# Detecting Text Lines in Handwritten Documents

Yi Li, Yefeng Zheng, and David Doermann
Laboratory for Language and Media Processing
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742-3275
E-mail: {liyi, zhengyf, doermann}@umiacs.umd.edu

## Abstract

*Although detecting text lines in machine printed documents is typically considered a solved problem, it is still a challenge to segment handwritten text lines in the general sense given no prior knowledge of script. This paper models text line detection as an image segmentation problem by enhancing text line structures using a Gaussian window and adopting the level set method to evolve text line boundaries. Experiments show that the method, which is script independent, achieves high accuracy for detecting text lines in heterogeneous handwritten documents.*

## 1 Introduction

Text line detection is a major component in a document image analysis system, which provides information for skew correction, zone segmentation, and character recognition. Algorithms for detecting text lines in machine printed documents have been explored and successfully adopted in many practical applications [1]. It is not easy however to extend these algorithms to handwritten documents where text lines are connected and close to each other (Fig. 1a). In some work on handwritten document image analysis, the authors typically assume that text lines are segmented, or the gap between two neighboring text lines is large enough that the conventional methods can be applied [2, 3]. In this section, we first review some previous work on text line detection in machine printed documents and the corresponding extensions for handwritten documents. Previous work on text line segmentation (or more general, layout analysis) can be roughly categorized as bottom-up or top-down. As a bottom-up approach, the connected component based methods [4, 5] merge neighboring connected components using a few simple rules based on the geometric relationship between neighboring blocks, such as distance, overlap, and size compatibility. Although they can process complex layouts, these methods are sensitive to topological changes of the connected components. For example, noise or handwritten strokes may cross two neighboring text lines making them inseparable at the connected component level. Another disadvantage is that they may be script dependent. For example, there is a space between neighboring words in English, but a Chinese text line is composed of a string of characters without word spaces. Therefore, it is not easy to derive script independent merging rules based on connected components. Projection based methods, such as X-Y cuts [6], may be one of the most successful top-down algorithms for machine printed documents. Because the gap between two neighboring text lines is typically significant, the projection of text lines is easily separable in the horizontal or vertical directions. However, these methods cannot be used in handwritten documents directly, unless gaps between two neighboring handwritten lines are significant or handwritten lines are straight [3]. Another disadvantage of the top-down approaches is that they cannot easily process complex non-Manhattan layouts.

Most text line segmentation approaches for machine printed documents are based on the assumption that text lines are straight. This fails in handwritten documents containing curvilinear text lines. Some approaches, such as the projection based methods, can be extended to deal with curved text lines. Tripathy and Pal [7] divide the image into multiple columns, use a projection based method in each column, and combine the results of adjacent columns into longer text lines. Generally, a better result is achieved compared with the naive projection method, but the results of two adjacent columns may be ambiguous so that it is still difficult to achieve a consistent result. Other researchers proposed different assumptions for their specific tasks [8] or for a specific script [9], but their methods are not general enough to process heterogeneous handwritten documents.

The support of this research by the US Department of Defense under contract MDA-9040-2C-0406 is gratefully acknowledged.

### 1.1 Overview of Our Approach

To make the problem tractable, we assume that the orientation of text lines are piecewise uniform, and skew correction can deskew the text lines, although a relatively large variation in orientation, e.g., up to 30°, is still allowed. Zone segmentation [10] may be necessary to separate text lines with significant different orientations (e.g., horizontal versus vertical) into different zones if they appear on the same document.

Fundamentally, text line detection is an image segmentation problem. Therefore, we model handwritten text lines in a general image segmentation framework as follows. We first enhance the text line structures by smoothing with a Gaussian window to convert a binary image to gray-scale, and then evolve an initial estimate of text line boundaries using the level set method [11]. In this paper, we make the following contributions:

1. Unlike previous methods, we convert the original binary image to gray-scale to enhance the text line structures.

2. Working on a smooth gray-scale image, we can leverage a state-of-the-art image segmentation approach, the level set method. With the priori knowledge that a text line is a horizontally elongated shape, we force the text line boundary to grow faster in the horizontal direction.

3. Our approach combines the advantages of the bottom-up and top-down approaches, i.e., script independence of the projection based methods and capability of processing a complex layout of the connected component based methods.

## 2 Level Set Based Detection

In this section we model text line detection as an image segmentation problem, and introduce the level set method [11].

### 2.1 Text Line Structure Enhancement

Previous algorithms work directly on a binary image. In our approach, we use Gaussian smoothing to convert a binary input to a gray-scale image, and a rectangular window is adopted. The height of the window is set smaller than the average height of the vertical gap between neighboring text lines to avoid blurring the gap. The width of the window can be set relatively large, so the horizontal gap between neighboring words or characters is blurred. The resulting image, $IM_{avg}$, has a low pixel intensity (dark) on the text lines, but a high intensity (light) for the vertical gaps (as shown in Fig. 1b). Therefore, the text line structures are enhanced. By smoothing the details of the handwritten strokes, we can achieve script independence. Since the resulting gray-scale image is smooth, gradient-based image segmentation methods can be applied.

Since the line structures are enhanced on $IM_{avg}$, we may simply binarize it to obtain an initial estimate of the text lines, $IM_{initial}$. The text lines we obtain here, however, may be fragmented (Fig. 1c). Therefore, we need to grow these partial text lines to generate longer ones. Since the level set method is an excellent tool for boundary evolution, which deals with topological changes naturally, it is well-suited for our application.

### 2.2 Text Line Boundary Evolution

Based on a partial differential equation (PDE), the level set method is an effective algorithm in image segmentation. The basic idea is to evolve the boundary by its partial derivatives and an external vector field. A general formulation [12] is as follows

$$\frac{\partial f}{\partial t} + \vec{S} \cdot \nabla f + V_N |\nabla f| = b_k |\nabla f|, \tag{1}$$

where $\frac{\partial f}{\partial t}$ denotes the move of the boundary frontier and it depends on the external vector field $\vec{S}$, gradient $\nabla f$, speed $V_N$ in the normal direction, and curvature $b_k$.

To evolve the text line boundary, we need an initial estimate of the text lines. The initial estimate need not be very accurate, and as long as at least one part of a text line is extracted, we can achieve a good final segmentation result. We use the binarization result of the smoothed gray-scale image, $IM_{initial}$, to initialize the level set method. The closed boundary of a text line is represented with an implicit function, which has negative values inside, positive values outside, and the zero value for the boundary. We then evolve the initial zero level set according to the PDE, so it can grow, merge, and stop automatically.

In our experiments, $IM_{avg}$ serves as the normal speed $V_N$, while the external vector field is set to zero. Using $IM_{avg}$ as the speed, the boundary will grow faster inside the text lines, where black pixel densities are large, while slower when it approaches to gaps. So, $IM_{avg}$ actually controls the speed of the boundary. In addition, we can force the boundary to grow faster in the horizontal direction by setting the horizontal grid distance larger than the vertical one. Curvature is also adopted as another factor in evolving the boundary. By our assumption, text lines are horizontal, therefore curvatures of left and right ends of a text line are larger than those of top and bottom, which means the text line tends to grow faster in the horizontal direction. Generally, we only need to evolve the boundary 50 iterations to

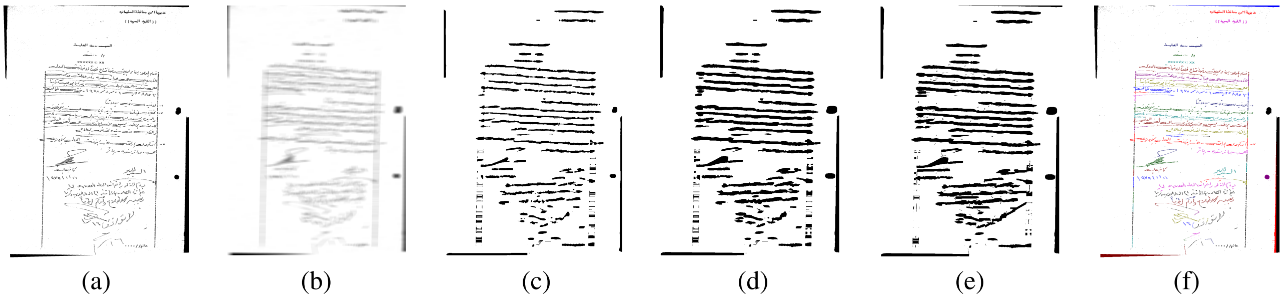


**Figure 1. Illustration of the proposed text line segmentation method. (a) A handwritten document; (b) Smoothed by a Gaussian window; (c) Initial estimate of text lines; (d) Result after 50 iterations of evolving using the level set method; (e) After connecting a few broken text lines in post-processing; (f) The final text line detection result.**

achieve a good result. For details of the level set method, please refer to [12].

### 2.3 Text Line Refinement

After growing the initial estimate of text lines, most text lines are segmented correctly. A few lines may be broken into several segments due to the large horizontal gaps between neighboring words. Since the number of broken segments is small and the line segments are often quite long, it is easy to group them in a post processing step as follows. We calculate the orientation of each line segment using the minimal mean squared error. We then merge the line segments if their orientations are compatible and their horizontal gap is less than a threshold. By the end of this step, we obtain the major text line structures (Fig. 1e). Finally, we group the isolated connected components to their closest major text line if the distance is not too large. Otherwise, we label those far from any major text line as noise. Fig. 1f shows an example of text line detection using different colors to distinguish neighboring text lines.

## 3 Experiments

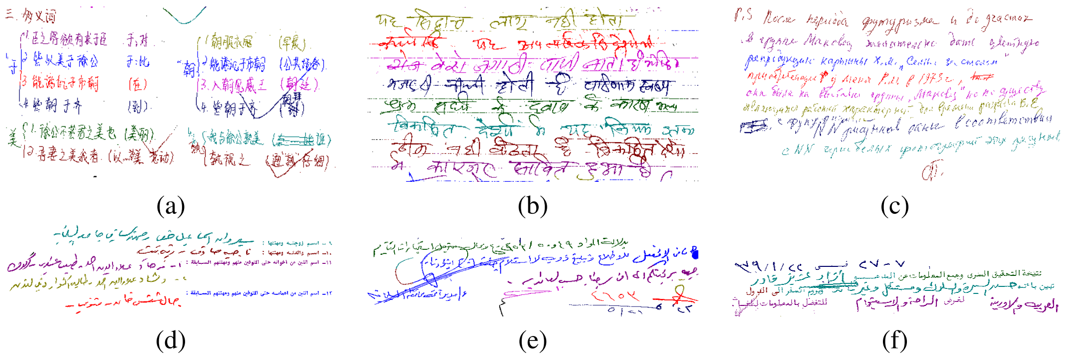
We tested our algorithm on more than 1000 heterogeneous handwritten documents in different scripts, such as Arabic, Hindi, and Chinese. We find our approach is robust, and good results are achieved for documents with different characteristics. Some results are shown in Fig. 2.

For a machine printed document, the text line segmentation results are often compared to the ground truth based on the bounding boxes. However, this simple approach does not apply for handwritten documents where text lines are curvilinear. Therefore, a polygon is used to represent the segmentation ground-truth. We have ground-truthed 100 handwritten Arabic documents.

To find the correspondence between the detection and ground-truth, we calculate the number of shared pixels for each pair of detected and ground-truthed text lines. The Hungarian algorithm is used to find one-to-one correspondence between the detected and ground-truthed text lines by maximizing the number of shared pixels [13]. We can evaluate the performance based on the hit rate, which is defined as the number of shared pixels normalized by the total number of pixels of the ground-truthed lines. Traditionally, the segmentation errors are classified as splits, merges, misses, and other complicated cases. By using the pixel-level hit rate and the Hungarian algorithm, different segmentation errors can be penalized by weighting by the number of involved pixels. We compare our approach with an improved connected component method [4], and the results are shown in Table 1. Comparing to the hit rate of 66% of the connected component based method, we achieve a much better result with a 92% hit rate.

We can also evaluate the performance at the text line level. If a ground-truthed line and the corresponding detected line share at least 90% pixels with respect to both of them, we claim the detection is correct. There are a total of 2,691 ground-truthed lines, and our approach can detect 2,303 (85.6%) of them correctly, as shown in the last column of Table 1. At the text line level, the connected component based method performs significantly worse. Only 951 (35.3%) text lines are detected correctly.

Figs. 2e and 2f show some failure examples. Most of the failures happen when two neighboring text lines are touched significantly. Other causes include signatures, the correction marks in the gap between two lines and the severe noise introduced while scanning step, etc. Since two lines are connected in only a few areas, post-processing may be exploited to segment them horizontally to improve the performance.



(a) (b) (c)

(d) (e) (f)

**Figure 2. Color examples of text line segmentation. (a) Handwriting with an irregular document layout; (b) Handwritten text lines with background rule lines; (c) Curvilinear handwritten text lines; (d) A document with mixed handwritten and machine printed text lines; (e) and (f) Failure examples.**

**Table 1. Quantitative comparison of our approach and a connected component based approach on 100 handwritten Arabic documents.**

| | Pixel Level Hit Rate | STD of Hit Rate | Detected Text Lines |
|---|---|---|---|
| Connected Component Based Approach [4] | 66% | 0.25 | 951 (35.3%) |
| Our Method | 92% | 0.04 | 2303 (85.6%) |

## 4 Conclusion and Future Work

We proposed a new approach for text line detection by adopting a state-of-the-art image segmentation technique. We first convert a binary image to gray scale using a Gaussian window, which enhances text line structures. Text lines are extracted by evolving an initial estimate using the level set method. Preliminary experiments show that our method is more robust comparing to a bottom-up connected component based approach [4]. We will compare our approach with the local projection based method [7]. Our method is language (or script) independent and this has been qualitatively confirmed by testing it on a few handwritten documents in several different languages, such as English, Chinese, and Korean. More quantitative evaluation experiments are necessary.

## References

[1] R. Plamondon and S. Srihari. Online and offline handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(1):63–84, 2000.

[2] U.-V. Martin and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In *Proc. Int'l Conf. Document Analysis and Recognition*, pages 159–163, 2001.

[3] R. Manmatha and J. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(8):1212–1225, 2005.

[4] K. Chen, S. Jaeger, G. Zhu, and D. Doermann. Doclib: A document processing research tool. In *Proc. Symposium on Document Image Understanding Technology*, pages 159–163, 2005.

[5] A. Simon, J.-C. Pret, and A. Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(3):273–277, 1997.

[6] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–12, 1992.

[7] N. Tripathy and U. Pal. Handwriting segmentation of unconstrained Oriya text. In *Int'l Workshop on Frontiers in Handwriting Recognition*, pages 306–311, 2004.

[8] U. Pal and P.P. Roy. Multioriented and curved text lines extraction from Indian documents. *IEEE Trans. System, Man and Cybernetics – Part B: Cybernetics*, 34(4):1676–1684, 2004.

[9] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane. Arabic hand-written text-line extraction. In *Proc. Int'l Conf. Document Analysis and Recognition*, pages 281–285, 2001.

[10] J. Sauvola, D. Doermann, and M. Pietikainen. Locally adaptive document skew detection. In *SPIE vol.3027*, pages 96–108, 1997.

[11] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2000.

[12] B. Sumengen. *Variational Image Segmentation and Curve Evolution on Natural Images*. PhD thesis, University of California, Santa Barbara, Sept. 2004.

[13] G. Liu and R.M. Haralick. Optimal matching problem in detection and recognition performance evaluation. *Pattern Recognition*, 35(3):2125–2135, 2002.