating those invariants within the handwriting along the writing direction.

The proposed implementation of the method builds upon two pillars: saliency of information and context dependent shape analysis. The concept of saliency is exploited for finding where the relevant information is located within the ink, avoiding any a priori assumption on the ink shape. Context dependent shape analysis is exploited for finding similar pieces of ink between the unknown and the reference set, without resorting to some predefined set of features to describe the ink shape, nor to classification to associate the ink with its interpretation.

The experimental results, although obtained on a small set of data produced by a single writer, confirm that the proposed approach is effective in providing correct interpretations even in case the unknown is not included in the reference.

One may argue, however, that the method usually provides many interpretations, from which the right one needs to be extracted, and that solving this problem requires the use of a dictionary. We have not yet investigated systematically this aspect, but preliminary experiments have shown that the problem is easier than it may appear, because most of the interpretations either contains orthographical errors or are non sense words. Thus, a spell-checking followed by a dictionary search should succeed in deleting them from the list of possible interpretations.

The analysis of the errors has also shown that most of them were on sample produced by the same writer. Looking at them, we realize that the writer was using invariants that were not included in the reference set. Thus, a larger reference set should be considered, so as to include as much handwriting variability as possible. On the other hand, as the reference set increases, the number of matches with an unknown becomes larger, leading to many more interpretations. So, the need to define more sophisticated criteria to rank the interpretations in such a way that the right one is pushed towards the top becomes indisputable.

Eventually, the analysis of the results in case of words whose correct interpretation ranks very low has shown that sometime good sequences receives low score because of gaps and overlaps that are due to imprecise labeling of the reference set. Improving the performance of the labeling algorithm for the reference set, will then be another goal of our future investigations.

ACKNOWLEDGEMENTS

The author would like to thank his colleagues, co-authors and students which, to various extent, have contributed to design, develop and test the algorithms for implementing the proposed method.

REFERENCES

[1] M. Kawato, Internal models for motor control and trajectory planning, *Current Opinion in Neurobiology*, 9, 1999, pp. 718-727.

[2] S. Grossberg and R.W. Paine, A neural model of corticocerebellar interactions during attentive imitation and predictive learning of sequential handwriting movements, *Neural Networks*, 13, 2000, pp. 999-1046.

[3] R. Plamondon and F.J. Maarse, An Evaluation of Motor Models for Handwriting. *IEEE Trans. on Systems, Man and Cybernetics*, 19,1060-1072.

[4] A. Alimi and R. Plamondon, R., Performance

Analysis of Handwritten Strokes Generation Models. *Preproc. of 3th Int. Workshop on Frontiers in Handwriting Recognition - IWFHR III*, (pp. 272-283). Buffalo, NY

[5] B. Bontempi and A. Marcelli, On-line segmentation of cursive script using an arclength representation. In *Handwriting and drawing research: Basic and applied issues*, Simner M.L., Leedham C.G. & Thomassen A.J.W.M. eds. IOS Press: Amsterdam, 1996, pp. 315-327.

[6] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag: New York, 1977, pp. 161-164.

[7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. PAMI*, vol. 20, 1998, pp. 1254–1259.

[8] C. De Stefano, G. Guadagno, A. Marcelli, A saliency-based segmentation method for on-line cursive handwriting, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18 (6), 2004, pp. 1139-1156.

[9] C. De Stefano, M. Garruto, L. Lapresa, A. Marcelli, Detecting Handwriting Primitives in Cursive Words by Stroke Sequence Matching, in: *Advances in Graphonomics*, A. Marcelli and C. De Stefano (eds.), Arezzo: Zona Editrice, June 2005, pp. 281-285.

[10] C. De Stefano, A. Marcelli, A. Santoro, On-line Cursive Recognition by Ink Matching, *Proc. IGS 2007*, Melbourne, Australia, 2007, pp. 23-27.

[11] S. Skina, *The Algorithm design manual*, Springer: New York, 1998, pp. 100-102.

[12] http://www.unipen.org/products.html