2009 10th International Conference on Document Analysis and Recognition

A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines

Zhixin Shi, Srirangaraj Setlur and Venu Govindaraju
Center for Unified Biometrics and Sensors
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260, U.S.A.

Abstract

In this paper, we present a new text line extraction method for handwritten Arabic documents. The proposed technique is based on a generalized adaptive local connectivity map (ALCM) using a steerable directional filter. The algorithm is designed to solve the particularly complex problems seen in handwritten documents such as fluctuating, touching or crossing text lines. The proposed algorithm consists of three steps. Firstly, a steerable filter is used to probe and determine foreground intensity along multiple directions at each pixel while generating the ALCM. The ALCM is then binarized using an adaptive thresholding algorithm to get a rough estimate of the location of the text lines. In the second step, connected component analysis is used to classify text and non text patterns in the generated ALCM to refine the location of the text lines. Finally, the text lines are separated by superimposing the text line patterns in the ALCM on the original document image and extracting the connected components covered by the pattern mask. Analysis of experimental results on the DARPA MADCAT Arabic handwritten document data indicate that the method is robust and is capable of correctly isolating handwritten text lines even on challenging document images.

1 Introduction

Text line extraction is a necessary step in any handwritten document recognition system. It segments a document image into distinct text lines to reveal the one dimensional natural reading sequence. The overall performance of a handwriting recognition system relies on the results of the process. Text line extraction is usually a straightforward process in machine printed documents but in the case of handwritten documents there exist many challenges. Some of the challenges are (i) variability in skew between different text lines, (ii) varying skew within a text line (fluctuating lines), (iii) overlapping text in pages with crowded writing where characters of adjacent text lines have overlapping bounding boxes, (iv) characters in one line touching text in adjacent lines, and (v) the presence of small symbols such as those seen in Arabic which float between text lines (see Fig. 1).

Generally text line separation algorithms first locate the lines and then segment them in their original logical order. A number of methods have been proposed in the literature. In the Projection Profile method[1, 2, 3] a histogram crossing an entire text block along a predetermined direction of the text lines is created. Then valleys that represent interline gaps are located to segment the text lines. Methods using Hough Transform are theoretically identical to the methods using projection profiles [4]. Along a set of selected angles straight lines are determined to fit the text elements with a measurement for the fit. The best fit gives the general skew angle and the location of the text lines. Another method uses nearest neighbor clustering of connected components [5]. Most of these approaches are designed mainly for machine printed documents. They are not directly adaptable to handwritten documents.

Unlike machine printed documents, handwritten documents have much more complex local structures. There have been prior methods designed for handwritten documents. Most of these methods segment the text lines by grouping the basic building elements of text such as pixels, connected components[6] or other structures including local minima detected from a chaincode structure [7]. The grouping algorithms are often based on heuristic rules[6], iterative learning algorithms[8] or searching in a tree structure[9]. Another local-global algorithm presented in [10] first partitions a document image into vertical strips. In each of these strips, the algorithm applies a projection profile algorithm with the assumption that the lines in a strip are almost all parallel to each other. This method deals with fluctuating or skewed text lines to some extent.

One of the problems in these methods is their depen-

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.79

Figure 1. Two sample handwritten Arabic document images: Image in the back includes text lines with variable skew and fluctuation, and the image in front shows a portion of a document with crowded text.

dency on isolation of the basic building elements such as strokes or connected components. When adjacent text lines touch each other, connected components that cross multiple lines have to be split, which is often difficult before the location of the text lines are available. An extreme example is that of Arabic handwriting written on lined paper. Very often one big connected component may consist of several lines of text. Another problem is that these methods generally take local decisions during the grouping process, and they sometimes fail to find the "best" segmentation when dealing with complex documents due to a tendency to be "trapped" by strong local features. When a document page includes very crowded writing, close neighboring connected components may not necessarily belong to the same line.

In this paper, we present a new text line extraction method for handwritten documents. The proposed technique is based on an *adaptive local connectivity map* (ALCM) generated using a steerable direction filter. The algorithm is designed for solving the particularly complex problems seen in handwritten documents including fluctuating, touching or crossing text lines.

Section 2 describes the steps involved in extracting text lines. Splitting of touching lines is also covered. Section 3 presents experimental results and Section 4 lists our conclusions.

2 Our Method

Humans are able to locate text lines in document images by detecting the text line patterns on reduced scale of the images. The touching or connections between text lines are sparse since they are usually made by oversized characters or characters with long ascenders or descenders running through the neighboring lines. On a reduced scale the line patterns appear distinct and the touching between lines loses prominence.

Based on these observations, an adaptive local connectivity feature was presented in [11] to change the scale of a document image. At each pixel, a connectivity measure is defined by cumulatively collecting its neighboring pixels' intensities along the horizontal direction. This connectivity measure can be intuitively understood to be the likelihood of a pixel belonging to a line. With the connectivity measure, the pixels in between lines are less likely to have an influence on the location of text lines.

In [12], a method was proposed using fuzzy runlength, in which a relaxed version of runlength computed for background pixels in a binary image was considered. The method emphasizes using background features in grouping and separating text lines. The method can efficiently extract text lines for complex documents including mixed objects of graphics, handwritten and printed text.

The methods described in both [12] and [11] cannot adequately handle fluctuating lines and lines with large skew.

The method proposed in this paper is a generalization of the adaptive local connectivity method presented in [11]. Instead of using a line segment filter along just the horizontal direction in generating the ALCM, we propose a generalized steerable directional filter. Using the new filter, local connectivity features are collected from multiple directions. The most likely local direction of the text line is captured by the maximum directional connectivity selected from the multiple directions.

Our method for text line location and extraction consists of the following steps. (1) Applying a steerable directional filter, we convert a down-sampled version of the input document image into an adaptive local connectivity map (ALCM), which is also a gray scale image. (2) We then apply a local adaptive thresholding algorithm on the ALCM to reveal the text line patterns in terms of connected components. (3) A grouping algorithm is used to easily group the connected components into location masks for each text line. (4) Extraction of the text lines is done by collecting the connected components corresponding to the location masks on the original binary document image. Small components are grouped into the spatially closest lines. In the case when a connected component touches more than one text line pattern in the ALCM, a splitting algorithm is applied to split the component into pieces and each of these is grouped with the closest text lines.

2.1 ALCM Using Steerable Filter

Let $f : R^2 \rightarrow R$ represent any given signal. The document image is the discrete version of this signal with the domain limited to $\{0, 1, ...n-1\} \times \{0, 1, ...m-1\}$ and values in $\{0, ...255\}$. Note that the image can be either binary or gray scale. Then, the adaptive local connectivity map is defined as a transform

$$\text{ALCM: } f \rightarrow A$$

by the convolution:

$$A(x, y) = \int_{R^2} f(x, y) G^{\theta_0}_{a,b}(x - t, y - s) dt ds \qquad (1)$$

where

$$G^{\theta_0}_{a,b}(x, y) = \begin{cases} 1 & \text{if } (x, y) \in E^{\theta_0}_{a,b} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where $E^{\theta_0}_{a,b}$ is an ellipse with semi-minor axis a, b and rotated by an angle θ_0:

$$E^{\theta_0}_{a,b} = \left\{ (x, y) | \{ \begin{array}{l} x < a \cos(\theta - \theta_0) \\ y < b \cos(\theta - \theta_0) \end{array} \text{ and } 0 \leq \theta < 2\pi \right\}$$

When we choose a longer than b, the ellipse $E^{\theta_0}_{a,b}$ is an elongated mask aligned with its long axis in θ_0 direction. Intuitively, using the steerable directional filter $G^{\theta_0}_{a,b}(x, y)$, ALCM is a convolution that aggregates the pixel intensities within the mask centered at (x, y). When the long axis of the filter is aligned with the direction of a text line, the ALCM value $A(x, y)$ at the pixel location inside the text line will be greater than the value along any other direction (see Fig. 2).

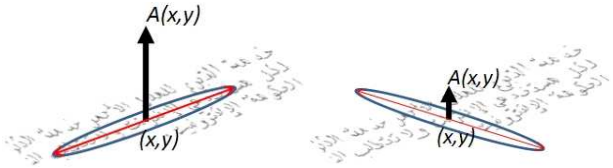

Figure 2. ALCM $A(x, y)$ using the same size filter aligned along different directions: Using the filter with direction along the direction of the text (left) the ALCM has a greater response (connectivity) than when using the filter in any other direction.

The implementation of the transform is as follows. For convenience, we first reverse the input image so that 255 represents the strongest level of intensity for foreground text. Most handwritten document images are scanned with resolutions ranging from 200 to 300dpi or higher. To determine the text line location, a lower resolution image is enough to retain the necessary information. We first downsample the image to 1/4 of its original size(1/2 in each direction). The size of the steerable filter in terms of a and b are generally chosen as follows. b is chosen to be a value less than the height of the text. a is chosen to be long enough to capture the location profile aggregate. Our experiments showed that 5 times of the text height is a reasonable estimate for a. The text height can be determined dynamically or by using a pre-set value. Our experiments found that our method can tolerate a large range of variation of a and b.

The direction parameter θ_0 plays a very important role in our method. The use of multiple directions in the filter allows the extraction of text lines with changing skew and fluctuation. In our experiments, for efficiency, we choose 5 directions – horizontal, slope ratio of 1 in 10 and 1 in 20 and their negations.

Finally, we re-scale the resulting ALCM values to a gray scale image with values ranging from 0 to 255 (see Figure 3).

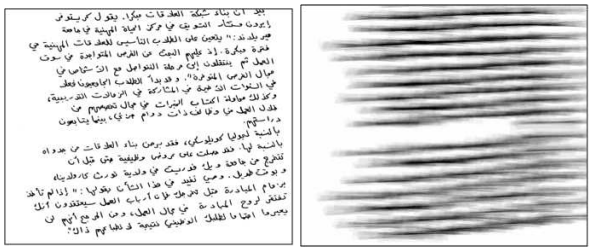

Figure 3. A portion of a handwritten Arabic document image with varying skew, and the ALCM generated on the same portion.

2.2 Location of Text Lines

Each pixel value in an ALCM image represents the cumulative intensity of the foreground pixels in an elliptical neighborhood around the pixel in the original document image. A pixel with higher value in the ALCM implies that the pixel is in a dense text region. We therefore binarize ALCM to two values for separation of highly-likely text areas from the background.

In [11] Otsu's global thresholding algorithm is used to binarize the ALCM and it has been shown that the algorithm works well for most English handwritten documents including historical manuscript images from the Library of Congress. However, in looking at handwritten Arabic document images, we have found that it is hard to locate distinctive text line patterns using any global thresholding algorithm on the ALCM due to crowded writing, variability in thickness of the strokes and irregular line spacing.

We have implemented a local adaptive thresholding algorithm similar to that presented in [13]. The algorithm determines a pixel's binary value by considering the pixel intensity distribution in a 5 neighborhood block structure.

The 5 neighborhood blocks are 5 $n \times n$ windows with one in the middle centered at the pixel under consideration and 4 other blocks adjacent to the corner of the center block. A weighted difference between the average pixel intensity in the middle block and that in the other 4 blocks is used to decide the center pixel's binary value. See [13] for the general algorithm. Our modification is in the implementation of the algorithm using configurable value n for the block size. Figure 4(a) shows the result of binarization using the adaptive thresholding algorithm.

The binarized ALCM image in Figure 4(a) consists of connected components which represent either the entire line or part of a line.

(a) (b)

Figure 4. Binarization of ALCM showing the patterns of text lines.

2.3 Extraction of Text Lines

The line patterns extracted from the ALCM are location masks for the actual text lines. The text lines in the original document image are extracted by collection and grouping of connected components. The connected components for the text in the original document image are generated. After up-sampling the line patterns in ALCM to the scale of the original image, we superimpose the line patterns on the document image. For each text line pattern, we collect all the connected components of text touching the pattern and these components together make up the text line.

If there are some connected components that do not touch any line pattern, they are grouped with the closest line. Figure 4(b) shows the line patterns that are superimposed on a document image and the text lines that are extracted are shown in different colors.

Some connected components may belong to more than one line pattern. These components represent characters that cross multiple text lines (see the red color components in Figure 5). Although these crossing pieces can be easily detected, it is a non trivial task to split them and group them with the lines that they belong to. To split the touching pieces we implemented the segmentation algorithm for splitting touching characters presented in [14]. For a touching piece, a reference line is drawn between the line patterns. The segmentation algorithm segments the contours of the piece into contour segments. Based on the location of the center of mass of the contour segments relative to the reference line, they are grouped into the corresponding text lines. The text images are recovered using the contour segments (see Figure 5).

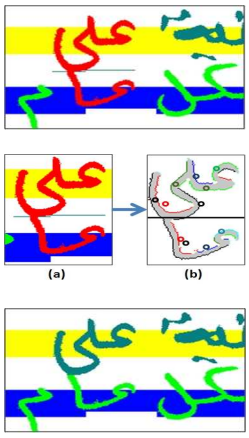

Figure 5. Splitting characters crossing multiple text lines. Top: Image showing detected multi-line character. Middle: Segmentation of the contours with the center of mass marked. Bottom: Character images are split and grouped with the closest text line.

3 Experiment

To test our method, we used a set of 45 randomly chosen handwritten Arabic document images from the DARPA MADCAT data. These images are written by multiple writers on blank paper, paper with pre-printed ruled lines and letterheads with company logos. The documents were scanned at 300dpi and binarized.

Various evaluation methods have been reported in the literature for line segmentation. Some use a manual verification of the results visually. Others use the number of pixels being included in a pre-defined region for each text line. For our performance evaluation, we apply a connected component based approach. Instead of defining the text lines in terms of regions covering the lines, we use connected components as the basic text objects. The number of connected components that are classified into the right text lines are counted for reporting the performance numbers.

The 45 pages in the test set contain a total of 1022 text lines. There were only two instances where two lines were incorrectly merged by the system. The total number of text connected components is 32,936. The number of incorrectly classified isolated components is 144. There are 178 connected components that touch more than one line. Among them, 14 pieces were incorrectly split and grouped. For the incorrectly split components, the error is counted twice. Therefore, in terms of correctly classified

connected components, the correct rate is $(32936 - 144 - 2 \times 14)/32936 = 99.5\%$. Evaluation on a larger data set is underway.

4 Conclusion

In this paper we present a novel method for extraction of text lines from complex handwritten documents. The method uses a new concept of steerable directional profile in building a generalized adaptive local connectivity map. Our experiments demonstrate the efficacy of this method.

5 Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency DARPA/IPTO (PLATO: A System for Taming MADCAT: Multilingual Automatic Document Classification Analysis and Translation) ARPA Order No. X103 Program Code No. 7M30 Issued through a subcontract from BBN Technologies Corp. under DARPA/CMO Contract # HR0011-08-C-0004.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency or the U.S. Government.

References

[1] T. Pavlidis and J. Zhou, "Page segmentation by white streams," *Proc. 1st Int. Conf. Document Analysis and Recognition (ICDAR),Int. Assoc. Pattern Recognition*, pp. 945–953, 1991.

[2] G. Ciardiello, G. Scafuro, M. T. Degrandi, M. R. Spada, and M. P. Roccotelli, "An experimental system for office document handling and text recognition," *Proc 9th Int. Conf. on Pattern Recognition*, pp. 739–743, 1988.

[3] S. C. S. G. Nagy and S. D. Stoddard, "Document analysis with expert system," *Procedings of Pattern Recognition in Practice II*, June 1985.

[4] S. N. Srihari and V. Govindaraju, "Analysis of textual images using the hough transform," *Machine Vision and Applications*, vol. 2, pp. 141–153, 1989.

[5] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.

[6] L. Likforman-Sulem and C. Faure, "Extracting text lines in handwritten documents by perceptual grouping," in *Advances in handwriting and drawing :a multidisciplinary approach*. C. Faure, P.Keuss, G.Lorette and A.Winter Eds, Europia,Paris, 1994, pp. 117–135.

[7] M. Feldbach and K. D. Tonnies, "Line detection and segmentation in historical church registers," in *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR '01)*. IEEE Computer Society, 2001, pp. 743–747.

[8] Y. Pu and Z. Shi, "A natural learning algorithm based on hough transform for text lines extraction in handwritten documents," in *Proceedings sixth International Workshop on Frontiers of Handwriting Recognition*, 1998, pp. 637–646.

[9] S. Nicolas, T. Paquet, and L. Heutte, "Text line segmentation in handwritten document using a production system," in *IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*. IEEE Computer Society, 2004, pp. 245–250.

[10] E. Bruzzone and M. C. Coffetti, "An algorithm for extracting cursive text lines," in *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*. IEEE Computer Society, 1999, p. 749.

[11] Z. Shi, S. Setlur, and V. Govindaraju, "Text extraction from gray scale historical document images using adaptive local connectivity map," in *ICDAR '05: Proceedings of the Seventh International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2005, pp. 794–798.

[12] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," in *DIAL '04: Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*. IEEE Computer Society, 2004, p. 306.

[13] E. Giuliano, O. Paitra, and L. Stringa, *Electronic character reading system*. U.S. Patent No. 4047152, Sep. 1977.

[14] Z. Shi and V. Govindaraju, "Segmentation and recognition of connected handwritten numeral strings," *Journal of Pattern Recognition*, vol. 30, no. 9, pp. 1501–1504, 1997.