Pattern Recognition 41 (2008) 3758 – 3772

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Text line detection in handwritten documents

G. Louloudis[a,*], B. Gatos[b], I. Pratikakis[b], C. Halatsis[a]

[a]*Department of Informatics and Telecommunications, University of Athens, 15784 Panepistimiopolis, Llissia, Athens, Greece*
[b]*Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", 153 10 Athens, Greece*

ARTICLE INFO

Article history:
Received 13 April 2007
Received in revised form 26 March 2008
Accepted 2 May 2008

Keywords:
Document image analysis
Handwritten text
Hough transform
Text line detection
Connected component analysis

ABSTRACT

In this paper, we present a new text line detection method for 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, partitioning of the connected component domain into three spatial sub-domains 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 handwritten document recognition that refers to the segmentation of a document image into distinct entities, namely text lines. The overall performance of a handwritten character recognition system strongly relies on the results of the text line detection process. In the case that text line detection does not give good results, this will affect the accuracy of the word segmentation as well as the text recognition procedure. Thus, there is a need for an optimal text line detection stage. Although in printed documents, text line detection is a rather straightforward process, in the case of handwritten documents there exist several challenges. Potential challenges encountered in this stage are the variability in the skew angle between different text lines (see Fig. 1) including the converse skew angles case in the same text line (see Fig. 2), the variability in skew directions (see Fig. 3), the presence of overlapping words (words of adjacent text lines that have overlapping bounding boxes), the adjacent text lines touching and finally, the presence of accents that may appear in certain languages (e.g. French, Greek).

Although many researchers have worked towards solving the handwritten text line detection problem they have not considered all the aforementioned challenges. For example, text line detection methods that rely on the projection profile [1–3] cannot confront with the problem of variable skew angles between different text lines and along the same text line as well as with the efficient distinction of vertically connected characters. Also, in most cases, ascenders and descenders from adjacent text lines are erroneously separated due to the presence of overlapping words. In the Hough-based approaches [4–6] although the problem of variability in the skew angle between different text lines can be solved, situations of different skew angles along the same text line are difficult to handle. Also, the separation of vertically connected characters is not considered in these approaches. Similar problems arise to other methodologies as well [7,9,16].

The proposed text line detection methodology for handwritten documents strives towards dealing with all aforementioned challenges. In particular, we present a novel methodology that comprises (i) an efficient block-based Hough transform in which voting occurs on the basis of equally spaced blocks after splitting of the connected components' bounding box; (ii) a partitioning of the connected component domain into three spatial sub-domains which impose a different strategy to the processing of the corresponding connected components, and (iii) the efficient separation of vertically connected parts of text lines.

The remainder of the paper is organized as follows. In Section 2, the related work is described. In Section 3, the proposed methodology for text line detection is detailed. Section 4 deals with the evaluation and the experimental results, and finally, in Section 5 concluding remarks are presented.

* Corresponding author. Tel.: +30 210 7275241; fax: +30 210 7275201.
E-mail addresses: louloud@mm.di.uoa.gr (G. Louloudis), bgat@iit.demokritos.gr (B. Gatos), ipratika@iit.demokritos.gr (I. Pratikakis), halatsis@di.uoa.gr (C. Halatsis)
URLs: http://www.di.uoa.gr, http://www.iit.demokritos.gr/cil.

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.05.011

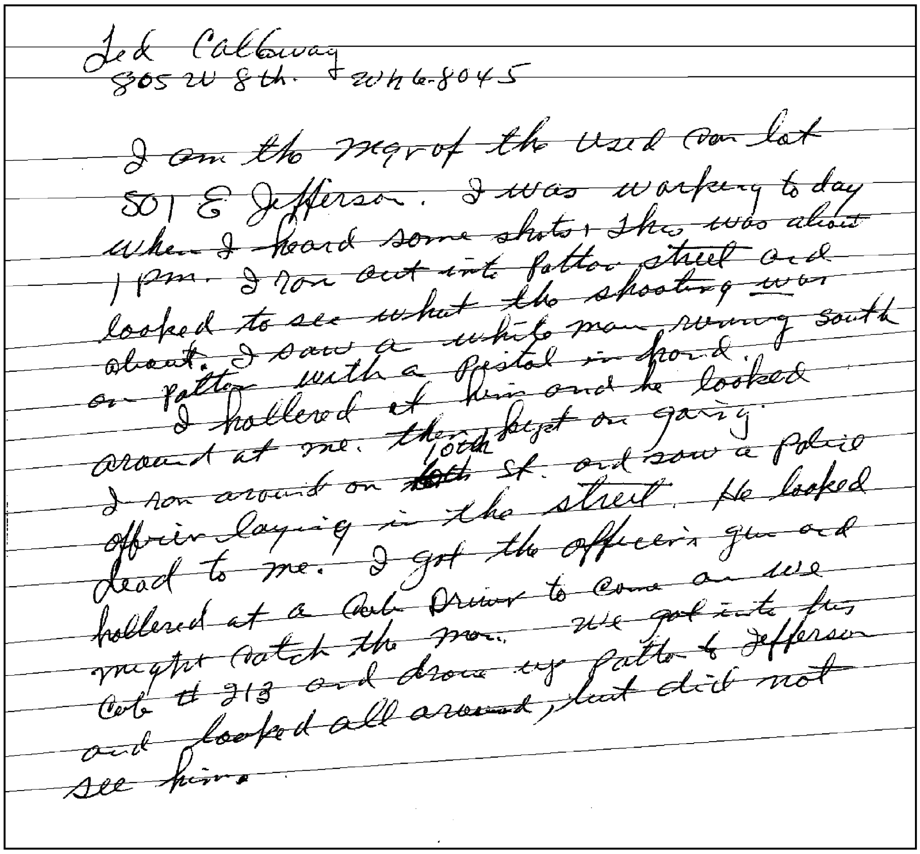
Ted Calloway
805 W 8th. 246-8045
I am the mgr of the used car lot
501 E Jefferson. I was working today
when I heard some shots & this was about
1 pm. I ran out into Patton street and
looked to see what the shooting was
about. I saw a white man running south
on Patton with a pistol in hand.
I hollered at him and he looked
around at me. then kept on going.
I ran around on 10th St. and saw a police
officer laying in the street. He looked
dead to me. I got the officer's gun and
hollered at a Cab Driver to come on we
might catch the man. We got into his
Cab # 213 and drove up Patton & Jefferson
and looked all around, but did not
see him.

Fig. 1. Example of an image wherein text lines have different skew angles along the page.

2. Related work

A wide variety of text line detection methods for handwritten documents has been reported in the literature. There are four categories that these methods may belong to: (i) methods that use projection profiles; (ii) methods based on the Hough transform; (iii) smearing methods; and (iv) other methods, which cannot be grouped in a unique category since they do not share a common guideline.

Methods that make use of the projection profiles include Refs. [1–3]. In Ref. [1], the projection profile is calculated by summing up intensities from all pixels found at each scan line. The corresponding profile is smoothed and the produced valleys are identified. These valleys indicate the space between the lines of the text. A different approach is considered in Ref. [2] where the initial image is partitioned into vertical strips. At each vertical strip, the histogram of horizontal runs is calculated. This technique assumes that text appearing in a single strip is almost parallel to each other. Srihari et al. [3] partition the initial image into vertical strips called chunks. The projection profile of every chunk is calculated. The first candidate lines are extracted among the first chunks. These lines traverse around any obstructing handwritten connected component by associating it to the text line above or below. This decision is made by either (i) modeling the text lines as bivariate Gaussian densities and evaluating the probability of the component for each Gaussian or (ii) the probability obtained from a distance metric.

Methods that make use of the Hough transform include Refs. [4–6]. The Hough transform is a powerful tool used in many areas of document analysis that is able to locate skewed lines of text. By starting from some points of the initial image the method extracts the lines that fit best to these points. The points considered in the voting procedure of the Hough transform are usually either the gravity centers [4,5] or minima points [6] of the connected components.

In more detail, Likforman [5] developed a method on a hypothesis-validation scheme. Potential alignments are hypothesized in the Hough domain and validated in the image domain. The centroids of the connected components are the units for the Hough transform. A set of aligned units in the image along a line with parameters (ρ, θ) is included in the corresponding cell (ρ, θ) of the Hough domain. Alignments including a lot of units, correspond to high peaked cells of the Hough domain. To take into account fluctuations of handwritten text lines, i.e., the fact that units within a text line are not perfectly aligned, two hypotheses are considered for each alignment and an alignment is formed from units of the *cell structure* of a *primary cell*. A cell structure of a cell (ρ, θ) includes all the cells lying in a cluster centered around (ρ, θ). Consider the cell $(\rho 0, \theta 0)$ having the greatest count of units. A second hypothesis $(\rho 1, \theta 1)$ is searched in the cell structure of $(\rho 0, \theta 0)$. The alignment chosen between these two hypotheses is the strongest one, i.e., the one which includes the highest number of units in its cell structure. Finally, a potential alignment is validated (or invalidated) using contextual information, i.e., considering its *internal* and *external* neighbors. An internal neighbor of a unit j is a within-Hough alignment neighbor. An external neighbor is a out of Hough alignment neighbor which lies within a circle of radius δj from unit j. Distance δj is the average distance of the internal neighbor distances from unit j. To be validated, a potential alignment may contain fewer external units than internal ones. This enables the rejection of alignments which have no perceptual relevance.

The Hough transform can also be applied to fluctuating lines of handwritten drafts such as in Ref. [6]. The Hough transform is first

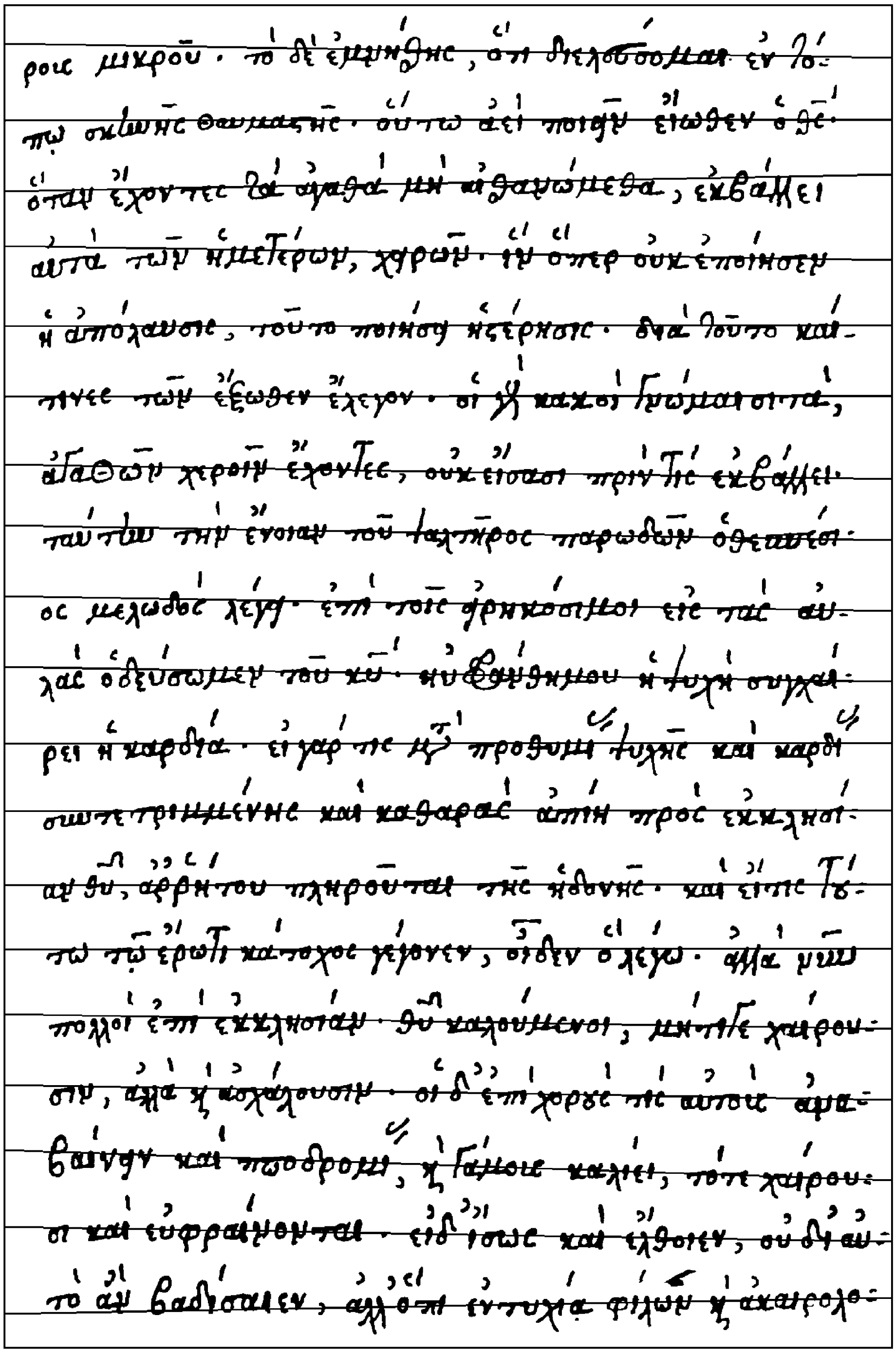

Fig. 2. Example of an image wherein text lines have converse skew angles.

applied to minima points (units) in a vertical strip on the left of the image. The alignments in the Hough domain are searched starting from a main direction, by grouping cells in an exhaustive search in six directions. Then a moving window, associated with a clustering scheme in the image domain, assigns the remaining units to alignments. The clustering scheme (*natural learning algorithm*) allows the creation of new lines starting in the middle of the page.

Smearing methods include the fuzzy RLSA [7] and the adaptive RLSA [8]. The fuzzy RLSA 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. The adaptive RLSA [8] is an extension of the classical RLSA in the sense that additional smoothing constraints are set in regard to the geometrical properties of neighboring connected components. The replacement of background pixels with foreground pixels is performed when these constraints are satisfied.

Other methodologies include the work of Nicolas et al. [9]. In this work, the text line extraction problem is seen from an artificial intelligence perspective. The aim is to cluster the connected components of the document into homogeneous sets that correspond to the

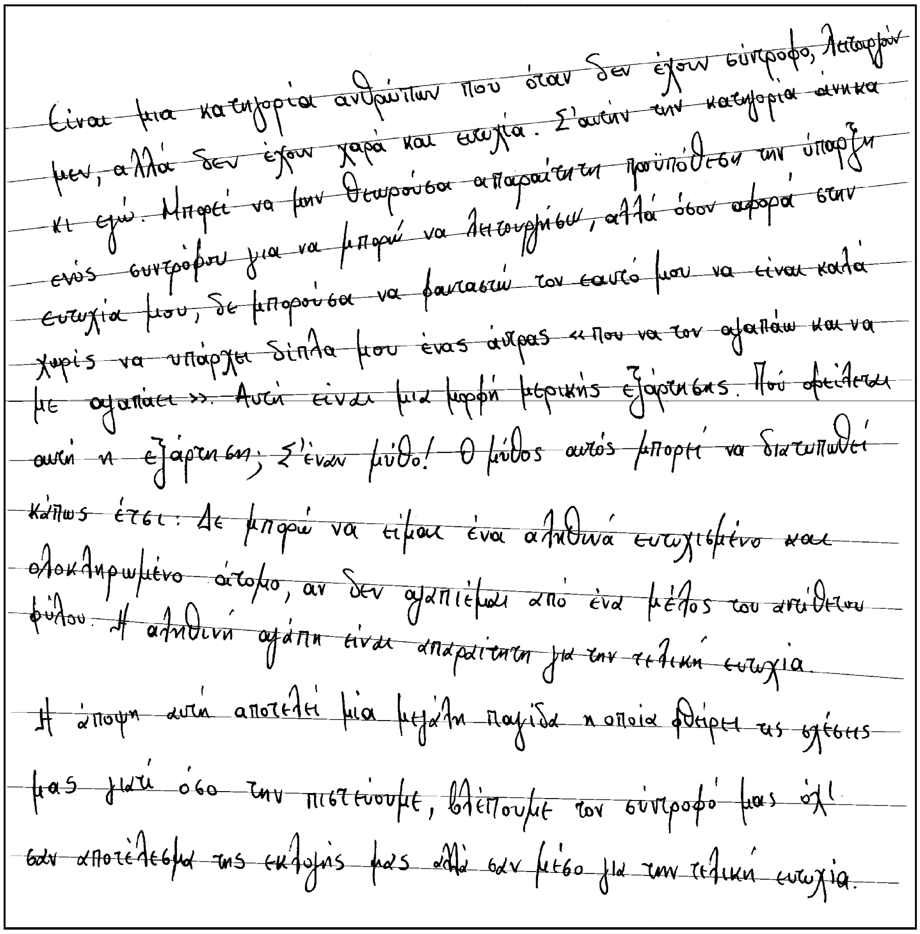

Fig. 3. Example of an image wherein text lines have both different skew direction along the page and converse skew angles.

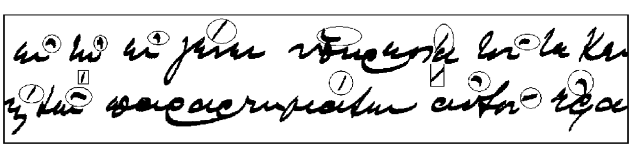

Fig. 4. An example of accents (in boxes) that are situated below the text line. All other accents are closed in ellipses.

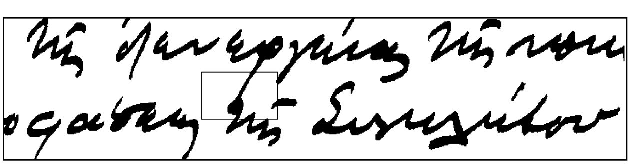

Fig. 5. An image portion that demonstrates adjacent touching text lines (in box).

text lines of the document. To solve this problem, a search over the graph that is defined by the connected components as vertices and the distances among them as edges is applied. A recent paper [10], makes use of the Adaptive Local Connectivity Map. The input to the method is a grayscale image. A new image is calculated by summing the intensities of each pixel's neighbors in the horizontal direction. Since the new image is also a grayscale image, a thresholding technique is applied and the connected components are grouped into location maps by using a grouping method. In Ref. [11], 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. Also, a min-cut/max-flow graph cut algorithm is used to split up text areas that appear to encompass more than one line of text. A merging of text lines containing relatively little text information to nearby text lines is then applied. Li [12] 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 Ref. [13] 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 the theory of Kalman filtering to detect text lines on low resolution images. Weliwitage et al. [14] describe a method that involves cut text minimization for segmentation of text lines from handwritten English documents. In doing so, an optimization technique is applied which varies the cutting angle and start location to minimize the text pixels cut while tracking between two text lines. In Ref. [15], a text line extraction technique is presented for multi-skewed document of handwritten English or Bengali text. It assumes that hypothetical water flows, from both left and right sides of the image frame, face obstruction from characters of text lines. The stripes of areas left unwetted on the image frame are finally labeled for extraction of text lines. Finally, in Ref. [16], a text line segmentation method for printed or handwritten historical Arabic is presented. Documents are first classified into two classes using a K-means scheme. These classes correspond to document complexity (easy or not easy to segment). A document which includes overlapping and touching characters is

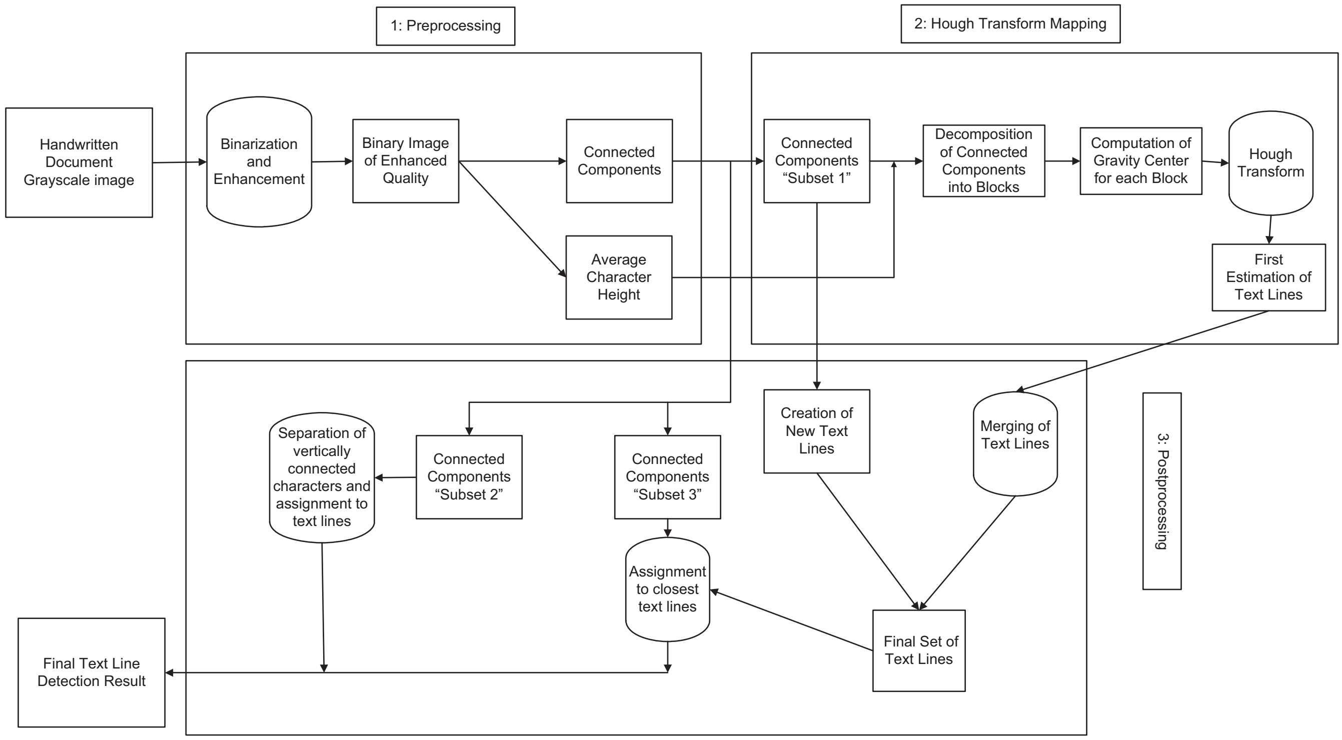

Fig. 6. Block diagram of the proposed method.

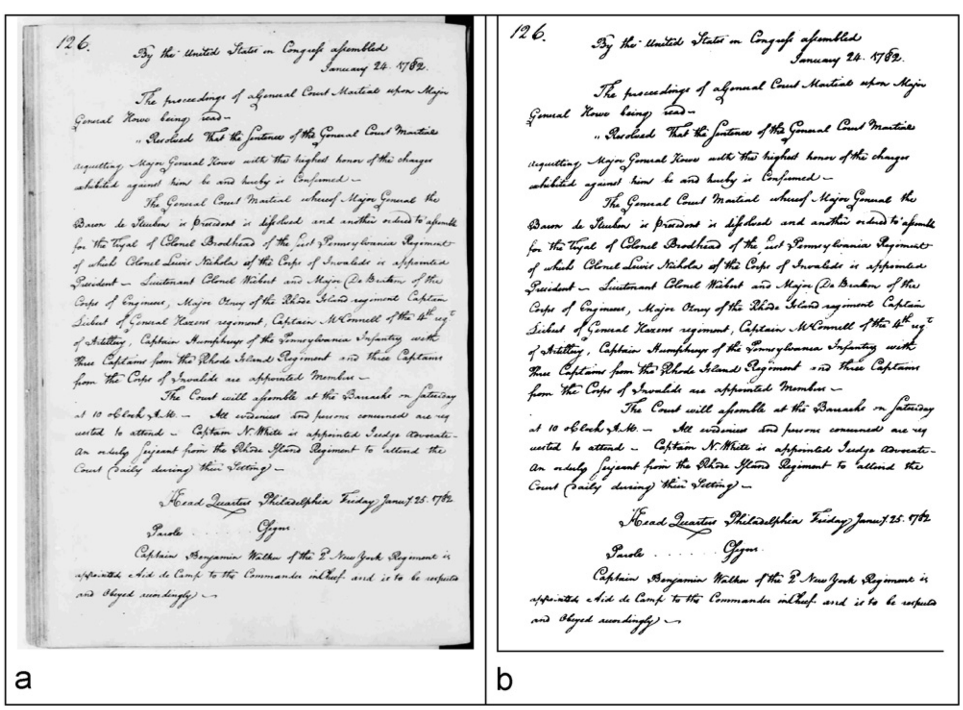

Fig. 7. (a) Original grayscale document image and (b) the resulting image after binarization.

divided into vertical strips. The extracted text blocks obtained by the horizontal projection are classified into three categories: small, average and large text blocks. After segmenting the large text blocks, the lines are obtained by matching adjacent blocks within two successive strips using spatial relationship. The document without overlapping or touching characters is segmented by making abstraction on the segmentation module of the large blocks. The authors claim 96% accuracy on a collection of 100 historical documents.

Although the abovementioned techniques have been proved efficient for certain problems, there are more challenges found in a text line detection process. For example, none of the above techniques deals with the problem of accents. Although accents do not appear

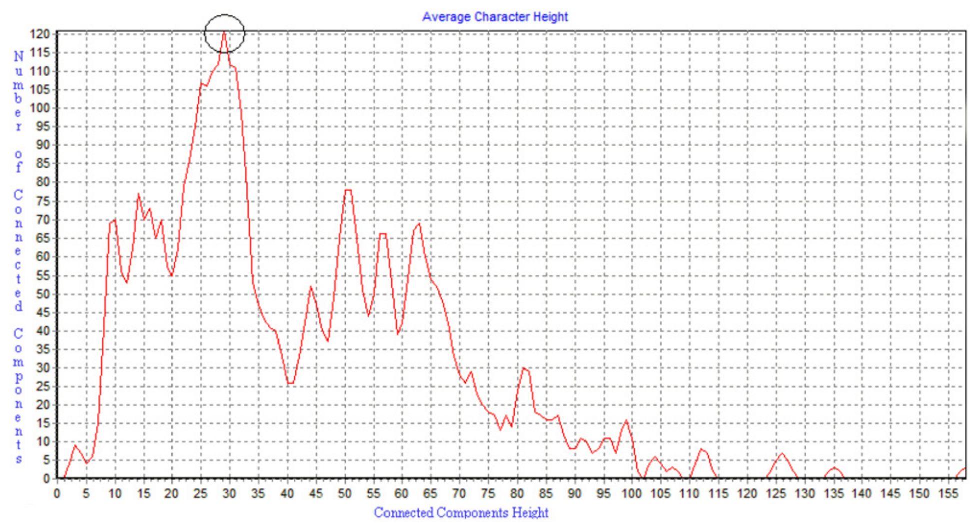

Fig. 8. The histogram of the connected components height. The maximum of histogram (in circle) corresponds to the average character height $AH = 28$.

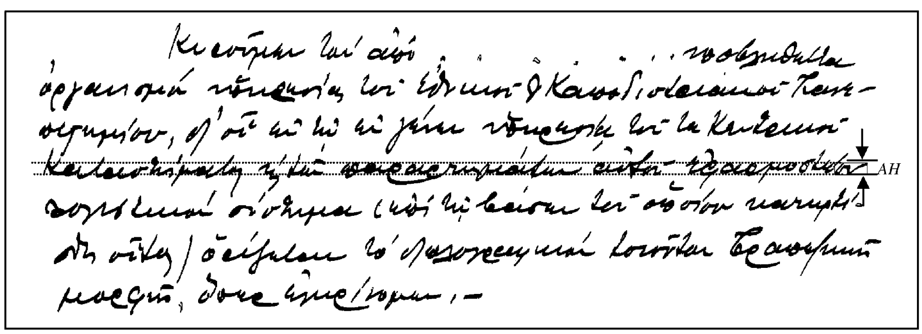

Fig. 9. The area delimited by the two lines indicates the computed connected components average height *AH*.

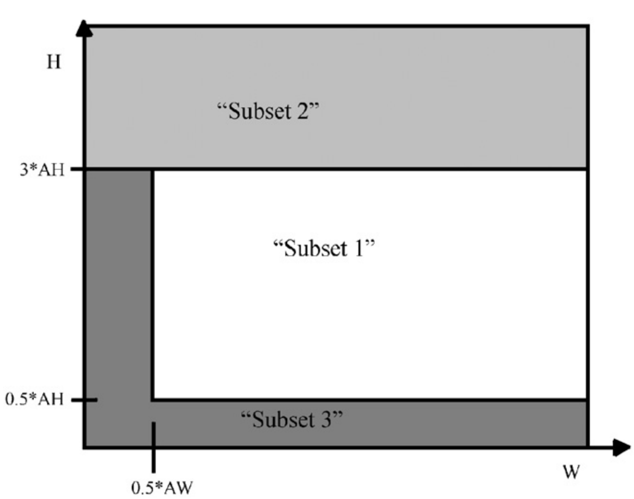

Fig. 10. The connected component domain partitioned to three sub-domains denoted as "Subset 1", "Subset 2", and "Subset 3", respectively.

in English documents it is a common constituent in documents of French or Greek language. Furthermore, most of these techniques do not strive towards solving the problem of vertically connected characters that results to text line merging (see Refs. [1,5,7]). In some of the above techniques [1,8,9], an assumption is made that all text lines have no skew angle or they have the same skew angle. Finally, in the Hough transform-based approach of Ref. [5], only one point from every connected component votes in the Hough domain. This may cause a serious problem in cursive multi-accented documents where one connected component can be a whole word. In this case, a whole word and a small accent have the same contribution in the Hough domain and this may lead to erroneous results. As will be described in the following section, our approach aims to solve the aforementioned challenges in an efficient way.

3. Methodology

The proposed methodology for text line detection in handwritten document images deals with the following challenges: (i) each text line that appears in the document may have an arbitrary skew angle (Fig. 1) and converse skew angle along the text line (Figs. 2 and 3); (ii) text lines may have different skew directions (Fig. 3); (iii) accents may be cited either above or below the text line (Fig. 4), and (iv) parts of neighboring text lines may be connected (Fig. 5).

To meet the aforementioned challenges, we propose a methodology which consists of three main steps (see Fig. 6). The first step includes binarization and image enhancement, connected component extraction, average character height estimation, and partitioning of the connected component domain into three distinct spatial sub-domains. 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 possible text lines which

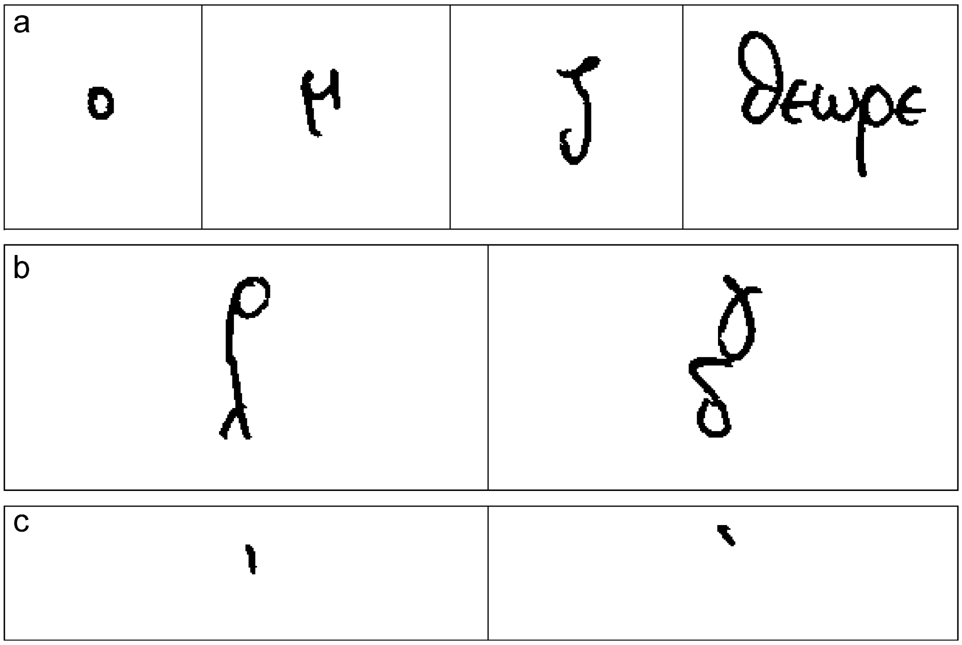

Fig. 11. Example connected components that are attributed to: (a) "Subset 1", (b) "Subset 2", and (c) "Subset 3".

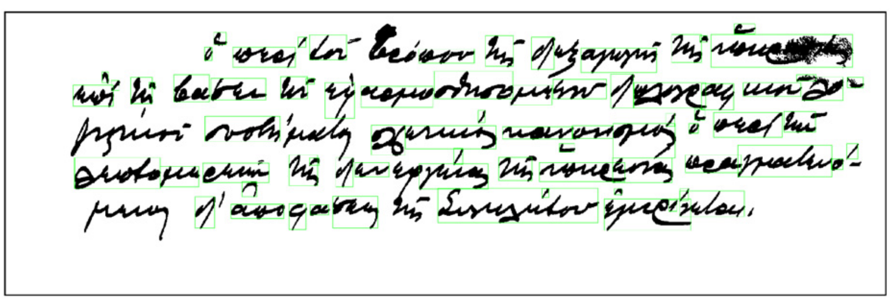

Fig. 12. Example connected components that are attributed to "Subset 1".

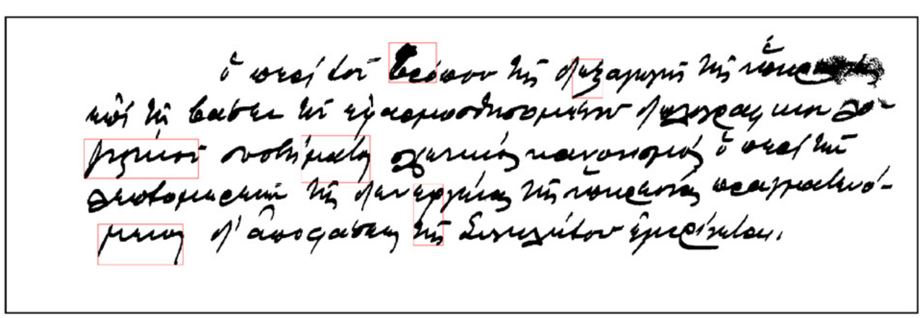

Fig. 13. Example connected components that are attributed to "Subset 2".

the previous step did not reveal, and, finally, to separate vertically connected parts and assign them to text lines. A detailed description of these stages is given in the following Sections 3.1–3.3.

3.1. Pre-processing

The preprocessing step consists of four stages. First, an adaptive binarization and image enhancement technique described in Ref. [17] is applied (see Fig. 7). Then, the connected components of the binary image are extracted [18] and the bounding box coordinates for each connected component are calculated. The average character height *AH* for the whole document image is calculated [19] (Figs. 8 and 9). We assume that the average character height equals to the average character width *AW*.

The connected components domain includes components of a different profile with respect to width and height since it is frequent to have components describing one character, multiple characters, a whole word, accents, and characters from adjacent touching text

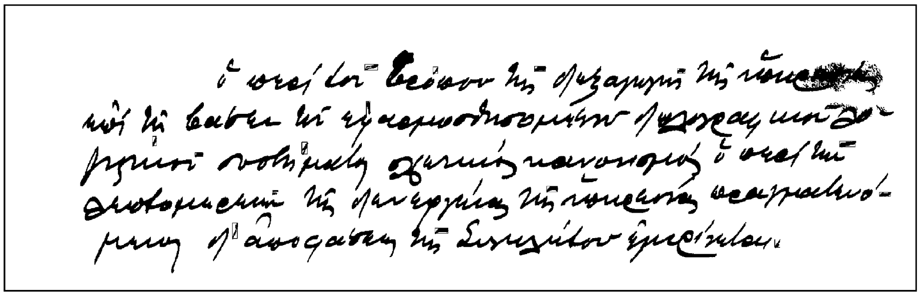

Fig. 14. Example connected components that are attributed to "Subset 3".

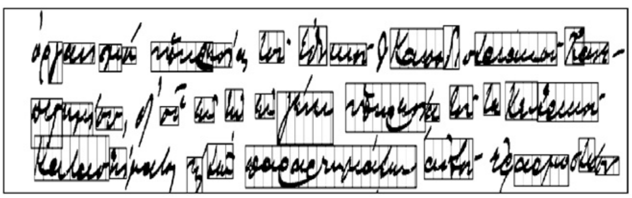

Fig. 15. Example that shows the "Subset 1" connected components partitioning into blocks of width *AW*.

lines. The aforementioned connected components variability has motivated us to divide the connected component domain into different sub-domains in order to deal with these categories separately. More specifically, in the proposed approach, we divide the connected components domain into three distinct spatial sub-domains denoted as "Subset 1", "Subset 2", and "Subset 3" (Fig. 10).

"Subset 1" (Figs. 11 and 12) contains all components which correspond to the majority of the characters with size which satisfies the following constraints:

$$(0.5 * AH \leqslant H < 3 * AH) \quad \text{AND} \quad (0.5 * AW \leqslant W) \tag{1}$$

where H and W denote the component's height and width, respectively, and AH, AW denote the average character height and the average character width, respectively. The motivation for 'Subset 1' definition is to exclude accents and components that are large in height and belong to more than one text line. As it is demonstrated in Fig. 11(a), connected components that contain ascenders and/or descenders will be included in this sub-domain.

"Subset 2" (Figs. 11 and 13) contains all large connected components. Large components are either capital letters or characters from adjacent text lines touching. The size of these components is described by the following equation:

$$H \geqslant 3 * AH \tag{2}$$

The motivation for 'Subset 2' definition is to grasp all connected components that exist due to touching text lines. We assume that the corresponding height will exceed three times the average character height (Fig. 11(b)).

Finally, "Subset 3" (Figs. 11 and 14) should contain characters as accents, punctuation marks, and small characters. The equation describing this set is:

$$\begin{aligned}&((H < 3 * AH) \quad \text{AND} \quad (0.5 * AW > W)) \quad \text{OR}\\&((H < 0.5 * AH) \quad \text{AND} \quad (0.5 * AW < W))\end{aligned} \tag{3}$$

The motivation for 'Subset 3' definition is that accents usually have width less than half the average character width or height less than half the average character height (Fig. 11(c)).

3.2. Hough transform mapping

In this step, the Hough transform takes into consideration only connected components that belong to sub-domain "Subset 1". Selection of this sub-domain (Fig. 10) is done for the following reasons: (i) it is guaranteed that components, which appear in more than one text line, will not vote in the Hough domain; (ii) it rejects components, such as accents, which have a small size. This avoids false text line detection by connecting all the accents above the core text line.

In our approach, instead of having only one representative point for every connected component (as in Refs. [4,5]), a partitioning is applied for each connected component lying in "Subset 1", in order to have more representative points voting in the Hough domain. In particular, every connected component lying in this subset is partitioned to equally sized blocks. The number of the blocks is defined by the following equation:

$$N_{\mathrm{b}} = \left\lceil \frac{W_{\mathrm{c}}}{AW} \right\rceil \tag{4}$$

where W_c denotes the width of the connected component and AW the average character width of all connected components in the image. It is easily observed that the width of each block is fixed and equals to the average character width AW with an exception of the last block which can be either equal to AW or smaller. An example of this partitioning is shown in Fig. 15. After this stage, we calculate the gravity center of the connected component contained in each block which is used in the voting procedure of the Hough transform.

The Hough transform [20] is a line to point transformation from the Cartesian space to the Polar coordinate space. A line in the Cartesian coordinate space is denoted as:

$$x \cos(\theta) + y \sin(\theta) = p \tag{5}$$

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 that was created following the previous procedure 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° letting θ take values in the range of $85–95^\circ$ while the resolution along p direction was set to $0.2 * AH$ [4].

After the computation of the accumulator array we proceed to the following procedure: we detect the cell (p_i, θ_i) having the maximum contribution and we assign to the text line (p_i, θ_i) all points that vote in the area $(p_i - 5, \theta_i), \ldots, (p_i + 5, \theta_i)$. 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, we remove from the Hough transform accumulator array all votes that correspond to this particular connected component. This procedure is repeated until the cell (p_i, θ_i) having the maximum contribution contains less than n_1 votes in order to avoid false detections. During

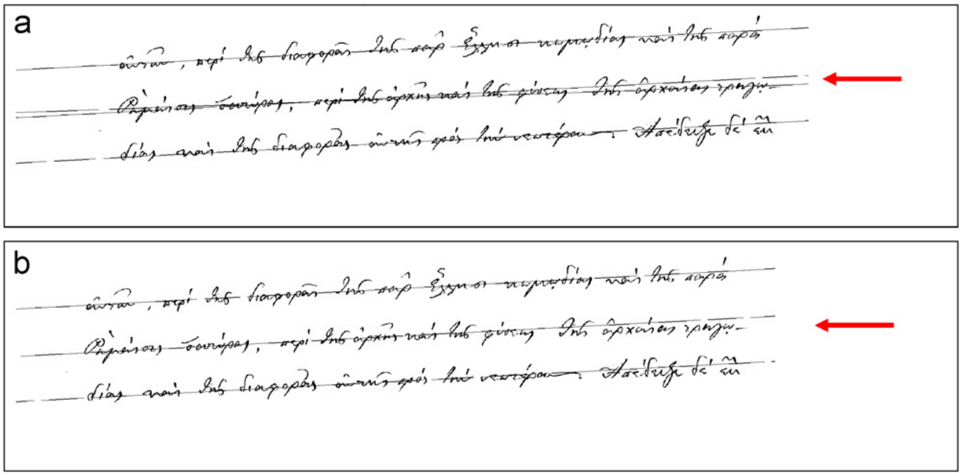

Fig. 16. (a) Two lines that are to be merged to define one text line (arrow) and (b) the result after the merging procedure.

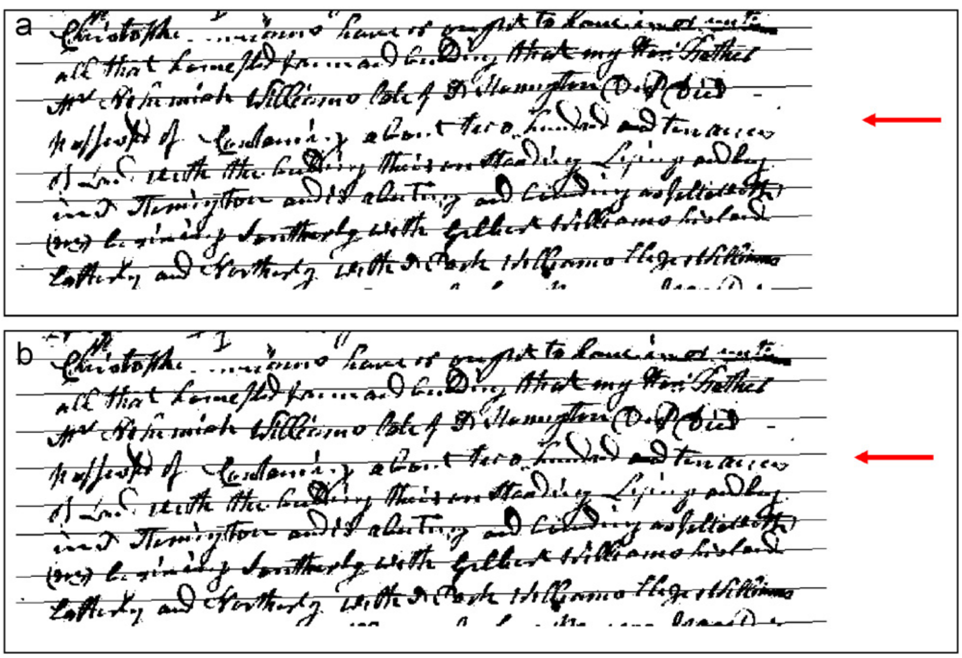

Fig. 17. (a) Part of a document image after the merging procedure. One text line is not detected (see arrow). (b) The creation of lines procedure correctly detects the text line.

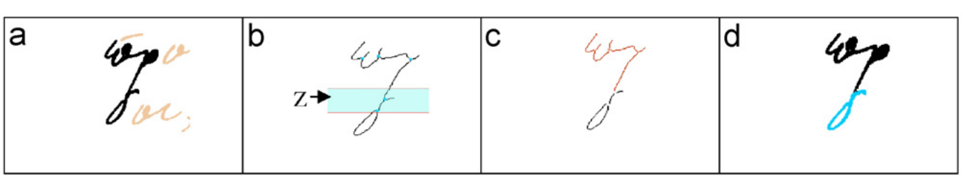

Fig. 18. Separating vertically connected characters: (a) a connected component that is likely to belong to vertically connected characters appearing into two successive text lines; (b) image skeleton, detected junction points and segmentation zone Z; (c) flagging the upmost skeleton component after removing all pixels in the 3×3 neighborhood of all junction points that lie inside the segmentation zone Z, and (d) final separation of the vertically connected characters.

the evolution of the procedure, the dominant skew angle of currently detected lines is calculated. In the case that a cell (p_i, θ_i) has 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. Post-processing

The post-processing step consists of two stages. At the first stage, (i) a merging technique over the result of the Hough transform is applied to correct possible false alarms (see Fig. 16) and (ii) connected components of "Subset 1" that were not clustered to any text

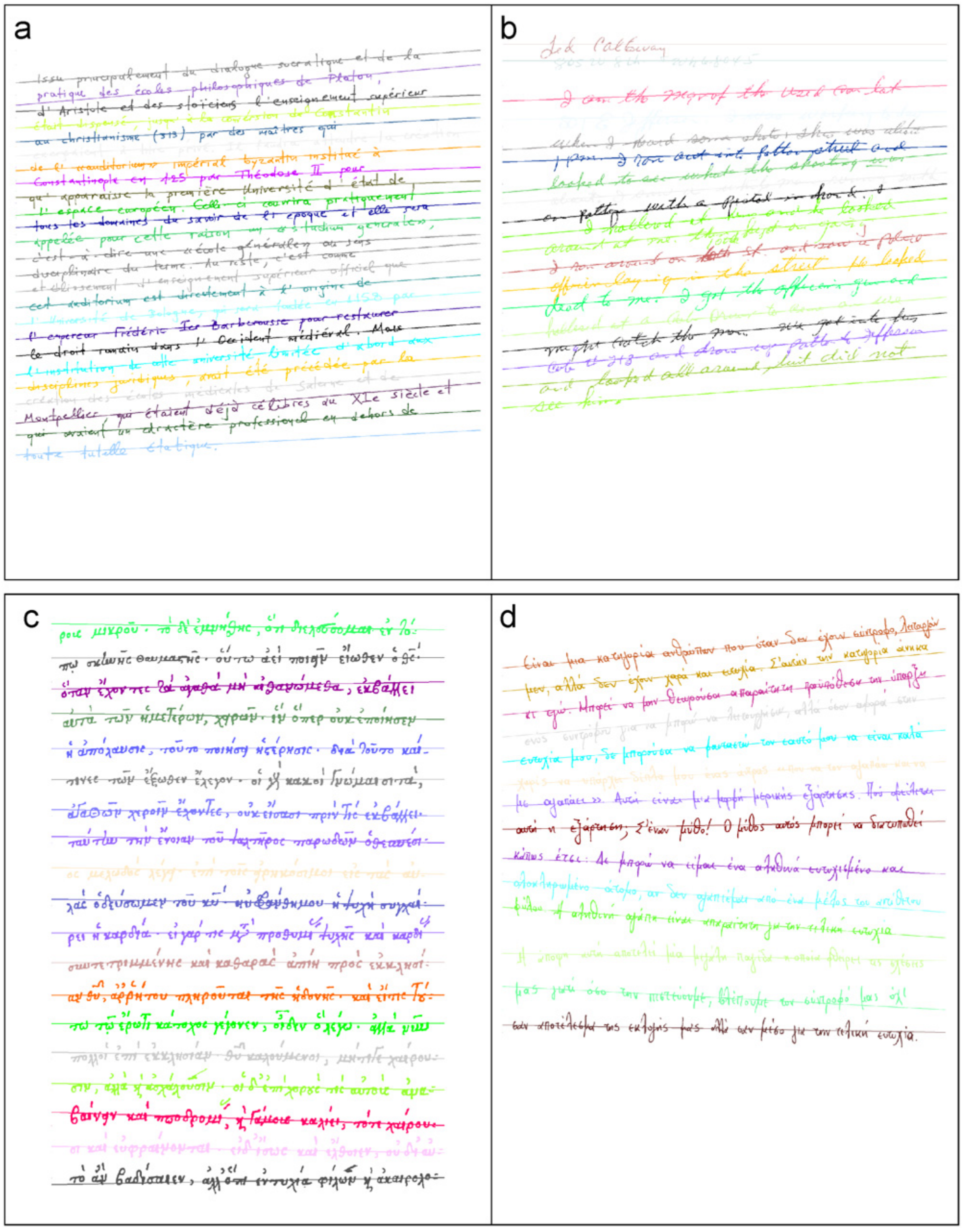

Fig. 19. The resulting text line detection of the proposed method on (a) French document; (b) English document; (c) and (d) Greek documents. Corresponding indicative 'one-to-one matches' are: (a) 25/25, (b) 18/19, (c) 19/19, and (d) 14/14.

line are checked to see whether they create a new text line that the Hough transform did not reveal.

Let $y = a_i x + b_i, \quad i \in \{1 \ldots N\}$ the set of lines extracted from the Hough transform in a top down order. The intersection of every line with the middle vertical line of the documents defines a set of points $A_i(x_i, y_i)$, where $x_i = I_x/2$ with I_x denoting the document image width and $y_i = a_i * x_i + b_i$. The average distance of adjacent lines is described by the following equation:

$$A_d = \frac{\sum_{i=1}^{N-1} |y_{i+1} - y_i|}{N-1} \tag{6}$$

For every pair of adjacent lines we calculate the distance $d = |y_{i+1} - y_i|$. If $d < A_d$ then all connected components which correspond to these lines are assigned to the same text line label.

Then, a grouping technique of the remaining connected components of Subset 1 is applied that utilizes the gravity centers of the corresponding blocks that result after the partitioning procedure described at the previous section. For every block with gravity center (x_j, y_j), we calculate the distance *dis* between (x_j, y_j) and the closest already detected line. This distance is defined as $dis = \min_i(|y_i - y_j|)$ where $y_i = a_i * x_j + b_i$ and $i = 1 \ldots N$. If distance *dis* has a value close to the average distance of adjacent lines ($dis \succ 0.9 * A_d$) then the corresponding block is considered as a candidate to belong to a new text line. To decide whether a connected component is assigned to a new text line, at least half of the corresponding blocks must be candidates to belong to the new text line.

Fig. 17 shows an image example describing the creation of new lines stage.

After the creation of the final set of lines, components lying in "Subset 3", which are usually punctuation marks or accents, as well as the unclassified components of "Subset 1" are grouped to the closest line.

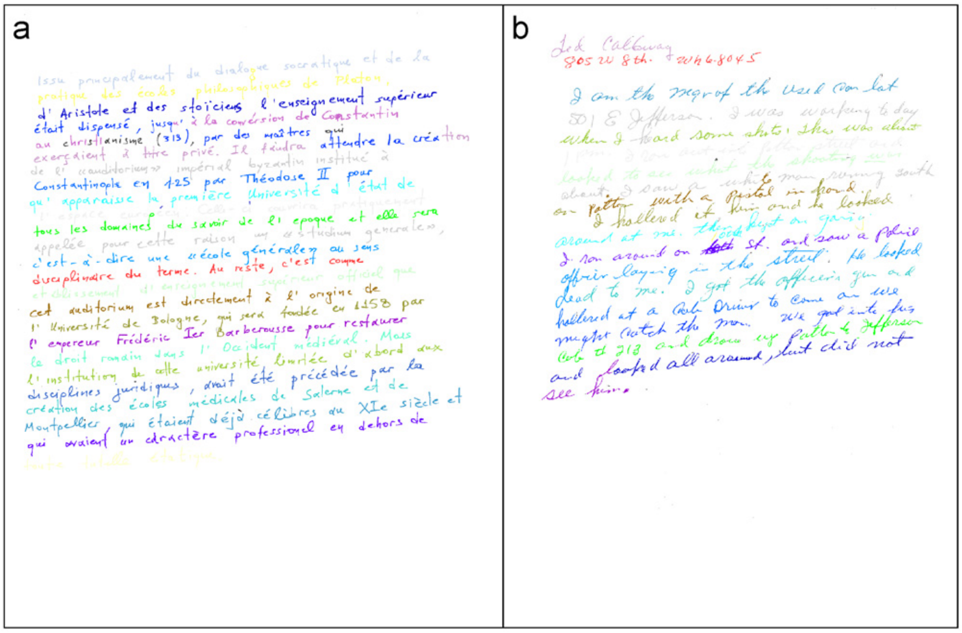

Fig. 20. Results of the Hough approach method [5] on the images that were used in here. Corresponding indicative 'one-to-one matches' in (a) are 22/25 while in (b) are 11/19.

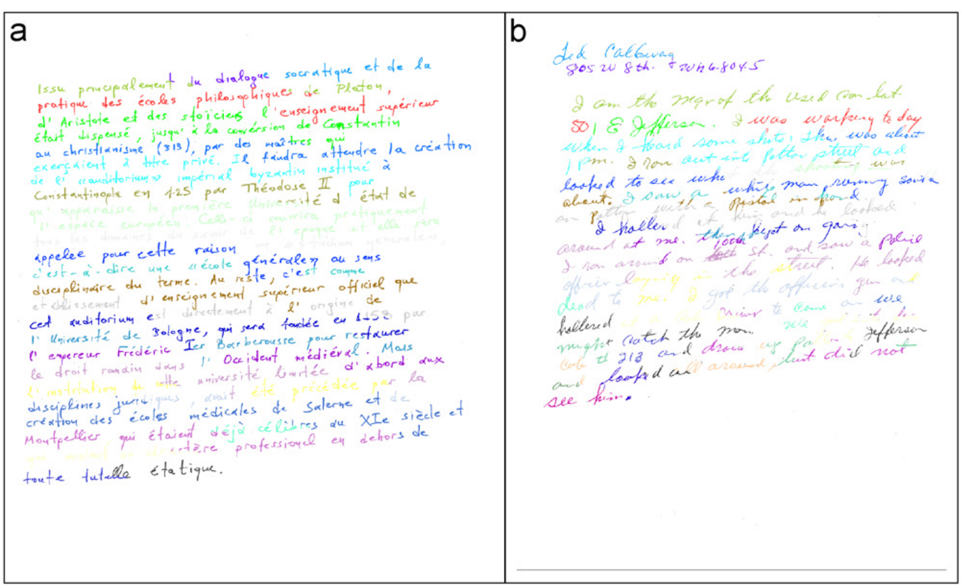

Fig. 21. Results of the fuzzy run-length approach [7] on the document images used in Fig. 20. Corresponding indicative 'one-to-one matches' in (a) are 0/25 while in (b) are 3/19.

The second post-processing stage deals with components lying in "Subset 2". This subset includes components whose height exceeds three times the average height *AH* (see Figs. 10 and 11(b)). All components of this subset mainly belong to two different text lines (see Fig. 18a). 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 [21] (see Fig. 18b).

Step 2: Define the segmentation zone *Z* according to the constraints $h_c/2 < y < 3 * h_c/2$ where h_c is the height of the connected component (see Fig. 18b).

Step 3: Remove from the skeleton image all pixels in the 3×3 neighborhood of all junction points that lie inside the segmentation zone *Z* (see Fig. 18c). If no junction point is present in the area, remove the skeleton pixels in the middle of the zone *Z*.

Step 4: Extract the connected components of the skeleton image [18] and flag the upmost component (see Fig. 18c).

Step 5: Separation of the initial connected component into two different segments is accomplished by examining if a pixel is closer to a flagged or to a non-flagged skeleton pixel of Step 4 (see Fig. 18d).

After the final step, the pixels of the two separated segments are assigned to two different text lines.

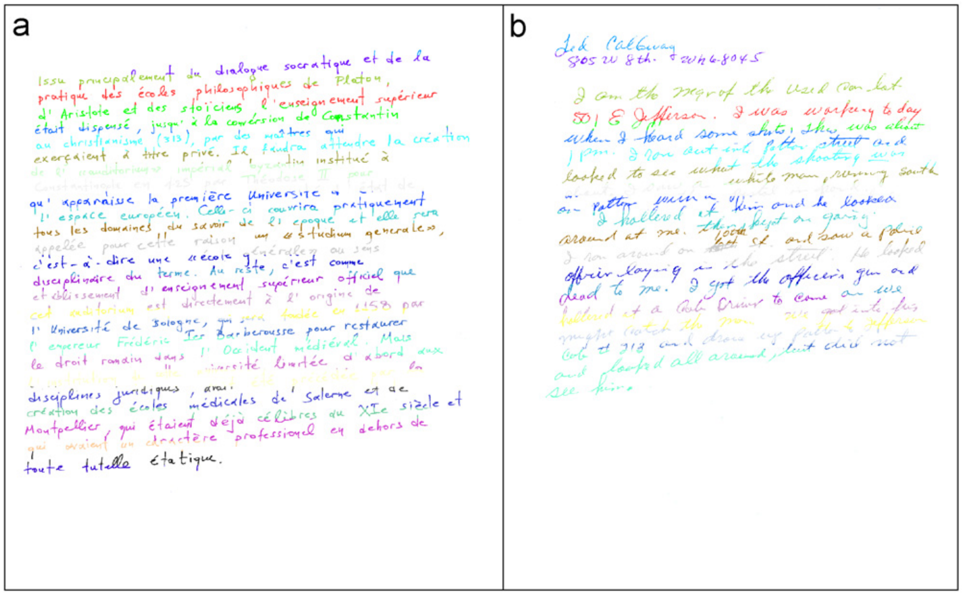

Fig. 22. Results of the projection profile approach [1] on the document images used in Fig. 20. Corresponding indicative 'one-to-one matches' in (a) are 0/25 while in (b) are 2/19.

4. Evaluation and experimental results

4.1. 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 [5,7,10]. 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, and ICDAR2005 page segmentation competitions [22,23]. 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 overlap of the labeled pixel sets as text lines and the ground truth.

Let I be 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:

$$\mathrm{MatchScore}(\mathbf{i},\mathbf{j}) = \frac{T(G_i \cap R_j \cap I)}{T((G_i \cup R_j) \cap I)} \tag{7}$$

The performance evaluator searches within the MatchScore table for pairs of one-to-one matches. We call a pair a one-to-one match if the matching score for this pair is equal to or above the evaluator's acceptance threshold *th*. A g_one-to-many match is a ground-truth text line that "partially" matches with two or more text lines in the detected result. A g_many-to-one match corresponds to two or more ground truth text lines that "partially" match with one detected text line. A *d*_one-to-many match is a detected text line that "partially" matches two or more text lines in the ground truth. Finally, a *d*_many-to-one match corresponds to two or more detected text lines that "partially" match one text line in the ground truth.

Table 1
Comparative experimental results

	Detection rate (%)	Recognition accuracy (%)	TLDM (%)
Fuzzy run-length [7]	83	78.5	80.7
Projection profiles [1]	70.2	73.7	71.9
Hough approach [5]	86.6	74.2	80
Proposed method	95.8	93.8	94.8

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, and w_6 are pre-determined weights, we can calculate the detection rate and recognition accuracy for as follows:

$$\mathrm{Det} = w_1 \frac{\mathrm{one2one}}{N} + w_2 \frac{\mathrm{g\_one2many}}{N} + w_3 \frac{\mathrm{g\_many2one}}{N} \tag{8}$$

$$\mathrm{Rec} = w_4 \frac{\mathrm{one2one}}{M} + w_5 \frac{d\mathrm{\_one2many}}{M} + w_6 \frac{d\mathrm{\_many2one}}{M} \tag{9}$$

where the entities one2one, g_one2many, g_many2one, *d*_one2many, and *d*_many2one are calculated from MatchScore table (Eq. (7)) following the steps of Ref. [24] and correspond to the number of one-to-one, g_one-to-many, g_many-to-one, *d*_one-to-many, and *d*_many-to-one, respectively.

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):

$$\mathrm{TLDM} = \frac{2 * \mathrm{Det} * \mathrm{Rec}}{\mathrm{Det} + \mathrm{Rec}} \tag{10}$$

4.2. Experimental results

The proposed text line detection method is tested on 152 handwritten English, Greek, French, and German documents originated from the historical archive of the University of Athens [25], from the Dallas library [26], and from the Handwriting Segmentation Competition of ICDAR2007 [8]. None of the documents include any non-text

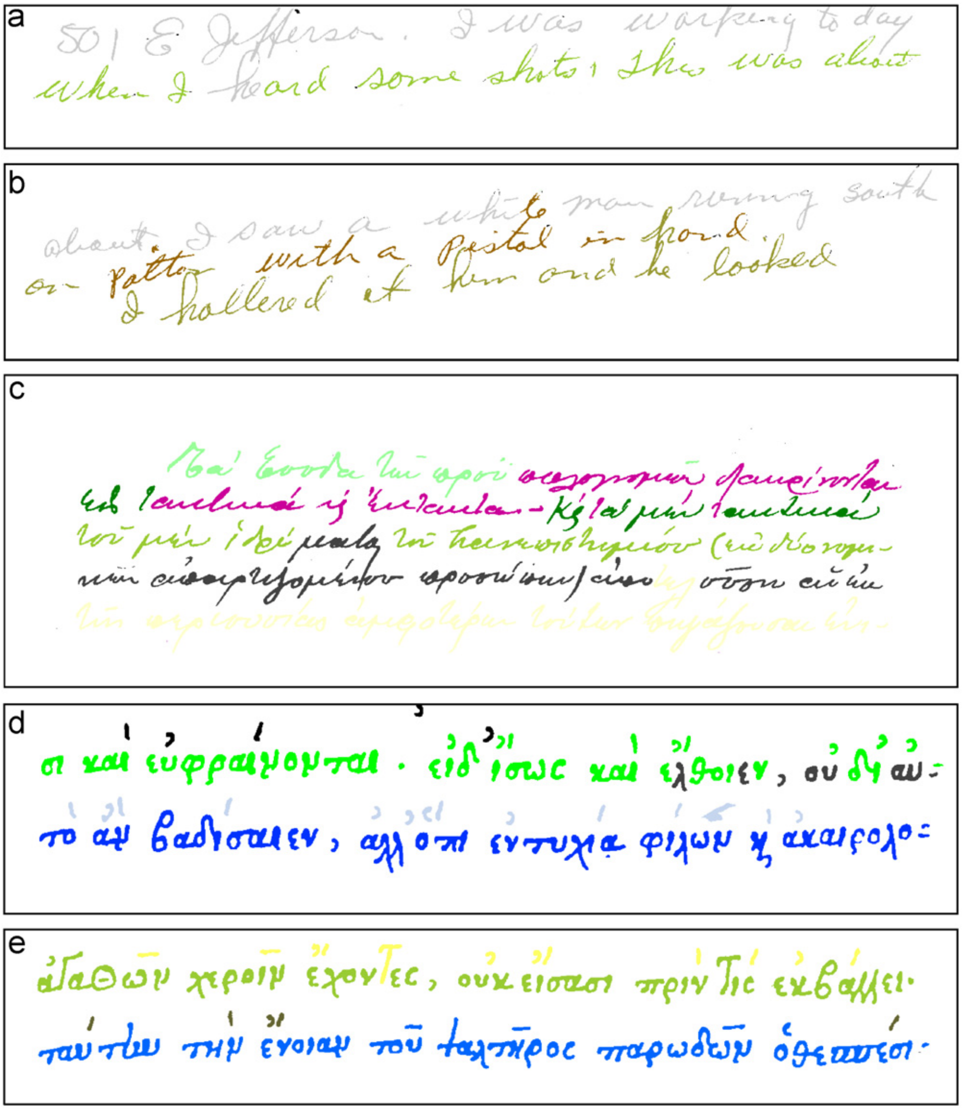

Fig. 23. Indicative errors of Hough approach [5]. (a)–(c) Merging of adjacent text line parts and lack of separation between vertically connected characters; (d) and (e) accents above the text line are considered as separate text lines.

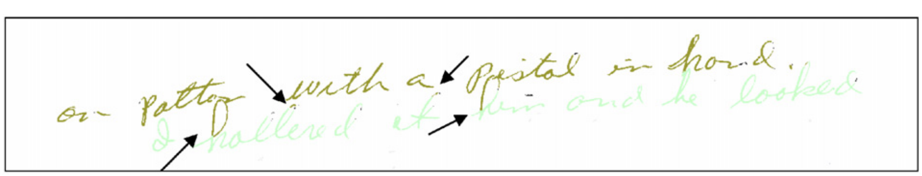

Fig. 24. Indicative errors of the proposed approach. The four arrows indicate the incorrect separation of vertically connected characters as well as small parts of the broken characters 'l' and 'f' that were incorrectly classified to the upper text line.

elements (such as lines, drawings, pictures, and logos). Almost all documents have two or more adjacent text lines touching in several areas. Part of them has constant skew among text lines while others have variable skew angles among text lines. Furthermore, there are document images having text lines with different skew directions as well as document images having text lines with converse skew angles along the same text line. The appearance of accents was common in Greek and French handwritten documents. All documents were written from different writers and in more than half of the documents the distance of adjacent text lines was very small leading to a highly dense text.

For all document images, we have manually created the corresponding text line detection ground truth. The total number of text lines appearing on those images was 3382. Parameters n_1 and n_2 in our methodology (Section 3.2) were experimentally defined to 5 and 9, respectively. Also, the evaluator's acceptance threshold *th* (Section 4.1) was defined to 0.9. Four examples of the proposed text line detection method are demonstrated in Fig. 19.

To check the effectiveness of our method, we based on the performance evaluation methodology described in Section 4.1 using $w_1 = w_4 = 1$ and $w_2 = w_3 = w_5 = w_6 = 0.25$. For the sake of comparison we also implemented a fuzzy run-length approach (such as

Table 2
Computational costs

Methodologies	Time (ms)
Fuzzy run-length [7]	46 928
Projection profiles [1]	741
Hough approach [5]	1203
Proposed method	1903

in Ref. [7]), a projection profile approach (such as in Ref. [1]) and a Hough transform based approach (such as in Ref. [5]).

For the method described in Ref. [7] we computed the fuzzy run-length. The resulting image was then binarized. After a merging phase, an id was attributed to each area that describes the core region of a text line. Finally, we gave to every pixel of the initial image the id of the text line previously detected.

For the projection profile approach [1] we calculated the horizontal projection profile by summing the foreground pixels in every scan line. After a smoothing procedure, we calculated the local minimums of that function that define the boundaries of a region which contains a text line.

In the case of the Hough transform approach [5], a voting scheme is used which takes into account the gravity center of the complete connected component.

In Figs. 20–22, text line detection results are shown in the case of applying Hough approach, fuzzy run-length, and projection profiles on the document images of Figs. 19(a) and (b).

Table 1 shows our comparative experimental results in terms of detection rate (Eq. (8)), recognition accuracy (Eq. (9)), and TLDM (Eq. (10)). As it can be observed from Table 1, the proposed methodology outperforms the other three approaches achieving a detection rate of 95.8% and a recognition accuracy of 93.8%.

As a general conclusion drawn from our experimental results, the fuzzy run-length method as well as the projection profile method fail to detect the text lines because they cannot deal with text lines having either arbitrary skew angles or consecutive lines very close to each other (dense text).

The Hough approach [5] has better performance than the other two methods in terms of detection rate (Eq. (8)) because the existence of skew in the document images does not affect text line detection. However, the fact that voting in the Hough domain considers all connected components as well as that only one point (the gravity center) per connected component votes in the Hough domain, are the main reasons for erroneous text line detection. Also, the Hough approach [5] does not separate vertically connected characters which leads to further false text line detections (Fig. 23).

Most of the errors made by our approach are due to misclassification of accents as well as to incorrect splitting of difficult cases of vertically connected characters (Fig. 24).

Table 2 shows the computational cost of all methodologies that were involved in our experimental work. The environment we used was Borland C++ Builder 6 on an Intel Core Duo at 1.8 Ghz with 512 Mbytes of Ram on Windows Xp. The typical size of the binary image was 2036×2280.

5. Concluding remarks

In this paper we present a new text line detection method for handwritten documents. The proposed methodology meets the following challenges: (i) each text line that appears in the document may have an arbitrary skew angle and converse skew angle along the text line; (ii) text lines may have different skew directions; (iii) accents may be cited either above or below the text line, and (iv) parts of neighboring text lines may be connected.

The main novelties of the proposed approach comprise (i) an efficient block-based Hough transform in which voting occurs on the basis of equally spaced blocks after splitting of the connected components' bounding box; (ii) a partitioning of the connected component domain into three spatial sub-domains which impose a different strategy to the processing of the corresponding connected components, and (iii) the efficient separation of vertically connected parts of text lines.

From our experimental results it is shown that the proposed methodology outperforms existing state-of-the-art methods for text line detection in handwritten documents.

Acknowledgment

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

References

[1] R. Manmatha, J.L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1212–1225.
[2] E. Bruzzone, M.C. Coffetti, An algorithm for extracting cursive text lines, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition, Bangalore, India, 1999, p. 749.
[3] M. Arivazhagan, H. Srinivasan, S.N. Srihari, A statistical approach to handwritten line segmentation, in: Document Recognition and Retrieval XIV, Proceedings of SPIE, San Jose, CA, February 2007, pp. 6500T-1-11.
[4] L.A. Fletcher, R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images, IEEE Trans. Pattern Anal. Mach. Intell. 10 (6) (1988) 910–918.
[5] L. Likforman-Sulem, A. Hanimyan, C. Faure, A Hough based algorithm for extracting text lines in handwritten documents, in: Proceedings of the Third International Conference on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 774–777.
[6] Y. Pu, Z. Shi, A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents, in: Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition, Taejon, Korea, 1998, pp. 637–646.
[7] Z. Shi, V. Govindaraju, Line separation for complex document images using fuzzy runlength, in: First International Workshop on Document Image Analysis for Libraries, 2004, p. 306.
[8] B. Gatos, A. Antonacopoulos, N. Stamatopoulos, ICDAR2007 handwriting segmentation contest, in: 9th International Conference on Document Analysis and Recognition (ICDAR'07), Curitiba, Brazil, September 2007.
[9] S. Nicolas, T. Paquet, L. Heutte, Text line segmentation in handwritten document using a production system, in: Proceedings of the 9th IWFHR, Tokyo, Japan, 2004, pp. 245–250.
[10] Z. Shi, S. Setlur, V. Govindaraju, Text extraction from gray scale historical document images using adaptive local connectivity map, in: Eighth International Conference on Document Analysis and Recognition, Seoul, Korea, 2005, pp. 794–798.
[11] D.J. Kennard, W.A. Barrett, Separating lines of text in free-form handwritten historical documents, in: Second International Conference on Document Image Analysis for Libraries (DIAL'06), pp. 12–23.
[12] Y. Li, Y. Zheng, D. Doermann, Detecting text lines in handwritten documents, in: The 18th International Conference on Pattern Recognition (ICPR'06), pp. 1030–1033.
[13] A. Lemaitre, J. Camillerapp, Text line extraction in handwritten document with Kalman filter applied on low resolution image, in: Second International Conference on Document Image Analysis for Libraries (DIAL'06), pp. 38–45.
[14] C. Weliwitage, A.L. Harvey, A.B. Jennings, Handwritten document offline text line segmentation, in: Proceedings of Digital Imaging Computing: Techniques and Applications, 2005, pp. 184–187.
[15] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, D.K. Basu, Text line extraction from multi-skewed handwritten documents, Pattern Recognition 40 (6) (2007) 1825–1839.
[16] A. Zahour, L. Likforman-Sulem, W. Boussalaa, B. Taconet, Text line segmentation of historical arabic documents, in: 9th International Conference on Document Analysis and Recognition (ICDAR'07), Curitiba, Brazil, September 2007.
[17] B. Gatos, I. Pratikakis, S.J. Perantonis, Adaptive degraded document image binarization, Pattern Recognition 39 (2006) 317–327.
[18] F. Chang, C.-J. Chen, C.-J. Lu, A linear-time component-labeling algorithm using contour tracing technique, Comput. Vision Image Understanding 93 (2) (2004) 206–220.
[19] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis, S. J. Perantonis, A segmentation-free approach for keyword search in historical typewritten documents, in: 8th International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea, August 2005.
[20] P. Hough, Methods and means for recognizing complex patterns, US Patent 3,069,654, 1962.

[21] H.J. Lee, B. Chen, Recognition of handwritten Chinese characters via short line segments, Pattern Recognition 25 (5) (1992) 543–552.
[22] A. Antonacopoulos, B. Gatos, D. Karatzas, ICDAR 2003 page segmentation competition, in: 7th International Conference on Document Analysis and Recognition (ICDAR'03), Edinburgh, Scotland, August 2003, pp. 688–692.
[23] A. Antonacopoulos, B. Gatos, D. Bridson, ICDAR2005 page segmentation competition, in: 8th International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea, August 2005, pp. 75–79.
[24] Phillips, A. Chhabra, Empirical performance evaluation of graphics recognition systems, IEEE Trans. Pattern Anal. Mach. Intell. 21(9) (1999) 849–870.
[25] ⟨http://pergamos.lib.uoa.gr/dl/object/col:histarch⟩
[26] ⟨http://jfk.ci.dallas.tx.us⟩

About the Author—georgios e. louloudis was born in 1979 in Athens, Greece. He received his Computer Science diploma in 2000 and his Master degree in 2002, both from the Department of Informatics and Telecommunications of University of Athens, Athens, Greece. He is currently a Ph.D. student in the Department of Informatics and Telecommunications, University of Athens, Greece, in the field of Handwritten Character Recognition. His main research interests are in Image Processing and Document Image Analysis, OCR, and Pattern Recognition.

About the Author—basilis g. gatos was born in 1967 in Athens, Greece. He received his Electrical Engineering Diploma in 1992 and his Ph.D. degree in 1998, both from the Electrical and Computer Engineering Department of Democritus, University of Thrace, Xanthi, Greece. His Ph.D. thesis is on Optical Character Recognition Techniques. In 1993 he was awarded a scholarship from the Institute of Informatics and Telecommunications, NCSR "Demokritos", where he worked till 1996. From 1997 to 1998 he worked as a Software Engineer at Computer Logic S.A. From 1998 to 2001 he worked at Lambrakis Press Archives as a Director of the Research Division in the field of digital preservation of old newspapers. From 2001 to 2003 he worked at BSI S.A. as Managing Director of R&D Division in the field of document management and recognition. He is currently working as a Researcher at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos", Athens, Greece. His main research interests are in Image Processing and Document Image Analysis, OCR, and Pattern Recognition. He has more than 60 publications in journals and international conference proceedings and has participated in several research programs funded by the European community. He is a member of the Technical Chamber of Greece.

About the Author—ioannis pratikakis received the Diploma degree in Electrical Engineering from the Demokritus University of Thrace, Xanthi, Greece, in 1992, and the Ph.D. degree in Applied Sciences from Vrije Universiteit Brussel, Brussels, Belgium, in 1998. From March 1999 to March 2000, he was at IRISA/ViSTA group, Rennes, France, as an INRIA postdoctoral fellow. He is currently working as a Research Scientist at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos", Athens, Greece. His research interests include 2D and 3D image analysis, image and volume sequence analysis and content-based image/3D models search and retrieval. He has published more than 60 papers in journals, book chapters, and papers in conference proceedings in the above areas. He has participated in more than 15 national and international R&D projects.

About the Author—constantin halatsis was born in Athens, Greece, in 1941. He received his B.Sc. degree in Physics in 1964 and his M.Sc. degree in Electronics in 1966, both from the University of Athens. In 1971 he received his Ph.D. degree in Computer Science from the University of Manchester, UK.
From 1971 to 1981 he was a researcher with the Computer Centre of NRC Demokritos in Athens. In 1981 he became Full Professor of Computer Science at the University of Thessaloniki, Greece, and in 1988 he was invited as Full Professor of CS at the Department of Informatics and Telecommunications of the University of Athens, where he is now since then. He has a wide area of research interests including logic design, computer architectures, fault-tolerant computing, artificial intelligence, and theory of computation. He is the author/co-author of more than 100 technical papers published in refereed international scientific journals and conference proceedings.