Image and Vision Computing 28 (2010) 590–604

Contents lists available at ScienceDirect

Image and Vision Computing

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

Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths

Nikos Nikolaou [a,b,*], Michael Makridis [a], Basilis Gatos [b], Nikolaos Stamatopoulos [b], Nikos Papamarkos [a]

[a] Department of Electrical and Computer Engineering, Democritus University of Thrace, 67 100 Xanthi, 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 7 May 2008
Received in revised form 21 May 2009
Accepted 22 September 2009

Keywords:
Text line segmentation
Word segmentation
Character segmentation
Historical machine-printed documents
Run Length Smoothing Algorithm

ABSTRACT

In this paper, we strive towards the development of efficient techniques in order to segment document pages resulting from the digitization of historical machine-printed sources. This kind of documents often suffer from low quality and local skew, several degradations due to the old printing matrix quality or ink diffusion, and exhibit complex and dense layout. To face these problems, we introduce the following innovative aspects: (i) use of a novel Adaptive Run Length Smoothing Algorithm (ARLSA) in order to face the problem of complex and dense document layout, (ii) detection of noisy areas and punctuation marks that are usual in historical machine-printed documents, (iii) detection of possible obstacles formed from background areas in order to separate neighboring text columns or text lines, and (iv) use of skeleton segmentation paths in order to isolate possible connected characters. Comparative experiments using several historical machine-printed documents prove the efficiency of the proposed technique.

1. Introduction

The recognition of historical machine-printed documents, such as old books, is essential for quick and efficient content exploitation of the valuable historical collections. In order to achieve accurate recognition results, a robust and efficient segmentation task must be involved. In this paper, we strive towards the development of efficient techniques in order to segment document pages resulting from the digitization of historical machine-printed sources.

This kind of documents often suffer from low quality and several degradations due to the old printing matrix quality or ink diffusion, and exhibit complex and dense layout (Fig. 1). Additionally, historical machine-printed documents have some notable characteristics which make the segmentation problem difficult and very challenging. We have defined a categorization for the machine-printed historical documents which is based on these characteristics. According to this categorization, historical machine-printed documents can be:

1. Multi column documents.
2. Noisy documents.
3. Documents with non-constant spaces between text lines, words and characters.
4. Documents with marginal text.
5. Documents in which various font sizes coexist.
6. Documents with ornamental characters and graphical illustrations.
7. Documents whose text is warped and/or skewed.

The above categorization is used in the experimental and evaluation section (Section 4) in order to measure the performance of our method under certain conditions. A similar list of characteristics is used in the work of Ramel et al. [1] where page layout analysis is performed on historical printed books.

During segmentation, the aim is to process text blocks in order to detect text lines, words and, finally, characters that will feed an OCR classifier. In the literature, the problem of segmenting historical machine-printed documents is tackled by the use of techniques such as the projection profiles [2] or the Run Length Smoothing algorithm (RLSA) [3] which are mainly designed for contemporary documents. As a result, the above mentioned problems inherent in historical machine-printed documents seriously affect the segmentation and, as a result, the recognition accuracy of the OCR system. To this end, in this paper we propose a new segmentation technique that is focused on the historical machine-printed document characteristics and is based on the following innovative aspects: (i) use of a novel Adaptive Run Length Smoothing Algorithm (ARLSA) in order to face the problem of complex and dense document layout, (ii) detection of noisy areas and

* Corresponding author. Address: Department of Electrical and Computer Engineering, Democritus University of Thrace, 67 100 Xanthi, Greece. Tel.: +30 2541076172.
E-mail addresses: nnikol@ee.duth.gr, nkleopas@gmail.com (N. Nikolaou), mmakridi@ee.duth.gr (M. Makridis), bgat@iit.demokritos.gr (B. Gatos), nstam@iit.demokritos.gr (N. Stamatopoulos), papamark@ee.duth.gr (N. Papamarkos).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.09.013

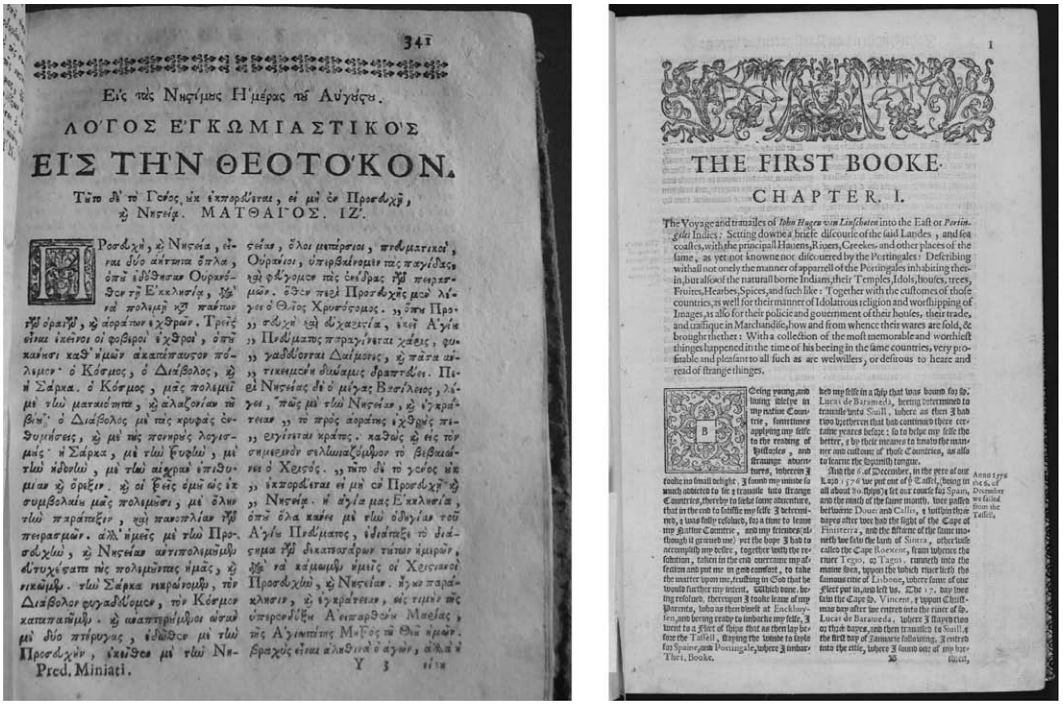


Fig. 1. Examples of historical machine-printed documents.

punctuation marks that are usual in historical machine-printed documents, (iii) detection of possible obstacles formed from background areas in order to separate neighboring text columns or text lines, and (iv) use of skeleton segmentation paths in order to isolate possible connected characters.

The rest of this paper is organized as follows: Section 2 discusses related work in this area. A detailed description of the proposed methodology is presented in Section 3, involving a description of Adaptive Run Length Smoothing in Section 3.1. Evaluation and experimental results are presented in Section 4 and, finally, Section 5 describes the conclusions.

2. Related work

Various document image segmentation techniques have been proposed in the literature. These techniques can be categorized based on the document image segmentation algorithm that they adopt. The most known of these segmentation algorithms are the following: *X–Y* cuts or projection profiles based [4], Run Length Smoothing Algorithm (RLSA) [5], component grouping [6], document spectrum [7], whitespace analysis [8], constrained text lines [9], Hough transform [10,11], Voronoi tessellation [12] and Scale space analysis [13]. All of the above segmentation algorithms are mainly designed for contemporary documents. For the case of historical and handwritten document segmentation, projection profiles [2,14], Run Length Smoothing Algorithm [3,15], Hough transform [14,16] and scale space analysis algorithms [13] are mainly used. Table 1 categorizes all of the aforementioned segmentation algorithms and depicts the way they have been used in document processing.

It should be noted that historical machine-printed and handwritten documents have some basic similarities regarding the difficulties of the segmentation problem. Local skew, overlapping text lines and connected characters are some examples of common segmentation problems in these types of documents. Also, degradations due to the old printing matrix quality or ink diffusion may produce a machine-printed document layout similar to the handwritten document layout. For this reason, segmentation techniques for handwritten documents are included in this section.

Since document segmentation is usually applied in three levels (text line segmentation, word segmentation and character segmentation), in the remaining of this section we have recorded the state-of-the-art works for each of these levels.

Text line segmentation is generally applied as a preprocessing step. Text lines can be used as input in word and character segmentation applications. In Ref. [2], a method which extracts

Table 1
Categorization of segmentation algorithms.

Segmentation algorithm	Printed documents	Handwritten documents	Historical documents	Page segmentation	Text line segmentation	Word segmentation	Character segmentation
X–Y cuts	•	•	•	•	•	•	•
RLSA	•	•	•	•	•	•	
Docstrum	•			•	•		
Whitespace analysis	•			•	•		
Constrained text lines	•			•	•		
Hough transform	•				•	•	
Voronoi	•			•			
Scale space analysis	•	•	•	•	•	•	

semantic-based content from historical machine-printed documents is proposed. Text line segmentation is based on the analysis of local minima and maxima patterns derived from the vertical projection profile of the gray scale image. The analysis takes into account the expected size of characters and each identified text line is verified by examining previously identified lines. The performance of this approach can be seriously affected by the existence of non-constant spaces between text lines as well as by the existence of warped or skewed text lines. In the work of Lemaitre [17], text line segmentation is applied on historical handwritten documents. Assuming that at a certain distance from the document text lines appear as line segments, the method is applied on low resolution gray scale images. Text line segments are extracted based on a technique that uses kalman filters. Li et al. [18] propose a text line detection technique for handwritten documents. The initial binary image is converted into a gray scale using a Gaussian filter in order to enhance the text line structures. By adopting the level set method, text line boundaries are initially estimated and through a segment merging procedure the final result is extracted. Kennard et al. [19] propose a technique which concerns also the text line segmentation in historical handwritten documents. In this technique, the foreground–background transitions count is used to determine possible text line areas. A min-cut/max-flow graph cut algorithm is then used to spit joint text lines and finally near components are merged. Although the above-mentioned techniques [17–19] 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 in English documents it is a common constituent in documents of French or Greek language. Text line extraction from cursive text is studied in [20]. It is based on the analysis of the horizontal run projections. The method groups and splits connected components in order to isolate text lines where ascending and descending characters overlap. This method cannot confront with the problem of variable skew angles between different text lines and along the same text line. An extended analysis on the issue of text line segmentation of historical documents can be found in [21].

Previous work in *word segmentation* includes several approaches for machine-printed and handwritten documents. Word segmentation is very important for segmentation-free approaches that entail the recognition of the whole word as in [22–24] where text line and word segmentation is used for creating an index based on word matching. For the case of historical machine-printed documents, Antonacopoulos and Karatzas [2] calculate and analyze the horizontal projection profile of each extracted text line segment in order to identify suitable spaces between words. Non-constant spaces between words can seriously affect the performance of this approach. In the work of Gatos et al. [3], word segmentation in historical machine-printed documents is based on a run length smoothing in the horizontal and vertical directions. For each direction, white runs having length less than a threshold are eliminated in order to form word segments. The main problem of this approach is the merging of neighboring words when the text is very dense. Manmatha et al. approach [13] is based on a scale space technique for word segmentation on handwritten documents of George Washington's collection. Word extraction for machine-printed documents is examined in the work of Park et al. [25]. A 3D neighborhood graph model analyses the structural information of documents and with the use of angle and distance constraints word components are identified. In the two previous techniques [13,25], accurate word extraction is based on text line segmentation but this is their main disadvantage in documents with warped or skewed text. Park and Govindaraju [26] present a methodology that takes advantage of the spacing between words in a phrase to aid the handwritten recognition process. The determination of word breaks is made in a manner that adapts to the writing style of the individual.

Character segmentation previous work concerns mostly handwritten text but methods for machine-printed text have also been proposed. Antonacopoulos and Karatzas [2] use the horizontal projection profile of each word segment for character segmentation in historical machine-printed documents. The first split point position is predicted based on the expected character box width and refined according to the location and strength of the projection minima. The next splitting positions are derived in the same manner using the information from the location of the previous separator. This approach cannot handle the case of overlapping characters, that is characters that their bounding boxes overlap horizontally. In the work of Nomura et al. [27], character segmentation and feature extraction is applied on degraded images of license plates. Horizontal projection profile is first used to merge character fragments and final character segmentation is guided by morphological operations. In Liang et al. [28] work, character segmentation is applied on modern machine-printed documents. Here, the touching characters problem is confronted using discrimination functions based on pixel and component profile projections. Also, contextual information and spell checking are used to improve recognition accuracy. Kavallieratou et al. [29] proposed a technique for handwritten character segmentation. A presegmentation stage locates all possible splitting points by detecting local minima in the horizontal projection profile. Final splitting points are determined by a transformation-based learning procedure. A cursive script character segmentation method is proposed in [30] in which knowledge of the structure of English letters and the structural characteristics between background and characters is investigated. Yanikoglu and Sandon [31] proposed a character segmentation technique for cursive handwritten text. Using linear programming, the weights of the extracted features are determined and a segmentation cost function for all possible character splitting points is formed. The final splitting points result after the evaluation of each cost function. The concept of water reservoir is introduced in the work of Pal et al. [32] where touching numeral segmentation is performed. A reservoir is the region that is formed around the touching point of two numerals and the analysis of its boundary determines the cutting point.

3. Proposed methodology

In this paper, we combine ideas from connected component processing, whitespace analysis and skeleton representation and introduce several innovative aspects in order to achieve a successful segmentation of historical machine-printed documents. The segmentation is accomplished at all levels including text line, word and character segmentation. The main innovative aspects introduced are the following:

- The use of a novel Adaptive Run Length Smoothing Algorithm (ARLSA) which is a modified version of the state-of-the-art RLSA [5] and efficiently groups homogeneous document regions.
- The definition of background obstacles that ALRSA is not allowed crossing in order to avoid merging neighboring text columns or text lines.
- Special handling of punctuation marks in order to help efficient text line detection.
- The use of skeleton paths in order to further isolate connected characters.

The proposed methodology is divided into five consecutive stages and explained in detail in the following sections: The whole method can be summarized at the following distinct steps:

1. Noise and punctuation marks removal.
2. Obstacles detection.
3. Text line segmentation.
4. Word segmentation.
5. Character segmentation.

An adaptive binarization technique [33] is first applied in order to produce a b/w image. This technique does not require any parameter tuning by the user and can deal with degradations which occur due to shadows, non-uniform illumination, low contrast, large signal-dependent noise, smear and strain. The average character height AH for the document image is then calculated [3] in order to be used for parameter tuning and font size independency in all processes. The b/w image is first deskewed and preprocessed by removing noisy black borders and noisy text areas appearing from neighboring pages [34] as well as small noisy components which can have a negative effect on the statistical measures of the document image.

Punctuation marks are initially eliminated and combined with a later stage result of the method in order for the text line segmentation stage to be completed more accurately. The resulted image of this stage is smeared with a proposed Adaptive Run Length Smoothing Algorithm (ARLSA) that helps grouping together homogeneous text regions. At the next stage, obstacles are detected and used in order to isolate different text lines and different text columns. Obstacles are considered as regions within a document that a horizontal run length procedure is not allowed to cross. Using obstacles, the initial text line segmentation result is efficiently calculated. By combining this result with the punctuation marks, the final text line segmentation is performed. In the next stage, each detected text line is processed independently to extract the word segments. Based on the histogram of horizontal distances between adjacent bounding boxes, a proper threshold value is calculated and used in order to merge components belonging to the same word. Finally, character segments are extracted from each word segment based on skeleton segmentation paths which are used to isolate possible connected characters. The whole proposed methodology is given in the flowchart of Fig. 2. In Section 3.1 the ARLSA is discussed before the detailed description of the method.

3.1. Adaptive Run Length Smoothing

The RLSA [5] is one of the most common algorithms used in page layout analysis and segmentation techniques. It operates on binary images in a specified direction (usually horizontal or vertical) by replacing a sequence of background pixels with foreground pixels if the number of background pixels in the sequence is smaller or equal than a predefined threshold $T_{\max}$. Its purpose is to create homogenous regions in the document image.

In the proposed segmentation technique, a modified version of the horizontal RLSA is proposed, the Adaptive Run Length Smoothing Algorithm (ARLSA), in order to overcome the drawbacks of the original algorithm, such as grouping inhomogeneous components or different slanted lines. ARLSA also works successfully with documents containing characters with variable font size. Before the application of ARLSA a connected component analysis is necessary. Two types of background pixels (white) sequences are considered. The first type concerns sequences which occur between two foreground pixels (black) which belong to the same connected component as shown in Fig. 3a. In this case, all background pixels of the sequence are replaced with foreground pixels. The second type of background pixels sequence occurs between two different connected components (Fig. 3b). In this case, constraints are set in regard to the geometrical properties of the connected components and the replacement with foreground pixels is performed when these constraints are satisfied.

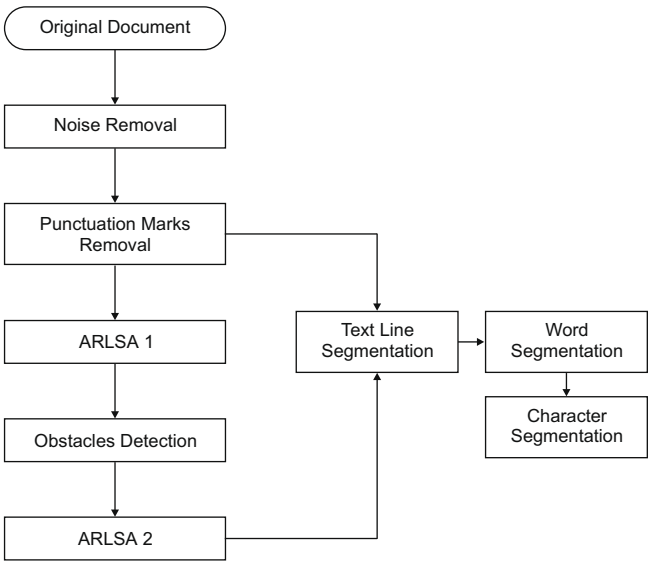


Fig. 2. Flowchart of the proposed method.

Fig. 3. The two types of background sequences: (a) belonging to the same connected component and (b) belonging to different connected components.

Let CC_i and CC_j be two connected components and $S(i,\ j)$ a horizontal sequence of background pixels between CC_i and CC_j.

We define the following four metrics:

- $L(S)$: length of the sequence $S(i,\ j)$, that is the number of white pixels.
- $H_R(S)$: height ratio between CC_i and CC_j, which is defined as follows:

$$H_R(S) = \frac{\max\{h_i,\ h_j\}}{\min\{h_i,\ h_j\}} \tag{1}$$

where $h_i,\ h_j$, the heights of CC_i and CC_j, respectively.

- $O_H(S)$: the horizontal overlapping between the bounding boxes of CC_i and CC_j, which is defined by the following equation:

$$O_H(S) = \max\{Yl_i,\ Yl_j\} - \min\{Yr_i,\ Yr_j\} \tag{2}$$

where $\{Xl_i,\ Yl_i\}$ and $\{Xr_i,\ Yr_i\}$ the coordinates of the upper left and down right corner of the CC_i's bounding box. The horizontal overlapping $O_H(S)$ is graphically demonstrated in Fig. 4 and it can be observed that when $O_H(S) < 0$, horizontal overlapping exists between the two connected components CC_i and CC_j.

- $N(S)$: a binary output function.
  $N(S)$ is set to 0 when in the 3×3 neighborhood of at least one pixel of the sequence $S(i,\ j)$, a third connected component $CC_k,\ k \neq i,\ j$ exists. Otherwise, $N(S)$ is set to 1. As shown in the example of Fig. 5, in the 3×3 neighborhood of the gray pixel sequences, no object other than CC_i and CC_j exists. In this case $N(S)$ is set to 1. In the white sequence pixels, this is not true, so $N(S)$ is set to 0.

Based on the above metrics, a sequence of background pixels is replaced with foreground pixels only if:

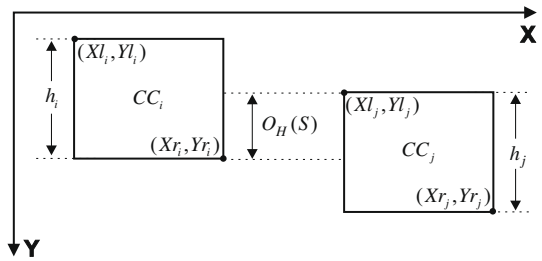


Fig. 4. Graphical depiction of horizontal overlapping.

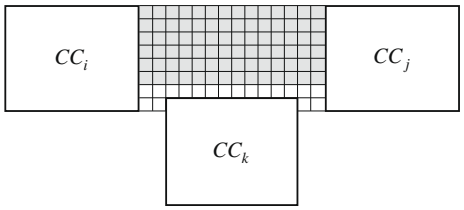


Fig. 5. Graphical depiction of the constraint based on the $N(S)$ function. Gray pixels represent the background pixels which will be replaced with foreground pixels.

$$(L(S) \leqslant T_l) \wedge (H_R(S) \leqslant T_h) \wedge (O_H(S) \geqslant T_o) \wedge (N(S) = 1) \tag{3}$$

where $T_l,\ T_h,\ T_o$ are predefined threshold values.

The length threshold T_l of each sequence is related to the heights of the connected components. If h_i and h_j express the heights of CC_i and CC_j then

$$T_l = a \cdot \min\{h_i,\ h_j\} \tag{4}$$

where a is a constant value.

Threshold T_h was set to 3.5 based on the following assumption: We consider that between a lowercase character such as "o" and a character with descender or ascender of the same font size, a height ratio between 2 and 3 is very common. Taking into account the fact that in historical documents the font size varies we concluded to value 3.5 for T_h which gives enough tolerance against font size variation between characters of the same text line.

The horizontal overlapping threshold T_o is expressed as the percentage of overlap in regard to the component with the smallest height, that is:

$$T_o = c \cdot \min\{h_i,\ h_j\} \tag{5}$$

where c is set to 0.4. This means that at least 40% of the shortest component height must be covered in order for a link to be established.

The constraint based on the function $N(S)$, ensures that background pixels will be transformed into foreground pixels only if in their 3×3 neighborhood no pixels of a third connected component exists as shown in Fig. 5. Its purpose is to prevent the creation of false links between objects and therefore the integration into component groups of unwanted objects. It is very helpful in historical and degraded documents where text line spacing is narrow and characters from different text lines overlap. In the example of Fig. 5, the application of the ARLSA, without taking into account this constraint, would result in the creation of a group containing all three components $CC_i,\ CC_j,\ CC_k$ which is obviously wrong.

The ARLSA in regard to the original algorithm can prevent the creation of inhomogeneous groups of components, namely to have large and small characters grouped together. Also, it has tolerance against warped or skewed text, that is components of different text lines that are close to each other at the horizontal direction is not likely to be linked even when T_l receives large values. This happens due to the horizontal overlapping constraint. An example which shows a comparison between the original RLSA and the proposed ARLSA for text-line detection is shown in Fig. 6.

The original document of Fig. 6a is processed with the RLSA where $T_{\max}$ is set to a small value equal to AH (average character height). The average character height (AH) of the document is calculated according to the technique proposed in [3] and it is also used during the character segmentation procedure. Small size text is merged into word or sets of words but large size text is maintained into individual characters (Fig. 6b). If a larger value of $T_{\max}$ is used, for example $7 \cdot \mathrm{AH}$, the resulted image is formed as Fig. 6c shows. Large text is correctly merged into text lines but the graphic element is linked with the small size text. In the case of Fig. 6d, in which the ARLSA is applied, all text elements were correctly formed into text lines without links with the graphic element. The value of factor a (Eq. 4) used here is 5. This is a large

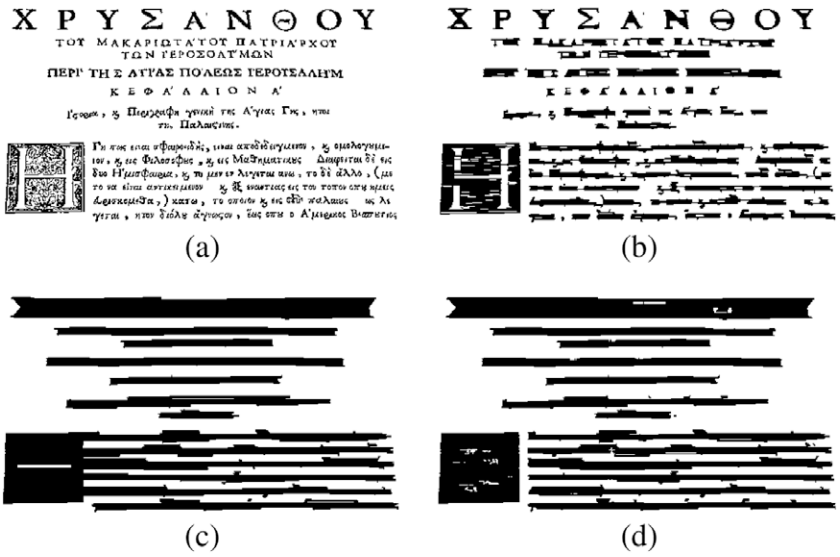


Fig. 6. RLSA – ARLSA comparison: (a) original image, (b) RLSA with small threshold ($T_{\max} = \mathrm{AH}$), (c) RLSA with large threshold ($T_{\max} = 7 \cdot \mathrm{AH}$) and (d) ARLSA ($a = 5,\ T_h = 3.5,\ c = 0.4$).

value and it ensures that even distant objects will be joined if the other three conditions of the ARLSA procedure are satisfied.

3.2. Noise and punctuation marks removal

The purpose of this stage is to remove small noisy connected components and to erase punctuation marks in order to improve the structure of the background and simplify the following procedures of the technique.

First, noisy elements are filtered out based on three characteristics of the connected components and their corresponding bounding boxes. For a connected component CC_i these characteristics are:

- The height of the bounding box of the CC_i, $H(CC_i)$.
- The elongation $E(CC_i) = \frac{\min\{H(CC_i),\ W(CC_i)\}}{\max\{H(CC_i),\ W(CC_i)\}}$. This measure shows the ratio of the shorter to the longer side of each bounding box.
- The density $D(CC_i) = \frac{P_{num}(CC_i)}{BB_{size}(CC_i)}$,

which is the ratio of the number of foreground pixels $P_{num}(CC_i)$ to the total number of pixels in the bounding box $BB_{size}(CC_i) = H(CC_i) \cdot W(CC_i)$.

Connected components with $H(CC_i) < \mathrm{AH}/3$, or $D(CC_i) < 0.08$, or $E(CC_i) < 0.08$ are considered as noisy elements and they are eliminated. These values have been selected very carefully so no character elements will be eliminated. With this type of filtering only very noisy non-character objects are removed.

The second type of filtering removes punctuation marks. It is based on the comparison of the connected components from two images, the initial document image and the resulted image after the application of the ARLSA with $a = 1.5$ (see Eq. 4). Let I_1 be the original image (see Fig. 7a), I_2 the image after the application of the ARLSA (see Fig. 7b). The number of pixels P_{I_2} of each connected component $CC_i \in I_2$ is calculated, that is the number of the black pixels. In the defined area of each $CC_i \in I_2$, the sum P_{I_1} of the corresponding black pixels of I_1 is also calculated. The ratio of these two sums is taken into account as in the following equation:

$$P_R = \frac{P_{I_2}}{P_{I_1}} \leqslant T_R \tag{6}$$

Components which correspond to pixel size ratio smaller than T_R are removed and a new image I_3 (see Fig. 7c) is produced. Punctuation marks are removed because they are mainly isolated objects. This means that after the application of the ARLSA their size is likely to remain constant or change by a small factor contrary to text components. The final result is obtained by an AND operation between images I_1 and I_3 as shown in Fig. 7d. A proper value for T_R was found to be 1.15. This means that P_{I_2} is expected to be 15% larger than P_{I_1}.

The punctuation marks eliminated in this stage are used in the text line segmentation stage in order to extract the final form of text lines.

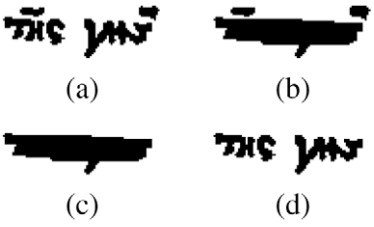


Fig. 7. Punctuation marks removal: (a) original image I_1, (b) application of ARLSA (I_2), (c) components with small pixel size ratio are removed (I_3) and (d) resulted image after the operation I_1 AND I_3.

3.3. Obstacles detection

The purpose of obstacles is to isolate different text lines and different text columns by defining regions within a document that a horizontal run length procedure is not allowed to cross. Obstacles are used in [9] where text line extraction is performed in documents of multiple text columns. In this paper, two types of obstacles are extracted, column and text line obstacles. Column obstacles are extracted to define different text column spaces or border text, while text line obstacles to locate regions between text lines which belong in the same text column.

So, after the application of noise removal, punctuation marks removal and the ARLSA with $a = 1.5$ (see Eq. 4), the algorithm detects on the resulting image two types of obstacles, column and text line obstacles. The application of ARLSA before the obstacle detection stage is very important because it prevents column obstacles from appearing between words or characters of the same line.

First, the histogram $h_v(w)$ of vertical white run lengths is constructed. If a document has text columns or borders large vertical white run lengths will appear between text columns or in the region of a border. White run lengths whose values are greater than

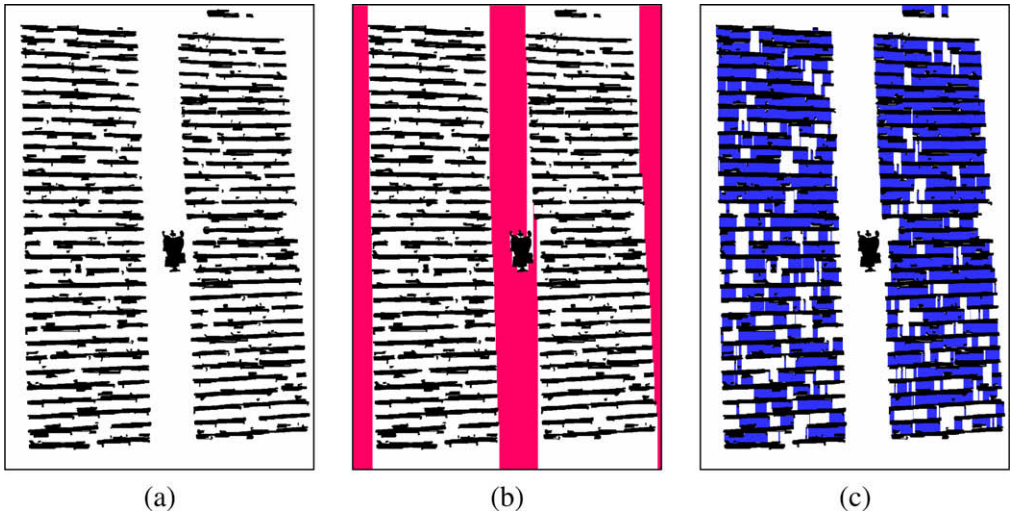


Fig. 8. Example of obstacle detection and text line estimation: (a) initial image, (b) column obstacles and (c) text line obstacles.

$k \cdot H$, where H is the document's height, are considered as column obstacles. Parameter k is a constant and its value is set to 1/3. With this value, the creation of obstacles between objects which belong to the same text line is avoided and yet, the space between different text columns is detected. An example of column obstacle detection is given in Fig. 8b.

For the case of text line obstacles, white run lengths whose values are smaller than $M_v = \arg\max h_v(w)$ represent the text line obstacles. M_v indicates the distances between components belonging to different text lines. Fig. 8c shows an example of text line obstacles.

Also, in Fig. 9, a more detailed text line obstacles detection example is presented. Blue pixels represent the obstacles while green pixels indicate the possible horizontal links between different connected components. All horizontal line segments that connect objects which belong to different text lines are blocked by the obstacles. In contrary, horizontal line segments between objects of the same text line are allowed to be linked.

3.4. Text line segmentation

Text line segmentation is performed by applying the ARSLA algorithm to the original image, after the noise and punctuation marks removal. Furthermore, ARLSA is constrained by text line and column obstacles. Therefore, ARLSA is performed only in cases where a background (white) pixel sequence does not include pixels that they have been also detected as obstacles. Constant a (see Eq. 4) is set to 5, which is a relative large value, so as distant parts of the same text line can be linked if the other three conditions of the ARLSA are satisfied. Obstacles prevent parts of different text lines to be linked despite of the large value of a.

Finally, the punctuation marks that were removed, as described in Section 3.2, are now combined with their nearest text line to complete the text line segmentation process. An example of text line segmentation is presented in Fig. 10.

3.5. Word segmentation

The word segmentation procedure of the proposed technique is applied independently to each text line detected from the previous stage of the algorithm. All connected components of a text line L_i are first sorted according to their x coordinate and the histogram H_d of the horizontal distances between adjacent bounding boxes is constructed. A negative value for a distance of vertically overlapped bounding boxes is considered to be zero.

In order to achieve a proper word segmentation of the components of the text line, adjacent connected components with distance smaller D_T are considered to belong to the same word. Threshold value D_T is defined by the following equation:

$$D_T = l + Max_v \tag{7}$$

where Max_V represents the peak of the histogram H_d and l a constant tolerance value. When l takes the value 2, the best word segmentation performance was measured. A word segmentation example is given in Fig. 11.

3.6. Character segmentation

Once the words within the text lines have been located, we use the following method in order to separate them into letters. Our algorithm is based on the segmentation algorithm described for touching numerals in [35]. The basic idea is that we can find possible segmentation paths linking the feature points on the skeleton of the word and its background.

We consider that the width of a letter cannot be less than MinCharWidth = AH/2 and more than MaxCharWidth = 1.5 · AH. Then, we calculate the connected components (CCs) of a word and apply the following steps in all the CCs that have their height to width ratio less or equal to 0.5, in order to separate them into letters (Fig. 12a and b).

Step 1: We calculate the skeleton of the CC and its background (Fig. 12c). For the skeletonization process, we use an iterative method presented in [36].

Step 2: We classify the skeleton in four different segments as follow (Fig. 12d):

- *Top-segment*: The segment generated from the upper part of the background region.
- *Bottom-segment*: The segment generated from the lower part of the background region.
- *Stroke-segment*: The segment generated from the black pixels of the CC.
- *Hole-segment*: The segment generated from the hole-region of the background.

Step 3: We locate the feature points of the skeleton (Fig. 12e). The different kinds of feature points are defined as follows:

- *Fork point*: The point on a segment which has more than two connected branches.
- *End point*: The point on a segment that has only one neighbour pixel.
- *Corner-point*: The point on a segment where the curvature of the segment changes sharply.

Step 4: In this step we construct all the possible segmentation paths (Fig. 12f). We simultaneously apply two different searches, downward search and upward search. In downward search, we construct all the segmentations paths which start from the feature points on the top-segment. Each segmentation path should start from a feature point on the top-segment, pass through one or two feature points on the stroke-segment and end at a feature point on the bottom-segment. The distance between two feature points (top-stroke, stroke-bottom or stroke–stroke) must be less than 0.8 · AH. Therefore, if no one feature point on the stroke-segment matches a feature point on the top-segment, a vertical path is constructed starting from this feature point on the top-segment until it touches the bottom-segment. Also, if no one feature point on the bottom-segment matches a feature point on the

(a) (b)

Fig. 9. Detailed text line obstacles detection: (a) document section with warped and skewed text, (b) text line obstacles (blue pixels) and horizontal line segments (green pixels) between different connected components.

stroke-segment, a vertical path is constructed starting from this feature point on the stroke-segment until it touches the bottom-segment. A similar process is applied to upward search in order to construct possible segmentation paths from bottom-segment to top-segment.

A segmentation path must satisfy the following constraints:

- Its length must be less than AH.
- Its width must be less than $(1/3) \cdot AH$.
- The ratio between the count of foreground pixels and the count of background on the segmentation path must be smaller than 3.
- There must be no stroke-segment between two matched feature points.
- If it is a vertical path, it must be cut the stroke-segment only in one point.

Step 5: After locating all the possible segmentation paths we decide which of them are the best segmentation paths (Fig. 12g). In order to achieve this, starting from the beginning of component or from the last segmentation path that was selected, we take into consideration only segmentation paths that result to characters with width

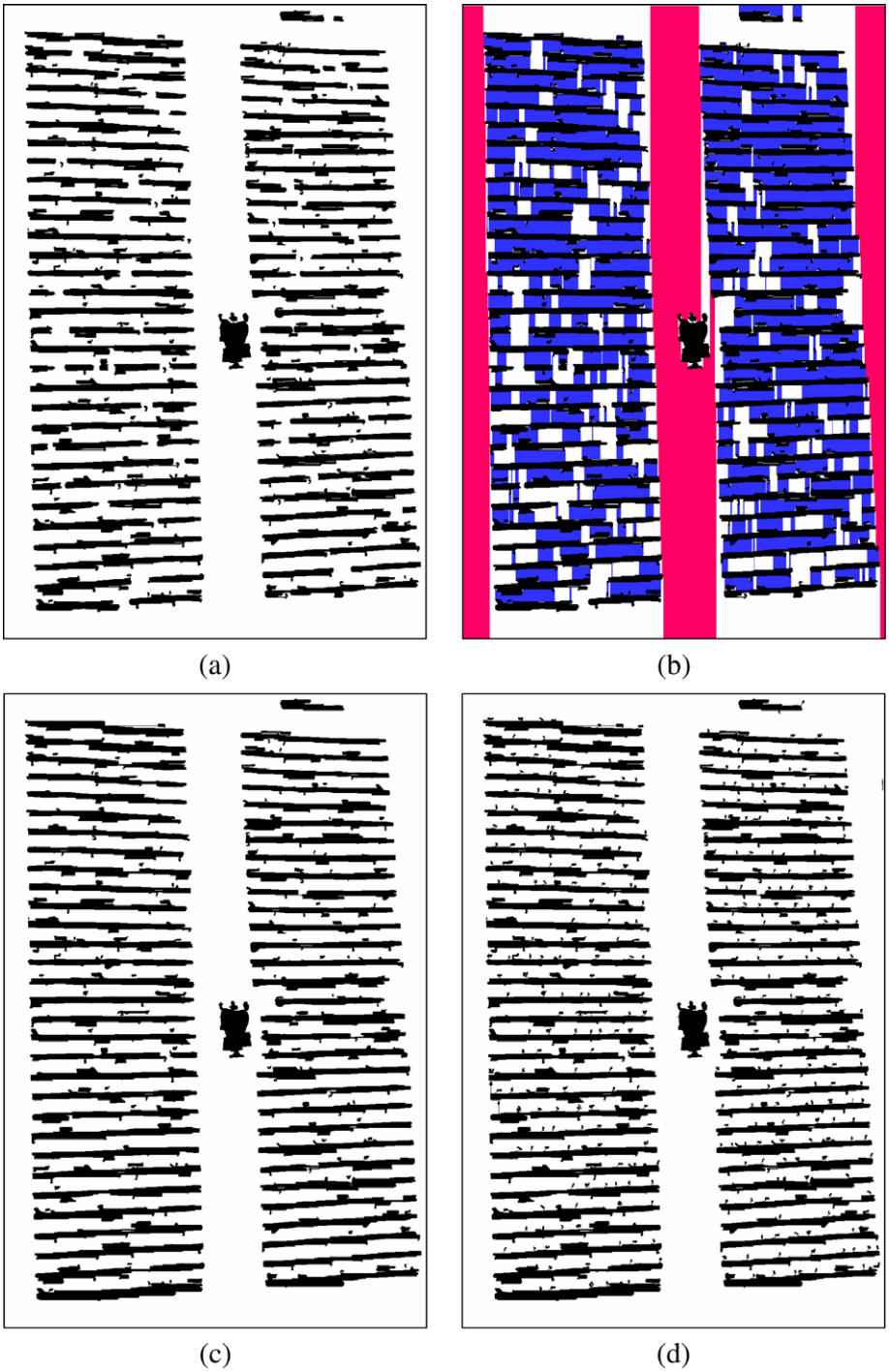


Fig. 10. Example of text line segmentation: (a) initial image, (b) column and text lines obstacles, (c) initial text line segmentation (d) final text line segmentation after combining image (c) with the punctuation marks.

in the limit of [MinCharWidth, MaxCharWidth]. Among these, the best segmentation path is selected as the one that minimizes the following criteria:

- The divergence of resulting letter's width from the expected width (AH).
- The divergence of resulting letter's height from the expected height (AH).
- The length of the segmentation path.
- The width of the segmentation path.

We repeat this process until the CC cannot be segmented into other letters or no other possible segmentation paths exist. Once the characters within the word have been located, in order to merge pieces of a broken character, we calculate all the CCs of the word which have width less than MinCharWidth and then we merge them with the nearest character.

4. Evaluation and experimental results

The proposed algorithm was tested on numerous historical and degraded machine-printed documents. The set of images used for the experiments and the evaluation procedure consists of Greek, English, French and Roman documents. The algorithm performed successfully even in cases with text of different size, or with text and non-text areas lying very near, or with warped text lines. Some problems occurred when the text line segmentation procedure failed. Examples of text line, word and character segmentation extracted with our method is given in Fig. 13a–f. In these examples, each segment is represented by a different color and for viewing reasons, in most of them a portion of the result is depicted.

Some common segmentation problems of the historical machine-printed documents are presented in these examples. In Fig. 13a, the document has marginal text and also the text is warped and skewed. In the example of Fig. 13b the text font size is very inhomogeneous and additionally the document contains graphical illustrations. Fig. 13c shows a word segmentation example of skewed document and Fig. 14d shows an English script word segmentation example. Finally, Fig. 13e and f demonstrate two character segmentation examples where it can be observed that joined characters are correctly segmented and also the punctuation marks are assigned to the correct character.

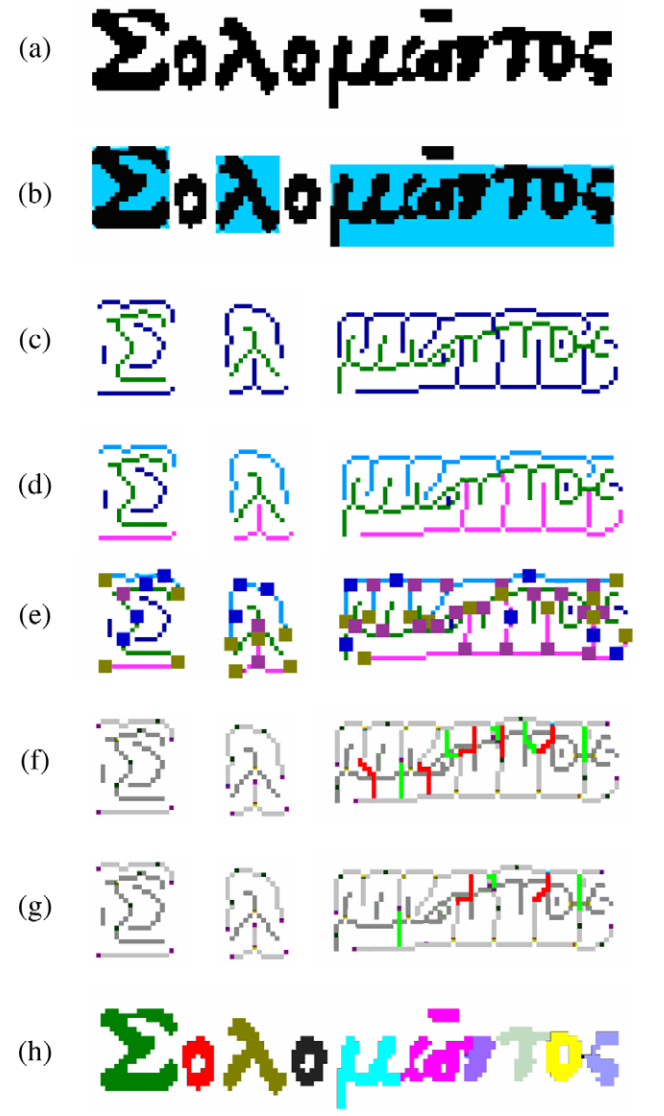


Fig. 12. Example of character segmentation: (a) original image, (b) Candidates CCs to be splitted, (c) skeleton of CCs and their background, (d) classification of skeletons, (e) feature points, (f) possible segmentation paths, (g) best segmentation paths and (h) final result of character segmentation.

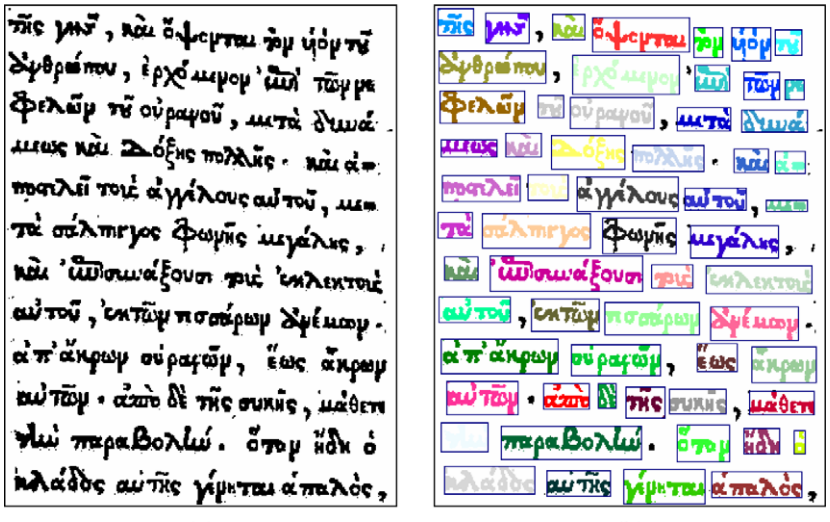


Fig. 11. Example of word segmentation result: (a) original document and (b) extracted word segments.

In order to compare the proposed technique with current state-of-the-art approaches, we implemented an RLSA and a projection profiles based technique. Similar approaches have been used for the word segmentation of historical and degraded machine-printed documents in [3,2]. Additionally we compare our technique with the commercial product FineReader Engine 8.1 [40] and the Open Source OCROpus software [41].

For the RLSA based approach [3], we examine the white runs existing in the horizontal and vertical directions. For each direction, white runs with length less than a threshold are eliminated. For text line detection, the horizontal length threshold is defined as 200% of the average character height while the vertical length threshold is defined as 10% of the average character height. For word detection, these thresholds are set to 50% and 10% of the average character height respectively, while for character detection these thresholds are set to 10% and 10% of the average character height respectively. For the projection profile approach [2], we calculate the horizontal projection profile by summing the foreground pixels in every scan line. After a smoothing procedure, we calculate the local minimums which define the boundaries of the regions which contain the text lines. For every detected text line, similar procedure based on vertical projections is used in order to detect words and characters.

For the purpose of the evaluation, we manually marked and extracted the ground truth on a set of 63 images for the case of evaluating the text line segmentation (3880 text line segments), 43 images for the case of word segmentation (18,654 word segments) and 18 images for the case of character segmentation (29,032 character segments).

The performance evaluation was based on counting the number of matches between the text line, word and character segments detected by the algorithm with those in the ground truth [37,38]. For each line, word, character in the resulted image of the proposed technique and the corresponding one in the groundtruth image a matching score is calculated. This is the ratio of the number of pixels that belong to this line, word, character in both images, to the

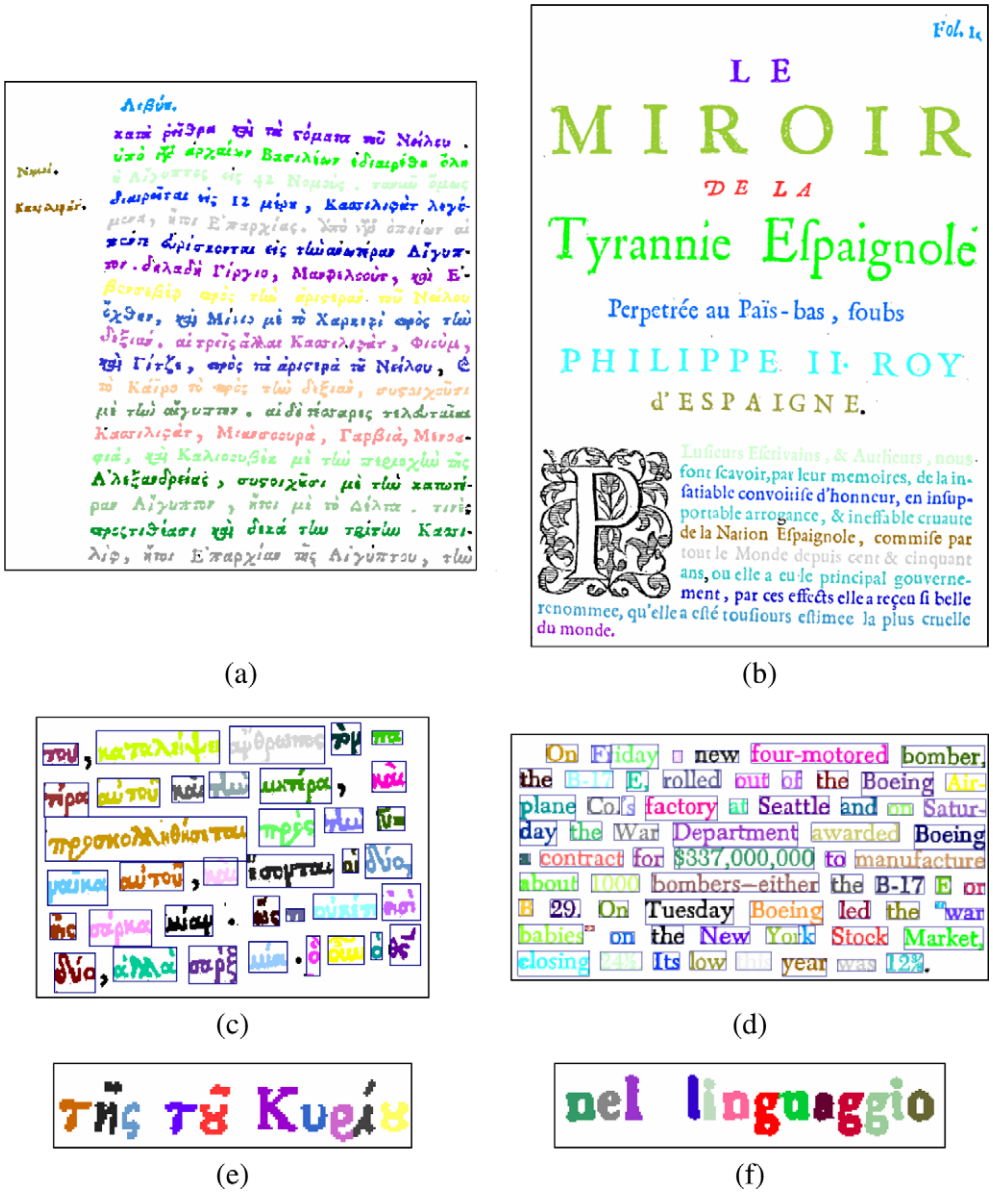


Fig. 13. Text line, word and character segmentation examples extracted by the propose method: (a) text line segmentation example of Greek historical document, (b) text line segmentation example of French historical document, (c) word segmentation example of Greek historical document, (d) word segmentation example of English historical document, (e) character segmentation example of Greek historical document and (f) character segmentation example of Roman historical document.

total number of pixels for this line word character according to the groundtruth. We consider a match only if the matching score is equal to or above the evaluator's acceptance threshold T_a (see [38]). The performance was recorded in terms of detection rate and recognition accuracy, while as an overall measure we used the *F*-measure which is a weighted harmonic mean of detection rate and recognition accuracy [39].

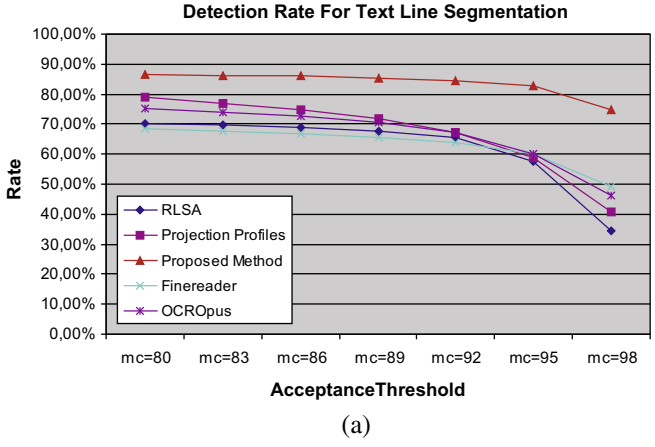
$$F\text{-measure} = \frac{2 \cdot D \cdot R}{D + R} \tag{8}$$

where D is the detection rate and R the recognition accuracy.

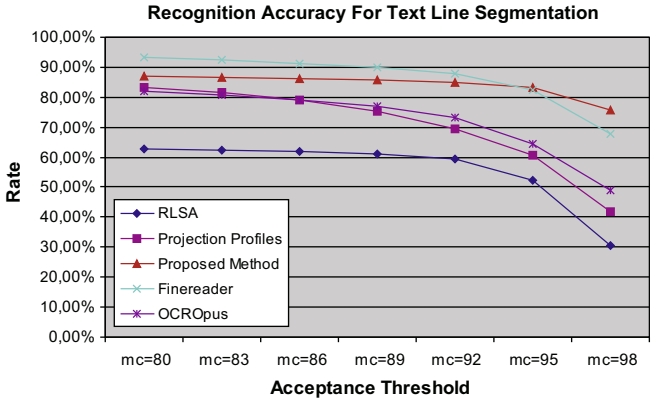
We compared our method with the four other approaches by using various acceptance threshold values which range from 80 to 98 and interval value equal to 3. The overall results are represented in the graphs of Figs. 14–16.

In almost all cases, the proposed technique outperforms the four other approaches.

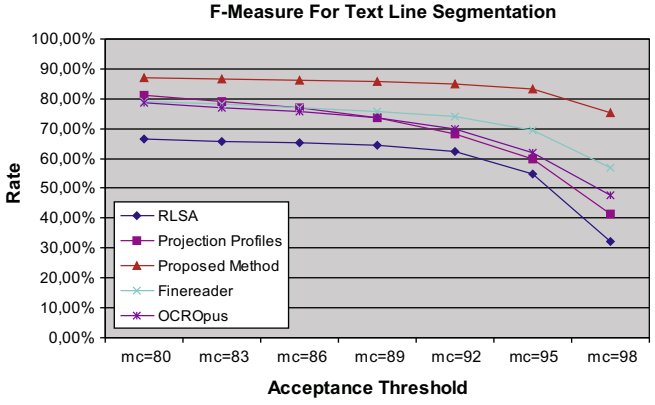
In the case of character segmentation, the RLSA based approach performs better than our method, considering the recognition accuracy measure. Also, in the cases of line and word segmentation, the FineReader technique performs better considering the recognition accuracy measure. Furthermore, it can be noticed that as T_a gets larger the efficiency of our method compared to the four other approaches becomes more obvious.



(a)

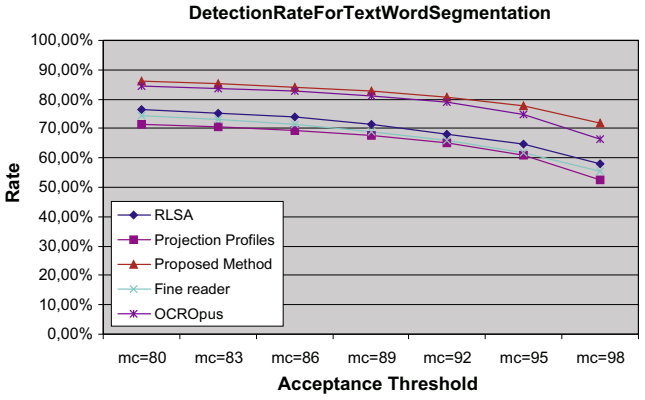


(b)

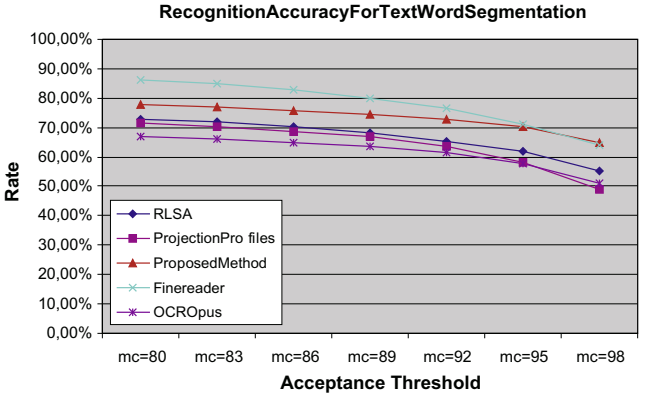


(c)

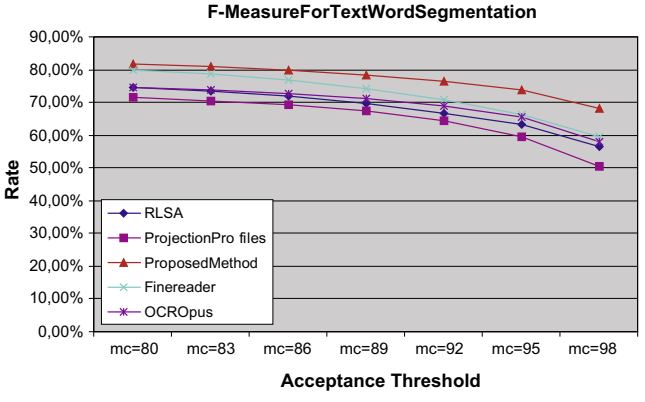
Fig. 14. Graphical depiction of the comparison results for text line segmentation using various acceptance threshold values: (a) detection rate, (b) recognition accuracy and (c) *F*-measure.



(a)



(b)



(c)

Fig. 15. Graphical depiction of the comparison results for word segmentation using various acceptance threshold values: (a) detection rate, (b) recognition accuracy and (c) *F*-measure.

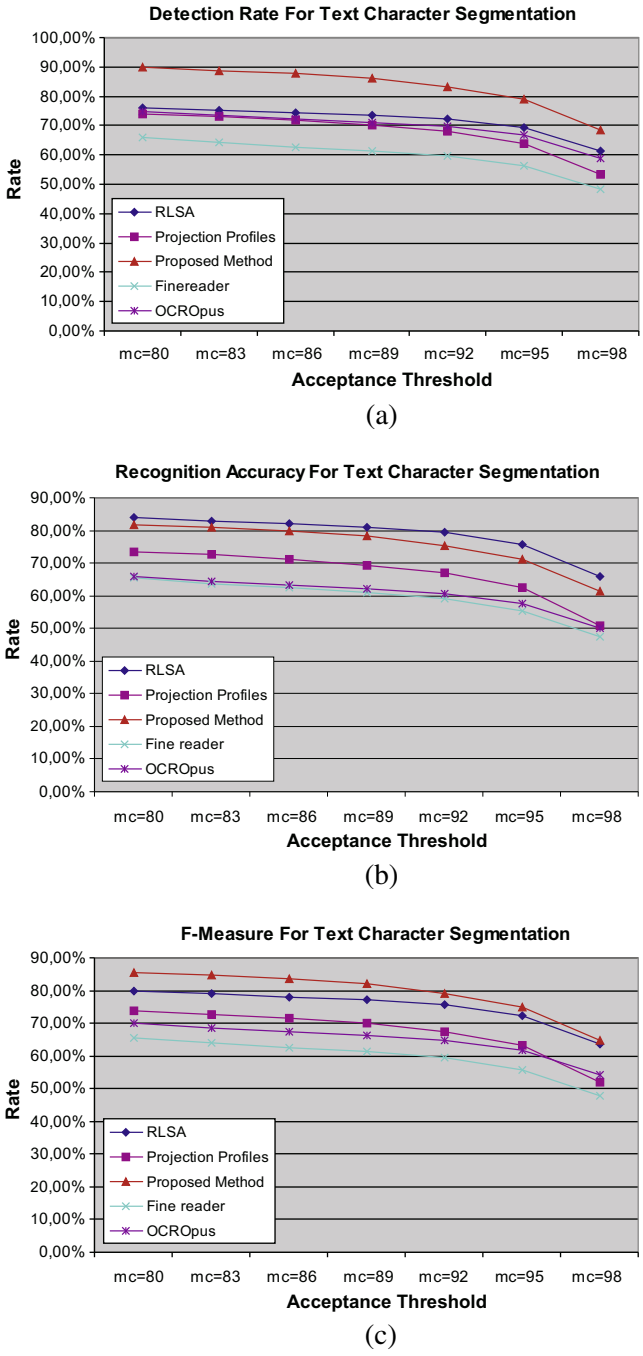


Fig. 16. Graphical depiction of the comparison results for character segmentation using various acceptance threshold values: (a) detection rate, (b) recognition accuracy and (c) *F*-measure.

Also, Tables 2–4 show comparative results where the acceptance threshold is set to a constant value, $T_a = 90$. Taking into account the *F*-measure metric, the proposed algorithm outperforms the four other state-of-the-art approaches in all segmentation cases.

Additionally, in order to evaluate the performance of the proposed method against the most common segmentation problems of historical machine-printed documents, we have defined the following seven categories of documents:

1. Multi column documents.
2. Noisy documents.

Table 2
Comparative results for text line segmentation (63 images – 3880 text lines).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)
RLSA based technique	67.5	54.6	60.4
Projection profiles based technique	67.8	75.1	71.3
FineReader technique	65.1	89.5	75.3
OCROpus technique	69.9	76.0	72.8
Proposed technique	**85**	**84.6**	**84.8**

Table 3
Comparative results for word segmentation (43 images – 18,654 word segments).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)
RLSA based technique	67.5	65.3	66.4
Projection profiles based technique	64.4	66.1	65.2
FineReader technique	68.0	78.8	73.0
OCROpus technique	80.5	62.8	70.5
Proposed technique	**81.5**	**74.6**	**77.9**

Table 4
Comparative results for character segmentation (18 images – 29,032 character segments).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)
RLSA based technique	71.2	80.2	74.4
Projection profiles based technique	68.6	68.6	68.6
FineReader technique	60.9	60.6	60.7
OCROpus technique	70.8	61.8	65.9
Proposed technique	**84.5**	**77**	**80.6**

Table 5
Results for the categories of segmentation problems (proposed technique).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)	Document category
Lines	88.5	87.3	87.9	Multi column
Words	82.7	72.2	77.1	
Characters	86.7	81.5	84.0	
Lines	80.3	75.8	78.0	Noisy
Words	75.1	71.9	73.5	
Characters	79.6	70.8	74.9	
Lines	83.6	83.8	83.7	Non-constant spaces
Words	78.1	70.7	74.2	
Characters	86.5	78.2	82.1	
Lines	79.1	80.3	79.7	Marginal text
Words	72.3	68.1	70.1	
Characters	86.0	78.6	82.1	
Lines	82.8	80.3	81.6	Various font sizes
Words	82.7	72.1	77.0	
Characters	88.6	76.5	82.1	
Lines	85.6	84.3	84.9	Ornamental characters and graphical illustrations
Words	81.0	72.0	76.2	
Characters	88.3	80.5	84.2	
Lines	83.6	84.2	83.9	Warped – skewed text
Words	76.7	66.6	71.3	
Characters	88.3	79.1	83.5	

3. Documents with non-constant spaces between text lines, words and characters.
4. Documents with marginal text.
5. Documents in which various font sizes coexist.
6. Documents with ornamental characters and graphical illustrations.
7. Documents whose text is warped and/or skewed.

We measured the detection rate, recognition accuracy and *F*-measure, as previously, separately for each category. Tables 5–9 show the results adopted for each of the above document category, for the proposed, the RLSA, the projection profile, the FineReader and the OCROpus technique, respectively. The results show that the proposed method is capable of dealing with all these types of segmentation problems and again outperforms the other four approaches in all cases.

As it can be observed from Table 6, the RLSA based technique does not perform well mainly with noisy documents (*F*-measure for lines, words and character segmentation: 43.7%, 48.3%, and 67.8%) and documents with non-constant spaces (*F*-measure for lines, words and character segmentation: 62.7%, 62.6%, and 75.3%). The projection profiles based method (Table 7) does not

Table 6
Results for the categories of segmentation problems (RLSA based technique).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)	Document category
Lines	71.8	64.7	68.1	Multi column
Words	66.9	66.6	66.8	
Characters	75.9	83.4	79.0	
Lines	45.5	42.1	43.7	Noisy
Words	46.6	50.2	48.3	
Characters	61.5	75.6	67.8	
Lines	63.4	62.1	62.7	Non-constant spaces
Words	63.5	61.7	62.6	
Characters	71.6	79.3	75.3	
Lines	55.9	60.7	58.2	Marginal text
Words	51.7	62.1	56.4	
Characters	75.2	83.6	79.2	
Lines	58.0	46.1	51.4	Various font sizes
Words	64.5	59.9	62.1	
Characters	77.1	81.1	79.1	
Lines	65.6	63.9	64.8	Ornamental characters and graphical illustrations
Words	63.3	66.4	64.8	
Characters	77.6	83.7	80.5	
Lines	67.2	65.7	66.4	Warped – skewed text
Words	58.8	59.4	59.1	
Characters	77.8	85.9	81.6	

Table 7
Results for the categories of segmentation problems (projection profiles based technique).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)	Document category
Lines	72.4	76.1	74.2	Multi column
Words	64.0	66.5	65.2	
Characters	73.3	73.6	73.4	
Lines	65.8	69.2	67.5	Noisy
Words	54.3	44.8	49.1	
Characters	61.1	57.7	59.3	
Lines	66.8	70.5	68.6	Non-constant spaces
Words	60.6	61.2	60.9	
Characters	69.8	69.2	69.5	
Lines	37.5	52.0	43.6	Marginal text
Words	40.9	51.7	45.6	
Characters	68.1	69.3	68.7	
Lines	54.0	59.0	56.4	Various font sizes
Words	61.9	56.5	59.1	
Characters	74.0	68.1	70.1	
Lines	66.1	69.9	67.9	Ornamental characters and graphical illustrations
Words	57.4	62.6	59.9	
Characters	71.12	67.1	69.0	
Lines	62.9	67.1	64.9	Warped – skewed text
Words	50.4	58.6	54.2	
Characters	67.7	69.4	68.5	

Table 8
Results for the categories of segmentation problems (FineReader technique).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)	Document category
Lines	64.6	90.3	75.3	Multi column
Words	70.6	81.8	75.8	
Characters	63.2	60.4	61.8	
Lines	47.5	80.3	59.7	Noisy
Words	53.1	70.5	60.8	
Characters	40.4	48.3	44.0	
Lines	57.5	87.0	69.2	Non-constant spaces
Words	55.5	70.1	61.9	
Characters	56.0	55.9	55.9	
Lines	53.9	82.3	65.1	Marginal text
Words	54.1	74.4	62.6	
Characters	61.9	73.1	67.0	
Lines	69.0	87.5	77.2	Various font sizes
Words	69.2	79.0	73.8	
Characters	66.7	65.1	65.9	
Lines	64.9	88.9	75.0	Ornamental characters and graphical illustrations
Words	63.1	76.7	69.2	
Characters	64.2	65.4	64.8	
Lines	53.8	86.3	66.3	Warped – skewed text
Words	50.2	72.3	59.3	
Characters	66.4	75.9	70.8	

Table 9
Results for the categories of segmentation problems (OCROpus technique).

	Detection rate (%)	Recognition accuracy (%)	*F*-measure (%)	Document category
Lines	68.7	76.9	72.6	Multi column
Words	82.9	67.1	74.2	
Characters	72.6	65.0	68.6	
Lines	74.8	73.8	74.3	Noisy
Words	67.3	53.2	59.4	
Characters	62.8	52.3	57.1	
Lines	64.9	73.1	68.8	Non-constant spaces
Words	74.9	55.6	63.8	
Characters	68.8	59.0	63.5	
Lines	44.7	66.2	53.7	Marginal text
Words	74.4	62.8	68.1	
Characters	71.4	67.3	69.3	
Lines	66.2	73.3	69.6	Various font sizes
Words	76.8	63.4	69.5	
Characters	74.0	62.8	67.9	
Lines	70.8	75.5	73.1	Ornamental characters and graphical illustrations
Words	77.9	62.0	69.0	
Characters	74.3	65.2	69.5	
Lines	57.5	69.2	62.8	Warped – skewed text
Words	76.7	59.4	66.9	
Characters	76.0	70.3	73.0	

perform well mainly with documents having marginal text (*F*-measure for lines, words and character segmentation: 43.6%, 45.6%, and 68.7%) and noisy documents (*F*-measure for lines, words and character segmentation: 67.5%, 49.1%, and 59.3%).

The FineReader technique (Table 8) does not perform well mainly with noisy documents (*F*-measure for lines, words and character segmentation: 59.7%, 60.8%, and 44.0%). Finally, the OCROpus technique (Table 9) does not perform well for documents with marginal text (*F*-measure for lines, words and character segmentation: 53.7%, 68.1%, and 69.3%).

In all these challenging cases (noisy documents, documents with non-constant spaces and marginal text) the proposed technique achieves an *F*-measure performance from 70% to 83% for all segmentation levels.

5. Conclusions

In this paper, we propose a novel methodology for text line, word and character segmentation in historical and degraded machine-printed documents. The proposed technique performs successfully even in cases with text of different size, or with text and non-text areas lying very near and with non-straight, warped and overlapping text lines. Comparative experiments using several historical machine-printed documents prove the efficiency of the proposed technique.

The method introduces the following innovative aspects: (i) use of a novel Adaptive Run Length Smoothing Algorithm (ARLSA) in order to face the problem of complex and dense document layout, (ii) detection of noisy areas and punctuation marks that are usual in historical machine-printed documents, (iii) detection of possible obstacles formed from background areas in order to separate neighboring text columns or text lines, and (iv) use of skeleton segmentation paths in order to isolate possible connected characters.

Although the proposed technique has been successfully tested for a wide variety of degraded documents, our intention is to extend it further, in order to perform even higher rates. A better filtering algorithm for removing noise and graphic elements and a preprocessing procedure that splits touching characters between neighboring text lines, would significantly improve the segmentation results. Furthermore, in cases of warped text, a dewarping algorithm would improve the overall performance of the proposed technique.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement No. 215064 (project IMPACT) as well as from the Greek Ministry of Research funded R&D project POLYTIMO.

References

[1] J.Y. Ramel, S. Leriche, M.L. Demonet, S. Busson, User-driven page layout analysis of historical printed books, International Journal on Document Analysis and Recognition 9 (2–4) (2007) 243–261.
[2] A. Antonacopoulos, D. Karatzas, Semantics-based content extraction in typewritten historical documents, in: Eighth International Conference on Document Analysis and Recognition, 2005, pp. 48–53.
[3] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, S.J. Perantonis, Keyword-guided word spotting in historical printed documents using synthetic data and user feedback, International Journal on Document Analysis and Recognition (IJDAR) 9 (2–4) (2007) 167–177 (special issue on historical documents).
[4] G. Nagy, S. Seth, Hierarchical representation of optically scanned documents, in: Seventh International Conference on Pattern Recognition, 1984, pp. 347–349.
[5] F.M. Wahl, K.Y. Wong, R.G. Casey, Block segmentation and text extraction in mixed text/image documents, Computer Graphics and Image Processing 20 (1982) 375–390.
[6] M. Feldbach, K.D. Tönnies, Line detection and segmentation in Historical Church registers, in: Proceedings of the 6th International Conference on Document Analysis and Recognition, 2001, pp. 743–747.
[7] L. O'Gorman, The document spectrum for page layout analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (11) (1993) 1162–1173.
[8] H.S. Baird, Background structure in document images, Document Image Analysis, World Scientific, 1994, 17–34.
[9] T.M. Breuel, Two geometric algorithms for layout analysis, in: Proceedings of the 5th International Workshop on Document Analysis Systems V, 2002, pp. 188–199.
[10] P.C.V. Hough, Methods and means for recognizing complex patterns, US Patent #3069654, 1962.
[11] R.D. Duda, P.E. Hart, Use of the Hough transform to detect lines and curves in pictures, Communications of the ACM 15 (1) (1972) 11–15.
[12] K. Kise, A. Sato, M. Iwata, Segmentation of page images using the area Voronoi diagram, Computer Vision and Image Understanding 70 (3) (1998) 370–382.
[13] R. Manmatha, J.L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1212–1225.
[14] J. He, A.C. Downton, User-assisted archive document image analysis for digital library construction, in: Seventh International Conference on Document Analysis and Recognition, 2003, pp. 498–502.
[15] Z. Shi, V. Govindaraju, Line separation for complex document images using fuzzy runlength, in: Proceedings – First International Workshop on Document Image Analysis for Libraries – DIAL 2004, 2004, pp. 306–312.
[16] L. Likforman-Sulem, A. Hanimyan, C. Faure, A Hough based algorithm for extracting text lines in handwritten document, in: Proceedings of ICDAR'95, 1995, pp. 774–777.
[17] 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, 2006, pp. 38–45.
[18] Y. Li, Y. Zheng, D. Doermann, Detecting text lines in handwritten documents, in: 18th International Conference on Pattern Recognition, 2006, pp. 1030–1033.
[19] 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, 2006, pp. 12–23.
[20] E. Bruzzone, M.C. Coffetti, An algorithm for extracting cursive text lines, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition, 1999, pp. 749–752.
[21] L. Linkforman-Sumem, A. Zahour, B. Taconet, Text line segmentation of historical documents: a survey, International Journal on Document Analysis and Recognition 9 (2–4) (2006) 1433–2833.
[22] Y. Lu, C. Tan, H. Weihua, L. Fan, An approach to word image matching based on weighted Hausdorff distance, in: Sixth International Conference on Document Analysis and Recognition, 2001, pp. 10–13.
[23] A. Marcolino, V. Ramos, M. Ramalho, J.C. Pinto, Line and word matching in old documents, in: Proceedings of the Fifth IberoAmerican Sympsium on Pattern Recognition, 2000, pp. 123–125.
[24] H. Weihua, C.L. Tan, S.Y. Sung, Y. Xu, Word shape recognition for image-based document retrieval, in: International Conference on Image Processing, 2001, pp. 8–11.
[25] H.C. Park, S.Y. Ok, Y.J. Yu, H.G. Cho, A word extraction algorithm for machine-printed documents using a 3D neighborhood graph model, International Journal on Document Analysis and Recognition 4 (2) (2001) 115–130.
[26] J. Park, V. Govindaraju, Use of adaptive segmentation in handwritten phrase recognition, Pattern Recognition 35 (1) (2002) 245–252.
[27] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, T. Shiose, A novel adaptive morphological approach for degraded character image segmentation, Pattern Recognition 38 (11) (2005) 1961–1975.
[28] S. Liang, M. Shridhar, M. Ahmadi, Segmentation of touching characters in printed document recognition, Pattern Recognition 27 (6) (1994) 825–840.
[29] E. Kavallieratou, E. Stamatatos, N. Fakotakis, G. Kokkinakis, Handwritten character segmentation using transformation-based learning, in: 15th International Conference on Pattern Recognition, vol. 2, 2000, pp. 634–637.
[30] X. Xiao, G. Leedham, Knowledge-based English cursive script segmentation, Pattern Recognition Letters 21 (10) (2000) 945–954.
[31] B. Yanikoglu, P.A. Sandon, Segmentation of off-line cursive handwriting using linear programming, Pattern Recognition 31 (12) (1998) 1825–1833.
[32] U. Pal, A. Belaid, Ch. Choisy, Touching numeral segmentation using water reservoir concept, Pattern Recognition Letters 24 (1–3) (2003) 261–272.
[33] B. Gatos, I. Pratikakis, S.J. Perantonis, Adaptive degraded document image binarization, Pattern Recognition 39 (3) (2006) 317–327.
[34] N. Stamatopoulos, B. Gatos, A. Kesidis, Automatic borders detection of camera document images, in: Second International Workshop on Camera-Based Document Analysis and Recognition (CBDAR'07) Curitiba, Brazil, 2007, pp. 71–78.
[35] Y. Chen, J. Wang, Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11) (2000) 1304–1317.
[36] H.J. Lee, B. Chen, Recognition of handwritten Chinese characters via short line segments, Pattern Recognition 25 (5) (1992) 543–552.

[37] A. Antonacopoulos, B. Gatos, D. Bridson, ICDAR2005 page segmentation competition, in: Eighth International Conference on Document Analysis and Recognition, 2005, pp. 75–79.
[38] I. Phillips, A. Chhabra, Empirical performance evaluation of graphics recognition systems, IEEE Transaction of Pattern Analysis and Machine Intelligence 21 (9) (1999) 849–870.
[39] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, Performance measures for information extraction, in: Proceedings of DARPA Broadcast News Workshop, 1999, pp. 249–252.
[40] ABBYY FineReader OCR. <http://finereader.abbyy.com/>.
[41] The OCRopus open source document analysis and OCR system. <http://code.google.com/p/ocropus/>.