Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map.

3 authors:

Zhixin Shi
University at Buffalo, The State University of New York
43 PUBLICATIONS **860** CITATIONS

SEE PROFILE

Srirangaraj Setlur
University at Buffalo, The State University of New York
51 PUBLICATIONS **536** CITATIONS

SEE PROFILE

Venu Govindaraju
University at Buffalo, The State University of New York
400 PUBLICATIONS **7,703** CITATIONS

SEE PROFILE

Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map

Zhixin Shi, Srirangaraj Setlur and Venu Govindaraju
Center of Excellence for Document Analysis and Recognition(CEDAR)
State University of New York at Buffalo, Buffalo, NY 14228, U.S.A.

Abstract

This paper presents an algorithm using adaptive local connectivity map for retrieving text lines from the complex handwritten documents such as handwritten historical manuscripts. The algorithm is designed for solving the particularly complex problems seen in handwritten documents. These problems include fluctuating text lines, touching or crossing text lines and low quality image that do not lend themselves easily to binarizations. The algorithm is based on connectivity features similar to local projection profiles, which can be directly extracted from gray scale images. The proposed technique is robust and has been tested on a set of complex historical handwritten documents such as Newton's and Galileo's manuscripts. A preliminary testing shows a successful location rate of above 95% for the test set.

1 Introduction

There exist numerous collections of handwritten historical manuscripts in libraries and museums around the world. They are a valuable resource for scholars and their indexing for archival and retrieval purposes is highly desired. Handwritten historical documents present many challenging research issues in document analysis and recognition. Among them, text line location and extraction are critical pre-processing for automated indexing systems using OCR.

Generally text line separation algorithms first locate the lines and then segment them in their original logical orders. A wide variety of methods have been proposed in the literature. The Projection Profile technique[1, 2, 3] creates a histogram crossing an entire text block along a predetermined direction of the text lines. Valleys are located for parallel straight lines in between text lines for separations. The methods using Hough Transform are theoretically identical to the methods using projection profiles [4]. Hough Transform is usually used for locating skewed text lines. It is applied on a set of selected angles and along each angle straight lines are determined with a measurement for the fit. The best fit for the lines gives the skew angles and the locations of the lines. Another method uses nearest neighbor clustering of connected components [5]. Most of these approaches are designed mainly for machine printed documents and provide good results on printed documents only. They are not directly adaptable to handwritten documents due to their reliance on global features in an image and are therefore more suited for well structured documents.

Unlike machine printed documents, handwritten documents have much more complex local structures with variations such as fluctuating text lines, skewed text lines, overlapping words and touching and/or crossing text lines. There are methods designed for text line segmentation for handwritten documents dealing with these difficulties. Due to the local rather than global structure characteristics in handwritten documents, the methods found in the literature for text line segmentation in handwritten documents are generally "bottom-up" and based on local analysis. 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]. One of the important concerns in the design of these methods is to overcome the fluctuation in text lines. 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 is based on local horizontal run projections and connected components grouping after partitioning a document image into vertical strips[10]. In each one 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.

There are two noticeable problems in these methods. One is their dependency on isolation of the basic building elements such as strokes or connected components. When adjacent text lines touch each other, connected components that crossing lines have to be split, which is often difficult before the text line locations become available. The second

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. Furthermore, most of these methods are designed for binary images. Due to the aging effect on historical manuscripts, most handwritten historical document images are of very low quality and sometimes difficult to read even by humans. It is very often also difficult to binarize the gray scale images while maintaining readability.

In this paper, we propose a novel text line location and extraction algorithm for a complex document. Unlike other line segmentation methods requiring a binary image, our method can be directly applied on gray scale images. The method can be considered a transform based method. For a gray scale handwritten document image, we first define a local projection profile at each pixel of the image and transform the original image into an adaptive local connectivity map. An adaptive local connectivity map (ALCM) is a gray scale image in which each pixel value at a pixel location represents a connectivity property for its neighboring pixels in the original document image. It is an accumulative integration of foreground pixel intensities. By thresholding the gray scale ALCM, clear text line patterns are revealed in the resulting binary image as connected components. The algorithm is designed primarily for handling difficult complex historical document images. The algorithm is also general enough to be used for any other types of images such as binary images, machine printed or even mixed script.

In Section 2 we provide a brief discussion of techniques inspiring our method. The basic principle of our approach will be described. In Section 3 the steps in locating text lines will be described in detail. Special care is taken in splitting touching lines. In Section 4 we will present experimental results and Section 5 lists our conclusions.

2 Background

Humans are able to locate text lines in document images relying on recognition of the geometrical layout features of the lines without recognizing character and understanding the document. Usually the text line patterns in an image can be easily detected by reducing the scale of the image. Reducing scale of a document image is equivalent to looking at the image from a farther distance. On a reduced scale the line patterns appear distinct and the touching between lines loses prominence. 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. Another observation of handwritten documents is that although the text lines may be fluctuating across an entire text block, there still exists a general orientation of a line.

Based on these observations, we use an adaptive local connectivity feature to change the scale of a document image. For each pixel we define its connectivity measure by cumulatively collecting its neighboring pixels' intensities along the horizontal direction. This connectivity measure can be intuitively understood as a popular measure of a pixel as the measure of how likely it is for a pixel to belong to a line. With the connectivity measure, the pixels in between lines are less likely to be influence of the location of text lines even if some of these pixels are part of the text that is between the lines.

In [11], a method proposed using fuzzy runlength, in which a relaxed version of runlength computed for background pixels in a binary image is considered. The method emphasizes using background feature 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 method proposed in this paper is inspired by several well known methods in the literature. Its cumulative measure is inspired by the projection profile method. Traditional projection profile method computs a global histogram over an entire text block. The computation is done on a binary image. Our connectivity measure is computed adaptively for each pixel in its neighborhood. The computation can be done on a gray scale image. Second method is runlength smearing. Runlength smearing[4] is usually used for tolerating noise and run-away black strokes. The desired runlength such as foreground runs are created by skipping small runs in background color. The expected result from this process is that the most of the foreground text characters are grouped together. The text lines and text blocks are extracted by using a connected component analysis approach. The method again works well for printed documents with mostly text. It will fail on documents with touching lines or connection between text lines and text blocks. The most inspiration part of the method is its local-global characteristic, which can be regarded as changing the scale of an image. Our method is also inspired by scale space techniques [12] in which non uniform Gaussian filers are used in convolutions on the image of a text line to create a smoothed/smeared image for extracting handwritten words. The filters are functions of two variables that act as scale parameters. By carefully choosing the parameters, the convolved image shows the separated words each as a distinct connected component.

3 ALCM Method

Our method for text line location and extraction consists the following steps. (1)We convert a gray scale document image into an adaptive local connectivity map (ALCM), which is also a gray scale image. (2)We then apply a

thresholding algorithm on the ALCM to reveal the text line patterns in terms of connected components. (3) Using a straightforward grouping algorithm we can easily group the connected components into location masks for each text line. (4) Finally, the text lines from a binarized version of the document image (using any standard thresholding algorithm) can be extracted by mapping the location masks back onto the binary image to collect the text line components. For those components touching multiple lines, a splitting algorithm is applied.

3.1 ALCM Transform

Let $f : R^2 \to R$ represent any given signal. Its discrete version with the domain limited to $\{0, 1, ...n - 1\} \times \{0, 1, ...m - 1\}$ and values in $\{0, ...255\}$ is our gray scale document image. Then the adaptive local connectivity map is defined as a transform

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

by a one-direction convolution:

$$A(x, y) = \int_R f(x, y)G_c(t - x, y)dt \qquad (1)$$

where

$$G_c(x, y) = \begin{cases} 1 & \text{if } |x| < c \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

The implementation of the transform is as follows. For convenience, we first reverse the input gray scale image so that 255 represents the strongest level of intensity for foreground text. Most handwritten historical images are scanned with resolution ranging from 200 to 300dpi. We first down-sample the image to 1/4 of its original size(1/2 in each direction). Then we scan the image along each of its scanlines twice, from left to right and right to left, using a sliding window of size $2c$, to compute the cumulative intensity at a pixel by adding up all the intensity values in a neighborhood of size $2c$. Finally, we re-scale the resulting image to a gray scale image with values ranging from 0 to 255.

In the above algorithm c is a value for determining the size of the sliding window prior to scanning the image for calculating ALCM transform. It can be set as a fixed value for an application running similar set of images or determined at run time dynamically. A good initial value of c for the sliding window would be 120 which is approximately equal to three time the average height of text. The average height of text in a document image could be roughly estimated using a projection profile on a portion of a text block or other connected component based methods, after a rough binarization. Our experiments found that our method can tolerate the variation of c in a quite large range.
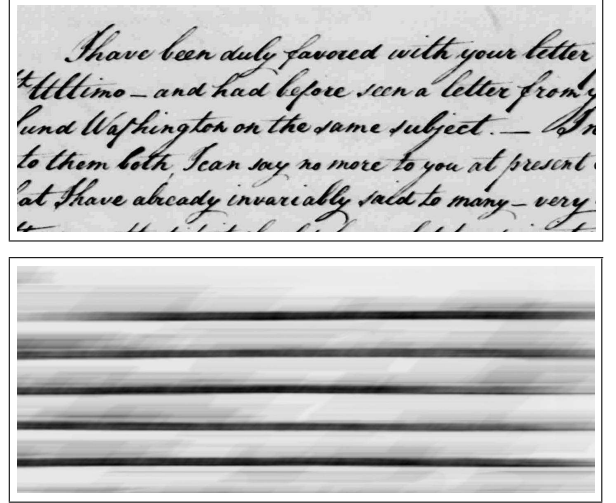

Figure 1. A portion of a handwritten manuscript from a letter by George Washington to William Milnor, and the ALCM generated on the same portion.

3.2 Locations of possible text lines

Each pixel value in an ALCM image represents the cumulative foreground pixel density for the original document image. As with a projection profile, a higher value in the ACLM means 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. Generally, binarization for document images is not an easy task especially when the document images are from handwritten historical manuscripts. Binarization for document images is used to not only separate the text in an image from its background but also keep the integrity of the writing for later recognition. Unlike document image binarization, binarization for our ALCM is much easier due to two reasons. Firstly, as a gray scale image, ALCM shows clear bi-modal pixel distribution most of the time, which allows for a global thresholding. Secondly, binarization for ALCM is for general patterns of text lines. Each pattern represents loosely a central location of a text line or part of a line. The binarization for ALCM is very tolerable for differences between results from different binarization algorithms. Figure 2 shows a binarization result using Otsu's algorithm [13].

The binarized ALCM image in Figure 2 consists of connected components wich represent either the entire line or part of a line. Instead of using the connected components to form the complete line pattern. We do some filtering and reconstruction as follows.

1. Filter out the small pieces. Based on an experiments, some small pieces whose width is significantly smaller
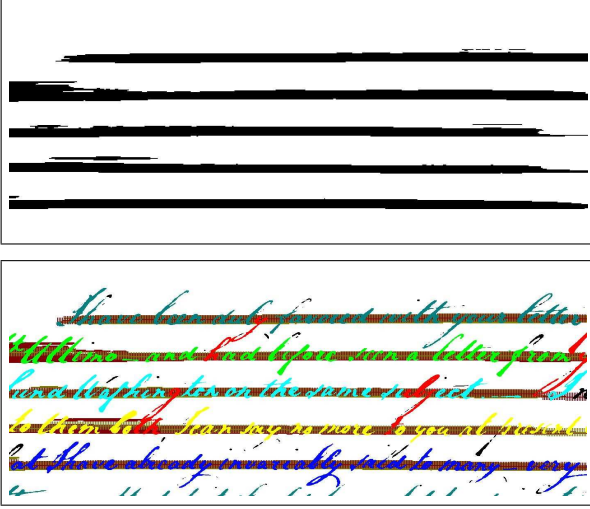
Figure 2. Binarization of ALCM show the patterns of text line.

than most other components. Usually they are also short. These pieces are redundant and should be filtered out.

2. For each connected component, we calculate its upper, lower profiles and also the center points. Filling in each pair of upper and lower profile points, we reconstruct the connected component replacing the original component.

In the ideal case, each connected component represents a complete text line. But sometimes a text line pattern is made up of 2 to 3 components and requires grouping. A straightforward approach would be using horizontal alignment and rules based on wether a grouping of two neighboring components is too wide for a line or not.

3.3 Extraction of Text

The text line patterns that we have extracted are location masks of text lines. The extraction of text from a gray scale image using these locations bring up two issues. The first is how we represent the text and how to get the text information. The second issue is dealing with touching characters crossing different lines.

Most document recognition systems require binary images of text lines, words and characters. However binarization of handwritten historical document images is still a challenge problem. Our proposed method for text line location in this paper suggests a possibility for local adaptive binarization. After we find the locations of the text lines, an adaptive binarization algorithm can be designed to focus on only the line locations. It allows us to not only concentrate

on a much smaller document region, but also avoid much of the non-text noise.

When a binarization is available for a document, extraction of the text lines can be done by a connected component collection and grouping. We first generate the connected component representation. For each text line pattern, we collect all the connected components touching the pattern, these components together make up the text line. If there are some connected components that do not touch any line pattern, we find the closest line pattern to group them with. Figure 2 shows the line patterns that are superimposed on top of a binary image and text lines in different colors found by collecting and grouping of the connected components.

Some connected components may belong to more than one line pattern. These components represent the touching characters crossing lines (see the red color components in Figure 2). Since these crossing pieces are easily detected, splitting them according to their relation to the nearby lines is trivial.

4 Experiment

To test our method, we used a set of 30 randomly chosen handwritten historical manuscript images downloaded from the Library of Congress. These images include Thomas Jefferson's manuscripts, George Washington's manuscripts and Samuel F. B. Morse manuscripts. Manual evaluationg of the results of our method on these images shows a correct location rate of 95%. An example is shown in Figure 3.
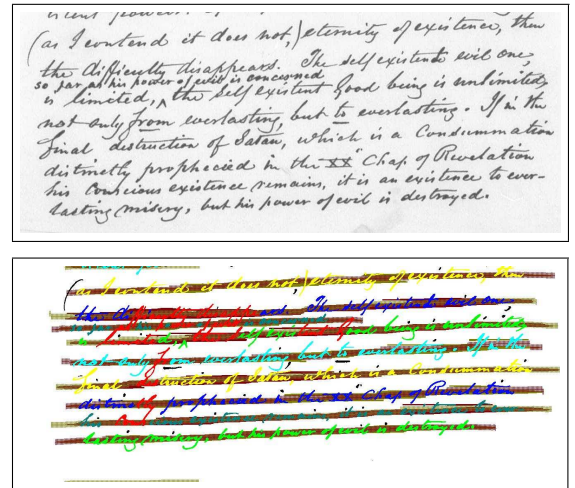

Figure 3. An example image from The Samuel F. B. Morse Papers at the Library of Congress. The image shows uneven lines with touching characters crossing the lines. Below is the result for showing the detected locations for the lines.

Among the failure cases, are images with severe damage, noise and low visual readability. One noticeable problem we would like to fix is the failure for some short text lines such as abbreviations, page numbers and salutes.

5 Conclusion

In this paper we present a novel method for complex handwritten document text line location. The method uses a new concept of adaptive local connectivity map which is a transform of an input image for re-scaling of the document. Our simple experiments demonstrate the success of the method. Further research includes using the method for document segmentation and other content location.

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. Degrandi, M. 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. S. G. Nagy and S. Stoddard, "Document analysis with expert system," *Procdings of Pattern Recognition in Practice II*, June 1985.

[4] S. 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 Anal. Mach. Intell.*, 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. Tönnies, "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 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.

[12] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," in *SCALE-SPACE '99: Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*. Springer-Verlag, 1999, pp. 22–33.

[13] N.Otsu, "A threshold selection method from gray level histogram," *IEEE Transactions in Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, 1979.