Improving the Use of Contours for Off-Line Cursive Script Segmentation

G. Boccignone, M. De Santo

DIIIE
Università Degli Studi di Salerno
84084 Fisciano (Sa) - Italy
{desanto, boccig}@dia.unisa.it

A. Chianese, A. Picariello

DIS
Università Degli Studi Federico II
Via Claudio, 21 80125 Napoli - Italy
chianese@nadis.dis.unina.it

Abstract. In this paper we show a new method for efficiently extracting information from contour analysis and demonstrate their use to face the Cursive Script Segmentation challenge. Expressly, our method improves the use of contour to detect the so-called baselines and to use them for a "zone" analysis of the text rows. Furthermore, we show how it is possible to use our method to solve critical situations due to slant and character overhanging. A final discussion about experimental results is also reported.

1. Introduction

Cursive Script Recognition (CSR) is a great challenge in the field of Optical Character Recognition. Several methods have been proposed in the literature for *off-line* CSR, but they can be collected into two main classes: analytical and global methods [1]. While global methods attempt to recognise an entire word, the analytical ones attempt to recognize the characters which form a word. In analytical methods, the character *segmentation,* i.e. the detection of characters within a word, is considered particularly critical. Many authors have used contour derived information for segmentation, due to the fact that it is relatively inexpensive to be obtained and moreover it can be extracted during other pre-processing phases, which are considered mandatory ones by a number of authors. Maier [2] 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 [1] 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. [3] detected the cutting points of the touching characters analyzing a function based on vertical pixels projection. Other authors [4,5] 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. In spite of their interesting characteristics, last quoted methods have been principally applied for segmenting *printed* characters, where main problems derive from the skew of text lines in the document. On the contrary, cursive script shows very significative and pervasive slant and overhanging problems and, what is worst, text lines that are not linear or horizontal. So, for CSR various techniques have been introduced to recover these problems. Bozinovic and

Srihari [6] used contour to formulate some heuristic considerations (e.g., presence of peaks and valleys, dots over i, holes, ...). Kimura et al. [7] proposed a methodology based on recursive classification/segmentation, using a 'similarity function' based on contours. Finally, there are other approaches [8] where the slant and overhanging problems have been faced using concavity and convessity analysis. As we noted, it is important to remark that all mentioned methods make a considerable amount of work to avoid slant and overhanging effects and, what is worst, this work is frequently lost for successive elaboration. Starting from these considerations, in this paper we introduce a new method which uses contour derived information in detecting cut points through "base-lines" determination which can be usefully utilized in the classification phase (see for ex. [5, 6]). The paper is organized as follows. In section, 2 we propose a preliminary analysis useful for base-lines extraction. In the same section, we furnish a new method for cut-points detection based on the idea that ligatures mostly occur in middle-zone of the text line. Finally, in section 3 we discuss some experimental results and in section 4 some concluding remarks.

2. The Method: Base-lines Detection and Segmentation through Contour Analysis

The first step of our method consists in detecting zones in which is possible to simply and reliably recognize the ligatures among characters. As a starting point, a ligature can be detected considering the histograms of the vertical thickness of the contour, described by the function called H(x) that, as reported in [4], is defined as the distance between the maximum and minimum y-coordinate of the contour of a word, for each x-coordinate. So, the ligatures could be found in the region in which H(x) has valleys surrounded by peaks. However, it is easy to note that H(x) gives wrong information in the case of characters with overhanging boxes. We can avoid these drawbacks considering that ligatures among characters occur in the *middle zone* of the text line and defining a new function based on the histograms calculated in the middle zone itself. In this work we start from the hypothesis that the input text has been previously organized in text lines. A text line, in occidental languages, is a collection of proximate successive words which span from left to right position in the document and may be segmented into three regions or zones: upper, lower and middle zone; in the literature this activity is known as 'base-line determination', and refers to the detection of the main body of a cursive line [6]. The upper zone refers to the ascender components of characters, the lower zone to the descender components, and the middle zone to the main body. Different methods have been used in the literature for base lines individuation, mostly based on histograms analysis [6, 7]: however, these methods are strongly influenced by the presence of writing and digitalization skew of the text line and by the quoted fact that they are not strictly horizontal or linear. We avoid the problems generated by histograms, using contour features that will be reused in successive phases. For this aim, it is useful to introduce some preliminary definitions. Let us consider a discrete plane $\{x_n, y_n\}$ and a closed region whose border is detected by Γ, described by the function Γ: $f(x_n, y_n) = 0$ where $(x_n, y_n) \in I \times I$, being I the integer set. Let $R_x(x^*_n)$ and $R_y(y^*_n)$ be subranges of **I** defined as

$$R_x(x^*_n) = [x^*_n - \Delta x_n , x^*_n + \Delta x_n] \qquad R_y(y^*_n) = [y^*_n - \Delta y_n , y^*_n + \Delta y_n]$$

We define *extremes* of Γ those points (x^*_n, y^*_n) for which one of the following condition occur:

(1) $f(x^*_n, y^*n) > f(x^*n+k, y^*n)$ (local maximum)
(2) $f(x^*n, y^*n) < f(x^*n+k, y^*n)$ (local minimum)
(3) $f(x^*n, y^*n) > f(x^*n, y^*n+r)$ (local left extreme)
(4) $f(x^*n, y^*n) < f(x^*n, y^*n+r)$ (local right extreme)

for each k Î Rx(x*n) and for each r Î Ry(y*n)

Using geometrical considerations, we also define ascender contour a set of points of G (i.e. and arc) which starts from a local minimum and finishes to the first successive local maximum, and descender contour an arc which starts from a local maximum and finishes to the first successive local minimum, upper contour the arc which starts from left extreme minimum and finishes to the first successive right extreme, and lower contour an arc which starts from right extreme and finishes to the first successive left extreme (figure 1).

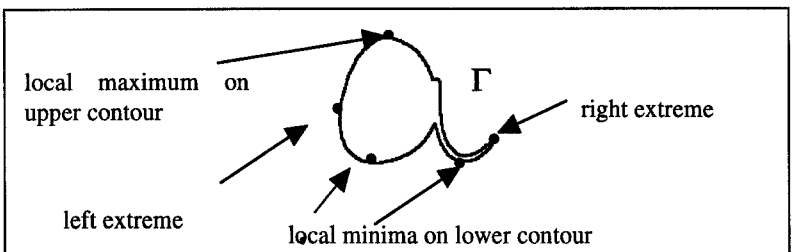


fig. 1: local minima, local maxima, upper and lower contours

Given a text line, let us define Σ as the set of local maxima detected on the upper contours of the line and σ as the set of local minima of lower contours. By means of a least square interpolation of Σ and σ, we may usefully approximate the upper-baseline and the lower base-line of our text. Note that our algorithm dinamically chooses the Δx_n and Δy_n ranges of the previous formulas taking care of avoiding the inclusion in Σ and σ of points which are near to "pen-up" and "pen-down" areas. In fact, these points do not represent significative information for base-lines determination but, on the contrary, may bring to distortions. Fig. 2 shows the steps for base-lines detection.

While it is clear that the accuracy of our approximation depends on the number of points considered, i.e. it is better if the text-line is long, we have found that experimental results show how, even if considering short lines (e.g., a single word), our method does not reduce the number of right cut points and does not introduce wrong cut points.

Furthermore, it is important to note that the use of this method to detect the middle zone avoid the effects of the skew and minimize those due to the unperfect linearity of text.

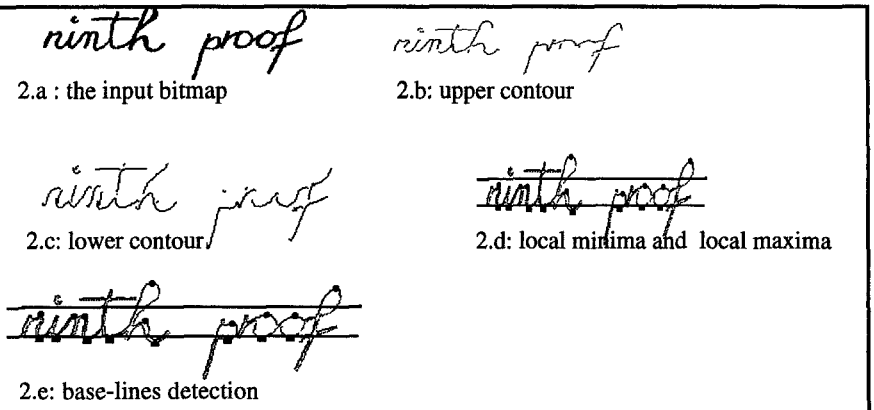


figure 2: text rows and zones extraction

In fact, our base-lines are generated starting from the analysis of the *extremes* of the contour, whose interpolation statistically represent the "direction" of the text-line in a way that is scarcely influenced by the quoted problems.

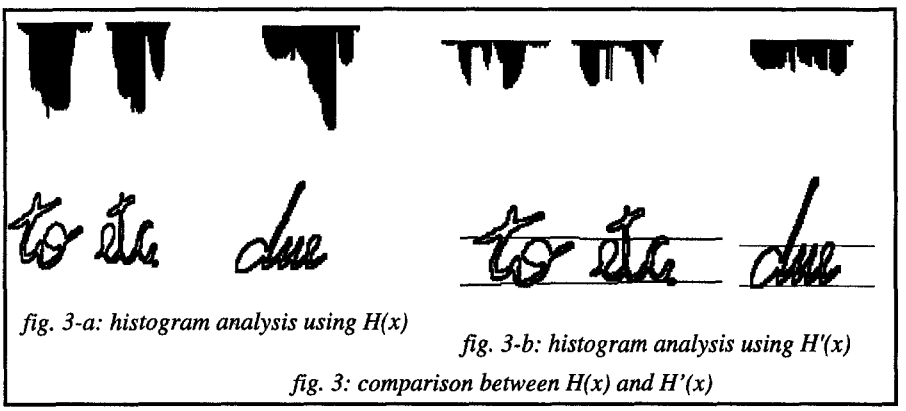


fig. 3-a: histogram analysis using H(x)

fig. 3-b: histogram analysis using H'(x)

fig. 3: comparison between H(x) and H'(x)

Now, using the base-lines, we can usefully introduce a new function for cut points detection. This function, that we call H'(x), is defined as the distance along y'-axis between the upper and the lower contours in the middle zone for each x'-coordinate. The new y'axis is obtained considering the new coordinate system which assumes the lower base line as its x'-axis. In figure 3 we show the application of H(x) and H'(x) functions to some word samples. It is easy to see that H(x) function fails in presence of horizontal strokes which span over successive characters in the word. In the same figure we illustrate an equivalent critical situation caused by 'slant'. An excessive slant, in fact, causes the overhanging among boundary boxes of 'd' and 'u'. Using the function H'(x), we can avoid this kind of problems.

3. Experimental results and discussion

We tested our Cursive Script Segmentation method on a database of some hundreds of handwritten mail addresses collected by students of our University (see fig. 4 for an example). The aim of our experiments was an analysis of safe cut points detected by our method by means of a comparison between the results obtained using H and H' functions and the average of points detected by human observers [9]. In particular we want to find: a) the number of undetected cut points; b) the number of intercharacter cut points detected within a single letter (typical of "m", "n", "u", etc.); c) the number of superfluous cut points detected within the same ligature. In tab. 1 we show the results of the described experimental work.

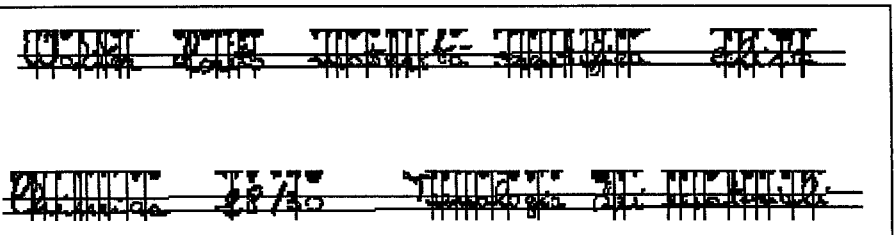
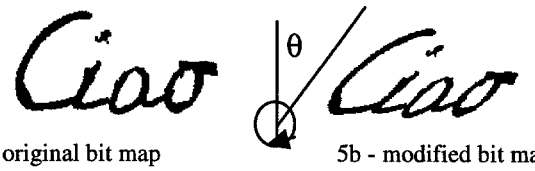


fig. 4:An Example from our database

	undetected cut points (%)	inter-character cut points (% on Total)	superfluous cut points - (% on Total)
H(x)	13	28	12
H'(x)	5	31	14

tab. 1: experimental results

The use of H' reduces the number of undetected cut points from 13% to 5%. This first positive result confirms the hypothesis that a large number of ligatures is located in the middle zone of the text line. Let us now consider the results relative to intercharacter cut points. Both H and H' functions exhibit an high number of wrong cut points due to the fact that the human observer uses semantical information to individuate true ligatures. Our approach slightly amplifies the number of inter character cut points because "m", "n", "u" characters are mainly located in the middle zone. For the same reasons, similar results can be observed for superflous cut points. However, it is important to remark that intercharacter cut points and superflous cut points can be recovered in an easier way than undetected cut points in successive steps of the recognition process. In a successive phase of our experiments, we start to quantitatively test the effectiveness of our method with respect to its sensitivity to the slant. To this aim, about one hundred of mail addresses samples have been transformed by means of a bit map rotation (see fig. 5), using graphical facilities.



5a - original bit map 5b - modified bit map

fig. 5 - quantitative study of the slant effects

Let us call Θ the rotation angle. We found that the performance of the method does not vary appreciably for Θ varying from 0° to 45°. In fact, our "middle zone" method avoids the main effect of the slant distorsion which is related to the increase of overhanging among characters due to ascender and descender components. When Θ spans over 45°, we report a quick degradation of the performance because of a propagation of overhanging effects to the middle zone components. We found these results encouraging because experience shows that cursive script rarely exhibits slant distortions with angles greater than 45°.

4. Conclusions

The discussed Cursive Script Segmentation method has been implemented by means of an experimental Object Oriented Environment System realized at the University of Salerno [10]. This environment allows a C++ programmer to develop image processing tasks through the availability of suitable class libraries. We are currently integrating the method with some heuristic knowledge to be applied in the character detection and we are starting a massive testing of the method itself.

References

1. E.Lecolinet and J.Crettez, "A Grapheme-based Segmentation Technique for Cursive Script Recognition", *ICDAR 1991*, Saint-Malo, 1991.
2. M. Maier, "Separating Characters in Scripted Documents", *ICPR 1986*, Paris 1986
3. S. Kahan, T. Pavlidis, H. S. Baird "On the recognition of printed characters of any font and size", IEEE Trans. Pattern Anal. and Machine Int., PAMI-9, 2, 1987.
4. F. Arcelli, A. Chianese, M. De Santo, A. Picariello, "Approaching Char. Segmentation Using Cont./Skel. Analysis",. *SCIA 1993, Tromso, 1993.*
5. S. Liand, M. Shridhar, M, Ahmadi, "Segmentation of touching char. in printed document recognition", Pattern Recognition, 27, 6, 1994
6. R.M. Bozinovic, S.N. Srihari, "Off-line Cursive Script Word Recognition", IEEE trans. on PAMI, PAMI-11, 1989.
7. Kimura, Shridhar, Chen "Improvements of a Lex. Dir. Algorithm for the Recogn. of Uncostrained Handw. Words", *ICDAR 1993*, Tokyo, 1993.
8. B. Plessis, et. al., "A Multiclassifier Combination Strategy for the Recognition of Handwritten Cursive Words",. *ICDAR, Tokyo* 1993.
9. A. Fisher, R. Bolles, "Perceptual organization and curve partitioning ", IEEE Trans. Pattern Anal. and Machine Int., PAMI-8, 1, 1986.
10. G. Boccignone, A. Chianese, M. De Santo, A. Picariello, "Building an O.O Env. for Image Analysis and Processing", *ICDAR 1993*, Tokyo, 1993.