Segmentation of Historical Handwritten Documents into Text Zones and Text Lines

Basilis Gatos, Georgios Louloudis and Nikolaos Stamatopoulos
Computational Intelligence Laboratory, Institute of Informatics and Telecommunications
National Center for Scientific Research "Demokritos"
GR-153 10 Agia Paraskevi, Athens, Greece
{bgat, louloud, nstam}@iit.demokritos.gr

Abstract— *In order to achieve accurate text recognition performance for historical handwritten document images, robust and efficient page segmentation is necessary. In this paper, we propose a text zone detection followed by a text line segmentation method suitable for historical handwritten documents. Our aim is to handle several challenging cases such as horizontal and vertical rule lines overlapping with the text, two column documents and characters of different text lines touching vertically. For text zone detection, we analyze vertical rule lines, connected components as well as vertical white runs while for text line segmentation, we enhance an existing approach based on Hough transform in order to better treat cases of vertical connected characters. Both methods have been proved very promising after an evaluation using a set of historical handwritten documents.*

Keywords- historical document image processing; page segmentation; text line segmentation

I. INTRODUCTION

Historical handwritten document collections are an important source of information related to the history of earlier times. In order to efficiently turn those collections to digital format and provide access to the embedded full-text content, automatic handwritten text recognition (HTR) procedure has to be involved to assist document transcription. Segmentation of historical handwritten documents into text zones and text lines is a crucial and challenging step in the HTR pipeline. Some factors that seriously affect it include the writing style and layout inconsistencies, local skew and adjacent text lines touching. Text lines are usually the input to the recognition module while it is usually essential to first detect the text zones in order to assist text line segmentation. In this paper, we present a novel segmentation module that can be used as input to the majority of HTR systems. It involves text zone as well as text line detection and can produce reliable results since it successfully handles several challenging cases such as horizontal and vertical rule lines overlapping with the text, two column documents and characters of different text lines touching vertically.

This work has been developed in the framework of the EU project tranScriptorium [1] which aims to develop innovative, cost-effective solutions for the indexing, search and full transcription of historical handwritten document images, using HTR technology.

II. RELATED WORK

A reading system requires the detection of main page elements as well as the discrimination of text zones from non-textual ones. Historical handwritten documents do not have strict layout rules and thus, a page segmentation method needs to be invariant to layout inconsistencies, irregularities in script and writing style, skew, fluctuating text lines and variable shapes of decorative entities. Most of the page segmentation state-of-the-art methodologies focus on machine-printed or modern handwritten documents and only a few deal with historical handwritten documents. Representative page segmentation methods can be found in [2].

A layout analysis of 19th Century handwritten documents is proposed in [3]. First, text line detection is modelled as an image segmentation problem by enhancing text line structure using Hough transform and a clustering of connected components so as to make text line boundaries appear. Then, snippets decomposition is performed for page layout analysis. This is based on a first step of content pages classification in five visual and genetic taxonomies, and a second step of text line extraction and snippets decomposition. Snippets are built by linking centroids of borders components. A layout analysis of handwritten historical documents in order to enable searching the archive of the Cabinet of the Dutch Queen (1798 – 1988) is presented in [4]. It is based on the detection of the rule lines of the tables and the page margins. It also uses contour tracing that generates curvilinear separation paths between text lines in order to preserve the ascenders and descenders. A layout analysis of Arabic historical document images using machine learning is proposed in [5]. Simple and discriminative features are extracted in a connected-component level and subsequently robust feature vectors are generated. A multi-layer perceptron classifier is then exploited to classify connected components to the relevant class of text. Finally, in [6] contextual models are used for complex handwritten page segmentation. Stochastic and contextual models are used in order to cope with local spatial variability, while some prior knowledge about the global structure of the document image is taken into account. Two tasks are defined: a) to label the main regions of the manuscripts such as text body, margins, header, footer, page number and marginal annotations, b) to detect pseudowords, deletions, diacritics and background.

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.84

An important handwritten document image processing task is the segmentation into text lines. The overall performance of a handwritten character recognition system strongly relies on the results of the text line segmentation process. Thus, the algorithm employed for this stage is critical for the overall recognition procedure.

We can group existing text line methods into four basic categories: methods making use of the projection profiles, methods that are based on the Hough transform, smearing methods and, finally, methods based on the principle of dynamic programming. Furthermore, several methods exist that cannot be clearly classified in a specific category.

Methods that make use of the projection profiles include [7-8]. In [7], 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. Arivazhagan et al. [8] 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.

Hough transform based methods include [9-11]. Hough transform is a powerful tool used in many areas of document analysis that is able to locate skewed lines of text. Starting from a set of 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 [9-10], or minima points [11] of the connected components.

Smearing methods mainly include the fuzzy RLSA [12] and the adaptive RLSA [13]. The fuzzy RLSA measure is calculated for every pixel on the initial image. By applying this measure, a new grayscale image is created, which is binarized and the text lines are extracted from the new image. The adaptive RLSA [13] 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.

Text line segmentation methods based on the dynamic programming principle were recently presented [14-15]. These methods try to segment text lines by finding an optimal path on the background of the document image travelling from the left to the right edge. Nicolaou and Gatos method [14] is based on the topological assumption that for each text line a path exists from one side of the image to the other which traverses a single text line. The image is first blurred and at a second step tracers are used to follow the white-most and black-most paths from left to right as well as from right to left. The final goal is to shred the image into text line areas. Saabni et al. [15] propose a method which computes an energy map of a text image and determines the seams that pass across and between text lines. Two different algorithms are described (one for binary and one for grayscale images) in [15]. Concerning the first algorithm (binary case), each seam passes on the middle and along a text line and marks the components that make the letters and words of it. At a final step, the unmarked components are assigned to the closest text line. For the second algorithm (grayscale case) the seams are calculated on the distance transform of the grayscale image.

Methods which cannot be grouped to the abovementioned categories include the work of Shi et al. [16]. In this work, the authors make use of the Adaptive Local Connectivity Map: the input to the method is a grayscale image, and 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. Li et al. [17] describe 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. In [18], 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, facing obstruction from characters of text lines. The stripes of areas left unwetted on the image frame are finally labeled for extraction of text lines. Yin and Liu [19], propose an approach which is based on minimum spanning tree (MST) clustering with new distance measures. First, the connected components of the document image are grouped into a tree by MST clustering with a new distance measure. The edges of the tree are then dynamically cut to form text lines by using a new objective function for finding the number of clusters. This approach is totally parameter-free and can apply to various documents with multi-skewed and curved lines. For a more detailed description of text line segmentation methods the interested reader should read [20].

III. Detection of Main Text Zones

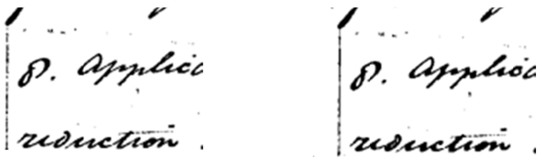
In this step, we aim to segment the historical handwritten page images and detect the main text regions in order to provide them as input to the text line segmentation module that is described in the next section. We first proceed to a binarization using a technique suitable for historical degraded documents [21]. The average character height *AH* is calculated from the binary image based on the histogram of heights of the connected components. Then, we detect vertical text zones by analyzing vertical rule lines as well as vertical white runs of the document image. Finally, we refine and horizontally restrict those zones based on connected component analysis.

A. Detection of Vertical Text Zones based on Vertical Rule Lines

Vertical rule lines are detected based on a fuzzy smoothing method. We first proceed to a vertical smoothing using the Run Length Smoothing Algorithm (RLSA) [22] in order to connect vertical broken rule lines (see Fig. 1). At a next step, pixels that belong to long vertical lines are detected by applying the vertical fuzzy RLSA [23]. 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 vertical direction”. By applying this measure, a new gray scale image is created, which is then binarized, and the vertical rule lines are extracted (see Fig. 2). In order to detect dominant vertical rule lines, we proceed to a vertical projection of the resulting image. Since we want to stress the existence of long vertical lines, we define the vertical black run profile *V*(*x*) as follows:

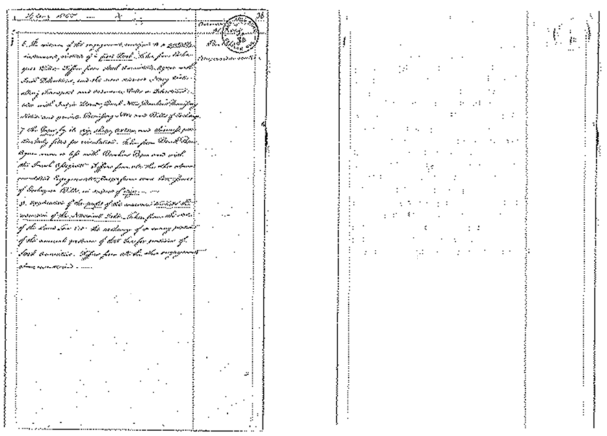
$$V(x) = \sum_{i=0}^{B(x)} L(i,x)^2 \tag{1}$$

where *B(x)* is the number of black runs in the column *x* and *L(i,x)* is the length of the *i*-th black run. An example of vertical projections is presented in Fig. 3a. By selecting only those image columns that have *V(x)* greater than a threshold *T*, we can define the vertical rule line segments of the image (see Fig. 3b). From our experiments, a sufficient value for *T* is (10x *AH*)^2 where *AH* is the average character height. The main text body can be defined in the zone between those consequent vertical rule lines that have maximum distance between them (see Fig. 3b).



(a) (b)

Figure 1. An image portion before (a) and after (b) vertical smoothing.

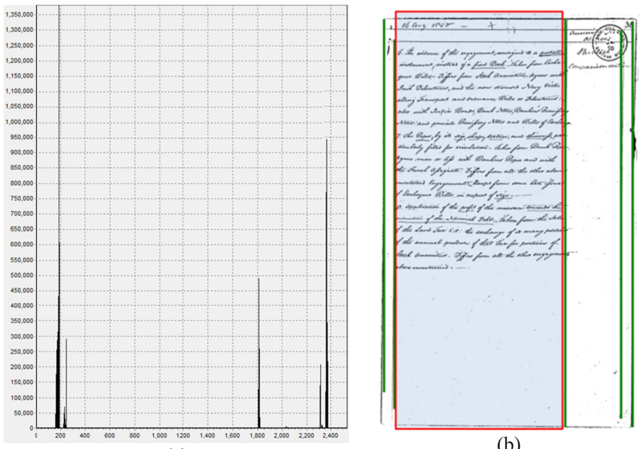


(a) (b)

Figure 2. Vertical fuzzy RLSA: (a) original image; (b) the fuzzy RLSA result.

B. Detection of Vertical Text Zones based on Vertical White Runs

Processing the background of the image can be used to efficiently detect the vertical text zones for the cases that vertical rule lines do not exist. In our approach, we first count the vertical white runs of the image that are of significant length (>image_height/3). Some examples of this procedure for single and two-column documents are presented in Fig. 4. At a next step, we detect and tag image columns that do not have long vertical white runs. If we calculate and sort all the lengths of successive tagged columns, then if the two longer lengths have almost the same value, the document is classified as two-column document. Otherwise, the main text area is detected. Examples of both cases are presented in Fig. 4.



(a) (b)

Figure 3. (a) Vertical black run projections of image in Fig. 2 and (b) the resulting vertical rule lines as well as the detected main text zone.

C. Horizontal Restriction and Refinement of Text Zones

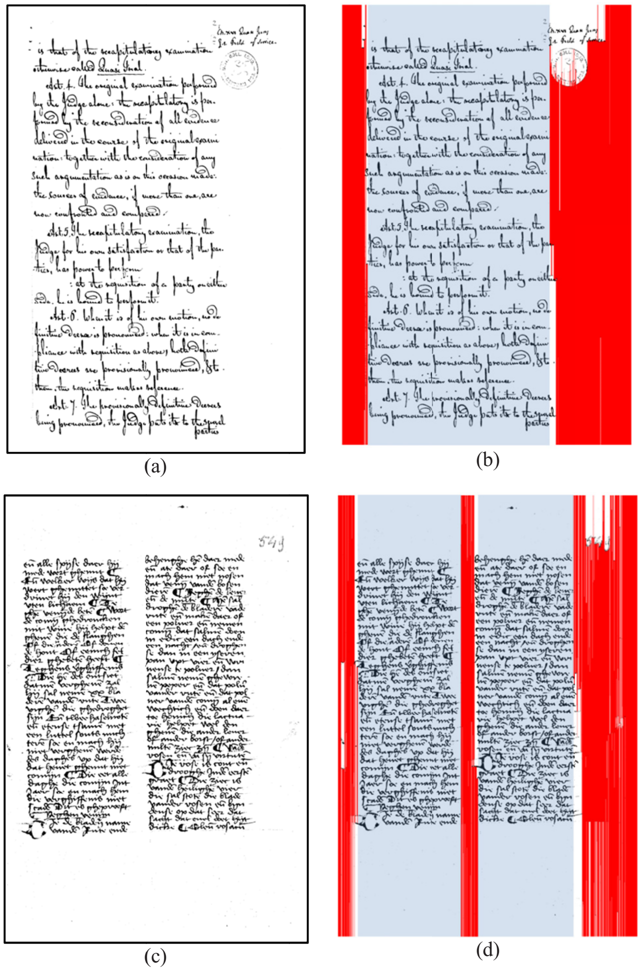
In order to have the exact position of text areas, we need to proceed to a refinement of the result, treating cases of text overlapping with horizontal or vertical rule lines as well as to horizontally restrict the already detected vertical text zones. We first proceed to removing pixels that belong to horizontal or vertical rule lines following the fuzzy RLSA based procedure described in Section III.A. Then, in order to restore character parts that overlap with horizontal or vertical rule lines, we follow the approach in [12, 23]. According to this approach, for the case of horizontal rule lines, if a vertical run is longer than the estimated width of the line, then it is retained. A similar approach is followed for the vertical rule lines.

At a next step, we calculate all connected components of the resulting image. A connected component *i* with bounding box $(cx_1^i, cy_1^i - cx_2^i, cy_2^i)$ belongs to a vertical text zone that is defined by x-offsets x_1 and x_2, only if $cx_2^i \geq x_1$ AND $cx_1^i \leq x_2$. Values x_1 and x_2 are then updated accordingly based on the limits of all valid connected components. In order to horizontally restrict the vertical text zones, we calculate the corresponding y-offsets y_1 and y_2 as follows:

$$y_1 = cy_1^i \text{ where } i{:}\, cy_2^i - cy_1^i \geq AH \text{ AND } cy_1^i = min \tag{2}$$

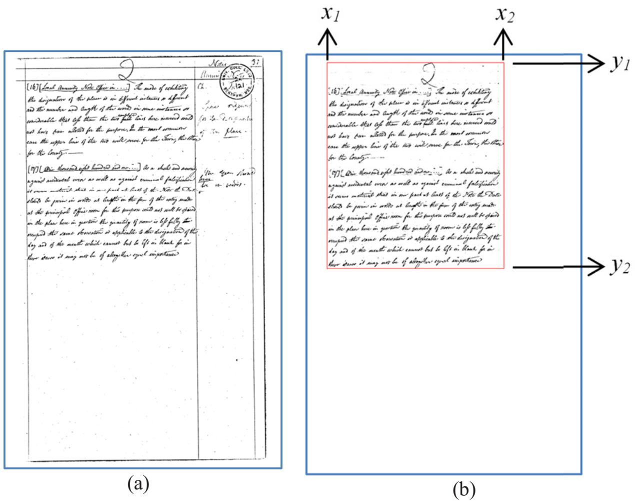
$$y_2 = cy_2^i \text{ where } i{:}\, cy_2^i - cy_1^i \geq AH \text{ AND } cy_2^i = max \tag{3}$$

An example of the detection of main text region is presented in Fig. 5.


(a) (b)

(c) (d)

Figure 4. (a),(c) Original images; (b),(d) their long vertical white runs (in red) and the detected main text zones.



(a) (b)

Figure 5. (a) Binarized image; (b) the detected main text region.

IV. Text Line Segmentation

The proposed method for text line segmentation in handwritten documents is an extension of the work presented in [10], consists of three main steps (pre-processing, Hough transform mapping and post-processing) and is applied to every text region provided by the previously described page segmentation procedure. The main novelty concerns the post-processing step in which an efficient vertical character separation method is presented.

The preprocessing step consists of four stages. Initially, vertical rule line removal is applied [23]. Then, the connected components are extracted and the bounding box coordinates for each connected component are calculated. The average character height ***AH*** is also estimated. The final stage concerns the division of the connected components domain into three sub-domains, namely "Subset 1", "Subset 2" and "Subset 3". "Subset 1" corresponds to the majority of the normal character components. The motivation for "Subset 1" definition is to exclude accents and components that are large in height and are likely to belong to more than one text line. "Subset 2" contains all large connected components. Large components are either capital letters or characters from adjacent text lines touching. Finally, "Subset 3" should contain characters as accents, punctuation marks and small characters (for details see [10]).

The Hough transform mapping step is responsible for the detection of lines that intersect with the connected components of each text line. Only connected components of "Subset 1" are fed to the Hough transform. The number of points per connected component that contribute to the Hough domain is proportional to its width [10]. A result of this step is presented in Fig. 6.

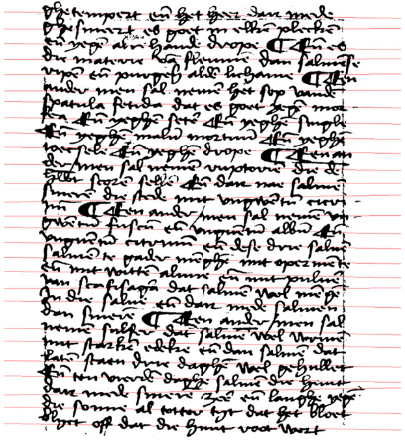


Figure 6. Result of the Hough Transform mapping step.

The post-processing step consists of two stages. At the first stage a merging technique over the result of the Hough transform is applied to correct possible false alarms and connected components of "Subset 1" that were not clustered to any text line are checked to see whether they create a new text line that the Hough transform did not reveal.

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*. All components of this subset mainly belong to *n* detected text lines (*n*>1). Our novel method for splitting these components consists of the following steps:

(A) Calculate y_i, which are the average *y* values of the intersection of detected line *i* and the connected component's bounding box (*i* =1…*n*) (see Fig. 7a, 7f).

(B) Compute the skeleton of the connected component (Fig. 7b, 7g), detect all junction points and remove them from the skeleton (Fig. 7c, 7h). If no junction point exists in the zone between y_i and y_{i+1}, remove all skeleton points in the center of the zone.

(C) Assign the skeleton connected components to the closest line y_i (Fig. 7d, 7i).

(D) Each pixel of the initial image (Fig. 7a, 7f) inherits the text line id of the closest skeleton pixel (Fig. 7e, 7j).

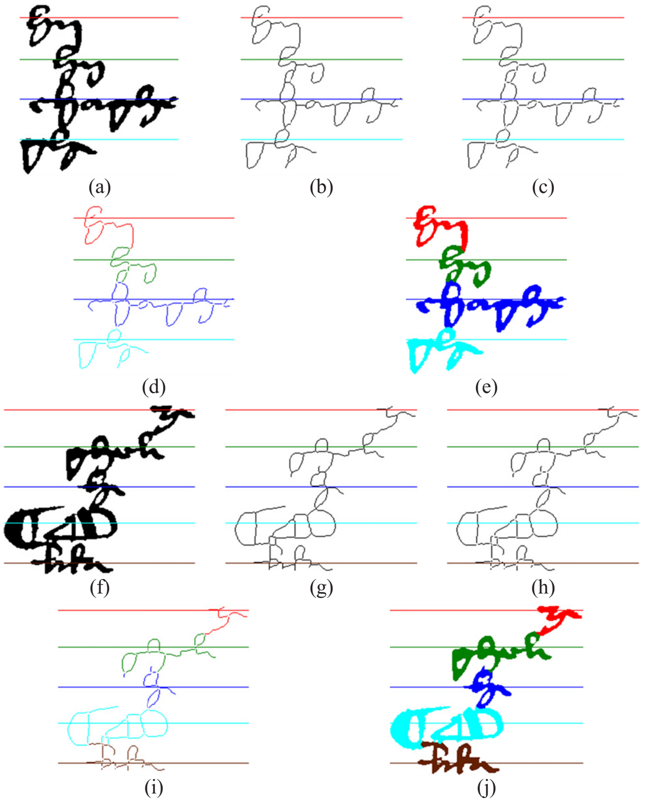

(a) (b) (c)
(d) (e)
(f) (g) (h)
(i) (j)

Figure 7. Steps for assigning pixels of vertically connected characters to corresponding text lines (colored lines): (a, f) Initial image, (b, g) skeleton, (c, h) removal of junction points, (d, i) assignment of skeleton connected components to closest line, (e, j) final result.

V. EXPERIMENTAL RESULTS

In order to record the efficiency of the developed text zone detection method, we created a set of 300 handwritten historical images from the tranScriptorium project [1] that have representative layout challenges (one or two column documents, side notes, horizontal and vertical rule lines, seals etc.). At a next step, we manually marked the main text zones. These correspond to one area for single column documents or two areas for two column documents (see Fig. 8).

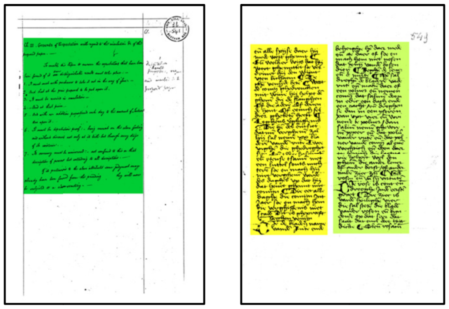


Figure 8. The ground truth of main document text zones.

The performance evaluation of the developed method is based on comparing the number and the coordinates of areas between the result and the ground truth. A result is assumed correct only if the number of areas is the same and the coordinates of the detected and ground truth areas are almost the same (their absolute difference is less than image_width / 30 for the x-coordinates and less than image_height / 30 for the y-coordinates). The results showed that in 254 out of the 300 images, the main text zones were correctly detected. This leads to an accuracy of 84.7%.

For the evaluation of the text line segmentation methods we follow the same protocol which was used in the ICDAR 2013 Handwriting Segmentation Competition [24]. The performance evaluation method is based on counting the number of one-to-one matches between text lines detected by the algorithm and text lines in the ground truth (manual annotation of correct text lines). We use a MatchScore table whose values are calculated according to the intersection of the ON pixel sets of the result and the ground truth. Let G_i the set of all points of the *i* ground truth text line, R_j the set of all points of the *j* text line result, *T(s)* a function that counts the elements of set s. Table *MatchScore(i,j)* represents the matching results of *i* ground truth text line and the *j* text line result as follows:

$$MatchScore(i,j) = \frac{T(G_i \cap R_j)}{T(G_i \cup R_j)} \tag{4}$$

We consider a region pair as a one-to-one match only if the matching score is equal to or above the evaluator's acceptance threshold Ta. If *N* is the count of ground-truth elements, *M* is the count of result elements, and *o2o* is the number of one-to-one matches, we calculate the detection rate (*DR*) and recognition accuracy (*RA*) as follows:

$$DR = \frac{o2o}{N}\ ,\ RA = \frac{o2o}{M} \tag{5}$$

The F-Measure metric (*FM*) can be extracted if we combine the values of detection rate and recognition accuracy:

$$FM = \frac{2 * DR * RA}{DR + RA} \tag{6}$$

The proposed text line segmentation has been tested on 433 handwritten document images derived from the tranScriptorium project [1]. For all these document images, we have manually created the corresponding text line segmentation ground truth using the Aletheia tool [25] and stored it using the PAGE xml format. The total number of text lines appearing on those images was 11468. The acceptance threshold for the text line segmentation evaluation is set to Ta = 0.95. For the sake of comparison, we also tested the winning method of the ICDAR2013 Handwriting Segmentation Competition [24] for the text line segmentation task as well as the method proposed by Nicolaou et al. [14]. Table I shows comparative experimental results in terms of detection rate (*DR*), recognition accuracy (*RA*) and F-Measure (*FM*). As it can be observed from Table I, the proposed method outperforms the other two approaches achieving detection rate 83.08%, recognition accuracy 86.35% and *FM* 84.68%.

TABLE I. COMPARATIVE EVALUATION RESULTS

Method	***N***	***M***	***o2o***	***DR* (%)**	***RA* (%)**	***FM* (%)**
Proposed	11468	11034	9528	83.08	86.35	84.68
Winner [24]	11468	11051	9418	82.12	85.22	83.64
Nicolaou [14]	11468	10022	8617	75.13	85.98	80.19

VI. CONCLUSIONS

In this paper, we propose a text zone detection followed by a text line segmentation method suitable for historical handwritten documents. For text zone detection, we analyze vertical rule lines, connected components as well as vertical white runs while for text line segmentation, we enhance an existing approach based on Hough transform in order to better treat cases of vertical connected characters. Both methods have been proved very promising after an evaluation using a set of historical handwritten documents since an accuracy of ~ 85% is recorded for both cases.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 600707 - tranScriptorium.

REFERENCES

[1] http://www.transcriptorium.eu

[2] F. Shafait, D. Keysers and T. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms", IEEE Trans. Patt. Anal. Mach. Intell. Vol. 30, no. 6, pp. 941-954, June 2008.

[3] V. Malleron, V. Eglin, H. Emptoz, S. Dord-Crousle, P. Regnier, "Text Lines and Snippets Extraction for 19th Century Handwriting Documents Layout Analysis", 10th Int. Conf. on Document Analysis and Recognition, pp. 1001-1005, 2009.

[4] M. Bulacu, R. van Koert, L.Schomaker, T. van der Zant, "Layout analysis of handwritten historical documents for searching the archive of the Cabinet of the Dutch Queen", 9th Int. Conf. on Document Analysis and Recognition, pp. 357-361, 2007.

[5] S. S. Bukhari, T. M. Breuel, A Asi, J. El-Sana, “Layout Analysis for Arabic Historical Document Images Using Machine Learning”, 13th Int. Conf. on Frontiers in Handwriting Recognition, pp. 635-640, 2012.

[6] S. Nicolas, T. Paquet and L. Heutte, "Complex Handwritten Page Segmentation Using Contextual Models“, 2nd Int. Workshop on Document Image Analysis for Libraries, pp. 46-59, 2006.

[7] E. Bruzzone, and M. C. Coffetti, "An Algorithm for Extracting Cursive Text Lines", 5th Int. Conf. on Document Analysis and Recognition, pp. 749-752, 1999.

[8] M. Arivazhagan, H. Srinivasan, and S. N. Srihari, "A Statistical Approach to Handwritten Line Segmentation", Document Recognition and Retrieval XIV, Proc. of SPIE, pp. 6500T-1-11, 2007.

[9] L. Likforman-Sulem, A. Hanimyan, and C. Faure, "A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents", 3rd Int. Conf. on Document Analysis and Recognition, pp. 774-777, 1995.

[10] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, "Text line and word segmentation of handwritten documents ", Pattern Recognition, Vol. 42, Issue 12, pp. 3169-3183, 2009.

[11] Y. Pu, and Z. Shi, " A Natural Learning Algorithm Based on Hough Transform for Text Lines Extraction in Handwritten Documents", 6th Int. Workshop on Frontiers in Handwriting Recognition, pp. 637-646, 1998.

[12] Z. Shi, and V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Runlength", 1st Int. Workshop on Document Image Analysis for Libraries, pp. 306-312, 2004.

[13] M. Makridis, N. Nikolaou and B. Gatos, "An Efficient Word Segmentation Technique for Historical and Degraded Machine-Printed Documents", 9th Int. Conf. on Document Analysis and Recognition, pp. 178-182, 2007.

[14] A. Nicolaou and B. Gatos, "Handwritten Text Line Segmentation by Shredding Text into its Lines", 10th Int. Conf. on Document Analysis and Recognition, pp. 626-630, 2009.

[15] R. Saabni, A. Asi, J. E. Sana, "Text line extraction for historical document images", Pattern Recognition Letters, Vol. 35, Iss. 1, pp. 23-33, 2014.

[16] Z. Shi, S. Setlur, and V. Govindaraju, "Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map", 8th Int. Conf. on Document Analysis and Recognition, pp. 794-798, 2005.

[17] Y. Li, Y. Zheng, and D. Doermann, "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents", IEEE Trans. Pattern Anal. Machine Intell. (T-PAMI), vol. 30, no. 8, pp. 1313-1329, August, 2008.

[18] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, and D.K. Basu, "Text Line Extraction from Multi – Skewed Handwritten Documents", Pattern Recognition, vol. 40, no. 6, pp. 1825-1839, June 2007.

[19] F. Yin, and C.-L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning", Pattern Recognition, Vol. 42, no. 12, pp. 3146-3157, December 2009.

[20] L. Likforman - Sulem, A. Zahour, and B Taconet, "Text Line Segmentation of Historical Documents: A Survey", International Journal on Document Analysis and Recognition (IJDAR), vol. 9, no.2-4, pp. 123-138, April 2007.

[21] B. Gatos, I. Pratikakis, and S.J. Perantonis, "Adaptive degraded document image binarization", Pattern Recognition, vol. 39, no. 3, pp. 317–327, Mar. 2006.

[22] F.M. Wahl, K.Y. Wong, and R.G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", Computer Graphics and Image Processing, vol. 20, pp. 375-390, 1982.

[23] Z. Shi, S. Setlur, and V. Govindaraju, "Image Enhancement for Degraded Binary Document Images" 11th Int. Conf. on Document Analysis and Recognition, pp.895-899, 2011.

[24] N. Stamatopoulos, G. Louloudis, B. Gatos, U. Pal and A. Alaei, "ICDAR2013 Handwriting Segmentation Contest", 12th Int. Conference on Document Analysis and Recognition, pp. 1402-1406, 2013.

[25] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Aletheia - An Advanced Document Layout and Text Ground-Truthing System for Production Environments", 11th Int. Conf. on Document Analysis and Recognition, pp. 48-52, 2011.