Improving the Use of Contours and Skeletons for Off-line Cursive Script Segmentation

A. Chianese*, M. De Santo** and A. Picariello*

* *DIS*

Università Degli Studi di Napoli 'Federico II'
Via Claudio, 21 80125 Napoli - Italy
{angchian, picus}@unina.it

** *DIIIIE*

Università Degli Studi di Salerno
84084 Fisciano (Sa) - Italy
desanto@dia.unisa.it

Abstract

Segmentation process represents a very difficult task for handwritten recognition aims, due to large variations involved in cursive scripts. This paper describes a method for off-line unconstrained handwritten segmentation, based on a combined use of contour and skeleton derived information. Experiments are discussed and successful segmentation results are reported.

1. Introduction

Handwriting recognition systems are generally divided into on-line and off-line systems. On-line systems [1] refer to those systems for which the recognition process occurs while the user is writing, typically by the motion of an optical pen across a surface. Off-line systems, on the contrary, perform handwritten text recognition after the human writing process is terminated and the recognition process involves the analysis of a text image obtained by digitizing, through a scanning device, the writing samples. The absence of dynamic information makes problems related to those last systems still a great challenge for researchers. The recognition rates for off-line handwritten reported in the literature [2][3][4] fluctuate between 50% and 96% depending on the context and on the limitative assumptions the several methods make. However, although is indubitably a critical step of recognition, only few works have presented into details the segmentation process and the results of this module in term of correct segmentation rate.

In the following we briefly summarize main previous results. One of the basic work in the field is that of Maier [5]: he introduced a scanning strategy for a classification of neighboring pixels which brings to achieve vertical cuts in the local minima of connecting strokes of the text. Lecolinet and Crettez [6] made a distinction between significant components of the script related to the characters in a word - 'graphemes' - and unessential components related to the so-called 'ligatures' among characters. They showed that most of the ligatures are the valleys of the upper outline of a given

word. Kahan et Al. [7] detected the cutting points of the touching characters analysing a function based on vertical pixel projection. Boccignone et Al. [8] improved the previous method adding the use of a function $H(x)$ defined as the difference between the top and the bottom profiles of external contour.

Bozinovic and Srihari [9] used contour to formulate some heuristic considerations (e.g., presence of peaks and valleys, dots over i, holes, ...). In their fundamental work, they propose an approach based on a perceptive model, so-called 'multilevel' [10], according to which an image containing handwritten is transformed through a hierarchy of representations. They also introduced some heuristic for intermediate-level segmentation and recognition process, such as the detection of the reference-lines (i.e. of those lines in which is concentrated the most of a text), the analysis of profile projections and a simple presegmentation based on lower contours of a word. Yanikoglu et Sandon [11], propose a character segmentation based on the use of the so-called 'separator lines', i.e. those separation lines obtained at a different degree of inclination and that touch the script only in one ribbon. In other word, segmentation points are obtained through the optimization of a cost function of the couple (α, P) , where α is the angle of the separation line, while P is the point intercepted on the line itself. The use of dynamic programming technique contributes to detect the different kind of connection among the letters forming a given word.

As previously discussed, the most part of the quoted authors scarcely discuss this phase results, while giving a key role to the segmentation process.

On the contrary, in this paper, we describe a segmentation method designed according to a combined strategy based upon contours and skeletons. This strategy consists into a preliminary extraction of cut points and a final validation of them: contours derived information are used to extract candidate cut points, while skeletons are used to segmentate unessential and essential part of a script word, for cut point validation.

The paper is organized as follows: in *section 2* we give an overall description of the method, describing the cut points extraction task using contour derived information (*section 2.1*), the ligature finding process (*section 2.2*) and the validation phase (*section 2.3*); experimental results and discussion are showed in *section 3*.

2. Method Description

The basic idea according to which the task is accomplished relies upon the assumption that cut points are localized, in the most of the cases, in the so-called middle-zone of a line. A number of authors define middle-zone as the zone on which the main body of a script line is localized. Our basic working hypothesis is that a ligature in the middle-zone connects most of letters. Our experiments showed that about 99.0% of use samples follow this rule. From the above assumption, we derive several heuristic criteria that are used in our algorithm.

- a) *In correspondence of a connection point, the thickness of the word in the middle zone always assumes a minimum* (see later for a better description).

Note that rule a) is *necessary but not sufficient*.

Rule a) captures the desired cut-points together with other points according to the particular shape of a letters composing the word. For this reason, it becomes necessary the introduction of other rules for extracted cut points validation.

In particular, the segmentation criteria assume the presence of ligatures among two letters, expressed by

b) *A cut points occurs only in correspondence of a ligature.*

Rule b) also states that two characters only touch in one contact point and that a ligature links those points. Or experiments have showed that about 98.0% of examined cases follow this rule.

Figure 1 shows a diagram of the proposed technique.

Fig. 1. System Diagram

2.1 Segmentation Points extraction

It is well known in the literature that a text line may be segmented into three zones: *middle, upper and lower zone*. The middle zone is that related to the main body of cursive script, while the remaining zones are related to those zones where one can found ascender (upper zone) and descender (lower zones) components of a character. The separation lines of middle zones among lower and middle and among upper and middle are also called base-lines, in particular lower and upper base-line.

Several techniques detect base lines through an analysis of the histogram of vertical pixels: we introduced a more efficient detection method calculating the straight lines of the local minima on lower contour (lower line) and local maxima on upper contour (upper line) [8].

In particular, given a set of contours forming a text line, we extracted the set M of local maxima on upper contour and the set μ of local minima on lower contour. By means of linear regression of M and μ we usefully approximate the upper base-line and the lower base line of our text line

The detection of base lines is used for a preliminary extraction of cut points, through the use of an appropriate thickness contour function $T(x)$.

The function $T(x)$ is defined as

$$T(x) = \{Top_{MZ}(x) - Bottom_{MZ}(x), x = 1, 2, \dots, N\}.$$

$Top_{MZ}(x)$ is the top profile of the external contour as seen from the top; while $Bottom_{MZ}(x)$ is the bottom profile as seen from the bottom. Both for top and bottom contour, if more then one contour occur at a given x co-ordinate, $T(x)$ takes into account only those located within the middle-zone or in the proximity of it.

$T(x)$ is quite noisy; so we perform a smoothing phase through a median filtering of window K . This transformation are performed to obtain a typical peak & valley signal (figure 2).

Fig. 2. An example from our database

The valleys of the thickness function, when surrounded by two peaks, are representative of those areas where thickness of the text line decreases, and their local minima can be considered candidate break points. Local maxima and local minima are calculated using the following formula based on the gradient of $T(x)$,

$$T_{xx}(x) = \frac{T(x+L) - 2T(x) + T(x-L)}{T(x)}$$

As suggested in [11], the candidate cut points are chosen among those points x^* : $T_{xx}(x^*) > \tau$. L and τ are parameters that can be tuned using experimental results

2.2 Ligature Extraction

The skeleton derived information is used to improve the previous results through the introduction of some validation criteria derived from the consideration that a valid segmentation point has to be located on ligatures. The use of skeletons for ligature findings has been suggested from those works which try to recover dynamic on-line information from static off-line inputs [12], [13].

An iterative thinning algorithm is applied to reduce the lines in the writing to one pixel width so that the strokes can be followed better. The algorithm used is essentially based upon a thinning algorithm due to Arcelli and Sanniti di Baja [14]. The skeletonization algorithm computes the so-called Medial Axis Transform (MAT), which converts an 8-connected figure into an 8-connected curve having unit width. MAT pixels are then labeled as End Points (EP), Normal Points (NP) and Branch Points (BP) depending on the fact the number of MAT pixel in their 3×3 neighborhood is one, two or greater then two.

However, The MAT does not faithfully reflect the original figures, but some distortions are introduced. We have to note that this is a very complex problem common to all thinning procedures. Several authors propose to recover this loss of information after the thinning phase, in a successive phase that we call Distortion Removal. Usually, after the skeletonization process, a polygonal approximation procedure is applied, with the aims of reducing digital line to a polyline made of k-connected segments, while reducing the amount of data to be processed.

Starting from the previously described considerations, ligatures may be regarded as the sequence of strokes which connect a BP to another BP, and satisfies the following basic conditions:

- strokes are not part of a loop
- the polyline obtained by the sequence of strokes may be approximated to a straight line.

Figure 3 gives an example of the ligature finding task performances.

Fig. 3. The Ligature Finding Task.

2.3 Validation

This phase has the aim of combining all the information extracted from the previously performed phases, so to validate or invalidate the extracted segmentation points. In particular we apply the following rules, based on several heuristic considerations:

- *if* one candidate cut points is located on a Ligature, *then* validate and move it into the minimum of stroke;
- *If* more then one candidate cut points is located on the same Ligatures, *then* validate only the cut point where the ligature assume a minimum, invalidate the others.

The rules put into evidence that a cut point always is located on a ligature, and that only one segmentation point can be assumed on a given ligature; anyway, a cut point is always located on the minimum of a single ligatures.

Figure 4 shows an example of how proposed method works on a sample coming from our experimental database.

Fig. 4. Segmentation Points before and after Validation

3. Experimental Results and Discussion

We use for our experiment a database built with about five hundred images containing handwritten produced by different students of the University of Salerno. The students were asked to write their addresses and some administrative requests (like curricula, certification and so on...). Text lines are assumed to be separated at the input of our segmentation process.

TAB I
PHASE I: CUT POINTS BEFORE VALIDATION

	<i>Human Detected Cut point</i>	<i>Machine Detected Cut point</i>	<i>Cut points detected by machine in the same position of a man</i>	<i>detected by man and not detected by the the machine</i>
Cut Points	1931	3386	1859	72

Table II report and I the results for character segmentation phase, before and after the use of the validation phase. The number of not detected cut points in the first phase is quite low (2.08 %) as respect to the results reported in the literature, while is clearly high the number of undesired cut points. The validation phase produce a generally stable result of the cut points detected by man and not detected by the the machine, while sensibly reducing the other points, also with the appreciable result of moving cut points in a more precise position.

TAB II
PHASE II: CUT POINTS AFTER VALIDATION

	<i>Human Detected Cut point</i>	<i>Machine Detected Cut point</i>	<i>Cut points detected by machine in the same position of a man</i>	<i>detected by man and not detected by the the machine</i>
Cut Points	1931	2745	1859	72

The experimental results confirm that in the most of cases, the assumed hypotheses are the right ones.

In particular:

- The most of characters are connected to their successive with a ligature: only few cases revealed the presence of no ligatures (e.g. some oo) or more than one contact point (such as some slanted ll, or sc).
- The most of ligatures occurs in the middle-zone
- The presegmentation+validation phases reduce significantly the number of bad cut points, so giving the classifier a more precise number of single characters.

Further work will be devoted to the analysis of the error causes in the various stages of the method and to a more precise ligature detection. Fuzzy logic based methods are actually under study.

References

- [1] C. C. Tappert, C.Y. Suen, T. Wakahara, "The State of the art in on-line handwriting recognition", IEEE PAMI 12 (8):787-808, 1990.
- [2] J-C Simon, "Off-line cursive word recognition", Proceedings of the IEEE, 80(7):1150-1161, July 1992

- [3] C. Y. Suen, R. Legault,, C. Nadal, M. Cheriet, L. Lam, "*Building a new generation of handwriting recognition systems*", Pattern Recognition Letters, pp. 303-315, 1993
- [4] Y. Lu and M. Shridar, "*Character Segmentation in Handwritten words - an overview*", Pattern Recognition, vol. 29, n. 1, 1996, pp. 77-96
- [5] M. Maier, "*Separating characters in Scripted Documents*" Proc. of ICPR 1986, Paris, 1986
- [6] E. Lecolinet, J.P. Crettex, "*A Grapheme Based Segmentation Technique for Cursive Script Recognition*", Proc. of 1. ICDAR 91, : 740-748, Saint Malo, France, 1991
- [7] S. Kahan, T. Pavlidis and H. S. Baird, "*On the recognition of printed characters of any font and size*", IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-9(2), pp. 274-287, (1987).
- [8] G. Boccignone, A. Chianese, M. De Santo and A. Picariello, "*Improving the use of Contours for Off-Line Cursive Script Segmentation*", Lecture Notes in Computer Science, vol. 974, 1995, pp. 545-550
- [9] R. M. Bozinovic, S. N. Srihari, "*Off-line cursive word recognition*", IEEE PAMI, 11(1): 68-83, January 1989
- [10] S. N. Srihari, R. M. Bozinovic, "*A multi-level perception approach to reading cursive script* ", Artificial Intelligence, 33:217-255, 1987
- [11] B. A. Yanikoglu, P. A. Sandon, "*Recognizing Off-line Cursive Handriting*", proc. of ICDAR'93, pp. 397-403
- [12] G. Boccignone, A. Chianese, L. P. Cordella and A. Marcelli, "*Recovering dynamic information from static handwriting*", Pattern Recognition, Vol. 26, n. 3, pp. 409-418
- [13] D. S. Doermann and A. Rosenfeld, "*Recovery of temporal Information from static Images of Handwriting*", Proc. 1992 IEEE Comp. Soc. Conf. On Computer Vision and Pattern Recognition, Urbana Champaign, Illinois, pp. 162-168.
- [14] C. Arcelli and G. Sanniti di Baja, "*A Thinning algorithm based on prominence detection*", Pattern Recognition 13, 1981, pp. 225-235