TEXT LINE EXTRACTION FOR HISTORICAL DOCUMENT IMAGES USING LOCAL CONNECTIVITY MAP

By

Omar Alaql

A thesis submitted to
Kent State University in partial fulfillment of
the requirements for the degree of
Master of Science in Computer Science

March 2014
Abstract

Vast amounts of valuable historical documents exist in libraries and in various National Archives that have not been exploited electronically. The analysis of historical documents presents specific difficulties with respect to other types of handwritten documents. Because of the low quality and the complexity of these documents, the document analysis remains an open research field. One of the major processes to analyze these documents is automatic text line extraction, which influences the accuracy of text recognition. The Center for Unified Biometrics and Sensors (CUBS) proposed one of the best-known approaches for text line extraction. In this research work, and starting with the concepts of CUBS approach, we propose an approach to extract text lines from the historical document images. The proposed approach is based on three local connectivity maps. One has the orientation angles of the text lines, and it is generated by using a dynamic steerable directional filter. This map is modified by using a mode filter to determine the paragraph map in the documents. Based on the values of the paragraph map, the adaptive local connectivity map (ALCM) is generated by using a static steerable directional filter to estimate the location of the text line. The proposed approach solves the problem of the ALCM binarization that the CUBS approach has, and gives the advantage of extracting the paragraphs in the document besides the text lines segmentation.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................................ III

LIST OF FIGURES ........................................................................................................................... VI

LIST OF TABLES ................................................................................................................................ IX

CHAPTER 1 INTRODUCTION ........................................................................................................... 10

CHAPTER 2 BACKGROUND ............................................................................................................. 12

2.1 An Overview on Optical Character Recognition (OCR) ......................................................... 12

2.1.1 OCR Steps .......................................................................................................................... 13

2.2 OCR for the Historical Documents .......................................................................................... 16

CHAPTER 3 STATE OF THE ART .................................................................................................. 21

3.1 Adaptive Local Connectivity Method ....................................................................................... 21

3.2 Hough Transform Method ...................................................................................................... 23

3.3 Level Set Method .................................................................................................................. 25

3.4 Partial Projection ................................................................................................................... 27

3.5 Affinity Propagation Method ................................................................................................ 29

CHAPTER 4 THE PROPOSED TEXT LINE EXTRACTION APPROACH FOR HISTORICAL DOCUMENTS ......................................................................................................................... 30

4.1 Steerable Dynamic Directional Filter ...................................................................................... 30

4.2 Mode Filter ............................................................................................................................ 34

4.3 Steerable Static Directional Filter .......................................................................................... 37

4.4 ALCM Thresholding .............................................................................................................. 39

4.5 Post-processing ..................................................................................................................... 41
4.5.1 Skew Correction ................................................................. 42
4.5.2 Horizontal Projection .......................................................... 42
4.6 Grouping the connected components ......................................... 45

CHAPTER 5 EXPERIMENTAL RESULTS AND ANALYSIS ...................... 46
5.1 Evaluation Methodology ............................................................ 46
5.2 Results and Analysis .................................................................. 48
  5.2.1 The Paragraphs Detection .................................................... 48
  5.2.2 Text Lines Segmentation ....................................................... 50
5.3 Tools ....................................................................................... 55

CHAPTER 6 CONCLUSIONS AND FUTURE WORK .............................. 57
REFERENCES .................................................................................. 58
LIST OF FIGURES

Fig 2-1: OCR System.................................................................................................................. 13
Fig 2-2: OCR Stages .................................................................................................................. 14
Fig 2-3: Preprocessing Stage ..................................................................................................... 15
Fig 2-4: Segmentation Stage ...................................................................................................... 15
Fig 2-5: The low quality of document ...................................................................................... 17
Fig 2-6: Complicated layouts .................................................................................................... 17
Fig 2-7: Skewness in the text lines ........................................................................................... 18
Fig 2-8: Small Components ....................................................................................................... 18
Fig 2-9 Noise in the document .................................................................................................. 19
Fig 3-1: Two different direction for the filter (From [1]) ......................................................... 22
Fig 3-2: The connected component of the text ........................................................................ 22
Fig 3-3: The CUBS method steps ............................................................................................ 23
Fig 3-4: The line representation ............................................................................................... 24
Fig 3-5: The final lines after the detection using the Hough Transform method (From [2]) .....
.................................................................................................................................................. 25
Fig 3-6: Level Set method steps............................................................................................... 26
Fig 3-7: The projection profile of the document and the red lines are the peaks ............ 27
Fig 3-8: The document columns, also shows the overlapping between text lines......... 28
Fig 3-9: Affinity Propagation method steps (From [3])......................................................... 29
Fig 4-1: The proposed approach stages .................................................................................... 31
Fig 4-2: The red line is the baseline of the text

Fig 4-3: The connectivity measure from six different directions. (e) has the maximum value

Fig 4-4: Input image before generating the paragraph map

Fig 4-5: The paragraph map of the image

Fig 4-6: The map before and after applying the mode filter

Fig 4-7: The kernel takes a direction that is different from the text line direction

Fig 4-8: The binarization of the ALCM

Fig 4-9: Thresholding results difference between the ALCM that was generated by our approach on the left and the ALCM of the CUBS approach on the right

Fig 4-10: Nine local thresholding results difference between the ALCM that was generated by our approach on the left and the ALCM of the CUBS approach on the right

Fig 4-11: Two patterns are falsely connected

Fig 4-12: The projection profile of the pattern

Fig 4-13: The problem of short line in the profile

Fig 4-14: The problem of narrow line in the profile

Fig 4-15: Smoothing the histogram

Fig 5-1: A sample image for each manuscript in the data set

Fig 5-2: Paragraph detection error is due to the neighboring paragraphs

Fig 5-3: Different orientation in the same paragraph

Fig 5-4: The detection rate in the paragraph and in the text lines
Fig 5-5: The error rate in the paragraph and in the text lines ........................................ 51
Fig 5-6: Spaces between text line are very narrow ...................................................... 52
Fig 5-7: Touching components in different text line ..................................................... 53
Fig 5-8: Paragraph overlaps ....................................................................................... 53
Fig 5-9: Low quality in some text lines ..................................................................... 54
Fig 5-10: The main actions of our tool ........................................................................ 56
LIST OF TABLES

Table 1 Experimental results for the paragraphs detection................................. 49
Table 2 Experimental results for text lines segmentation with post-processing......... 50
Table 3 Experimental results for text lines segmentation without post-processing..... 52
Table 4 Text Lines Detection Errors...................................................................... 55
CHAPTER 1

Introduction

Libraries and museums around the world contain an enormous amount of historical documents. Since they are a valuable resource for scientists and researchers, those documents need to be exploited electronically in order to make them accessible and searchable. The analysis and the recognition of the historical documents present many challenging research issues. Among these issues, text lines segmentation in the optical character recognition system is a critical process for translating the documents. The text lines segmentation of the historical documents is more complex due to the effect of aging and the lack of constraints on page layout.

The Center for Unified Biometrics and Sensors (CUBS) at the State University of New York at Buffalo proposed an approach for the text lines segmentation for the handwritten documents [1]. This approach, submitted by Z. Shi, S. Setlur and V. Govindaraju, uses the ALCM map, which is a gray scale image where each pixel value represents a connectivity property for its neighboring pixels in the original image, in order to extract patterns that represent regions of text line. This approach is robust and effective. It is considered to be one of the most effective techniques to extract text lines from documents. In two contests for handwritten segmentation [4, 5], the CUBS approach achieved the highest results in text line segmentation performance evaluation. However, this approach has difficulties and limitations when it comes to the binarization
of the ALCM images, especially when text lines in the document are very close to each other. That means, its performance on the historical document is lower since touching lines in these document is very common.

In this research work, we propose an approach for the text lines segmentation for the historical documents using steerable directional filters, and it is based on the concept of the adaptive local connectivity map (ALCM). The approach is proposed to enhance and increase the capability of the CUBS approach to deal with the historical documents and solve the problem of the binarization of the ALCM. Furthermore, the proposed approach gives the advantage of extracting the paragraphs in the document besides the text lines extraction. The proposed approach extracts each paragraph in the document first by determining the orientation of each paragraph. The orientation is determined by applying a mode filter on a mask that represents the direction angle value for each pixel. The angles’ values are assigned by using a steerable dynamic directional filter that is defined in this research work. Then, text lines in each extracted paragraph are segmented by using the text patterns technique that is presented in [1].

This research work is organized as follows. Chapter 2 briefly describes the optical character recognition system and its difficulties with the historical documents. In addition, some techniques for text lines segmentation are presented in chapter 3. Then, chapter 4 describes the proposed approach stages in more details. Finally, the results of the experiment and the analysis are presented in chapter 5 and the conclusion with the future work is drawn in chapter 6.
CHAPTER 2

Background

Some backgrounds will be presented in this chapter in order to put the reader in the right context. First, we give an idea about the optical character recognition system where the documents segmentation is part of the system, and then we present the historical documents in terms of analyzing them by the OCR system.

2.1 An Overview on Optical Character Recognition (OCR)

Optical character recognition, usually abbreviated to OCR, is the electronic translation of images into machine-editable text. These images can be handwritten or typewritten text. Thus, OCR in computer science refers to the branch that involves reading text from paper and translating the images into a form that the machine can manipulate.

One of the most challenging research areas in the field of image processing in recent years is the optical recognition systems due to the complexity of images that need to be analyzed. For instance, historical documents are the most complex and require more research compared to other types of documents. Several research works have been done to evolve newer techniques and methods that would provide higher accuracy rate and reduce the processing time.
2.1.1 OCR Steps

There are four major stages which must be done in order to apply any optical characters recognition:

1) Preprocessing.

2) Segmentation.

3) Feature extraction.

4) Training and recognition.

There are other stages that have been added in many research works such as in [6, 7], but in our perspective, these are the most general and most common stages in the OCR system.
Preprocessing:

Preprocessing aims to produce data that are easy for segmentation. There are several processes that are usually done in this stage, such as noise reduction, normalization of the data, and compression in the amount of data to be analyzed.
Segmentation:

Segmentation is the isolation of various writing units, such as paragraphs, sentences, words, or letters. This stage is the most important stage in the OCR system because it has a direct effect on the recognition rate. Most of the failures in the OCR system occur in the segmentation stage due to the huge variation in writing styles and structure.
• **Representation**

The feature extraction stage is a process that extracts the most relevant information from the text image which helps the recognition stage to recognize the text. Thus, this information consists of the features of each symbol that is needed to distinguish it from other symbols.

• **Recognition**

The recognition stage is the last and the main decision making stage. It is a classification process that identifies each unknown symbol and assigns it into a predefined class. This classification is based on the extracted features which are the output of the previous stage.

### 2.2 OCR for the Historical Documents

The large amount of historical manuscripts existing in libraries, museums, and private houses around the world are valuable human heritage. These historical documents have a special importance for our cultural heritage. Due to the expansion of digital libraries, the need for reliable and accurate systems for processing the historical documents has become more and more significant because these procedures help scholars easily access and analyze digital copies of historical documents. During the last decades a lot of research has been done in the field of optical character recognition systems, and a lot of OCR systems have been released as well. Most of these systems process machine printed documents successfully, but when it comes to handwritten documents the results have not been satisfactory. Moreover, when it comes to historical documents, such systems are
unable to process these documents [8]. Thus, historical documents processing is a challenging task for various reasons:

1. Lack of standard alphabets and presence of unknown fonts.
2. Due to the aging effect on historical documents, most of these documents have a low quality as shown in Fig 2-5.

Fig 2-5: The low quality of document

Fig 2-6: Complicated layouts
3. The lack of constraints on page layout. Fig 2-6 shows different layouts with different text orientations.

4. The complexity of handwriting. In some cases, it is hard for human beings to read them.

5. The variability of skew between the different text-lines and within the same text-line (Fig 2-7). Skewness is inherent in the historical documents.

Fig 2-7: Skewness in the text lines

Fig 2-8: Small Components
6. Spaces between lines are narrow and variable.

7. The existence of small components, such as dots and diacritics (e.g., Arabic script), between consecutive text-lines (Fig 2-8).

8. Many text lines may touch each other. Characters in one line touching text in adjacent lines.

9. Distinguishing noise from text. Such distortions may lead to several joint characters being interpreted as one single character (Fig 2-9).

10. It is very often also difficult to binarize the gray scale images while maintaining readability.

    Moreover, since we did our experiment of the proposed approach on Arabic documents, the Arabic language presents many challenges to the OCR system. These challenges can be summarized as following:

    a. Letter connectivity: In Arabic text, consecutive letters in word are connected to one another by a baseline. In this case each letter has several shapes according to its position on the word, beginning, middle, or at the end.
b. Position-dependent letter shaping: in addition to the letter body, most Arabic letters contain dots above or below its body, such as Thaa three dots above it. Some of the dots might be missed in the Pre-processing.

c. Different writing style: There are many different writing style, such as (Nuskah) for typewritten, (Ruqqah) for handwritten and a few others that are usually used such as (Koffy, Thuluth and Diwani). The handwritten style rules usually are not followed. That will cause more difficulties for recognition [9].
CHAPTER 3
State Of The Art

A large number of approaches have been proposed for text lines segmentation. We present in this section some of these approaches where each one has a different strategy. Also, we take into account the capability of using these approaches on the historical documents.

3.1 Adaptive Local Connectivity Method

Z. Shi, S. Setlur and V. Govindaraju [10] proposed an algorithm to extract the text lines of the complex historical documents using the adaptive local connectivity map (ALCM) technique. ALCM is a gray scale image where each pixel value represents a connectivity property for its neighboring pixels in the original image. It shows how likely each pixel belongs to the text line, which means that the pixels with higher values are pixels in the text region. Thresholding the ALCM discloses clear text-line patterns. A grouping algorithm is used to group the connected components into location masks for each text line. Text lines can be extracted by mapping the text connected components (Fig 3-2) onto the location masks. In [1], Z. Shi, S. Setlur and V. Govindaraju proposed a new filter which is cumulatively collects each pixel neighboring intensities from multiple directions, as shown in Fig 3-1, instead of using only the horizontal direction. The
maximum directional connectivity selected from these multiple directions captures the direction of the text line. Fig 3-3 shows the method steps.

In [11] a min-cut/max-flow graph cut algorithm is used to split up text-line patterns that are got from the ALCM. Vertically-aligned pixels are set to be more easily split than horizontally-aligned pixels by giving the low edge cost between vertical neighbors, and the high edge cost between horizontal neighbors.

This method deals with fluctuating or skewed text lines, but it fails in the step of grouping components when the spaces between words are highly varying. Also, it is difficult to achieve a good binarization of ALCM.

**Fig 3-1:** Two different direction for the filter (From [1])

**Fig 3-2:** The connected component of the text
3.2 Hough Transform Method

The Hough transform technique is a well known technique to detect lines and curves in digital images. Any black pixel has an infinite number of lines that could pass through this pixel. Each line can be expressed in the slope-intercept form as

\[ y = mx + c \]

Where, \( m \) is the slope of the line, and \( c \) is the intercept on \( y \) axis made by the line. Each line has a unique value \( (m, c) \) which is called its accumulator. There is a vote for the accumulator when the line passes through a black pixel. The local maximum accumulator is the best line that can lie on the text line. The problem of this representation of the lines is that \( m \) and \( c \) are infinite. To solve this problem, there is another representation used and formed as

\[ x \cos \theta + y \sin \theta = \rho \]
For each line, there is a perpendicular drop from the origin to the line as it shown in Fig 3-4. \( \theta \) is the angle that the perpendicular makes with the x-axis and \( \rho \) is the length of the perpendicular. Thus, each line has a unique accumulator \((\rho, \theta)\).

In Likforman-Sulem et al. [12], connected components are extracted first from the input image, and the centroids of the connected components are the units for the Hough transform. The corresponding cell \((\rho, \theta)\) of the Hough domain contains a set of aligned units in the image along a line with parameters \((\rho, \theta)\). Two hypotheses are used to solve the problem which is units within a text line are not perfectly aligned. Units of the cell structure of a primary cell form an alignment. The cell structure of a cell \((\rho, \theta)\) is a cluster of cells where the center is \((\rho, \theta)\). The primary cell is the cell that includes the highest number of units in its cell structure. Then, contextual information is used to validate the potential alignments. The potential alignments that contain more near neighbors units than far neighbors are validated, and the other alignments are ignored.
This method is able to detect the text lines with different directions, but it fails with the highly fluctuating lines.

3.3 Level Set Method

An approach for handwritten text line segmentation using level sets has been presented in [13]. In this approach, probability density function PDF of pixel values is estimated first. Each pixel represents the probability that it belongs to a text line. For example, a pixel at the center of a text line is more likely to be black than at a text line gap. On the probability map, peaks represent text lines, while valleys are the boundary between neighboring text lines. After estimating the probability map, the level set method is used to analyze the probability map. The level set method determines the boundary of neighboring text lines by evolving an initial estimate. In the level set method, negative values are inside the text line, positive values are outside, and a zero value for the
boundary. The zero level set automatically grows, merges, and stops to the final text line boundary by tracking them. Fig 3-6 shows the steps of the method.

Fig 3-6: Level Set method steps
This method is script independent. It performs well on many scripts such as Arabic, Chinese, Korean, and Hindi. Also, this method is robust to noise. Nevertheless, it suffers from the high computational cost, and the adjacent text lines may merge if they significantly touch each other. In addition, this approach has made some modification in the growing and merging criteria of level set, keeping straight horizontal assumption of text lines. Thus, vertical text lines that may exist in same document will not be extracted.

3.4 Partial Projection

In any projection based approach the projection profile is obtained by summing pixel values along the horizontal axis. Then, the text lines can be determined by finding the gaps in the profile.

Fig 3-7: The projection profile of the document and the red lines are the peaks

For skewed or moderately fluctuating text lines, the image may be divided into vertical strips such as the approach that is presented in [14]. In this approach a partial contour following based method is used to detect the separating lines. This method is an
enhancement of another approach that is called partial projection [15] where the overlapping of text lines is not extracted correctly. In the partial projection method the page of the document is subdivided into columns. Horizontal projection is applied on the columns to determine the minimal values of the projection histograms. This minimal value indicates the potential separating lines. A horizontal stroke is drawn on each minimal value inside a column. Two adjacent lines are separated by linking these strokes. The partial contour following method solves the case where the minimal values do not exist due to the overlapping between the text lines as shown in Fig 3-8.

Fig 3-8: The document columns, also shows the overlapping between text lines

This method works correctly for the texts with tabulations, some noise, incomplete lines and skew in text orientation. However, the method cannot be used for the handwritten document with different text lines orientations such as having vertical and horizontal text lines in the same document. Also, it does not deal with touching lines.
3.5 Affinity Propagation Method

Kumar et al. [3] proposed an algorithm to extract text lines from unconstrained handwritten Arabic documents. The algorithm first estimates local orientation at each primary component of a word to build a sparse similarity graph. The centroids of the components are defined as local coordinates system. At each coordinate, the region is divided into five regions to find local orientation. The Breadth-First Search algorithm is applied to find disjoint sets in the similarity graph. There exists a path from each node to every other node in each set. Each disjoint set estimates a text line. The clustering method that is called Affinity propagation (AP) is used as well to refine these text lines by finding the shortest path between each node in the local orientation graph and every other node.

This algorithm is fast and performs well unless the words from different lines touch each other.

Fig 3-9: Affinity Propagation method steps (From [3])
CHAPTER 4

The Proposed Text Line Extraction Approach for Historical Documents

The proposed approach consists of several stages. First, apply a steerable dynamic directional filter to convert the image to a Local Connectivity Directions Map (LCDM). Second, apply a mode filter to extract each paragraph in the document and its orientation. Then, convert the original document image into an adaptive local connectivity map ALCM by applying a steerable static directional filter based on the local connectivity directions map. After that, the ALCM is converted from gray scale image to a binary image where each component indicates the region of a text line which is called pattern. Then, validate these patterns by using the projection profile technique. Finally, the connected components in the original image are grouped based on these patterns locations. In the following sections is a description for each stage of the proposed approach.

4.1 Steerable Dynamic Directional Filter

Steerable dynamic directional filter is a new proposed filter that converts the image to a local connectivity directions map (LCDM) that contains angles’ values. Each value at each pixel in the map has an angle value which is the most likely local direction of the text line or the space between the text lines. It is captured by the maximum or the minimum, respectively, directional connectivity selected from the multiple directions.
The connectivity measure is defined by cumulatively collecting, at each pixel, its neighboring pixels’ intensities along these multiple directions. The ALCM calculation is described in section 3.1. Since the objective of this filter is determining the orientation...

**Fig 4-1**: The proposed approach stages
of the whole paragraph, two connectivity measures are calculated, the maximum and the minimum directional connectivity that are selected from the multiple directions. For convenience, the input image is reversed so that 255 represent the strongest level of intensity for the foreground text. The minimum directional connectivity value is taken if the pixel intensity of the input image is 0, and the maximum directional connectivity value otherwise. Thus, the directions of both, the text lines and the spaces between them are extracted. The purpose of that is making the next stage of this approach, which is the mode filter, easier to apply because the mode filter is extracting the mode direction of the whole paragraph.

The kernel of the filter can take two shapes, rectangle or ellipse. Ellipse is used with languages that have a baseline connecting the letters in each word as shown in Fig 4-2, such as Arabic and Farsi and Urdu. Ellipse shape gives more focus on the baseline. Rectangle shape is used with other languages. The height of the kernel is chosen to be a value less than the height of the text, and width should be at least twice the height.

For the implantation, six different angles (Fig 4-3) are used here to estimate the orientation 0, 30, 60, 90, 120, and 150. The number of the angles that are tested has a significant effect on the complexity of the algorithm since the mode filter in the next stage is counting how often each value is repeated in each neighboring pixel. Also, if we take too many angles, the efficiency of the mode filter will be affected negatively.

Fig 4-2: The red line is the baseline of the text
The algorithm of generating local connectivity directions map LCDM by using the steerable dynamic directional filter is as follows:

For each pixel with coordination x,y
   For each direction Q
      Find d = the value of ALCM with rotation angle Q and coordination x,y
      If d>maximum
         Max=Q
      If d<minimum
         Minimum=Q
      If f(x,y) is white pixel
         LCDM(x,y)=minimum
      Else
         LCDM(x,y)=maximum

Fig 4-3: The connectivity measure from six different directions. (e) has the maximum value
4.2 Mode Filter

Mode filter is a filter that changes the pixel value to the most repeated value in the neighboring pixels. The filter’s objective is to estimate the boundary of each paragraph and its orientation. This filter is applied on the LCDM that is generated from the previous stage. The filter has several iterations. The process is repeated until values become stable. In our case, pixels values that appear most often in the LCDM expand until they reach the paragraph boundary. At the end of this stage, each region in the paragraph map indicates one separate paragraph. The document that in Fig 4-4 has six different paragraphs, three of them are horizontal, and the other ones have different orientations. After generating the paragraph map by using the mode filter, as it is shown in Fig 4-5, the paragraph reigns are extracted with direction value for each one. Also, Fig 4-6 shows one vertical line and its LCDM pixels values before and after applying the filter. 90 degree appears most often in the map expanded until it reached the boundary of this text line. Furthermore, 90 degree indicates that this text line is a vertical line.
Fig 4-4: Input image before generating the paragraph map

Fig 4-5: The paragraph map of the image
Fig 4-6: The map before and after applying the mode filter.
4.3 Steerable Static Directional Filter

The steerable static directional filter works the same as the steerable directional filter that is presented in [1], but instead of rotating the kernel with multiple directions for each pixel, the rotation angle is taken from the paragraph map that is generated from the previous stage. The filter converts the original image into an adaptive local connectivity map (ALCM) which is a convolution that aggregates the pixel intensities within all pixels in the kernel. The definition of this transform can be defined as the CUBS definition by the following convolution:

$$A(x, y) = \int_{\mathbb{R}^2} f(x, y) G_{a,b}^{PM(x,y)}(x - t, y - s) dt ds$$

Where

$$G_{a,b}^{PM(x,y)}(x, y) = \begin{cases} 
1 & \text{if } (x, y) \in E_{a,b}^{PM(x,y)} \\
0 & \text{otherwise} 
\end{cases}$$

$$E_{a,b}^{PM(x,y)} = \left\{(x, y) \mid \left\{ \begin{array}{l}
x < a \cos(\theta - PM(x,y)) \\
y < b \cos(\theta - PM(x,y)) \\
\end{array} \right. \text{ and } 0 \leq \theta < 2\pi \right\}$$

Where E is an ellipse and the semi-minor is a, b. PM(x,y) is the value of (x,y) in the paragraph map, which is the direction angle.
The ALCM values are re-scaled after being generated to a gray scale image with values ranging from 0 to 255. This filter is designed for solving the particular thresholding problem seen in the steerable filter that is presented in [1]. By using their filter, it is difficult to binarize the ALCM when the text lines are close to or touching each other each because the pixels between the text lines will have high density values that can be greater than the values in the text line region. What makes this happen is that the kernel takes more often angles between 30 and 70, or between 120 and 150 to capture the maximum value as shown in Fig 4-7. In our proposed filter, this problem is solved by making the angle rotation for the kernel uniform in each paragraph in the document. Which means that rotation angle of the kernel is the same in text line regions or in the spaces between them. Therefore, the range between the density values of the ALCM in these two regions becomes larger, so that makes the binarization easier.

The algorithm of generating adaptive local connectivity map (ALCM) by using the steerable static directional filter is as follows:

For each pixel with coordination \(x, y\)

- Take rotation angle \(Q = \text{paragraph\_map}(x, y)\)
- Rotate the kernel by \(Q\)
- \(\text{ALCM}(x, y) = \text{aggregates the pixel intensities within the Kernel centered at } (x; y)\)
4.4 ALCM Thresholding

In order to reveal the text line patterns, the ALCM is converted from gray scale image to a binary image where each component indicates the region of a text line. As we mentioned in section 4.3, the binarization of the ALCM becomes much easier by using our approach. Many binarization algorithms, that were not applicable to be used in the CUBS approach, can be used now with our approach. Fig 4-9 shows the results of the ALCM thresholding that is extracted from our approach and the CUBS approach by using 15 different thresholding methods, (a) Huang [16], (b) Intermodes [17], (c) IsoData [18], (d) Li [19], (e) MaxEntropy [20], (f) Mean [21], (j) MinError [22], (h) Minimum [17], (i) Moments [23], (k) Otsu [24], (l) Percentile [25], (m) RenyiEntropy [20], (n) Shanbhag [26], (j) Triangle [27], (k) Yen [28]. Furthermore, fig 4-10 shows the results of using nine local thresholding methods, (a) Bernsen [29], (b) Contrast [30], (c)
Mean, (d) Median, (e) MidGray, (f) Niblack [31], (g) Otsu [24] (h) Phansalkar [32], (i) Sauvola [33].

**Fig 4-8:** The binarization of the ALCM

**Fig 4-9:** Thresholding results difference between the ALCM that was generated by our approach on the left and the ALCM of the CUBS approach on the right
4.5 Post-processing

The initial text patterns need to be validated before using them to group the original image connected components. The projection profile strategy is used to make the validation. The projection profile will segment the patterns that are badly connected. Fig 4-11 shows an example of two patterns that should be segmented. The first step that has to be done before making the projection to get the histogram is that each pattern must be
extracted in order to apply the projection on each one separately. Each connected component in the ALCM represents one pattern.

4.5.1 Skew Correction

Before making the projection, each pattern is rotated to correct the skew and get the projection profile. By using the paragraph map values, which are the angles of the direction of each paragraph, the correction is done by making a rotation for each pattern based on the values of the angles.

4.5.2 Horizontal Projection

The projection profile for each pattern is extracted in order to segments the text patterns. Each minimum of the profile curve is a potential segmentation point. Fig 4-12 shows one of the paragraph and its profile.
There are some drawbacks that make finding the maximum and the minimum in the profile more complicated. A short line will provide a low peak that might be ignored as shown in Fig 4-13.
Furthermore, very narrow patterns, or patterns that include overlapping will not produce significant peaks. Fig 4-14 shows an example of these patterns.

Fig 4-14: The problem of narrow line in the profile

In this research work, an algorithm is proposed to smooth the projection profile as shown in Fig 4-15. After smoothing the profile, each minimum is a segmentation point. This algorithm solves the problem of narrow and short patterns. The algorithm consists of the following steps:

1. Calculate the average of peaks and bottoms.
2. Set a new parameter with a zero initial value.
3. Pass all the points in profile. If the difference between the current point and the parameter value is greater than the average, add the current point to the new profile and set the parameter equal the current point value.
The final stage in our method is grouping all connected components of each line into one group which indicates the final segmentation of the text line. We used the same technique that is used in the CUBS method which groups the connected components based on the text pattern that is generated from previous stage. The grouping strategy contains the following steps:

1. Extract the central points of all connected components in the original image.
2. If the central point is inside a text pattern, the connected component of this point is grouped to be a part of this text line.
3. If the central point is outside the text patterns’ location, the connected component is grouped to a pattern that has a border nearest to this point.
CHAPTER 5
Experimental Results and Analysis

We have tested our method on nine Arabic manuscripts consisting of 10848 text lines. Each manuscript has a different script and different level of complexity. The images were taken from the department of manuscripts in Imam Muhammad Ibn Saud Islamic University library \[34\]. The manuscripts numbers in the library archive are 2522, 1758, 1755, 4634, 5754, 4780, 4064, 2512, and 1579. Fig 5-1 shows a sample image for each manuscript. We present in this section the evaluation of the method, then the analysis of the results, and finally the tools used in our experiment.

5.1 Evaluation Methodology

Several evaluation techniques have been used for text line segmentation. Some of these are applicable only to printed documents and cannot be used for the handwritten documents because the performance evaluation is based on the bounding box \[35\]. Since our method is proposed mainly for the historical documents, we use the evaluation technique that is appropriate to measure the performance of the handwritten documents. It is similar to \[35\] \[13, 36\], MatchScore matrix is calculated by counting the number of matches between the pixels in the ground-truth object, whether paragraph or text line, and the detected object.
Fig 5-1: A sample image for each manuscript in the data set
Where $G$ is a set of ground-truth pixels, and $R$ is a set of detected objects. If the match score is greater than 0.95, the line is marked as correctly detected object. The number of correctly detected objects (CDO) over the ground-truth objects (GTO) gives the detection rate (DR) which is defined as:

$$DR = \frac{CDO}{GTO}$$

In addition, the number of objects that are not detected over the detected objects (DO) gives the error rate (ER) which is defined as:

$$ER = 1 - \frac{CDO}{DO}$$

5.2 Results and Analysis

5.2.1 The Paragraphs Detection

Among 553 ground-truth paragraphs, 543 paragraphs are detected. 518 are correctly detected, and 25 paragraphs are falsely detected as shown in Table 1. This detection includes the boundary and the orientation for each paragraph. Therefore, the detection rate is 93.67% and the error rate is 4.6%.
The false detection in our method has three causes. 74% of the paragraphs that are not correctly detected have overlap with other paragraphs that have the same orientation, or the space between these paragraphs and the other paragraphs is narrow as shown in Fig 5-2.

Table 1 Experimental results for the paragraphs detection

<table>
<thead>
<tr>
<th></th>
<th>GTO</th>
<th>DO</th>
<th>CDO</th>
<th>DR</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript_2522</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Manuscript_1758</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Manuscript_1755</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Manuscript_4634</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Manuscript_5754</td>
<td>74</td>
<td>74</td>
<td>73</td>
<td>98.64%</td>
<td>1.35%</td>
</tr>
<tr>
<td>Manuscript_4780</td>
<td>56</td>
<td>56</td>
<td>53</td>
<td>94.64%</td>
<td>5.35%</td>
</tr>
<tr>
<td>Manuscript_4064</td>
<td>88</td>
<td>88</td>
<td>81</td>
<td>92.04%</td>
<td>7.95%</td>
</tr>
<tr>
<td>Manuscript_2512</td>
<td>75</td>
<td>71</td>
<td>69</td>
<td>92%</td>
<td>2.81%</td>
</tr>
<tr>
<td>Manuscript_1579</td>
<td>53</td>
<td>47</td>
<td>35</td>
<td>66.60%</td>
<td>25.53%</td>
</tr>
<tr>
<td>Total Paragraphs</td>
<td>553</td>
<td>543</td>
<td>518</td>
<td>93.67%</td>
<td>4.6%</td>
</tr>
</tbody>
</table>

In addition, 14% of these paragraphs have wrong orientation because of having more than one orientation in the same paragraph (Fig 5-3). Moreover, 11% of the paragraphs are not detected correctly due to the low quality.

Fig 5-2: Paragraph detection error is due to the neighboring paragraphs.
5.2.2 Text Lines Segmentation

Among 10883 ground-truth text lines, 10597 text lines are detected. 10322 are correctly detected and 275 text lines are falsely detected as shown in Table 2. Therefore, the detection rate is 94.84% and the error rate is 2.59%. This detection rate is after the post-processing stage that is presented in section 4.3. Obviously, the paragraph detection stage has a significant effect on the text lines segmentation results. Fig 5-4 shows the detection rate for the paragraphs and the text lines. In addition, Fig 5-5 shows the error rate as well.

**Table 2 Experimental results for text lines segmentation with post-processing**

<table>
<thead>
<tr>
<th>Manuscript_2522</th>
<th>GTO</th>
<th>DO</th>
<th>CDO</th>
<th>DR</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1094</td>
<td>1093</td>
<td>1092</td>
<td></td>
<td>99.81%</td>
<td>0.09%</td>
</tr>
<tr>
<td>Manuscript_1758</td>
<td>1171</td>
<td>1170</td>
<td>1157</td>
<td>98.80%</td>
<td>1.11%</td>
</tr>
<tr>
<td>Manuscript_1755</td>
<td>1349</td>
<td>1349</td>
<td>1333</td>
<td>98.81%</td>
<td>1.18%</td>
</tr>
<tr>
<td>Manuscript_4634</td>
<td>1374</td>
<td>1360</td>
<td>1329</td>
<td>96.62%</td>
<td>2.27%</td>
</tr>
<tr>
<td>Manuscript_5754</td>
<td>1659</td>
<td>1567</td>
<td>1503</td>
<td>90.59%</td>
<td>4.08%</td>
</tr>
<tr>
<td>Manuscript_4780</td>
<td>700</td>
<td>698</td>
<td>657</td>
<td>93.85%</td>
<td>5.87%</td>
</tr>
<tr>
<td>Manuscript_4064</td>
<td>1631</td>
<td>1521</td>
<td>1498</td>
<td>91.84%</td>
<td>1.51%</td>
</tr>
<tr>
<td>Manuscript_2512</td>
<td>1155</td>
<td>1135</td>
<td>1097</td>
<td>94.97%</td>
<td>3.34%</td>
</tr>
<tr>
<td>Manuscript_1579</td>
<td>750</td>
<td>704</td>
<td>656</td>
<td>87.46%</td>
<td>6.81%</td>
</tr>
<tr>
<td>Total Text Lines</td>
<td>10883</td>
<td>10597</td>
<td>10322</td>
<td>94.84%</td>
<td>2.59%</td>
</tr>
</tbody>
</table>
Fig 5-4: The detection rate in the paragraph and in the text lines

Fig 5-5: The error rate in the paragraph and in the text lines
Table 3 shows the results without applying the post-processing stage. These results show the importance of this stage especially for Manuscript_4634 and Manuscript_2512 because the spaces between the text lines in these two manuscripts are very narrow as shown in Fig 5-6. Thus, most of the text patterns in these manuscripts need to be segmented.

Table 3 Experimental results for text lines segmentation without post-processing

<table>
<thead>
<tr>
<th>Manuscript</th>
<th>GTO</th>
<th>DO</th>
<th>CDO</th>
<th>DR</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript_2522</td>
<td>1094</td>
<td>1093</td>
<td>1092</td>
<td>99.81%</td>
<td>0.09%</td>
</tr>
<tr>
<td>Manuscript_1758</td>
<td>1171</td>
<td>1170</td>
<td>1066</td>
<td>91.03%</td>
<td>8.88%</td>
</tr>
<tr>
<td>Manuscript_1755</td>
<td>1349</td>
<td>1349</td>
<td>1303</td>
<td>96.59%</td>
<td>3.40%</td>
</tr>
<tr>
<td>Manuscript_4634</td>
<td>1374</td>
<td>1360</td>
<td>324</td>
<td>23.58%</td>
<td>76.17%</td>
</tr>
<tr>
<td>Manuscript_5754</td>
<td>1659</td>
<td>1567</td>
<td>1339</td>
<td>80.71%</td>
<td>14.55%</td>
</tr>
<tr>
<td>Manuscript_4780</td>
<td>700</td>
<td>698</td>
<td>592</td>
<td>84.57%</td>
<td>15.18%</td>
</tr>
<tr>
<td>Manuscript_4064</td>
<td>1631</td>
<td>1521</td>
<td>1452</td>
<td>89.02%</td>
<td>4.53%</td>
</tr>
<tr>
<td>Manuscript_2512</td>
<td>1155</td>
<td>1135</td>
<td>744</td>
<td>64.41%</td>
<td>34.44%</td>
</tr>
<tr>
<td>Manuscript_1579</td>
<td>750</td>
<td>704</td>
<td>622</td>
<td>82.93%</td>
<td>11.64%</td>
</tr>
<tr>
<td>Total Text Lines</td>
<td>10883</td>
<td>10597</td>
<td>8534</td>
<td>78.41%</td>
<td>19.46%</td>
</tr>
</tbody>
</table>

Fig 5-6: Spaces between text line are very narrow.
There are four causes of the false detection for the text lines segmentation in this experiment:

1. Some connected components are grouped to the neighbor text line because the text lines are touching each other. This touching merges two letters or more in one connected component, and these letters are located in different text lines. 36% of the errors are caused by this.

![Fig 5-7: Touching components in different text line](image)

2. There are some overlaps between text lines in two different paragraphs. Two or more text lines are falsely merged in one text line if they are in different paragraphs and these paragraphs are close to each other and have the same orientation. Fig 5-8 shows the error of detection in the last two text lines and how they merged into one text line. 44% of the errors caused by this reason.

![Fig 5-8: Paragraph overlaps](image)
3. Some text lines are located in a paragraph that has wrongly detected orientation. As a result of the false detection, most of the text lines inside this paragraph will be falsely detected. This causes 11% of the errors.

4. The last reason is the low quality of some text lines as shown in Fig 5-9. 9% only of the errors are due to the low quality.

![Low quality in some text lines](image)

**Fig 5-9:** Low quality in some text lines

Table 4 shows the errors that are caused by the four reasons for each manuscript. As we can see that most of the manuscripts have errors with touching lines and the paragraphs overlap, whether the complexity of the manuscript is very high and low. These two reasons are the common errors that need to be fixed in our approach. The third reason which is the wrong paragraph orientation occurs mostly in the last two manuscripts, because many paragraphs in these manuscripts have more than one direction in the same paragraph. For the low quality reason, two manuscripts in the data set have low quality images. This is related to the quality of scanning. Therefore, the last reason is not related to our approach performance.
### Table 4 Text Lines Detection Errors

<table>
<thead>
<tr>
<th>Manuscript_2522</th>
<th>Touching Lines</th>
<th>Paragraphs Overlaps</th>
<th>Wrong Paragraph Orientation</th>
<th>Low Quality Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript_1758</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Manuscript_1755</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Manuscript_4634</td>
<td>46</td>
<td>26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Manuscript_5754</td>
<td>54</td>
<td>102</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Manuscript_4780</td>
<td>34</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Manuscript_4064</td>
<td>2</td>
<td>95</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Manuscript_2512</td>
<td>15</td>
<td>13</td>
<td>14</td>
<td>40</td>
</tr>
<tr>
<td>Manuscript_1579</td>
<td>20</td>
<td>13</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>200</td>
<td>249</td>
<td>61</td>
<td>51</td>
</tr>
<tr>
<td>Percentage</td>
<td>35.65%</td>
<td>44.38%</td>
<td>10.87%</td>
<td>9.09%</td>
</tr>
</tbody>
</table>

#### 5.3 Tools

A tool was built for the proposed method by using the library of OpenCV [37] in C language with a graphical user interface by QT library [38] in C++ language. Many actions can be done by this tool as shown in (Fig 5-10).
Fig 5-10: The main actions of our tool
CHAPTER 6

Conclusions and Future Work

Extraction of text lines from the historical document images is one of the important associated problems of the OCR systems. Presence of skewed text lines, touching lines, and the lack of constraints on page layout, always make it difficult for accurate extraction of the text lines from the historical documents. We presented a novel method, which is based on the concept of the adaptive local connectivity map (ALCM), for segmenting text lines in the historical document images. Our experiments demonstrate the efficacy of this method. Initial results of this work yielded an accuracy of 94.84% for the text line extraction, and 93.67% for the paragraph extraction. From early analysis, we find that many of our errors are caused by the touching components and the neighboring paragraphs that have the same orientation. Our method is to be further improved by refining the extracted paragraphs and segmenting the touching components. We will continue our tests and comparisons with other methods, such as the CUBS method [1] and the level set method [13].
References


58


*QT Project*. Available: http://qt-project.org/