

A Hybrid Method for Text Line Extraction in Handwritten Document Images

Ehsan Kiumarsi

Electrical Engineering Faculty
Shahid Bahonar University, Iran
kiumarsi.e@gmail.com

Alireza Alaei

Griffith Institute for Tourism, and School of ICT
Griffith University, Australia
a.alaei@griffith.edu.au

Abstract—Text line segmentation in handwritten document image, as one of the preliminarily steps for document image recognition, is a challenging problem. In this paper, a hybrid method for text line extraction in handwritten document images is presented. Initially, a connected component (CC) labelling method following by a CC filtering is employed to extract a set of CCs from the input document image. A new distance measure is introduced to compute normal distances between the extracted CCs. By traversing the normal distance matrix from both the right and left directions, half-chains of CCs are constructed. The CCs half-chains are merged to obtain CCs full-chains. From the extracted full-chains separator lines are obtained. A gradient metric is proposed to detect and remove touching text lines. Using remaining separator lines the adaptive projection profile of the image is computed. Based on the projection profile, coarse text line extraction is performed. Finally, a fine text lines extraction is performed by applying a postprocessing step. To evaluate the method, two benchmarks named ICDAR2013 handwriting segmentation contest, and Kannada datasets composed of handwritten document images in English, Greek, Bengali, and Kannada languages were considered for experimentation. Experimental results indicate a promising performance was obtained compared to some of the state-of-the-art methods.

Keywords: Text line extraction, handwritten document image, connected component grouping, projection profile analysis

I. INTRODUCTION

Handwritten text line extraction is one of the major problems in document image analysis. Handwritten documents commonly include multi-skewed text lines, different inter-line distances and touching text lines. There are various methods presented in the literature to tackle the problem of text line segmentation in handwritten document images [1-11]. In general, text line segmentation methods in the literature can be categorized into four main groups: i) projection-based, ii) smearing-based, iii) Hough transform-based, and iv) grouping methods.

Projection profile is a classical approach in text line extraction and many researchers used it in different ways. In projection-based methods, text region and white space in a document image are detected based on the image projection profile. The projection profiles can be calculated globally or locally in different orientations from the whole image or different parts of document image. The approach is simple and time complexity is relatively low in simple projection based methods [5]. In [1], a block-based projection profile method was introduced. At first, a document image was divided into vertical strips. Then, based on the strips' vertical projection profile, text zones

were detected. Text zones in adjacent strips with some refinements were merged to form text lines' regions. In [2], after a preprocessing step, in which small CCs were removed, and large CCs were partitioned into medium CCs based on the vertical projection profile, the image energy map was calculated based on the Euclidean distance between CCs. By connecting the center of the CCs placed in the similar energy map, the base lines of the text were obtained. Small pieces were also linked to the closest region.

In smearing based methods document images were blurred to interconnect text regions in each text line in order to make text line separation task easier. Global smearing, affects all parts of image equally. It may cause improper connections between adjacent text lines, especially in the case of close and wavy text lines. Therefore, smearing was performed on stripes obtained by splitting each document image into a number of strips. This process generally includes more regulating parameters and needs more computation time. In [3], a document image was initially divided into a number of vertical strips based on the gap between text lines. Then, a piece-wise filtering algorithm followed by a binarization method was performed on each strip. Employing a dilation method, text lines were extracted based on separating lines obtained by applying a thinning algorithm. At last, touching and overlapping CCs were detected and separated based on the average CCs' height. In [4], employing a skew correction the document image was smeared. Then, the smeared image was binarized to indicate regions of text lines. These regions form text line masks. However, not all the masks necessarily contain a single text line. So, based on the frequency and average of the vertical run-length on each mask, a criterion was obtained for evaluating the same mask. Each of the masks that satisfies the condition was selected as a text separator mask and the smearing operation was repeated with a less degree of smearing. When all of the text line masks were specified or there was no effect on the image while the degree of filtering was decreased the operation stops. After a post-processing step to connect multi-part masks, text lines were separated by the obtained masks.

In Hough transform based methods, the skew angle of text lines are detected through the Hough transform concept enabling text line extraction process. Therefore, this type of method is usually a combination of Hough transform and another text line extraction method. In [5], a hybrid algorithm of Hough transform and grouping method was proposed. In the first step after preprocessing, CCs were classified into three subsets based on their sizes. The middle-size CCs in the next step were used as the candidates to determine initial text line paths. The paths

were detected using the Hough transform method. Finally, the rest of CCs were assigned to the nearest text line and potential faults were compensated. Hough transform-based methods cannot perform well on documents containing multi-directional text lines. Moreover, they may not work well in case of text images in languages with many dots or strokes. In addition, this type of methods has a high time complexity.

In the grouping methods presented in the literature, a document image was divided into a list of text parts, e.g., connected components and they were then grouped to construct text lines. In [6], at the first step a CC labelling technique was employed and distances between CCs were calculated. By employing a minimum spanning tree clustering on the calculated distances the CCs were grouped to extract text lines. In [7], a CC grouping was introduced using vertical overlapping of CCs to extract Arabic handwritten text lines. Initially, the bounding boxes of CCs were obtained. Based on the vertical overlapping of each CC with its neighbours, text lines were extracted. Grouping methods provide a better performance on wavy text lines. This type of methods, however, does not work well on touching text lines. On the other hand, because of making local decision and working on image particles, such as connected components, it is probable to make failure in crowded text images [8]. Moreover, the methods in this category are often based on heuristic, which reduces their robustness.

In [9], a method is presented using fully convolutional network (FCN) to extract the text lines. In this method, the document image passes through some convolution and pooling layers. In this process, the line map of the image is obtained using some filters. This line map is then binarized with the Niblack method. With Skeleton operator, the paths of the text lines are obtained. Of course, these paths might be partitioned in few places that are connected before performing on the image. If a connected component is intersected only with a text string, it corresponds to the same text line. If a CC does not intersect with any text string, it is linked to the nearest text line. Otherwise, it is separated using the LAG method in [16].

In this paper, a hybrid method based on the combination of grouping and projection profile approach for text line extraction from handwritten document is proposed. Because of the high precision performance of the grouping approach, we used a grouping method in the first step of the proposed method to find initial text lines. Since, there are some touching or overlapping text lines, some groups of CCs may be assigned to more than one text line. To handle this problem, the text lines containing this kind of CCs are removed from the initial detected text lines. Then, using the paths of the remaining text lines, an adaptive projection profile is employed on the image to extract all the text lines. Adaptive projection profile is employed to obtain more precise and reliable projection profiles for the document images composed of text lines with various skews. Finally, a post-processing step is performed to compensate probable failures in the proposed text line extraction method. As all the steps used in the proposed text line segmentation method are of linear complexity, time complexity of the method is of $O(n)$, where n is the number of pixels in the image. Therefore, the proposed method is quite fast.

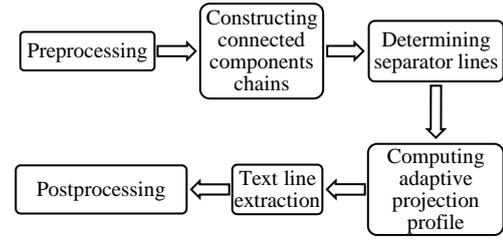


Fig. 1. Block diagram of the proposed method.

The rest of the paper is laid out as follows: In Section II, the proposed method is discussed in detail. Section III describes the dataset as well as experimental results and findings. Finally, Section IV provides some conclusions and future work.

II. PROPOSED METHOD

The block diagram of the proposed hybrid text line segmentation method is presented in Fig. 1. As shown in Fig. 1 the proposed method consists of six steps including preprocessing, creating connected components chains, determining separator lines, calculating projection profile, text line extraction, and postprocessing. A CC analysis is first employed on a binary document image and small CCs are also removed from the document image. In the next step, initial CC half-chains are created from both the left and right sides of the document image by connecting the CCs. Then, each pair of half-chains with at least one common CC are merged to make complete CC chains. The CC chains are used to form line paths. Line paths are estimated to a first order polynomial. By using these polynomials, a projection profile of document image is calculated and image is split based on the project profile valleys. Finally, multi labelled

CCs are treated to extract text lines. Details of each step are presented in the following subsequent subsections.

A. Preprocessing

Starting with a binary handwritten document image (Fig. 2.a) a CC labelling is performed on the input image to extract all the CCs. In cases where the document images are color or gray, the Otsu threshold method can be used to convert the images to binary. Small CCs, such as dash and dot, with sizes (number of pixels) less than a threshold are then removed from the image. The threshold value is computed based on the average area (0.1 of the average area) of all the CCs of the image. We noted that in the case of Arabic/Persian documents which generally have a lot of dots, dots are essential for understanding the meaning of documents, but these small CCs have generally no role in text line path specification.

As a result, the remaining n CCs in the document image are defined as set C .

$$C = \{CC_1, CC_2, \dots, CC_n\} \quad (1)$$

B. Constructing Connected Components Chains

As a text line is generally composed of a set of adjacent CCs, a function as F is defined to determine an adjacent connected component say CC_y for the given CC, e.g., CC_x , in a text line:

$$F(CC_x) = CC_y \quad (2)$$

Every CC, e.g., CC_k , is composed of a number of pixels, p_{kl} , represented by their coordinates (i_{kl}, j_{kl}) which is defined as follows:

$$CC_k = \{p_{k1}, p_{k2}, p_{k3}, \dots\}, \quad p_{kl} = (i_{kl}, j_{kl}) \quad (3)$$

To measure distances between CCs, two points on the right and left sides of a CC, are determined (Fig. 3). To define these two points, boundaries of every CC are defined as follows:

$$i_{kU} \triangleq \{i_{kl} | i_{kl} \leq i_{kv} \forall p_{kv} \in CC_k\} \quad (4)$$

$$i_{kD} \triangleq \{i_{kl} | i_{kl} \geq i_{kv} \forall p_{kv} \in CC_k\} \quad (5)$$

$$j_{kL} \triangleq \{j_{kl} | j_{kl} \leq j_{kv} \forall p_{kv} \in CC_k\} \quad (6)$$

$$j_{kR} \triangleq \{j_{kl} | j_{kl} \geq j_{kv} \forall p_{kv} \in CC_k\} \quad (7)$$

where i_{kU} and i_{kD} are, respectively, the upper and lower boundaries, and j_{kL} and j_{kR} represent the left and right boundaries of the CC_k , accordingly. It may be noted that j_{kL} and j_{kR} are the minimum and maximum values of j_{kl} obtained from the set of the points in CC_k . The bounding box of each CC is defined using these four points. The Most Right Point (MRP_k) and the Most Left Point (MLP_k) of the k -th CC are computed using the following equations.

$$MLP_k = \left(\frac{i_{kU} + i_{kD}}{2}, j_{kL} \right) \quad (8)$$

$$MRP_k = \left(\frac{i_{kU} + i_{kD}}{2}, j_{kR} \right) \quad (9)$$

Moreover, the CCs obtained from the document image are ordered from the left to right based on the left point positions:

$$j_{kL} \leq j_{hL} \quad \forall k < h \quad (10)$$

In (8) and (9), the vertical values are defined as the mid-values of the upper and lower boundaries, as in case of the touching CCs or the CCs representing upper case characters with a large height two adjacent text lines are more likely merged.

Distance between two points can be measured using either horizontal, vertical, or Euclidean distance between the two points. Considering the horizontal distance criterion alone makes CCs stacked in vertical chains which is also misleading. Moreover, the vertical criterion alone is not feasible, as two text lines -due to the slope of the text lines- can also fall into one category in addition to the missing of some CCs. The Euclidean distance measure seems to be a good measure to compute the distance between CCs. But since text lines paths may have a skew angle of less than 45 degrees, the vertical distance between the CCs is more important than the horizontal distance between them. Hence, we propose to emphasize the effect of vertical distance in the Euclidean distance equation with a coefficient. Therefore, in our proposal the distance between the h -th and k -th CCs is computed as follows:

$$d_{hk} = \sqrt{a * (i_h - i_k)^2 + (j_h - j_k)^2} \quad (11)$$

where i and j are vertical (y) and horizontal (x) coordinates, respectively, and a is a weight coefficient. As text lines are usually written horizontally, coefficient a must be much greater than 1. The optimal value for

coefficient a is found based on experimental results on training data. By computing distances between all the CCs a symmetric distance matrix is obtained.

The following functions are defined to determine the nearest CC of every CC from the left to right and from the right to left directions:

$$F(CC_h) = \{CC_k | h < k, d_{hk} \leq d_{hc} \forall c > h\} \quad (12)$$

$$F(CC_k) = \{CC_h | h < k, d_{hk} \leq d_{ck} \forall c < k\} \quad (13)$$

To construct half-chains CCs, the traversing process starts from the CCs located in the left and right sides of the document. As the end points of the text lines in a document image are not given, we must stop traversing before the end of the text lines. Therefore, in searching from each side, we only examine a fraction of the total number of CCs; $p * n$. Parameter n is the number of CCs and p is a real number between zero and one to control the search process. Employing these functions on the constructed distance matrix, half-chains are formed (Fig. 4). Then, every two sets of CCs (half-chains) with at least one CC in common are merged to construct a full-chain. Fig. 2.b represents the result of constructing connected components chains.

C. Determining Drawing Lines

Merging half-chains may result in some multi-line groups, especially in cases of touching CCs between two text lines or short text lines, e.g., see the last text line shown in Fig. 2.b. These groups of CCs need to be determined and treated, as their paths do not accurately simulate baselines. To do so, the route of each group of CCs in a full-chain is created by connecting MLP_k to MRP_k points of the CC_k followed by connecting the most right point of CC_k (MRP_k) to the most left point (MLP_{k+1}) of its adjacent CC (CC_{k+1}) on each chain of CCs. Here route and path are different. A path is a straight line that crosses most of the connected components of a text line, whereas a route is not a straight line as it is created by connecting MRPs to MLPs.

As two text lines in a single group are usually overlap vertically, their horizontally adjacent CCs have a strong gradient in a group route. The gradient (slope) threshold (Eq. 14) is defined in such a way that can remove text lines with unconventional slope paths. However, to avoid deleting the lines with slopes higher than the average slope a y-intercept is added to the threshold value. This value is associated with the maximum slope of the lines to have a kind of adaptive threshold value. The higher the maximum slope of the lines, the greater the threshold value and the higher the variance, the lower the threshold value.

$$g_{th} \triangleq b * \frac{\max(G)}{\text{std}(G)} + \text{mean}(G) \quad (14)$$

where G is the set of the maximum gradients obtained from all the routes in the document image and b is a coefficient. A route with a maximum gradient less than g_{th} is considered as a single line route and a route with a maximum gradient greater than or equal to g_{th} is determined as multi-line. The resultant remaining text line paths are shown in Fig. 2.c. Some single text line routes may also be removed in this process. However, it does not

affect the next processing steps in the proposed text line extraction method.

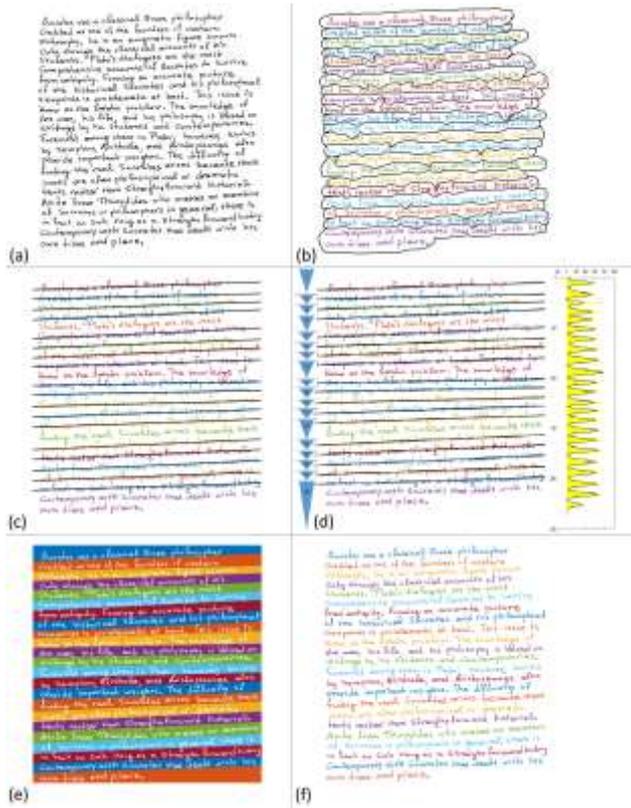


Fig. 2. (a) a sample binary handwritten document image, (b) primary connected component grouping for text line path detection, (c) selected text line paths to calculate adaptive projection profile, (d) text line paths with travelling range for each one and calculated projection profile, (e) extracted text lines using the projection profile coming with some failures in high connected components, (f) the final detected text lines.



Fig. 3. A CC sample and its MRP_k and MLP_k points.

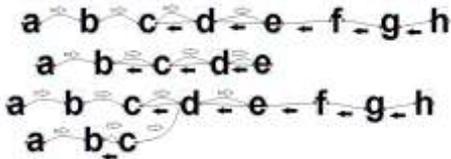


Fig. 4. CCs and their determined bounding box, the most right and left points and the route of each text line.

D. Computing Adaptive Projection Profile

Using the separator lines orientation the adaptive horizontal projection profile of the document image is computed (Fig. 2.d). Due to the persistent and random change of the number of text pixels in the image rows, projection profile has many minima and maxima. However, some of these extremes are not appropriate for the purpose of separating text lines. Therefore, the projection profile should be smoothed so that for each text

line only one maximum remains. Some researchers have used Gaussian [10] and moving average [11] filters to smooth the projection profile signal. Here we use the Savitzky-Golay [12] method, as this filter preserves the maxima better. In Fig. 2.d, the projection profile is shown in yellow and the smoothed in black line. In the projection profile, the peaks indicates text lines and valleys are inter line gaps. The Savitzky-Golay FIR smoothing filter with a second order polynomial and frame size of $f = 69$ is performed to smooth the projection profile result.

E. Text Line Extraction

To segmenting text lines, the maximums in projection profile are detected. Then, the image is split in the minimum point of the projection profile between each pair of maximum (Fig. 2.e). The original image is used here to consider all the CCs, as some small CCs were initially removed in the pre-processing step. Some long CCs may cut incorrectly in this process that need to be detect and then handled properly.

F. Postprocessing

There is a possibility that some CCs, especially the CCs with higher height, are incorrectly split into two parts and assigned to more than one text line (Fig. 2.e). Therefore, a post processing is performed to compensate, such mis-segmentation results, and improve the performance of the proposed text line segmentation method. In order to assign an optimal label to a multi-labelled CC, the label which belongs to the part with the size greater than $\%a_{th}$ of the CC's area is allotted to the multi-labelled CC.

Moreover, some small CCs eliminated initially from the original document image may provide erroneous text line extraction results. These are usually small CCs, less than three pixels and appear in a long horizontal row with a low density. Therefore, two criteria are considered for comparing the area and density of the determined text lines. The normalized area and density of recognized text lines are measured and compared to a threshold. The normalized area is defined as the area of text line CCs, divided by mean area of all the CCs of the image. The pixel density of a text line is defined as its normalized area divided by its major axis length. To do this, the normalized area and density of each text line are compared to a_{thtl} and dns_{tl} , respectively. A text line with an area or density smaller than the thresholds is merged to its upper adjacent text line. The final result obtained employing the proposed text line segmentation on the document image is demonstrated in Fig. 2.f.

III. EXPERIMENTAL RESULTS AND DISCUSSION

A. Dataset and Evaluation Metrics

To evaluate the proposed text line extraction method a benchmark dataset composed of 350 handwritten document images was considered for experimentation [13]. The dataset contains 200 document images in train set and 150 document images in test set. These texts were written in three languages; English, Greek and Bangla. None of the document images contains any non-text components, whereas touching/overlapping components

can frequently be seen in the images. Furthermore, a dataset of Kannada handwritten documents was considered in our experiments. The Kannada dataset contains 228 handwritten document images [15] written by 57 individuals.

Evaluation metrics used in the literature [13] were considered to compute the accuracy of the proposed text line extraction method. It involves three metrics; *Detection Rate (DR)*, *Recognition Accuracy (RA)* and *Performance Metric (FM)*. A *Match Score* which varies between 0 and 1 was also used to specify whether a text line is recognized correctly or not. In our evaluation process, the *Match Score* was set to 0.95.

B. Implementation and Parameter Setting

The proposed text line segmentation method was implemented using Matlab. In the proposed method, different parameters are used that needs to be tuned during the training phase. To set the parameters used in the proposed method a set of experiments were performed on the train dataset, considering different values for each parameter. To tune a parameter, such as a , different values were considered for a to obtain the best result, where the rest of the parameters were kept constant with some reasonable values according to them. To set the second parameter, the value of the first parameter was fixed with the value obtained in the previous step and in the same way as before the rest of the parameters were kept constant. The best values for the parameters obtained based on these experiments are demonstrated in Fig. 5. The segmentation results based on the FM metric obtained from different parameters using 200 document images of the training dataset are demonstrated in Fig. 5.

From the results shown in Fig. 5(a) it is clear that a value between 20 and 100 for the parameter a resulted in stable text line segmentation results. As shown in Fig. 5(b), the proposed method performed well when the parameter p were considered a value greater than 0.7. Parameter f was set to 69 based on the experimental results (Fig. 5(d)). The proposed method provided the best results when the parameter k was set to 2 or 3 (Fig. 5(c)). Furthermore, the proposed method is not sensitive to the other parameters and provided satisfactory text line segmentation results as demonstrated in Fig. (e-h).

C. Experimental Results

In testing experiments, the parameters were fix with the values, which provide the best results on training dataset. The proposed method was first tested on 150 test document images used in [13]. Among 2,649 text lines in the test dataset, 2,553 text lines were correctly segmented. As results, text line segmentation of 96.37%, 96.26% and 96.32% for DR , RA and FM were obtained on the ICDAR 2013 test dataset [13], respectively. Some of the results obtained from the proposed text line segmentation method are also shown in Fig. 6.

The experimental results on Kannada dataset indicate that out of 4,851 Kannada text lines, 4,704 text lines were correctly segmented. As results, DR , RA , and FM obtained for Kannada dataset are 96.96%, 97.08%, and 97.02%, respectively.

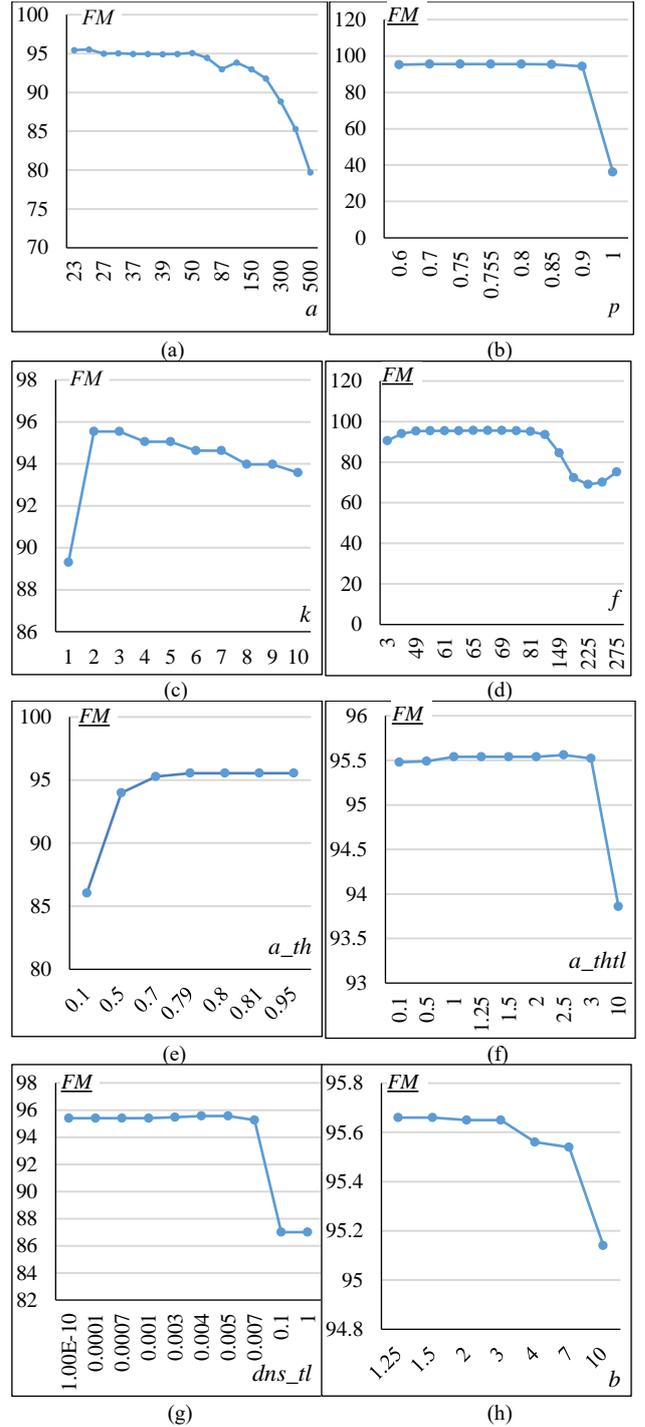


Fig. 5. Text line segmentation results (FM) obtained from the experiments on the train dataset based on different values for the parameters.

It is also worth noting that the line segmentation process in our proposed method takes on an average less than 1.7 seconds computing time when a personal computer with a 2.4 GHz Intel processor and 4 GB RAM were used for the experimentation. The low computational time in the proposed method is because i) for each two CCs, just one distance measurement is computed, and ii) the nearest neighbour is found among a small subset of all CCs distances instead of all the distances.



(a) (b) (c)
 Fig. 6. Sample results of the tested document images written in (a) English, (b) Bangla, and (c) Greek.

TABLE I. COMPARISON OF THE RESULTS OBTAINED FROM THE PROPOSED METHOD WITH SOME OF THE RESULTS REPORTED IN [13]

Method \ Metric	<i>M</i>	<i>o2o</i>	<i>DR</i> (%)	<i>RA</i> (%)	<i>FM</i> (%)
MSHK	2696	2428	91.66	90.06	90.85
QATAR-a	2626	2404	90.75	91.55	91.15
CVC	2715	2418	91.28	89.06	90.16
NCSR (SoA)	2646	2447	92.37	92.48	92.43
ILSP (SoA)	2685	2546	96.11	94.82	95.46
Proposed Method	2652	2553	96.37	96.26	96.32

D. Comparative Analysis

The text line segmentation method proposed in this research work is compared with some state-of-the-art methods. Table I provides a comparative analysis of the results obtained from the test data used in [13]. From the results provided in Table I it is evident that the proposed method is superior to those mentioned methods in terms of all the three evaluation metrics.

IV. CONCLUSION AND FUTURE WORK

In this paper, a new method for text line extraction in handwritten document images is presented. The method is based on the grouping and projection profile analysis. By introducing a new distance measure for *CC* grouping and an adaptive projection profile, a comparable performance with a low computation is achieved. The proposed text line extraction method may fail to accurately extract text lines from a document image when determined separator line, especially in cases with very low inter line gap, is not the best-fitted on intended baseline and splits a single text line into two parts.

For the future work, we plan to investigate on such a problem proposing a more efficient method to assign unusual connected components (very small and large) to the text line that they actually belong.

REFERENCES

[1] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, "Handwritten document image segmentation into text lines and words," *Pattern Recognit.*, 43(1), 2010, pp. 369–377.
 [2] X. Zhang and C. L. Tan, "Text Line Segmentation for Handwritten Documents Using Constrained Seam Carving," *In proceedings of the ICFHR*, 2014, pp. 98-103.

[3] A. Alaei, U. Pal, P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," *Pattern Recognit.*, 44(4), Apr. 2011, pp. 917–928.
 [4] W. Swaileh, K. A. Mohand and T. Paquet, "Multi-script iterative steerable directional filtering for handwritten text line extraction," *In proceedings of the ICDAR*, 2015, pp. 1241-1245.
 [5] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line detection in handwritten documents," *Pattern Recognit.*, 41(12), Dec. 2008, pp. 3758–3772.
 [6] F. Yin, and Ch. L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," *Pattern Recognit.*, 42(12), Dec. 2009, pp. 3146–3157.
 [7] Y. Boulid, A. Souhar and M. Y. Elkettani, "Arabic handwritten text line extraction using connected component analysis from a multi agent perspective," *In proceedings of the ISDA*, 2015, pp. 80-87.
 [8] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten arabic text lines," *In proceedings of the ICDAR*, 2009, pp. 176–180.
 [9] Q. N. Vo and G. Lee, "Dense prediction for text line segmentation in handwritten document images," *In proceedings of the ICIP*, 2016, pp. 3264-3268.
 [10] R. Manmatha, J.L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, *IEEE Trans. Pattern Anal. Mach. Intell.* 27(8), 2005 1212–1225.
 [11] M. Arivazhagan, H. Srinivasan, S.N. Srihari, A statistical approach to handwritten line segmentation, in: Document Recognition and Retrieval XIV, In proceedings of the SPIE, 2007, pp. 6500T-1-11.
 [12] A. Savitzky, M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures", *Anal. Chem.*, 36, pp. 1627-1639, 1964.
 [13] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," In proceedings of the ICDAR, 2013, pp. 1402-1406.
 [14] G. Seni, E. Cohen, "External word segmentation of off-line handwritten text lines," *Pattern Recognition*, 27(1),1994, pp. 41–52.
 [15] A. Alaei, U. Pal, and P. Nagabhushan, Dataset and ground truth for handwritten text in four different scripts, *Int. J. Patt. Recogn. Artif. Intell.*, (2012), pp. 26.
 [16] N.P. Carter, "Automatic recognition of printed music in the context of electronic publishing," Blostein D, BairdH(eds), *Structured Document Image Analysis*. Springer, Heidelberg, 1992, pp. 405-434.