Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 9, Number 1, Spring 2020, pp. 46-60
Received: 04/07/2019, Revised: 01/19/2020, Accepted: 04/22/2020

Script-Independent Handwritten Text line Segmentation Using Directional 2D Filters

Majid Ziaratban
Department of Electrical Engineering, Faculty of Engineering, Golestan University, Gorgan, Iran.
m.ziaratban@ gu.ac.ir

Corresponding author address: Majid Ziaratban, Department of Electrical Engineering, Faculty of Engineering, Golestan University, Gorgan, Iran, Post Code: 49188-88369.

Abstract- Text line segmentation is an important stage of the optical character recognition (OCR) algorithms. To analyze and recognize a document, text lines have to be segmented accurately. Text line segmentation of handwritten documents is more difficult than that of machine-printed ones. Curved and multi-skewed text lines, overlapping text lines, and very small text lines are the main challenges. Most of the proposed approaches did not consider local features of text lines in a document image. In our proposed method, both global and local features are considered. The proposed method is based on using directional 2D anisotropic filters. The parameters of our method are tuned based on a main global parameter which is computed for each document, separately. Hence, the proposed method is a dataset-independent method. A document is divided into several blocks for which some local characteristics are calculated. In each block, text regions are detected by using local characteristics such as the block skew. In order to estimate the skew of text regions in a block, a novel text block skew estimation algorithm is proposed in this paper. Experimental results show that the proposed method outperforms all the state-of-the-art methods on three standard datasets. Our final F-Measure are 0.54%, 0.03%, and 0.02% greater than the winner of ICDAR2013 text line segmentation contests on ICDAR2013, ICDAR09, and HIT-MW datasets, respectively. The experiments proved that the proposed method can accurately segment text lines of complicated handwritings.

Keywords- Text line segmentation, handwritten documents, script-independent method, directional 2D filters.

I. Introduction

Segmentation of a document image into text lines is a problem that is not solved completely, particularly for unconstraint handwritten documents. Inaccurate segmentation of text lines reduces the final text recognition performance. Text line (TL) segmentation algorithms have challenges with multi-skewed TLs, curved TLs, correlated/overlapping TLs, and very small TLs.

In machine-printed texts and simple/regular documents, text lines are simply separable by using the methods based on horizontal projection analysis [1]. For documents with more complexity, the performance of these methods dramatically decreases. To improve the accuracy and efficiency, a number of methods were proposed based on piece-wise horizontal projection analysis [2]–[7] in various scripts. In these methods, a document image is divided into vertical strips in which text line parts are detected. By merging detected text line parts, text lines are constructed. Although, these approaches work more accurate than the conventional horizontal projection-based methods, they have problems with large skewed documents, overlapping text lines, and the documents containing incomplete text lines. Furthermore, setting the value of strip width is another problem of these methods [8]. Some recent methods such as [9] and [7] tried to improve the accuracy of projection profile-based methods.

Minimal spanning tree clustering-based method was proposed to extract multi-skewed text lines [10]. Text lines should be written with large enough space between them to be accurately separable by this method. Li et. al. proposed a probability density function-based method by which overlapping text lines were properly extracted. But, multi-skewed text lines were not accurately segmented [11].

A number of text line segmentation methods used Hough transform to detect directions of text lines [12]–[14]. Since the Hough transform requires straight lines (generally, regular shapes) to detect, curved text lines could not be detected by the Hough transform. To overcome this problem, in [13] and [14] a block based Hough transform was used.

Recently, some methods based on run-length analysis [15][16], active contours [17], morphology operations [18]–[20], level set [11], k-shortest paths optimization [21], textures [22], recurrent neural networks [23], convolutional neural networks [24]–[27], graph analysis [28], gradient/derivative [29][30], and dynamic programming [31], have been proposed to segment text lines.

Ziaratban and Faez proposed a script-independent text line extraction method in which the image was divided into several overlapping blocks [8]. The skew of each block was estimated using a directional gradient algorithm. Some representative points were found in each block by processing a horizontal projection profile of the de-skewed block. A clustering algorithm was proposed and applied on representative points of the whole document image to extract the overall text line paths.

Some state-of-the-art methods participated in text line segmentation contests [32][33] and were tested using standard datasets. The result of the proposed method is compared with the results of these approaches. More details about these approaches can be found in [33].

GOLESTAN Method was proposed by Ziaratban and won the third place in the ICDAR2013 contest. In this method, a document image was first blurred by using a 2D Gaussian filter. The blurred image was divided into several overlapping blocks. Skew angles of blocks were computed and the blocks were de-skewed. The block skew estimation algorithm was similar to the one in [8]. Text line regions were extracted by applying an adaptive thresholding algorithm. The binarized block was rotated back to its original skew. By merging extracted text regions, overall text line paths were obtained. Separator boundaries were made by skeletonization of the background of the path image [34].

Most of the state-of-the-art methods participated in text line segmentation contests do not consider local characteristics of text lines and apply their algorithms to the whole document image. To the best of our knowledge, none of the previous methods calculated and used local text line skews for better text line segmentation. In this paper, a novel text block skew estimation algorithm is proposed. Previous methods were almost designed to overcome only one of the main challenges (curved /multi-skewed TLs, and correlated/overlapping TLs). We try to solve both challenges simultaneously.

In the proposed method, first, a vertical centralization algorithm is proposed and applied on CCs (Connected Components) to reduce effects of CCs influences in other adjacent text lines. To overcome the segmentation of multi-skewed and curved text lines, a document image is divided into several overlapping blocks using the approach introduced in [8],[34]. Instead of extraction of text lines from the whole image, for each block, text line sub-regions are extracted with respect to the estimated local skew. A novel skew estimation algorithm is proposed to detect the skew of each block, separately. To overcome the segmentation of overlapping/touching text lines, probability maps of text line pixels are used in the proposed method. An adaptive thresholding approach [34] is used to extract text regions for each block. Merging extracted text regions construct text line regions. After merging text regions, some adjacent text lines may incorrectly connect to each other and make a combined text line region. Two algorithms are proposed to segment text lines of a combined text line region. Moreover, an algorithm is proposed to detect small missing text lines. Skeleton of the background of the detected text line regions constructs separator boundaries by which text lines are separated. Finally, to decide about the CCs which have junctions with the separator boundaries, a CC assignment algorithm is proposed. The block diagram of the proposed method is shown in Fig.1.

The remaining of the paper is organized as follows: in Section II, the proposed methodology for text line detection and segmentation is described. Section III deals with the experimental results and quantitative comparisons of the proposed method and the state-of-the-art methods on three standard datasets. Finally, concluding remarks are presented in Section IV.

II. PROPOSED TEXT LINE SEGMENTATION METHOD

To have a script-independent text line segmentation method, the parameters should be tuned based on some characteristics of an input image. We use the overall height of CCs, h_{cc} , which is an important and useful parameter of a document image and is defined as the average value of height of CCs [34].

2.1. Vertical centralization of CCs

As mentioned earlier, one of the challenging problems in the text line (TL) segmentation is the overlapping TLs. In our method, an algorithm is proposed to reduce the effects of CCs pixels which are not usually close to the overall hypothetical path (baseline) of a text line. To this end, each CC is vertically centralized by using the following equation.

$$CC_{VC}(x,y) = \begin{cases} 0.5\,CC(x,y) & h_i > 2\,h_{CC} \\ CC(x,y)\exp\left(-\frac{x\prime-0.5\,h_i}{2\sigma^2}\right) & (h_i \leq 2\,h_{CC}) \wedge (h_i + w_i > h_{CC}) \\ 0.5\,CC(x,y)\exp\,(-\frac{x\prime-0.5\,h_i}{2\sigma^2}) & elsewhere \end{cases} \quad (1)$$

where $CC(x,y)$ and $CC_{VC}(x,y)$ are the foreground pixels of an original CC and the vertically centralized CC, respectively. h_i and w_i are the height and width of i-th CC, respectively. $I_{VC}(x,y)$ is the image containing vertically centralized CCs. Usually in Gaussian kernels, the value of σ is set to 0.33 of the length of the Gaussian window. Hence, the value of σ in Eq. (1) is set to $0.33h_i$. Fig.2 shows the result of the vertical centralizing of CCs for an instance text image.

2.2. Block skew estimation

Several document skew estimation methods have been proposed that most of them estimates the overall skews for the whole document. In addition, in the conventional methods, the skews of the whole texts in a document should not have wide varieties. To overcome the cases of multi-skewed documents, we have to divide a document into some blocks. Since the text components in a block is very lower than the whole document, a robust and reliable skew estimation approach is needed.

In the proposed method, the most probable angle among the angles assigned to a set of pixels (selected foreground pixels) of a block is considered as the skew of a block. In other words, our block skew estimation method is a pixel-level process.

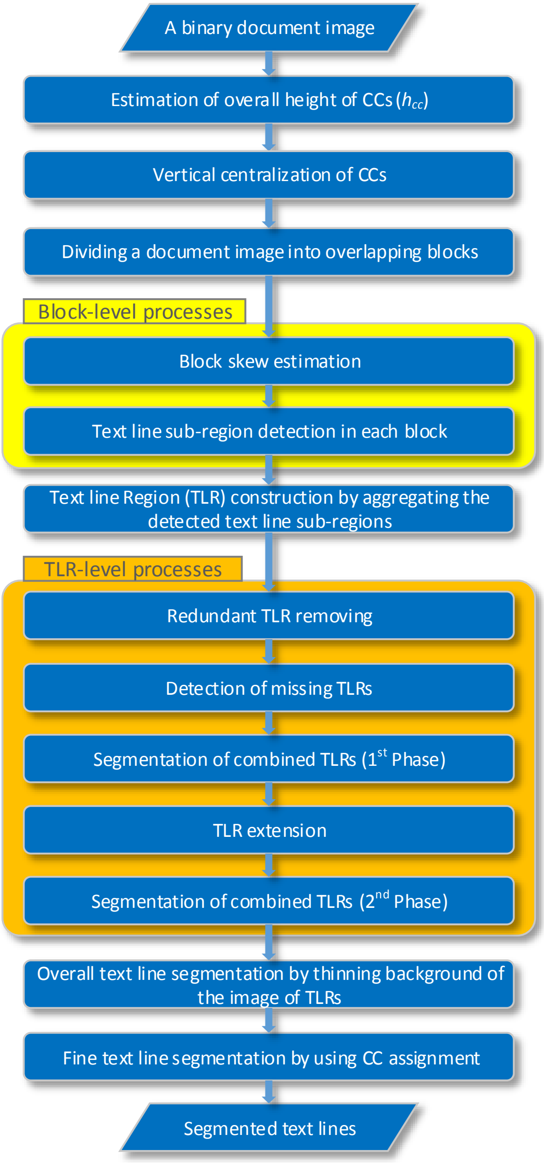

A binary document image
Estimation of overall height of CCs (h_{cc})
Vertical centralization of CCs
Dividing a document image into overlapping blocks
Block-level processes
Block skew estimation
Text line sub-region detection in each block
Text line Region (TLR) construction by aggregating the detected text line sub-regions
TLR-level processes
Redundant TLR removing
Detection of missing TLRs
Segmentation of combined TLRs (1st Phase)
TLR extension
Segmentation of combined TLRs (2nd Phase)
Overall text line segmentation by thinning background of the image of TLRs
Fine text line segmentation by using CC assignment
Segmented text lines

Fig.1. Proposed method

Hence, to have more reliable block skew estimation, a set of reliable pixels is determined (by applying a mask on images) to be used for the block skew estimation. To obtain the mask image, first an input image containing vertical centralized CCs, $I_{VC}(x,y)$, is smoothed by a small 2D Gaussian filter (G_s) and then the blurred image is binarized.

$$G_s(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{x^2}{\sigma_x^2}-\frac{y^2}{\sigma_y^2}\right) \qquad (2)$$

$$I_s(x,y) = I_{VC}(x,y) * G_s(x,y) =$$

$$= \sum_{x'}\sum_{y'} I_{VC}(x'-x, y'-y)G_s(x',y') \qquad (3)$$

In the mask image, we want to have some regions bounding the main body of written words. Hence, the pixels of the mask image should be set to zero in the regions in which the probability of occurrence of text elements is very low. The height of the smoothing filter should be around the heights of the main bodies of characters/words. Therefore, the filter height is set to h_{cc}. Since the overall shape of words is usually wider in horizontal direction, the width of the filter is set to 1.5×filter height. In Gaussian filters, the values of σ are usually set to $\frac{1}{5}$ of the kernel size. Hence, the values of σ_x and σ_y are set to $0.2h_{cc}$ and $0.3h_{cc}$, respectively. A kernel of the filter for a sample text image is shown by a yellow rectangle in Fig.1(c).

Fig.1 (d) shows the smoothed image, I_s, of I_{VC} for the sample text image. Three different thresholds are applied to the smoothed image in Fig.1 (d) to create the mask image. Yellow, Blue, and red regions in Fig.1 (e) obtained by using thresholds equal to 0.08, 0.12, and 0.16 for binarization, respectively. As mentioned earlier, the goal in this stage is to find the regions bounding the main body of written words. From Fig.1 (e), by choosing low threshold values, some regions located between text lines may be incorrectly selected as the reliable pixels for the skew estimation. On the other hand, choosing high threshold values creates regions that do not sufficiently covers the main body of written words. Fig.1 (f) shows that the threshold value equal to 0.12 creates regions that covers the main body of texts and are suitable for the block skew estimation stage.

$$R(x,y) = \begin{cases} 0 & I_s(x,y) < Th \\ 1 & I_s(x,y) \geq Th \end{cases} \qquad (4)$$

where R is the mask image in which the foreground pixels are the corresponding reliable pixels of $I(x,y)$ which will be used for the block skew estimation. A sample document image from the ICDAR13 dataset [33] is deformed by a *fish-eye* transformation and used in this paper to demonstrate the ability of the proposed method in segmentation of complex text lines. The deformed sample image and its computed reliable pixels are given in Fig.3 (a) and (b), respectively.

Pixel-level skews are only calculated for reliable pixels and will be used for the block skew estimation. In other words, the angles corresponding to only the non-zero pixels of $R(x,y)$ are used for the block skew estimation. The pixel-level skew angles are calculated as follows:

First, $I_{VC}(x,y)$ is filtered by directional 2D Gaussian filters, $G_L^t(x,y)$, in various directions. G_L^t is the rotated version of the basic filter G_L^0 where *t* is the rotation angle. G_L^0 is a 2D Gaussian filter whose width is much longer than its height. The height of G_L^0 should be large enough to cover the text parts of a text line. On the other hand, to avoid overlapping two consecutive lines, it must be narrow. So the height of is set to $1.5h_{cc}$. To have a sufficiently elongated distribution for occurrence of pixels of a text line, the width of G_L^0 is considered to be 10 times higher than its height. The basic kernel, G_L^0, for a sample text image is shown by a yellow rectangle in Fig.2(a).

$$I_L^t(x,y) = I_{VC}(x,y) * G_L^t(x,y) \qquad (5)$$

Fig.3 (c), (e), and (g) show directional filtered images corresponding to *t* = -5, t = 0, and *t* = 5 degrees, respectively. A morphological *opening* operation is applied to $I_L^t(x,y)$ to increase differences between values of pixels of the text regions which are located in the directions around *t* and the ones which are in other directions. The structural element, S^t, used in the opening process is a directional single-width straight line in the direction equal to *t*. The length of S^t should be at least equal to $\frac{Line\_Thickness}{\sin\Delta\theta}$ pixels to reduce the values of pixels of TL parts with skews greater than $t+\Delta\theta$ or lower than $t-\Delta\theta$ degrees (Fig.3(c)). By assuming $Line\_Thickness = h_{cc}$ and $\Delta\theta = 5$ degrees, the minimum length of S^t is obtained equal to $11.5h_{cc}$ pixels. In our experiments, the length of S^t has been set to $13h_{cc}$ pixels.

$$I_{L,o}^t(x,y) = I_L^t(x,y) \circ S^t \qquad (6)$$

where (∘) is the morphological opening operator. Fig.3 (d), (f), and (h) illustrate results of applying the opening operator on Fig.3 (c), (e), and (g), respectively.

$I_{L,o}^t(x,y)$ is then divided to several overlapping blocks. In a very big block, text line parts may have none-uniform manners. In other hand, very small blocks contain insufficient text components for skew estimation and text region detection. Usually, in a document image, three successive text lines have similar characteristics. In normal document images, the vertical distance between successive text lines is about $3h_{cc}$ to $5h_{cc}$. To have blocks containing uniform text line parts, the width of blocks should be no longer than about $18h_{cc}$. Hence, in the proposed method, the height (H_b) and width (W_b) of overlapping blocks are set to $12h_{cc}$ and $15h_{cc}$ pixels, respectively. Also, the overlapping amount between adjacent blocks is set to 80%.

In our notation, $I_{L,o,\{i,j\}}^t(x',y')$ is a horizontally *i*-th and vertically *j*-th block of $I_{L,o}^t(x,y)$, respectively. (x,y) and (x',y') are the coordinates of a pixel in a whole image and in a text block, respectively. A skew value is assigned to each pixel of the document image located in non-zero regions of $R(x,y)$ as follows:

$$\Phi(x,y) = \begin{cases} \tau(x,y) & R(x,y) = 1 \\ undefined & elsewhere \end{cases} \qquad (7)$$

where $\tau(x,y)$ is the angle that causes the pixel values of $I_{L,o}^{\tau}(x,y)$ to be the maximum value of $I_{L,o}^t(x,y)$ among various *t* values.

$p_{\{i,j\}}(\tau)$ is the probability distribution of τ in $\Phi_{\{i,j\}}(x',y')$. $p_{\{i,j\}}(\tau)$ is smoothed by a 1×5 averaging mask and yields $\bar{p}_{\{i,j\}}(\tau)$. Fig.4(a) shows pixel-level skews computed for all pixels of the instance image. As shown in this figure, the skews corresponding to the pixels located in the spaces between text lines are not reliable.

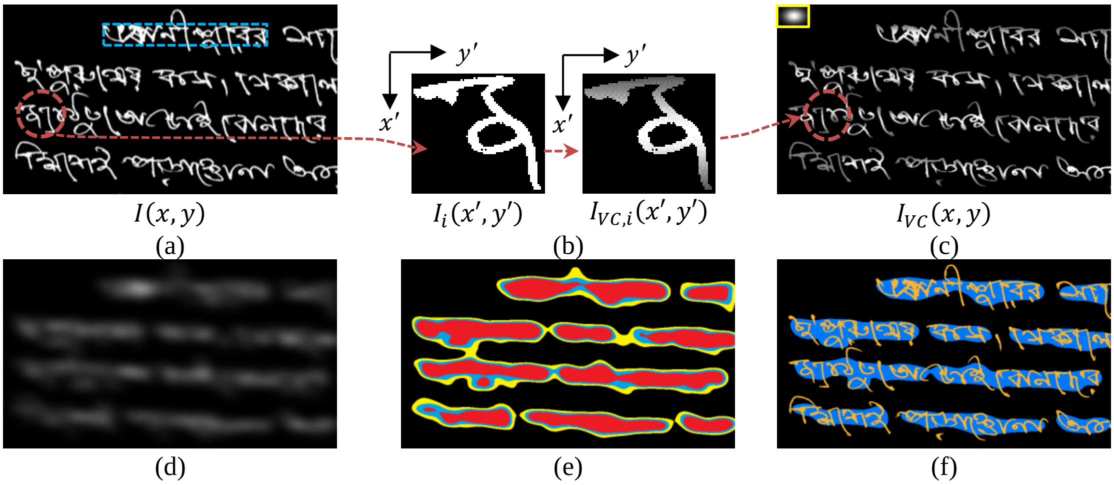
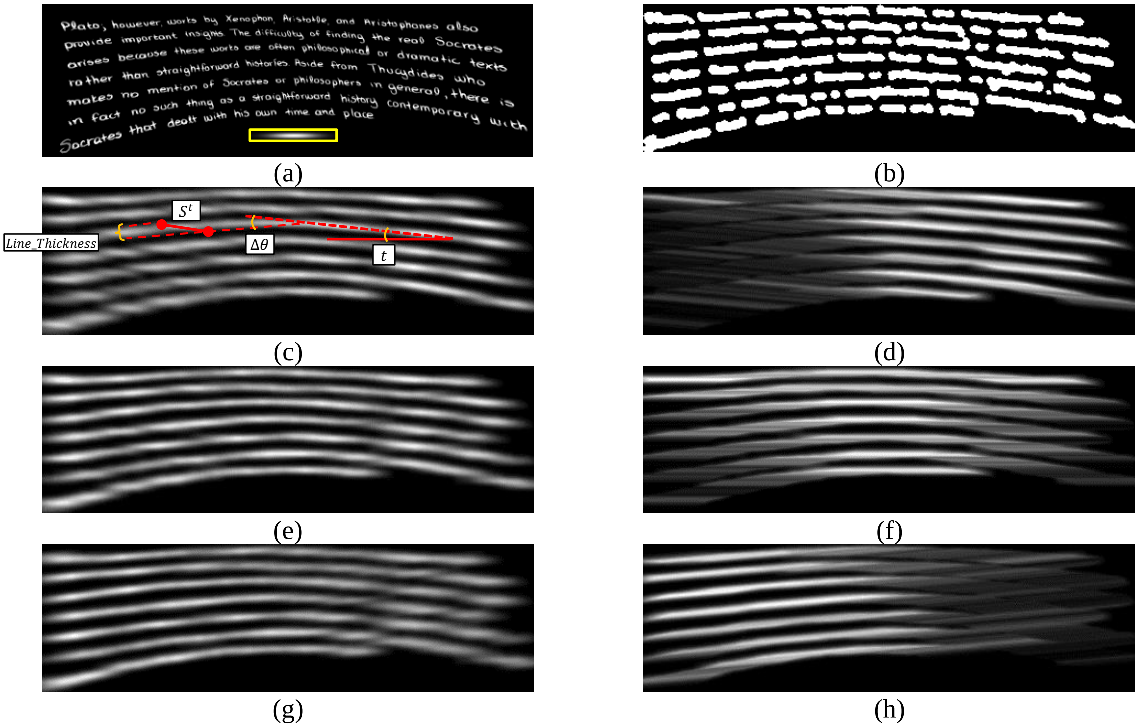


Fig.2. (a)-(c) Vertical centralizing of connected components, (d) Smoothed image. (e) Binarized image of (d) by three different thresholds. (f) Blue regions contain reliable pixels for the block skew estimation


Plato, however, works by Xenophon, Aristotle, and Aristophanes also
provide important insights. The difficulty of finding the real Socrates
arises because these works are often philosophical or dramatic texts
rather than straightforward histories. Aside from Thucydides who
makes no mention of Socrates or philosophers in general, there is
in fact no such thing as a straightforward history contemporary with
Socrates that dealt with his own time and place
S^t
Line_Thickness
$\Delta\theta$
t

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

Fig.3. (a) A sample manually deformed document image after CC vertical centralization. (b) Reliable pixels for skew estimation. (c), (e), and (g) are directional blurred images in directions of -5, 0, and +5 degrees, respectively. (d), (f), and (h) are the results of applying the opening operator on (c), (e), and (g), respectively.

The computed $\Phi(x,y)$ for the sample image is illustrated in Fig.4(b). Blue pixels in this figure are zero pixels of $R(x,y)$ and hence, are not used for block skew estimation. Fig.4(b) shows that the non-zero pixels of $R(x,y)$ are suitable pixels for block skew estimation. $\Phi_{\{i,j\}}(x',y')$, $p_{\{i,j\}}(\tau)$, and $\bar{p}_{\{i,j\}}(\tau)$ for two instance blocks of Fig.4(b) are shown in Fig.5. Peak value in $\bar{p}_{\{i,j\}}(\tau)$ determines the skew of the block $I_{\{i,j\}}(x',y')$.

$$\vartheta_{i,j} = \arg_\tau \max\{\bar{p}_{\{i,j\}}(\tau)\} \qquad (8)$$

The skews for two blocks shown in Fig.5(a) and (c) are estimated equal to +7 and -7 degrees, respectively. Although the proposed algorithm estimates reliable skews, errors may occur in some cases. To avoid diffusion of the skew estimation errors to the following steps, a modified averaging is applied to the set of calculated $\vartheta_{i,j}$ and the final block skew $\theta_{i,j}$ is computed as follows:

$$\theta_{i,j} = \begin{cases} \overline{\vartheta_3}_{\ i,j} & |\vartheta_{i,j} - \overline{\vartheta_3}_{\ i,j}| < 5° \\ \overline{\vartheta_5}_{\ i,j} & else\ where \end{cases} \qquad (9)$$

where $\overline{\vartheta_k}_{\ i,j}$ is the average skew of k×k neighbor blocks around the block $I_{\{i,j\}}$.

Fig.6 illustrates $\vartheta_{i,j}$ and $\theta_{i,j}$ computed for all blocks of the instance image. Black regions in this figure belong to the blocks for which the number of foreground pixels is lower than %1 of total number of the block pixels and hence, these blocks are not processed.

2.1. Text line sub-regions detection

In this step, $I_{VC}(x,y)$ is first blurred with very long directional 2D Gaussian filters.

$$I_{VL}^t(x,y) = I_{VC}(x,y) * G_{VL}^t(x,y) \qquad (10)$$

G_{VL}^t is a rotated version of G_{VL}^0. The value of σ_x should be small enough to avoid touching the adjacent text lines (between $0.1h_{cc}$ to $0.4h_{cc}$). The value of σ_y should be large enough (greater than a block width). In our method, the values of σ_x and σ_y are set to $0.2h_{cc}$ and $20\,h_{cc}$, respectively. The blurred images by applying very long 2D Gaussian filters in directions of +7 degrees and -7 degrees on the instance image shown in Fig.3(a) are given in Fig.7(a) and (b), respectively.

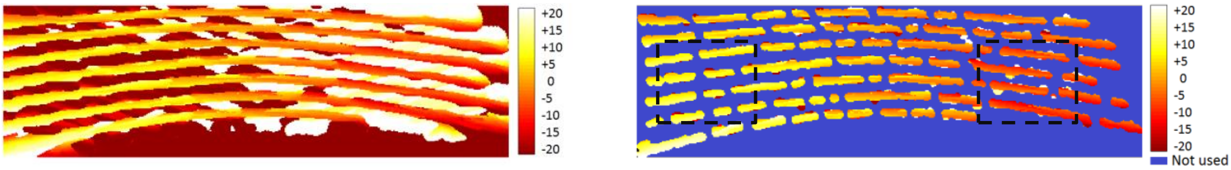
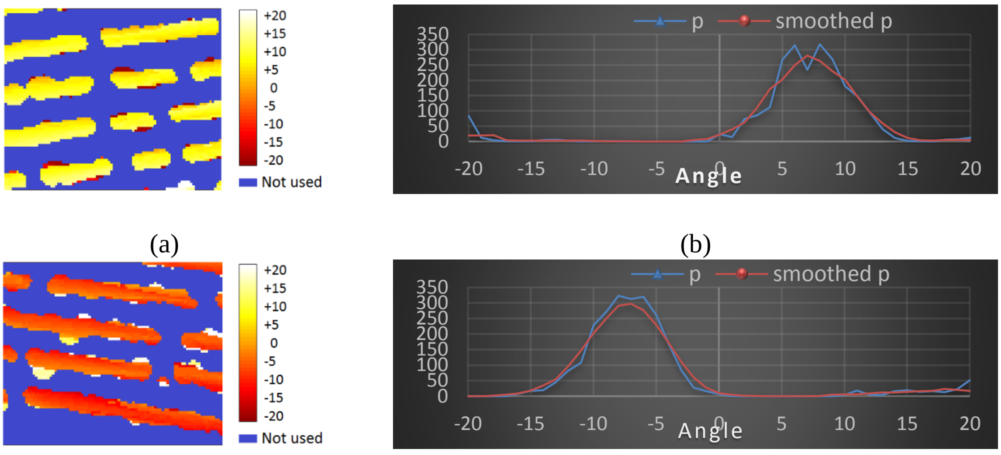


Fig.4: Pixel-level skews obtained for all pixels (left) and the ones for reliable pixels (right)



(a) (b)

(c) (d)

Fig.5. Block skew estimation. (left column) pixel-level skews of reliable pixels for two sample blocks. (right column) probability distributions of skew angles ($p_{\{i,j\}}(\tau)$, and $\bar{p}_{\{i,j\}}(\tau)$ are illustrated with blue and red markers, respectively)

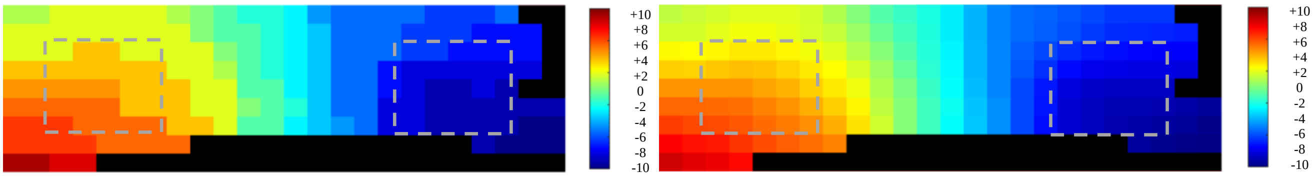


Fig.6. Computed $\vartheta_{i,j}$ (a) and $\theta_{i,j}$ (b) for all blocks of the instance image

The locations of two blocks shown in Fig.5 (a) and (c) are illustrated by red squares in the blurred images in Fig.7. Since the skew angle of the block $I_{\{i,j\}}(x,y)$ is estimated and known, we use $I^{\theta_{i,j}}_{VL,\{i,j\}}(x',y')$ for detection of text line sub-regions. This filtered block is de-skewed to have a horizontal distribution of text line pixels.

The differences between pixel values of text line regions and background in $I^{\theta_{i,j}}_{VL,\{i,j\}}(x',y')$ are the greatest among all directional filtered blocks, $I^{t}_{VL,\{i,j\}}(x',y')$. Hence, in $I^{H}_{VL,\{i,j\}}(x',y')$, text line regions and background are horizontally separable.

$$I^{H}_{VL,\{i,j\}}(x',y') = rotate\{I^{\theta_{i,j}}_{VL,\{i,j\}}(x',y')\ ,-\theta_{i,j}\} \qquad (11)$$

By using suitable local thresholds, more reliable text line sub-regions are extracted. An adaptive thresholding is used to extract more accurate text line sub-regions [34]. The maximum value of each row of $I^{H}_{VL,\{i,j\}}(x',y')$ is stored in $Q_{\{i,j\}}(x')$. The adaptive threshold values are set based on local peaks and valleys of the $Q_{\{i,j\}}$. For *k*-th peak value of $Q_{\{i,j\}}$, m_k , the previous valley, v_k, and next valley,v_{k+1}, are detected and the *k*-th threshold is computed as follows:

$$Th_{\{i,j\},k} = \max\{0.3(Q_{\{i,j\}}(m_k) - q_{min}) + q_{min}\ ,0.02\} \qquad (12)$$

$$q_{min} = \max\{Q_{\{i,j\}}(v_k)\ ,Q_{\{i,j\}}(v_{k+1})\} \qquad (13)$$

In other words, we compute a particular threshold for each horizontal strip of a filtered block $I^{H}_{VL,\{i,j\}}$. The k-th horizontal strip is bounded between v_k and v_k+1. By using the proposed adaptive thresholding, text regions corresponding to highly correlated text lines and very short text lines are accurately detected and extracted. A minimum value (0.02) is considered for $Th_{\{i,j\},k}$ to avoid extracting very weak text regions which are usually resulted from dots, diacritics, and noise.

To obtain more reliable sub-regions and avoid occurring faults, for each block, the only sub-regions which are located horizontally between $\frac{1}{4}W_b$ and $\frac{3}{4}W_b$, and vertically between $\frac{1}{5}H_b$ and $\frac{4}{5}H_b$, are considered and the other sub-regions are ignored. H_b and W_b are the height and width of text blocks, respectively.

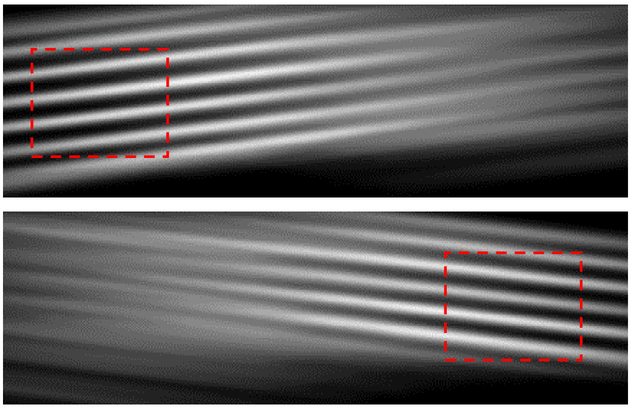


Fig.7. (top) and (bottom) are the images blurred by applying very long 2D Gaussian filters in directions of +7 degrees and -7 degrees on the instance shown in Fig.3(a), respectively.

The result of the adaptive thresholding of the block $I^{H}_{VL,\{i,j\}}$ is a binary image, $B^{H}_{\{i,j\}}$, containing horizontal sub-regions. The obtained $B^{H}_{\{i,j\}}$ is rotated back by $\theta_{i,j}$ degrees to create sub-regions of $I^{\theta_{i,j}}_{VL,\{i,j\}}$:

$$B^{\theta_{i,j}}_{\{i,j\}}(x',y') = rotate\left(B^{H}_{\{i,j\}}(x',y'),\theta_{i,j}\right) \tag{14}$$

Fig.8(a) and (b) illustrate the extracted text line sub-regions, $B^{\theta_{i,j}}_{\{i,j\}}(x',y')$, corresponding to the blocks shown in Fig.5(a) and Fig.5(c), respectively.

2.2. Text line Region construction

To construct the text line regions of the whole input image, $I(x,y)$, an empty binary image, $B(x,y)$, is considered. The corresponding non-zero pixels in $B^{\theta_{i,j}}_{\{i,j\}}(x',y')$ are set to 1 in $B(x,y)$. Doing the above procedure for all blocks, $B(x,y)$ is completely constructed and contains text line regions (TLRs) of the input image. As shown in Fig.8(c), curved and skewed TLRs are correctly detected.

In some cases, a small text line may be ignored in the TLR detection phase. In addition, two text line regions corresponding to overlapping text lines may connect together and construct a combined TLR. Moreover, in some cases, some redundant TLRs may be created. The solutions proposed to overcome these problems are described in the following.

2.4.1. Redundant TLR removing

In our method, to remove redundant TLRs, the smallest acceptable text line is assumed to have at least three CCs. The length and thickness of a CC are usually around $2h_{cc}$ and $0.1h_{cc}$ pixels, respectively.

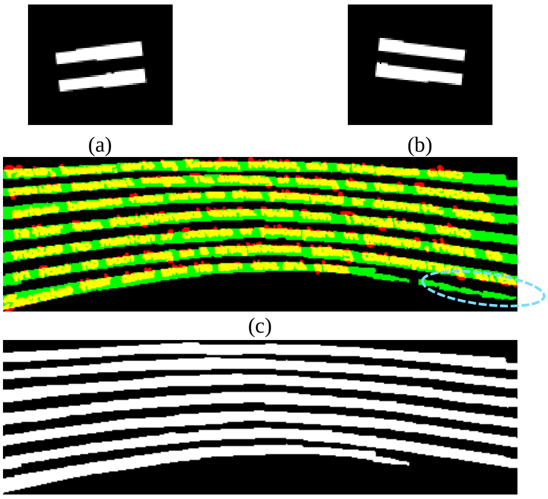


Fig.8. (a) and (b) are extracted text line sub-regions corresponding to the blocks shown in Fig.5(a) and 5(c), respectively. (c) Extracted text line regions by merging text line sub-regions extracted from the whole image blocks. (d) Remaining TLRs after redundant TLR removing

Hence, the smallest text line have about $0.6h^{2}_{cc}$ pixels. Therefore, a detected TLR whose corresponding CCs have totally lower than $0.3h^{2}_{cc}$ pixels is considered as a redundant TLR. In Fig.8 (c), the marked TLR covers no CCs and is removed (Fig.8 (d)).

2.4.2. Detection of missing TLRs

To detect missing TLRs, the remaining CCs which are not covered by the detected TLRs, are considered. This set of the remaining CCs are divided into a number of subsets in which the distance between adjacent CCs is no longer than h_{cc}.

As mentioned earlier, the smallest acceptable text line should have about $0.6h^{2}_{cc}$ pixels. To have more reliable decisions, a subset whose foreground pixels are greater than $0.8h^{2}_{cc}$ pixels is considered as a new text line and its corresponding TLR is constructed by applying a closing morphological operator with a structural element of size $1 \times h_{cc}$ on the subset.

An instance Chinese document image from the HIT-MW dataset [35] is shown in Fig.9(a). Blue regions are the current detected TLRs. The TLR corresponding to the last text line has not been detected. The subset with large enough foreground pixels is shown in Fig.9(b). The green region in Fig.9(c) is the detected missing TLR.

2.4.3. Segmentation of connected TLRs (First phase)

In some cases, adjacent TLRs may incorrectly connect together. Usually, the thickness of a combined TLR at a connection area is wider than those of single TLRs.

Hence, in the proposed method, undesirable TLR connections are detected by comparing the thickness of each TLR with the global TLR thickness. Usually the variance of the thicknesses of the detected TLRs are very low. Three global parameters: TLRs thickness (T_{TLR}), between-region space width (W_{BRS}), and between-baseline distance (D_{bb}) are calculated as follows:

$$D_{bb} = \arg_{d} \max \left\{p\left(d = d_{q}(y)\right)\right\} \tag{15}$$

$$W_{BRS} = \arg_{\mathrm{w}} \max \left\{p\left(w = w_{q}(y)\right)\right\} \tag{16}$$

$$T_{TLR} = \arg_{t} \max \left\{p\left(t = t_{q}(y)\right)\right\} = D_{bb} - W_{BRS} \tag{17}$$

where q and y are the index of TLR and the index of the image column, respectively. $d_{q}(y)$ is the vertical distance between the center of TLR_{q} and that of TLR_{q+1} in y-th column of B. The index of TLRs increases from top to bottom. $w_{q}(y)$ is the vertical distance between the most bottom pixel of TLR_{q} and the most top pixel of TLR_{q+1} in y-th column of B. $t_{q}(y)$ is the vertical distance between the most top pixel and the most bottom pixel of TLR_{q} in y-th column of B. A

TLR is considered as a combined TLR if the following condition is satisfied:

$$t_Q(y) \geq D_{bb} + 0.6\, T_{TLR} \tag{18}$$

Let (x_{cut}, y_{cut}) be the coordinate of a cutting pixel of TLR_Q (which is a combined TLR). y_{cut} is the y that satisfies Eq.(18) and x_{cut} is found as follows:

$$x_{cut} = S_Q(y_{cut}) + \frac{1}{2}(D_{bb} + T_{TLR}) \tag{19}$$

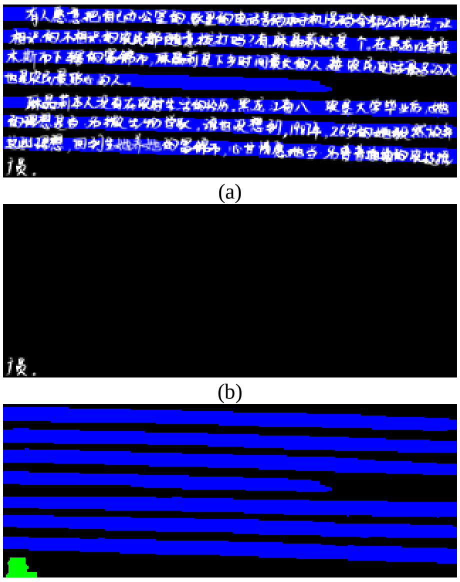
where $S_Q(y)$ is the x-coordinate of the most top pixel in y-th column of TLR_Q. The local skew of a TLR around a cutting pixel (x_{cut}, y_{cut}) is computed as follows:

$$\varphi_{cut} = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1}\left(\theta_{i+i_c,j+j_c}\right) \tag{20}$$

where $I_{\{i_c,j_c\}}$ is the image block that the cutting pixel is located near the center of this block.

The cutting procedure is repeated iteratively. In each iteration, a part of a combined TLR is cut at the cutting pixel (x_{cut}, y_{cut}) with respect to the local skew φ_{cut} and then a new cutting pixel is found. The iteration continues until no new cutting pixel is found. In our method, the cutting length has been set to $6T_{TLR}$.

Fig.10(a) and Fig.11(a) show two instance document images. In both instances, the combined TLRs are correctly detected. Red points in Fig.10(b) are the locations of cutting pixels. The green line in Fig.10(c) shows the cutting length corresponding to the leftmost cutting pixel in Fig.10(b).



(a)

(b)

(c)

Fig.9. Missing TLR Detection (a) An instance Chinese document image [35]. Blue regions are the current detected TLRs. The last text line has been ignored in the TLR detection phase. (b) The CCs subset with large enough foreground pixels. (c) Detected missing TLR (the green region)

Fig.10(d) shows that the first phase of the combined TLR segmentation algorithm segments TLRs of the combined TLR in Fig.10(a). For the instance in Fig.11(a), two text lines have not been completely segmented. Because, not enough cutting pixels have been found for this combined TLR. The remaining combined TLR will be segmented in the second phase (section 2.4.5). In Fig.11, the remaining connected region of the combined TLR (Fig.11(b)) has been zoomed and shown in Fig.11(c).

2.4.4. Segmentation of connected TLRs (Second phase)

In this phase, for each column of a TLR, the number of vertical 0-to-1 transitions is computed and stored in $n_S(y)$. The leftmost column, y_L, and rightmost column, y_R, of a connection area in a combined TLR can be found as follows (Fig.11(c)):

$$y_L = min\{y \mid n_S(y) = 2 \;\; \wedge \;\; n_S(y+1) = 1\} \tag{21}$$

$$y_R = max\{y \mid n_S(y) = 2 \;\; \wedge \;\; n_S(y-1) = 1\} \tag{22}$$

A combined TLR is segmented into two separate TLRs by using a hypothetical separator line. The coordinates of the left and right ends of the separator line are (x_L, y_L) and (x_R, y_R), respectively. x_L and x_R are calculated as follows:

$$x_L = \frac{s_2(y_L) + e_1(y_L)}{2} \tag{23}$$

$$x_R = \frac{s_2(y_R) + e_1(y_R)}{2} \tag{24}$$

where $s_i(y)$ and $e_i(y)$ are the vertical coordinates of i-th 0-to-1 and 1-to-0 transitions in y-th column of a TLR, respectively. Fig.11(c) shows the left and right ends of the hypothetical separator line. Fig.11(d) illustrates the result of the second phase of the combined TLR segmentation algorithm applied on the remaining combined TLR (Fig.11(b)).

2.4.5. TLR length extension

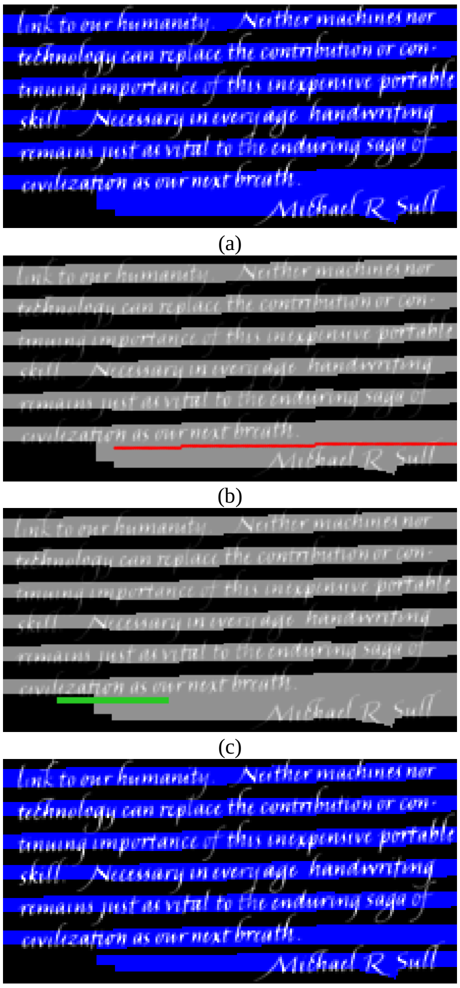
The length of TLRs are extended in both end sides (left and right) to be sure about a TLR covers the whole length of a text line, and also to connect two TLRs which are in the same direction and originally belong to the same text line. Fig.11(d) shows the extended TLRs.

2.3. CC assignment

By thinning the background of the image containing the final TLRs, separator boundaries are obtained, by which the text lines are segmented. Fig.12 shows the separator boundaries for the instance document image.

Some parts of words may be written in between-text line spaces and intersect with separator boundaries. In these cases, it should be decided to which side of the separator boundary the intersected CC belongs. The intersected CCs of an instance document image have been illustrated by red color in Fig.13(a). In this figure, the blue regions and green lines are the detected TLRs and the separator boundaries, respectively.

In our approach, four features (h_i, j_i, n_i, and c_i) are extracted from the intersected CCs of the training set. A number of rules are extracted from the training samples based on these features to assign an intersected CC to its correct text line. Generally, two types of assignments are considered. An intersected CC may completely assign to one side of the separator boundary. In addition, it may be segmented from the intersection point and each part is assigned to the corresponding text line. The extracted CC assignment rules are as follows:

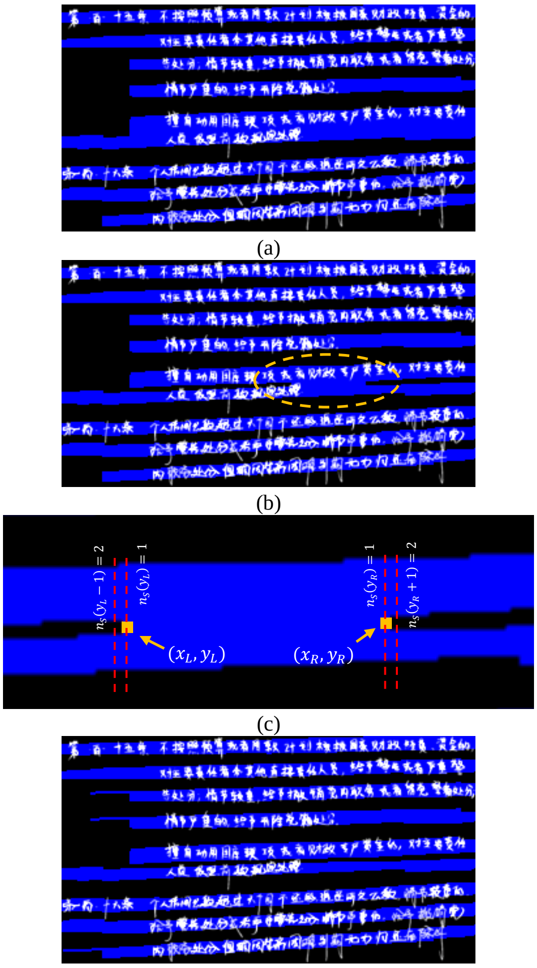

(a)

(b)

(c)

(d)

Fig.10. (a) An instance from the ICDAR2009 [32] dataset. (b) Red points are the cutting pixels. (c) The cutting amount corresponding to the leftmost cutting pixel. (d) The result of the first phase of combined TL segmentation algorithm.



(a)

(b)

(c)

(d)

Fig.11. (a) An instance from the the HIT-MW [35] dataset. (b) The result of the first phase of combined TLR segmentation algorithm. (c) The extended TLRs and the separator line (the red line). (d) The result of the second phase of the combined TLR segmentation algorithm.

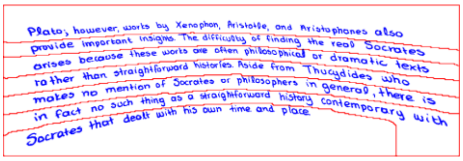


Fig.12. Segmented text lines

$$\begin{aligned}
&if\ \ (n_1 > 3\,n_2)\ \wedge\ (c_2 = 0) && \Rightarrow\ CC \rightarrow TL_1\\
&if\ \ (h_1 > 3\,h_2)\ \wedge\ (n_1 > 1.5\,n_2) && \Rightarrow\ CC \rightarrow TL_1\\
&if\ \ (c_1 = 1)\ \wedge\ (c_2 = 0)\ \wedge\ (n_1 > 1.5\,n_2) && \Rightarrow\ CC \rightarrow TL_1\\
&if\ \ [(c_1 = 1) \vee (j_1 == 1)]\ \wedge\ [(c_2 = 0) \wedge (j_2 = 0)] && \Rightarrow\ CC \rightarrow TL_1\\
&if\ \ (h_1 > 0.4h_{cc})\ \wedge\ (h_2 < 0.25h_{cc})\ \wedge\ (n_2 < 3h_{cc}) && \Rightarrow\ CC \rightarrow TL_1\\
&if\ \ (c_1 = 0)\ \wedge\ (c_2 = 1)\ \wedge\ (n_2 > 3\,n_1) && \Rightarrow\ CC \rightarrow TL_2\\
&if\ \ (c_1 = 0)\ \wedge\ (c_2 = 1)\ \wedge\ (h_2 > 3\,h_1)\ \wedge\ (h_1 < 0.4h_{cc}) && \Rightarrow\ CC \rightarrow TL_2\\
&if\ \ (c_1 = 0)\ \wedge\ (c_2 = 1)\ \wedge\ (h_2 > 0.4h_{cc}) \wedge && \\
&\qquad\quad (h_1 < 0.25h_{cc})\ \wedge\ (n_1 < 3h_{cc}) && \Rightarrow\ CC \rightarrow TL_2
\end{aligned}$$

where h_i and n_i are respectively the height and the number of pixels of CCP_i and CCP_i is the part of the CC in i-th side of a separator boundary (i=1,2). c_i and j_i are logical parameters. c_i is set when the distance between CCP_i and the skeleton of TL_i is lower than $0.2h_{cc}$. j_i is set when the CCP_i has any branches or junction points.

Obviously, if none of the above conditions are satisfied, the intersected CC is not completely assigned to one side of the separator boundary. Fig.13(b) illustrates the result of the CC assignment algorithm for the instant shown in Fig.13(a). In this figure, the parts of the intersected CCs which is assigned to the same text line have been shown in the same color. It can be seen that most of the intersected CCs are assigned to the correct text line.

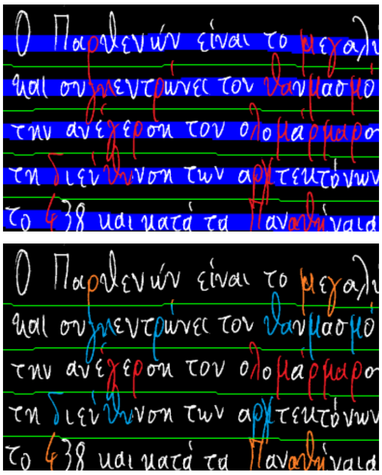


Fig.13. (top) Intersected CCs with separator lines. (bottom) Result of the proposed CC assignment algorithm on the intersected CCs

III. Experiments

In order to evaluate the performance of the proposed method, three standard datasets (HIT-MW [35], ICDAR09 [32], and ICDAR13 [33]) were used. In the ICDAR2013 contest, the set of first 200 document images (from #1 to #200) was available for participants. The test set consists of the next 150 images (from #201 to #350) which was not available. The test set was published after the contest was completed.

In our experiments, parameters tuning and CC assignment rules extraction were done by using only the first 200 document images (from #1 to #200) of the ICDAR2013 dataset. The remaining ICDAR2013 document images and the whole images of HIT-MW and ICDAR09 datasets were used as the test set. ICDAR09 and HIT-MW consist of 100 Latin and 848 Chinese document images, respectively.

The performance evaluation method is based on counting the number of one-to-one matches between the detected text lines and the correct text lines in the ground truth [33]. A *MatchScore* is used to detect the number of one-to-one matches. The value of the *MatchScore* is calculated according to the intersection of the foreground pixel sets of the detected and the ground truth text lines.

Let G_j be the set of foreground pixels inside the j-th ground truth text line, R_i be the set of foreground pixels inside the i-th detected text line, and $T(s)$ be a function that counts the number of non-zero members of set s. $MatchScore(i,j)$ represents the matching amounts of the j-th ground truth text line and the i-th detected text line [33]:

$$MatchScore(i,j) = \frac{T(R_i \cap G_j \cap I)}{T\left((R_i \cup G_j) \cap I\right)} \tag{25}$$

A successful one-to-one match is considered only if the matching score is equal to or above the acceptance threshold T_a. The value of the acceptance threshold was considered T_a = 0.95 [33]. Let N, M, and $o2o$ be the number of ground-truth text lines, detected text lines, and successful one-to-one matches, respectively. The detection rate (DR) and recognition accuracy (RA) are defined as follows [33]:

$$DR = \frac{o2o}{N} \tag{26}$$

$$RA = \frac{o2o}{M} \tag{27}$$

A performance metric FM is defined as follows [33]:

$$FM = \frac{2.DR.RA}{DR+RA} \tag{28}$$

Our previous work [34] was submitted to the ICDAR 2013 handwriting segmentation contest [33] and won the first and third places in the overall segmentation and the text line segmentation competitions, respectively, among 10 professional participant methods. The proposed method outperforms our previous method as well as other methods.

Experimental results on ICDAR2013 [33], ICDAR2009 [32], and HIT-MW [35] datasets are reported in Table 1, Table 2, and Table 3, respectively.

Table 1
Experimental results on the ICDAR2013 dataset [33]

	DR (%)	RA (%)	FM (%)
Graph Analysis [28]	76.97	75.44	75.97
CVC	91.28	89.06	90.16
MSHK	91.66	90.06	90.85
QATAR-a	90.75	91.55	91.15
QATAR-b	91.73	93.14	92.43
NCSR [37]	92.37	92.48	92.43
Algorithm [38]	93.58	92.29	92.93
Dynamic Programming [31]	-	-	93.1
Weighted-Gradient [29]	96.4	94.3	94.8
ILSP [5]	96.11	94.82	95.46
Algorithm [9]	96.37	96.26	96.32
LRDE	96.94	97.57	97.25
TEI [39]	97.77	96.82	97.30
IRISA [40]	97.85	96.93	97.39
CUBS [16]	97.96	96.94	97.45
LAG-Horizontal [25]	97.51	97.73	97.62
Ziaratban [8]	98.02	98.05	98.04
GOLESTAN-a,b [34]	98.23	98.34	98.28
NUS	98.34	98.49	98.41
LAG-Vertical [25]	98.45	98.68	98.56
INMC [36]	98.64	98.68	98.66
Proposed method	**99.16**	**99.24**	**99.20**

TABLE 2
EXPERIMENTAL RESULTS ON THE ICDAR2009 DATASET [32]

	DR (%)	RA (%)	FM (%)
Algorithm [41]	77.59	77.21	77.40
Algorithm [42]	94.47	94.61	94.54
Algorithm [10]	95.86	95.51	95.68
Algorithm [38]	98.31	98.05	98.18
Algorithm [43]	98.59	98.59	98.59
GOLESTAN-a,b [34]	99.10	98.94	99.02
ILSP [5]	99.16	98.94	99.05
Ziaratban [8],[44]	99.33	99.28	99.31
CUBS [16]	99.55	99.50	99.53
INMC [36]	99.60	99.63	99.62
Proposed method	**99.61**	**99.69**	**99.65**

TABLE 3
EXPERIMENTAL RESULTS ON THE HIT-MW DATASET [35]

	DR (%)	RA (%)	FM (%)
Algorithm [1]	65.38	55.62	60.11
Algorithm [3]	92.07	92.28	92.17
Algorithm [45]	95.92	96.86	96.39
Algorithm [10]	98.03	97.53	97.78
Ziaratban [44]	98.34	98.25	98.30
Ziaratban [8]	98.66	98.58	98.62
Algorithm [38]	99.68	99.75	99.71
INMC [36]	99.78	99.88	99.83
Proposed method	**99.80**	**99.90**	**99.85**

The results reported in Table 1 were taken from [33] and the ones in Table 2 and Table 3 were taken from [36]. The overall performance of the proposed method was obtained 0.54% greater than that of the INMC method [36] on the ICDAR2013 dataset. All parameters of our method were set based on the overall height of CCs for each document image and hence, no manual parameter tuning was done for applying the method on various datasets. The proposed method presented the best performance on three different datasets. The reason is the method considers both global and local features of document images. Our block based approach makes us to segment multi-skewed text lines. By using suitable directional 2D anisotropic Gaussian filters with proper parameters, overlapping and connecting text lines were accurately segmented. Another approach proposed to improve separability of overlapping TLs is the vertical centralization of CCs.

Variations of detection rate (DR), recognition accuracy (RA), and performance metric (FM) versus six different acceptance threshold values on three datasets have been illustrated in Fig.14. As can be seen in Fig.14, by considering the acceptance threshold equal to 90%, the performance of the method improves and reaches near 99.8% for all three datasets. It demonstrate that the proposed method accurately has detected TLRs and segmented text lines.

A document image with FM value lower than 100% is considered as a text line segmentation error. Fig.15 and Fig.16 illustrate some of these errors. Generally, two types of errors were seen in the segmentation results. The number of errors occurred in each dataset are reported in Table 4.

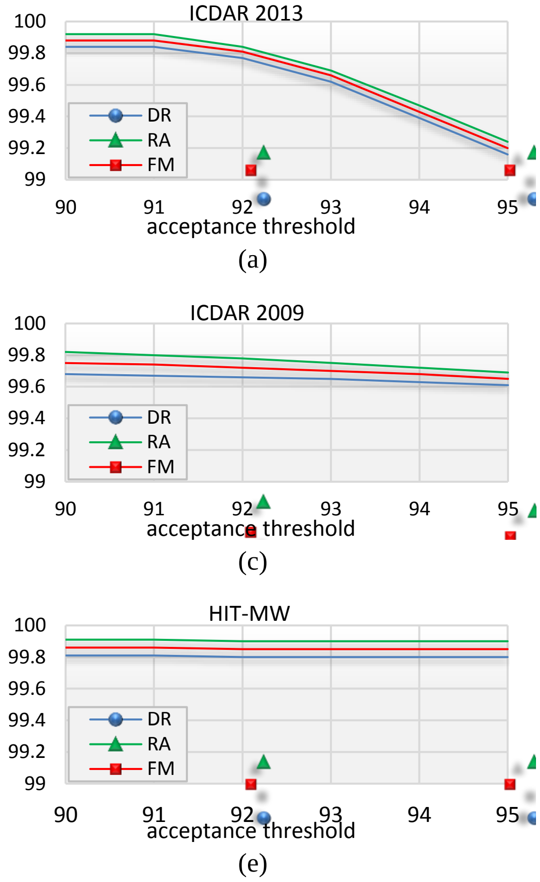


Fig.14. The detection rate (*DR*), recognition accuracy (*RA*), and Performance metric (FM) variations vs. various acceptance threshold values obtained on (a) the ICDAR 2013 [33], (b) ICDAR 2009 [32], and (c) HIT-MW datasets.

From Table 4, only 14 small text lines out of 13518 text lines were not correctly detected. As shown in Fig. 16, the text lines of these documents were correctly detected. But the *MatchScore* of some of the detected text lines were obtained a little lower than 95%. The problems of detecting very small missing text lines and increasing the *MatchScore* of the overlapping text lines can be solved by applying some post-processing.

To evaluate the robustness of the proposed method about the skews of handwritten text lines, another experiment was performed. In this experiment, both handwriting images and ground truth files of the ICDAR2013 dataset were rotated manually with various skews. The results of the text line segmentation of these rotated documents have been reported in Fig.17. The results in this figure demonstrate that the proposed method accurately segments the rotated text lines. The text line segmentation performance (F-Measure) of the proposed method is upper than 98% even for the documents manually rotated by +/-20 degrees. The segmentation results of two highly rotated documents have been illustrated in Fig.19. The documents shown in Fig.19(a) and (b) were manually rotated by -15 and +20 degrees, respectively.

In order to illustrate the benefits of the proposed method compared to our previous methods ([8] and [34]), two other comparisons were performed on manually complicated data. The first comparison was done on manually rotated documents of the ICDAR2013 dataset. Fig.18 shows the *F-Measure* values vs. skew angles obtained by Ziaratban [8], GOLESTAN-a,b [34], and the proposed method.

The second comparison was performed over manually distorted documents of the ICDAR2013 dataset and the results have been reported in Table 5. The manually distorted documents are available for research purposes[1]. Fig.20 shows some manually distorted documents. The results in Fig.18 and Table 5 demonstrate that the proposed method outperformed the methods proposed in [8] and [34]. Some novelties of the proposed method compared to the previous methods ([8],[34]) which caused significant performance improvement (particularly in complicated document images) are as follows:

- Vertical centralization of text parts
- Pixel-based block skew estimation and skew correction (Fig.6)
- Redundant TLR removing
- Detection of missing TLRs
- Segmentation of connected TLRs
- CC assignment

Conclusion

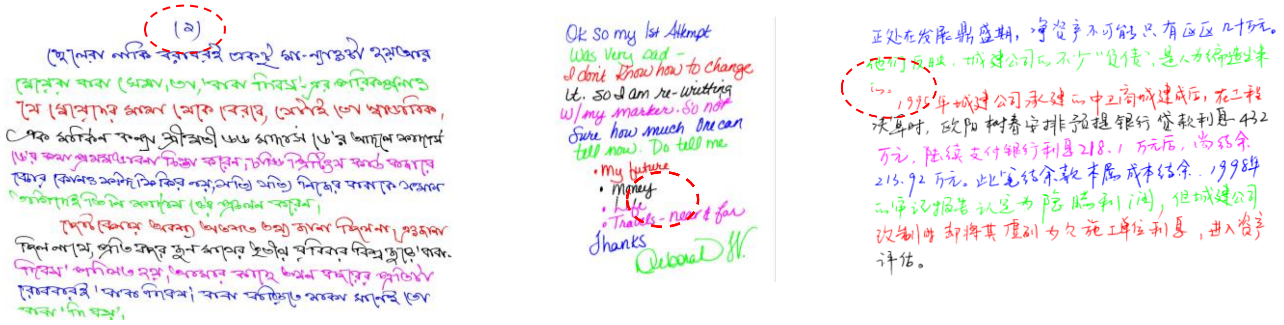
In this paper, we proposed a script-independent text line segmentation method for handwritten documents. Local skews were estimated and used to overcome the challenge of curved and multi-skewed text line segmentation. Our proposed skew estimation algorithm is based on pixel-level skews which is only calculated for the pixels with high probability of occurrence in final text regions. Directional 2D Gaussian filters were used to emphasize the regions containing texts and to decrease the effect of pixels located in between-text line spaces. The proposed algorithm for vertical centralization of CCs improved separability of the overlapping text lines. Both vertical centralization and pixel-level block skew estimation methods have been proposed in this manuscript and have not been introduced and used before. By applying an adaptive thresholding algorithm to the directional filtered blocks, text sub-regions for each block were detected. Text regions were constructed by merging all detected sub-regions.

TABLE 4
NUMBER OF TEXT LINE SEGMENTATION ERRORS

Dataset	# of document images	# of text lines	# of missing TLs	# of remaining combined TLs	# of documents with FM<100% caused by overlapping CCs
ICDAR2013	150	2649	2	0	9
ICDAR2009	100	2249	2	0	6
HIT-MW	848	8620	10	0	5
Total	1098	13518	14	0	20

TABLE 5
EXPERIMENTAL RESULTS ON MANUALLY DISTORTED IMAGES OF ICDAR2013 DATASET

	Ziaratban [8]	GOLESTAN-a,b [34]	Proposed method
ICDAR2013 [33]	98.04	98.28	99.20
Manually distorted ICDAR2013	80.37	82.86	90.84



(a) ICDAR2013 (b) ICDAR2009 (c) HIT-MW

Fig. 15. Missing text lines

[1] http://uupload.ir/view/xau2_distorted_document_images_of_icdar2013.rar

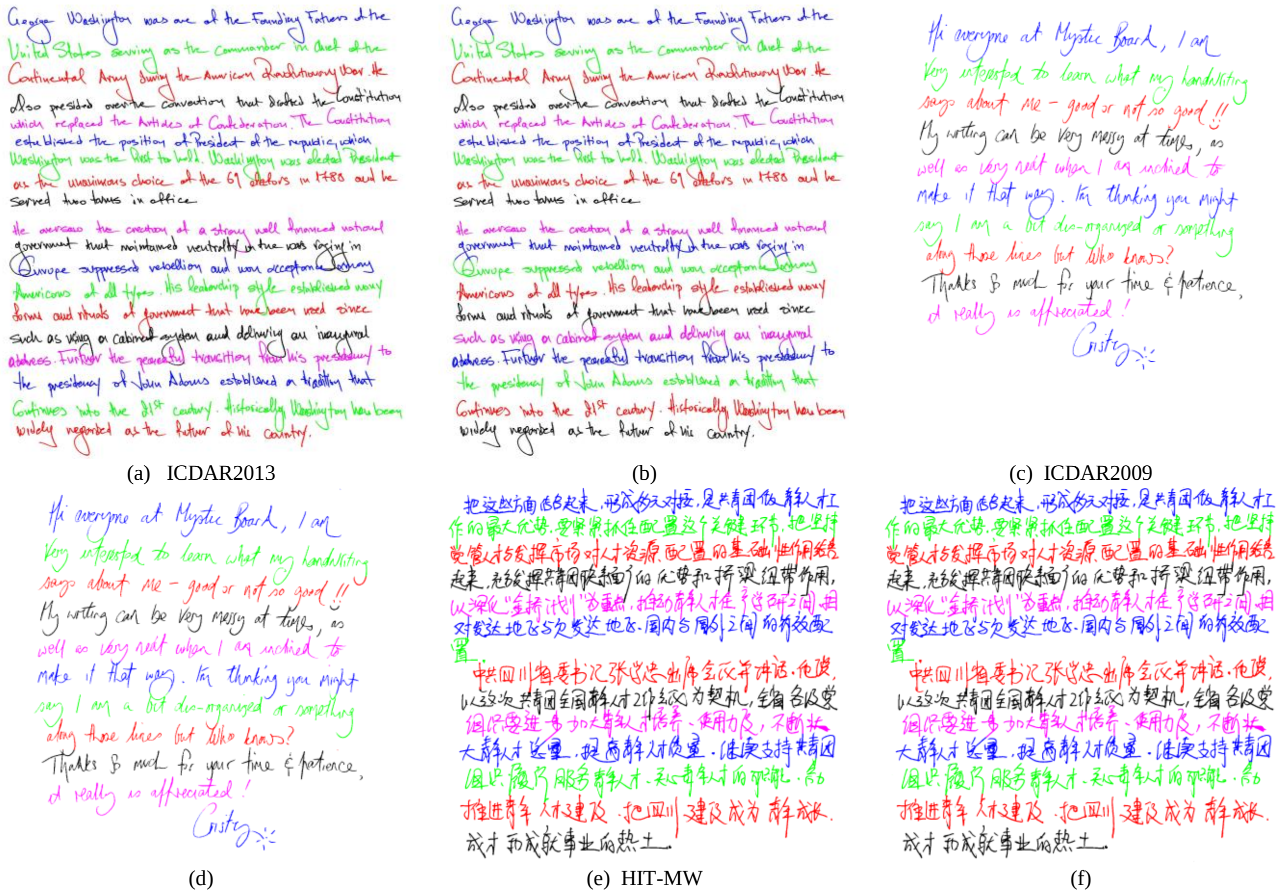


Fig. 16. The text line segmentation errors arise from the highly overlapping CCs. (left column) ground truth files. (right column) the proposed text line segmentation results.

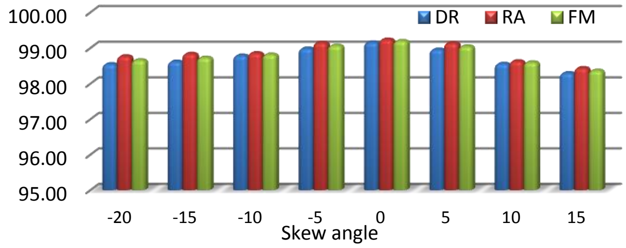


Fig. 17. The text line segmentation performance of manually skewed document images of the ICDAR2013 dataset

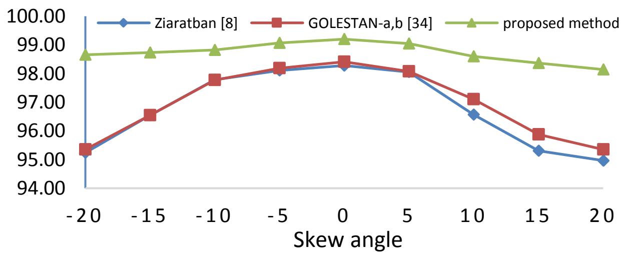


Fig. 18. The text line segmentation performance of manually skewed document images of the ICDAR2013 dataset

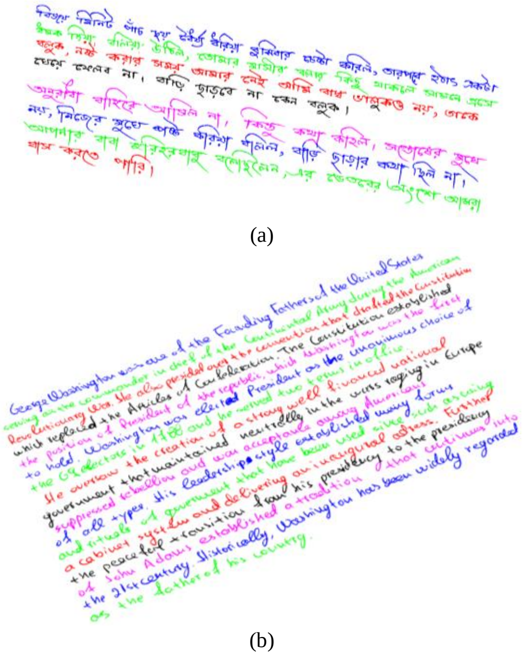


Fig. 19. The text line segmentation results of manually skewed documents by (a) -15 and (b) +20 degrees

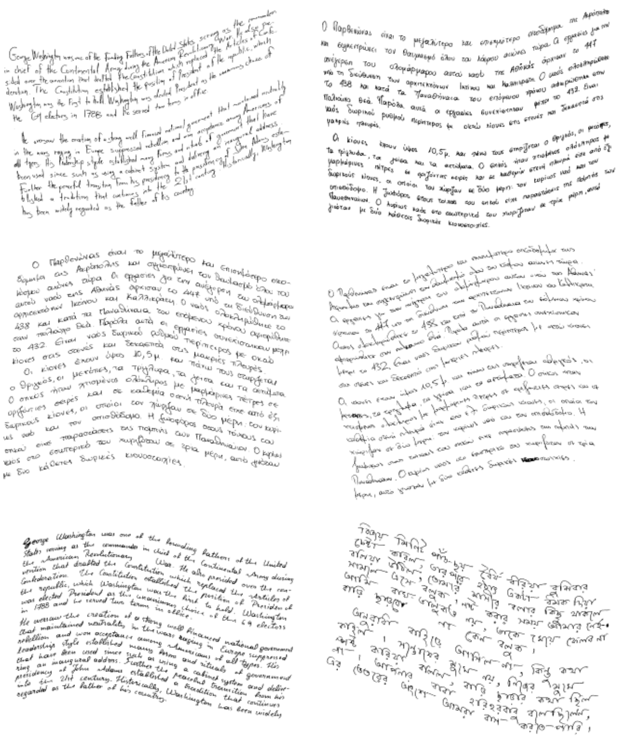


Fig. 20. Some manually distorted images of ICDAR2013 documents

A combined text region segmentation and a small missing text region detection algorithms were proposed to improve the text region detection accuracy. Moreover, the proposed CC assignment method decided about the CCs which had junctions with text lines separator boundaries. All the parameters of our method were tuned based on a global parameter which was calculated for each document image, separately. Experimental results showed that the proposed method outperformed all the previous state-of-the-art methods. The good results over three standard datasets including documents in different languages and with wide varieties in writing styles demonstrated that our method is a script and dataset independent method. In addition, the results on manually skewed and distorted documents (Fig. 18 and Table 5) proved that the proposed method can accurately segment text lines of complicated documents.

REFERENCES

[1] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1162–1173, 1993.

[2] A. Zahour, L. Likforman-Sulem, W. Boussalaa, and B. Taconet, "Text Line segmentation of historical Arabic documents," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2007, vol. 1, pp. 138–142.

[3] M. Arivazhagan, "A statistical approach to line segmentation in handwritten documents," *Doc. Recognit. Retr. XIV, Proc. SPIE, San Jose, CA, USA*, vol. 6500, p. 65000T–1–11, 2007.

[4] 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.

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

[6] B. B. Chaudhuri and S. Bera, "Handwritten text line identification in Indian scripts," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009, pp. 636–640.

[7] A. Zohrevand, J. Sadri, Z. Imani, and M. R. Yeganezad, "Line Segmentation in Persian Handwritten Documents Based on a Novel Projection Histogram Method," in *International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 2019, pp. 70–74.

[8] M. Ziaratban and K. Faez, "Adaptive script-independent text line xtraction," *IEICE Trans. Inf. Syst.*, vol. E94–D, no. 4, pp. 866–877, 2011.

[9] E. Kiumarsi and A. Alaei, "A Hybrid Method for Text Line Extraction in Handwritten Document Images," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 241–246.

[10] 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.

[11] 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.

[12] G. Louloudis, B. Gatos, I. Pratikakis, and K. Halatsis, "A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents," *Proc. of the Tenth IWFHR*, no. i, pp. 515–520, 2006.

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

[14] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of Historical Handwritten Documents into Text Zones and Text Lines," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2014–Decem, pp. 464–469, 2014.

[15] Z. Shi and V. G. V. Govindaraju, "Line separation for complex document images using fuzzy runlength," *First Int. Work. Doc. Image Anal. Libr. 2004. Proceedings.*, 2004.

[16] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten Arabic text lines," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009, pp. 176–180.

[17] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Script-independent handwritten textlines segmentation using active contours," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009, pp. 446–450.

[18] S. S. Maddouri, F. Ghazouani, and F. B. Samoud, "Text lines and PAWs segmentation of handwritten Arabic document by two hybrid methods," *2014 1st Int. Conf. Adv. Technol. Signal Image Process. ATSIP 2014*, pp. 310–315, 2014.

[19] A. A. Shahraki, A. E. Ghahnavieh, and S. A. Mirmahdavi, "A Morphological Approach to Persian Handwritten Text Line Segmentation," *2014 UKSim-AMSS 16th Int. Conf. Comput. Model. Simul.*, pp. 298–301, 2014.

[20] A. Alaei, U. Pal, and P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," *Pattern Recognit.*, vol. 44, no. 4, pp. 917–928, 2011.

[21] L. Wang, W. Fan, J. Sun, and S. Uchida, "Globally Optimal Text Line Extraction based on K- Shortest Paths algorithm," *2016 12th IAPR Work. Doc. Anal. Syst.*, pp. 335–339, 2016.

[22] K. Chen, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Robust text line segmentation for historical manuscript images

using color and texture," in *Proceedings - International Conference on Pattern Recognition*, 2014, pp. 2978–2983.

[23] B. Moysset, C. Kermorvant, C. Wolf, and J. Louradour, "Paragraph text segmentation into lines with Recurrent Neural Networks," *13th Int. Confrence Doc. Anal. Recognit. - ICDAR'15*, pp. 456–460, 2015.

[24] B. Barakat, A. Droby, M. Kassis, and J. El-sana, "Text Line Segmentation for Challenging Handwritten Document Images Using Fully Convolutional Network," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 374–379.

[25] Q. N. Vo, S. H. Kim, H. J. Yang, and G. S. Lee, "Text line segmentation using a fully convolutional network in handwritten document images," *IET Image Process. Res.*, vol. 12, no. 3, pp. 438–446, 2018.

[26] B. Seraogi *et al.*, "Employing CNN to Identify Noisy Documents Thereafter Accomplishing Text Line Segmentation," in *Proceedings of TENCON*, 2018, no. October, pp. 28–31.

[27] P. Schone, C. Hargraves, J. Morrey, R. Day, and M. Jacox, "Neural Text Line Segmentation of Multilingual Print and Handwriting with Recognition-Based Evaluation," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 265–272.

[28] I. Setitra and A. M. X, "Angle Minimization and Graph Analysis for text line segmentation in handwritten documents," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 453–458.

[29] V. Khare *et al.*, "Weighted-Gradient Features for Handwritten Line Segmentation," in *International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3651–3656.

[30] D. Aldavert and M. Rusinol, "Manuscript Text Line Detection and Segmentation using Second-Order Derivatives," in *IAPR International Workshop on Document Analysis Systems*, 2018, pp. 293–298.

[31] V. Bosch, V. Romero, A. H. Toselli, and E. Vidal, "Text Line Extraction Based on Distance Map Features and Dynamic Programming," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 357–362.

[32] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR 2009 Handwriting Segmentation Contest," 2009, pp. 1393–1397.

[33] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 1402–1406, 2013.

[34] M. Ziaratban and F. Bagheri, "Extracting local reliable text regions to segment complex handwritten textlines," in *Iranian Conference on Machine Vision and Image Processing, MVIP*, 2013, pp. 70–74.

[35] T. Su, T. Zhang, and D. Guan, "Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text," *Int. J. Doc. Anal. Recognit.*, vol. 10, no. 1, pp. 27–38, 2007.

[36] J. Ryu, H. Il Koo, and N. I. Cho, "Language-independent text-line extraction algorithm for handwritten documents," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1115–1119, 2014.

[37] 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.

[38] H. Il Koo and N. I. Cho, "Text-line extraction in handwritten chinese documents based on an energy minimization framework," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1169–1175, 2012.

[39] A. Nicolaou and B. Gatos, "Handwritten text line segmentation by shredding text into its lines," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009, pp. 626–630.

[40] A. Lemaitre, J. Camillerapp, and B. Couasnon, "Handwritten text segmentation using blurred image," *DRR - Doc. Recognit. Retr. XXI*, 2014.

[41] E. Kavallieratou, N. Dromazou, N. Fakotakis, and G. Kokkinakis, "An integrated system for handwritten document image processing," *Int'l J. Pattern Recognit. AI*, vol. 17, pp. 617–636, 2003.

[42] J. S. Cardoso, A. Capela, A. Rebelo, and C. Guedes, "A connected path approach for staff detection on a music score," in *Proceedings - International Conference on Image Processing, ICIP*, 2008, pp. 1005–1008.

[43] M. Diem, F. Kleber, and R. Sablatnig, "Text line detection for heterogeneous documents," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2013, pp. 743–747.

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

[45] 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.

مجله علمی پژوهشی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۹، شماره ۱، بهار ۱۳۹۹، صفحه ۴۶-۶۰
دریافت: ۱۳۹۸/۰۱/۱۸، بازنگری: ۱۳۹۸/۱۰/۲۹، پذیرش: ۱۳۹۹/۰۲/۲۳

جداسازی سطرهای دستنویس مستقل از زبان نوشتار با استفاده از فیلترهای جهتی دوبُعدی

مجید زیارت‌بان
گروه مهندسی برق، دانشکده مهندسی، دانشگاه گلستان،گرگان، ایران.
m.ziaratban@gu.ac.ir

* نشانی نویسنده مسئول: مجید زیارت‌بان، گرگان، خیابان شهید بهشتی، دانشگاه گلستان، دانشکده مهندسی، کد پستی: ۱۵۷۵۹-۴۹۱۳۸

چکیده- جداسازی سطرها بخش مهمی از الگوریتم‌های بازشناسی متن است. به منظور تحلیل و بازشناسی یک متن، ابتدا باید سطرهای نوشته شده از یکدیگر جدا شوند. جداسازی سطرها در متن‌های دستنویس بسیار سخت‌تر از متن‌های تایپی است. سطرهای منحنی شکل و با زاویه‌های مختلف نسبت به محور افقی، سطرهای دارای همپوشانی و سطرهای بسیار کوتاه مهمترین چالش‌ها در مبحث جداسازی صحیح سطرهای دستنویس هستند. اغلب روش‌های موجود خصوصیات محلی سطرها در یک متن را در نظر نمی‌گیرند. در روش پیشنهادی، هم خصوصیات کلی و هم محلی در نظر گرفته می‌شوند. روش پیشنهاد شده مبتنی بر استفاده از فیلترهای دو بُعدی گوسی است. کلیه پارامترهای روش پیشنهادی بر اساس یک متغیر کلی بدست می‌آیند که این پارامتر کلی برای هر متن بصورت جداگانه محاسبه می‌شود. بنابراین روش پیشنهادی مستقل از مجموعه داده‌ها است. یک متن ورودی ابتدا به چندین بلاک تقسیم می‌شود که برای هر بلاک، ویژگی‌های محلی استخراج می‌شود. در هر بلاک، نواحی متنی با استفاده از ویژگی‌های محلی از جمله میزان چرخش در بلاک، شناسایی می‌شوند. به منظور تخمین صحیح زاویه چرخش نواحی متنی در یک بلاک، یک الگوریتم تخمین زاویه در این مقاله پیشنهاد می‌شود. نتایج آزمایش‌ها نشان می‌دهند که روش پیشنهادی از تمامی روش‌های خوب دیگر و بر روی سه مجموعه داده استاندارد دقت بهتری ارائه می‌کند. نتایج بدست آمده برای روش پیشنهادی در مجموعه داده های ICDAR2013 ، ICDAR09 وHIT-MW ، به ترتیب به میزان ۰/۵۴، ۰/۰۳ و ۰/۰۲ درصد نسبت به نتایج برنده مسابقه جداسازی سطرها در ICDAR2013 بهبود داشته است. همچنین آزمایش‌ها نشان می‌دهند که روش پیشنهادی می‌تواند سطرهای متن‌های دستنویس با میزان پیچیدگی زیاد را نیز با دقت کافی از هم جدا نماید.

واژه‌های کلیدی: جداسازی سطرها، متن دستنویس، روش مستقل از زبان نوشتاری، فیلترهای جهتی دوبُعدی.