

Segmentation and Analysis of Double-Sided Handwritten Archival Documents

Ruini Cao, Chew Lim Tan, Qian Wang, Peiyi Shen

School of Computing, National University of Singapore,
Lower Kent Ridge Crescent, Singapore, 119260
{caoruini, tancl, wangqial, shenpy}@comp.nus.edu.sg

Abstract. Historical handwritten documents are preserved in good condition in many national archives or libraries. One problem that many archivists are facing is the sipping of ink through the pages of certain double-sided handwritten documents after long periods of storage. This paper addresses this problem and develops a novel algorithm to extract clear textual images from interfering and overlapping areas. With the critical observation that the edges of the sipping strokes from the reverse side are not as sharp as those on the front side, we adopt the edge detection approach to suppress unwanted background patterns. Firstly, an improved Canny edge detector with edge orientation constraint is proposed. These improvements could link more weak foreground edges without introducing noises. Secondly, a new edge expansion model is presented for recovering broken edges of the words or characters on the front side. Finally, the outline of the whole document analysis system is illustrated. The segmentation results of real images are shown and evaluated.

1 Introduction

Documents have been the dominant information medium in the human society. This makes document image analysis an important research area in the field of image processing and pattern recognition. Traditionally, text extraction is the segmentation of text from the background. But this paper introduces a rather different problem, that is, how to extract clear text strings on the front side from the seriously sipping, dominating, overlapping and interfering images originating from the reverse side.

The motivation of this paper comes from a request from the National Archives of Singapore which keeps large amount of historical documents, quite a lot of which are double-sided handwritten documents with ink sipping through the pages through long periods of storage. The result is that the handwritten characters from the reverse side appear as noise on the front side and even interfere with the front side characters. As the original copies of these historical documents are carefully preserved and only binary or reduced gray-level microfilm images of these documents are directly available for public reading. Two original images and their corresponding binary microfilm copies are shown in Figure 1. We can see that reading of the contents in the binary copies is extremely difficult if not impossible. Thus, there is a request from the National Archives of finding some way to remove such interfering noises to produce readable copies for public viewers.

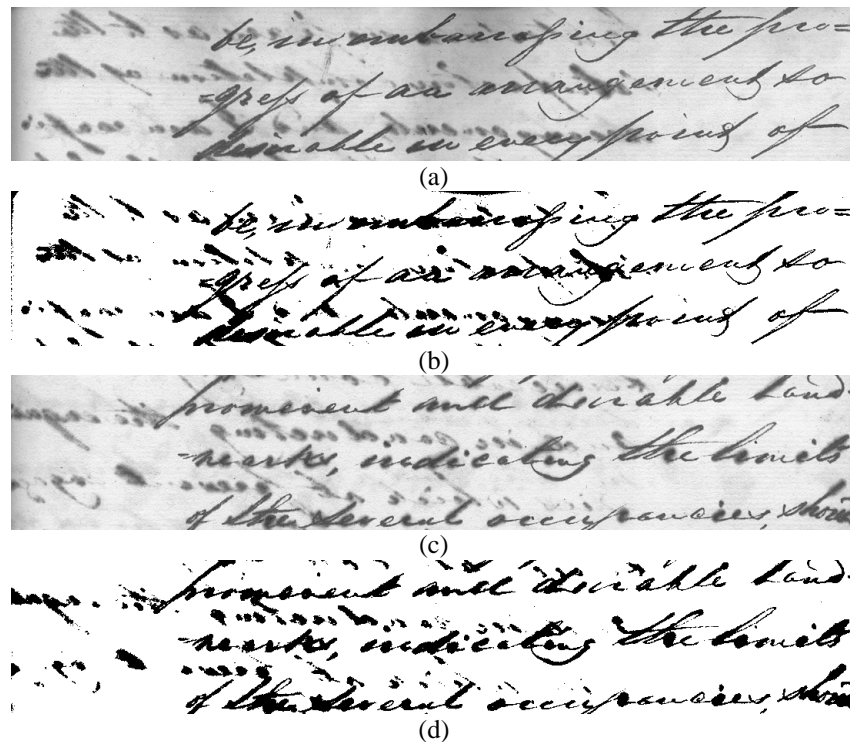


Figure 1. Two samples of the original archival documents (a, c) and their binary copies (b, d) (interfering, dimming and overlapping)

Many segmentation and binary approaches have been reported in the literature.^[1,2] In considering the increasing number of digital libraries on the internet that help researchers in humanities to compare old manuscripts with printed matters, Negishi et al.^[3] presented several automatic thresholding algorithms based on Otsu's method^[4] in extracting the character bodies from the noisy background. They dealt with terribly dirty and considerably large images, and cases where the gray levels of the character parts overlap with that of the background. Also, for complex background, Liu and Srihari^[5] presented a thresholding algorithm based on texture features to extract characters from the run-length featured texture background. Their proposed algorithm utilized two fundamental attributes of document images, in that, the characters normally occupy a separable gray-level range in the gray-scale histogram and that the text images contain highly structured-stroke units. Similar works could be seen in Liang and Ahmadi's algorithm^[6], which adopts a morphological approach to extract text strings from regular periodic overlapping text/background images. In their work different typical geometric background patterns were used as the mathematical morphological masks. White and Rohrer's^[7] method might be more traditional. It is basically an image thresholding technique based on boundary characteristics to suppress unwanted background patterns.

While the above efforts are indeed valuable and relevant, they cannot be applied directly to the present problem due to the varying degree of interference from the background. Firstly, the interfering strokes appear in varying intensities relative to the foreground text in different documents, in some cases, the background looks even darker than the foreground; Secondly, the edges of the foreground strokes are sometimes overshadowed by the interfering strokes rendering the foreground text almost unrecognizable even by human eyes (Figure 1. b & d). Thus a totally different approach has to be resorted for our archival documents. Due to the property of the paper material of these archival documents, it is observed that the edges of the strokes that sipped from the reverse side are not as sharp as those on the front side (Figure 1. a & c). This prompts us to adopt an edge detection algorithm followed by the use of boundary characteristics to suppress unwanted interfering strokes. The edge detection algorithm chosen here is an improved Canny edge detector^[8] as its double-threshold method could provide us the selection of the front stroke edges and its candidates.

The paper is organized as follows: In section 2, we describe the improved Canny edge detector with edge-orientation constraint to recover the weak edges of the foreground words and characters; In section 3, by analyzing the problems encountered in section 2, we present a new edge expansion model to restore the broken foreground edges in the overlapping area; In section 4, we illustrate the outline of the whole practical system; Section 5 discusses and evaluates the experimental results of the proposed method, demonstrating that our algorithm significantly improves the segmentation quality; Section 6 concludes this paper.

2 Improved Canny Edge Detector

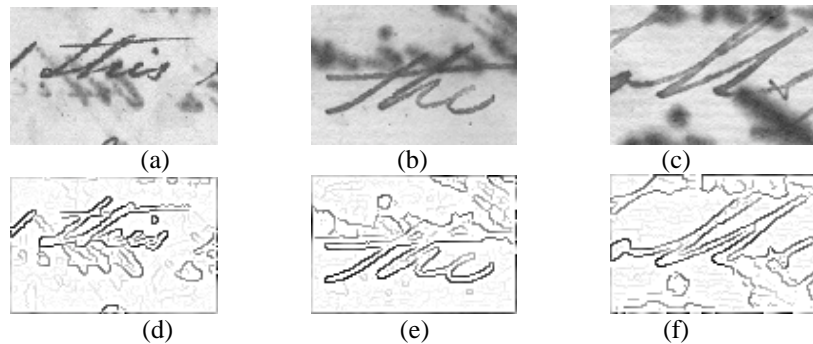


Figure 2. Sample images with different properties: (a) ~ (c) the original images; (d) ~ (f) the magnitude of all detected edges

Three typical samples of original documents are shown in Figure 2. It should be noted that, while usually, the foreground writing appears darker than the background image, as shown in sample image Figure 2(a), there are cases where the foreground and background have similar intensities as shown in Figure 2(b), or worst still, the background is more prominent than the foreground as shown in Figure 2(c). Therefore, using only the intensity value is not enough to differentiate the foreground from the background. Figure 2(d), (e) and (f) show the magnitude of the detected

edges of the three images respectively. The magnitude of the gradient is converted into the gray level value. The darker the edge is, the larger the gradient magnitude is. It is obvious that most of the top strong edges correspond to foreground edges. So, by double-thresholding the edges with hysteresis as in Canny edge detector, we can detect most of the edges on the front side.

2.1 Traditional Canny Edge Detection

Traditional canny edge detector is implemented as its double-threshold method could provide us the selection of the front stroke edges (high-thresholded edges) and its candidates (low-thresholded edges). The algorithms is described as follows (Figure 3):

1. Convert the scanned image into gray-scale, 8 bit.
2. Gaussian filtering.
3. Compute the gradient magnitude and orientation using finite-difference approximations for the partial derivatives.
4. Apply nonmaxima suppression to the gradient magnitude
5. High-level threshold edge detection
6. Low-level threshold edge detection
7. Edge linking: only the edges connected with the edges detected in step6 could be regarded as the candidates of linking.

Figure 3. Traditional canny edge detection

The above edge detection method works in detecting most of the foreground edges. The detected edges for the three samples (Figure 2) are shown in Figure 4.

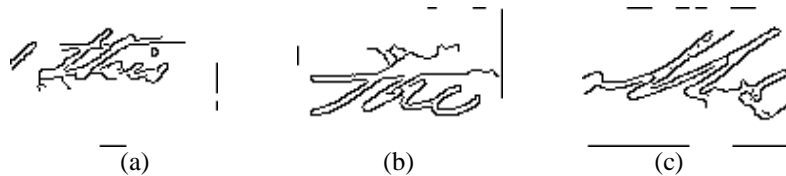


Figure 4. Traditional canny edge detection, (a) detected edges for the image shown in Fig. 2(a), high-level threshold = 0.980, low-level threshold = 0.900; (b) detected edges for the image shown in Fig. 2(b), high-level threshold = 0.973, low-level threshold = 0.820; (c) detected edges for the image shown in Fig. 2(c), high-level threshold = 0.972, low-level threshold = 0.820

Note that the algorithm fails when the interfering edges are present nearby and edge tracing is misled along the interfering edges. And in the overlapping area, the foreground edges are often broken and the resultant boundary is not complete.

2.2 Orientation Constraint

Since there exist many broken foreground edges in the overlapping area, it is necessary to pick them up in recovering the complete edges of the foreground characters. Unfortunately, traditional canny edge detector fails due to the interference of nearby edges^[9]. In many cases, in order to detect the “weak” foreground strokes in the seriously overlapping/overshadowed area, the low-level threshold often has to be traded off to favor the foreground edges over the interfering stroke. In the case that interfering strokes are “strong” enough to be detected in the low-threshold stage and adjacent with the seeds of the front edges, these “strong” interfering strokes would be regarded as the foreground edges in the final results. Thus, in practice, the low-threshold (*denoted as low-threshold1*) is often selected a little higher so as not to introduce too much “strong” interfering strokes and the result is that some “weak” foreground edges, especially in the overlapping area, are often lost.

In order to connect more weak foreground edges and reduce the risk of linking noisy edges detected in the low-level threshold stage, we lower the low-threshold (*denoted as low-threshold2*) and superimpose orientation constraint on edge linking. This is based on the observation that the transition between the orientation of the foreground edge and that of the “strong” interfering strokes is not smooth. The orientation constraint facilitates selecting the appropriate candidate fragment of the foreground edges in a smooth stroke, and at the same time prevents introducing the interfering strokes. It should be noted that, in the interfering cases, the maximum gradient intensity reflects the strength of the strokes, sharper or blurred; the orientation similarity reflects the smoothness property of a stroke. It is obvious that a character includes both smooth and non-smooth strokes. So the combination of the two strategies of *low-threshold1* and *low-threshold2* could be recommended to our problem.

The traditional canny edge detector is revised like this: adding three more steps onto the traditional canny edge detection algorithm (Figure 5).

8. New low-level threshold (*low-threshold2*) edge detection.
9. Edge linking: with the edges in step 6 being “seeds”, link the edges detected in step 8 with orientation constraint.
10. Combine the two edge images.

Figure 5. Improved linking strategy in canny edge detection algorithm

With this improvement, we could raise *low-threshold1* a little in step 6 of the traditional Canny edge detector. Accordingly, some interfering strokes will be successfully filtered out. And *low-threshold2* could be set at a rather low value in practice. In this way, more details of the front strokes will be recovered without introducing noises in the dimming area (Figure 6).

3 A New Edge Expansion Model

While edges are connected using the improved canny edge detector, there are other edge breakages that are many pixels apart whose gradient magnitude is weak,

especially in the overlapping/overshadowed area. Thus, an edge expansion model is here proposed to connect such broken edges. In this model, the Cartesian plane is divided into 16 sectors corresponding to 16 different edge extension directions as shown in Figure 7. First, to facilitate the processing of the edges, thinning algorithm is used. After the thinning operation, edge break points can be determined by checking whether an edge pixel has one and only one neighbor edge pixel.

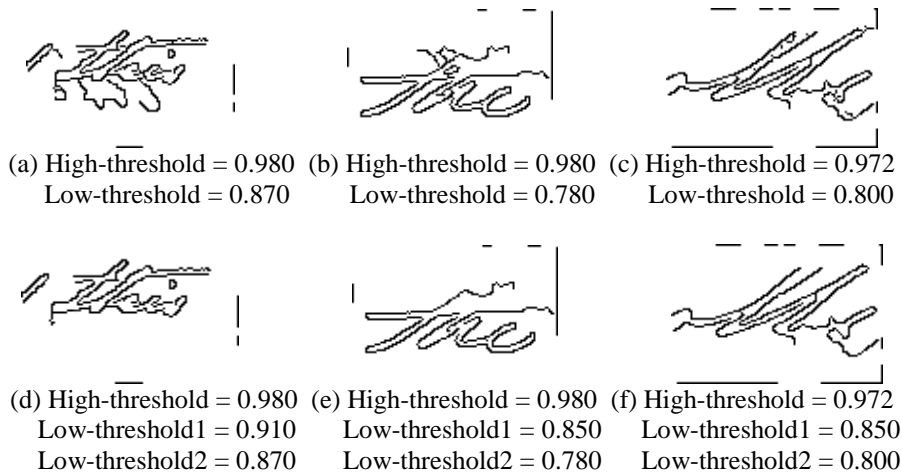


Figure 6. (a), (b) and (c): traditional Canny edge detection results; (d), (e) and (f): improved Canny edge detection results: orientation constraint, low-threshold1 is adopted in step 6, low-threshold2 is adopted in step 8

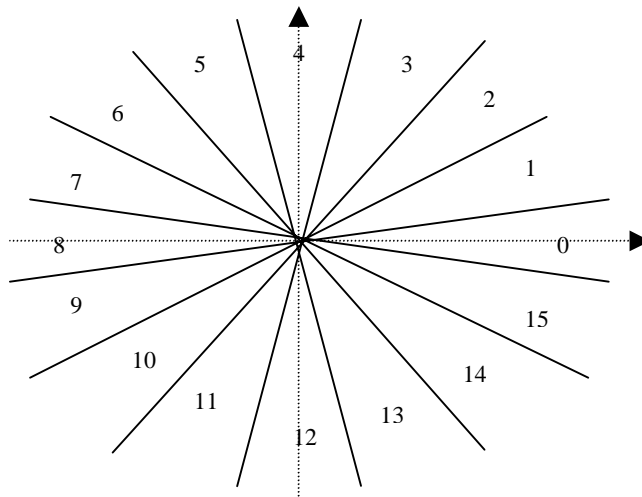


Figure 7. Cartesian plane with 16 angle regions

To determine the edge extension direction, the information stored in the neighboring pixels of a broken edge point may be used. As we know, after doing edge thinning, each broken edge point (for simplicity we call it P0) has one and only one neighbor pixel (P1). The way in which these two pixels are connected is useful but it may also mislead the edge extension work. What we should do is to take some more nearby pixels into consideration. Here two more neighboring pixels of P1 other than P0 (we call them P2 and P3) are picked and they are assigned a lower weight than P1 for their indirect connections with P0. In case P1 has only one neighboring pixel other than P0, one neighboring pixel of P2 is picked as P3 (Figure 8).

The detailed computation for the edge extension direction is like this: first the directions from P2 to P0 and from P3 to P0 are calculated and the average direction is recorded as D1. Next compute the direction from P1 to P0, which is recorded as D2. The average of D1 and D2 is used to estimate the extension direction. This computation assigns a higher weight to the direction from P1 to P0 compared to the other two directions.

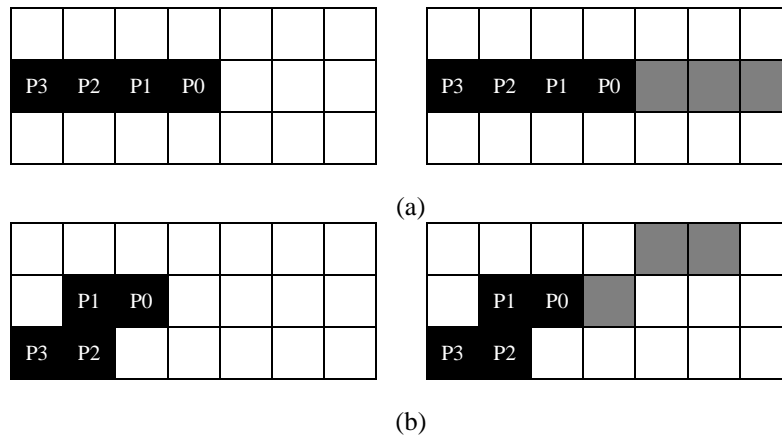


Figure 8. Two edge extension examples

The next step is to add new edge pixels based on the edge extension direction. For 16 different directions, there are 16 different patterns to be added, which are shown in Figure 9, and the black pixel is the original edge break point. For each extension direction, the pixel pattern to be added consists of three new pixels. Two examples (shown in Figure 8) illustrate the use of pattern table to do broken edges extension. In example Figure 8(a), a piece of edge falls on one straight line. Directions from P3 to P0, P2 to P0 and P1 to P0 are all 0, thus the average direction is 0. Pattern for direction 0 is chosen and added on the image. On the other hand, example Figure 8(b) is an edge, which looks like part of a curve. Directions from P3 to P0, from P2 to P0 and from P1 to P0 are 1, 2 and 0 respectively. So the approximate direction should be $((1 + 2) / 2 + 0) / 2 \approx 1$ and pattern 1 is used for edge extension.

The experiment results of the use of the edge expansion model are shown in Figure 10. We can see that the resultant edges are more complete.

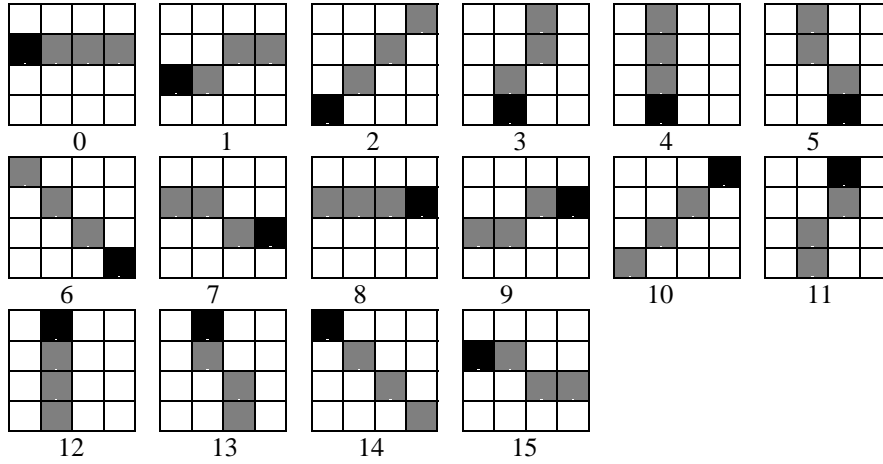


Figure 9. Patterns added for different edge extension directions

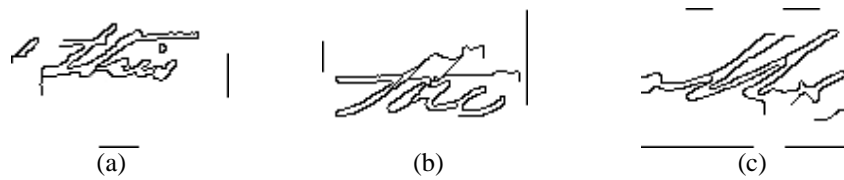


Fig. 10 Edge recovery based on the edge expansion model

4 The Outline of the Practical System

The essential aim of our project is to obtain clean and readable copies for public viewing of the archival documents. The outline of the whole practical system is shown in Figure 11. The image recovery stage is illustrated in Figure 12, which restores the original front side image based on the detected edges. The neighboring pixels within the 7x7 window centered on each edge are recovered. The size of the window being set at 7 is based on the statistic width of the strokes in the documents. Though isolated fragments of the foreground edges still exist after the above-mentioned edge expansion and connection, they could still be remedied in this recovery stage. The restored front side images for the three samples are shown in Figure 13.

5 Experiment Observation and Discussion

The performance of our approach has been evaluated based on the scanned images of historical handwritten documents provided by the National Archives of Singapore. The images were scanned in 150 dpi and saved in TIF format without compression. The cleaned up images were visually inspected to assess the readability of the words extracted. Here, 12 typical images are adopted in illustrating the performance of the system. The two evaluation metrics: precision and recall are used to measure the performance of the system. Precision and Recall are defined as follows:

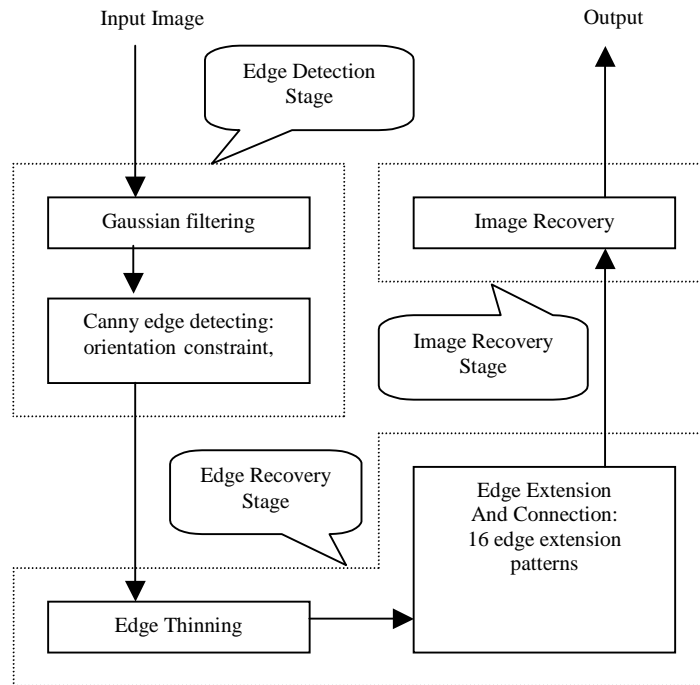


Figure 11. The flowchart of the document image segmentation system

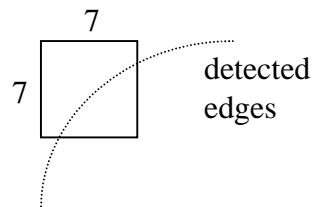


Figure 12. Restoration of the foreground image

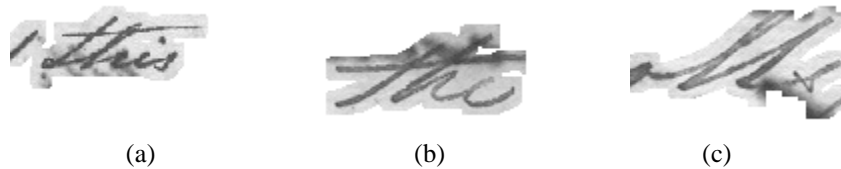


Fig. 13 Restored foreground text images: (a), (b) and (c) segmented result for the images Fig. 2(a), (b) and (c) respectively

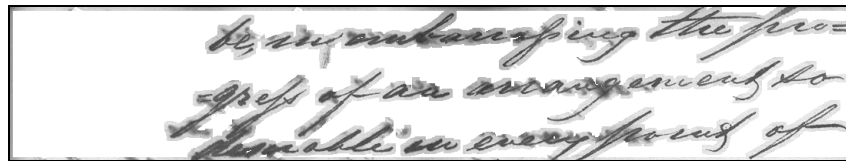
Precision = Number of Detected Correct Words / Total Number of Detected Words,
Recall = Number of Detected Correct Words / Total Number of Words.

where *total number of words* includes all the words in the foreground image, while the *total number of detected words* means the sum of the detected correct words and incorrect words (interfering words). If some characters in a foreground word are lost or not recovered properly, the whole word is considered lost. If parts of characters coming from the back are available, the total number of incorrect words will be increased by 1. Precision reflects the performance of removing the interfering strokes of the system and recall reflects the performance of restoring the foreground words of the system. The higher the precision, the lesser is the number of detected interfering strokes. The higher the recall, the more the foreground words are detected.

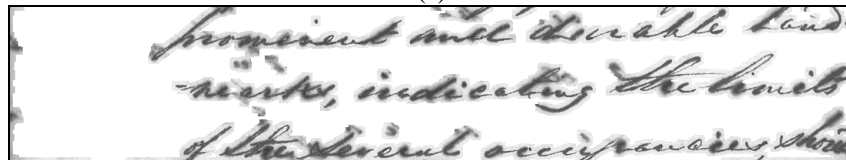
Table 1 shows the evaluation of the segmentation results of the 12 testing images. The average precision and recall is 85.7% and 100% respectively. In practice, the *low-threshold2* can be selected lower so that the foreground words can be completely detected without introducing noises. All the foreground words are detected and thus the recall is 100%. In fact, sometimes the interfering stroke is stronger than that of the foreground such that the edges of the interfering strokes would be erroneously regarded as the front “seed”. So the average precision is just 85.7%. Therefore, how to separate the front “seed” from all the high thresholded edges is still a problem in our further work. Another problem is that the fragments of the interfering strokes still exist in the segmentation results. Further processing in removing these fragments will be studied.

Table1. Evaluation of the system in 12 testing images

Image number	1	2	3	4	5	6	7	8
Total words	15	14	10	16	18	10	12	15
Precision	100%	77.8%	62.5%	100%	85.7%	100%	80%	93.7%
Recall	100%	100%	100%	100%	100%	100%	100%	100%
Image number	9	10	11	12				Average
Total Words	11	10	12	18				
Precision	78.6%	50%	100%	100%				85.7%
Recall	100%	100%	100%	100%				100%



(a)



(b)

Fig. 14 Segmentation results of the test images shown in Fig. 1

The segmentation results of the original images (shown in Figure 1) are shown in Figure 14. Since most of the interfering area could be removed successfully, the Ostu's threshold is adopted to binarize the segmentation images. The binary images of the segmentation results are shown in Figure 15. The binary images of other segmentation results are shown in Figure 16. And we can see that the appearance of the binary images is much cleaner and the foreground strokes are more readable.

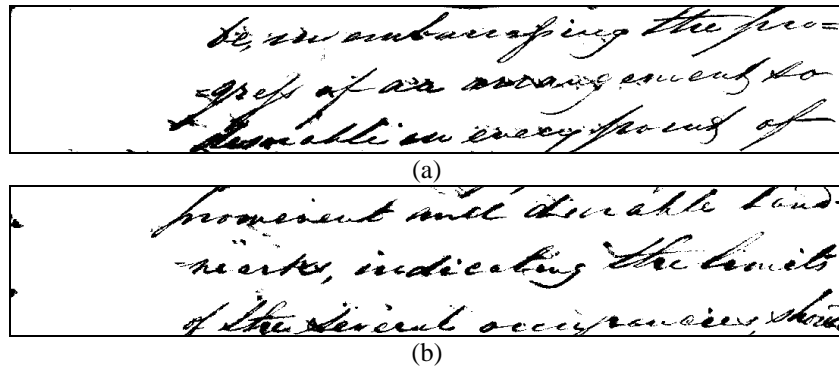


Fig. 15 The binary images of the segmentation results shown in Fig. 14.



Figure 16. (a), (b), (g), (h): original test images; (c), (d), (i), (j): the binary microfilm copies of (a), (b), (g), (h) separately; (e), (f), (j), (k): the binary images of segmentation results by the proposed algorithm of (a), (b), (g), (h) respectively, Ostu's threshold method⁴.

6 Conclusion and Further Work

The paper describes a method for the removal of interfering images. This method is especially designed for old handwritten documents by using the fact that the edges of the interfering images caused by ink sipping from the reverse side are not as sharp as those of the foreground images. The algorithm performs well and can improve the appearance of the original documents greatly. Currently we are working on the development of a local adaptive threshold method to restore the foreground image from the resultant segmentation images. Another approach will be also attempted. That is to superimpose the mirror-image rendition of the reverse page image with the front page image. In this way, corresponding mapping of the strokes of the two superimposed images will aid in identifying characters from either side of the paper.

References

1. O. D. Trier and A. K. Jain, Goal-directed evaluation of binarization methods, *IEEE Trans. on PAMI*, Vol. 17, No. 12, Dec. 1995, pp. 1191-1201
2. M. Thulke, V. Margner, and A. Dengel, A general approach to quality evaluation of document segmentation results, *The 3rd IAPR Workshop on Document Analysis Systems, DAS'98*, Nagano, Japan, November 1998, Springer-Verlag Berlin Heidelberg, Seong-Whan Lee, Yasuaki Nakano (Eds.), pp. 43-57
3. H. Negishi, J. Kato, H. Hase, and T. Watanabe, Character extraction from noisy background for an automatic reference system, *Proc. of the 5th Int. Conf. on Document Analysis and Recognition, ICDAR'99*, Sep. 1999, Bangalore, India, the Printing House, pp. 143-146
4. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. System, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66
5. Y. Liu, S. N. Srihari, Document image binarization based on texture features, *IEEE Trans. on PAMI*, Vol. 19, No. 5, May 1997, pp 540-544
6. S. Liang, M. Ahmadi, A morphological approach to text string extraction from regular periodic overlapping text/background images, *Graphical Models and Image Processing, CVGIP*, Vol. 56, No. 5, Sep. 1994, pp. 402-413
7. J. M. White, G.D. Rohrer, Image thresholding for optical character recognition and other applications requiring character image extraction, *IBM J. Res. Dev.* 27(4), 1983, pp. 400-410
8. J. Canny, A computational approach to edge detection, *IEEE Trans. on PAMI*, Vol. 8, No. 6, Nov. 1986, pp. 679-689
9. S. Casadei, S. K. Mitter, A hierarchical approach to high resolution edge contour reconstruction, *Proc. of IEEE Conf. On Computer Vision and pattern Recognition, CVPR'96*, June 18-20, 1996, San Francisco, California, IEEE Computer Society Press, pp. 149-154