International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.3 (2015), pp.291-302
http://dx.doi.org/10.14257/ijsip.2015.8.3.27

Connected Component feature Analysis based Handwritten Uyghur Text Lines Detection and Separation Algorithm

Kamil Moydin, Yi Xiaofang and Askar Hamdulla

Institute of Information Science and Engineering, Xinjiang University, Urumqi, China
kamilmoydin@gmail.com
College of Software, Xinjiang University, Urumqi, China
askarhamdulla@gmail.com

Abstract

This paper presents a Uyghur text-line separation algorithm based on classified connected components of Uyghur Handwritten scripts. In order to get the location of main text-lines, this paper proposes an adaptive painting and thinning algorithm. To insure the efficiency of text-lines segmentation, the post-processing of text-line detection procedures are introduced. The experimental results show that the algorithm is strongly robust for segmentation of text-lines with having some skewness, touching and overlapping, and small strokes remained.

Keywords: *Uyghur; Handwritten Documents; Text-line Segmentation; Robustness; Overlapping and Touching*

1. Introduction

Text-line segmentation is an essential pre-processing stage for handwriting recognition in many Optical Character Recognition (OCR) systems. It provides critical information for the segmentation of the text area, keyword matching, character segmentation and recognition. The segmentation result will directly affect the subsequent extraction identification. Although the printed text-line segmentation and recognition technology is relatively mature, the text-line segmentation for the handwritten documents is still quite challenging. The variations in inter-line distance, presence of inconsistent baseline skew, touching and overlapping text lines make this task more crucial and complex [1].

For the segmentation of lines from handwritten text, survey papers are available [2-3]. A lot of work has been carried out to segment lines of some languages script. What's more there are varied and some well developed techniques for them [4-5]. But very little work has been done for the minority scripts in China. Until now, only a few papers are available for Uyghur segmentation of handwritten scripts.

Uyghur as one of the main languages in Xinjiang, Documents written in Uyghur contain a large number of discrete points and additional part. In such documents, a connected component can be a whole word, a sub-word, a character, a dot, a component of connected dots, or a stroke. The presence of dots, strokes, overlapping and touching makes challenge for the segmentation of Uyghur script. (See from Figure 1. to Figure 4.).

This paper presents a new text line segmentation method for handwritten documents according to the characteristics of Uyghur script. The proposed technique is based on connected component feature, which divides the connected components into three parts. The rest of the paper is organized as follows:

Section 2 describes problems associated with line segmentation. Section 3 describes the method to be proposed. Experiments and results are discussed in section 4.

2. Challenges

When dealing with handwritten text, line segmentation has to solve some obstacles that are uncommon in modern printed text. The most predominant ones are:

- Skewed lines: lines of text in general are not straight

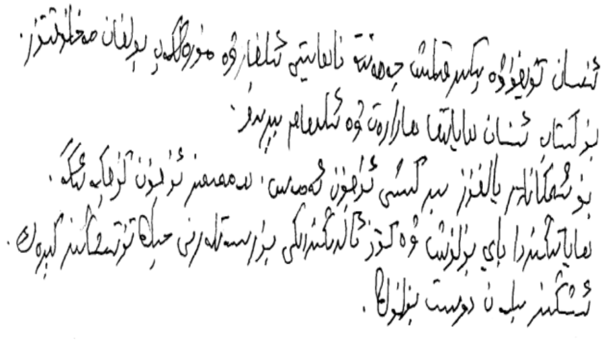


Figure 1. Skewed Lines

- Non-standard discrete strokes

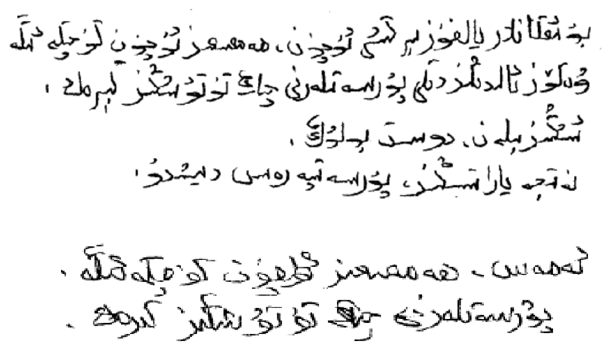


Figure 2. Discrete Strokes

- overlapping components

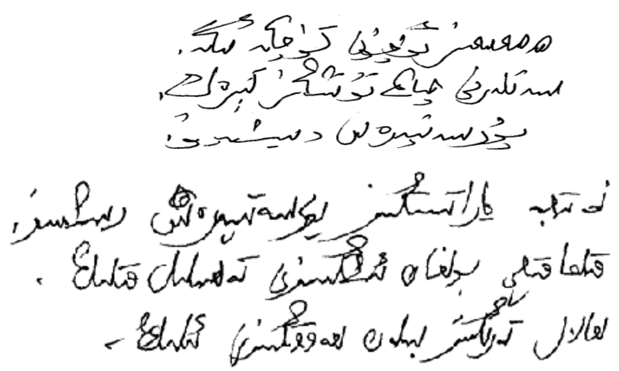


Figure 3. Illustration of Overlapping Components

- touching components

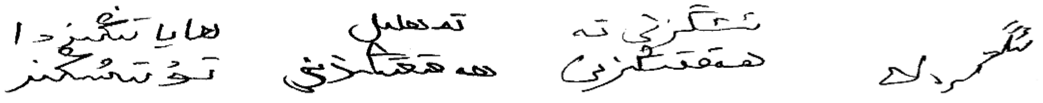


Figure 4. Illustration of Touching Components

3. The Proposed Method

Although many methods on handwritten line segmentation have been published in the literature for Latin and non-Latin scripts, only few papers are available on text-line segmentation of handwritten Uyghur documents.

There are mainly three basic categories. Methods lying in the first category make use of the Hough transform, Hough transform considers any image to compose of straight lines. It creates an angle, offset plane in which the local maxima are assumed to correlate with text lines. Hough transform has trouble detecting curved text lines. The second category makes use of projections. The method of horizontal projection of the whole text is suitable for segmentation of the text with straight lines and with large gap in lines. This method cannot segment handwritten document because it contains touching lines, overlapping lines or fluctuating lines [6]. The third category deals with methods that use a kind of smearing. The short white runs are filled with black pixels intending to form large bodies of black pixels, which will be considered as text line areas. Smearing methods can’t deal well with touching and overlapping components. In some methods that do not lie in these categories, the text line extraction problem is seen from an Artificial Intelligence perspective. The aim is to cluster the connected components of the document into homogeneous sets that correspond to the text lines of the document [7].

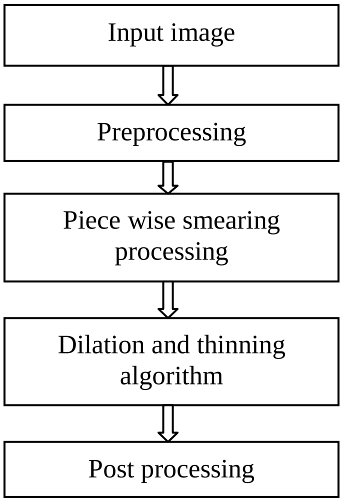

Input image
Preprocessing
Piece wise smearing processing
Dilation and thinning algorithm
Post processing

Figure 5. Block Diagram of the Proposed Method

To meet the aforementioned challenges, we propose a methodology which consists of four main steps (see Figure 5). The first step includes connected component extraction, average character height estimation, and partitioning of the connected component domain into three distinct spatial sub-domains. In the second step, a piece wise smearing operation is performed while the third step, a thinning Algorithm is used for the detection of potential text lines and the last step is used to correct possible splitting and detect possible text lines which the previous step did not reveal, and finally, to separate vertically connected parts and assign them to text lines. A detailed description of these stages is given in the following Sections 3.1–3.4.

3.1 Pre-processing

The preprocessing step consists of three stages. First, remove noise from handwritten image.

$$if \quad W_i \le 2 \,\&\, H_i \le 2 \qquad then \quad B_i = 0 \tag{1}$$

Where W_i denotes the width of the i th connected component, H_i denotes the height of the i th connected component and $B_i \in \{0,1\}$ is the value of i th connected component (0 for black and 1 for white).

Then, the connected components of the binary image are extracted [18] and the bounding box coordinates for each connected component are calculated. The average character height AH and AW for the whole document image is calculated [8].

$$AW = \frac{1}{n}\sum_{i=1}^{n} W_i \quad AH = \frac{1}{n}\sum_{i=1}^{n} H_i \tag{2}$$

Where AW , AH denote the average character width and the average character height.

At last, we divide the connected components domain into three distinct spatial sub-domains denoted as “Subset 1”, “Subset 2”, and “Subset 3”.

“Subset 1” should contain characters as additional parts, punctuation marks, and small characters. The equation describing this set is:

$$((W_i < 0.5 * AW)\, and\, (H_i < 3 * AH\ ))$$
$$OR$$
$$((H_i < 0.5 * AH) and (W_i > 0.5 * AW)) \tag{3}$$

The motivation for ‘Subset 1’ definition is that additional parts usually have width less than half the average character width or height less than half the average character height.

“Subset 2” contains all large connected components. Large components are either capital letters or characters from adjacent text lines touching. The size of these components is described by the following equation:

$$(\ 0.5 * AH \le H_i < 3 * AH\ )\ \&\ (W_i < 0.5 * \mathrm{AW}\ ) \tag{4}$$

The motivation for ‘Subset 2’definition is to exclude some small strokes and large components which belongs to more than one text lines.

Finally, “Subset3” contains all large connected components. Large components are either capital letters or characters from adjacent text lines touching. The size of these components is described by the following equation:

$$H_i \ge 3 * AH \tag{5}$$

The motivation for ‘Subset 3’ definition is to grasp all connected components that exist due to touching text lines. We assume that the corresponding height will exceed three times the average character height.

3.2 Piece Wise Smearing Algorithm

At first, we divide the Subset 2 into vertical stripes of width AW. In our work, stripes are considered from left to right. Width of the last stripes may differ from AW (see Figure 6). Subsequent to the division of the input image into stripes, the gray value of each pixel in each row of a stripe is modified by changing it with the average gray value of all pixels present in that row of the stripe. The average gray value (G_{ki}) in each row-stripe is computed using the following formula:

$$GBW = 255.*BW \tag{6}$$

$$G_{ki} = \frac{\sum_{j}^{n} GBW_{ij}}{DW_k} \tag{7}$$

Where BW denotes the Subset 2, The G_{ki} is the average gray value of all the pixels placed in the i th row and kth stripe, GBW_{ij} is the gray value in the i th row and jth column of the input image BW and DW_k is the width of k th stripe. Results after applying the piece wise smearing algorithm are demonstrated in Figure 7.

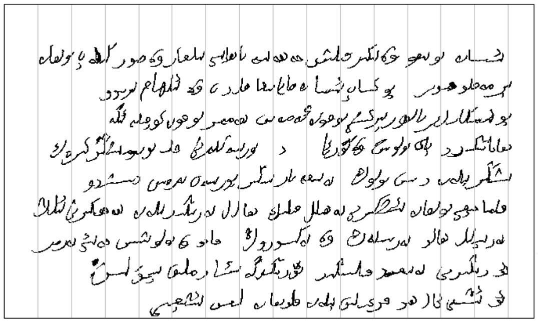


Figure 6. Vertical Strips

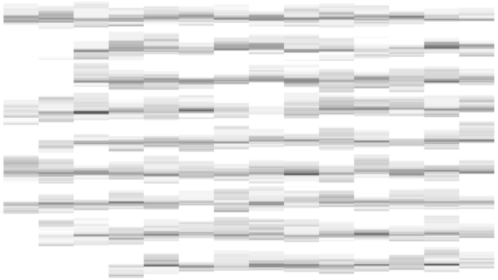


Figure 7. Results after Smearing

The gray value of each pixel is range from 0-255, according to the style of Uyghur handwritten text. The text on the left tend to leave a lot of empty space, and the distribution of the number of words of each line of text are different. Therefore, the resultant gray-scale image is converted to binary by applying the Otsu's method [9] on each stripe separately (see Figure 8.). The black and white rectangles represent the foreground (text regions) and background, respectively. This image is smoothed stripe-wise by a set of smoothing operations to fill white space between two consecutive black regions by black and to remove very thin black areas by converting them to white. This is done to avoid having redundant segmentation lines that may produce improper text line segmentation. The white space between two consecutive black regions is filled by black pixels based on the following criterion:

$$if \quad HW_{ki} \le AHW_k \ \& \ HW_{ki} \le 2AH \qquad then \quad BW_{ki} = 0 \qquad (8）$$

Where k=1 to no. of stripes, AHW_k denotes the average heights of white areas in the kth stripe and HW_{ki} denotes the height of the i th white region in the k th stripe. $BW_{ki} \in \{0,1\}$ is the value of i th white area in the k th stripe(0 for black and 1 for white).

In order to obtain the location of main text region, some black thin rectangles are converted into white based on the following criterion:

$$if \quad HB_{ki} \le T \qquad then \quad BB_{ki} = 1 \qquad (9）$$

where HB_{ki} denotes the height of the i th black region in the k th stripe, T is a threshold obtained from the heights of black areas, and VBikA{0,1} is the value of the i th black area in the k th stripe. The threshold T is dynamically computed with respect to each input image.

For our datasets, T=1/3 is the average heights of black areas to each input image. $BB_{ki} \in \{0,1\}$ is the value of i th black area in the k th stripe(0 for black and 1 for white).

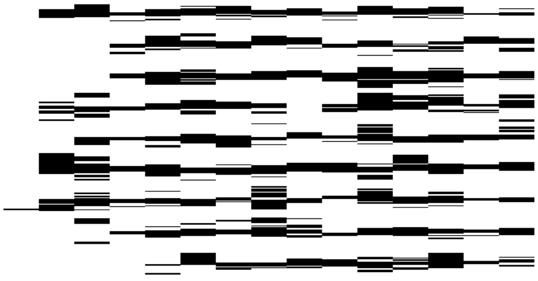


Figure 8. Results of Binary Operation

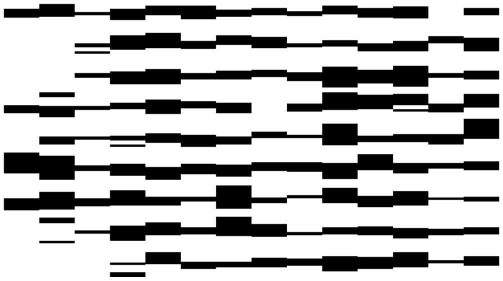


Figure 9. Results of Filling and Removing

With a clear observation it was noticed that the width of the stripe should actually be dependent on the gaps (white space) between the text lines and it should be dynamically calculated for each text-page. In other words, if the text-lines are closely written then the width of the stripe should be very small and vice versa [10]. So in the next step, the statistical mode of the heights of all white rectangles of the smoothed image is computed. Based on this value, the image of "subset 2" is divided into stripes again and the proposed piece wise smearing algorithm is applied once more. The results are presented in Figure 10.

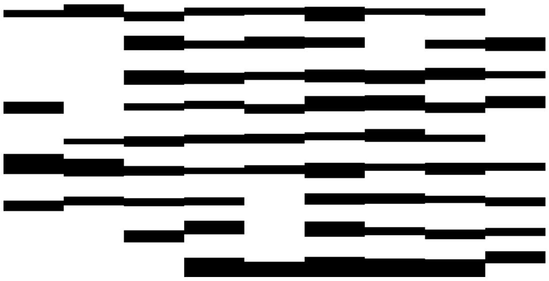


Figure 10. Results of More Smearing

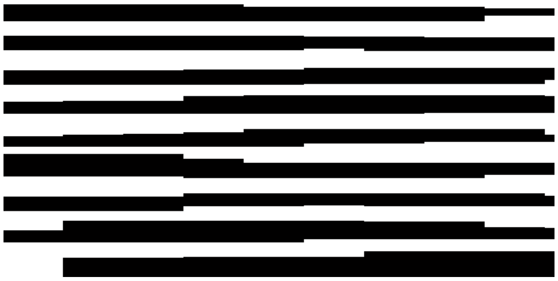


Figure 11. Results of Dilation

3.3 Dilation and Thinning Algorithm

In order to make each line as a single component, the remaining black rectangular areas are dilated to connect the black rectangular areas with the next stripes. Here we use a structuring element of length 4xSW and a width of one. Where SW is the width of the strip which divided once again, the length of the structural element for dilation is experimentally selected based on our 210 datasets. The result of the dilation operation on the image given in Figure 10 is shown in Figure 11.

The complementary image is then thinned by applying the algorithm proposed in [11] to obtain some candidate text line location. The candidate thinning lines indicate the text region information. In some cases, some candidate lines may have some junction points, sobel filter with the following 3×3 mask [12] is employed for this purpose. Finally, based on the morphological operations, the residual lines to be deleted, the region of main text lines are located. The result of the thinning and trimming operation is shown in Figure 12.

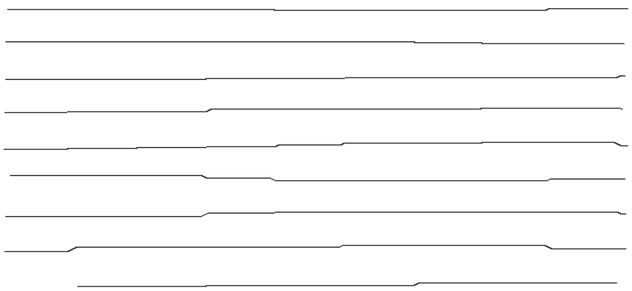
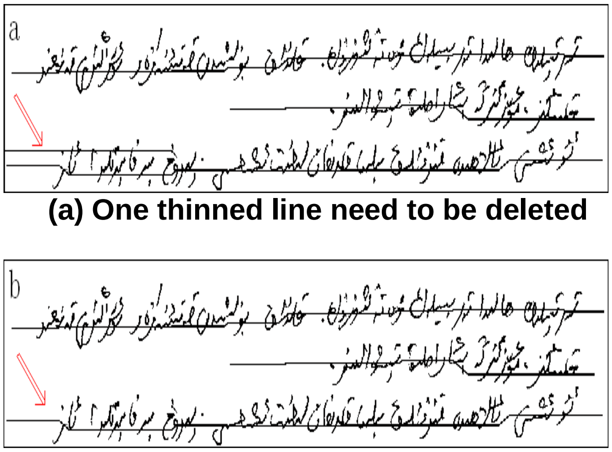


Figure 12. Results of Thinning and Trimming on Foreground Part of Figure11

3.4 Post-processing for Text Line Detection

The post-processing step consists of four stages. At the first stage, a merging technique over the result of dilation and thinning Algorithm is applied to correct possible false alarms (see Figure 13(a)).



(a) One thinned line need to be deleted

(b) The results of merging procedure

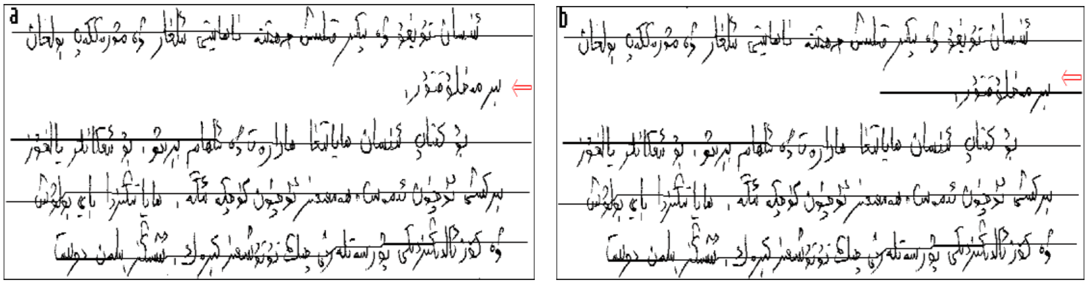
Figure 13. Results of First Step Post-processing Operations

In the process of building text thinning lines, two text thinning lines may appear on the same line, and the two are in close proximity, see Figure 13(a). Therefore, it will cause problems for the post character segmentation and coloring. So we compute the distance between two adjacent lines: $dis\_1 = | y_i - y_{i-1} |$ and $dis\_2 = | y_{i+1} - y_i |$, Where y_i denotes the gravity height of the ith thinning lines. Merging technique for the i th lines are applied according to the following criterion:

$$\begin{array}{l} if \quad dis\_1 < 0.5 \times A_d \ \& \ dis\_2 > 0.5 \times A_d \\ or \ \ dis\_1 > 0.5 \times A_d \ \& \ dis\_2 < 0.5 \times A_d \end{array} \qquad (10)$$

Then delete line.

Where A_d denotes the average distance of adjacent lines. Merging lines using the proposed algorithm for problem document shown in Figure 13(b).



(a) One thinning line is not detected **(b) the creation of new thinning line**

Figure 14. Results of Second Step Post-processing Operations

The second stage deals with some text lines which are not detected see Figure 14 (a). In the process of building text thinning lines, duo to connected components less than normal in some text regions, some text lines are not detected after morphological operations. Therefore, the undetected regions are located in order to solve this kind of problem. And then, the algorithm for construction of new thinning lines is applied, Figure 14b shows the creation of new thinning lines. The algorithm is shown in Figure 15.

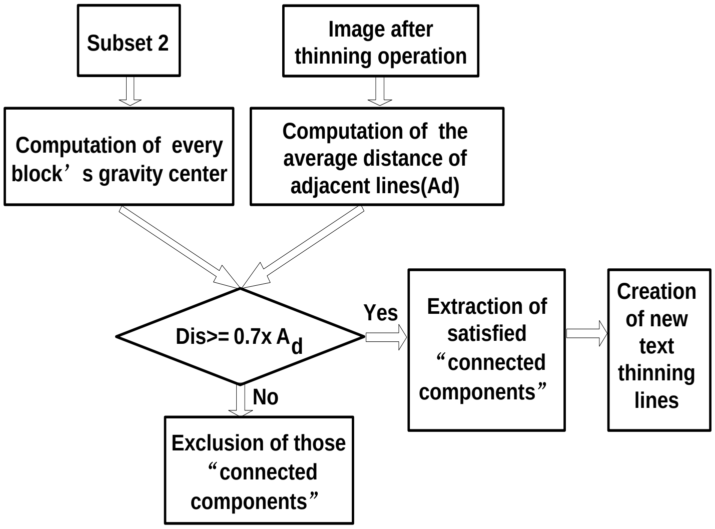

Subset 2
Image after thinning operation
Computation of every block’ s gravity center
Computation of the average distance of adjacent lines(Ad)
Dis>= 0.7x A_d
Yes
No
Extraction of satisfied “connected components”
Creation of new text thinning lines
Exclusion of those “connected components”

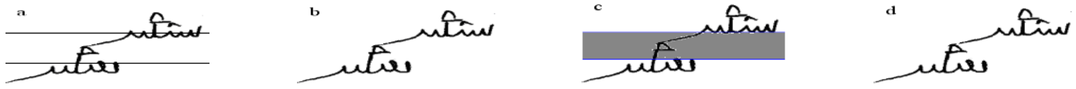
Figure 15. The Line Detection Algorithm

The third stage deals with components lying in “Subset3”. This subset includes components whose height exceeds three times the average height AH. All components of this subset may belong to two different text lines (see Figure 16(a)). The procedure we follow to separate vertically connected characters consists of the following steps:

Step1: Extract the corresponding connected component which belongs to two text lines or more text lines (see Figure 16(b)).

Step2: Define the segmentation zone Z according to the constraints: $1*H_c \le y \le 2*H_c$, where H_c is the height of the connected component (see Figure 16(c)).

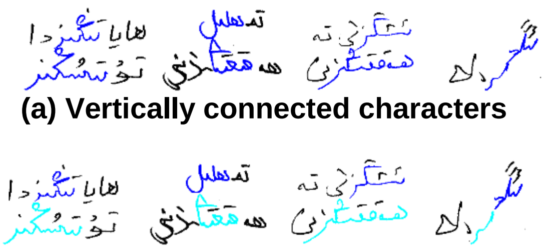
Step3: Erosion operation is employed for this segmentation zone, and the structural element se=strel(disk,1). After the erosion, check the connected component whether remain is one component. If not, remove the pixels in the middle of the zone Z. If the segmentation zone are divided into several parts, each part is grouped to the closest lines(see Figure16(d)).



(a) a connected component that belongs to two successive text lines is located; (b) extract the corresponding connected component; (c) define the segmentation zone Z; (d) results after character segmentation

Figure 16. Separating Vertically Connected Characters

Some results of the proposed algorithm with different samples are shown in Figure 17.



(a) Vertically connected characters

(b) Results of segmentation

Figure 17. Some Results of the Proposed Algorithm with Different Samples

The last stage, after getting the location of main text region, coloring with these main parts (see Figure 18), some connected components of "Subset 1" and some small discrete strokes that were not clustered to any text lines are grouped to the closest line(see Figure 19).

4. Experiments and Results

The experiments are performed on various handwritten text images in Uyghur Script. The images with some degree of skewness, less line gap, more gaps in words etc. are considered. For experiments, we considered only single column document pages. By viewing the results on the computer's display, we calculate line segmentation accuracy manually. Accuracy of line extraction algorithm is measured according to the following rules [13]:

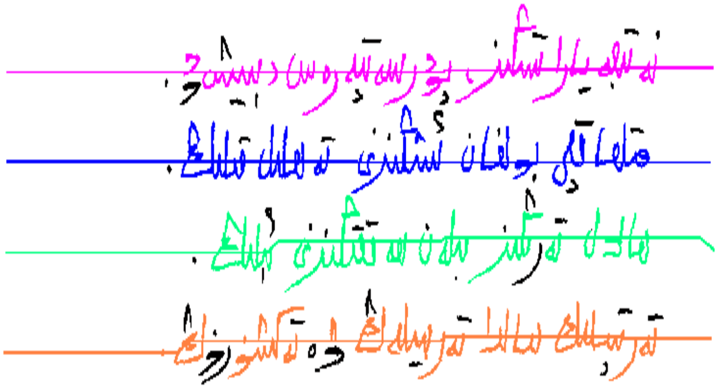


Figure 18. Main Parts of Each Line have been shown in Different Color

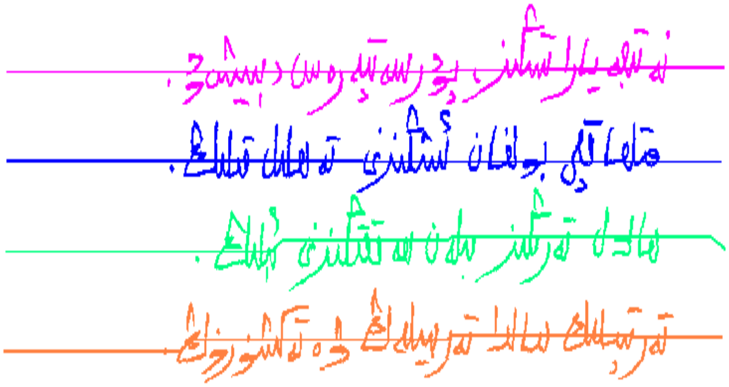


Figure 19. Unclassified Strokes are Grouped to the Closest Line

$$recall\_rate = \frac{correctly \quad detected \quad lines}{ground \quad truth \quad lines} \tag{11}$$

$$error\_rate = \frac{false \quad lines}{detected \quad lines} \tag{12}$$

Distributions of experimental results of line segmentation are given in Table 1. From Table 1, it is evident that the results reported in the proposed technique outperformed the results of all other algorithms. But the problem of incorrectly segmented lower or upper zone characters remains there, at the same time, the segmentation algorithm of touching characters need to be perfected.

Table 1. Comparison of our Results with Different Algorithms Used in [6, 10] and Tested on 210 Text-pages (2563 Ground Truth Lines)

	Detected lines	Correct detection	Recall rate(%)	Error rate(%)
Projection profiles[6]	2217	1963	76.58	15.96
Method of document[10]	2514	2431	94.84	3.30
Proposed method	2552	2474	96.53	3.04

Acknowledgements

This work has been supported by the National Natural Science Foundation of China under grant of 61462080.

References

[1] A. Kumar and S. R. Jindal, "Segmentation of Handwritten Gurmukhi Text into Lines", IJCA Proceedings on International Conference on Recent Advances and Future Trends in Information Technology, iRAFIT, vol. 9, (**2012**) April, pp. 13-17,

[2] L. Likforman-Sulem, A. Zahour and B. Taconet, "Text line segmentation of historical documents: a survey", International Journal of Document Analysis and Recognition (IJDAR), vol. 9, no. 2-4, (**2007**) April, pp. 123-138.

[3] Z. Razak, K. Zulkiflee, M. Y. I. Idris, E. M. Tamil, M. Noorzaily, M. Noor, R. Salleh, M. Yaakob, Z. M. Yusof, and M. Yaacob, "Off-line Handwriting Text Line Segmentation: A Review", IJCSNS International Journal of Computer Science and Network Security, vol. 8, no. 7, **(2008),** July.

[4] I. Bar-Yosef, N. Hagbi, K. Kedem and I. Dinstein, "Line segmentation for degraded handwritten historical documents", 10th International Conference on Document Analysis and Recognition, 2009. ICDAR '09, Barcelona, (**2009**), July 26-29, pp. 1161-1165

[5] A. Nicolaou and B. Gatos, "Handwritten Text Line Segmentation by Shredding Text into its Lines", ICDAR '09 Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, IEEE Computer Society Washington, DC, USA, (**2009**), July, pp. 626-630.

[6] R. Manmatha and J. L. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents", IEEE Transactions on Pattern Analysis and Machine Intelligent, vol. 27, no. 8, (**2005**), pp. 1212-1225.

[7] S. Jindal and G. S. Lehal, "Line segmentation of Handwritten Gurmukhi Manuscripts", DAR '12 Proceeding of the workshop on Document Analysis and Recognition, ACM New York, NY, USA, (**2012**), December, pp. 74-78.

[8] G. Louloudis, B. Gatos and I. Pratikakis, "Text line detection in handwritten documents", Pattern Recognition, vol. 41, (**2008**), pp. 3758-3772.

[9] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Transaction on System, vol. 9, no. 1, (**1979**), pp. 62-69.

[10] A. Alaei, U. Pal and P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation Pattern Recognition", vol. 44, (**2011**), pp. 917–928.

[11] L. Lam, L. Seong-Whan and Y. Suen Ching, "Thinning methodologies—a comprehensive survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 9, (**1992**), p. 879.

[12] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", second edition, Prentice Hall, India, (**2002**).

[13] Y. Fei and L. Chenglin, "Handwritten Chinese Text Line Segmentation by Clustering with Distance Metric Learning", Pattern Recognition, vol. 42, no. 12, (**2009**), pp. 3146-3157

Authors

Kamil Moydin, he 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. He was a visiting scholar in the Osaka Institute of Technology, Japan from 1994 to 1996. In 2002, he got the position of associate professor in Xinjiang University. His research interests include computer network, pattern recognition, and digital image processing.

Yi Xiaofang, he received his B.E. and M.S. degree in electronics, signal and information processing from Xinjiang University, China, in 2010 and 2013, respectively. His scientific interest includes handwriting identification. Currently, he is a research assistant at the Key Laboratory of Intelligent Information Processing, Xinjiang University, China. His research interests include text lines detection and segmentation techniques, writer identification and verification.

Askar Hamdulla, he received B.E. in 1996, M.E. in 1999, and Ph.D. in 2003, all in Information Science and Engineering, from University of Electronic Science and Technology of China. In 2010, he was a visiting scholar at Center for Signal and Image Processing, Georgia Institute of Technology, GA, USA, tutored by Professor Biing-Hwang (Fred) Juang. Currently, he is a professor in the School of Software Engineering, Xinjiang University. He has published more than 120 technical papers on speech synthesis, natural language processing and image processing. He is a senior member of CCF and an affiliate member of IEEE.