An Adaptive Threshold Algorithm for Offline Uyghur Handwritten Text Line Segmentation

Eliyas Suleyman[1], Palidan Tuerxun[1], Kamil Moydin[2], and Askar Hamdulla[2(✉)]

[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
askar@xju.edu.cn

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.

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

1 Introduction

Text line segmentation is significant stage of offline handwritten document recognition and analysis. Correctness of segmented text lines would influence the process and result of subsequent stages directly [1]. 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 [2]. Unlike machine printed documents, 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.

Modern Uyghur script is an alphabetic script which has 32 basic characters, written from right to left [3]. 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 [10]. 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.

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 302–312, 2019.
https://doi.org/10.1007/978-3-030-32216-8_29

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 [4]. 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 [1]. 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, Papavassiliou et al. proposed an algorithm based on the piece-wise projection [6]. The algorithm 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 2016, Bal et al. proposed a text line segmentation algorithm based on projection [7]. 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. This approach correctly segmented 95.65% text lines. Due to the chosen threshold is constant, it is not adaptable for various handwritten document.

In 2017, Ptak et al. proposes an algorithm based on projection with a variable threshold [9]. 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. The testing result shows that the algorithm is not able to detect and segment the touched and short text lines in the Polish document.

In this paper, a projection based adaptive threshold estimation algorithm 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. After preprocessing the document image, horizontal projection of preprocessed image is calculated, and

thresholding is performed according to projection peaks. After measuring threshold, each text line is segmented according to each previously determined threshold and the line separators are drawn at the valley point of each neighbor text lines in the original image. The major steps of proposed algorithm are shown in Fig. 1.

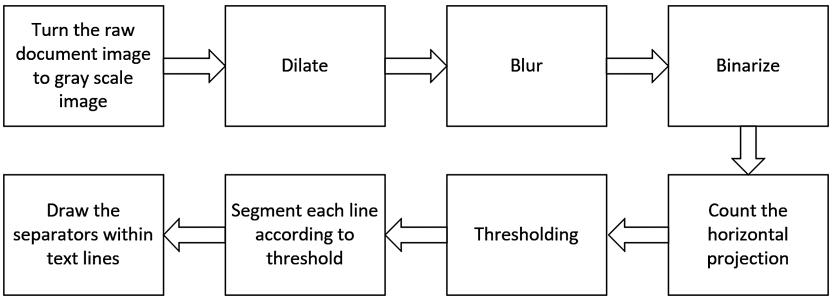
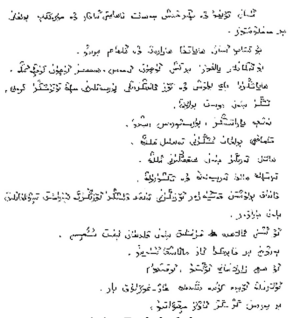


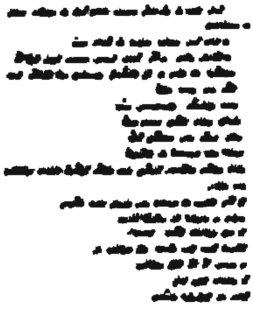
Fig. 1. Major steps of proposed algorithm

3.2 Preprocessing

Preprocessing aims to eliminate harmful or insignificant content and enhance useful features in document image. Thus, it improves generality of sample representation and performance of subsequent works [5]. Before the proposed text-line segmentation algorithm is applied, preprocessing is performed using the basic image processing technique which includes gray scaling, dilation, smoothing that contains noise removal and binarization which is utilized twice in proposed work. Firstly, weighted gray scaling method is used to turn the raw document image to gray image and binarized it afterward. Next, dilation is performed to thickening the text in document image. Then smoothing is applied to dilated image to eliminate insignificant stain points and smooth the document image in order to minimize local extrema in projection profile. Secondary binarization is conducted to at final step. Figure 2 shows the contrast of initial image and processed image.



(a). Initial image



(b). Processed image

Fig. 2. Before and after processing the initial image

Noise Removal and Smoothing

Noise removal is important to any kind of image processing task [8], especially for handwritten document images. Since binarized image is dilated, consequently, noisy points are also becoming bigger that could affect subsequent processing. 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. This means window size must be chosen adequately; otherwise, filtering will lose important information in image. In this paper, we use mean filtering to perform the noises removing. 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. For every pixel in image, the filter would calculate average value of corresponding window and replace the original value to the calculated one.

$$p_i = \frac{1}{m^2} \sum_{m=1}^{m^2} p_m \tag{1}$$

Where p_i is the center of kernel, p_m refers to the m-th visited pixel in a blurring kernel and m indicates the size of kernel. Besides, we also used mean filter to minimize the local extrema (minima and maxima points) in projection profile. 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.

Binarization

Otsu thresholding method is used for image binarization [11].

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

Weights ω_0 and ω_1 are probabilities of two classes, which refers text lines and the background, separated by a threshold t, and σ_0^2 and σ_1^2 is variance of these two classes. In this work, binarization also enhances the generality of the text lines in our document image. The projection after binarization on each differently blurred images are shown in Fig. 3. The black straight line indicates the mean value of the projection.

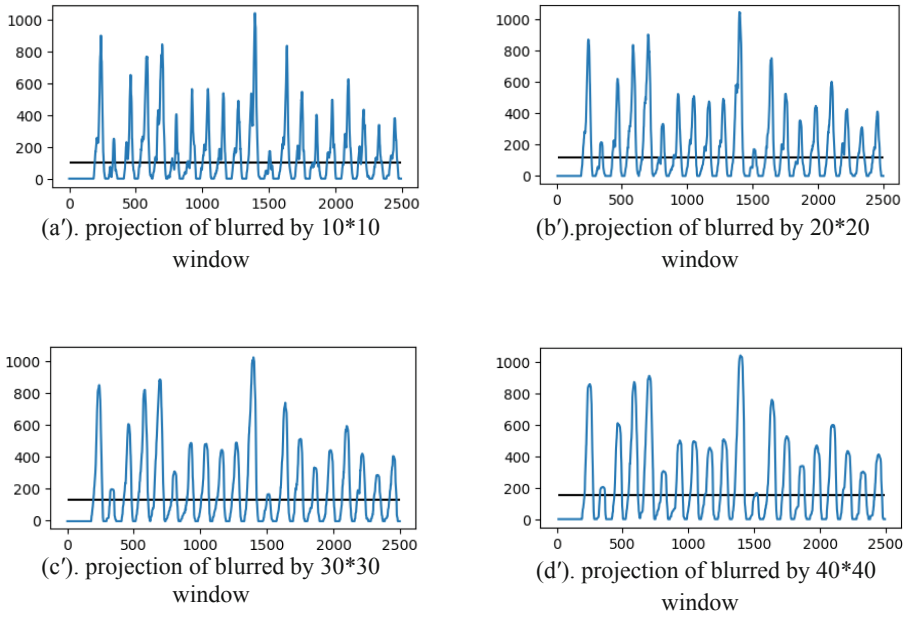


Fig. 3. Projections after binarization

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 [9]. 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 extracted to set P. Next, thresholding is performed as follows: visit each location $P(i)$ in set P; for given $P(i)$, the algorithm would take the half of current peak's value as threshold T [9].

$$T_p = \frac{P(i)}{2} \tag{3}$$

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 lesser than threshold, then the location of this projection value is assumed as starting point and added to set S and break the traverse loop. Then, the ending points are determined same way using forward visiting of projection values and the estimated ending point is added to set E.

The pre-estimated text-line intervals and tip points (starting, ending) are checked to confirm their validity and correctness by the following algorithm. First, traverse each element in set S and set E, then get start point $S(i)$ and end point $E(i)$, to calculate midpoint M_i of each interval using equation below;

$$M_i = \frac{S(i) + E(i)}{2} \tag{4}$$

Second, get next interval's start point $S(i+1)$, if it is greater or equal to M_i, then accept it as a true interval, otherwise it is omitted (5). This process makes the performance of interval selection more acceptable.

$$\begin{cases} S(i+1) \geq M_i \ true \\ S(i+1) < M_i \ false \end{cases} \tag{5}$$

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 between two adjacent estimated text-line positions which is between the current interval's ending point and next interval's starting point.

3.4 Algorithm

Step 1: Read a handwritten document image as a multi-dimensional array.
Step 2: Convert the raw image to gray scale image as G[][].
Step 3: Dilate the gray scale image G[][] and store it into matrix D[][].
Step 4: Blur the dilated matrix D[][] and store it into matrix P[][].
Step 5: Binarize the blurred matrix P[][] and store it into matrix B[][].
Step 6: Calculate the horizontal projection profile of binarized matrix B[][] and store the projection vector into HPP[].
Step 7: The peaks', which is above the mean value of projection, location is added to set P.
Step 8: For element in set P, calculate the threshold by multiplying 0.5 to peak value. Visit the elements of the HPP[] vector from current location forwardly and reversely to determine ending point and starting point, respectively. Where projection value is lesser than threshold is measured as starting point or ending point and add to set S and set E.
Step 9: Check the intervals whether it is overlapped by observing the starting point of interval whether it is greater than the next interval's midpoint. If it is greater, then accept it. If it is not delete the starting point and ending point from set S and set E.
Step 10: Draw a straight line at the valley point between two adjacent intervals' valley point.
Step 11: End.

4 Experimental Result

4.1 Database

To verify the proposed algorithm, we collected 210 Uyghur handwritten document image 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. Figure 4 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.

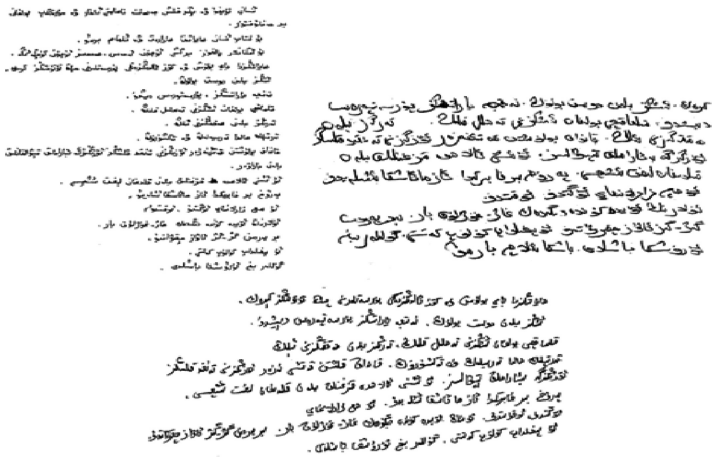


Fig. 4. Three samples of database

4.2 Evaluation Method

In this paper, we calculated precision, recall and the F-measure to evaluate the performance of proposed algorithm [12]. 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.

$$P = \frac{L_c}{L_s} \tag{6}$$

$$R = \frac{L_c}{L_t} \tag{7}$$

$$F = \frac{2PR}{P+R} \tag{8}$$

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

4.3 Result and Analysis

There are three parameters is taken in 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 experiment results of text-line segmentation on our dataset are shown in Fig. 5 and Table 1. For comparison, we evaluated the participant algorithm on our database.

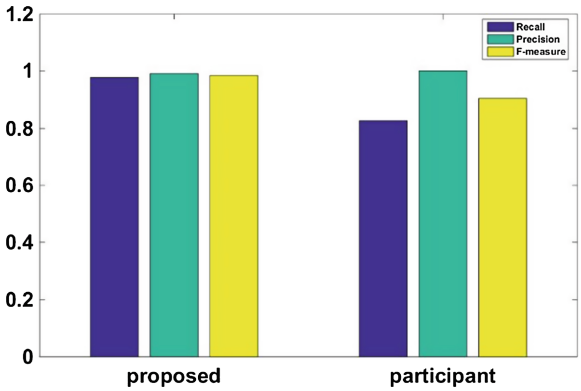


Fig. 5. Comparison of algorithms

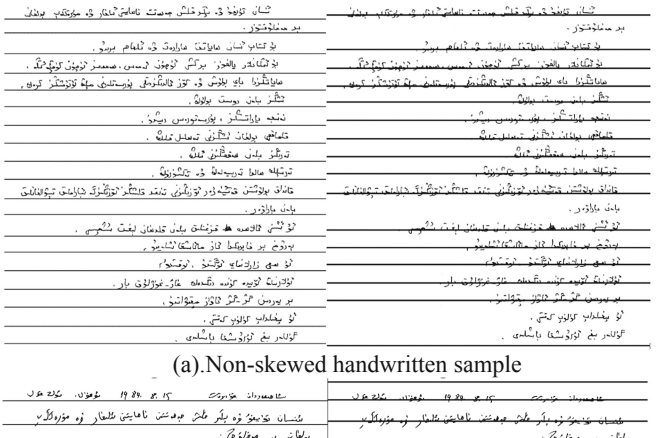
Table 1. Result of experiments

	Proposed	Participant
Recall	97.70%	82.75%
Precision	99.01%	99.99%
F-measure	98.35%	90.56%

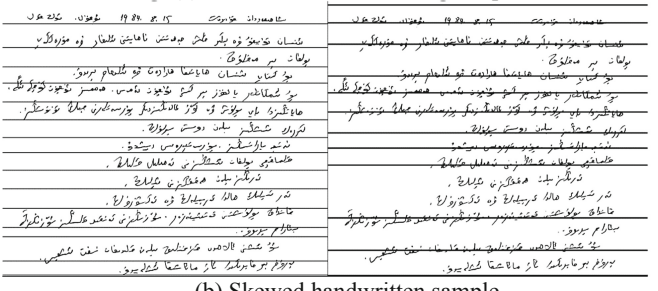
In the participant algorithm [9], the Polish document image is preprocessed including turning the original image to gray scale image, binarization and noise reduction. Then count the projection profile of preprocessed image and sort it with descending order. Then visit each value of sorted projection 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 lesser 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. 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, 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. For checking the 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 the intervals which is segmented previously, even it is not severely overlapped. In our checking mechanism, we consider each two adjacent intervals and observe the letter intervals start point whether it is greater than the former's midpoint.

According to results of the two segmentation algorithms in Table 1, the 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 the participant algorithm is not as strong as the proposed algorithm in text-line detection perspective. The 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



(a).Non-skewed handwritten sample



(b).Skewed handwritten sample

Fig. 6. Result of two different algorithms

effects of the compared algorithm are illustrated in Fig. 6. 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.

5 Conclusion

This paper proposed a novel approach 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. The experimental results show robustness of the proposed algorithm. 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. 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 styled text lines. Another factor that makes the performance of the algorithm decline is incorrect peak extraction from calculated projection profile. To develop more comprehensive and general text-line segmentation algorithm is the main content of our next work.

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

References

1. Razak, Z., et al.: Off-line handwriting text line segmentation: a review. Int. J. Comput. Sci. Netw. Secur. **7**, 12–20 (2008)
2. Yanikoglu, B., Sandon, P.A.: Segmentation of off-line cursive handwriting using linear programming. Pattern Recogn. **31**(12), 1825–1833 (1998)
3. Abliz, A., Simayi, W., Moydin, K., Hamdulla, A.: A survey on methods for basic unit segmentation in off-line handwritten text recognition. Int. J. Future Gener. Commun. Netw. **9**, 137–152 (2016)
4. Li, Y., Zheng, Y., Doermann, D., et al.: A new algorithm for detecting text line in handwritten documents. Proc. IWFHR La Baule **2**, 35–40 (2006)
5. Nie, L., Jiang, D., Guo, L.: A compressive sensing-based approach to end-to-end network traffic reconstruction utilising partial measured origin-destination flows. Trans. Emerg. Telecommun. Technol. **26**, 1108–1117 (2015)
6. Papavassiliou, V., et al.: Handwritten document image segmentation into text lines and words. Pattern Recogn. **43**(1), 369–377 (2010)

7. Bal, A., Saha, R.: An improved method for handwritten document analysis using segmentation, baseline recognition and writing pressure detection. Proc. Comput. Sci. **93**, 403–415 (2016)
8. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **1**(1), 1–12 (2018)
9. Ptak, R., Żygadło, B., Unold, O.: Projection-based text line segmentation with a variable threshold. Int. J. Appl. Math. Comput. Sci. **27**(1), 195–206 (2017)
10. Al-Dmour, A., Zitar, R.A.: Word extraction from arabic handwritten documents based on statistical measures. Int. Rev. Comput. Softw. **11**(5), 436–444 (2016)
11. Ohtsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
12. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS One **13**(5), 1–23 (2018)