A Multi-scale Text Line Segmentation Method in Freestyle Handwritten Documents

Yangdong Gao, Xiaoqing Ding, Changsong Liu
State Key Laboratory of Intelligent Technology and Systems
Department of Electronic Engineering, Tsinghua University
Beijing, P.R. China
{gaoyd, dxq, lcs}@ocrserv.ee.tsinghua.edu.cn

Abstract

Text lines in free-style handwritten documents are often curved, touch or overlap with each other, which presents a challenge for text line segmentation. In this paper, we proposed a novel text line segmentation method that utilizes the advantages of algorithms in both the small scale and large scale. A path is dynamically detected between each pair of neighboring text lines to separate them. During the process, the line-separating path's coordinate in each step is determined by a three-stage multi-scale method that combines (1) a simple local minima search algorithm, (2) the technique based on following the contour of the foreground component and (3) the piecewise projection profile. Without training, our method has achieved a high segmentation accuracy on plenty of samples, which proves its strong adaptability to various line conditions. Experimental results show that the proposed method outperforms traditional methods.

Keywords: handwritten document; text line segmentation; document image analysis;

I. INTRODUCTION

As a critical preprocessing stage in handwritten document recognition, text line segmentation exports results which will affect the overall performance of recognition system. There still exist several challenges in the text line segmentation for handwritten document. Unlike machine printed document, the free-style handwritten text lines are often curved, have various skew angles, no uniform direction, touch or overlap with each other.

So far a wide variety of algorithms for text line segmentation in handwritten documents have been reported in the literature, but the challenges mentioned above have not been solved satisfactorily. Existing line segmentation methods can be roughly categorized as top-down projection based methods, bottom-up component grouping based methods or a hybrid of the two. Although projection profile based methods have been successfully applied for machine-printed documents, the methods do not work when text lines have variable skew angles. The piecewise projection approach [1] divides the document into vertical stripes, detects local minimums of the projection profile in each stripe and groups them to form the final text line segmentation. However, due to the demand for a lot of empirical parameters, the robustness and general performance of the approach are hard to guarantee. The Hough-based algorithms [2] can not deal with the problem of different skew angles in the same text line. The approaches that rely on connected component (CC) grouping are sensitive to topological changes of the components. For example, the minimum spanning tree clustering method [3] will fail to gain acceptable segmentation result if the neighboring lines touch or overlap seriously.

To solve the aforementioned challenges, we adopt a novel method that utilizes advantages of algorithms in both the small scale and large scale (Fig. 1). For each pair of the text lines, there exists a path from the left to right border that separates those two text lines [4]. When the space between neighboring lines is wide enough, the path's coordinate in the next step is gained by a simple local minima search algorithm. The small-scale algorithm can trace the boundary curve of the text line even if it changes frequently. When the space becomes narrow, the optimal solution in larger scale will provide a more reliable segmentation result. Therefore, the path will try to bypass the CC following its contour. However, when the neighboring text lines overlap seriously, piecewise projection is employed to determine the coordinates of the path.

The remainder of the paper is organized as follows. In Section 2, the proposed method for text line segmentation is detailed. Section 3 deals with the evaluation method and the experimental results are shown in Section 4. Finally, in Section 5 the conclusion is presented.

II. PROPOSED METHOD

Our text line segmentation method consists of three main steps: preprocessing, detecting the line-separating paths, post processing.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.135

IEEE computer society

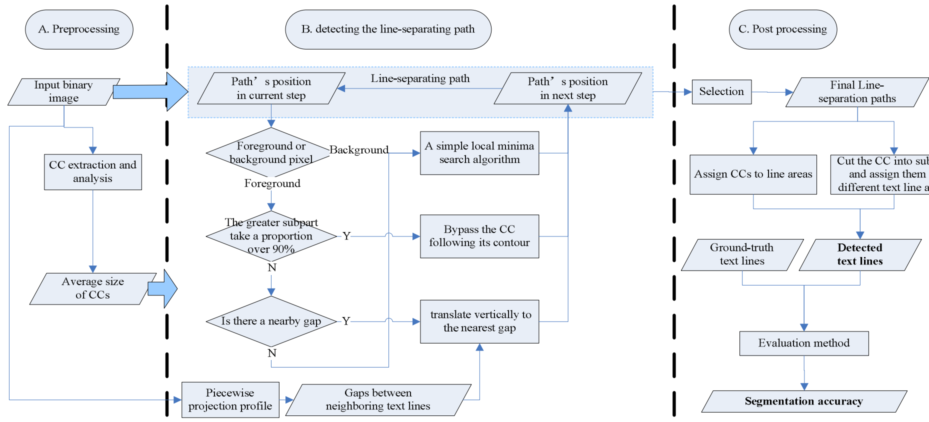


Figure 1. Block diagram of the proposed method.

A. Preprocessing

This step includes CC extraction and the estimation of characters' average size. First, CCs of the input binary document image are extracted. Most of the CCs are not complete characters, but just elements or strokes. The mean height and width, which act as important parameters in whole process, are estimated excluding the smaller-size portion. We select the top 5% heights of all the CCs and get their mean value *MaxheightCC*. Then the CCs with greater height than *MaxheightCC* × *0. 1* are used to calculate the mean height *meanHeightCC*. The mean width *meanWidthCC* is gained by similar method.

In [4], the input image is blurred to suppress noise before further processing. In practicing experiment, we find it's a hard task to determine the size of the blurring window, and the result estimated by [4] is not stable or robust enough. It is more advisable to choose specific noise-decreasing approaches according to the particular situation of document images. So the blurring process is removed from our method.

B. Detecting the line-separating paths

In this section, we detect a line-separating path between each pair of neighboring text lines to separate them from the left to the right bounder and reversely. The multi-scale method includes three sub algorithms.

1) A simple local minima search algorithm:

The *img(y, x)* is a single pixel of the input binary document image and the value of *img(y, x)* is defined as

$$img(y,x)=\begin{cases}1 & \text{if foreground pixel}\\ 0 & \text{if background pixel}\end{cases} \quad (1)$$

When the space between neighboring lines is wide enough, there exists a path along the background pixels. Such path is detected using a simple local minima search algorithm discussed subsequently. The equation (2) inspired by [4] is recursively used to find each coordinate of paths dynamically.

$$\mathrm{y}(k+1)=\begin{cases}y(k)-1 & \textit{if} \quad imgAbove<imgBelow\\ y(k) & \textit{if} \quad imgAbove=imgBelow\\ y(k)+1 & \textit{if} \quad imgAbove>imgBelow\end{cases} \quad (2)$$

The *y(k)* represents the vertical coordinate of the path where *k* is its horizontal coordinate. *imgAbove* and *imgBelow* (Fig. 2) are the sums of foreground pixels in a *(n+1)* × *(n+1)* rectangular region with the center at *(y(k)-D, k)* and *(y(k)+D, k)* respectively.

$$imgAbove=\sum_{y=y(k)-D-n/2}^{y(k)-D+n/2}\ \sum_{x=k-n/2}^{k+n/2} img(y,x) \quad (3)$$

$$imgBelow=\sum_{y=y(k)+D-n/2}^{y(k)+D+n/2}\ \sum_{x=k-n/2}^{k+n/2} img(y,x)$$

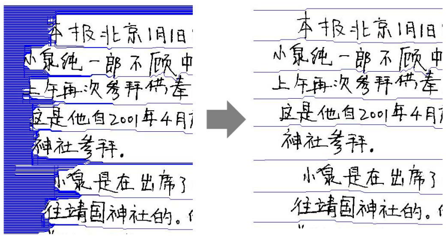
D is given by (4):

$$D=\frac{meanHeightCC}{2} \quad (4)$$

Figure 2. Red arrows represent the effects of the foreground pixels on the paths drawn in cyan. The rectangular regions of *imgAbove* and *imgBelow* are marked by blue rectangules.

During each step of the search, the path's next vertical coordinate *y(k+1)* is gained only based on the information of current vertical coordinate *y(k)*, *imgAbove* and *imgBelow*, so this technique can be considered as a small-scale technique compared with followings.

The document image is divided vertically into 3 parts: the left, middle and right. The projections of the left and right parts are calculated and their local minimal points are selected to be the initial points of the paths. In this way the computation amount is reduced compared with [4], in which the paths start from every point along the border of the image, (see Fig. 3 (a)).

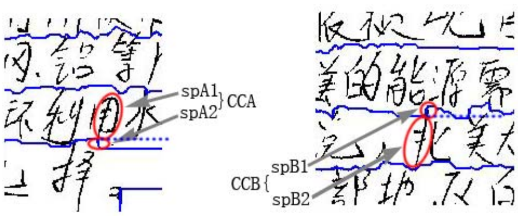


(a) (b)

Figure 3. Reduce the number of paths by selecting initial points along the left border. (a) previous approach in [4]. (b) improved approach in our method.

2) The technique based on following the contour of the foreground CC:

When the space between neighboring text lines becomes narrow, the path is very likely to meet with foreground pixels. When this happens, we assume the CC is cut into subparts along the horizontal direction by the path and its extension line (Fig. 4).



(a) (b)

Figure 4. The CCs are cut into subparts along the horizontal direction by the path and its extension line (drawn by dashed blue lines). 4 subparts (sp) are circled by red ellipses in this figure. CCA is cut into spA1 and spA2 in (a). CCB is cut into spB1 and spB2 in (b).

If the greater subpart of the CC takes a proportion over 90% in both height and pixel number, the path will bypass the CC following the contour of the CC (Fig. 5).

(a) (b)

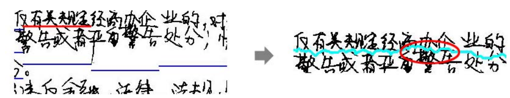
Figure 5. Green curves represent the contour of the CCs. Red ellipses mark the area where the path bypasses the foreground pixels following the contour of the CCs.

3) The piecewise projection profile:

When the greater subpart of the CC takes a proportion less than 90% either in height or pixel number, the path probably runs into the central region of a text line, or the text lines overlap seriously in the area. The piecewise projection profile, a classical technique in larger scale, is employed here. Firstly, the document image is divided into N stripes, and N is designed to be smaller than that in published papers. Then the local minimums of projection profile in each stripe are selected as "gaps" between neighboring text lines. Suppose the yp_i represents the vertical coordinate of a gap in the same stripe the path's current position belonging to. yd_i represents the vertical coordinate of a gap in the neighboring stripe. The mD is the average vertical interval of all gaps gained by the piecewise projection profile. The path's next vertical coordinate is gained using (5).

$$\begin{array}{lll} if & |y(k)-yp_i|<0.5\times mD & y(k+1)=yp_i \\ elseif & |y(k)-yd_i|<0.25\times mD & y(k+1)=yd_i \\ else & & y(k+1)\leftarrow Eq(2) \end{array} \quad (5)$$

If the vertical distance between the path's current position and the nearest gap is smaller than a threshold, the vertical coordinate of the gap will be assigned to that of the path in the next step (Fig. 6).



(a) (b)

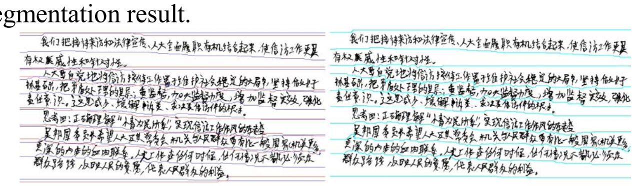
Figure 6. The part of the path circled by red ellipse in (b) find out the correct segmenatation position wth the help of the gap in the neighboring stripe drawn in red in (a).

In this technique, even more information of a stripe of the image is needed to determine the path's next coordinate, which means the piecewise projection profile is in larger scale.

On the whole, we strive to avoid foreground pixels in the procedure of detecting line-separating paths. When the text lines overlaps seriously in the region and the path can't avoid the foreground pixel, the path attempts to pass through the foreground region along the nearest gap gained by the piecewise projection profile.

C. *Post processing*

In this stage, we select some of the paths from adjacent ones based on their altitude, smoothness and the frequency of intersections with others (Fig. 7). Empirical analysis of the experiments shows the path that is rare to meet with foreground pixels and smoother will give a more satisfying segmentation result.



(a) (b)

Figure 7. The paths from the left to right border are drawn blue while the paths from the right to left border are drawn red in (a). The selected line-separating paths are drawn cyan in (b).

In the final step, the foreground CCs are assigned to different text line areas segmented by the line-separating paths. Sometimes if the CC belongs to multiple text lines areas, the segmentation of it along the path will be considered in this step.

Some examples are shown in Fig. 8.

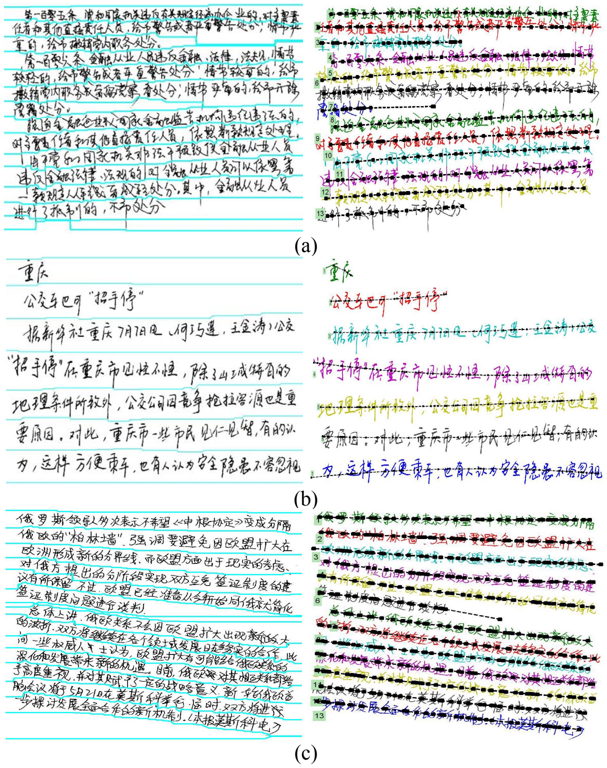


Figure 8. Several segmentation results on HIT-MW. (a) The heights of the text lines vary widely. Moreover the first and the second text lines overlap with each other. (b) The text lines have different lengths. (c) The text lines have various directions and skew angles.

III. PERFORMANCE EVALUATION METHOD

Some researchers perform the evaluation of text line segmentation algorithm based on visual criteria. Such evaluation method is not adopted due to its tediousness and strong subjectivity in this paper. To avoid the shortages, we use an automatic performance evaluation that has been followed in many document segmentation competitions [5] and published papers. Based on counting the number of the matching pixels between ground-truth text lines and detected text lines, the evaluation method [6] below provides evaluation results at both pixel level and line level.

Let the number of ground-truth lines be N_{gt}, the number of detected lines be N_d and $N_m=max(N_{gt}, N_d)$. A $N_m \times N_m$ square matrix *MatchScore* is constructed whose element $MS(i, j)$ represents the number of matching pixels of the i^th ground-truth line and the j^th detected line. A line is allowed to be matched to a dummy one and the corresponding $MS(i, j)=0$. The one-to-one correspondence can be expressed as an array C:

$$\begin{array}{lcccc} \text{index of ground-truth lines} & 1 & 2\ \cdots & i & \cdots \\ & \updownarrow & \updownarrow & \updownarrow & \\ \text{index of detected lines} & C(1) & C(2)\cdots & C(i) & \cdots \end{array}$$

Our purpose here is to find a C_m by which the sum of matching pixels $S(C)$ reaches maximum:

$$S(C) = \sum_{k}^{N_m} MS\left(k, C(k)\right) \tag{6}$$

$$C_m = \arg \sum_{k}^{N_m} MS\left(k, C(k)\right) \tag{7}$$

In theory, the enumeration of one-to-one correspondences is as many as the permutation of N_m. Thanks to the Hungarian algorithm [7], the global optimal solution can be obtained in an acceptable time.

The S_{gt} represents the sum pixels of all the ground-truth text lines. Then the overall correct rate at pixel level is defined as

$$PL = \frac{S(C_m)}{S_{gt}} \tag{8}$$

At the line level, the i^th ground-truth text line is claimed to be segmented correctly only if the i^th ground-truth text line and the corresponding detected one share at least 90 percent of the pixels with respect to both of them [6]:

$$\frac{MS\left(i, C_m(i)\right)}{\sum_{j=1}^{N_m} MS\left(i, j\right)} \geq 0.9 \tag{9}$$

and

$$\frac{MS\left(i, C_m(i)\right)}{\sum_{k=1}^{N_m} MS\left(k, C_m(i)\right)} \geq 0.9 \tag{10}$$

The overall detection rate DR and recognition accuracy RA at line level are defined as:

$$DR = \frac{N_{match}}{N_{gt}} \tag{11}$$

$$RA = \frac{N_{match}}{N_d} \tag{12}$$

Where N_{match} is the number of text lines which satisfy (9) and (10).

IV. EXPERIMENTAL RESULTS

The proposed text line segmentation is tested mainly on HIT-MW [8], a freestyle, unconstrained Chinese handwritten text image database written by multiple writers. In this database, we can find some cases such as skewed, overlapping and touching text line that are very common in typical handwritten documents. Totally, it contains 853 handwritten samples containing 8664 text lines with ground truth at pixel level for each image, and no image includes any non-text content.

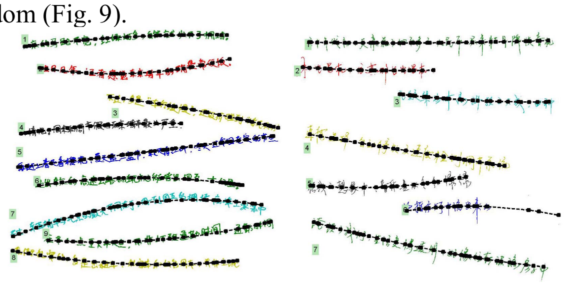
Based on the evaluation method described above, our algorithm has gained a high accuracy. The overall correct rate is 99.71% at pixel level while at line level the detection rate and recognition accuracy are 98.68% and 98.76%, respectively.

TABLE I. COMPARATIVE RESULTS OVER THE HIT-MW DATASET

Method	DR(%)
Docstrum	65.38
Piecewise projection based	92.07
MST clustering(with post processing)	98.02
Block-based	98.34
our method	98.68

A table of segmentation results by other method [9] and our method is given to illustrate the predominance of ours. In general, the proposed method outperforms the previously reported ones over the HIT-MW dataset. Besides, unlike other methods which need large number of training data to determine the parameters, in our method the key parameters (e.g. the size of character) are gained by the estimations. Moreover, the proposed method shows impressive performance in the cases that the text lines have overlapping components, arbitrary skew angles or various-width spaces between them.

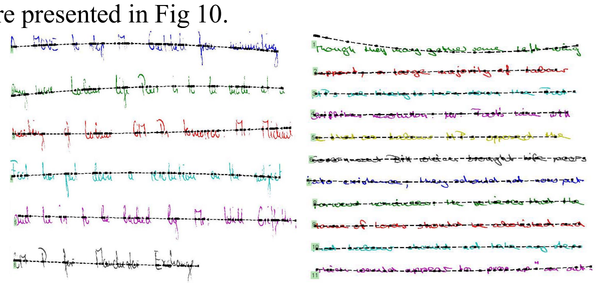
In order to demonstrate the robustness and adaptability of our algorithm, we collect a few samples with more writing freedom (Fig. 9).



(a) (b)

Figure 9. exapmles of documents with more writing freedom.

Additionally, we also test our method on the English handwritten document dataset to check its effectiveness on western-character handwritten documents. Unfortunately the detailed correct rate is not listed for there is no sufficient ground truth at pixel level for these dataset. A few examples are presented in Fig 10.



(a) (b)

Figure 10. exapmles of documents written in English.

We use the same parameter settings in all of these experiments mentioned in this paper.

V. CONCLUSION

In this paper an effective text line segmentation method for handwritten documents is proposed. It can solve the problems of text lines having various skew angles, touching or overlapping with each other. The algorithms in different scale are adaptively chosen to provide the segmentation results. Requiring no training, our method has proved its strong adaptability to various line conditions. We achieve satisfying segmentation results on plenty of samples using the same parameter settings. Experimental results show that the proposed method has obvious superiority over traditional methods.

ACKNOWLEDGMENT

This work was supported by the National Basic Research Program of China (973 program) under Grant No. 2007CB311004 and the National Natural Science Foundation of China under Grant Nos. 60872086.

REFERENCES

[1] Bar-Yosef. I, Hagbi. N, Kedem. K, Dinstein. I, “Line segmentation for degraded handwritten historical documents,” Proc. 10th ICDAR, pp.1161-1165, 2009.

[2] G. Louloudis, B. Gatos, I. Pratikakis, K. Halatsis, “A Block-Based Hough Transform Mapping for Text Line Detection in HandWritten Documents” Proc. 10th International Workshop on Frontiers in Handwriting Recognition, Oct. 2006.pp.515–520.

[3] F. Yin, Cheng-Lin Liu, “Handwritten Chinese text line segmentation by clustering with distance metric learning,” Pattern Recognition 42: 3169-383, 2009.

[4] A. Nicolaou, B. Gatos, “Handwritten Text Line Segmentation by Shredding Text into lines,” Proc. 10th ICDAR, pp.626-630, 2009.

[5] B.Gatos, N. Stamatopoulos, G. Louloudis, “ICDAR2009 Handwriting Segmentation Contest,” Proc. 10th ICDAR, pp.1393-1397, 2009.

[6] Y. Li, Y. Zheng, D. Doermann, S. Jaeger, “Script-Independent text line segmentation in freestyle handwritten documents,” IEEE Trans. Pattern Analysis and Machine Intelligence. 30(8): 1313-1329, 2008.

[7] G. Liu, R. M,Haralick, “Optimal Matching Problem in Detection and Recognition Performance Evaluation,” Pattern Recognition, vol. 4, no. 3, pp. 205-217, 2002.

[8] Tonghua. Su, Tianwen. Zhang, Dejun. Guan, “HIT-MW Dataset for Offline Chinese Handwritten Text Recognition”, Proc. 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, Oct. 2006.pp.515–520.

[9] Majid. Ziaratban, Karim. Faez, “An Adaptive Script-Independent Block-Based Text Line Extraction”, 20th International Conference on Pattern Recognition, pp.249-252, Istanbul, Aug. 2010.