IEICE TRANS. INF. & SYST., VOL.E94–D, NO.4 APRIL 2011

PAPER

Adaptive Script-Independent Text Line Extraction

Majid ZIARATBAN[†a)], *Student Member and* **Karim FAEZ**[†], *Member*

SUMMARY In this paper, an adaptive block-based text line extraction algorithm is proposed. Three global and two local parameters are defined to adapt the method to various handwritings in different languages. A document image is segmented into several overlapping blocks. The skew of each block is estimated. Text block is de-skewed by using the estimated skew angle. Text regions are detected in the de-skewed text block. A number of data points are extracted from the detected text regions in each block. These data points are used to estimate the paths of text lines. By thinning the background of the image including text line paths, text line boundaries or separators are estimated. Furthermore, an algorithm is proposed to assign to the extracted text lines the connected components which have intersections with the estimated separators. Extensive experiments on different standard datasets in various languages demonstrate that the proposed algorithm outperforms previous methods.

key words: *script-independent, block-based, text line extraction, connected component assignment*

1. Introduction

To recognize texts in a document image, first the text lines must be extracted. The results of the text line extraction stage affect the overall performance of the optical character recognition (OCR) tasks. Different sizes of characters and words, variations in the distances between consecutive text lines, touching text lines and non-uniform skew angles even in the same line (*multi-skewed* text line) make the text line extraction difficult. The frequently used approaches are based on horizontal projection profiles (HPP) [1]–[3], piece-wise HPP [4]–[11], Hough transform [12]–[15], and blurring [16], [17]. Algorithms have been designed and proposed for different languages such as English [2], [5], [14], [16], [18], [19], Hindi [7], [10], [20]–[22], Japanese [23]–[26], Chinese [19], [27], and Arabic [3], [8], [9]. The approaches in which the whole document image is processed globally cannot extract these complicated text lines with an acceptable accuracy. Two main categories of the challenge exist in the handwritten text line segmentation problem. Different skews of text lines in the same document image even in the same text line (curvilinear text lines) is the first category of the challenges. The second one is the different distances between text lines. Variations of between-text line gaps may occur along the two consecutive text lines. Overlapping and touching text lines are also in the second category.

Manuscript received September 17, 2010.
Manuscript revised November 25, 2010.

[†]The authors are with Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran.
a) E-mail: m_ziaratban@aut.ac.ir

DOI: 10.1587/transinf.E94.D.866

HPP-based method [1], X-Y cut [1] and Docstrum [28] are the approaches which have been designed for machine-printed text lines and are not suitable for complex handwritten text line segmentation. These approaches cannot extract overlapping text lines, text lines with different skews, and curved text lines. Hough transform-based methods can segment text lines with different skews in the same page, but are not very effective for multi-skewed and curved text lines [14]. In the piece-wise HPP-based approaches, a text image is divided into a number of vertical strips with the same width. In each strip, by using the HPP technique, the text regions and the gaps between these regions are detected. The estimated separators in each strip must be combined with the ones in the consecutive strips. Piece-wise HPP-based methods can handle multi-skewed text lines, but are not very accurate in segmentation of overlapping or touching text lines. Setting a proper value for the strips width is another problem in these methods.

In the case of touching text lines, the blurring-based approaches such as probability density function (PDF)-based [16], Mumford-Shah model [18], and shredding [17] methods have been presented better results than others. These methods will blur text images with a horizontal filter to obtain the horizontal distribution of text line pixels. In these approaches, overlapping and touching text lines with very small skews are separated. The drawback of these blurring-based methods is that they suppose the text lines to be horizontal with very small skews. In other words, if the skews of text lines are great, by applying the horizontal filter to a document image, text lines are interlaced and could not be segmented. MST (minimal spanning tree) clustering-based algorithm [27] can extract multi-skewed text lines, but it requires large gaps between text lines.

In our approach, to overcome multi-skewed and touching text line extraction problems, each text page is divided into a number of overlapping blocks. Text regions in each block are detected and some information about these regions is extracted. Text lines are extracted by using the information about the detected text regions. The organization of the rest of the paper is as follows: A brief overview of the recent works is given in Sect. 2. The proposed text line segmentation and connected-component (CC) assignment algorithms are described in detail in Sects. 3 and 4, respectively. Experimental results are presented in Sect. 5. Finally, concluding remarks are given in Sect. 6.

2. Related Works

Basu et al. [20] proposed an approach based on the hypothetical water flow technique. This algorithm extracts curved text lines, but due to the nature of the algorithm, sufficient gap between text lines is required [20]. Moreover, for different document images, the value of two parameters of the algorithm, θ (the flow angle) and k (radius of the structure element) must be adjusted manually [20]. Experiments were done on a set of 1191 handwritten Bengali and English text lines. The success rate was 90.34% and 91.44% for handwritten Bengali and English text lines, respectively.

A PDF-based algorithm was proposed by Li et al. [16]. They achieved 98%, 97%, and 98% pixel-level hit rate (PLHR) for Chinese, Hindi, and Korean handwritten text line extraction, respectively. They assumed that the skews of text lines are less than 10 degrees. The most errors occurred in touching text lines [16]. Du et al. in [18] found that different initial conditions for the algorithm of [16] will produce different segmentation results. Du et al. [18] proposed an algorithm by using Mumford-Shah model and tested it on the same dataset as Li et al. used in [16]. They reached to 98%, 98%, and 96% PLHR for Chinese, Hindi, and Korean handwritten texts, respectively. They assumed that text lines are horizontal and used horizontal structure elements for connecting separated text lines and also segmenting the touching lines in the post-processing stage [18]. Hence, this method works effectively only on the text lines with very small skews.

Yin and Liu [27] proposed a handwritten Chinese text line segmentation algorithm by using minimal spanning tree (MST) clustering with distance metric learning. In their algorithm, the connected components of a document image were grouped into a tree structure to find the text lines based on the clusters. In experiments on a dataset of 803 handwritten Chinese pages, they achieved a correct rate of 98.02% of text line detection. This algorithm was very sensitive to the between-word and between-line gaps. If the gaps between words were large or the distances between text lines were not large enough, the clustering algorithm produced incorrect results.

Recently, Papavassiliou et al. [5] proposed an algorithm based on piece-wise HPP technique. They over segmented strips and reclassify the detected text regions by applying the Viterbi algorithm. Experimental results on the ICDAR07 dataset [29] showed that 98.46% of text lines were detected correctly. They carried out the experiments with various values for strip widths and found that the strip width equal to 5% of the document image width produces the best results. It seems that this way of value setting for strip width is not very accurate, because of different sizes of document widths for the same writer and also different average size of words and gaps between text lines in the same size of images. It is better that the value of strip width be set based on the features of the handwritten text instead of the document image width.

3. Proposed Method

In the proposed algorithm, to extract text lines with various skews in the same text page and multi-skewed text line, a text page is divided into a number of blocks. The skew angle and between-line gap for each block are estimated separately. Using the computed block skew angle, text regions in each block are detected. Repeating this procedure for all blocks of the text page, a number of data points are obtained. Each data point includes information about the distance between text lines, skew angle, and x-y coordinate of the detected text regions. The data points are used to estimate the path of each text line. To have more accurate path estimation, it is better the blocks have some overlaps. In our experiments, the adjacent blocks have 80% overlaps. The flowchart of the proposed text line extraction algorithm is given in Fig. 1. More discussions about different parts of our algorithm are presented in the following sections.

3.1 Global Parameters Estimation

In the proposed method, to have better results, particularly in the case of multi-skewed text lines, the text image is divided into a number of overlapping blocks. These blocks should contain some parts of text lines which have near-straight paths and uniform skews. In very small blocks, there is not enough foreground pixels to have an accurate skew estimation. On the other hand, in a very big block, variations of writing paths may be high and the text line segments may have non-uniform skews. To adapt our method for various kinds of handwritings in different languages and also for different scanning resolutions, the required block size should be estimated for each text page with respect to the text features. Experiments on different datasets show that the text line segments including about 8 to 13 connected-components (CC) usually have near-straight paths and uniform skews. In our study, the block width is set to be equal to the width of 10 CCs. Therefore, the effective width of the CCs in the text image is required to be

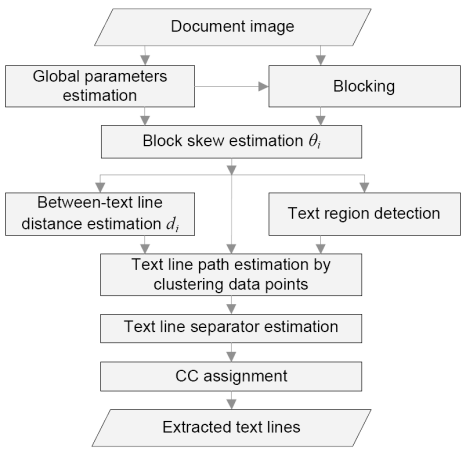

Fig. 1 Flowchart of the proposed text line extraction algorithm.

estimated. In our study, three global parameters are defined and computed as follows:

- Pen width (w_p): To calculate the pen width, the run lengths of foreground pixels in columns of the document image are computed. The most frequent run length is considered as the pen width. To prevent the influence of noise on the other parameters estimation stage, very small CCs for which the summation of height and width values is smaller than $2w_p$ pixels are considered as noise and are not allowed entering in the next two parameters estimation stage.
- Effective CC width (w_{cc}): To compute w_{cc}, the widths of all CCs in the document image are obtained and stored in $\boldsymbol{W}$. The value of w_{cc} is calculated as follows:

$$w_{cc} = \sum_{k \in K} kHST_w(k) \Big(\sum_{k \in K} HST_w(k) \Big)^{-1}, \tag{1}$$

$$K = \left\{ k | HST_w(k) > \frac{1}{4} max\{HST_w\} \right\}, \tag{2}$$

  where HST_w is the histogram of $\boldsymbol{W}$ and includes the distribution of the CCs versus their widths. $max\{HST_w\}$ is the maximum number of the CCs which have the same width. Equation (2) eliminates the effects of the CCs for which their widths are very different from the width of the majority of CCs.
- Effective CC height (h_{cc}): This parameter is estimated in the similar way as the w_{cc}.

According to the calculated parameters, the block size ($p \times q$) is set to $10w_{cc} \times 10w_{cc}$ ($p = q = 10w_{cc}$).

3.2 Block Skew Estimation

Several approaches have been developed to estimate the overall skew angles of handwritten texts. These approaches require long enough text lines to estimate the skew angle. Thus, they cannot be used to estimate the skew of the text blocks. In this paper, a skew estimation method for the text blocks is proposed. The basic idea behind the proposed method is that if a text block is blurred in various directions, the blurred block in the direction equal to the block skew angle has the greatest contrast. Text blurring is carried out by filtering the image with anisotropic 2D Gaussian kernels [16]. The window size of the basic kernel (corresponding to $\theta = 0$) is considered as $h_{cc} \times 15w_{cc}$. The vertical and horizontal standard deviations of the basic filter are set to $\sigma_y = \frac{1}{5}h_{cc}$ and $\sigma_x = 5w_{cc}$, respectively. The filter bank consists of the rotated versions of the basic kernel in several directions. To increase the speed and to have robust and accurate estimations, only the blurred image corresponding to the text blocks in which the number of foreground pixels are greater than 1% of total number of the block pixels are processed. In our proposed method, the contrast of a blurred block is calculated based on the directional gradient of the blurred block. In this way, the document image $I(x, y)$ is first blurred in various directions. $F_j(x, y)$ is the blurred image in the j-th direction. Then, F_j is divided into overlapping blocks of size $p \times q$. $F_{j,i}(x', y')$ is the i-th block of $F_j(x, y)$. x' and y' are the horizontal and vertical coordinates in each block, respectively. Directional gradient of $F_{j,i}$ is computed as follows:

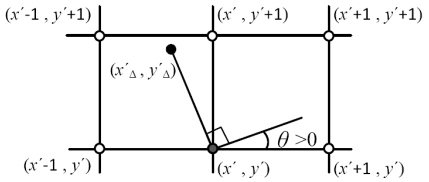

Fig. 2 Coordinate of the pixel which is used in the directional gradient calculation.

$$G_{j,i}(x', y') = F_{j,i}(x', y') - F_{j,i}(x'_\Delta, y'_\Delta), \tag{3}$$

where $G_{j,i}$ is the directional gradient of $F_{j,i}$. As shown in Fig. 2, (x'_Δ, y'_Δ) is the coordinate of the pixel which has one pixel distance from (x', y') with respect to θ and θ is the direction of the blurring. As this figure shows, x'_Δ and y'_Δ are not generally integer values. Therefore, the value of $F_{j,i}(x'_\Delta, y'_\Delta)$ is computed based on the interpolation of the values of the adjacent pixels as follows:

$$F_{j,i}(x'_\Delta, y'_\Delta) = \begin{cases} (1 - \sin\theta)(1 - \cos\theta)F_{j,i}(x', y') \\ + (1 - \sin\theta)(\cos\theta)F_{j,i}(x', y' + 1) \\ + (\sin\theta)(1 - \cos\theta)F_{j,i}(x' - 1, y') \quad \text{if } 0 \le \theta \le 90 \\ + (\sin\theta)(\cos\theta)F_{j,i}(x' - 1, y' + 1) \\ \\ (1 + \sin\theta)(1 - \cos\theta)F_{j,i}(x', y') \\ + (1 + \sin\theta)(\cos\theta)F_{j,i}(x', y' + 1) \quad \text{if } -90 \le \theta \le 0 \\ + (-\sin\theta)(1 - \cos\theta)F_{j,i}(x' + 1, y') \\ + (-\sin\theta)(\cos\theta)F_{j,i}(x' + 1, y' + 1) \end{cases} \tag{4}$$

By inserting $F_{j,i}(x'_\Delta, y'_\Delta)$ from (4) into (3), the directional gradient image is calculated. The summation of absolute value of the gradient image is considered as the directional contrast of $F_{j,i}$:

$$Cont_{j,i} = \sum_{x'=1}^{q} \sum_{y'=1}^{p} abs(G_{j,i}(x', y')). \tag{5}$$

The skew angle of the i-th block θ_i is the J-th element of Θ and Θ is the set of the blurring directions.

$$J = \arg\max_j \{Cont_{j,i}\}. \tag{6}$$

Directional blurred blocks and the absolute of the directional gradient blocks of a sample text block are illustrated in the first and second rows of Fig. 3, respectively. As this figure shows, the blurred block in the direction equal to the text skew has the greatest contrast. Figure 4 shows the directional contrast values for the sample text block versus various directions. The maximum value in this chart is obtained in $\theta = -13$ degrees which is equal to the skew of the

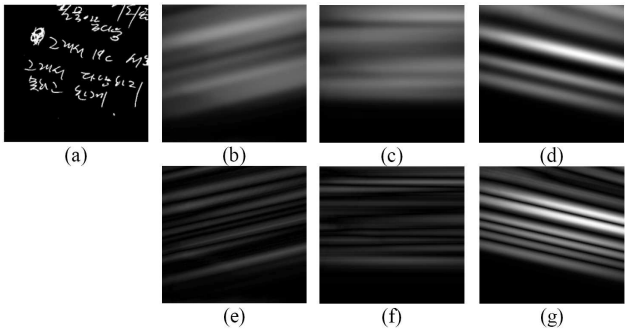

Fig. 3 (a) A sample text block from the UMD dataset [18], (b), (c), and (d) blurred blocks in directions of 13, 0 and −13 degrees, respectively. (e), (f), and (g) are the absolute of directional gradient of (b), (c), and (d), respectively.

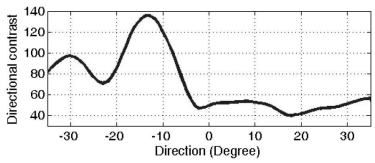

Fig. 4 Directional contrast of the blurred blocks in various directions.

block. By defining and using directional contrast parameter, the proposed block skew estimation algorithm can estimate skews of text line segments which are rotated with arbitrary angles even the near-vertical text line segments. But the absolute skew values of text line segments in the datasets used in our experiments are not greater than 35 degrees. Hence, we consider the directions between −35 and 35 degrees.

3.3 Estimation of the Distance between Text Lines for Each Block

As shown in the flowchart (Fig. 1), after the block skew estimation, another parameter of a text block is estimated. This parameter is the between-text line distance which is expressed as d_i for the i-th text block, $I_i(x', y')$. The text block is first de-skewed by using the estimated skew angle θ_i. The value of d_i is calculated based on the distances between the consecutive peaks of the HPP of the de-skewed text block. To have smoother HPP for better peak detection, an averaging mask with length equal to $\frac{1}{2}h_{cc}$ is applied to the HPP. The between-line distance d_i for the i-th block $I_i(x, y)$ is calculated as follows:

$$d_i = \begin{cases} \frac{1}{n_i - 1} \sum_{k=1}^{n_i - 1} d_{i,k} & \text{if } n_i \geq 2 \\ 5w_{cc} & \text{if } n_i = 1 \end{cases} \quad (7)$$

where n_i is the number of peaks in the smoothed HPP of the de-skewed text block. $d_{i,k}$ is the distance between the k-th and $(k + 1)$-th peaks of the smoothed HPP. The de-skewed text block for the example in Fig. 3 (a) is depicted in Fig. 5 (a). Four peaks are detected in the smoothed HPP of the de-skewed block shown in Fig. 5 (c). The value of d_i for this block is the average value of $d_{i,1}$, $d_{i,2}$, and $d_{i,3}$.

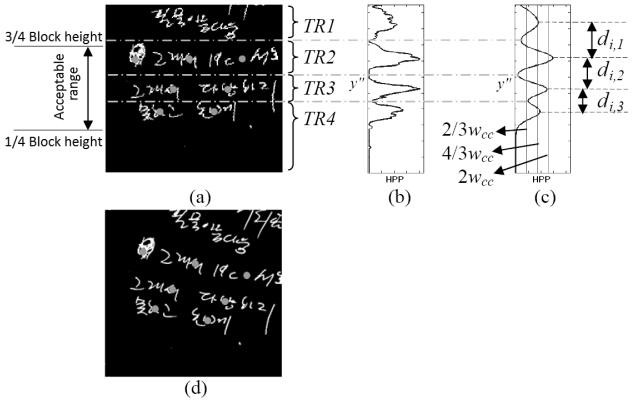

Fig. 5 (a) de-skewed text block, (b) HPP of the de-skewed text block, (c) smoothed HPP of the de-skewed text block, (d) extracted representatives in the text block (red dot-markers).

Table 1 The numbers and coordinates of representatives with respect to m_k which is the k-th peak value of the smoothed HPP.

	Numbers of representatives	Representatives coordinates
$m_k < \frac{2}{3}w_{cc}$	0	-
$\frac{2}{3}w_{cc} \leq m_k < \frac{4}{3}w_{cc}$	1	$(x''_{i,k}, y''_{i,k})$
$\frac{4}{3}w_{cc} \leq m_k < 2w_{cc}$	2	$(x''_{i,k} - w_{cc}, y''_{i,k})$
		$(x''_{i,k} + w_{cc}, y''_{i,k})$
$2w_{cc} \leq m_k$	3	$(x''_{i,k} - 2w_{cc}, y''_{i,k})$
		$(x''_{i,k}, y''_{i,k})$
		$(x''_{i,k} + 2w_{cc}, y''_{i,k})$

3.4 Text Region Detection

As shown in Fig. 5 (a), by using the valleys of the smoothed HPP of the de-skewed text block, different text regions can be simply separated. A text region is a horizontal strip of the de-skewed block which is vertically bounded between two consecutive valleys. Text regions in the most top and most bottom of a de-skewed text block may have slight remaining skews. Therefore, we used only the text regions which are located between $\frac{1}{4}$ and $\frac{3}{4}$ of the height of the de-skewed blocks. The most top detected text region in Fig. 5 (a), *TR1*, is not in this acceptable range and is rejected. For each detected text region, a number of representatives are extracted proportional to the value of the corresponding peak in the smoothed HPP. If all pixels in a row of a text block are foreground pixels, the value of the HPP corresponding to this row will be equal to the block width. But for handwritten text lines, this value does not usually exceed from $\frac{1}{3}$ of the block width. Since the width of the text blocks in our method is equal to $10w_{cc}$ pixels, peak values of HPPs do not usually exceed from $\frac{10}{3}w_{cc}$. We consider three thresholds to extract proper numbers of representatives for text regions. The thresholds are $\frac{2}{3}w_{cc}$, $\frac{4}{3}w_{cc}$, and $2w_{cc}$ (Fig. 5 (c)). Table 1 shows the numbers and coordinates of representatives with respect to the peak values of the smoothed HPP. In this table, m_k is the k-th peak value in the smoothed HPP.

$(x''_{i,k}, y''_{i,k})$ is the coordinate of the gravity center of the k-th detected text region in the i-th de-skewed text block.

The coordinate of the r-th representative in the text block $I_i(x', y')$ is computed as follows:

$$\begin{bmatrix} x'_{i,r} \\ y'_{i,r} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} x''_{i,r} - \frac{q}{2} \\ y''_{i,r} - \frac{p}{2} \end{bmatrix} + \begin{bmatrix} \frac{q}{2} \\ \frac{p}{2} \end{bmatrix}, \tag{8}$$

where $(\frac{q}{2}, \frac{p}{2})$ is the coordinate of the center of the blocks. θ_i is the estimated skew of the i-th block. Red dot-markers in Figs. 5 (a) and 5 (d) show the representatives in the de-skewed text block and their new coordinates in the text block, respectively. Let (x_i^0, y_i^0) be the coordinate of the bottom-left corner of the i-th text block in the text image $I(x, y)$. The coordinates of the representatives of the i-th text block in the text image $I(x, y)$ are obtained as follows:

$$\begin{bmatrix} x_{i,r} \\ y_{i,r} \end{bmatrix} = \begin{bmatrix} x'_{i,r} \\ y'_{i,r} \end{bmatrix} + \begin{bmatrix} x_i^0 \\ y_i^0 \end{bmatrix}. \tag{9}$$

3.5 Text Line Path Construction

After detecting the representatives in all blocks, a number of data points are obtained. Let $DP_c = \{x_c^*, y_c^*, \theta_c^*, d_c^*\}$ be the c-th data point corresponding to the r-th representative of the i-th text block. Then, x_c^*, y_c^*, θ_c^*, and d_c^* are equal to $x_{i,r}$, $y_{i,r}$, θ_i, and d_i, respectively. The writing paths of text lines are constructed based on these data points. Before the path construction, a preprocessing is required to eliminate unsuitable data points. In the block skew estimation stage, if a block consists of two or more text line segments with different skews, the estimated skew may not be accurate and correct for all text line parts. Thus, some unreliable representatives may be extracted from these text blocks. An example of these cases is illustrated in Fig. 6. The yellow rectangle in Fig. 6 (a) shows a block including text line segments with different skews. Extracted data points are shown with their directions in Fig. 6 (b). It can be seen that there are

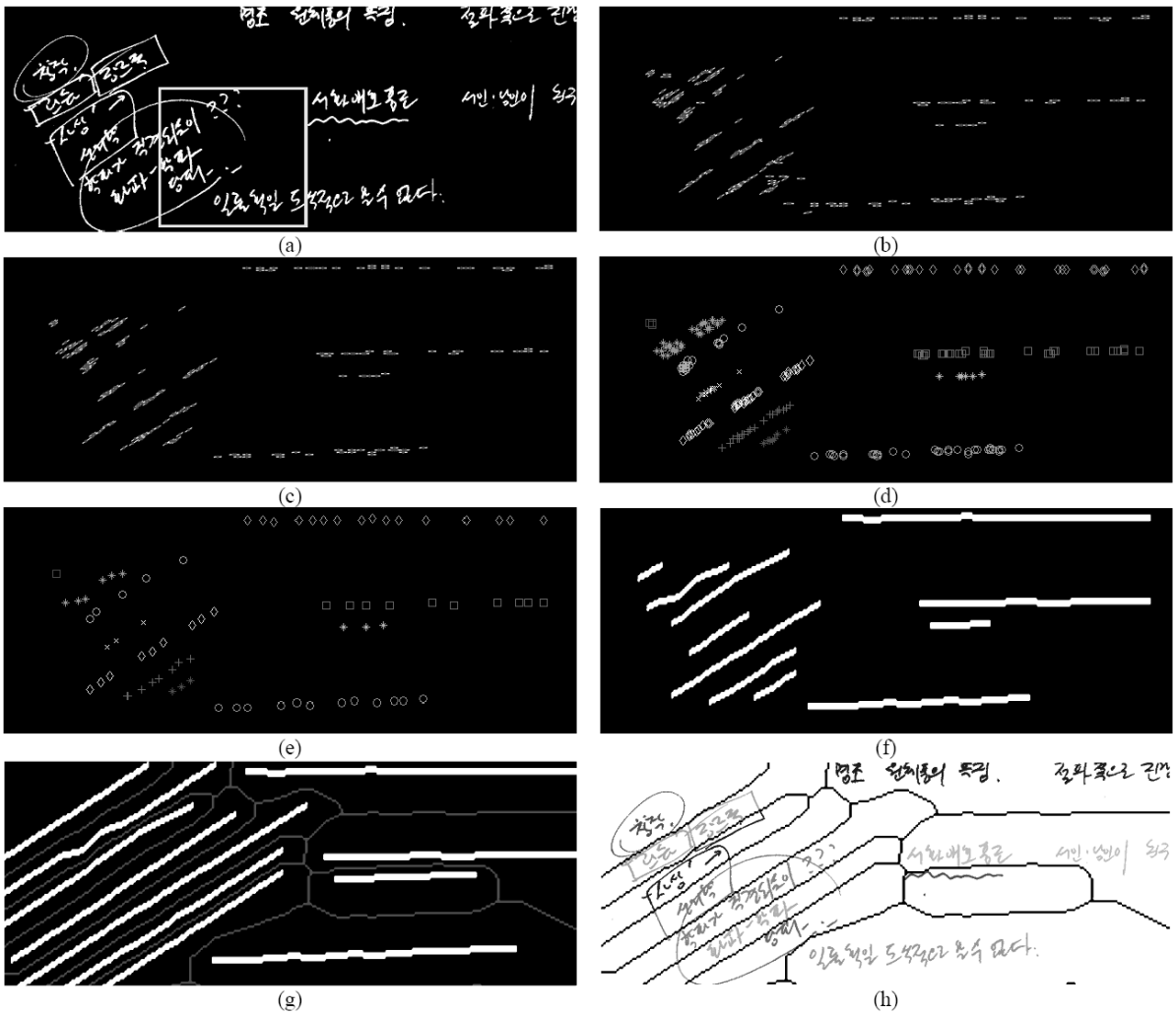

Fig. 6 (a) A part of a complex Korean handwritten text in the UMD dataset [16], (b) extracted data points, (c) reliable data points after removing unsuitable data points, (d) clustered data points, (e) average clustered data points, (f) constructed paths, (g) extended paths and the separators, (h) extracted text lines.

some unsuitable data points with incorrect skews. To extract suitable data points, a neighborhood of size $6h_{cc} \times 6w_{cc}$ is considered around each data point. The differences between the angles of all data points and the center data point are calculated. In the neighborhood, the number of data points, for which the angle difference is lower than a predefined value (10 degrees in our algorithm), is computed as n_d. Let n_a be the total number of data points in the neighborhood. A data point is considered as the reliable data point if $\frac{n_d}{n_a} > 0.5$; otherwise is eliminated. By applying the above process to the data points of our example, the reliable data points are detected and depicted in Fig. 6 (c). Now, we can construct the paths of text lines by using these reliable data points. The algorithm of the text line path construction is a clustering algorithm as follows:

Step 1: The most left data point (The data point whose x-coordinate is the minimum) is considered as the first member (new member) of the first path.

Step 2: If any new member (the first member of the new created cluster or the data points which are added to the current cluster in the Step 3) belongs to the current path, then, for each new member go to Step 3; else, go to Step 4.

Step 3: Check the membership conditions between all un-clustered data points and the new members of the current path. The un-clustered data points which satisfy the membership conditions are considered as new members of the corresponding path. Go to Step 2.

Step 4: If there are any un-clustered data points, then, consider the most left un-clustered data point as the first member of the new path and go to Step 3; else, go to Step 5.

Step 5: For each constructed path determine the data points which are at two ends of the path. If two ends of two different paths satisfy merging conditions, then, these two paths are merged.

Three membership conditions in ***Step 3*** are as follows:

$$1: \ abs(\theta_c^* - \theta_u^*) < 10, \tag{10}$$

$$2: \ \Delta y''_{c,u} \leq \frac{d_c^*}{3}, \tag{11}$$

$$3: \ \Delta x''_{c,u} \leq 3w_{cc}, \tag{12}$$

where θ_c^* and θ_u^* are the direction of the clustered data point DP_c and un-clustered data point DP_u, respectively. Figure 7 illustrates $\Delta y''$ and $\Delta x''$ for two sample data points DP_a and DP_b. To calculate $\Delta y''_{c,u}$ and $\Delta x''_{c,u}$, the x-y axes is first rotated with the angle of θ_c^* degrees. In the rotated axes (x''–y'' axes), the distance of the un-clustered data point (DP_u) from the x''-axis and y''-axis are computed as $\Delta y''_{c,u}$ and $\Delta x''_{c,u}$, respectively.

$$\Delta y''_{c,u} = abs(-(x_u^* - x_c^*)\sin\theta_c^* + (y_u^* - y_c^*)\cos\theta_c^*), \tag{13}$$

$$\Delta x''_{c,u} = abs((x_u^* - x_c^*)\cos\theta_c^* + (y_u^* - y_c^*)\sin\theta_c^*). \tag{14}$$

The first condition (10) must be hold because the adjacent data points of the same text line must have approximately the same directions. If $\Delta y''_{c,u} = 0$, it means that DP_u is exactly in the same direction of the text region corresponding to DP_c. In the third membership condition (12), the maximum allowable distance is set to a small value ($3w_{cc}$) to prevent connecting the data points which are in the same direction but far from each other, because they may be members of different text lines. By applying the third condition, for text lines in which the gaps between words are high, some disconnected paths may be constructed instead of a single continuous path. These path parts are merged to each other in Step 5. Three merging conditions are defined as follows:

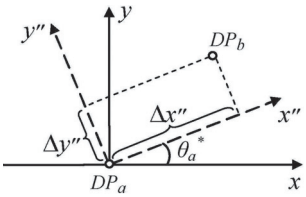

Fig. 7 Distances DP_b from the horizontal and vertical axes in the rotated coordinates with respect to the direction of DP_a.

$$1: \ abs(\theta_{e1}^* - \theta_{e2}^*) < 5, \tag{15}$$

$$2: \ \Delta y''_{e1,e2} \leq \frac{d_{e1}^*}{3}, \tag{16}$$

$$3: \ \Delta x''_{e1,e2} \leq 6w_{cc}, \tag{17}$$

where DP_{e1} and DP_{e2} are two data points at two ends of two different paths. In other words, the path parts, which are in the same direction and are not far from each other, are merged.

To achieve smoother paths, average coordinates of adjacent data points of the same cluster are calculated and used instead of the adjacent data points. In this way, data points of the same cluster are sorted from left to right and the first data point (the most left data point) is selected. $\Delta x''$ values between the first data point and other adjacent date points are calculated by using Eq. (14). The adjacent data points for which the value of $\Delta x''$ is lower than $0.5w_{cc}$ are chosen. Suppose that $(k-1)$ data points satisfy this condition. The average x-y coordinate of these k data points (the first data point and $(k-1)$ adjacent data points) are computed. This *average data point* is used instead of these k data points. In the next step, $(k+1)$-th data point is selected and the above procedure is performed. This procedure is continued until we reach to the last data point of the cluster. Clustered data points and average data points for an example text image are shown in Figs. 6 (d) and 6 (e), respectively.

3.6 Text Line Extraction

The path of each text line is obtained by connecting the consecutive average data points of the same cluster to each other. Each constructed path is extended at its two ends to be sure that the path covers the whole text line. The extension direction at each end of the path is set to the direction of the data point at the corresponding end. The maximum extension length is considered to be $5w_{cc}$ pixels at each end. The extension is allowed only if the extended path does not

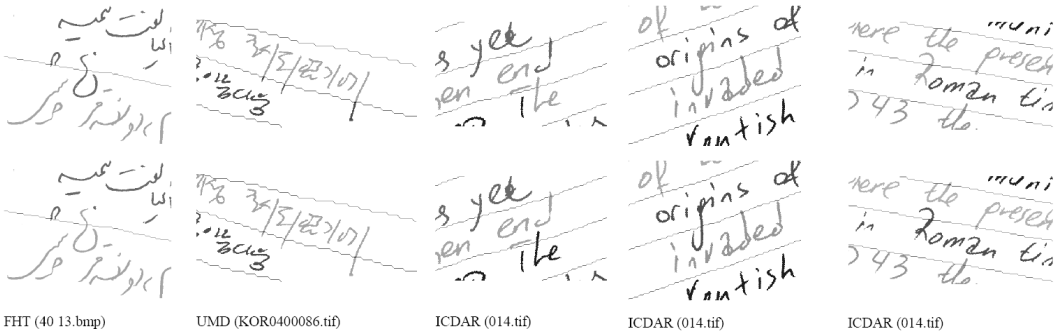

Fig. 8 CC assignment results for non-horizontal text lines by using (first row) the method in [5] and (second row) our proposed method.

touch any other paths. In other words, if the extended path touches other paths, the extension is not correct and will be cancelled. To obtain the separators or text line boundaries, the background of the image including the paths is thinned. Constructed paths, extended paths, and extracted text lines for a difficult Korean handwritten text are illustrated in Figs. 6 (f), 6 (g), and 6 (h), respectively. The text image in Fig. 6 (a) includes text lines with different skews and also touching text lines. As Fig. 6 (h) shows, the proposed method accurately extracts touching text lines as well as the text lines with large differences in their skews.

4. CC Assignment

After estimating the text line separators, there may be some intersections between the separators and handwritten CCs. In these cases, the whole CC may belong to one of the text lines which are separated by the estimated separator. Another possibility is that a part of the CC is belonged to one text line and the rest is a member of the other text line. Papavassiliou et al. [5] proposed an algorithm to assign CCs to text lines. They considered a neighborhood area around the intersection point. This area must include parts of both text lines (j-th and $(j+1)$-th text lines). They defined N^b_j and N^b_{j+1} as the number of foreground pixels of lines j and $(j+1)$ in the area, respectively. Similarly, N^a_j and N^a_{j+1} are defined as the number of foreground pixels assigned to lines j and $(j+1)$ which lie at the same horizontal line with the pixels of the CC. The ratios $r_k = N^a_k/N^b_k$ for $k = j$ and $(j+1)$ are considered as a decision measurement. In their approach, if both ratios are below a threshold (they experimentally set to 0.4), the CC is completely assigned to the text line having the greatest overlap with the CC.

If only one ratio is over the threshold, the CC is assigned to the text line with the highest ratio. If both ratios are over the threshold, the algorithm tries to find any junction point which lies near the separator between the j-th and $(j+1)$-th text lines. The distance of the junction point from the separator line must be less than half of the mean height of all CCs in the document image. If there are more junction points, the nearest one to the separator is selected. The CC in the junction point is broken into some parts. If there is no junction point, the CC is divided into two parts in the intersection point with the separator. Each new CC part is separately assigned to text lines by applying the aforementioned constraint. For more details, please refer to [15]. The main drawback of the CC assignment algorithm proposed in [15] is that the text lines are considered horizontal and only the horizontal overlap between the CC and text lines are computed. Therefore, this algorithm cannot make a correct decision for assigning CCs in the skewed text lines.

In the proposed CC assignment algorithm, the skews of text lines are considered. In more details, the neighborhood area is first de-skewed and then the horizontal overlap between the CC and the de-skewed text lines are computed. The skew of the neighborhood area around the intersection point is calculated based on the directions of the adjacent data points. We use average of angles of four nearest data points to the intersection point as the skew of the texts in the neighborhood area. Figure 8 shows some examples of non-horizontal text lines. In this figure, the first and second rows are the CC assignment results by applying the method of [15] and the proposed method, respectively. The samples in Figs. 8 (a) and (b) were written non-horizontally. The samples in Figs. 8 (c), (d), and (e) are parts of image 014.tif from the ICDAR dataset and are rotated artificially. It can be seen that in all examples, the proposed method demonstrated the best CC assignment results.

5. Experimental Results

5.1 Datasets

Four datasets were used to evaluate the proposed text line extraction method. The datasets are FHT [30], HIT-MW [31], UMD [16], and ICDAR [29]. The FHT includes 1129 Farsi handwritten forms and 7186 text lines written by 282 persons. The HIT-MW contains 853 Chinese handwritten forms including 8664 text lines. About 780 participants filled the forms. The UMD consists of 100 Chinese, 96 Hindi and 100 Korean forms. The ICDAR dataset includes 100 forms in English, French, German and Greek languages.

5.2 Evaluation Methodology

Three evaluation metrics were used in our experiments: *DR*, DR_2, and *PLHR*. Detection rate (*DR*) is the frequently used metric and is equal to the rate of the detected text lines. A result text line is considered as a detected text line if its corresponding *MatchScore* is greater than 0.95. *MatchScore* is the number of matches between the pixels of a result text line and the pixels of the corresponding ground-truth text line.

$$MatchScore(i, j) = T(G_i \cap R_j)(T(G_i \cup R_j))^{-1}, \quad (18)$$

where $T(s)$ counts the number of foreground pixels of s. G_i and R_j are the i-th ground-truth and j-th resulting text lines, respectively. The result text lines and ground-truth text lines must have one-to-one correspondence. Since the number of text lines in the result and ground-truth sets may be different, either a result or a ground-truth text line is allowed to be matched with a dummy line. The percentage of correctly detected text lines out of the ground-truth lines gives the correct detection rate (recall rate) [27]. Another version of the detection rate (DR_2) was defined and used in [16]. In this evaluation metric, a result text line is claimed to be correct if (19) and (20) are satisfied.

$$T(G_i \cap R_j)(T(G_i))^{-1} \geq 0.9, \quad (19)$$

$$T(G_i \cap R_j)(T(R_j))^{-1} \geq 0.9. \quad (20)$$

In other words, a result text line is considered as a correct detected line, if it and the corresponding ground-truth line share at least 90 percent of the pixels with respect to both of them [16]. By using *DR* and DR_2 metrics, performances are evaluated at the text line level. Furthermore, a pixel-level evaluation metric was defined in [16]. This metric was called *pixel-level hit rate* (*PLHR*) and is equal to the number of shared pixels between the best matched ground-truth and the resulting text lines divided by total number of foreground pixels in the ground-truth [16].

The comparative results over the FHT, HIT-MW, UMD, and ICDAR datasets are reported in Tables 2, 3, 4, and 5, respectively. The proposed method outperformed all other approaches over different datasets. To compare the different approaches, some text line segmentation results produced by various methods are shown in Fig. 9 to Fig. 14. Results of the Hough transform-based [12] and fuzzy RLSA [32] methods on a cursive English handwritten document with overlapping text lines are illustrated in Figs. 9 (a) and (b) [12], respectively. In Fig. 9 (a), the second text line has not been detected. In addition, the 6th and 7th text lines have not been separated. Also as shown in Fig. 9 (b), some parts of consecutive text lines have not been correctly segmented. The proposed method has accurately segmented these overlapping text lines. In Fig. 9 (c), esti-

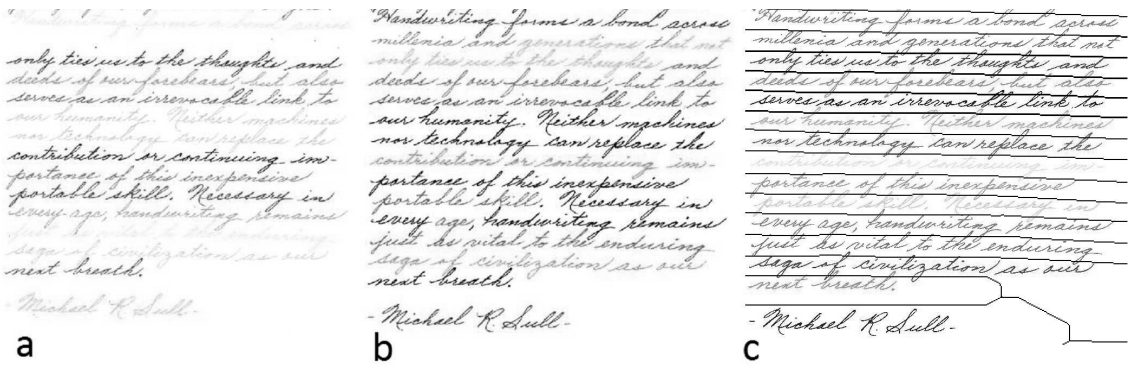

Fig. 9 Segmentation results on a cursive English handwritten document of the ICDAR dataset produced by (a) Hough method with post-processing [12], (b) Fuzzy RLSA method [32], and (c) the proposed method.

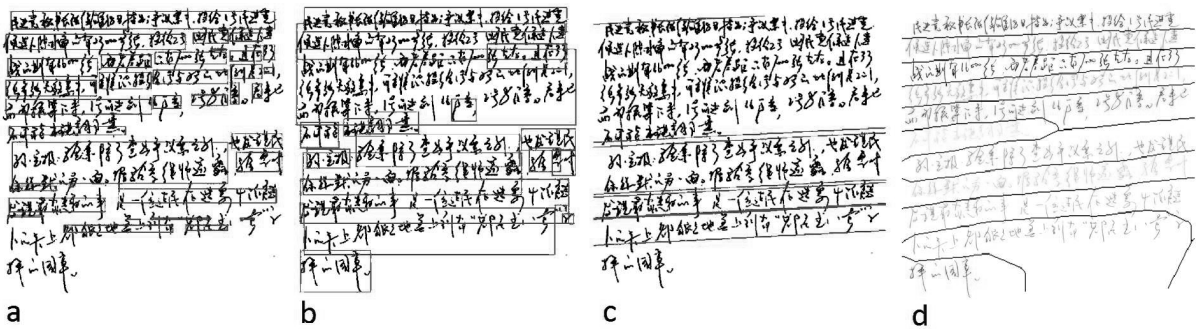

Fig. 10 Segmentation results on a Chinese document image of the HIT-MW dataset produced by (a) X-Y cut [1], (b) Docstrum [28], (c) stroke skew correction [33], and (d) the proposed method.

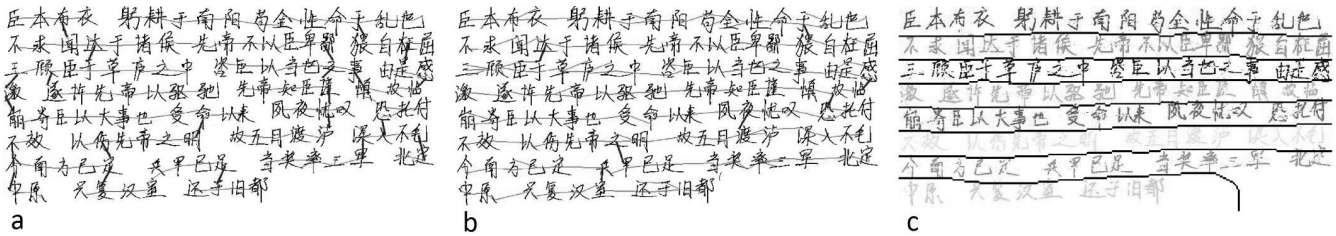

Fig. 11 The results of (a) the MST clustering with hand-crafted metric [34], (b) the MST clustering with learned metric [27], and (c) our proposed method.

Table 2 Comparative results over the FHT dataset.

	DR(%)	DR_2(%)	*PLHR*(%)
HPP-based [1]	62.59	80.91	93.03
Piece-wise projection-based [4]	87.77	96.18	97.16
Proposed method	93.13	98.93	99.06

Table 3 Comparative results over the HIT-MW dataset,

	DR(%)
X-Y cut [1]	45.07
Stroke skew correction [33]	55.34
Docstrum [28]	65.38
Piece-wise projection-based [4]	92.07
MST clustering [27]	95.75
MST clustering+post processing [27]	98.02
Proposed method	98.66

Table 4 Comparative results over the UMD dataset.

	PLHR(%)			DR_2(%)		
	CHN	HIN	KOR	CHN	HIN	KOR
HPP-based [1]	58	43	85	57	49	84
Docstrum [28]	94	74	80	83	72	78
Locally adaptive CC [35]	64	78	83	67	80	81
PDF-based [16]	98	97	98	92	95	96
Mumford-Shah model [18]	98	98	96	-	-	-
Proposed method	99	99	99	99	99	97

Table 5 Comparative results over the ICDAR dataset.

	DR(%)
RLSA	44.3
Projections	68.8
DUTH-ARLSA	73.9
BESUS	86.6
PARC	92.2
LLA	95.2
UoA-HT	95.5
ILSP-LWSeg	97.3
Hough transform-based with post processing [12]	97.4
Piece-wise projection profile and Viterbi-based [5]	98.5
Shredding [17]	98.9
Proposed method	99.3

mated separators are illustrated with black boundaries. The CCs which have intersections with the separators are correctly assigned to the text lines.

Text lines of a Chinese document with small skew and different gaps between text lines have been segmented by X-Y cut [1] and Docstrum [28], and stroke skew correction [33] algorithms and the corresponding results are shown in Figs. 10 (a), (b), and (c) [27], respectively. These approaches have been designed for machine-printed and simple handwritten documents, and are not suitable for complex text line segmentation. Figure 10 (d) shows that the proposed method has extracted correct text lines.

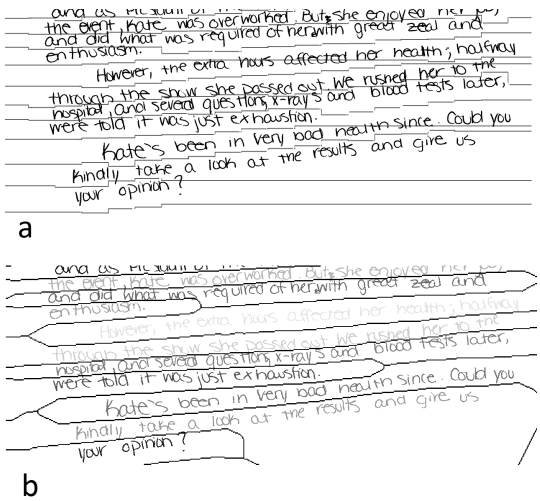

Fig. 12 The results of (a) piece-wise projection method [4], and (b) our proposed method.

The results of two MST clustering methods by using hand-crafted metric [34] and learned metric [27] on a Chinese text image are shown in Figs. 11 (a) and (b) [27], respectively. These methods are sensitive to the gaps between the words in the same text line and also the distances between the text lines. In Fig. 11 (a), some text lines have been erroneously divided into a number of text line parts. In addition, some consecutive text lines are incorrectly connected to each other. Blue lines show these incorrect connections. By using the learned metric, no text line has been divided into segments, but incorrect connections between text lines have been remained. As shown in Fig. 11 (c), the proposed method has correctly segmented the text lines.

Figure 12 (a) [4] shows the result of the piece-wise projection method [4] on a text image with various skews. It can be observed that two text lines have not been correctly extracted. The proposed method estimates the skew of each block separately. Text regions in each block are detected based on the peaks of the smoothed HPP of the deskewed block. Therefore, as shown in Figs. 12 (c), 13 (c), 14 (b), and 14 (d), text lines with large skews are accurately extracted by using the proposed method. Mumford-

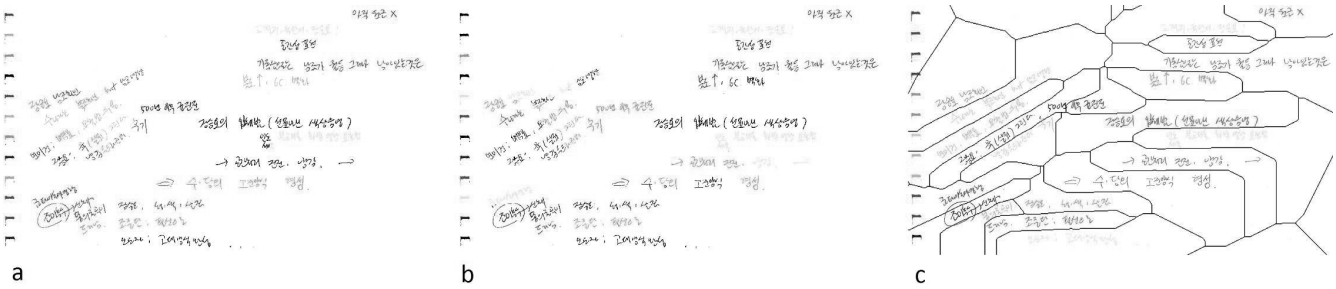

Fig. 13 Segmentation results on a difficult Korean document image of the UMD dataset produced by (a) Mumford-Shah model [18], (b) PDF-based method [16], and (c) the proposed method.

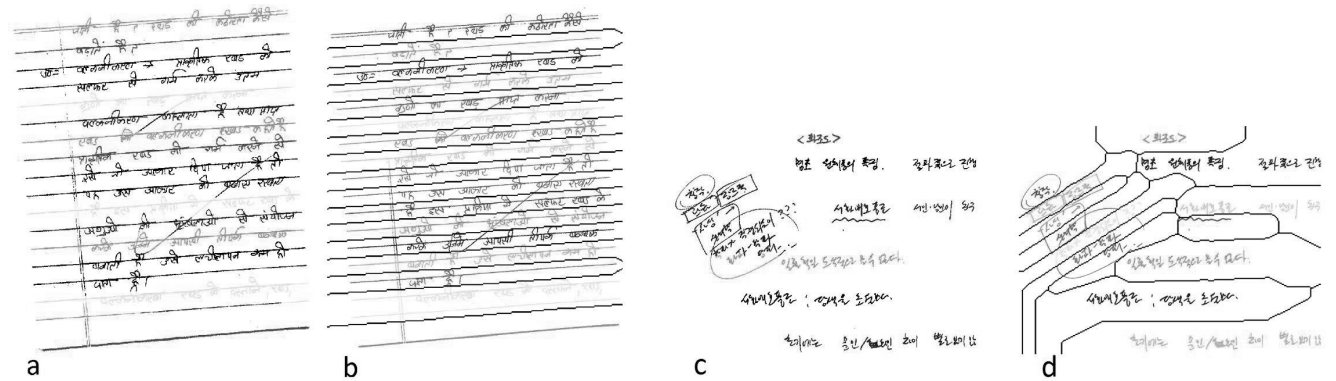

Fig. 14 Segmentation results on Hindi (a, b) and difficult Korean (c, d) document images of the UMD dataset. The results have been produced by the PDF-based method [16] (a, c) and our proposed method (b, d).

Shah model [18] and PDF-based [16] method use horizontal blurring filters. Hence, as shown in Fig. 13 (a) [18], Fig. 13 (b) [16], Fig. 14 (a) [16], and Fig. 14 (c) [16], these approaches cannot extract text lines with large skews. The text image used in Figs. 14 (c) and (d) includes wide range of text line skews. Moreover, the text lines with large skews have some connections. The proposed method has accurately extracted these complex text lines.

In our algorithm, to overcome the multi-skewed text lines problem, a document image is divided into several overlapping blocks. The sizes of blocks are set based on the estimated parameters of the text image, so that in each block, text lines are approximately uniform. The skew of each block is estimated as well as the between-text line distance. To extract touching and overlapping text lines, a text block is first de-skewed and then the HPP of the de-skewed text block is computed. Using the smoothed HPP of the de-skewed text block, touching or overlapping text regions with very small gaps between text lines are separated (Figs. 5 and 6). The distances between text lines of the blocks are considered in the data point clustering phase. Hence, the algorithm can adapt itself with various gaps between text lines.

Moreover, the proposed CC assignment algorithm assigns to the text lines the CCs which have intersections with the estimated separators. The assignment decision is done by considering the skew of text lines. Hence, it can work properly even in the non-horizontal text lines. Defining and using three overall parameters (pen width and effective width and height of CCs) and two local parameters (skew angle and between-text line distance for each block) adapt the proposed algorithm with various handwritings even in different languages.

6. Conclusion

In this paper, a novel script-independent text line extraction algorithm was proposed. This method outperformed all other approaches. The reason is that our method adapts itself with the documents global features such as the effective height and width of CCs. Furthermore, unlike other approaches, in our algorithm, in different parts of text lines, text regions are detected with respect to the corresponding estimated skew angles. Since the skew angle and the distance between text lines are estimated for each block of the document image, the multi-skewed text lines with various distances between text lines are accurately segmented by using the proposed method. Furthermore, our CC assignment algorithm works effectively over non-horizontal text lines.

References

[1] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," Computer, vol.25, no.7, pp.10–22, July 1992.
[2] R.P. dos Santos, G.S. Clemente, T.I. Ren, and G.D.C. Calvalcanti,

"Text line segmentation based on morphology and histogram projection," Proc. 10th International Conference on Document Analysis and Recognition, pp.651–655, 2009.

[3] K. Saeed and M. Albakoor, "Region growing based segmentation algorithm for typewritten and handwritten text recognition," Applied Soft Computing, vol.9, no.2, pp.608–617, 2009.

[4] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to line segmentation in handwritten documents," Proc. SPIE Document Recognition and Retrieval XIV, pp.1–11, 2007.

[5] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, "Handwritten document image segmentation into text lines and words," Pattern Recognit., vol.43, no.1, pp.369–377, 2010.

[6] T. Stafylakis, V. Papavassiliou, V. Katsouros, and G. Carayannis, "Robust text-line and word segmentation for handwritten documents images," Proc. International Conference on Acoustics, Speech and Signal Processing, pp.3393–3396, 2008.

[7] B.B. Chaudhuri and S. Bera, "Handwritten text line identification in Indian scripts," Proc. 10th International Conference on Document Analysis and Recognition, pp.636–640, 2009.

[8] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, "Arabic handwritten text-line extraction," Proc. 6th International Conference on Document Analysis and Recognition, pp.281–285, ICDAR, USA, 2001.

[9] A. Zahour, L. Likforman-Sulem, W. Boussalaa, and B. Taconet, "Text line segmentation of historical arabic documents," Proc. 9th International Conference on Document Analysis and Recognition (ICDAR'07), pp.138–142, Curitiba, Brazil, Sept. 2007.

[10] N. Tripathy and U. Pal, "Handwriting segmentation of unconstrained Oriya text," Proc. International Workshop on Frontiers in Handwriting Recognition, pp.306–311, 2004.

[11] E. Bruzzone and M.C. Coffetti, "An algorithm for extracting cursive text lines," Proc. 5th International Conference on Document Analysis and Recognition, pp.749–752, Bangalore, India, 1999.

[12] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," Pattern Recognit., vol.42, no.12, pp.3169–3183, 2009.

[13] V. Malleron, V. Eglin, H. Emptoz, S. Dord-Crousle and P. Regnier, "Text lines and snippets extraction for 19th century handwriting documents layout analysis," Proc. 10th International Conference on Document Analysis and Recognition, ICDAR, pp.1001–1005, 2009.

[14] G. Louloudis, B. Gatos, and C. Halatsis, "Text line detection in unconstrained handwritten documents using a block-based Hough transform approach," Proc. Ninth International Conference on Document Analysis and Recognition, ICDAR, vol.2, pp.599–603, 2007.

[15] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line detection in handwritten documents," Pattern Recognit., vol.41, no.12, pp.3758–3772, Dec. 2008.

[16] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger, "Script-independent text line segmentation in freestyle handwritten documents," IEEE Trans. Pattern Anal. Mach. Intell., vol.30, no.8, pp.1313–1329, 2008.

[17] A. Nicolaou and B. Gatos, "Handwritten text line segmentation by shredding text into its lines," Proc. 10th International Conference on Document Analysis and Recognition, pp.626–630, 2009.

[18] X. Du, W. Pan, and T.D. Bui, "Text line segmentation in handwritten documents using Mumford-Shah model," Pattern Recognit., vol.42, no.12, pp.3136–3145, 2009.

[19] M. Ziaratban and K. Faez, "An adaptive script-independent block-based text line extraction," Proc. International Conference on Pattern Recognition, pp.249–252, 2010.

[20] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, and D.K. Basu, "Text line extraction from multi-skewed handwritten documents," Pattern Recognit., vol.40, no.6, pp.1825–1839, 2007.

[21] U. Pal and S. Datta, "Segmentation of Bangla unconstrained handwritten text," Proc. International Conference on Document Analysis and Recognition, vol.2, pp.1128–1132, 2003.

[22] U. Pal and P.P. Roy, "Multioriented and curved text lines extraction from Indian documents," IEEE Trans. Syst. Man Cybern. B, Cybernetics, vol.34, no.4, pp.1676–1684, 2004.

[23] S. Tsuruoka, Y. Adachi, and T. Yoshikawa, "The segmentation of a text line for a handwritten unconstrained document using thinning algorithm," Proc. International Workshop on Frontiers in Handwriting Recognition, pp.505–510, Amsterdam, 2000.

[24] H. Goto and H. Aso, "An algorithm for extraction of rules and field-separators in document image," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J78-D-II, no.12, pp.1935–1939, Dec. 1995.

[25] H. Goto and H. Aso, "Robust and fast text-line extraction using local linearity of the textline," Systems and Computers in Japan, 01.26, no.13, pp.21–31, Nov. 1995. (Translated from IEICE Trans. Inf. & Syst., vol.J78-D-II, no.3, pp.465–473, March 1995.)

[26] H. Yang and S. Ozawa, "Extraction of bibliography information based on the image of book cover," IEICE Trans. Inf. & Syst., vol.E82-D, no.7, pp.1109–1116, July 1999.

[27] F. Yin and C.L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," Pattern Recognit., vol.42, no.12, pp.3146–3157, 2009.

[28] L. O'Gorman, "The document spectrum for page layout analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol.15, no.11, pp.1162–1173, 1993.

[29] www.iit.demokritos.gr/˜bgat/HandSegmCont2009

[30] M. Ziaratban, K. Faez, and F. Bagheri, "FHT: An unconstraint Farsi handwritten text database," Proc. International Conference on Document Analysis and Recognition, ICDAR09, pp.281–285, 2009.

[31] T. Su, T. Zhang, and D. Guan, "Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text," International Journal on Document Analysis and Recognition, vol.10, pp.27–38, 2007.

[32] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," Proc. First International Workshop on Document Image Analysis for Libraries, pp.306–312, 2004.

[33] T. Su, T. Zhang, H. Huang, and Y. Zhou, "Skew detection for Chinese handwriting by horizontal stroke histogram," Proc. International Conference on Document Analysis and Recognition, pp.899–903, 2007.

[34] F. Yin and C.L. Liu, "Handwritten text line extraction based on minimal spanning tree clustering," Proc. International Conference on Wavelet Analysis and Pattern Recognition, vol.3, pp.1123–1128, 2007.

[35] S. Jaeger, G. Zhu, D. Doermann, K. Chen, and S. Sampat, "DOCLIB: A software library for document processing," Proc. SPIE Document Recognition and Retrieval XIII, pp.63–71, 2006.

Majid Ziaratban received B.Sc. and M.Sc. degrees in Electronic Engineering from Guilan University in 2002 and Amirkabir University of Technology in 2005, respectively. Currently, he is a Ph.D. student at the Electrical Engineering Department of Amirkabir University of Technology, Tehran, Iran. His research interests include Farsi and Arabic document analysis and recognition, computer vision and pattern recognition. He is a reviewer of the International Journal of Document Analysis and Recognition.

Karim Faez received his B.S. degree in Electrical Engineering from Tehran Polytechnic University as the first rank in June 1973, and his M.S. and Ph.D. degrees in Computer Science from University of California at Los Angeles (UCLA) in 1977 and 1980 respectively. Prof. Faez was with Iran Telecommunication Research Center (1981–1983) before joining Amirkabir University of Technology in Iran. He was the founder of the Computer Engineering Department of Amirkabir University in 1989 and he has served as the first chairman during April 1989–Sept. 1992. Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, research and Technology (during 1988–1996). His research interests are in Pattern Recognition, Image Processing, Neural Networks, Signal Processing, Farsi Handwritten Recognition, Earthquake Signal Processing, Fault Tolerant System Design, Computer Networks, and Hardware Design. He is a member of IEEE, and ACM.