Text Line Detection in Unconstrained Handwritten Documents Using a Block-Based Hough Transform Approach

G. Louloudis[1], B. Gatos[2], C. Halatsis[1]

[1]Department of Informatics and Telecommunications, University of Athens, Greece
http://www.di.uoa.gr, louloud@mm.di.uoa.gr, halatsis@di.uoa.gr
[2]Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos", GR-153 10 Ag. Paraskevi, Athens, Greece
http://www.iit.demokritos.gr/~bgat/, bgat@iit.demokritos.gr

Abstract

In this paper we present a new text line detection method for unconstrained handwritten documents. The proposed technique is based on a strategy that consists of three distinct steps. The first step includes image binarization and enhancement, connected component extraction and average character height estimation. In the second step, a block-based Hough transform is used for the detection of potential text lines while a third step is used to correct possible splitting, to detect text lines that the previous step did not reveal and, finally, to separate vertically connected characters and assign them to text lines. The performance evaluation of the proposed approach is based on a consistent and concrete evaluation methodology.

1. Introduction

Text line detection is a critical stage towards unconstrained handwritten document recognition that refers to the segmentation of a document image into distinct entities, namely text lines. The problems that can be encountered in this stage are the difference in the skew angle between lines on the page, overlapping words (words of adjacent lines that have overlapping bounding boxes) and adjacent lines touching. Furthermore, the frequent appearance of accents in many languages (eg. French, Greek) makes the text line detection a challenging task.

In this paper, we present a new text line detection method for unconstrained handwritten documents. The main novelties of the proposed approach are (i) the partitioning of the connected component space into three subsets each treated in a different manner, (ii) the splitting of the bounding box of the connected components into equally spaced blocks each of them voting in the Hough domain and (iii) the efficient separation of vertically connected characters.

The paper is organized as follows: in Section 2, the related work is described. In Section 3 the method to segment text lines is detailed. Section 4 deals with the performance evaluation methodology. In Section 5, we present the experimental results and, finally, Section 6 describes conclusions and future work.

2. Related work

A wide variety of text line detection methods for handwritten documents has been reported in the literature. There are mainly three basic categories that these text line detection methods fall in. Methods lying in the first category make use of the Hough transform ([1], [2], [3]). In these methods, by starting from some points of the initial image, the lines that fit best to these points are extracted. The points considered in the Hough transform are usually either the gravity centers [2] or minima points [3] of the connected components. Methods lying in the second category make use of projections ([4], [5]). In [5], the histogram of the pixels' intensities at each scan line is calculated. The produced bins are smoothed and the corresponding valleys are identified. These valleys indicate the space between the lines of the text. Finally, the third category deals with methods that use a kind of smearing. In [6], a fuzzy runlength is used to segment lines. This measure is calculated for every pixel on the initial image and describes how far one can see when standing at a pixel along horizontal direction. By applying this measure, a new grayscale image is created which is binarized and the lines of text are extracted from the new image.

There are also some methods that do not lie in the previous categories. A recent paper [7] makes use of the Adaptive Local Connectivity Map. The input to the method is a grayscale image. In this method, a new image is calculated by summing the intensities of the neighbors in each pixel in the horizontal direction. A thresholding technique is applied in the new image and the connected components are grouped into location maps by using a grouping method. In [8], the method to segment text lines uses the count of foreground/background transitions in a binarized image to determine areas of the document that are likely to be text lines. Yi Li [9] describes a technique that models text line detection as an image segmentation problem by enhancing text line structures using a Gaussian window and adopting the level set method to evolve text line boundaries. The method described in [10] is based on a notion of perceptive vision: at a certain distance, text lines can be seen as line segments. This method is based on an extractor that detects text lines on low resolution images. This extractor is based on the theory of Kalman filtering.

3. Methodology

Text line detection in unconstrained handwritten document images deals with the following challenges: (i) each line that appears in the document may have an arbitrary skew angle; (ii) Accents may be cited either above or below the text line; (iii) Parts of neighboring text lines may be connected and (iv) Cursive words usually consist of connected characters.

To meet the aforementioned challenges, we propose a methodology which consists of the following three steps. The first step includes preprocessing for binarization and image enhancement, connected component extraction and average character height estimation. In the second step, a block-based Hough transform is used for the detection of potential text lines while a third step is used to correct possible false alarms, to detect possible text lines that the previous step did not reveal and, finally, to separate vertically connected characters and assign them to text lines. These stages are described in detail in Sections 3.1-3.3.

3.1. Pre-processing

First, an adaptive binarization and image enhancement technique described in [11] is applied. Then, the connected components of the binary image are extracted following approach [12] and for every connected component, the bounding box coordinates are calculated. Finally, the average character height *AH* for the document image is calculated [13]. We assume that the average character height equals to the average character width *AW*.

3.2. Hough Transform Mapping

In this stage, the Hough transform takes into consideration a subset (denoted as "Subset 1" in Fig. 1) of the connected components of the image. This subset includes all components with size identified by the following constraints:

$$\begin{aligned} &0.5 * AH < H < 3 * AH \\ &1.5 * AW < W \end{aligned} \quad (1)$$

where *H, W* denote the component's height and width, respectively, and *AH*, *AW* denote the average character height and the average character width, respectively.

This subset is chosen for the following reasons: (i) it is required to ensure that components which appear in more that one line will not vote in the Hough domain; (ii) components, such as accents, which have a small size must be rejected from this stage because they can cause a false text line detection by connecting all the accents above the core text line.

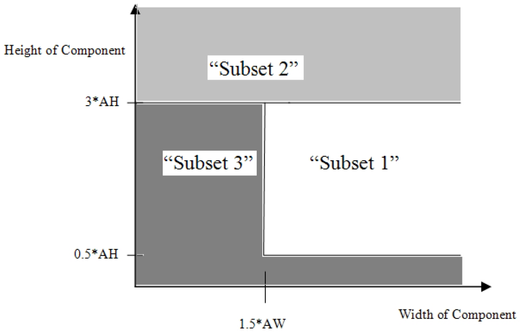


Figure 1. The connected component space partitioned to 3 subsets denoted as "Subset 1", "Subset 2" and "Subset 3".

In our approach, instead of having only one representative point for every connected component (as in [1], [2]), a partitioning is applied for each connected component lying in "Subset 1", so as to have more representative points voting in the Hough domain. This is accomplished by partitioning every connected component of the above set to equally- sized blocks. The width of each block is defined by the average character width *AW*. An example is shown in Fig. 2. After the creation of blocks, we calculate the gravity center of the connected component contained in each block. The set of all this points contributes to the Hough transform.

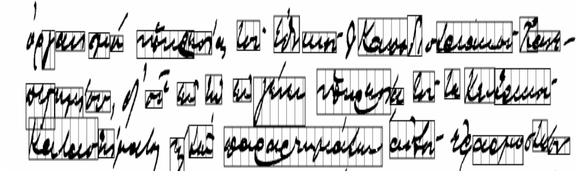


Figure 2. An example showing the connected components partitioning to blocks of width *AW*. All connected components not placed in a bounding box correspond to either “Subset 2” or “Subset 3”.

The Hough transform is a line to point transformation from the Cartesian space to the Polar coordinate space. A line in the Cartesian coordinate space is described by the equation:

$$x\cos(\theta) + y\sin(\theta) = p \qquad (2)$$

It is easily observed that the line in the Cartesian space is represented by a point in the Polar coordinate space whose coordinates are p and θ. Every point in the subset that was created above corresponds to a set of cells in the accumulator array of the (p,θ) domain. To construct the Hough domain the resolution along θ direction was set to 1 degree letting θ take values in the range 85 to 95 degrees and the resolution along p direction was set to 0.2*AH (as in [1]).

After the computation of the accumulator array we proceed to the following procedure: We detect the cell (ρ_i,θ_i) having the maximum contribution and we assign to the text line (ρ_i,θ_i) all points that vote in the area $(\rho_i-5,\theta_i)..(\rho_i+5,\theta_i)$. To decide whether a connected component belongs to a text line, at least half of the points representing the corresponding blocks must be assigned to this area. After the assignment of a connected component to a text line, all votes that correspond to this particular connected component are removed from the Hough transform accumulator array. This procedure is repeated until the cell (ρ_i,θ_i) having the maximum contribution contains less than n_1 votes in order to avoid false alarms. During the evolution of the procedure, the dominant skew angle of currently detected lines is calculated. In the case that the cell (ρ_i,θ_i) having a maximum contribution less than n_2 $(n_2>n_1)$, an additional constraint is applied upon which, a text line is valid only if the corresponding skew angle of the line deviates from the dominant skew angle less than 2° .

3.3. Postprocessing

The postprocessing procedure consists of two stages. At the first stage, (i) a merging technique over the result of the Hough transform is applied to correct some false alarms and (ii) connected components of “Subset 1” that were not clustered to any line are checked to see whether they create a new line that the Hough transform did not reveal. After the creation of the final set of lines, components lying in “Subset 3” as well as the unclassified components of “Subset 1” are grouped to the closest line.

The second stage deals with components lying in “Subset 2”. This subset includes components whose height exceeds three times the average height (see Fig.1). All components of this subset mainly belong to two text lines (see Fig. 3a). The procedure we follow to separate vertically connected characters consists of the following steps:

Step 1: Extract the skeleton of the corresponding connected component and detect all junction points [14] (see Fig. 3b).

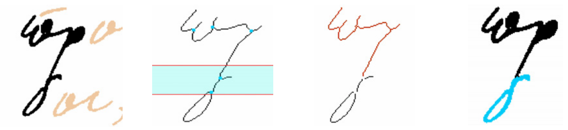
Step 2: Define the segmentation zone Z according to the contraints $h_c/2<y<3*h_c/2$ where h_c is the height of the connected component (see Fig. 3b).

Step 3: Remove from the skeleton image all pixels in the 3x3 neighbor of all junction points that lie inside the segmentation zone Z (see Fig. 3c).

Step 4: Extract the connected components of the skeleton image [12] and flag the upmost component (see Fig. 3c).

Step 5: Separation of the initial connected component into two different segments is accomplished by examining if a pixel is nearer to a flagged or to a non flagged skeleton pixel of step 4 (see Fig. 3d).

After the final step, the pixels of the two segments are assigned to the corresponding text line.



(a) (b) (c) (d)

Figure 3. Separating vertically connected characters: (a) a connected component that is likely to belong to vertically connected characters appearing into two successive text lines; (b) image skeleton, detected junction points and segmentation zone *Z*; (c) flagging the upmost skeleton component after removing all pixels in the 3x3 neighbor of all junction points that lie inside the segmentation zone *Z* and (d) final separation of the vertically connected characters.

4. Performance Evaluation Methodology

In the literature, the performance evaluation of a text line detection algorithm is mainly based on visual criteria in order to calculate the percentage of the correct segmented text lines ([2], [6], [7]). Manual observation of the segmentation result is a very tedious, time consuming and not in all cases unbiased process. To avoid user interference, we propose an automatic performance evaluation technique based on comparing the text line detection result with an already annotated ground truth. Similar evaluation strategies have been followed in several document segmentation competitions, such as ICDAR2003 & ICDAR2005 Page Segmentation Competitions ([15], [16]). The text line performance evaluation is based on counting the number of matches between the areas detected by the algorithm and the areas in the ground truth. We use a MatchScore table whose values are calculated according to the intersection of the ON pixel sets of the result and the ground truth.

Let *I* the set of all image points, G_i the set of all points inside the *i* text line ground truth region, R_j the set of all points inside the *j* text line result region, □*(s)* a function that counts the elements of set *s*. Table *MatchScore(i,j)* represents the matching results of the *i* ground truth region and the *j* result region as follows:

$$MatchScore\,(\mathrm{i},\,\mathrm{j}) = \frac{T(\,G_i \cap R_j \cap I\,)}{T(\,(G_i \cup R_j\,) \cap I\,)} \tag{3}$$

If *N* is the count of ground-truth text lines, *M* is the count of result text lines, and w_1, w_2, w_3, w_4, w_5, w_6 are pre-determined weights, we can calculate the detection rate and recognition accuracy for as follows:

$$Det = w_1 \frac{one2one}{N} + w_2 \frac{g\_one2many}{N} + w_3 \frac{g\_many2one}{N} \tag{4}$$

$$\mathrm{Rec} = w_4 \frac{one2one}{M} + w_5 \frac{d\_one2many}{M} + w_6 \frac{d\_many2one}{M} \tag{5}$$

where the entities *one2one*, *g_one2many*, *g_many2one*, *d_one2many* and *d_many2one* are calculated from *MatchScore* table (eq. 3) following the steps of [17].

A global performance metric for text line detection can be defined if we combine the values of detection rate and recognition accuracy. We can define the following *Text Line Detection Metric* (*TLDM*):

$$TLDM = \frac{2 * Det * \mathrm{Rec}}{Det + \mathrm{Rec}} \tag{6}$$

5. Experimental Results

The proposed text line detection method is tested on 50 unconstrained handwritten English and Greek documents. We experimented with the proposed methodology using 25 document images taken from the historical archives of the University of Athens and 25 document images taken from the Kennedy archive that is hosted in Dallas library. For all images, we have manually created the corresponding text line detection ground truth. The total number of text lines appearing on those images was 947. Parameters n_1 and n_2 in our methodology (Section 3.2) were experimentally defined to 5 and 9, respectively. An example of the proposed text line detection method is demonstrated in Fig.4.

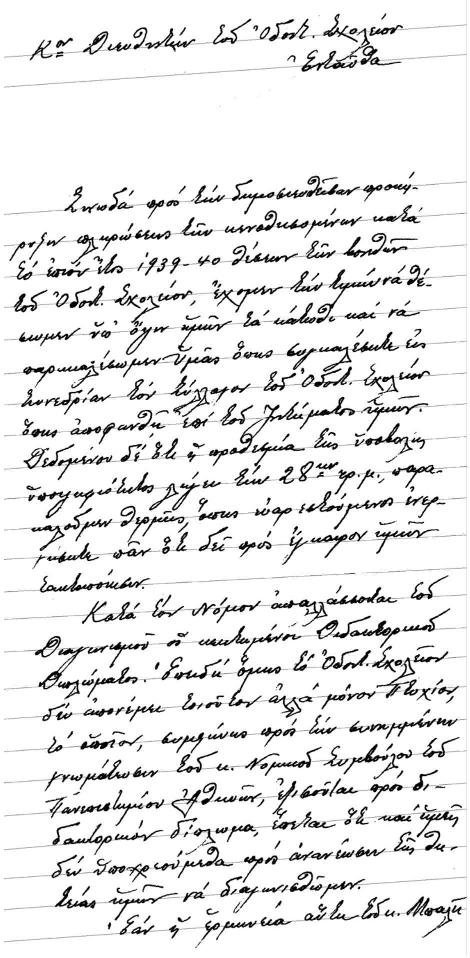


Figure 4. An image example showing the final lines created from the proposed method.

To check the effectiveness of our method we based on the performance evaluation methodology described in Section 4 using w_1=w_4=1, w_2=w_3=w_5=w_6 =0.25. For the sake of comparison we also implemented a fuzzy runlength approach (such as in [6]) and a projection profile approach (such as in [5]). Table 1 depicts our comparative experimental results in terms of detection rate, recognition accuracy and TLDM (see Section 4). As it can be observed from Table 1, the proposed methodology outperforms the other two approaches achieving a detection rate of 93.1% and a recognition accuracy of 96%. Most of the errors made by our approach are due to misclassification of accents as well as to possible skew angle changes along the text line.

Table 1. Comparative experimental results

	Detection Rate	Recognition Accuracy	TLDM
Fuzzy Run-Length	83.6%	73.1%	78%
Projection Profiles	70%	57.1%	63%
Our Method	93.1%	96.0%	94.5%

6. Conclusions and future work

In this paper we present a new text line detection method for unconstrained handwritten documents. The main novelties of the proposed approach consist of (i) the partitioning of the connected component space into three subsets each treated in a different manner, (ii) the splitting of the bounding box of the connected components into equally spaced blocks each of them voting in the Hough domain and (iii) the efficient separation of vertically connected characters.

Future work concerns the implementation of a method that will handle the difference of the skew angle along the text line. Another issue to handle is to find ways that correctly classify accents as they appear to cause most of the errors.

Acknowledgments

This work was supported in part by the Greek Secretariat for Research and Development under the PENED 2003 framework.

References

[1] L. A. Fletcher, R. Kasturi, “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images”, *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.10, No.6,* November 1988, pp. 910-918.

[2] Laurence Likforman-Sulem, Anahid Hanimyan, Claudie Faure, “A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents”, *Proceedings of the Third International Conference on Document Analysis and Recognition,* Montreal, Canada, 1995, pp. 774-777.

[3] Y. Pu and Z. Shi, “ A Natural Learning Algorithm Based on Hough Transform for Text Lines Extraction in Handwritten Documents”, *Proceedings of the 6 International Workshop on Frontiers in Handwriting Recognition*, Taejon, Korea,1998, pp. 637-646.

[4] Elisabetta Bruzzone, Meri Cristina Coffetti, “An Algorithm for Extracting Cursive Text Lines”, *Proceedings of the Fifth International Conference on Document Analysis and Recognition,* Bangalore, India, 1999, pp. 749.

[5] R. Manmatha, J. L. Rothfeder, “A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents”, *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.27, No.8,* August 2005, pp. 1212-1225.

[6] Z. Shi, V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Runlength", *First International Workshop on Document Image Analysis for Libraries,* 2004, pp. 306.

[7] Z. Shi, S. Setlur, and V. Govindaraju, “Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map”, *Eighth International Conference on Document Analysis and Recognition,* Seoul, Korea, 2005, pp. 794-798.

[8] Douglas J. Kennard, William A. Barrett, “Separating Lines of Text in Free-Form Handwritten Historical Documents”, *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pp. 12-23.

[9] Yi Li, Yefeng Zheng, David Doermann, “Detecting Text Lines in Handwritten Documents”, *The 18th International Conference on Pattern Recognition (ICPR’06),* pp. 1030-1033.

[10] Aurelie Lemaitre, Jean Camillerapp, “Text Line Extraction in Handwritten Document with Kalman Filter Applied on Low Resolution Image”, *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pp. 38-45.

[11] B. Gatos, I. Pratikakis and S. J. Perantonis, “Adaptive Degraded Document Image Binarization”, *Pattern Recognition, Vol. 39*, pp. 317-327, 2006.

[12] Fu Chang, Chun-Jen Chen, Chi-Jen Lu, “A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique”, *Computer Vision and Image Understanding, Vol. 93, No.2,* February 2004, pp. 206-220.

[13] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis and S. J. Perantonis, “A Segmentation-free Approach for Keyword Search in Historical Typewritten Documents”, *8th International Conference on Document Analysis and Recognition (ICDAR'05)*, Seoul, Korea, August 2005

[14] H.J. Lee, B. Chen “Recognition of Handwritten Chinese Characters via Short Line Segments”, *Pattern Recognition 25 (5),* 1992, pp. 543-552.

[15] A. Antonacopoulos, B. Gatos and D. Karatzas, "ICDAR 2003 Page Segmentation Competition", *7th International Conference on Document Analysis and Recognition (ICDAR'03)*, pp. 688-692, Edinburgh, Scotland, August 2003

[16] A. Antonacopoulos, B. Gatos and D. Bridson, "ICDAR2005 Page Segmentation Competition", *8th International Conference on Document Analysis and Recognition (ICDAR'05)*, pp. 75-79, Seoul, Korea, August 2005

[17] I. Phillips and A. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems," *IEEE Transaction of Pattern Analysis and Machine Intelligence, Vol. 21, No. 9*, pp. 849-870, September 1999.