Wireless Networks (2021) 27:3483–3495
https://doi.org/10.1007/s11276-019-02221-1

An adaptive threshold algorithm for offline Uyghur handwritten text line segmentation

Eliyas Suleyman[1] · Askar Hamdulla[2] · Palidan Tuerxun[1] · Kamil Moydin[2]

Published online: 2 January 2020

Abstract
This paper presents an effective text-line segmentation algorithm and evaluates its performance on Uyghur handwritten text document images. Projection based adaptive threshold selection mechanism is implemented to detect and segment the text lines with different valued thresholds. The robustness of the proposed algorithm is admirable that experiments on 210 Uyghur handwritten document image including 2570 text lines got correct segmentation by 97.70% precision and 99.01% recall rate and outperformed the compared classic text-line segmentation algorithm on same evaluation set. Additionally, the proposed algorithm is tested on the public handwriting dataset and get 98.05% correct segmentation rate which is robust and promising.

Keywords Text line segmentation · Adaptive thresholding · Offline Uyghur handwritten documents

1 Introduction

With the prevalence of computers and scanners, tremendous books and handwritings of its copies are being digitally available. In order to make these document can be accessed easily, various techniques is utilized and some of it are already playing major role in commercial application. Text line segmentation is significant stage of offline handwritten document recognition and analysis [1]. Correctness of segmented text lines would influence the process and result of subsequent stages directly [2]. Text-line segmentation on document images of printed texts is easily handled by using simple projection method and a statistically estimated threshold. However, it is not a promising way to segment handwritten document images [3–5]. Unlike machine printed documents [6], due to high diversity in writing habits of different writers, distances within text lines are irregular and existence of touching and overlapping text lines makes this work challenging.

✉ Askar Hamdulla
askar@xju.edu.cn

[1] School of Software, Xinjiang University, Urumqi 830046, People's Republic of China

[2] Institute of Information Science and Engineering, Xinjiang University, Urumqi 830046, People's Republic of China

Modern Uyghur script is an alphabetic script which has 32 basic characters and it is written from right to left [7]. Almost each letter has several special ascenders or descenders which distinguish them from similar letter forms. Due to the cursive nature of Uyghur script, the special symbol may appear connected, overlapped not only in a word and text-line, but also between neighboring text-lines, as well. This makes text line segmentation more difficult than printed texts or other scripts of isolated styles.

Traditional projection-based text-line segmentation method uses a confirmed constant threshold to separate different and neighboring text lines [7]. It is suitable for machine printed text images due to equal or regular spatial distance between neighboring text lines. Yet, its effectiveness is not acceptable for handwritten documents.

In this paper, we propose a novel approach for text line segmentation based on projection and adaptive thresholding mechanism. The proposed method has proven its effectiveness and robustness during the experiments on handwritten text images of text-lines with different styles, lengths, skewing and touching degrees. Rest of the paper is organized as follows: some previous works are recalled in Sect. 2. In Sect. 3, the proposed method is described in detail. Discussion on the conducted experiments and evaluation methods are given in Sect. 4. Section 5 draws brief conclusion then.

2 Related work

In 2006, Li Et Al proposed an approach based on smearing [8]. They first convert a binary image to gray scale image using a Gaussian window. Then, text lines are extracted by evolving an initial estimate using level set method [2]. The algorithm correctly detected 85.6% of 2691 ground-truth text lines. The segmentation error caused by adjacent text lines and over-lapping text line makes this algorithm less compatible.

In 2009, Vassilis Papavassiliou Et Al proposed an algorithm based on the piece-wise projection [9]. The algorithm is tested on the benchmarking datasets of IDCAR07 handwriting segmentation contest, correct rate of the segmentation reached 95.67%. Although the segmentation is mostly correct, over-segmentation is occurred.

In Bal and Saha [10] proposed a text line segmentation algorithm based on projection. All Rising section in the projection is measured and the average value of rising section is treated as threshold. The algorithm is tested on the IAM database which contains more than 550 text images which has different writer. This approach correctly segmented 95.65% text lines. Due to the chosen threshold is constant, it is not adaptable for various handwritten document and it is not able to segment severely sloped text line.

In Ptak et al. [11] proposes an algorithm based on projection with a variable threshold. This method can segment handwritten text lines which text lines are in similar length. However, performance of segmentation declines when text lines are short or touched. The author tested the algorithm on their own collected Polish document images, which contains similar length text lines document and random length text lines.

In this paper, a projection based adaptive threshold algorithm for text line segmentation is proposed.

3 Methodology

3.1 Framework

The first-hand collected Uyghur handwritten text samples are preprocessed using common preprocessing techniques including turning the original image to the gray scale image, dilation, binarization and noise removal [12]. After the document image is preprocessed, horizontal projection of preprocessed image is calculated, and thresholding is performed according to projection peaks and its locations. After measuring threshold, each text line is segmented according to each previously determined threshold and the line separators are drawn at the valley point, which is determined according to horizontal projection profile, of each neighbor text lines in the original image. The major steps of proposed algorithm are shown in Fig. 1.

3.2 Preprocessing

Preprocessing technique aims to eliminate and minimize harmful or insignificant content and enhance useful features in images, especially for document images [13]. Thus, it improves generality of sample representation and performance of subsequent works. Before the proposed text-line segmentation method is applied, preprocessing is performed using turning the original image (color image) to the gray scale image, dilation, noise removal and binarization which is used twice.

3.2.1 Gray scaling

In order to calculate a projection profile, original document image should be turned to binary image, thus, gray scaling is performed before binarization. Therefore weighted sum method is used to conduct gray scaling. Commonly, a color image contains three channels, each channel stores the 2-dimentional array which represents red, green and blue [14]. The gray scale image is determined by calculating a weighted sum of three channels components for every pixel of color image. Therefore, three dimensional tensor became two dimensional array that stores the result of calculation which is the final gray scale image.

$$S = 0.2989 \times R + 0.587 \times G + 0.1140 \times B \tag{1}$$

3.2.2 Dilation

Dilation is one of the basic operations in mathematical morphology [15]. The dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image [16]. The dilation of A by B is defined by:

$$A \oplus B = \bigcup_{b \in B} A_b \tag{2}$$

where A_b is the translation of A by B.

The dilation process is highly dependent on its structuring element [17]. If it is not suitable for particular situation in image, dilation process may cause unpromising result which is different from expectations [18]. Thus, the structuring element must be defined properly. The dilation kernel used in this work is shown in Fig. 2. By using the kernel shown below, representation of text in document image became conspicuous.

In this paper, dilation is used to thicken the texture of text in document image and keep the main area of the text,

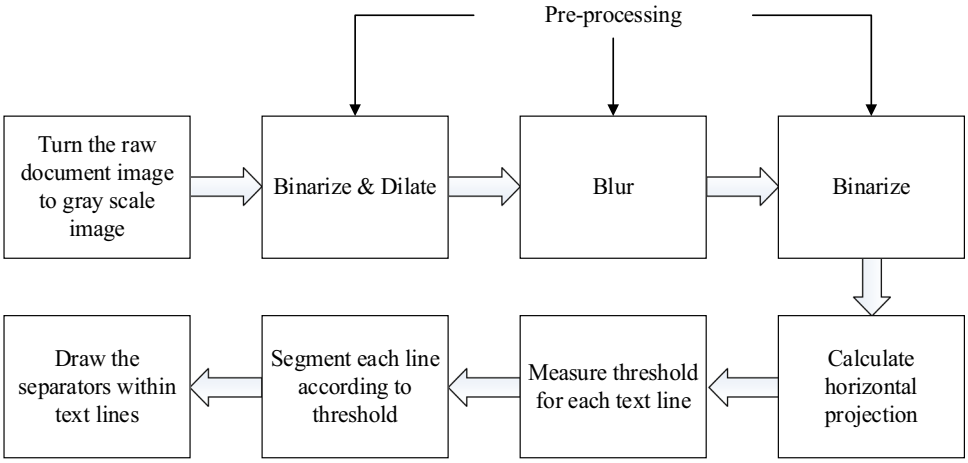


Fig. 1 Major steps of proposed algorithm

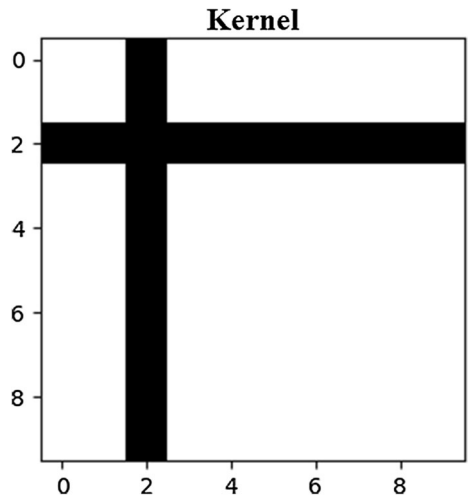


Fig. 2 Dilation kernel

which allows the proposed algorithm easier to extract vital information like peak points and valley points, and distinguish each text line. Fig 3 compares binary image of a text document and its dilation effect.

3.2.3 Noise removal

Noise removal is important to any kind of image processing task [19], especially for handwritten document images [20]. Generally, scanned handwritten document image contains some kind of noisy points which caused by dirt or during the scanning process. These points are harmful for the entire process of algorithm. Since binarized image is dilated, consequently, noisy points are also became bigger that could affect subsequent work. Filtering is a prevalent way to minimize or remove the noise in images. Each filter commonly contains a corresponding window. With the expansion of window size, result of filter would be vaguer

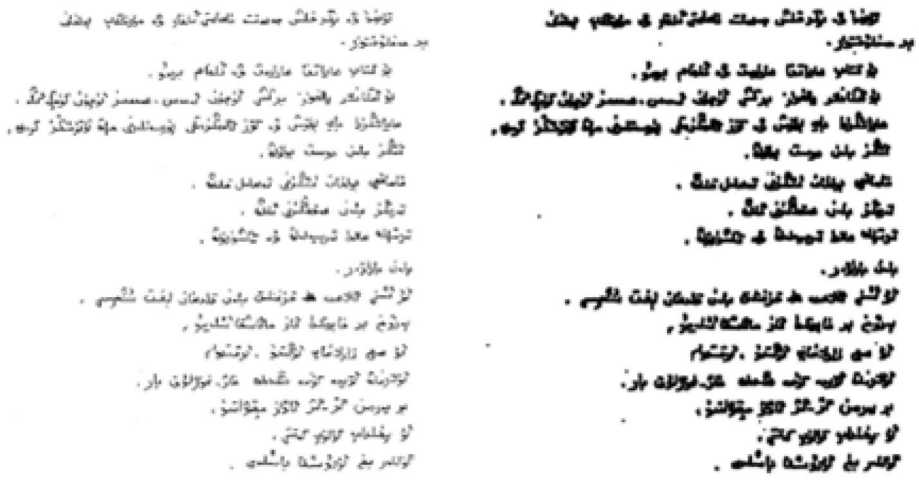


Fig. 3 Before and after dilation

(a) Original image **(b)** Dilated image

[21]. This means window size must be chosen appropriately; otherwise, the document image will lose important information in the process of filtering. In this paper, we use mean filter to perform noise removal. Mean filtering is a simple, intuitive and easy to implement method of smoothing images i.e. reducing the amount of intensity variation between one pixel and the next [22]. Thus, noisy points in blank area in document image can be weakened or eliminated. For every pixel in image, the filter would calculate average value of corresponding window and replace the original value to the calculated one.

$$O(x,y) = \frac{1}{mn} \sum_{(s,t)\in S} I(s,t) \tag{3}$$

Besides, we also used mean filter to minimize the local extrema (minima and maxima points) in projection profile which is calculated after whole preprocessing technique is done. Some different blurring parameters are tested to observe their blurring effects, setting window size to 30 by 30 pixels gave the best blurring effect and is selected as blurring parameter in later experiments. Handwritten document image after smoothing by different window sizes parameters are compared in Fig. 4.

3.2.4 Binarization

Document image binarization is a crucial phase which is able to segment the text and the background by eliminating remaining unimportant information [23]. The histogram of original gray image and image blurred by 30*30 window is shown in Fig. 5. As the second histogram shows, after the document image is blurred, gray level in each black pixel is reduced [24]. Moreover, the gray level of pixels, which is near to black ones, are increased. This means that threshold of binarization should be chose correctly.

Thus, Otsu thresholding method is used for image binarization [25].

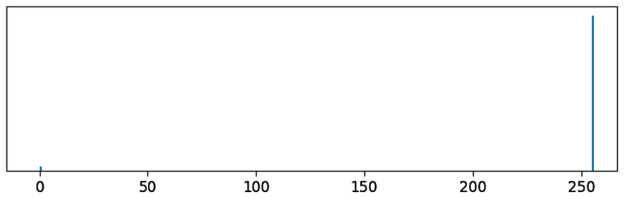
(a) blurred by 10*10 window

(c) blurred by 30*30 window

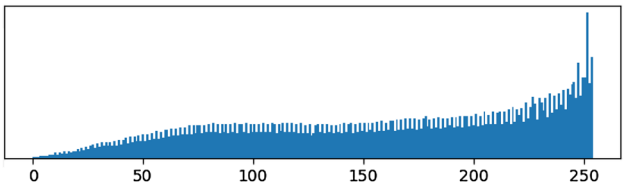
(b) blurred by 20*20 window

(d) blurred by 40*40 window

Fig. 4 Differently blurred images



(a) Histogram of original gray image



(b) Histogram of blurred gray image

Fig. 5 Histogram of document image

$$\sigma_{\omega}^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (4)$$

Weights ω_0 and ω_1 are probabilities of two classes, which refers text lines and the background or black pixel and white pixel, separated by a threshold t, and σ_0^2 and σ_1^2 and variance of these two classes [26].

In this work, binarization also enhances the generality of the text lines in our document image. Four images, which are differently blurred, after binarization effect are shown in Fig. 6.

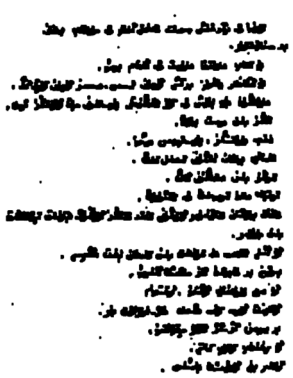
As the binary image shows, some noisy points are removed, text area in document image became smoother than original image. This is very conducive to compute a smooth projection profile. The projection after binarization on each differently blurred images are shown in Fig. 7.

3.3 Text line segmentation

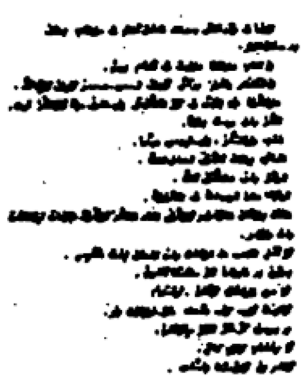
Widely acknowledged text line segmentation method based on projection calculates the average gap between successive text lines, then define a constant threshold to separate these text lines [27, 28]. However, when threshold is constant, touched or near text lines might be omitted. Therefore, the process of defining threshold must be adaptive to different gaps between each neighbor text line couples.

In this work, after calculating horizontal projection profile H from the preprocessed image, significant peaks' location which might represent each potential text lines are

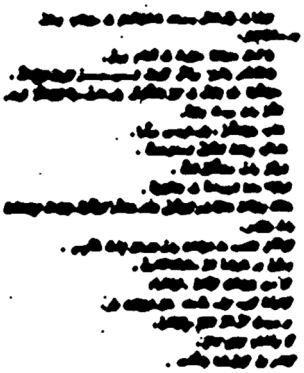
Fig. 6 Binarization effect



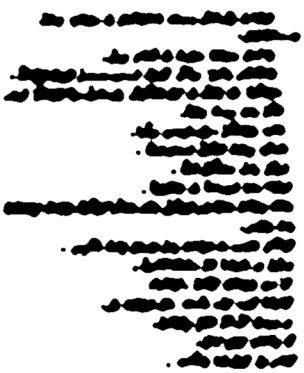
(a) Binarization of image blurred by 10*10 window



(b) Binarization of image blurred by 20*20 window



(c) Binarization of image blurred by 30*30 window



(d) Binarization of image blurred by 40*40 window

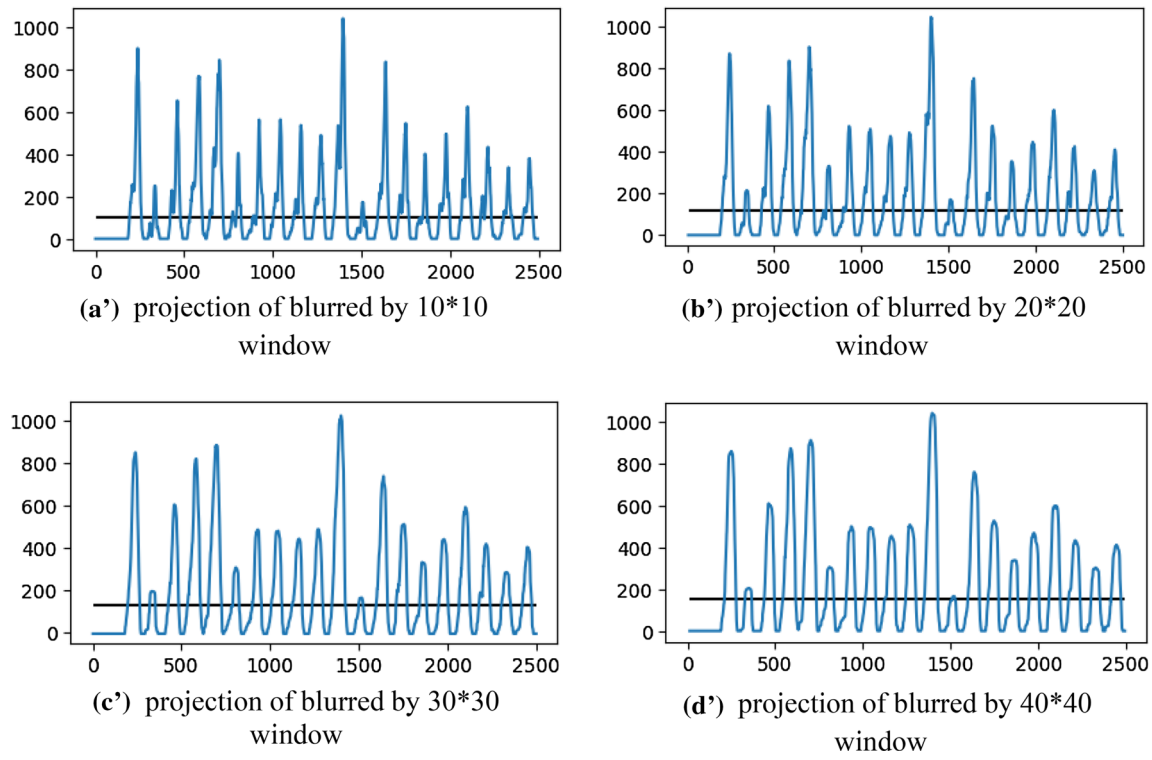


Fig. 7 Projections after binarization

extracted to set P and P'. Next, thresholding is performed as follows: visit each element $P(i)$ in set P; for given $P(i)$, take the half of the peak value and give it to threshold T. In general, visit each peak's location, then get its value and take the half of it and treat it as threshold.

$$T_p = P(i) \cdot \frac{1}{2} \tag{5}$$

Since each threshold is differently measured form peaks of horizontal projection values, the threshold will have different values for each neighbor text lines. After measuring each threshold, the projection values are visited reversely from the current peak location. If the currently visited projection value is less than threshold, then the location of this projection value is assumed as starting point of a text line and added to set S and break the loop. Then, the ending points are determined same way using forward visiting of projection values and the ending point is added to set E, correspondingly. However, these intervals, which composed by starting points and ending points, are not totally reliable for determine each potential text line. Therefore, interval inspection is performed to remove the intervals that do not represent text line. The pre-estimated text-line intervals and tip points (starting, ending) are checked to confirm their validity and correctness by the following algorithm.

First, visit each element in set S and set E, for given start point $S(i)$ and end point $E(i)$, to calculate midpoint M_i of each interval using equation below;

$$M_i = \frac{S(i) + E(i)}{2} \tag{6}$$

Second, get next interval's start point $S(i+1)$, if it is greater or equal to M_i, the algorithm see these two intervals as true intervals, which means they are not overlapped with each other, then accept it as a true interval, otherwise it is seen as false interval (7) and it will be added to the previous interval, which is the process of combination of two intervals. This process makes the performance of interval selection more acceptable.

$$\begin{cases} S(i+1) \geq M_i & true \\ S(i+1) < M_i & false \end{cases} \tag{7}$$

After modifying set S and E straight lines are drawn to separate the text-lines in the document image. The separator lines are drawn horizontally at valley points, which is in the horizontal projection profile, between two adjacent estimated text-line positions.

In the respect of computation complexity, firstly, the projection calculation is depend on the height and width (rows and columns) of document image. Then, due to every peak is extracted by a projection vector (one dimensional array), thus, peak extraction stage is linear. Moreover, line

drawing is also linear. Thus, final equation of time complexity is:

$$O(n) = rc(w * h) \tag{8}$$

where r and c refers to row and column of binarized document image, where w and h refers to the width and height of the filter.

3.4 Algorithm

The pseudo code of proposed algorithm is shown in Table 1.

Step 1: Read a handwritten document image as a multi-dimensional array;
Step 2: Convert the raw image to gray scale image and binarize the gray image;
Step 3: Dilate the binarized image;
Step 4: Blur the dilated image;
Step 5: Binarize the blurred image;
Step 6: Calculate the horizontal projection profile of binarized image;
Step 7: Add peaks, which is above the mean value of projection, to set P and their locations are stored into set P'.
Step 8: For each element in set P, calculate the threshold by taking half of the peak value. Visit the elements of projection vector from currently visiting peak's location forwardly and reversely to determine ending point and starting point, respectively. Where projection value is less than threshold is measured as starting point or ending point and the location of these are added to set S and set E.
Step 9: For each interval, calculate the mid point M_i. Compare it with next interval's start point. If it is greater or equal to M_i, accept it as a true interval. Otherwise it is seen as false interval and it will be added to the previous interval
Step 10: Draw a straight line at the valley point between two adjacent intervals according to HPP.
Step 11: End.

Table 1 Pseudo code of algorithm

Text line Segmentation Algorithm
1. Count the horizontal projection profile $\boldsymbol{H}$ preprocessed image $\boldsymbol{I_p}$
2. $\boldsymbol{M} \leftarrow \frac{1}{\boldsymbol{H.length}()} \sum_{\boldsymbol{n}=1}^{\boldsymbol{H.length}()} \boldsymbol{H}(\boldsymbol{n})$
3. $\boldsymbol{P} \leftarrow \emptyset, \boldsymbol{S} \leftarrow \emptyset, \boldsymbol{E} \leftarrow \emptyset, \boldsymbol{i} \leftarrow 1, \boldsymbol{j} \leftarrow 1, \boldsymbol{k} \leftarrow 1, \boldsymbol{entrance} \leftarrow \boldsymbol{false}, \boldsymbol{t_r} \leftarrow 0, \boldsymbol{b} \leftarrow 0, \boldsymbol{mid} \leftarrow 0$
4. for each row in $\boldsymbol{H}$
5. if $\boldsymbol{H}(\boldsymbol{i}) > M$ and *entrance* is *false* then
6. $entrance \leftarrow falce$
7. if $\boldsymbol{H}(\boldsymbol{i}) > b$ then
8. $\boldsymbol{b} \leftarrow \boldsymbol{H}(\boldsymbol{i})$
9. if $\boldsymbol{H}(\boldsymbol{i}) < M$ and *entrance* is *true* then
10. $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \boldsymbol{b}$
11. $entrance \leftarrow false$
12. while $\boldsymbol{P.length}()$ is not 0 do
13. $\boldsymbol{t_r} \leftarrow \frac{\boldsymbol{P}(1)}{2}$
14. for each corresponding row of $\boldsymbol{P}(1)$ to first row do
15. if $\boldsymbol{H}(\boldsymbol{j}) < \boldsymbol{t_r}$ then
16. $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup$ corresponding row of $\boldsymbol{H}(\boldsymbol{j})$, break
17. for each corresponding row of $\boldsymbol{P}(1)$ to first row do
18. if $\boldsymbol{H}(\boldsymbol{j}) > \boldsymbol{t_r}$ then
19. $\boldsymbol{E} \leftarrow \boldsymbol{E} \cup$ corresponding row of $\boldsymbol{H}(\boldsymbol{j})$, break
20. Delete $\boldsymbol{P}(1)$ from $\boldsymbol{P}$
21. while $\boldsymbol{k} \neq \boldsymbol{S.length}()$ do
22. $\boldsymbol{mid} \leftarrow \frac{\boldsymbol{S}(\boldsymbol{k}) + \boldsymbol{E}(\boldsymbol{k})}{2}$
23. if $\boldsymbol{mid} > S(k+1)$ then
24. Delete $\boldsymbol{S}(\boldsymbol{k})$ from $\boldsymbol{S}$ and $\boldsymbol{E}(\boldsymbol{k}+1)$ from $\boldsymbol{E}$

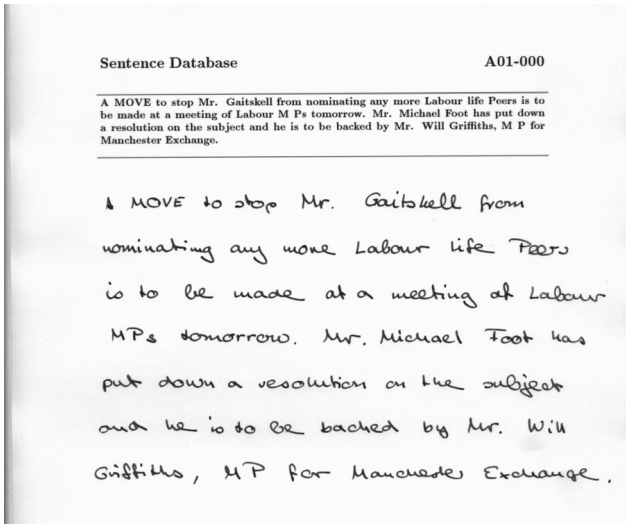
4 Experimental result

4.1 Database

To verify the proposed algorithm, we collected 210 Uyghur handwritten document images including 2570 text lines. The collected handwritten documents are written by different writers that each document varies in length and handwriting styles. The handwriting styles in the established database are broadly categorized into three types: (1) neatly written text-lines with random lengths; (2) similar length of text-lines in casual style that contain many overlapping and ligatures; (3) skewed normal handwriting. Fig 8 shows some typical examples of the mentioned handwriting styles in the database. Each document image is separately stored in TIF format. The pixel intensity of the samples also varies between 1477 × 944 to 2175 × 2277.

Additionally, we also collected the Polish handwritten document images from website that is given by Ptak et al. [11]. Dataset include 29 pairs of Polish document image which has 58 images in total. They generally put these document images into two different classes which are documents that contains short length of text lines and documents that almost has equal length of text lines. In the database, each document is stored as pair. Each pair has random length text line version and mostly identical length text line version. In the document, writing style is divergent from image to image. Some are very neatly written, but severely sloped, which is multidirectional. Some are not sloped but written in extremely casual style. Thus, running test on this data set is also able to evaluate the performance of proposed algorithm due to the dataset's challenging feature.

Finally, the proposed algorithm is also tested on the public offline handwriting dataset, the IAM dataset [29], to evaluate its performance. It includes unconstrained handwritten text, which were scanned at a resolution of 300 dpi and saved as PNG images with 256 gray levels. The sample of IAM database is illustrated in Fig. 9.


Sentence Database A01-000

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, M P for Manchester Exchange.

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour MPs tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, MP for Manchester Exchange.

Fig. 9 Sample of IAM English handwriting dataset

4.2 Evaluation method

In this paper, we calculated precision, recall and the F-measure to evaluate the performance of proposed algorithm [30]. Precision is based on manually counting the total segmented text lines and correctly segmented text lines, recall is based on counting the total text lines and the correctly segmented text lines. Then, the F-measure is calculated according to precision and recall.

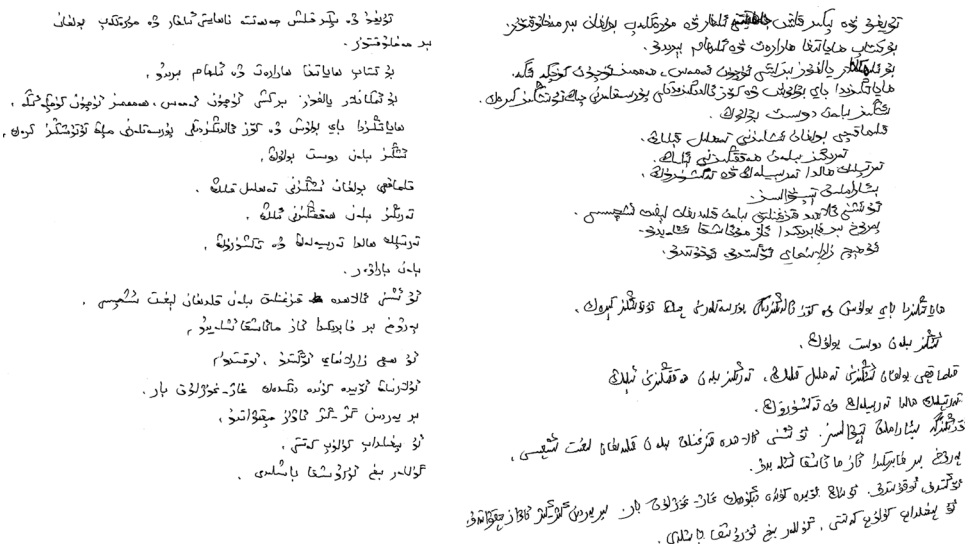


Fig. 8 Three samples of database

$$P = \frac{L_c}{L_s} \tag{9}$$

$$R = \frac{L_c}{L_t} \tag{10}$$

$$F = \frac{2PR}{P + R} \tag{11}$$

where L_c and L_s denote the correctly segmented text lines and total segmented text lines, respectively. Where L_t refers to the total lines in document image.

4.3 Result and analysis

Several algorithms including projection based are tested on introduced datasets to compare with proposed algorithm. Brief introduction of algorithms and its segmentation mechanism is depicted below.

There are three parameters is taken to the participant algorithm which is the input image, windows size of filter and the relative threshold. The optimum values of parameters are given that the window size takes 9 and the relative threshold takes 0.5. The experimental results of text-line segmentation on our dataset are shown in Fig. 10 and Table 2. For comparison, we evaluated the participant algorithm on our database.

In the participant algorithm [11], the Polish document image is preprocessed including turning the original image to gray scale image, binarization and noise reduction. Then

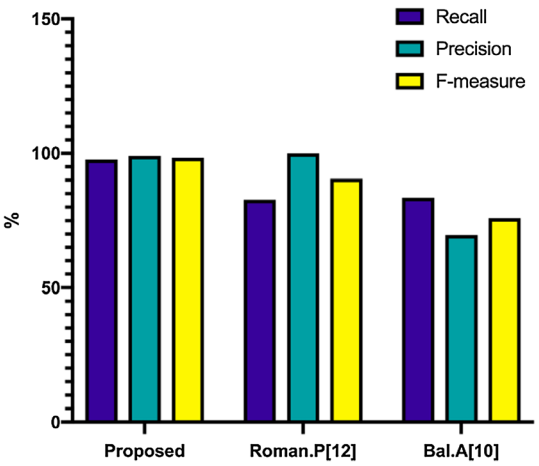


Fig. 10 Comparison of algorithms

Table 2 Result of experiments

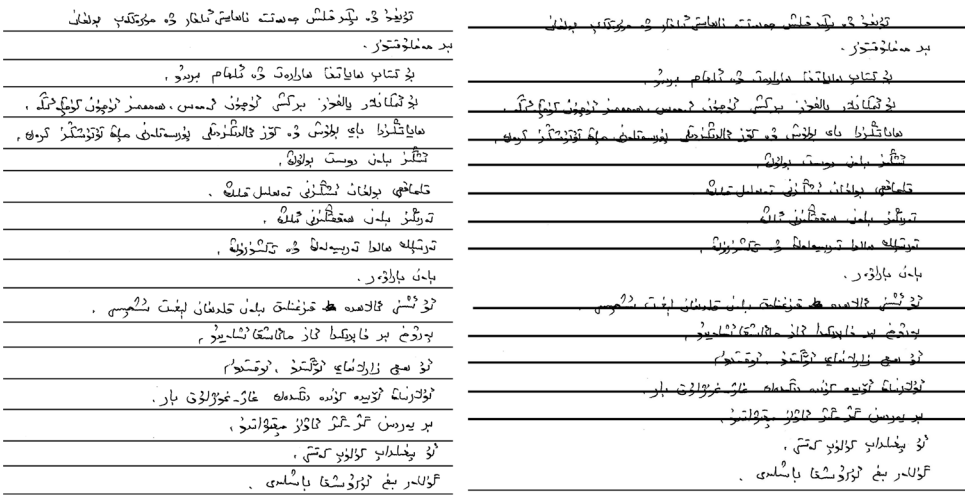
	Proposed (%)	Roman [11] (%)	Bal [10] (%)
Recall	97.70	82.75	83.54
Precision	99.01	99.99	69.53
F-measure	98.35	90.56	75.92

count the projection profile of preprocessed image and sort it with descending order. Then visit each value of sorted projection to determine the threshold. Each time the algorithm chooses a threshold, text lines would be segmented afterward. If the text lines are already segmented, the algorithm would continue to the next iteration. The algorithm stops when the current value of projection is less than 1/10 of maximum value of projection.

In contrast, our algorithm's preprocessing stage has one more step which is dilation. This guarantees the important features of text in document image not to be removed by the noise reduction process. In the respect of threshold measuring, we extract each location corresponding to the significant peaks to determine the threshold rather than sorting the entire projection profile. In text line extraction stage, our algorithm starts visiting from the location of a significant peak, terminates when algorithm find a starting point or an ending point of one interval, rather than visiting all values of projection. In the respect of checking extracted text lines whether it is correctly segmented, we conducted checking mechanism that is totally different from the participant algorithm. The participant algorithm simply just omits if the currently segmented text lines overlaps with intervals which is segmented previously, even it is not severely overlapped. In our checking mechanism, we consider each two adjacent intervals and observe the current interval's start point that whether it is greater than the next interval's midpoint.

According to results of the two segmentation algorithms in Table 2, proposed algorithm outperformed the participant algorithm in recall and F-measure. Although the precision of the participant algorithm is higher than the proposed algorithm, its recall rate is much lower than proposed algorithm. This means method [11] is not strong as the proposed algorithm in the respect of text-line detection. Segmentation precision of the participant algorithm is high for neatly styled text-lines, but it is observed not strong enough to detect sufficient text lines. Some text-line segmentation effects of two compared algorithm are illustrated in Fig. 11. In sample (a), which is neatly written handwriting sample, the participant algorithm is unable to detect and segment short text lines. Although the text lines in sample (b) is mostly similar in the respect of length, the casual writing style and skewed text lines affected the participant algorithm's accuracy. Even the participant algorithm detected one of the skewed text lines, the segmentation is incorrect. But our algorithm segments the all text lines in both sample properly.

Proposed algorithm and the algorithm [11] are also tested on the polish handwriting documents. In this experiment, proposed segmentation method still outperformed the compared method. However, the result of both algorithm is not promising due to testing dataset's feature



(a) Non-skewed handwritten sample

(b) Skewed handwritten sample

Fig. 11 result of two different algorithms

is very challenging and segmentation condition is extreme. The result shows that the proposed algorithm detected and segmented most text lines in this Polish document image. However, in proposed algorithm, same error occurred because text lines are skewed. Although our algorithm detected every text line in the image, the segmentation is not correct. Since skewed text lines affected the extraction of significant peaks of projection profile. Algorithm [11] is not sensitive to short text line and when it exist in document, the algorithm is not able to segment these text lines. Finally, the recall rate of proposed algorithm and algorithm [11] are 63.23% and 38.06%, respectively.

We tested several algorithm on our Uyghur documents and Table 3 is the result ofeach algorithm. It can be seen from the result that the proposed algorithm is alsobetter than other compared algorithms.

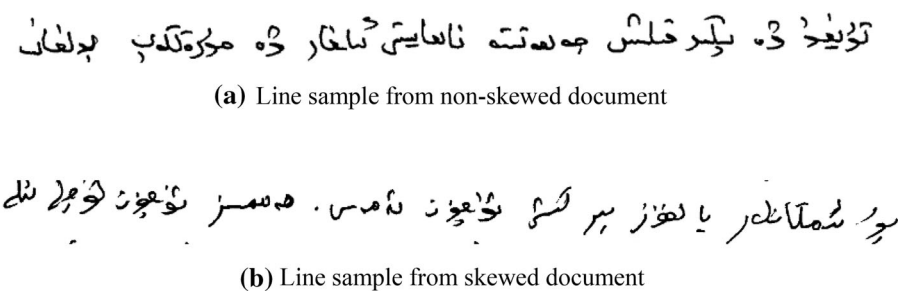
In addition, proposed algorithm is also tested on IAM public handwriting dataset. Theexperimental result shows that out method is also promising on public handwriting-dataset. As the Table 4 shows that proposed algorithm's

Table 4 Comparison of algorithms on IAM dataset

	Proposed (%)	Bal [10] (%)
Accuracy	98.05	95.65
Over	0.00	1.45
Under	1.95	2.9

Table 3 Comparison of algorithms on Uyghur dataset

Algorithm	Accuracy (Correctly segmented text lines) (%)
Simple projection	76.58
Projection with variable threshold	82.75
Piece-wise projection	88.74
Smearing-based	94.84
Coloring-based	96.33
Proposed	**97.70**



(a) Line sample from non-skewed document

(b) Line sample from skewed document

Fig. 12 result of two different algorithms

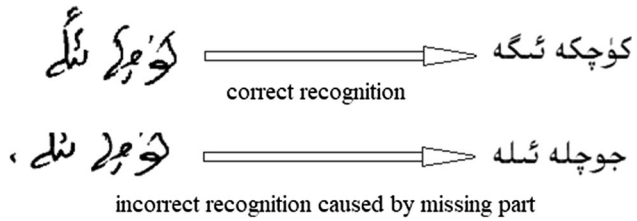


Fig. 13 result of two different algorithms

performance is also betterthan other recent approach using same dataset.

In the final stage of segmentation process, detected text lines will be separated from original image. At the same time, every separated text line will stored as individual line image. Some of segmented line images are shown in Fig. 12

As it can be seen from the separation results, sample A, which is written neatly and has significant gap between each text line, is separated easily with all of its contents and did not miss any significant information during the separation process. Thus, in the recognition stage [31], this will enhance the recognition accuracy by providing a whole text line. In contrast, due to skewness of some text lines in other type of document images, the separated line image lost some important information which includes part of words or characters even the line is accurately detected. In this scenario, handwriting recognizers would be affected directly and cause incorrect recognition. Consequence of this kind of segmentation is illustrated in Fig. 13.

5 Conclusion

This paper proposed a novel approach, which is not effected by the length of text lines in handwritten document, for off-line Uyghur handwritten text line segmentation using projection based adaptive threshold selection. The proposed algorithm is verified on 210 different Uyghur handwritten document images and 27 pairs of Polish document image, which is 58 images in total, including 1474 text lines. The experimental results shows robustness of the proposed text line segmentation algorithm. In our dataset, Recall rate of the proposed text-line segmentation algorithm is observed as 97.70% which is much higher than 82.35% recall of the compared algorithm. In Polish document dataset, the final recall rate of proposed algorithm is 63.23% which is twice as accurate as algorithm [11]. Finally, in the IAM public handwriting dataset, proposed algorithm is also better than the recent approach. The increase of segmentation rate means that the subsequent stages will be done in more reliable way. However, there are some disadvantages in proposed algorithm due to its simple projection-based mechanism. If the written direction of document is severely skewed, the performance of the proposed algorithm would decline or even unable to segment skewing text lines. Another factor that makes the performance of the algorithm decline is incorrect peak extraction from calculated projection profile, since the existence of overlapping text lines and nearly written neighboring text lines. To develop more comprehensive and general text-line segmentation algorithm, that is able to segment skewed text lines, is the main content of our next work.

Acknowledgements This work has been supported by the National Natural Science Foundation of China (under Grant of 61462080 and 61662076) and Ph.D. Scientific Research Startup Project of Xinjiang University.

References

1. Saabni, R., Asi, A., & El-Sana, J. (2014). Text line extraction for historical document images. *Pattern Recognition Letters, 35,* 23–33.
2. Razak, Z., Zulkiflee, K., Idris, M. Y. I., et al. (2008). Off-line handwriting text line segmentation: A review. *International Journal of Computer Science and Network Security, 7,* 12–20.
3. Yanikoglu, B., & Sandon, P. A. (1998). Segmentation of off-line cursive handwriting using Lunear programming. *Pattern Recognition, 31*(12), 1825–1833.
4. Sanchez, A., Suarez, P. D., & Mello, C. A. B., et al. (2008). Text line segmentation in images of handwritten historical documents. In *First workshops on image processing theory, tools and applications*, 2008. IPTA 2008. IEEE.
5. Basu, S., Chaudhuri, C., Kundu, M., et al. (2007). Text line extraction from multi-skewed handwritten documents. *Pattern Recognition, 40*(6), 1825–1839.

6. Saabni, R., & El-Sana, J. (2011). Language-independent text lines extraction using seam carving. In *2011 international conference on document analysis and recognition, ICDAR 2011*, Beijing, China. IEEE, 2011.
7. Abliz, A., Simayi, W., Moydin, K., & Hamdulla, A. (2016). A survey on methods for basic unit segmentation in off-line handwritten text recognition. *International Journal of Future Generation Communication and Networking, 9,* 137–152.
8. Li, Y., Zheng, Y., Doermann, D., et al. (2006). A new algorithm for detecting text line in handwritten documents. *Proc Iwfhr La Baule, 2,* 35–40.
9. Papavassiliou, V., Stafylakis, T., Katsouros, V., et al. (2010). Handwritten document image segmentation into text lines and words. *Pattern Recognition, 43*(1), 369–377.
10. Bal, A., & Saha, R. (2016). An improved method for handwritten document analysis using segmentation, baseline recognition and writing pressure detection. *Procedia Computer Science, 93,* 403–415.
11. Ptak, R., Żygadło, B., & Unold, O. (2017). Projection-based text line segmentation with a variable threshold. *International Journal of Applied Mathematics and Computer Science, 27*(1), 195–206.
12. Jiang, D., Li, W., & Lv, H. (2017). An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. *Neurocomputing, 220,* 160–169.
13. Jiang, D., Huo, L., & Song, H. (2018). Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. *IEEE Transactions on Network Science and Engineering, 1*(1), 1–12.
14. Huo, L., Jiang, D., Zhu, X., et al. (2019). An SDN-based fine-grained measurement and modeling approach to vehicular communication network traffic. *International Journal of Communication Systems, 5,* 1–12.
15. Jiang, D., Wang, W., Shi, L., et al. (2018). A compressive sensing-based approach toend-to-end network traffic reconstruction. *IEEE Transactions on NetworkScience and Engineering*, *5*(3), 1–12.
16. Huo, L., Jiang, D., & Lv, Z. (2017). Soft frequency reuse-based optimization algorithm for energy efficiency of multi-cell networks. *Computers and Electrical Engineering, 66,* 316–331.
17. Wang, F., Jiang, D., & Qi, S. (2019). An adaptive routing algorithm for integrated information networks. *China Communications, 7*(1), 196–207.
18. Lei, C., Jiang, D., Song, H., et al. (2018). A lightweight end-side user experience data collection system for quality evaluation of multimedia communications. *IEEE Access, 6*(99), 15408–15419.
19. Sun, M., Jiang, D., Song, H., et al. (2017). Statistical resolution limit analysis of two closely spaced signal sources using Rao test. *IEEE Access, 99,* 1.
20. Jiang, D., Huo, L., Lv, Z., et al. (2018). A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. *IEEE Transactions on Intelligent Transportation Systems, 10,* 3305–3319.
21. Jiang, D., Wang, W., Shi, L., et al. (2018). A compressive sensing-based approach to end-to-end network traffic reconstruction. *IEEE Transactions on Network Science and Engineering, 5*(3), 1–12.
22. Dingde, J., Liuwei, H., Ya, L., et al. (2018). Fine-granularity inference and estimations to network traffic for SDN. *PLoS ONE, 13*(5), e0194302.
23. Ntirogiannis, K., Gatos, B., & Pratikakis, I. (2014). A combined approach for the binarization of handwritten document images. *Pattern Recognition Letters, 35,* 3–15.
24. Wang, F., Jiang, D., Wen, H., et al. (2019). Adaboost-based security level classification of mobile intelligent terminals. *The Journal of Supercomputing, 75,* 7460–7478.
25. Ohtsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics, 9*(1), 62–66.
26. Huo, L., & Jiang, D. (2019). Stackelberg game-based energy-efficient resource allocation for 5G cellular networks. *Telecommunication Systems, 3,* 1–12.
27. Al-Dmour, A., & Zitar, R. A. (2016). Word extraction from Arabic handwritten documents based on statistical measures. *International Review on Computers and Software, 11*(5), 1–10.
28. Manmatha, R., & Rothfeder, J. L. (2005). A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1212–1225.
29. Marti, U., & Bunke, H. (2002). The IAM-database: An english sentence database for off-line handwriting recognition. *International Journal on Document Analysis and Recognition, 5,* 39–46.
30. Jiang, D., Zhang, P., Lv, Z., et al. (2016). Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. *IEEE Internet of Things Journal, 99,* 1.
31. Jiang, D., Wang, Y., Lv, Z., et al. (2019). Big data analysis-based network behavior insight of cellular networks for industry 4.0 applications. *IEEE Transactions on Industrial Informatics*. https://doi.org/10.1109/tii.2019.2930226.

Eliyas Suleyman received his undergraduate degree in software engineering from Xinjiang University. Now he is graduate student at Xinjiang University. His field of interests includes image processing and handwriting document analysis.

Askar Hamdulla is a professor in Institute of information science and engineering of the Xinjiang University. He received his Master's and Doctoral degree both in Communication and Information system in 1999 and 2003, respectively. Now he is associate dean of Institute of information science and engineering. His research interest focus on natural language processing, computer vision.

Palidan Tuerxun received her M.S. degree in 1996 from Liaoning University, China and her Ph.D. degree in 2015 from Northwestern University, China. Since 1992, she has been working as a teacher at Xinjiang University. Her research interests are machine learning and natural language processing.

Kamil Moydin received his B.E. and M.S. degree in radio electronics and computer science from Xinjiang University, China, and Osaka Institute of Technology, Japan in 1983 and 1998, respectively. He has been working as a teacher in School of Information Science and Engineering, Xinjiang University since 1983. His research interests include computer network, pattern recognition, and digital image processing.