

# Document Image Enhancement

Su Bolan



SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2012 August

# Document Image Enhancement

**Su Bolan**

A Thesis Submitted For the Degree of

*Doctor of Philosophy*

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2012 August

I would like to dedicate this thesis to  
my beloved parents and Zhang Xi  
for their endless support and encouragement.

*It is the time you have wasted for your rose that makes your rose so  
important.*

*Antoine de Saint Exupery "Little Prince"*

## Acknowledgements

First of all, I express my most sincere appreciation to my PhD supervisors Professor Tan Chew Lim in School of Computing, National University of Singapore and Dr. Lu Shijian. They are very kind and provide me a research environment which is full of freedom. Their wide knowledge and constructive advice have inspired me with various ideas to tackle the difficulties and attempt new directions. In particular, their understanding and help in every aspect have supported me through the chaos and confusion in those difficult days. This thesis would not have been possible without their generous contributions.

I thank all of my lab fellows for all of great ideas, hard work, discussions and arguments during my research study in the Center of Information Mining and Extraction (CHIME) of School of Computing, National University of Singapore. They are Dr. Sunjun, Dr. Li Shimiao, Dr. Gong Tianxia, Dr. Wang jie, Dr. Liu Ruizhe, Dr. P Shivakumara, Mohtarami Mitra, Chen Qi, Situ Liangji, Trung Quy Phan, Chen Bin, Huang Yun, Zhang Wei, who helped me in academic or non-academic aspects.

I wish to extend my warmest thanks to all friends that came across my life during my four years study in Singapore. I wouldn't have some many memorable moments in my life without you. I wouldn't able to ride out the difficulties without your helps. I am sorry I can only list some of them here: Wang Guangsen, Li Xiaohui, Fang Shunkai, Zheng Hanxiong, Zhou Zenan, Zheng Manchun, Wang Chundong, Chen Wei, Deng Chengzi, Cheng Yuyao... Life is a journey, not a destination. It is you make my journey in Singapore so colorful.

Last but not least, I wish to express my special gratitude to my parents, who always love me unconditionally, and my beloved Zhang Xi, who gives me a lot of delighted hours and always companies me in my bright and dark time.

## Abstract

Document image enhancing aims to improve the document image quality, which not only enhance human perception, but also facilitate the subsequent automated image processing. Document image enhancing is a difficult problem, because : 1) The information it aims to recover could be lost in many cases; 2) Different ways of image distortion could lead to the same degraded document image. This thesis focuses on three aspects of the document enhancement techniques including document image binarization, web image recognition and document image deblurring. we have proposed several document enhancement techniques that have been tested on some public datasets and shown superior performance.

First, we developed a set of binarization techniques that aim to improve the binarization performance. In addition, we also proposed frameworks to improve the existing document image binarization techniques. Second, We proposed a robust text recognition technique for web images. Third, we proposed an image blur detection and classification technique that makes use of singular value feature and alpha channel feature. We also developed a motion deblurring technique for document images.

# Contents

<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction of Document Image Enhancement</b>	<b>1</b>
1.1 Background and Motivation . . . . .	1
1.2 Scope of Study . . . . .	3
1.3 Organization of this thesis . . . . .	4
<b>2 Literature Review of Document Image Binarization</b>	<b>6</b>
2.1 Previous Work . . . . .	7
2.2 Challenges on Degraded Document Image Binarization . . . . .	10
<b>3 Document Image Binarization using Local Maximum and Minimum</b>	<b>13</b>
3.1 Contrast Image Construction . . . . .	14
3.2 High Contrast Pixel Detection . . . . .	18
3.3 Historical Document Thresholding . . . . .	19

---

<b>4</b>	<b>Document Image Binarization using Background Estimation</b>	<b>22</b>
4.1	Document Background Estimation . . . . .	23
4.2	Stroke Edge Detection . . . . .	25
4.3	Threshold Estimation and Post-Processing . . . . .	27
<b>5</b>	<b>A Robust Adaptive Document Image Binarization Technique for Degraded Document Images</b>	<b>28</b>
5.1	Contrast Image Construction . . . . .	30
5.2	Text Stroke Edge Pixel Detection . . . . .	33
5.3	Local Threshold Estimation . . . . .	35
5.4	Post-Processing . . . . .	36
<b>6</b>	<b>Experiments and Discussions of the Proposed Binarization Methods</b>	<b>38</b>
6.1	Evaluation Metrics . . . . .	39
6.2	Experiments on competition datasets . . . . .	42
6.3	Testing on Bickley diary dataset . . . . .	45
<b>7</b>	<b>Learning Frameworks For Document Image Binarization</b>	<b>53</b>
7.1	A Learning Framework using K-means Algorithm . . . . .	54
7.1.1	Uncertain Pixel Detection . . . . .	54
7.1.2	Uncertain Pixel Classification . . . . .	57
7.1.3	Experiments . . . . .	58
7.2	Combination of Document Image Binarization Techniques . . . . .	59
7.2.1	Feature Extraction . . . . .	61
7.2.2	Combination of Binarization Results . . . . .	62

7.2.3	Experiments . . . . .	64
7.3	A Learning Framework using Markov Random Field . . . . .	65
7.3.1	Uncertain Pixels Detection . . . . .	66
7.3.2	Edge Pixels Detection . . . . .	66
7.3.3	Uncertain Pixels Classification . . . . .	67
7.3.4	Experiments . . . . .	69
<b>8</b>	<b>Enhancement of Web Images for Text Recognition</b>	<b>71</b>
8.1	Introduction . . . . .	71
8.2	Literature Review . . . . .	72
8.3	Text Recognition on Web Images . . . . .	73
8.3.1	Pre-Processing . . . . .	74
8.3.2	Image Smoothing and Binarization . . . . .	75
8.3.3	Detection of Character Components . . . . .	80
8.3.4	Skew Correction and Text Recognition . . . . .	83
8.4	Experiments . . . . .	83
<b>9</b>	<b>Document Image Deblurring</b>	<b>88</b>
9.1	Mathematical Model of Image Blur . . . . .	88
9.2	Image Deblurring as an Ill-posed Problem . . . . .	91
9.3	Related Work . . . . .	92
9.4	Blurred Image Region Detection and Classification . . . . .	95
9.4.1	Image Blur Features . . . . .	96
9.4.2	Experiments and Applications . . . . .	101
9.5	Restoration of Motion Blurred Document Images . . . . .	105
9.5.1	Alpha Channel Map . . . . .	106



9.5.2	Restoration of Motion blur image . . . . .	109
9.5.3	Experiments . . . . .	111
<b>10</b>	<b>Conclusions and Future Work</b>	<b>114</b>
10.1	Conclusions . . . . .	114
10.2	Contributions of my thesis work . . . . .	115
10.3	Future Research Direction . . . . .	117
<b>11</b>	<b>Publications arising from this work</b>	<b>119</b>
	<b>References</b>	<b>122</b>

# List of Figures

2.1	Two degraded document image examples, which are obtained from Document Image Binarization Contest (DIBCO) [1] dataset . . .	6
2.2	Binarization Results using Otsu’s method of images in Figure 2.1	7
2.3	Binarization Results using Niblack’s method of images in Figure 2.1	8
2.4	Binarization Results using Sauvola’s method of images in Figure 2.1	9
3.1	The flowchart of Binarization using local maximum and minimum	14
3.2	The gradient and contrast map: (a) The traditional image gradient that is obtained using Canny’s edge detector [2]; (b) The image contrast that is obtained by using the local maximum and minimum [3];(c) One column of the image gradient in Figure 3.2(a) (shown as a vertical white line);(d) The same column of the contrast image in Figure 3.2(b). . . . .	16
3.3	High contrast pixel detection: (a) Global thresholding of the gradient image in Figure 3.2(a) by using Otsu’s method; (b) Global thresholding of the contrast image in Figure 3.2(b) by using Otsu’s method. . . . .	18
4.1	The flowchart of Binarization using background estimation . . . .	23

5.1	The flowchart of the proposed adaptive document image binarization technique . . . . .	29
5.2	Contrast Images constructed using the local image gradient [4] in (a), the local image contrast [5] in (b), and our proposed method in (c), respectively. . . . .	31
5.3	Binary contrast images, canny edge maps and their corresponding combined Edge maps, respectively. . . . .	34
6.1	Five degraded document image examples taken from DIBCO dataset series and Bickley diary dataset. . . . .	46
6.2	Binarization Results of the sample document image in Figure 6.1(a) produced by different methods. . . . .	47
6.3	Binarization Results of the sample document image in Figure 6.1(b) produced by different methods. . . . .	48
6.4	Binarization Results of the sample document image in Figure 6.1(c) produced by different methods. . . . .	49
6.5	Binarization Results of the sample document image in Figure 6.1(d) produced by different methods. . . . .	50
6.6	Binarization Results of the sample document image in DIBCO 2011 dataset produced by different methods. . . . .	51
6.7	Binary results of the badly degraded document image from Bickley diary dataset shown in Figure 6.1(e) produced by different binarization methods and the ground truth image. . . . .	52

7.1	Binarization results of the document image in Figure 7.1(a). The left images in Figure 7.1(b-d) are produced by testing methods, the right images are produced by proposed framework. . . . .	58
7.2	F-measure values of ten different document images in DIBCO 2009 dataset. . . . .	59
7.3	The overall flowchart of our proposed document binarization combination framework. . . . .	60
7.4	The flowchart of combination of two binarization results. . . . .	63
7.5	Two degraded document image examples and corresponding binarization results produced by Otsu’s method, Sauvola’s method and our proposed combination framework, respectively. . . . .	65
7.6	Binarization results with/without our MRF framework . . . . .	70
8.1	Some low quality web image examples . . . . .	72
8.2	A column of image pixels taken from Figure 8.1(f) which is shown in blue. The vertical index denotes the pixel intensity, the horizontal index denotes the image pixel index. The smoothed line is represented in red. . . . .	76
8.3	Smoothed images of the original images in Figure 8.1 . . . . .	79
8.4	Binary images of the original images in Figure 8.1 . . . . .	81
8.5	An example of skew correction. (a) shows the original web image, the binary image with a red line denotes the text orientation calculated using PCA is shown in (b). (c) shows the rotated result.	84
8.6	Some web image examples that cannot be recognized . . . . .	87
9.1	The model of Image Blurring, which is adopted from [6] . . . . .	91

- 9.2 Illustration of the blur map constructed by a singular value feature:  
(a,c) show a pair of example images that suffer from defocus blur  
and motion blur; (b,d) show the corresponding blur maps that are  
constructed based on the proposed singular value feature. . . . . 95
- 9.3 Framework of the proposed image blurred region detection and  
classification technique. . . . . 96
- 9.4 A pair of example images suffering from motion blur image and  
defocus blur and their corresponding  $\nabla\alpha$  distributions in Hough  
space (a clear white circle region appears in  $\nabla\alpha$  distribution of the  
defocus blur image as highlighted by a red color circle in (b)). . . 97
- 9.5 Selected samples of blurred/non-blurred image regions from our  
dataset. . . . . 99
- 9.6 Illustration of the Recall-Precision curve of our classification method.  
(a) the recall-precision curve of 'blur' in blur/non-blur classifica-  
tion using singular value feature. (b) the recall-precision curve  
of 'defocus blur' in motion/defocus blur classification using alpha  
channel feature. . . . . 99
- 9.7 Illustration of blurred and non-blurred image region extractions by  
several example images: the red curves separate the blurred and  
non-blurred images regions where the image in (a) has a blurred  
background and the image in (b) has a blurred foreground. . . . . 101

9.8 Blurred region extraction using different thresholds. The document image in (a) contains defocus blur of different extents. Its corresponding singular value map is shown in (b), those regions with different blur degrees are highlighted in different color. (c) and (d) show the two extracted blurred image regions of (a) when the threshold is set at 0.91 and 0.76, respectively. . . . . 102

9.9 Comparison of blurred image region extraction: (a-c) show the blurred image regions that are extracted by using Levin’s method [7], Liu et al.’s method [8], and our proposed method, respectively. The images in (b) and (c) are adopted from [8] . . . . . 103

9.10 Images ranked based on the estimated blurry degree D: the proposed D in Equation 9.12 captures the image blurry degree properly, i.e., images are blurred more severely with the increase of D. . . . . 104

9.11 Procedure of constructing the alpha channel map. (a) is the input blurry document, and is divided into blocks, which is illustrated by red solid lines, and the overlapped border is illustrated by blue dot lines, (b) shows one block taken from (a), (c) and (d) illustrate the histogram distribution and corresponding Gaussian mixture distribution of (b), respectively. . . . . 107

9.12 Distribution of  $\nabla\alpha$  on 2D  $(\nabla\alpha_x, \nabla\alpha_y)$  coordinate and Hough domain, the origin is in the center. . . . . 110

9.13 Restoration Results of motion blurred document images using different methods. The first column is the blurred images, the second column is the corresponding recovered images by cepstrum method, the third column is the corresponding recovered images by proposed method, the last column is the origin clear images. . . . . 112

9.14 Four motion blurred document image examples in the first column and corresponding recovered images by our proposed method in the second column, Shan et al.'s method [9] in the third column and Qi's method [6] in the fourth column, respectively. . . . . 113

# List of Tables

2.1	Document Image Binarization Methods . . . . .	11
6.1	Evaluation Results of the dataset of DIBCO 2009 . . . . .	43
6.2	Evaluation Results of the dataset of H-DIBCO 2010 . . . . .	43
6.3	Evaluation Results of the dataset of DIBCO 2011 . . . . .	44
6.4	Evaluation Results of H-DIBCO 2012 . . . . .	45
6.5	Evaluation Results of Bickley diary dataset . . . . .	46
7.1	Evaluation results of Sauvola’s, Niblack’s, Otsu’s methods and proposed framework . . . . .	56
7.2	Evaluation Results of the dataset of DIBCO 2009 . . . . .	63
7.3	F-Measure evaluation of our proposed framework . . . . .	69
8.1	Evaluation of the recognition results on the Robust Reading Competition Dataset using Google Tesseract OCR . . . . .	83
8.2	Evaluation of the recognition results on the Robust Reading Competition Dataset using Abbyy OCR . . . . .	84
8.3	Recognition results of the web images in Figure 8.1 using Google Tesseract . . . . .	85



8.4 Recognition results of the web images in Figure 8.1 using Abbyy . 86

# Chapter 1

## Introduction of Document Image Enhancement

### 1.1 Background and Motivation

There is huge amount of textual information that is embedded within images. For example, more and more documents are digitalized everyday via camera, scanner and other equipment, many digital images contain texts, and a large amount of textual information is embedded in web images. It would be very useful to turn the characters from image format to textual format by using optical character recognition (OCR). This converted text information is very important for document mining, document image retrieval and so on. However, in many cases, the document images cannot be directly fed to an OCR system due to the following reasons:

- The original document papers suffer from different kinds of degradation including smear, ink-bleeding through and intensity variation, especially

---

for historical documents.

- The process of obtaining digital images from the real world is not perfect. There are many factors that may cause image distortion, such as incorrect focal length, over/under exposure, camera shaking/object movement, low resolution, etc.
- The web images in the internet are often susceptible to certain image degradation such as low resolution and small size, which is specially designed for faster network transmission rate, computer-generated-character artifacts, and special effects on images to attract visual attention.

Document Image Enhancement is a technique that improves the quality of a document image to enhance human perception and facilitate subsequent automated image processing. It is widely used in the pre-processing stage of different document analysis tasks. Document image enhancement problem is essentially an ill-posed problem, because a number of enhanced images can be generated from the same input image. Moreover, the quality of enhancement techniques is mainly judged by human perception, which makes the quantitative measures hard to be applied.

The main aim of this study is to propose some document image enhancement techniques for better accessibility to the textual information embedded in the images. The specific objectives of this research are to:

- Propose some document binarization techniques for degraded document images that achieved good performance for degraded documents and can be used in different document analysis applications.

- 
- Develop better frameworks for improving and combining existing binarization methods by employing domain knowledge and image statistics.
  - Explore enhancement techniques of low quality images for better text recognition performance of web image.
  - Study blurred region detection and classification techniques that can be used in different multimedia analysis applications and investigate restoration methods for blurred document images.

The proposed techniques can be used in different applications, such as optical character recognition, document image retrieval, optical musical recognition, image segmentation, depth recovery and image retrieval.

## 1.2 Scope of Study

There are many different kinds of document enhancement techniques which handle differently distorted document images, such as document image dewarping [10] and document image super-resolution [11]. In this thesis, we focus on three aspects of the document enhancement techniques: document image binarization, web image enhancement and document image deblurring. These techniques are widely used in different kinds of applications. I explored these topics during my Ph.D. study and proposed better document image enhancement techniques for different document images.

---

## 1.3 Organization of this thesis

The following is a road map of the remaining chapters of this thesis.

The first part of this thesis discusses a few topics on the document image binarization area. First, a literature review of the existing binarization techniques is provided in **Chapter 2**. Generally speaking, these binarization techniques can be divided into two categories, namely global thresholding methods and local thresholding methods. However, there are a number of limitations of the state-of-the-art techniques, which decrease their performance on some kinds of degraded document images due to smear and smug, low contrast, ink bleed-through and so on and so forth. In order to address these problems, we propose a set of binarization techniques for degraded document images in **Chapter 3**, **Chapter 4**, and **Chapter 5**. In **Chapter 6**, we conduct a few experiments to demonstrate the effectiveness of our proposed binarization techniques. Furthermore, we illustrated a set of learning framework to improve the existing binarization techniques in **Chapter 7**.

The second part of this thesis covers other document image enhancement techniques during my Ph.D. study. In particular, **Chapter 8** deals with the text recognition problem on web images. We propose a robust text recognition technique for web images that make use of L0 smoothing, further work need to be done to improve the accessibility of the textual information embedded in web images. **Chapter 9** discusses the area of document image deblurring. We propose a blurred region detection and classification method to effectively segment blur/non-blur and motion/defocus blur regions for further processing. In addition, a motion blur restoration technique is provided to address the motion blur

---

problem in document images.

**Chapter 10** summarizes the current and potential contributions of this research work and discusses the future research directions. The publications that arise from my research work are also listed in the end.

## Chapter 2

# Literature Review of Document Image Binarization

Document image binarization is usually performed in the preprocessing stage of different document image processing related applications such as optical character recognition (OCR) and document image retrieval. It converts a gray-scale document image into a binary document image and accordingly facilitates the ensuing tasks such as document skew estimation and document layout analysis. As

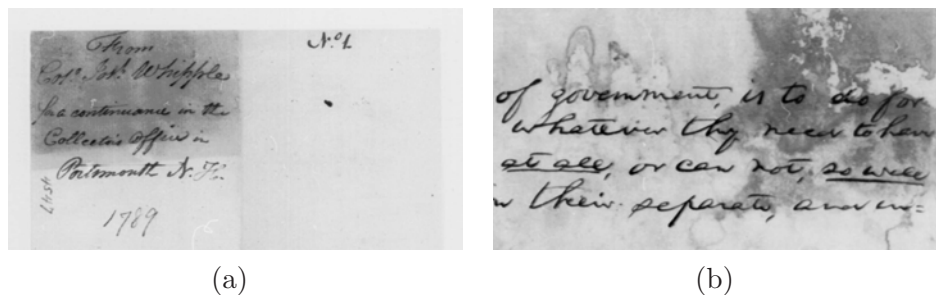
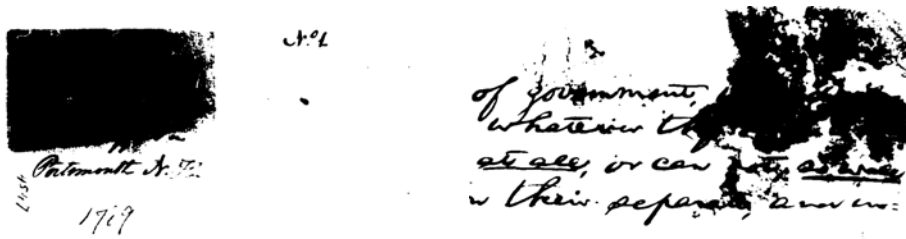


Figure 2.1: Two degraded document image examples, which are obtained from Document Image Binarization Contest (DIBCO) [1] dataset



(a) Binarized image of Figure 2.1(a) (b) Binarized image of Figure 2.1(b)

Figure 2.2: Binarization Results using Otsu's method of images in Figure 2.1

more and more text documents are scanned, fast and accurate document image binarization is becoming increasingly important.

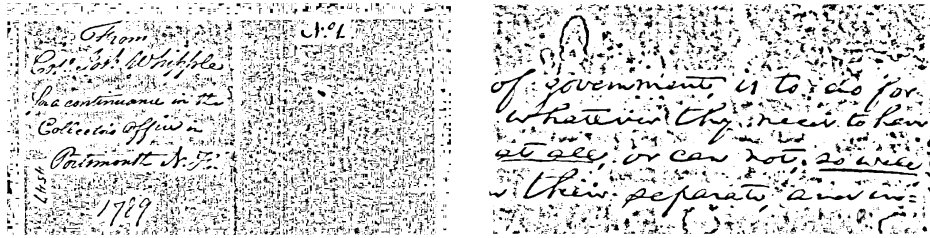
## 2.1 Previous Work

Generally speaking, the binarization techniques are either global or local. The global binarization techniques assign a single threshold for the whole document image, and the local binarization techniques find a threshold for each pixel in the document image. One of the famous global thresholding methods is Otsu's method [12], which is a histogram shape-based image thresholding technique. Otsu's method tries to estimate a global threshold that minimizes the intra-class variance, which is defined as a weighted sum of variances of the two classes:

$$\delta_{\omega}^2(t) = \omega_1(t)\delta_1^2(t) + \omega_2(t)\delta_2^2(t). \quad (2.1)$$

where the term  $\omega_i$  is the probabilities of the two classes separated by a threshold





(a) Binarized image of Figure 2.1(a) (b) Binarized image of Figure 2.1(b)

Figure 2.3: Binarization Results using Niblack's method of images in Figure 2.1

$t$  and the variances of these classes  $\delta_i$ . The term  $\omega_i$  is defined as follows:

$$\omega_1 = \sum_{i=1}^{t-1} p(i); \omega_2 = \sum_{i=t}^n p(i); \quad (2.2)$$

where the variable  $p(i)$  denotes the number of pixels with gray value level  $i$ .

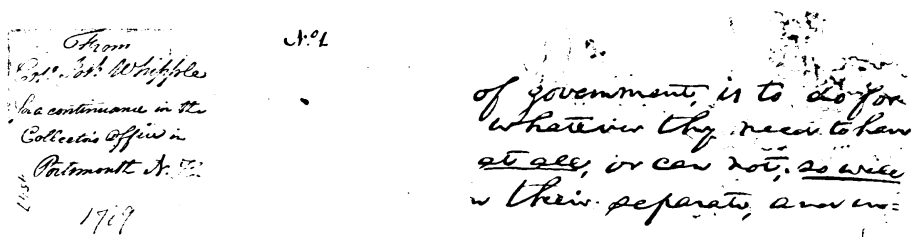
And one of the famous local thresholding methods is Niblack's method [13], which estimates the local threshold by using the local mean  $m$  and the standard variation  $s$ . The local threshold is computed as follows:

$$T = m + k \cdot s, \quad (2.3)$$

where the parameter  $k$  is a user defined parameter and it normally lies between -1 and 0.

The main drawback of this window-based thresholding approach is that the thresholding performance depends heavily on the window size and hence the character stroke width. Sauvola et. al. [14] later modify the formula in Equation 2.3 and propose a new thresholding formula as follows:

$$T = m \cdot (1 + k \cdot (\frac{s}{R} - 1)), \quad (2.4)$$



(a) Binarized image of Figure 2.1(a) (b) Binarized image of Figure 2.1(b)

Figure 2.4: Binarization Results using Sauvola’s method of images in Figure 2.1

where the parameter  $R$  refers to the dynamic range of the standard deviation and the parameter  $k$  instead takes a positive value between 0 and 1. The new thresholding formulas reduce the background noise greatly, but it requires the knowledge of document contrast to set the parameter  $R$  properly.

Figures 2.2, 2.3, 2.4 show the binarization results of the sample document images in Figure 2.1. As shown in the results, Otsu’s method [12] requires a bimodal histogram pattern and so cannot handle these document image with severe background variation. Adaptive thresholding methods such as Niblack’s/Sauvola’s [13, 14] method may either introduce a certain amount of noise or fail to detect the document text with a low image contrast.

Many works [15, 16] have been reported to deal with the high variation within historical document images. As many historical documents do not have a clear bimodal pattern, global thresholding [12, 17, 18] is usually not a suitable approach for the historical document binarization. Adaptive thresholding [13, 14, 19, 20, 21, 22], which estimates a local threshold for each document image pixel, is usually a better approach to handle the high variation associated with historical document images. For example, the early window-based adaptive thresholding techniques [13, 14] estimate the local threshold by using the mean

---

and the standard variation of image pixels within a local neighborhood window.

There are other approaches have been developed. Background Subtraction [23, 24] tries to subtract a background from the degraded images and use it to binarize the document images, however it is hard to model the document background and separate it from foreground text. Image contrast and edge information [25] which are good indicators of text strokes are used to remove the non-uniform background, although it is difficult to identify the difference between text stroke edges and document background noise. Some domain knowledge such as Texture feature [26] and cross section sequence graph analysis [27] can also be used to produce better results. But they requires some prior knowledge to the testing document images. Decomposition method [28] tried to divide the document images into smaller regions which are more uniform and easier to be binarized. Energy-based method [29] employs graph-cut algorithm to segment text information by minimizing Laplacian energy. In conclusion, these approaches combine different types of image information and domain knowledge and are often complex and time consuming. Table 2.1 shows most state-of-the-art document image binarization techniques with their strengths and weaknesses.

## 2.2 Challenges on Degraded Document Image Binarization

Though document image binarization has been studied for many years, the thresholding of degraded document images is still an unsolved problem. This can be explained by the fact that the modeling of the document foreground/background

---

Table 2.1: Document Image Binarization Methods

Methods	Pros	Cons
Global Thresholding	Fast, Produce good results on clean documents	Fail on degraded images
Local Thresholding	Works on degraded documents	Sensitive to window size
Background Subtraction	Produce good results when foreground varies	Performance decreased when background non-uniform
Image Contrast	Produce good results when background varies	Performance decreased when foreground non-uniform
Domain Knowledge	Preserve text info using domain knowledge	Hard to extract proper domain knowledge
Energy Based	Simple but effective	Need to tune a few parameters

is very difficult due to various types of document degradation such as uneven illumination, image contrast variation, bleeding-through, and smear as illustrated in Figure 2.1. The recent Document Image Binarization Contests (DIBCO) [1, 30] held under the framework of the International Conference on Document Analysis and Recognition (ICDAR) 2009 and 2011 and the Handwritten Document Image Binarization Contest(H-DIBCO) [31] held under the framework of the International Conference on Frontiers in Handwritten Recognition (ICFHR) show recent efforts on this issue. These contests partially reflect the current efforts on this task as well as the common understanding that further efforts are required for better document image binarization solutions.

Many practical document image binarization techniques have been applied on the commercial document image processing systems. These techniques perform well on the documents which do not suffer from serious document degradation. However, the degraded document image binarization is not fully explored and

---

still needs further research.

## Chapter 3

# Document Image Binarization using Local Maximum and Minimum

This chapter presents a simple but efficient historical document image binarization technique that is tolerant to different types of document degradation such as uneven illumination and document smear. The proposed technique makes use of the image contrast that is evaluated based on the local maximum and minimum. The overall flowchart is shown in Figure 3.1. Given a document image, it first constructs a contrast image and then extracts the high contrast image pixels by using Otsu's global thresholding method. After that, the text pixels are classified based on the local threshold that is estimated from the detected high contrast image pixels. The proposed method has been tested on the dataset that is used in the recent DIBCO contest series. Experiments show that the proposed method outperforms most reported document binarization methods.

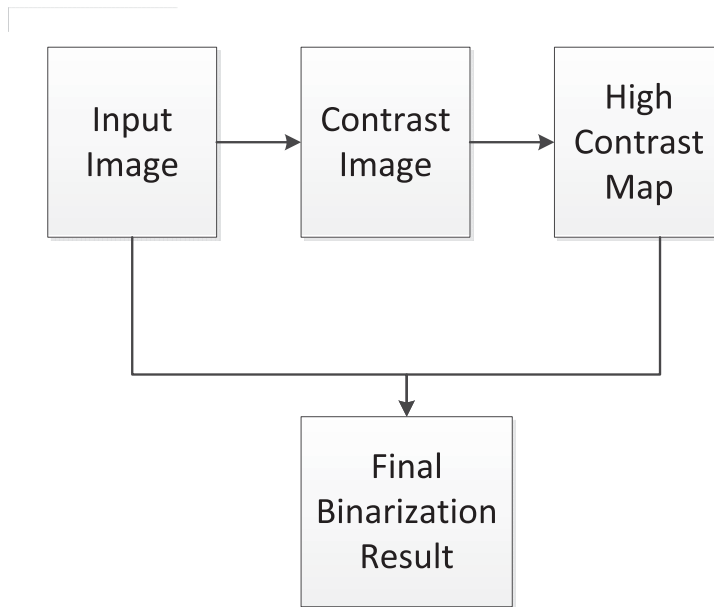


Figure 3.1: The flowchart of Binarization using local maximum and minimum

### 3.1 Contrast Image Construction

The image gradient has been widely used in the literature for edge detection [2]. However, the image gradient is often obtained by the absolute image difference within a local neighborhood window, which does not incorporate the image intensity itself and is so sensitive to the image contrast/brightness variation. Take an unevenly illuminated historical document image as an example. The gradient of an image pixel (around the text stroke boundary) within bright document regions may be much higher than that within dark document regions. To detect the high contrast image pixels around the text stroke boundary properly, the image gradient needs to be normalized to compensate for the effect of the image contrast/brightness variation. At the same time, the normalization suppresses the variation within the document background as well.

---

In the proposed technique, we suppress the background variation by using an image contrast that is calculated based on the local image maximum and minimum [3] as follows:

$$D(x, y) = \frac{f_{max}(x, y) - f_{min}(x, y)}{f_{max}(x, y) + f_{min}(x, y) + \epsilon} \quad (3.1)$$

, where the terms  $f_{max}(x, y)$  and  $f_{min}(x, y)$  refer to the maximum and the minimum image intensities within a local neighborhood window. In the implemented system, the local neighborhood window is a  $3 \times 3$  square window. The term  $\epsilon$  is a positive but infinitesimally small number, which is added in case the local maximum is equal to 0.

The image contrast in Equation 3.1 lowers the image background and brightness variation properly. In particular, the numerator (i.e. the difference between the local maximum and the local minimum) captures the local image difference that is similar to the traditional image gradient [2]. The denominator acts as a normalization factor that lowers the effect of the image contrast and brightness variation. For image pixels within bright regions around the text stroke boundary, the denominator is large, which neutralizes the large numerator and accordingly results in a relatively low image contrast. But for image pixels within dark regions around the text stroke boundary, the denominator is small, which compensates the small numerator and accordingly results in a relatively high image contrast. As a result, the contrasts of image pixels (lying around the text stroke boundary) within both bright and dark document regions converge close to each other and this facilitates the detection of high contrast image pixels lying around the text stroke boundary.



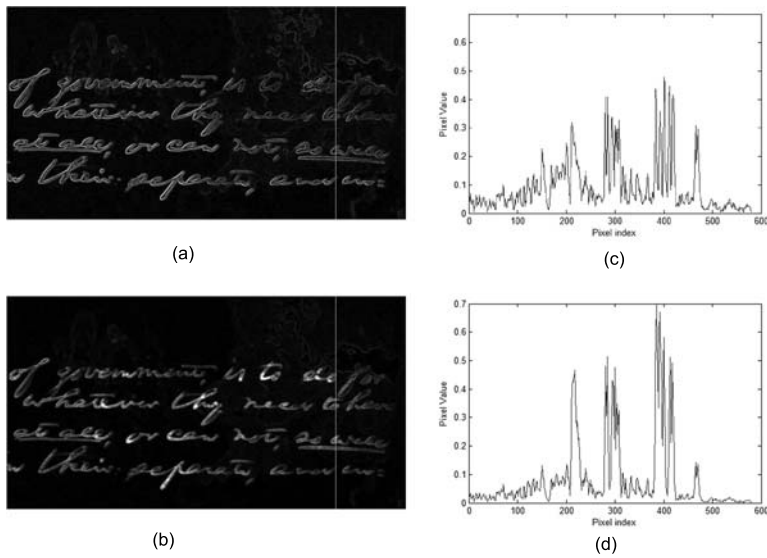


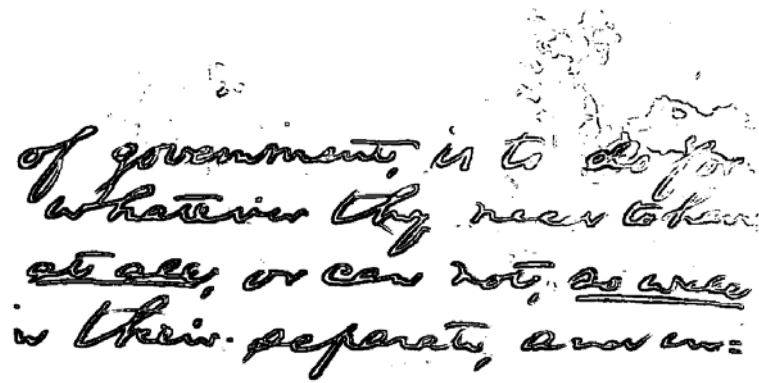
Figure 3.2: The gradient and contrast map: (a) The traditional image gradient that is obtained using Canny's edge detector [2]; (b) The image contrast that is obtained by using the local maximum and minimum [3];(c) One column of the image gradient in Figure 3.2(a) (shown as a vertical white line);(d) The same column of the contrast image in Figure 3.2(b).

---

At the same time, the image contrast in Equation 3.1 suppresses the variation within the document background properly. For document background pixels, the local minimum is usually much brighter than that of the image pixels lying around the text stroke boundary. As a result, the contrast of the document background pixels will be suppressed due to the high denominator. With the same reason, the image pixels with similar image gradient lying around the text stroke boundary in dark regions will have a much higher image contrast. This enhances the discrimination between the image pixels around the text stroke boundary and those within the document background region with high variation because of the document degradation.

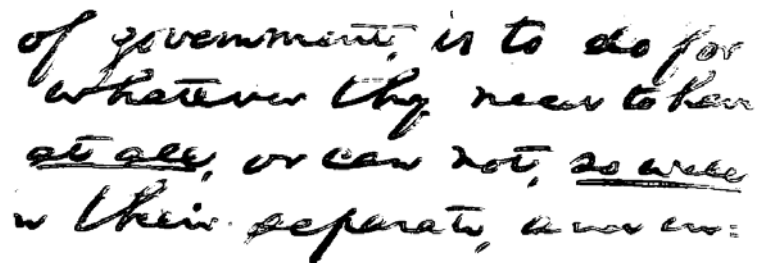
Figure 3.2 illustrates the difference between the image gradient and the image contrast defined in Equation 3.1. In particular, Figure 3.2(a) and 3.2(b) show the gradient image and the contrast image, respectively. As Figure 3.2(a) shows that the image gradient around text stroke boundary varies visibly from the bright document regions to the dark document regions. However,, as shown in Figure 3.2(b), the image contrast around text stroke boundary varies little from the bright document regions to the dark document regions. At the same time, the discrimination between the image contrast around the text stroke boundary and that around the document background is much stronger compared with the discrimination of the the image gradient around the text stroke boundary and that around the document background. These two points can be further illustrated in Figure 3.2(c) and 3.2(d) where the same column of the gradient image in Figure 3.2(a) and the contrast image in Figure 3.2(b) is plotted, respectively.

---



of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answers:

(a)



of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answers:

(b)

Figure 3.3: High contrast pixel detection: (a) Global thresholding of the gradient image in Figure 3.2(a) by using Otsu's method; (b) Global thresholding of the contrast image in Figure 3.2(b) by using Otsu's method.

## 3.2 High Contrast Pixel Detection

The purpose of the contrast image construction is to detect the desired high contrast image pixels lying around the text stroke boundary. As described in the last subsection, the constructed contrast image has a clear bi-modal pattern where the image contrast around the text stroke boundary varies within a small range but is obviously much larger compared with the image contrast within

---

the document background. We therefore detect the desired high contrast image pixels (lying around the text stroke boundary) by using Otsu's global thresholding method.

Figure 3.3(a) and (b) show the binarization results of the gradient image in Figure 3.2(a) and the contrast image in Figure 3.2(b), respectively, by using Otsu's global thresholding method. As Figure 3.3(b) shows, most of the high contrast image pixels detected through the binarization of the contrast image correspond exactly to the desired image pixels around the text stroke boundary. On the other hand, the binarization of the gradient image in Figure 3.3(a) introduces a certain amount of undesired pixels that usually lie within the document background.

### 3.3 Historical Document Thresholding

The text pixels can be classified from the document background pixels once the high contrast image pixels around the text stroke boundary are detected properly. The document thresholding from the detected high contrast image pixels is based on two observations. First, the text pixels should be close to the detected high contrast image pixels because most detected high contrast image pixels lie around the text stroke boundary. Second, based on the assumption that foreground text pixels get low gray scale values, the intensity of most text pixels should be close or lower than the average intensity of the detected high contrast image pixels within a local neighborhood window. This can be similarly explained by the fact that most detected high contrast image pixels lie around the text stroke boundary.

For each document image pixel, the number of the detected high contrast

---

image pixels is first determined within a local neighborhood window. The document image pixel will be considered a text pixel candidate if the number of high contrast image pixels within the neighborhood window is larger than a threshold. The document image pixel can thus be classified based on its intensity relative to that of its neighboring high contrast image pixels as follows:

$$R(x, y) = \begin{cases} 1 & N_e \geq N_{min} \quad \&\& I(x, y) \leq E_{mean} + E_{std}/2 \\ 0 & \text{otherwise} \end{cases}, \quad (3.2)$$

where the terms  $E_{mean}$  and  $E_{std}$  are the mean and the standard deviation of the image intensity of the detected high contrast image pixels (within the original document image) within the neighborhood window that can be evaluated as follows:

$$E_{mean} = \frac{\sum_{neighborhood} I(x, y) * (1 - E(x, y))}{N_e},$$

$$E_{std} = \sqrt{\frac{\sum_{neighborhood} ((I(x, y) - E_{mean}) * (1 - E(x, y)))^2}{N}}$$

where the variable  $I$  refers to the input document image and the variable pair  $(x, y)$  denotes the position of the document image pixel under study. The parameter  $E$  refers to the binary high contrast pixel image where the term  $E(x, y)$  is equal to 0 if the document image pixel is detected as a high contrast pixel. The parameter  $N_e$  refers to the number of high contrast image pixels that lie within the local neighborhood window. So if  $N_e$  is larger than  $N_{min}$  and  $I(x, y)$  is smaller

---

than  $E_{mean} + E_{std}/2$ , the term  $R(x, y)$  is set at 1. Otherwise, the term  $R(x, y)$  is set at 0.

There are two parameters that need to be set properly, namely, the size of the neighborhood window and the minimum number of the high contrast image pixels  $N_{min}$  within the neighborhood window. These two parameters are both correlated to the width of the text strokes within the document image under study. In particular, the size of the neighborhood window should not be smaller than the text stroke width. Otherwise the text pixels within the interior of the text strokes may not be classified correctly because the local neighborhood window may not enclose enough high contrast image pixels. At the same time, the minimum number of the high contrast image pixels (within the neighborhood window) should be around the size of the local neighborhood window based on the double-edge structure of the character strokes. The calculation of the text stroke width will be discussed on Section [5.3](#).

## Chapter 4

# Document Image Binarization using Background Estimation

This chapter presents a document binarization technique using background estimation and stroke edge information. This document binarization technique is based on the observations that the text documents usually have a document background of the uniform color and texture and the document text within it has a different intensity level compared with the surrounding background. The technique makes use of the document background surface and the text stroke edge information. The overall flowchart is shown in Figure 4.1 It first estimates a document background surface through an iterative polynomial smoothing procedure. The text stroke edges are then detected by combining the local image variation and the estimated document background surface. After that, the document text is segmented based on the local threshold that is estimated from the detected stroke edge pixels. At the end, a series of post-processing operations are performed to further improve the document binarization performance.

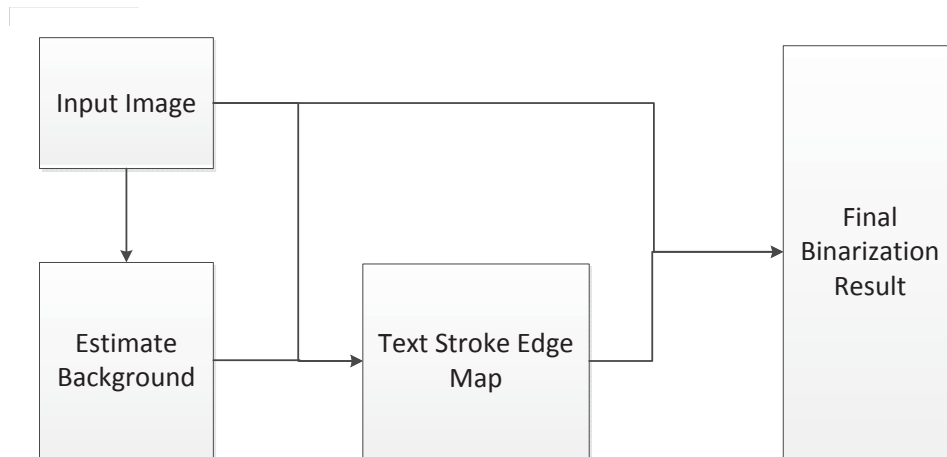


Figure 4.1: The flowchart of Binarization using background estimation

## 4.1 Document Background Estimation

The polynomial smoothing procedure can be summarized as follows. First, a set of equidistant pixels are sampled from a document row/column. The signal at each sampling pixel is estimated by the median intensity of the document image pixels within a local one-dimensional neighborhood window. The initial polynomial smoothing setting can therefore be specified as follows:

$$\begin{aligned}
 X_i &= k_s \times i \\
 Y_i &= f_{mdn}([I(X_{f_{rnd}}(i - 5 \cdot k_s)), \dots, I(X_{f_{rnd}}(i + 5 \cdot k_s))]), i = 1, \dots, N
 \end{aligned}
 \tag{4.1}$$

where the term  $k_s$  denotes the sampling step as well as the size of the local neighborhood window. Functions  $f_{mdn}(\cdot)$  and  $f_{rnd}(\cdot)$  denote a median and a rounding functions, respectively. The variables  $X_i$  and  $Y_i$  refer to the position of the  $i_{th}$  sampling pixel and the sampled image intensity at that sampling pixel,



---

respectively. The sampling index  $i$  changes from 1 to  $N$ .  $N$  refers to the number of the image pixels sampled from the document row/column under study. The number  $N$  is determined by the sampling step  $k_s$ . Our experiments show that the document thresholding performance changes little when  $k_s$  changes from 1 to 6.

---

**Algorithm 1** Polynomial Smooth of one row/column of a document image

---

**Require:**

One row/column image pixels,  $I(i)$

**Ensure:**

A smoothing polynomial of the background of the document image row/column under study

- 1: Sample the image data from the document row/column under study as specified in Equation 4.1.
  - 2: Fit a smoothing polynomial of the initial order  $d_0$  to the sampled image data.
  - 3: Evaluate the fitting maximum error between the sampled data and the fitted smoothing polynomial. Remove the sampling point with the maximum fitting error if the maximum fitting error is larger than a predefined threshold.
  - 4: Re-fit a smoothing polynomial of a higher order  $d_i$  to the remaining data points.
  - 5: Repeat the previous two steps iteratively until the maximum fitting error is small than the predefined threshold or the number of the remaining data points is small than  $d_i$ .
  - 6: **return** The final smoothing polynomial.
- 

And the background surface of the document row/column under study can thus be estimated iteratively as shown in Algorithm 1. As described in Algorithm 1, we pre-define a threshold to stop the iterative polynomial smoothing procedure. In our implemented system, the pre-defined threshold is set at 10 because the intensity difference between the document text pixels and the document background pixels is usually much larger than 10. In addition, we set the initial polynomial order  $d_o$  at 6 based on the observation that the polynomial of

---

order 6 in the initial iteration is usually sufficient to track the image variation within the document background. Furthermore, we increase the polynomial order adaptively (after each smoothing iteration) as follows to estimate the document background surface accurately:

$$d_n = d_o + f_{rnd}(k_t \times n), \quad (4.2)$$

where  $n$  denotes the iteration number and  $f_{rnd}(\cdot)$  refers to a rounding function.  $d_o$  and  $d_n$  denote the order of the initial smoothing polynomial and the smoothing polynomial at the  $n_{th}$  iteration, respectively. Parameter  $k_t$  specifies the increase speed of the polynomial order that can be set between 0.1 and 0.2.

We therefore further perform a column-by-column smoothing procedure to correct the estimation error that is introduced through the row-by-row smoothing procedure. The column-by-column smoothing is very similar to the row-by-row smoothing as described in Algorithm 1. The only difference is that the image data are sampled not from the original document image but from the document background surface estimated in the row-by-row smoothing stage.

## 4.2 Stroke Edge Detection

Then we detect the text stroke edges by combining the local image variation and the document background surface. In particular, we first compensate the document contrast variation (resulting from document degradation such as uneven illumination and smear) by using the estimated document background surface as

---

follows:

$$\bar{I} = \frac{C}{BG} \times I, \quad (4.3)$$

where the term  $C$  is a constant that controls the brightness of the compensated document images. In our implemented system, it is set at the median intensity of the document image under study to preserve the original document brightness. The parameter  $BG$  stands for the estimated document background surface. And the stroke edge pixels usually correspond to the ones that have the maximum local variation in either horizontal or vertical direction evaluated as follows:

$$\begin{aligned} V_h(x, y) &= |\bar{I}(x, y + 1) - \bar{I}(x, y - 1)| \\ V_v(x, y) &= |\bar{I}(x + 1, y) - \bar{I}(x - 1, y)| \end{aligned}, \quad (4.4)$$

where the term  $\bar{I}$  denotes the normalized document image under study. We therefore first detect a number of candidate text stroke edge pixels by the ones having the maximum local variations in either horizontal or vertical direction. Next, the local image variation of each candidate text stroke edge pixel can be evaluated as follows:

$$V(x, y) = V_h(x, y) + V_v(x, y), \quad (4.5)$$

where the terms  $V_h(x, y)$  and  $V_v(x, y)$  refer to the local image variations in horizontal and vertical directions, respectively. The local image variation of all detected candidate stroke edge pixels usually has a bimodal histogram pattern. And then the real text stroke edge pixels can therefore be detected by using Otsu's

---

method [12].

### 4.3 Threshold Estimation and Post-Processing

The classification step is the same as illustrate in subsection 3.3. Then in the post-processing step, we correct the thresholding error by using the estimated document background surface and some document domain knowledge. In particular, we observe that most falsely classified text pixels correspond to the background pixels that are often slightly darker than the (improperly detected) surrounding text stroke edge pixels. One typical characteristic of such classification error is that most misclassified document image pixels are much brighter than the surrounding text pixels. Generally, the image difference of the real text components is much larger compared with that of those misclassified text components. The segmented non-text components can therefore be identified and removed by using Otsu's Algorithm. In addition, we also observed that the document image thresholding often introduces certain amount of single-pixel holes, concavities, and convexities along the text stroke boundary. And the final binarization result after the post-processing where most thresholding error is corrected properly.

## Chapter 5

# A Robust Adaptive Document Image Binarization Technique for Degraded Document Images

This chapter presents a document binarization technique that extends our previous local maximum minimum method (which is discussed in Chapter 3) and the method used in the latest DIBCO 2011. The proposed method is simple, robust and capable of handling different types of degraded document images with minimum parameter tuning. It makes use of the adaptive image contrast that combines the local image contrast and the local image gradient adaptively and therefore is tolerant to the text and background variation caused by different types of document degradation. In particular, the proposed technique addresses the over-normalization problem of the local maximum minimum algorithm that is discussed in Chapter 3. At the same time, the parameters used in the algorithm can be adaptively estimated.

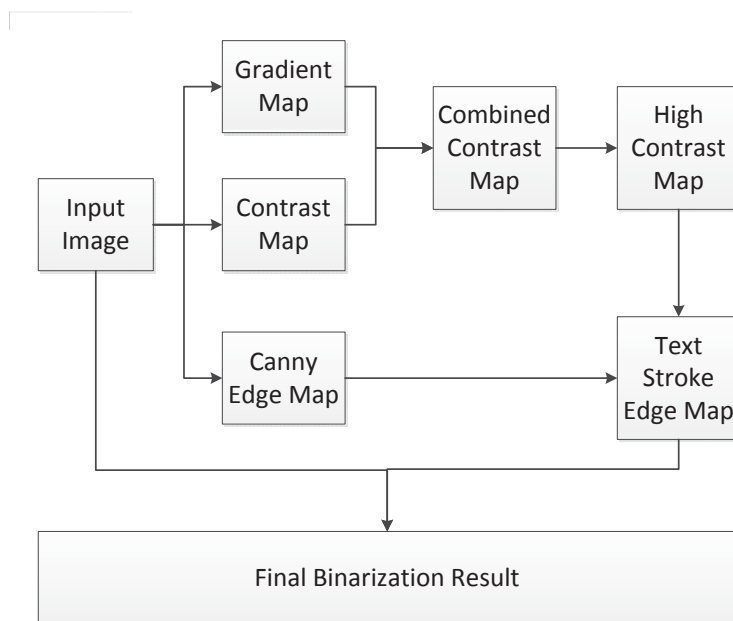


Figure 5.1: The flowchart of the proposed adaptive document image binarization technique

---

The overall flowchart is shown in Figure 5.1. Given a degraded document image, an adaptive contrast map is first constructed and the text stroke edges are then detected through the combination of the binarized adaptive contrast map and the canny edge map. The text is then segmented based on the local threshold that is estimated from the detected text stroke edge pixels. Some post-processing is further applied to improve the document binarization quality.

## 5.1 Contrast Image Construction

The image gradient has been widely used for edge detection [4] and it can be used to detect the text stroke edges of the document images effectively that have a uniform document background. On the other hand, it often detects many non-stroke edges from the background of degraded document that often contains certain image variations due to noise, uneven lighting, bleed-through, etc. To extract only the stroke edges properly, the image gradient needs to be normalized to compensate for the image variation within the document background.

In our earlier method in Chapter 3, The local contrast evaluated by the local image maximum and minimum is used to suppress the background variation. However, the image contrast has one typical limitation that it may not handle document images with the bright text properly. This is because a weak contrast will be calculated for stroke edges of the bright text. To overcome this over-normalization problem, we combine the local image contrast with the local image gradient and derive an adaptive local image contrast as follows:

$$C_a(i, j) = \alpha C(i, j) + (1 - \alpha)(I_{max}(i, j) - I_{min}(i, j)), \quad (5.1)$$



(a) Local Image Gradient



(b) Local Image Contrast



(c) Proposed Method

Figure 5.2: Contrast Images constructed using the local image gradient [4] in (a), the local image contrast [5] in (b), and our proposed method in (c), respectively.



---

where the term  $C(i, j)$  denotes the local contrast in Equation 3.1 and the term  $(I_{max}(i, j) - I_{min}(i, j))$  refers to the local image gradient that is normalized to  $[0, 1]$ . The variable  $\alpha$  is the weight between local contrast and local gradient that is controlled based on the document image statistical information. Ideally, the image contrast will be assigned with a high weight (i.e. large  $\alpha$ ) when the document image has significant intensity variation. Otherwise, the local image gradient will be assigned with a high weight.

We model the mapping from document image intensity variation to  $\alpha$  by a power function as follows:

$$\alpha = \left(\frac{Std}{128}\right)^\gamma, \quad (5.2)$$

where the term  $Std$  denotes the document image intensity standard deviation, and the variable  $\gamma$  is a pre-defined parameter. The power function has a nice property in that it monotonically and smoothly increases from 0 to 1 and its shape can be easily controlled by different  $\gamma$ . The variable  $\gamma$  can be selected from  $[0, \infty]$ , where the power function becomes a linear function when  $\gamma = 1$ . Therefore, the local image gradient will play the major role in Equation 5.1 when  $\gamma$  is large and the local image contrast will play the major role when  $\gamma$  is small.

Figure 5.2 shows the contrast map of the sample document images in Figure 6.1 (b) and (d) that are created by using local image gradient [2], local image contrast [5] and our proposed method in Equation 5.1, respectively.

The use of the local image contrast produces a better result as shown in Figure 5.2(b) compared with the result by the local image gradient as shown in Figure 5.2(a) (because the normalization factors in Equation 3.1 helps to sup-

---

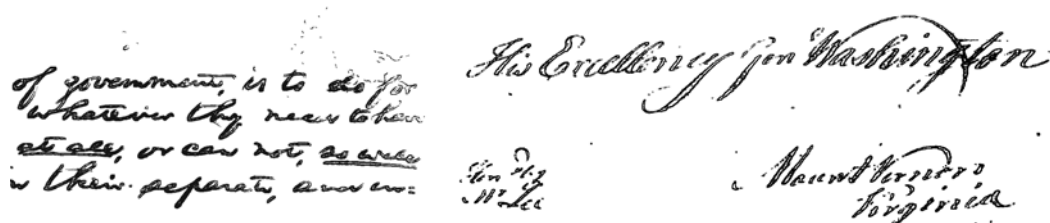
press the noise at the upper left area of Figure 5.2(a)). But the use of the local image contrast removes many light text strokes improperly in the contrast map as shown in Figure 5.2(b) whereas the use of local image gradient is capable of preserving those light text strokes as shown in Figure 5.2(a). As a comparison, the adaptive combination of the local image contrast and the local image gradient in Equation 5.1 can produce proper contrast maps for document images with different types of degradation as shown in Figure 5.2(c).

## 5.2 Text Stroke Edge Pixel Detection

The purpose of the contrast image construction is to detect the stroke edge pixels of the document text properly. The constructed contrast image has a clear bi-modal pattern, where the adaptive image contrast computed at text stroke edges is obviously larger than that computed within the document background. We therefore detect the text stroke edge pixel candidate by using Otsu's global thresholding method.

The purpose of the contrast image construction is to detect the stroke edge pixels of the document text properly. The constructed contrast image has a clear bi-modal pattern [5], where the adaptive image contrast computed at text stroke edges is obviously larger than that computed within the document background. We therefore detect the text stroke edge pixel candidate by using Otsu's global thresholding method. For the contrast images in Figure 5.2(c), Figure 5.3(a) shows a binary map by Otsu's algorithm [12] that extracts the stroke edge pixels properly.

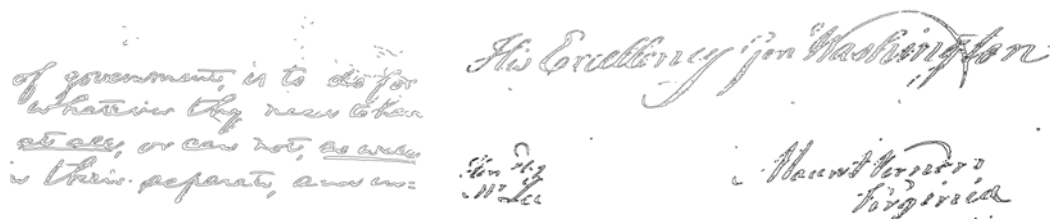
As the local image contrast and the local image gradient are evaluated by



(a) Binary Contrast Map



(b) Canny Edge Map



(c) Combined Text Stroke Edge Map

Figure 5.3: Binary contrast images, canny edge maps and their corresponding combined Edge maps, respectively.

the difference between the maximum and minimum intensity in a local window, the pixels at both sides of the text stroke will be selected as the high contrast pixels. The binary map can be further improved through the combination with the edges by Canny's edge detector [2]. It should be noted that Canny's edge detector by itself often extracts a large amount of non-stroke edges as illustrated in Figure 5.3(b) without tuning the parameter manually. In the combined map, we keep only pixels that appear within both the high contrast image pixel map and canny edge map. The combination helps to extract the text stroke edge

---

pixels accurately as shown in Figure 5.3(c).

### 5.3 Local Threshold Estimation

The text can then be extracted from the document background pixels once the high contrast stroke edge pixels are detected properly. Two characteristics can be observed from different kinds of document images as described in Chapter 3: First, the text pixels are close to the detected text stroke edge pixels. Second, there is a distinct intensity difference between the high contrast stroke edge pixels and the surrounding background pixels. The document image text can thus be extracted based on the detected text stroke edge pixels as follows:

$$R(x, y) = \begin{cases} 1 & I(x, y) \leq E_{mean} + \frac{E_{std}}{2} \\ 0 & \text{otherwise} \end{cases}, \quad (5.3)$$

where the terms  $E_{mean}$  and  $E_{std}$  are the mean and standard deviation of the intensity of the detected text stroke edge pixels within a neighborhood window  $W$ , respectively.

The size of the neighborhood window  $W$  is closely related to the stroke width  $EW$ . Generally, a larger local window size will help to reduce the classification error that is often induced by the lack of edge pixels within the local neighborhood window. The term  $W$  can be set around  $2EW$  because a larger local neighborhood window will increase the computational load significantly. The term  $EW$  can be estimated from the detected stroke edges as stated in Algorithm 2.

First the edge image is scanned horizontally row by row and the edge pixel candidates are selected as described in step 3. As the edge pixels should have

---

**Algorithm 2** Edge Width Estimation

---

**Require:** The Input Document Image  $I$  and Corresponding Binary Text Stroke Edge Image  $Edg$

**Ensure:** The Estimated Text Stroke Edge Width  $EW$

- 1: Get the *width* and *height* of  $I$
  - 2: **for** Each Row  $i = 1$  to *height* in  $Edg$  **do**
  - 3:   Scan from left to right to find edge pixels that meets the following criteria:
    - its label is 0(background)
    - the next pixel is labeled as 1(text).
  - 4:   Examine the intensities of those pixels selected in Step 3, and remove those pixels have a lower intensity than the following pixels next to it.
  - 5:   Match the remaining pixels into pairs, and calculate the distance between the two pixels in pair.
  - 6: **end for**
  - 7: Construct a histogram of those calculated distances.
  - 8: Use the most frequently occurring distance as the estimated stroke edge width  $EW$ .
- 

higher intensities than the following few pixels (which should be the text stroke pixels), those improperly detected edge pixels are removed in step 4. After that a histogram is constructed that records the frequency of the distance between two adjacent candidate pixels. The stroke edge width  $EW$  can then be approximately estimated by using the most frequently occurring distances of the adjacent edge pixels.

## 5.4 Post-Processing

Once the initial binarization result is derived as described in previous subsections, the binarization result can be further improved by incorporating certain domain knowledge as described in Algorithm 3. First, the isolated foreground pixels that do not connect with other foreground pixels are filtered out to make the edge

---

pixel set precisely. Second, the neighborhood pixel pair that lies on symmetric sides of a text stroke edge pixel should belong to different classes (i.e., either the document background or the foreground text). One pixel of the pixel pair is therefore labeled to the other category if both of the two pixels belong to the same class. Finally, some single-pixel artifacts along the text stroke boundaries are filtered out by using several logical operators as described in Chapter 4.

---

**Algorithm 3** Post Processing Procedure

---

**Require:** The Input Document Image  $I$ , Initial Binary Result  $B$  and Corresponding Binary Text Stroke Edge Image  $Edg$

**Ensure:** The Final Binary Result  $B_f$

- 1: Find out all the connect components of the stroke edge pixels in  $Edg$
  - 2: Remove those pixels that do not connect with other pixels.
  - 3: **for** Each remaining edge pixels  $(i, j)$ : **do**
  - 4:   Get its neighborhood pairs:  $(i - 1, j)$  and  $(i + 1, j)$ ;  $(i, j - 1)$  and  $(i, j + 1)$
  - 5:   **if** The pixels in the same pairs belong to the same class **then**
  - 6:     Assign the pixel with lower intensity to foreground class, and the other to background class.
  - 7:   **end if**
  - 8: **end for**
  - 9: Remove single-pixel artifacts along the text stroke boundaries after the document thresholding.
  - 10: Store the new binary result to  $B_f$ .
-

## Chapter 6

# Experiments and Discussions of the Proposed Binarization Methods

We participated in the recent Document Image Binarization Contests (DIBCO) [1, 30] held under the framework of the International Conference on Document Analysis and Recognition (ICDAR) 2009 and 2011 and the Handwritten Document Image Binarization Contest(H-DIBCO) [31] held under the framework of the International Conference on Frontiers in Handwritten Recognition (ICFHR) 2010. Our submitted method that based on the background estimation method in Chapter 4 performs the best among entries of 43 algorithms submitted from 35 international research groups in DIBCO 2009. In addition, our submitted methods based on our local maximum-minimum method in Chapter 3 achieved one of the top two scores among 17 submitted algorithms in H-DIBCO 2010 and the second best results among 18 submitted algorithms in the DIBCO 2011. The robust

---

adaptive binarization method described in Chapter 5 also has been submitted to the latest Document Image Binarization H-DIBCO 2012 contest held under ICFHR 2012.

In this chapter, A few experiments are designed to demonstrate the effectiveness and robustness of our proposed methods by quantitatively comparing our proposed methods, namely local maximum and minimum method(LMM) in Chapter 3, background estimation method(BE) in Chapter 4 and robust adaptive binarization method (RAB) in Chapter 5, with other state-of-the-art techniques on DIBCO 2009, H-DIBCO 2010 and DIBCO 2011 datasets. Finally, the proposed technique is further evaluated over a very challenging Bickley diary dataset<sup>1</sup> [32]. Figure 6.1 shows some degraded document image examples taken from those datasets.

## 6.1 Evaluation Metrics

The binarization performance is evaluated by using F-Measure, pseudo F-Measure, Peak Signal to Noise Ratio (PSNR), Negative Rate Metric (NRM), Misclassification Penalty Metric (MPM) and Distance Reciprocal Distortion (DRD) that are adopted from DIBCO 2009, H-DIBCO 2010 and DIBCO 2011 [1, 30, 31]. Due to the lack of ground truth data in some datasets (such as the skeleton ground truth), not all of the metrics are applied on every image. In particular, the F-Measure is defined as follows:

$$FM = \frac{2 * RC * PR}{RC + PR} \quad (6.1)$$

<sup>1</sup><http://www.comp.nus.edu.sg/~dfanbo/projects/BinarizationShop/dataset.htm>



---


$$RC = \frac{CTP}{CTP + CFN}$$

$$PR = \frac{CTP}{CTP + CFP}$$

where the terms  $RC$  and  $PR$  refer the recall and the precision of the method in Equation 6.1. The terms  $CTP$ ,  $CFP$ , and  $CFN$  denote the numbers of true positive pixels, false positive pixels, and false negative pixels, respectively. This measure evaluates how well an algorithm can retrieve the desire pixels.

The pseudo F-Measure is defined as follows [33]:

$$pFM = \frac{2 * pRC * PR}{pRC + PR} \quad (6.2)$$

$$pRC = \frac{\sum_{i,j} SG(i,j)\dot{B}(i,j)}{\sum_{i,j} SG(i,j)}, \quad (6.3)$$

where the term  $pRC$  refers the pseudo recall of the method in Equation 6.2, the term  $PR$  is the same as in Equation 6.1. The term  $SG$  denotes the skeletonized ground truth image that has 0 at background and 1 in text, respectively. The term  $B$  denotes the resultant binary image. This measure evaluates how well an algorithm can preserve the character skeleton.

The measure PSNR is defined as follows:

$$PSNR = 10 \log\left(\frac{C^2}{MSE}\right) \quad (6.4)$$

$$MSE = \frac{\sum_{x=1}^M \sum_{y=1}^N (I(x,y) - I'(x,y))^2}{MN},$$

where the term  $C$  is a constant that denotes the difference between foreground

---

and background. This constant can be set to 1. The term  $PSNR$  measures how close the resultant image to the ground truth image.

The measure NRM is defined as follows:

$$NRM = \frac{NR_{FN} + NR_{FP}}{2} \quad (6.5)$$

$$NR_{FN} = \frac{N_{FN}}{N_{FN} + N_{TP}},$$

$$NR_{FP} = \frac{N_{FP}}{N_{FP} + N_{TN}},$$

where the terms  $N_{TP}$ ,  $N_{FP}$ ,  $N_{TN}$ , and  $N_{FN}$  denote the number of true positives, false positives, true negatives, and false negatives respectively. This metric measures pixel mismatch rate between the ground truth image and resultant image.

The measure MPM is defined as follows:

$$MPM = \frac{MP_{FN} + MP_{FP}}{2} \quad (6.6)$$

$$MP_{FN} = \frac{\sum_{i=1}^{N_{FN}} d_{FN}^i}{D}$$

$$MP_{FP} = \frac{\sum_{j=1}^{N_{FP}} d_{FP}^j}{D},$$

where the terms  $d_{FN}^i$  and  $d_{FP}^j$  denote the distance of the  $i^{th}$  false negative and the  $j^{th}$  false positive pixel from the contour of the ground truth segmentation. The normalization factor  $D$  is the sum over all the pixel-to-contour distances of the ground truth object. This metric measures how well the resultant image represents the contour of ground truth image.

---

The measure DRD is defined as follows [34]:

$$DRD = \frac{\sum_{k=1}^S DRD_k}{N} \quad (6.7)$$

$$DRD_k = \sum_{i=-2}^2 \sum_{j=-2}^2 2|GT_k(x+i, y+j) - B_k(x, y)| \times W_{Nm}(i, j),$$

where the term  $DRD_k$  denotes the distortion of the  $k$ -th flipped pixel and calculated using a weight matrix  $W_{Nm}$  that defined in [34]. The pair  $(x, y)$  denotes the index of the  $k$ -th flipped pixel, The terms  $B$  and  $GT$  denote the resultant binary image and ground truth image, respectively. The term  $N$  refers to the number of the non-uniform (which contains background and text pixels)  $8 \times 8$  blocks in the GT image. This metric properly correlates with the human visual perception and measures the distortion for all the flipped pixels.

## 6.2 Experiments on competition datasets

In this experiment, we quantitatively compare our proposed method with other state-of-the-art techniques on DIBCO 2009, H-DIBCO 2010 and DIBCO 2011 datasets. These methods include Otsu’s method [12], Sauvola’s method [14], Niblack’s method [13], Bernsen’s method [25], Gatos et al.’s method [24].

The three datasets are composed of the same series of document images that suffer from several common document degradations such as smear, smudge, bleed-through and low contrast. The DIBCO 2009 dataset contains ten testing images that consist of five degraded handwritten documents and five degraded printed documents. The H-DIBCO 2010 dataset consists of ten degraded handwritten

Table 6.1: Evaluation Results of the dataset of DIBCO 2009

Methods	Otsu	Sauvola	Niblack	Bernsen	Gatos et al.	LMM	BE	RAB
F-Measure(%)	78.72	85.41	55.82	52.48	85.25	91.06	91.24	93.5
PSNR	15.34	16.39	9.89	8.89	16.5	18.5	18.6	19.65
NRM( $\times 10^{-2}$ )	5.77	6.94	16.40	14.29	10	7	4.31	3.74
MPM( $\times 10^{-3}$ )	13.3	3.2	61.5	113.8	0.7	0.3	0.55	0.43

Table 6.2: Evaluation Results of the dataset of H-DIBCO 2010

Methods	Otsu	Sauvola	Niblack	Bernsen	Gatos et al.	LMM	BE	RAB
F-Measure(%)	85.27	75.3	74.1	41.3	71.99	85.49	86.41	92.03
pseudo F-Measure(%)	90.83	84.22	85.4	44.4	74.35	92.64	88.25	94.85
PSNR	17.51	15.96	15.73	8.57	15.12	17.83	18.14	20.12
NRM( $\times 10^{-2}$ )	9.77	16.31	19.06	21.18	21.89	11.46	9.06	6.14
MPM( $\times 10^{-3}$ )	1.35	1.96	1.06	115.98	0.41	0.37	1.11	0.25

documents. The DIBCO 2011 dataset contains eight degraded handwritten documents and eight degraded printed documents. In total, we have 36 degraded document images with ground truth.

The evaluation results are shown in Table 6.1 and 6.2. As Table 6.1 and 6.2 show, our proposed methods achieve the highest scores in F-Measure, pseudo F-Measure, PSNR, and NRM and MPM under DIBCO dataset series. This means that our proposed methods produce a higher overall precision and preserve the text strokes better. In addition, our proposed methods also perform better than the 43 document thresholding algorithms submitted to the DIBCO 2009 [1] under DIBCO 2009 dataset and the 17 submitted algorithms in the H-DIBCO 2010 [31] under the H-DIBCO 2010 dataset. Figures 6.2, 6.3, 6.4 and 6.5 further show the binarization results of the four example document images in Figure 6.1 by using the eight document binarization methods. It is clear that our proposed methods extract the text better than the other comparison methods.

Besides the comparison methods mentioned above, our proposed methods are also compared with the top three algorithms, namely Lore et al.’s method [35], our submitted method and N. Howe’s method [29] for the DIBCO 2011 dataset. The quantitative results are shown in Table 6.3. As Table 6.3 shown, Our pro-

Table 6.3: Evaluation Results of the dataset of DIBCO 2011

Methods	F-Measure(%)	PSNR	DRD	MPM
Otsu [12]	82.22	15.77	8.72	15.64
Sauvola [14]	82.54	15.78	8.09	9.20
Niblack [13]	68.52	12.76	28.31	26.38
Bernsen [25]	47.28	7.92	82.28	136.54
Gatos et al. [24]	82.11	16.04	5.42	7.13
LMM	85.56	16.75	6.02	6.42
BE	81.67	15.59	11.24	11.40
Lelore et al. [35]	80.86	16.13	104.48	64.43
our submitted	85.2	17.16	15.66	9.07
N. Howe [29]	88.74	17.84	5.37	8.64
RAB	87.8	17.56	<b>4.84</b>	<b>5.17</b>

posed RAB method performs the best in terms of DRD and MPM, which means that our proposed technique maintains good text stroke contours and provides best visual quality. In addition, our proposed RAB method also performs well when being evaluated in pixel level. The F-Measure and PSNR of our proposed RAB method are very close to the highest scores, which is also shown in Table 6.3.

Figure 6.6 further shows one example image from the DIBCO 2011 dataset and its corresponding binary results produced by different methods. As shown in Figure 6.6, Bernsen’s method, Niblack’s method and Lelore et al.’s method fail to produce reasonable results. In addition, most of the methods including N. Howe’s method induce some background noise in the final results. our LMM method and our submitted method instead remove too much character strokes. On the other hand, our proposed RAB method produces a binary result with better visual quality and contains most of the text information.

We also participant in the latest H-DIBCO 2012 contest held under the ICFHR 2012. Our submitted method [36], which is described in Chapter 5, achieved the top three performance among all the submitting algorithms. As the testing images

---

Table 6.4: Evaluation Results of H-DIBCO 2012

Rank	Method	Score	FM(%)	p-FM(%)	PSNR	DRD	Time
1	6	172	89.47	90.18	21.80	3.440	29.66
2	11	340	92.85	93.34	20.57	2.660	0.26
<b>3</b>	<b>4.a</b>	<b>412</b>	<b>91.54</b>	<b>93.30</b>	<b>20.14</b>	<b>3.048</b>	<b>0.09</b>
4	13	435	90.38	95.09	19.30	3.348	0.92
5	7	494	91.85	92.19	19.65	3.056	0.66
6	8	501	89.98	92.07	19.44	3.761	19.44

are not released yet,

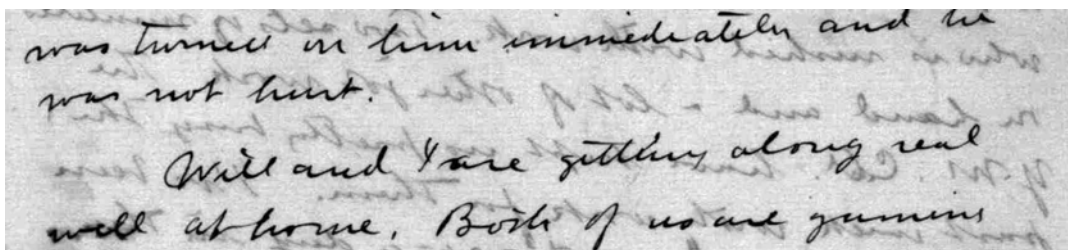
### 6.3 Testing on Bickley diary dataset

In the last experiment, we evaluate our methods on the Bickley diary dataset to show its robustness and superior performance. The images from Bickley diary dataset are taken from a photocopy of a diary that is written about 100 years ago. These images suffer from different kinds of degradation, such as water stains, ink bleed-through, and significant foreground text intensity and are more challenging than the previous DIBCO dataset series.

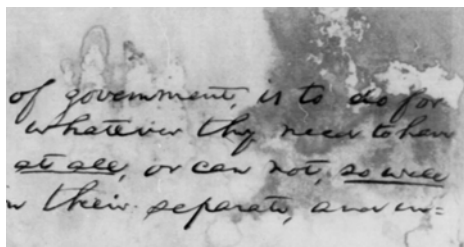
We use seven ground truth images that are annotated manually using Pix Labeler [37] to evaluate our proposed method with the other methods. Our proposed method achieves average 78.54% accuracy in terms of F-measure on the seven images with binary ground truth, which is at least 10% higher than the other seven methods. Detailed evaluation results are illustrated in Table 6.5. Figure 6.7 shows one example image from the Bickley diary dataset and its corresponding binarization results generated by different methods. It is clear that the proposed RAB algorithm performs better than other methods by preserving most textual information and producing the least noise.

Table 6.5: Evaluation Results of Bickley diary dataset

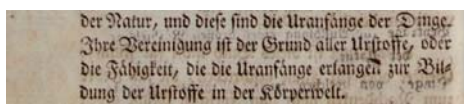
Methods	Otsu	Sauvola	Niblack	Bernsen	Gatos et al.	LMM	BE	RAB
F-Measure(%)	50.42	64.60	67.71	52.97	69.13	66.44	34.65	78.54
PSNR	7.58	11.62	9.79	7.71	11.44	10.76	3.54	13.15
NRM( $\times 10^{-2}$ )	21.41	23.26	9.52	18.86	21.89	17.50	40.78	12.92
MPM( $\times 10^{-3}$ )	196.98	28.97	105.17	193.35	36.57	72.08	370.15	16.71



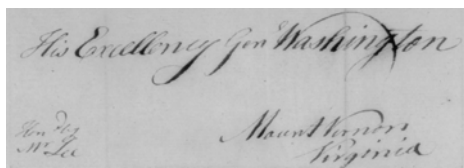
(a)



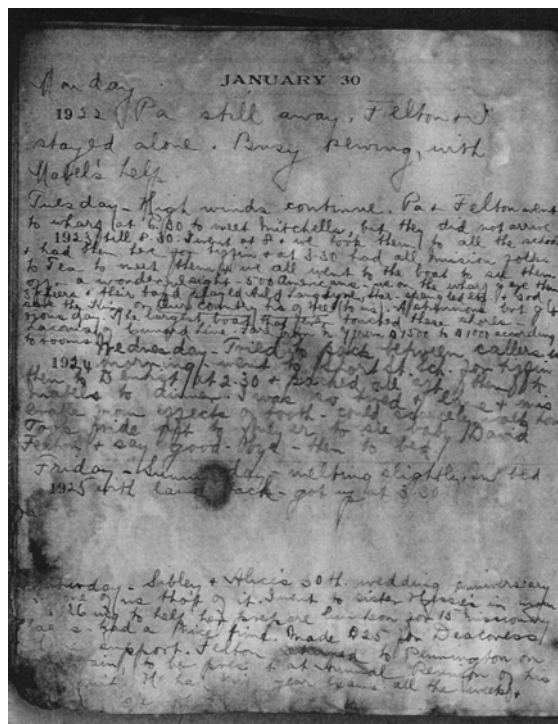
(b)



(c)



(d)



(e)

Figure 6.1: Five degraded document image examples taken from DIBCO dataset series and Bickley diary dataset.

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(a) Otsu's method

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(c) Niblack's method

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(e) Gatos et al.'s method

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(g) BE

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(b) Sauvola's method

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(d) Bernsen's method

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

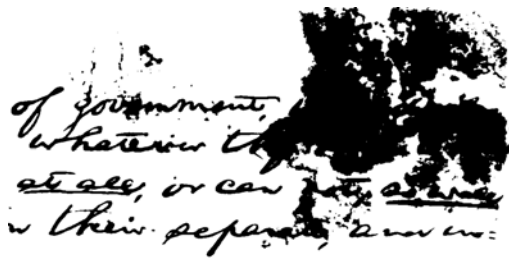
(f) LMM

was turned in him immediately and we  
was not hurt.  
Will and I are getting along real  
well at home. Both of us are getting

(h) RAB

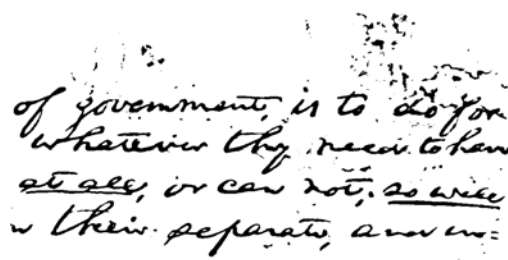
Figure 6.2: Binarization Results of the sample document image in Figure 6.1(a) produced by different methods.





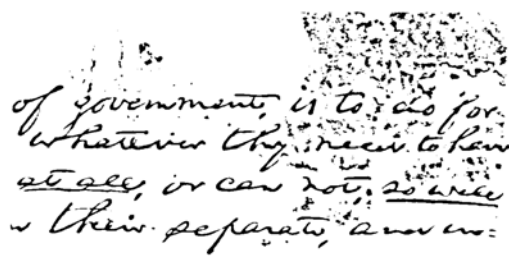
of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(a) Otsu's method



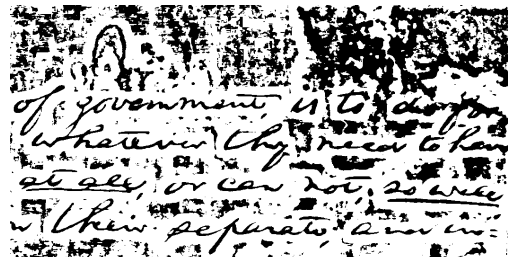
of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(b) Sauvola's method



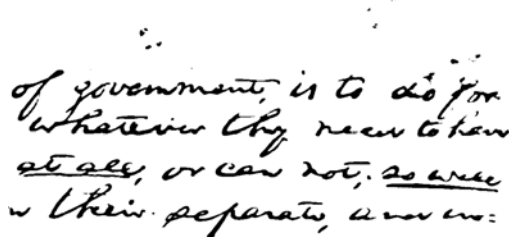
of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(c) Niblack's method



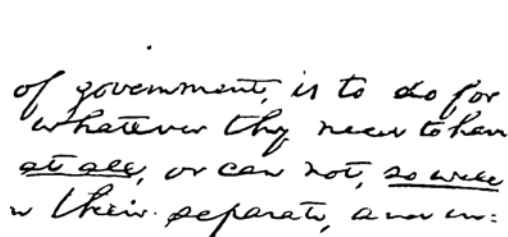
of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(d) Bernsen's method



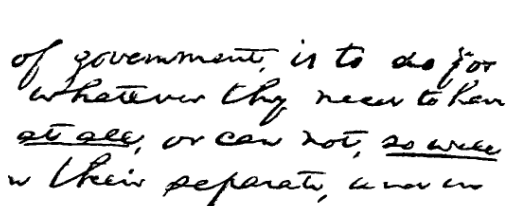
of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(e) Gatos et al.'s method



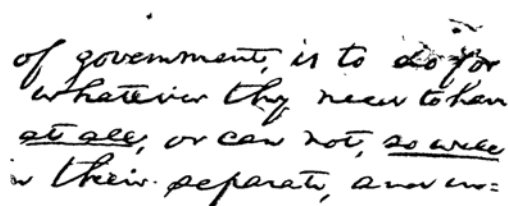
of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(f) LMM



of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(g) BE



of government, is to do for  
whatever they need to have  
at all, or can not, so well  
in their separate, answer:

(h) RAB

Figure 6.3: Binarization Results of the sample document image in Figure 6.1(b) produced by different methods.

---

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(a) Otsu's method

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(c) Niblack's method

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(e) Gatos et al.'s method

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(g) BE

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(b) Sauvola's method

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(d) Bernsen's method

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(f) LMM

der Natur, und diese sind die Uraufänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraufänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

(h) RAB

Figure 6.4: Binarization Results of the sample document image in Figure 6.1(c) produced by different methods.

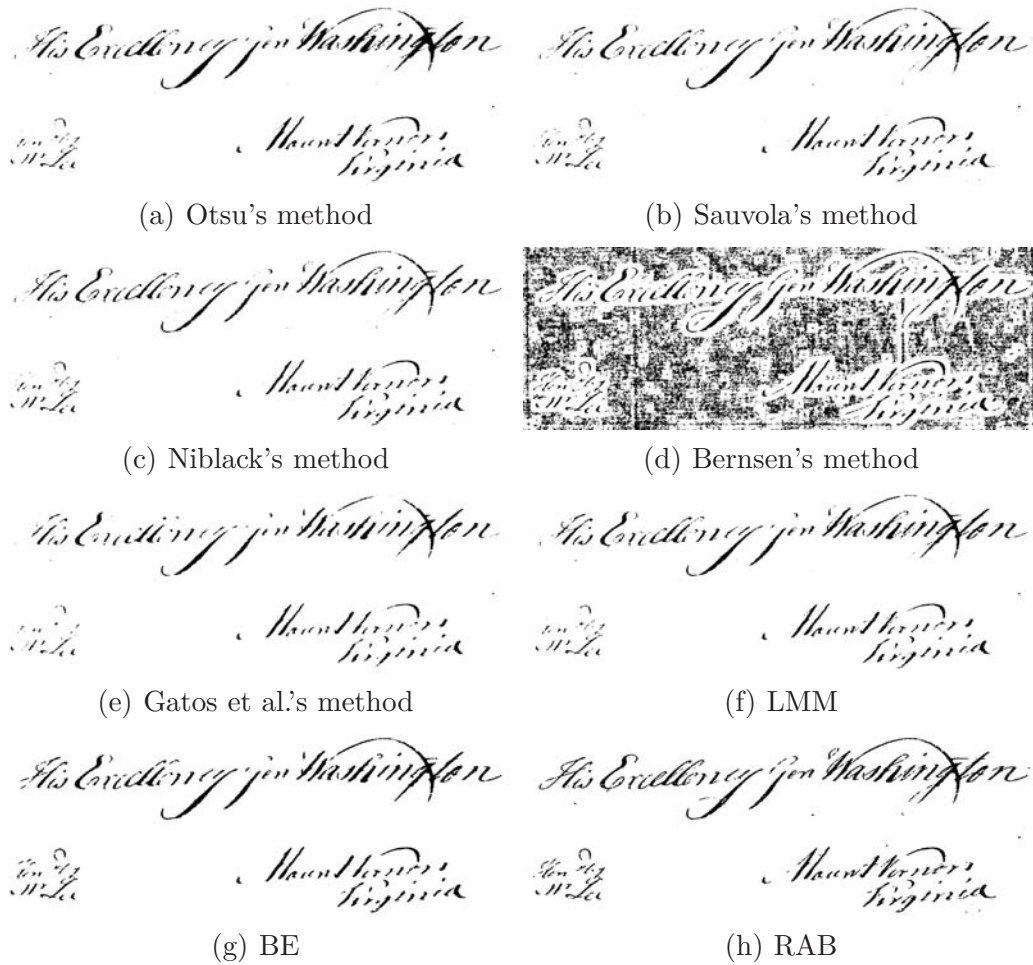


Figure 6.5: Binarization Results of the sample document image in Figure 6.1(d) produced by different methods.

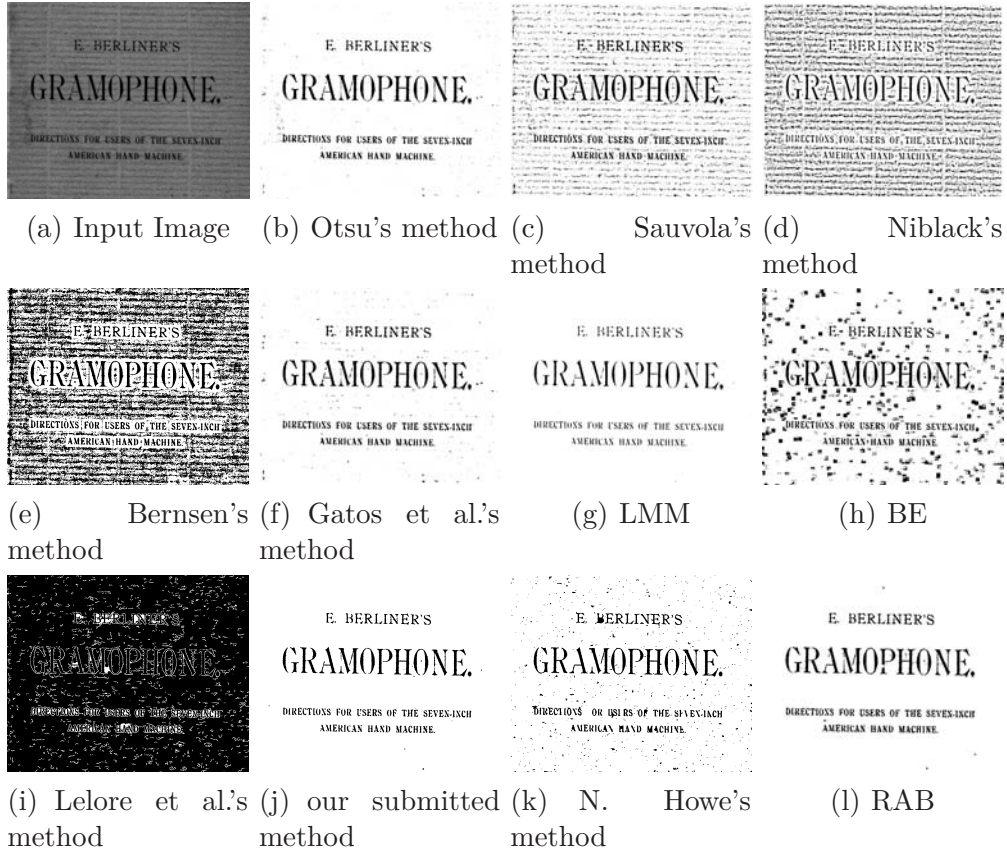


Figure 6.6: Binarization Results of the sample document image in DIBCO 2011 dataset produced by different methods.

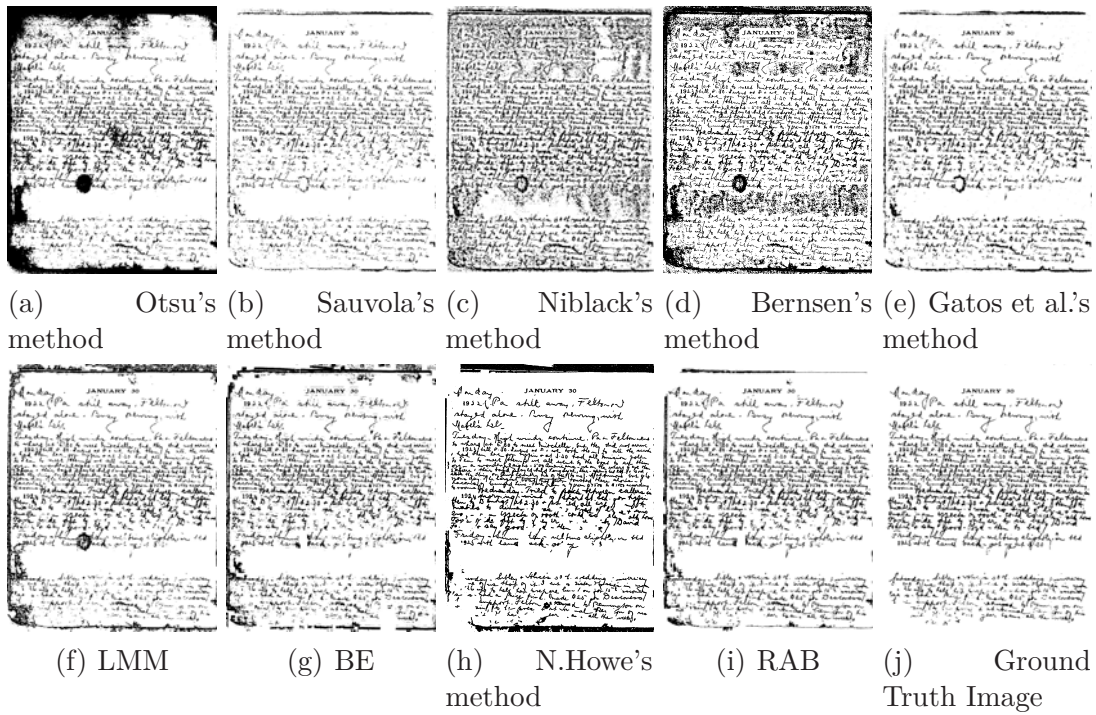


Figure 6.7: Binary results of the badly degraded document image from Bickley diary dataset shown in Figure 6.1(e) produced by different binarization methods and the ground truth image.

## Chapter 7

# Learning Frameworks For Document Image Binarization

This chapter presents a set of novel document image binarization frameworks that improve the performance of reported document image binarization methods. The proposed frameworks share the same idea that the correctly classified pixels by a given binarization method can be used to relabel those wrongly classified pixels. The document image pixels are first clustered into three categories instead of two based on given binarization methods, namely foreground pixels, background pixels and uncertain pixels. The features of the foreground and background pixels are extracted to classify those uncertain pixels. These proposed frameworks treat the binarization as a learning problem and has been tested on the dataset that is used in the recent DIBCO contest series.



---

## 7.1 A Learning Framework using K-means Algorithm

This section presents a learning framework that improves the performance of reported document image binarization methods using K-means algorithm. The proposed framework divides the image pixels into three categories based on given binarization method. It first clusters the foreground and background pixels into different classes using k-means algorithm, and assigns labels to these clusters as foreground or background. Then the uncertain pixels are classified with the same label of its nearest cluster. This framework treats the binarization as a learning problem and has been tested on the dataset that is used in the recent DIBCO contest series [1].

### 7.1.1 Uncertain Pixel Detection

When a threshold surface is generated by a given document binarization method, a pixel which is far away from the threshold surface has a high probability to be correctly classified, and most of the misclassified pixels are near the threshold surface. The distance from a pixel to the threshold surface is defined as the intensity difference between the pixel and its local threshold. So we divide the document image pixels into three categories as follows:

$$P(x) = \begin{cases} foreground, & T(x) - I(x) > T1 \\ background, & I(x) - T(x) > T2 \\ uncertain, & otherwise \end{cases} , \quad (7.1)$$

---

where the term  $P(x)$  refers to one pixel in the document images, the term  $I(x)$  refers to its intensity, the term  $T(x)$  refers to its local threshold, the terms  $T1, T2$  are two predefined thresholds. If the distance of a pixel to the threshold surface is larger than a threshold, the pixel is viewed as correctly classified. The other pixels are viewed as uncertain pixels. And we need to build a classifier from these fixed pixels to help us correctly classify those uncertain pixels, which will be explained in the later subsection.

The historical binarization records of the given document binarization method can be used to select suitable values for  $T1, T2$ . We can first estimate the binarization recall and precision of a given method from the training images. And the precision allows us to find the information of the number of background pixels which are classified as foreground. The recall can be used to estimate the number of foreground pixels which are classified as background. So the misclassified rates of background and foreground pixels are estimated as follows:

$$\begin{aligned}
 R_{fg} &= 1 - prec \\
 R_{bg} &= \frac{(1 - rec) \times NUM_{fg}}{NUM_{bg}},
 \end{aligned}
 \tag{7.2}$$

where the terms  $R_{fg}, R_{bg}$  denote the misclassified rates of foreground and background, respectively. The terms  $NUM_{fg}, NUM_{bg}$  denote the total numbers of pixels of foreground and background, respectively. The terms  $prec, rec$  denote the precision and recall of given binarization method, respectively.

Since most of the misclassified pixels are near the threshold surface, we should select thresholds  $T1, T2$  to make the number of uncertain background and fore-



Table 7.1: Evaluation results of Sauvola’s, Niblack’s, Otsu’s methods and proposed framework

Method	F-Measure(%)	PSNR	NRM( $\times 10^{-2}$ )	MPM( $\times 10^{-3}$ )
Sauvola	80.44	16.98	7.84	3.4
Improved Sauvola	89.12	17.59	3.95	1.0
Niblack	55.82	9.89	16.40	61.5
Improved Niblack	80.97	15.32	9.11	3.75
Otsu	78.72	15.34	5.77	13.3
Improved Otsu	83.88	17.91	3.92	1.65

ground pixels similar to the number of misclassified background or foreground pixels. First, we build two histograms for the distances of foreground and background pixels classified by a given method. Then the thresholds are selected separately for the background and foreground, to ensure that the number of uncertain background and foreground pixels satisfy the misclassified rates of background and foreground, which is shown as follows:

$$\begin{aligned}
 T1 &= \left\{ \min(t), \left| \frac{\sum_{0 \leq i \leq t} hist_{fg}(i)}{\sum_{0 \leq j \leq 255} hist_{fg}(j)} \geq R_{fg} \right\} \\
 T2 &= \left\{ \min(t), \left| \frac{\sum_{0 \leq i \leq t} hist_{bg}(i)}{\sum_{0 \leq j \leq 255} hist_{bg}(j)} \geq R_{bg} \right\} \right.
 \end{aligned} \tag{7.3}$$

where the terms  $T1, T2$  refer to the thresholds in Equation 7.1, the terms  $R_{fg}, R_{bg}$  refer to the misclassified rates in Equation 7.2, the terms  $hist_{fg}, hist_{bg}$  refer to the distance histograms built for foreground and background pixels, respectively. The term  $\min(t)$  denotes the smallest distance in distance histogram that makes the fraction of uncertain pixel number and total pixel number bigger than misclassified rate.

---

### 7.1.2 Uncertain Pixel Classification

After the fixed pixels are ready, we project these pixels into a feature space, and cluster them into different classes. There are two distinct differences between background pixels and text pixels. One is pixel intensity as the text pixels are usually darker than background pixels; the other is local image contrast as the text edge pixels usually have larger contrast than other pixels. So we use pixel intensity and local contrast as two features for each pixel. The local contrast is defined as: [3]

$$D(x, y) = \frac{f_{max}(x, y) - f_{min}(x, y)}{f_{max}(x, y) + f_{min}(x, y)}, \quad (7.4)$$

where  $f_{max}(x, y)$  and  $f_{min}(x, y)$  refer to the maximum and the minimum image intensities within a local neighborhood window. In the implemented system, the local neighborhood window is a  $3 \times 3$  square window. The two features need to be normalized into  $[0, 1]$ .

Because either the background or foreground pixels are usually non-uniform as shown in the figure, we need to cluster them into several classes. We use k-means algorithm [38] for background and foreground pixels separately. And each cluster is labeled background or foreground. The cluster number can be set 2 or 3, to satisfy the variations of most of document images.

The foreground and background clusters represent the global characteristic of text pixels and background pixels. These global foreground and background information are then used to build a classifier to classify the uncertain pixels. Each uncertain pixel will be labeled as either background or foreground based on its nearest cluster. The distance between a uncertain pixel and a cluster is

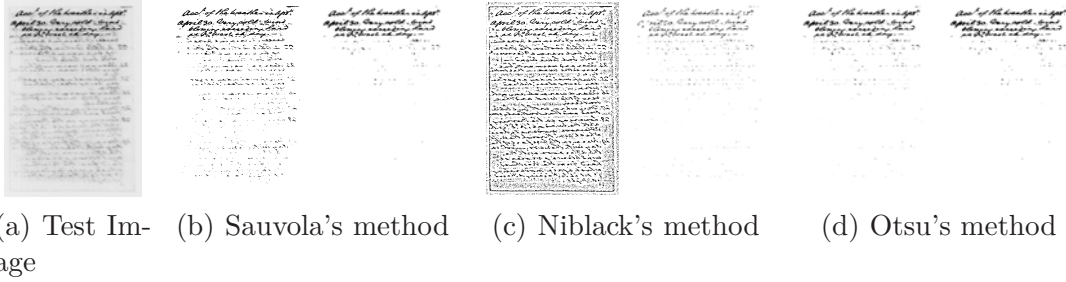


Figure 7.1: Binarization results of the document image in Figure 7.1(a). The left images in Figure 7.1(b-d) are produced by testing methods, the right images are produced by proposed framework.

defined as the Euclidean distance of the uncertain pixel and the center of that cluster in the feature space. The classifier is shown as follows:

$$P = \begin{cases} foreground, & C_{min} \in foreground \\ background, & C_{min} \in background \end{cases}, \quad (7.5)$$

$$C_{min} = \arg \min Dis(C_i, P)$$

where the term  $P$  denotes one uncertain pixel, the term  $C_i$  denotes the center of a cluster, the term  $Dis$  calculates the Euclidean distance between the pixel and the cluster center in the feature space, and the term  $C_{min}$  denotes the nearest cluster of  $P$ .

### 7.1.3 Experiments

The proposed framework has been tested over the dataset that is used in the recent Document Image Binarization Contest (DIBCO), 2009. We apply our framework to Sauvola's, Niblack's and Otsu's methods [12, 13, 14]. As shown in

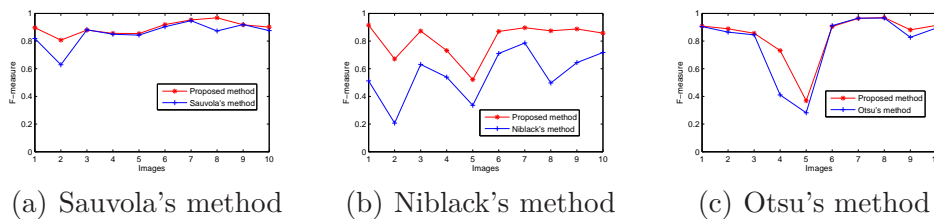


Figure 7.2: F-measure values of ten different document images in DIBCO 2009 dataset.

Figure 7.1, most of the noises generated by testing results are removed by our proposed framework.

We also apply four widely used evaluation measures that adapted from DIBCO report [1] to the testing methods and proposed framework. The evaluation results are shown in Table 7.1. And Figure 7.2 compares the F-measure values of testing methods and improved results by our proposed framework for ten different images in DIBCO 2009 dataset. Our proposed framework significantly improves the performance of these testing methods in term of these four measures.

## 7.2 Combination of Document Image Binarization Techniques

This section presents a novel document image binarization combination framework that improves the performance of reported document image binarization methods. The proposed framework divides the image pixels into three categories based on the binary results of given document binarization methods. All the pixels are then projected into a feature space. The pixels in foreground and background sets can be viewed as correctly labeled samples, and used to determine the

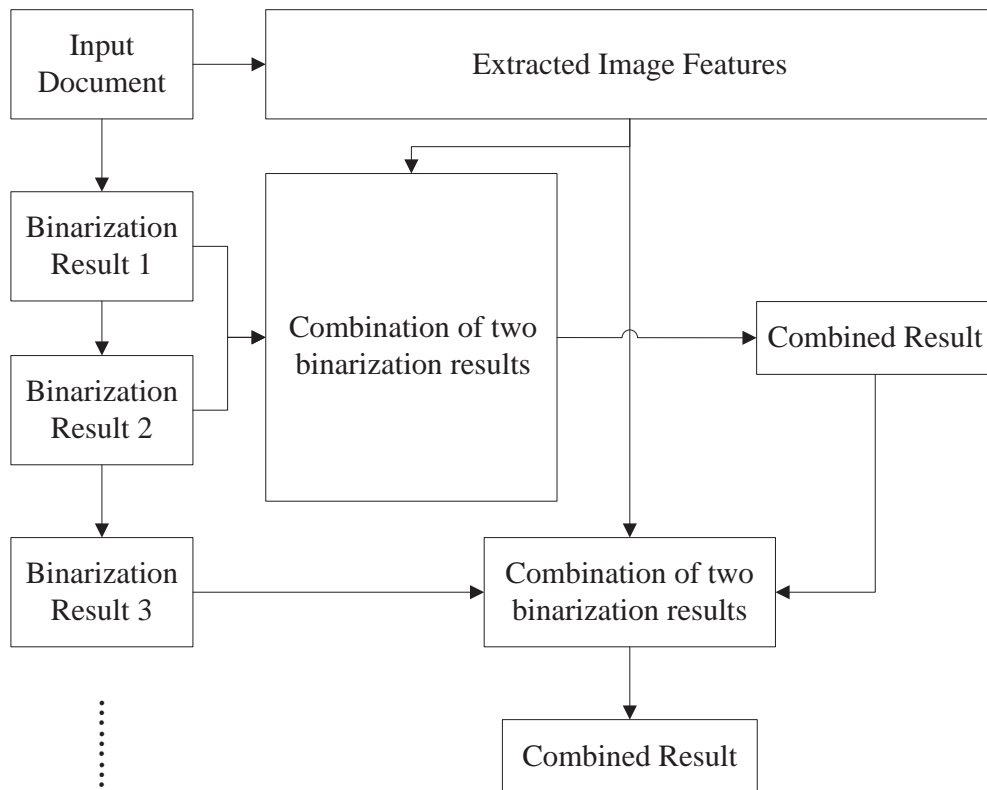


Figure 7.3: The overall flowchart of our proposed document binarization combination framework.

---

label of those uncertain pixels. A classifier is then applied to iteratively classify those uncertain pixels into foreground and background. The overall flowchart of our proposed document binarization combination framework is illustrated in Figure 7.3. Experiments over the dataset of recent DIBCO 2009 [1] and H-DIBCO 2010 [31] demonstrate superior performance of our proposed framework. Experimental results show that the proposed framework can improve the reported binarization methods significantly.

### 7.2.1 Feature Extraction

Feature extraction is one of the most important steps in classification. Projecting the image pixels into an appropriate feature space makes the foreground text and document background easier to separate. For document image binarization, the two most frequently used features are intensity and contrast. In our proposed framework, we define the contrast presentation as follows:

$$Con(x, y) = \frac{f_{max}(x, y) - I(x, y)}{f_{max}(x, y) + \epsilon}, \quad (7.6)$$

where the term  $I(x, y)$  denotes the intensity of pixel  $(x, y)$ . The term  $f_{max}(x, y)$  refers to the maximum image intensities within a local neighborhood window. The term  $\epsilon$  is a positive but small number, which is added in case the local maximum is equal to 0. Furthermore, the term  $Con(x, y)$  denotes the contrast value of the estimating pixel  $(x, y)$ . In our implementation, the local neighborhood window is set to  $10 \times 10$ . The contrast defined in Equation 7.6 preserves the ability to suppress the background variation while assigns a more accurate contrast value to document pixels.

---

Another feature used in our framework is the pixel intensity  $I$  of the document image. Therefore, the pixel  $(x, y)$  is projected to a 2D feature space  $[Con, I]$ , where the term  $Con$  denotes the contrast feature, and the variable  $I$  denotes the intensity feature.

## 7.2.2 Combination of Binarization Results

After the image pixels are projected into a feature space, we need to use the pixels in the foreground/background set to determine the categories of the uncertain pixels. It is not suitable to compare the examining uncertain pixel with the whole foreground/background set, due to the high variation within both the foreground and background of the degraded document image. So the uncertain pixel is examined under a local neighborhood window, the pixel is set to background or foreground depending on its distance to local background pixels and foreground pixels, which is defined as follows:

$$P(x) = \begin{cases} foreground, & \frac{Con(x)}{Con_F} > \frac{Con_B}{Con(x)} \parallel \frac{I_F}{I(x)} > \frac{I(x)}{I_B} \\ background, & otherwise \end{cases}, \quad (7.7)$$

where the term  $P(x)$  denotes one uncertain pixel, the terms  $Con(x), I(x)$  denote the corresponding contrast and intensity features, respectively. The two terms  $Con_F, I_F$  refer to the mean contrast and intensity feature values of foreground pixels within a local neighborhood window, respectively. Furthermore, the two terms  $Con_B, I_B$  refer to the mean contrast and intensity feature values of background pixels within a local neighborhood window, respectively. Since the terms  $Con_F > Con_B$  and  $I_B > I_F$ ,  $\frac{Con(x)}{Con_F} > \frac{Con_B}{Con(x)}$  and  $\frac{I_F}{I(x)} > \frac{I(x)}{I_B}$  mean that distance between local contrast mean value and local intensity mean value of foreground

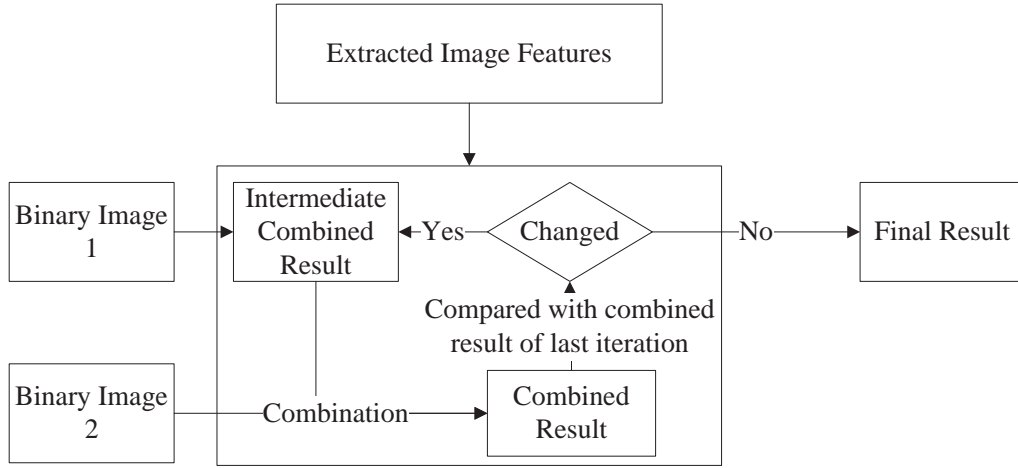


Figure 7.4: The flowchart of combination of two binarization results.

Table 7.2: Evaluation Results of the dataset of DIBCO 2009

Method	F-Measure(%)	PSNR	NRM( $\times 10^{-2}$ )	MPM( $\times 10^{-3}$ )
1.Otsu	78.72	15.34	5.77	13.3
2.Sauvola	85.41	16.39	6.94	3.2
3.Combined of 1 and 2	86.62	16.76	3.99	4.1
4.Gatos	85.25	16.50	10	0.7
5.LMM	91.06	18.50	7	0.3
6.Combined of 4 and 5	91.86	18.72	3.97	0.4
7.BE	91.24	18.66	4.31	0.55
8.Combined of 5 and 7	93.18	19.60	3.34	0.31

and the examining uncertain pixel is smaller than that of background and the examining uncertain pixel, respectively.

The local neighborhood window is set to  $3 \times 3$  in our implementation. There may be no foreground and background pixels within a neighbor window of an uncertain pixel. So we use an iterative strategy to update the foreground/background sets. Only those uncertain pixels that have foreground or background pixels within its neighbor window will be classified into foreground or background in each iteration. The procedure repeats until all the uncertain pixels are classified,



---

which is shown in Figure 7.4. The input is two binarization resultant images, one is selected as the initial combined result, then the document pixels are divided into three categories using the intermediate combined result and the other binary image. Then some of the uncertain pixels are classified to form the new intermediate combined result, this procedure repeats until the combined result doesn't change, then the final result are produced. It usually takes around 10 iterations to coverage according to experiments.

### 7.2.3 Experiments

The proposed method has been tested over the dataset recent DIBCO 2009 [1] and H-DIBCO 2010 [31]. The DIBCO 2009 dataset and H-DIBCO 2010 dataset contain 20 historical document images suffering from different kinds of degradations in total. We apply our framework to different well-known document binarization methods, including Otsu's, Sauvola's, Gatos's methods [12, 14, 24] and our previous LMM and BE methods discussed in Chapter 3 and 4. And the four evaluation measures (F-Measure, PSNR, NRM, MPM) adapted from DIBCO's report [1] are used to compare the performance of the testing methods and proposed framework.

The evaluation results are shown in Table 7.2. As shown in Table 7.2, our proposed framework can produce better results than other methods by combining our previous LMM method and BE method. And the combined results can perform better in terms of F-Measure, PSNR, NRM than the two origin methods separately. This means a higher precision and better text stroke contour can be obtained after combination. Figure 7.5 shows two binarization results produced

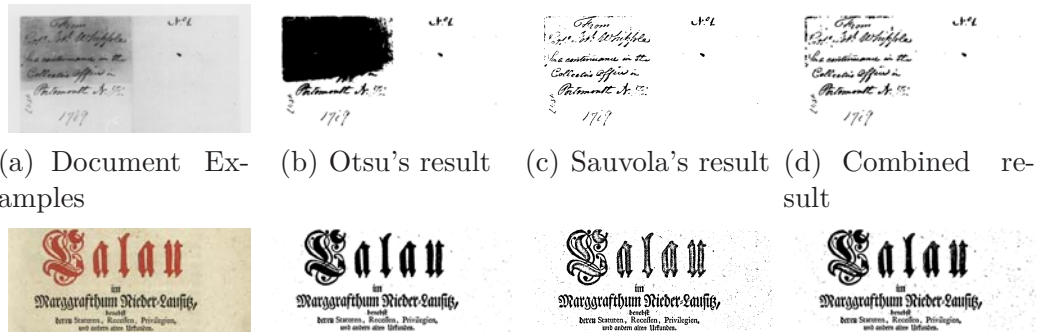


Figure 7.5: Two degraded document image examples and corresponding binarization results produced by Otsu’s method, Sauvola’s method and our proposed combination framework, respectively.

by combining Otsu’s [12] and Sauvola’s [14] method.

### 7.3 A Learning Framework using Markov Random Field

With the binarization results of existing methods, the proposed framework divides the document image pixels into three categories, namely, document background pixels, document foreground (text) pixels, and uncertain pixels. First, a threshold surface is obtained by an existing binarization method for the input degraded document. Document image pixels are then divided into three categories based on their distances to the threshold surface. Those pixels with large distance to threshold surface are classified into either background or foreground categories because they usually have a higher classification rate. The rest is classified as uncertain pixels, which are finally classified by the Markov Random Field (MRF) model by incorporating the edge information.

---

### 7.3.1 Uncertain Pixels Detection

To determine the uncertain pixels, we need to create the threshold surface from the given binarization method. Generally speaking, the binarization methods classify the pixels by comparing its feature vector  $\mathbf{f}$  with the corresponding threshold vector  $\mathbf{t}$  using some distance metrics. Those pixels with larger distances should have higher possibility to be correctly classified. On the other hand, those pixels that are close to the threshold surface might need to re-classify, and be viewed as uncertain pixels. This is defined as follows:

$$P(x) = \begin{cases} \text{foreground/background,} & \text{dis}(f_x, t_x) > T \\ \text{uncertain,} & \text{otherwise} \end{cases}, \quad (7.8)$$

where the term  $P(x)$  refers to one pixel in the document images, the term  $f_x$  refers to its feature vector, the other term  $t_x$  refers to its corresponding threshold vector, the term  $dis()$  denotes the distance metric that defined by the given binarization method, and the variable  $T$  is a predefined threshold. If the distance between the feature vector and threshold vector is larger than a threshold, the pixel is viewed as correctly classified. The others are viewed as uncertain pixels. The threshold  $T$  is set to determine the number of pixels will be labeled as the uncertain pixels, which is related to the number of pixels that might be misclassified by the given binarization methods.

### 7.3.2 Edge Pixels Detection

Canny edge detector [2] is used to extract the text stroke edge information of the document image. However, Canny's detector by itself often extracts a large

---

amount of non-stroke edges, since the high and low thresholds of Canny’s method is not easy to selected. We then incorporate the edge pixels detected by Canny’s method with the high contrast pixels extracted by our previous local maximum and minimum method in Chapter 3. The image contrast is evaluated by the equation 3.1, which suppresses the intensity variation of the degraded document images, and captures the high contrast pixels that are usually close to the text stroke edge more precise. In the combined map, we keep only pixels that appear within both the high contrast image pixel map and canny edge map.

### 7.3.3 Uncertain Pixels Classification

Since an image can be represented as a graph  $G = V, E$ , where the term  $V$  is the set of all pixels  $p$  and  $E$  is the set of all adjacent relationships between a pixel and its neighborhood pixels, the image binarization problem can be treated as assigning a label  $l$  (either foreground or background) to each image pixels and interpreted using Markov Random Field model. The solution is to find out a label set  $L$  for each image pixels in  $V$  by minimizing the following energy function [39]:

$$E(L) = \sum_{p \in V} E_1(l_p) + \lambda \sum_{(p,q) \in E} E_2(l_p, l_q), \quad (7.9)$$

where the term  $E_1$  refers to the likelihood energy, which denotes the cost when pixel  $p$  is assigned to label  $l$ , and the term  $E_2(x, y)$  refers to the prior energy that represents the cost when the adjacent pixels  $p, q$  are assigned to label  $l_p, l_q$ , respectively.

The two energy terms  $E_1$  and  $E_2$  are defined based on the preliminary binarization results and detected edge images.

---

The likelihood energy  $E_1(l_p)$  reflects how likely the image pixel  $p$  belongs to label  $l_p$  (foreground or background). There should be large costs associated to assign a different label to those pixels already in the foreground/background sets, which are determined in previous steps as illustrated in previous sections. In addition, the cost that assigns foreground label to those pixels that are far away from the detected edges should be large, because the text pixels should be close to the text stroke edge. Moreover, those uncertain pixels should be assigned to a label with similar intensities locally.

The prior energy  $E_2$  reflects the smoothness cost between the adjacent pixels. The smoothness cost penalizes those adjacent pixels with different labels in non-edge zones. On the other hand, those adjacent pixels with the same label at edge should be penalized too.

The  $E_1(l_p)$  and  $E_2(l_p, l_q)$  is defined as follows:

$$\left\{ \begin{array}{ll} E_1(l_p) = l_p \cdot \infty & \forall p | EN_p == 0 \\ E_1(l_p) = l_p & \forall p \in \text{foreground} \\ E_1(l_p) = 1 - l_p & \forall p \in \text{background} \\ E_1(l_p) = \frac{dis_p^B \cdot l_p + dis_p^F \cdot (1 - l_p)}{dis_p^F + dis_p^B} & \forall p \in \text{Uncertain} \end{array} \right. , \quad (7.10)$$

$$\left\{ \begin{array}{ll} E_2(l_p, l_q) = \frac{1 - |l_p - l_q|}{dis(p, q) + 1} & E_p \cdot E_q == 1 \\ E_2(l_p, l_q) = \frac{|l_p - l_q|}{dis(p, q) + 1} & \text{otherwise} \end{array} \right. , \quad (7.11)$$

where the term  $l_p$  is either 0 (foreground) or 1 (background), the term  $EN_p$  denotes the number of edge pixels within a local neighborhood window of  $p$ , the terms  $dis_p^F, dis_p^B$  denotes the distance between the intensity of  $p$  and the mean intensity of the foreground pixels and background pixels within a local neigh-

---

Table 7.3: F-Measure evaluation of our proposed framework

F-Measure(%)	DIBCO09	HDIBCO10	DIBCO11	Average
Sauvola	85.41	75.3	82.54	81.08
Improved Sauvola	89.88	86	85.64	87.17
Niblack	55.82	74.1	68.52	66.15
Improved Niblack	82.21	80.27	76.36	79.62
Otsu	78.72	85.27	82.28	82.09
Improved Otsu	81.98	87.13	83.55	84.22

neighborhood window, respectively. The term  $dis(p, q)$  denotes the intensity difference between  $p$  and  $q$ , the term  $E_p$  denotes whether  $p$  belong to edge (1) or not (0). The last three conditions in Equation 7.10 imply that  $EN_p$  does not equal 0. The final binarization results is obtained by solving the energy function in Equation 7.9 using graph cuts methods [40].

### 7.3.4 Experiments

We test our proposed framework on the datasets of DIBCO09, H-DIBCO10 and DIBCO11 [1, 30, 31] using the established binarization methods, including Otsu’s method [12], Niblack’s method [13] and Sauvola’s method [14].

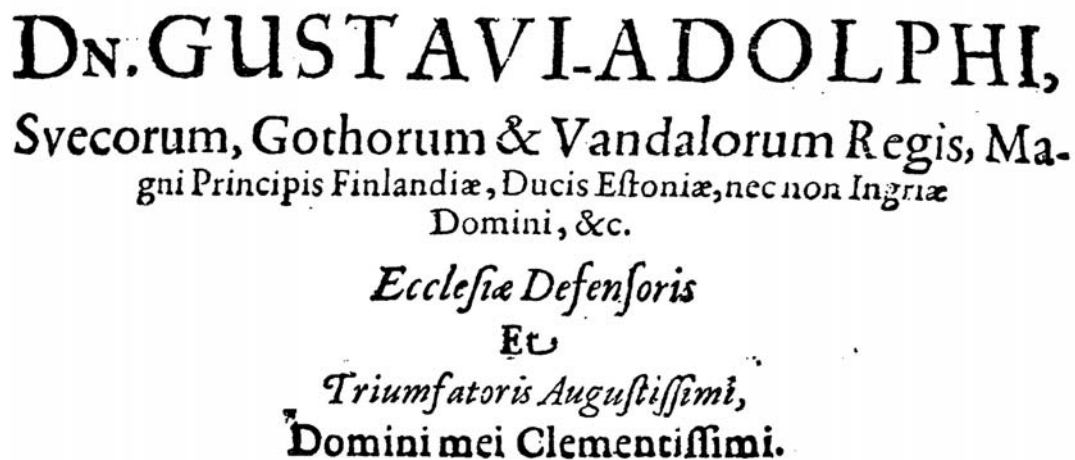
We quantitatively analyze the performance of our proposed framework on the three datasets using F-Measure that was adapted from the recent DIBCO 09 [1]. The evaluation results are shown in Table 7.3. Figure 7.6 shows a binarization example with and without our proposed Markov Random Field framework. Our proposed framework significantly improves the performance of these testing methods.



(a) One degraded document image example



(b) Binarization Using Sauvola's method



(c) Binarization using Sauvola's method with our framework

Figure 7.6: Binarization results with/without our MRF framework

# Chapter 8

## Enhancement of Web Images for Text Recognition

### 8.1 Introduction

Images on the internet are increasing tremendously during these years. Many of these images contain text information that cannot be found in other places of the web pages [41, 42]. The recognition of the textual information within web images is very helpful for a better understanding of the contents of web pages. As these images with text embedded are used in the internet for different purposes, text recognition in web images can be applied on different kinds of applications, such as web page indexing and retrieval, web page content filtering [43]. It will become even more important as the textual information within web images is contributing more and more due to the future network development.





Figure 8.1: Some low quality web image examples

## 8.2 Literature Review

Many techniques have been proposed for text extraction and recognition on videos and natural scene images [44, 45, 46], but much fewer efforts have been reported for the recognition of the text within web images [43, 47, 48, 49]. Karatzas and Antonacopoulos [49] tried to identify the character regions by layer decomposition based on approximate human color perception. Lopresti and Zhou [48] make use of the polynomial surface fitting and fuzzy tuple classifiers to recognition the low resolution text in web images. And some techniques [50, 51, 52] are also proposed for image based Spam email filtering.

---

Compared with other images, web images are often more susceptible to certain specific image degradations including low resolution and small size for faster network transmission rate, computer-generated-character artifacts, and special effects on images for attractiveness purpose. As a result, the techniques developed for video/natural scene images often fail to produce satisfactory results when they are directly applied for web images. The latest Robust Reading Competition in Born-Digital Images (Web and Email) held under the framework of International Conference on Document Analysis and Recognition (ICDAR) 2011 [43] shows current research progress on this area. The contest consists of three tasks, i.e. text localization, text segmentation and word recognition in web image. The third recognition task aims to convert the textual information from bitmap format to ASCII format where the text regions are assumed to have been cropped within web images by certain text localization algorithms. The baseline word recognition method provided by competition organizers achieved around 63% recognition rate on the test dataset. The low recognition rate can largely be attributed to several types of document degradation including low resolution, complex background, low contrast, and non-uniform color as illustrated in Figure 8.1.

### 8.3 Text Recognition on Web Images

The proposed technique only focuses on web images that are composed mainly of text regions. Furthermore, the proposed technique assumes that the text has been located by text localization techniques, and aims to improve the quality of the text regions to guarantee better OCR results. We choose the *RGB* color model in our study.

---

First, the input web images are resized and applied with contrast stretching in the preprocessing step. Second, L0 norm smoothing [53] is used to suppress variation within background and text and enhance the resolution. Third, the smoothed images are binarized on each color channel separately. Fourth, the images are divided into eight fragments, each of which contains pixels with the same intensity. Connected component analysis is then applied on each fragment to filter out non-character regions. Finally the images are recognized by OCR engines after skew correction.

### 8.3.1 Pre-Processing

The input web images need to be resized before applying the smoothing step so that the sizes of characters are large enough for processing. We simply use the bicubic interpolation [54] to resize the input image from  $(h, w)$  to  $(H, W)$ , where the terms  $(h, w)$  and  $(H, W)$  denote the height and width of the original image and enlarged image, respectively. The term  $H$  is a user defined fixed height to control character size of the input web image, and  $W$  is proportional to  $H$ , and calculated by  $\frac{w}{h} \times H$ . The  $H$  is set to 300 in our experiments.

The intensity differences of some web images between text and background are small, the contrast of the input web images need to be stretched for better results. The intensity ranges of the input images are rescaled to  $[0, 255]$  using Equation 8.1 at each channel to obtain a significant edge contrast at the text boundaries after smoothing.

$$I'_p = 255 \times \frac{I_p - \text{Min}(I)}{\text{Max}(I) - \text{Min}(I)}, \quad (8.1)$$

---

where the variable  $I$  denotes the original image intensity, the variable  $I'$  denotes the rescaled image intensity, the variable  $p$  refers to the pixel index, the terms  $Max(I)$ ,  $Min(I)$  denote the maximum and minimum intensities of image  $I$ , respectively.

### 8.3.2 Image Smoothing and Binarization

To guarantee a good recognition result, the text and background of the web images should have uniform color. The text recognition task would become much easier if the pixel intensities within the same category (background or text) are equal, intensity differences only occur at the boundary between text and background. Take a line of image pixels as an example. The pixel intensities are desired to be smoothed as the red line as shown in Figure 8.2. The smoothed image line should have as less intensity levels as possible subject to difference between the smoothed line and the original line. In ideal cases, each smoothed line has only one or two intensity levels that denote the background or text. There are usually low resolution and multi-color text in web images. Thus it is not effective to do binarization directly on the web images, and we can express the problem by the following objective function instead:

$$\min_S \left( \sum_i (S_i - I_i)^2 + \lambda L(S) \right), \quad (8.2)$$

where the term  $I$  denotes the intensities of an original image line, the term  $S$  denotes the intensities of the smoothed image line, the term  $i$  denotes the pixel index, the term  $L(S)$  denotes the number of image intensity levels of  $S$ . The term  $\lambda$  is the weight factor that controls the smoothing degree. By minimize

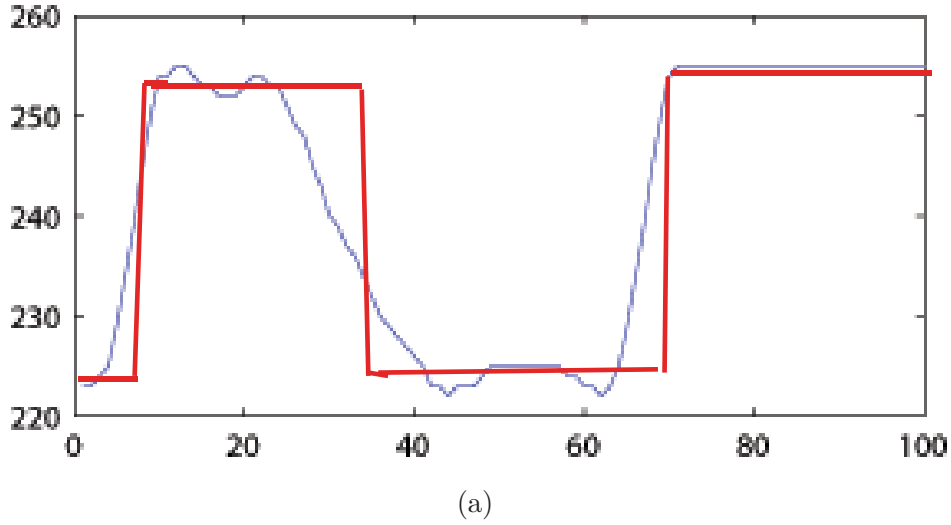


Figure 8.2: A column of image pixels taken from Figure 8.1(f) which is shown in blue. The vertical index denotes the pixel intensity, the horizontal index denotes the image pixel index. The smoothed line is represented in red.

Equation 8.2, the smoothed image line  $S$  is obtained.

However, it is difficult to solve the objective function given by Equation 8.2 because the second term  $L(S)$  is not functional and involves global statistical information. The gradient descent and other discrete optimization methods cannot be applied to this objective function directly.

The pixels in the same intensity level should locate in near regions since the testing web images are composed of background regions and text regions. Then the number of intensity levels  $L(S)$  should be close to the number of intensity change. To simplify this objective function, We replace Equation 8.2 by a new objective function as shown in Equation 8.3.

$$\min_S \left( \sum_i (S_i - I_i)^2 + \lambda N(S' \neq 0) \right), \quad (8.3)$$

---

where the term  $N(S' \neq 0)$  denotes the number of non zero gradient of  $S$ . The term  $S'$  refers to the gradient of  $S$ , which is computed by  $S_i - S_{i-1}$ . This function is equivalent to the L0-norm Smooth proposed by Xu et al. [53], and a close-form solution can be found by half quadratic splitting scheme [55] as described in Xu et al.'s paper [53]. We can only find an approximation solution since the L0-norm regularized optimization problem is computationally intractable.

So an auxiliary vector  $g$  with the same size of  $S$  is introduced to rewrite the objective function of Equation 8.3 as follows:

$$\min_{S,g} (\sum_i ((S_i - I_i)^2 + \beta(S'_i - g_i)^2) + \lambda N(g \neq 0)), \quad (8.4)$$

where the term  $N(g \neq 0)$  denotes the number of non-zero items in  $g$ . the term  $\beta$  is the parameter that controls the similarity between  $g$  and  $S'$ . Equation 8.4 is equivalent to Equation 8.3 when  $\beta$  is arbitrarily large.

We can solve Equation 8.4 by minimizing  $g$  and  $S$  iteratively. At every iteration, one variable is minimized given the value of the other variable obtained in last iteration.

### Minimizing S

We can omit the term  $\lambda N(g_i \neq 0)$  that does not involve  $S$  and obtain the following equation for solving  $S$ .

$$\min_S (\sum_i ((S_i - I_i)^2 + \beta(S'_i - g_i)^2)), \quad (8.5)$$

This function is quadratic and can be solved easily by gradient descent. We can also directly yield the solution by derivative operator. Since  $S'_i$  is expressed by  $S_i - S_{i-1}$ , we calculate the partial derivative of the objective function for each  $S_i$

---

and let it be 0, then the value of  $S_i$  is obtained as follows:

$$S_i = \frac{\beta}{\beta + 1} S_{i-1} + \frac{\beta g_i + I_i}{\beta + 1}, \quad (8.6)$$

where the term  $S_0$  equals to  $\frac{\beta g_i + I_i}{\beta + 1}$  if we let  $S_{-1}$  equal to 0.

### Minimizing $g$

The term  $(S_i - I_i)^2$  can be omitted since it is not related to  $g$ . It is noticed that  $N(g \neq 0)$  can be expressed by a summation of Boolean functions, which is defined as follows:

$$B(g_i) = \begin{cases} 1 & g_i \neq 0 \\ 0 & g_i = 0 \end{cases}, \quad (8.7)$$

where the term  $g_i$  denotes the  $i_{th}$  item of  $g$ , the boolean function  $B(g_i)$  returns 1 only if  $g_i \neq 0$ , and 0 when  $g_i = 0$ . Then the  $\sum$  operator can be further moved outside of the brackets, and the objective function is decomposed at the pixel level, which is shown in Equation 8.8

$$\sum_i \min_g (\beta(S'_i - g_i)^2 + \lambda B(g_i)) \quad (8.8)$$

This function can be solved empirically, it is easy to prove that the solution of  $g_i$  is given by Equation 8.9 [53].

$$g_i = \begin{cases} S'_i & \text{if } S_i'^2 > \frac{\lambda}{\beta} \\ 0 & \text{otherwise} \end{cases} \quad (8.9)$$



Figure 8.3: Smoothed images of the original images in Figure 8.1

The value of  $\frac{\lambda}{\beta}$  is related to web image contrast, and controls the smooth degree. In each iteration, the smoothing procedure will set  $g_i$  to 0 if its corresponding gradient magnitude  $S_i'^2$  is smaller than  $\frac{\lambda}{\beta}$ . The larger the value of  $\frac{\lambda}{\beta}$ , the smoother the resultant  $S$ . And  $\frac{\lambda}{\beta}$  should not be larger than the maximum gradient of  $I$ , otherwise it may end up with a complete flat image.

Because the gradient between text and background is more important to be evaluated locally [5] and the intensity variation of web images is smaller than that of natural images. If we examine the gradient globally and use the 2-D smoothing manner, some text regions with small intensity differences to the background might be filtered out after 2-D smoothing globally. It is better to apply the



---

smoothing locally on each image line to retain the text information. In the proposed technique, we applied the smoothing procedure two rounds for each web image. The web image is smoothed first row by row, and then column by column at the second round.

After the web images are smoothed, they can be easily binarized on each color channel separately by binarization techniques. Figure 8.3 shows the results of the images in Figure 8.1 after smoothing and binarization. The contrast of those images in Figure 8.1 are enhanced, and the complex background are removed as shown in (c) and (e). The web images are then decomposed into several fragments based on color levels and classified in the next section.

### 8.3.3 Detection of Character Components

We make use of the shape characteristics of text components to segment text regions from the smoothed images as shown in Figure 8.1. Since the images are binarized in each channel, the value of each image pixel on each channel is either 0 or 1. There are eight different color levels at most, the color levels vary from  $[0, 0, 0]$  to  $[1, 1, 1]$ . The smoothed images are then segmented into eight fragments, where each fragment contains only image pixels with the same RGB color values. Usually there are less than eight fragments in a web image after segmentation, because most of the web images may contain only little color levels.

Then the character component detection is applied on each fragment separately. Instead of examining the statistics information only for each connected component separately [46], we further consider the whole fragment information since the text pixels are usually at the same fragment. And a few features are

---

**lufthansa.com out**

(a)

(b)

**Service society**

(c)

(d)

**Point FULL**

(e)

(f)

Figure 8.4: Binary images of the original images in Figure 8.1

---

used for character region detection as described below.

**Proportion of the pixels in the testing fragment that are also in the boundary components.**

The boundary components refer to the components that touch the image boundary. The text embedded in the web images should locate at the central part of the web images surrounding by background regions. It is obvious that the fragment belongs to background if most of its components locate at the border of the images. If the proportion is larger than a threshold  $T_{border}$ , the whole fragment is considered as background regions, otherwise only the components that touch the border are removed.

**Proportion of the pixels in the filled area of all connected components that are also in those component regions.**

The filled area of a component refers to the area with all the holes filled of the connected component. Since the text strokes are usually strengthened for better visualization, most of those character components of web image fragment do not contain too large hole inside. If the proportion is smaller than a threshold  $T_{filled}$ , the whole fragment is considered as non text regions.

**The variation and mean of the heights of connected components within the same fragment.**

The web images are resized into the same height in the preprocessing step and the characters of the same web images have similar heights, so the characters should have similar sizes. If a fragment is classified as text, the variation of heights of those text components should be smaller than a threshold  $T_{var}$ , and average heights of text components should be larger than a fixed height  $T_{mean}$ .

The non-character regions of each image fragment are filtered out based on

---

Table 8.1: Evaluation of the recognition results on the Robust Reading Competition Dataset using Google Tesseract OCR

Methods	Edit Distance	Precision(%)
Baseline	484.5	34.64
Bilateral	468.4	36.06
Binarization	839.8	6.54
VideoText	269.3	55.45
Proposed	317.4	61

the detection criterion. The remaining character components are then combined together to generate the binary images contain only text. Figure 8.4 shows the corresponding binary images of those in Figure 8.1.

### 8.3.4 Skew Correction and Text Recognition

The orientation of the web image text may not be horizontal. The OCR engine may fail to recognize the skew text. The text characters are arranged one by one along the text line, the text pixels are spread out along that direction. So we can use principal component analysis (PCA) [56] to find out the highest variance direction, which should be also the orientation of the text. Figure 8.5 shows a skew correction example, where the text line direction is found by PCA correctly, and the rotated image is shown in (b).

After the skew correction, the binary web images are fed into the OCR engine to obtain the final text recognition result.

## 8.4 Experiments

Our proposed technique is tested using the recent Robust Reading Competition for Born-Digital Images dataset [43]. The testing dataset contains 918 web im-

Table 8.2: Evaluation of the recognition results on the Robust Reading Competition Dataset using Abbyy OCR

Methods	Edit Distance	Precision(%)
Baseline	232.8	63.51
Bilateral	228.6	68.63
Binarization	621.1	27.56
VideoText	272.8	61.33
Proposed	<b>190.1</b>	<b>72.33</b>

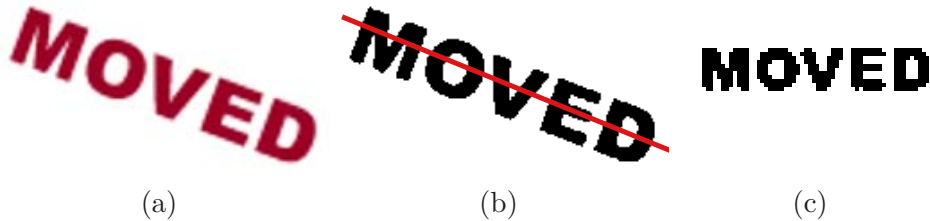


Figure 8.5: An example of skew correction. (a) shows the original web image, the binary image with a red line denotes the text orientation calculated using PCA is shown in (b). (c) shows the rotated result.

ages. Those images have already been extracted from the located text of origin images. Some of these images are easy to recognize, but quite a few of them are challenging. These images usually have various types of problems that decrease its visual quality significantly, such as low resolution, and text artifacts.

During the experiments, we set  $T_{border} = 0.8$ ,  $T_{filled} = 0.1$ , and  $T_{var} = \frac{H}{2}$ ,  $T_{mean} = \frac{H}{6}$ , where the term  $H$  is the resized height described in Pre-Processing section. Ground truth texts for the testing dataset are provided by the competition organizers, so we can use Levenshtein distance (edit distance) to compare the difference between the ground truth text and recognized text. The total edit distance for the whole dataset of one method is defined as follows:

Table 8.3: Recognition results of the web images in Figure 8.1 using Google Tesseract

Methods	(a)	(b)	(c)	(d)	(e)	(f)
GT	lufthansa.com	out	Service	society	Point	FULL
Baseline	lufthansa	nm	Is	(no text)	Lti??U	1
Bilateral	N \hnnsa.com	nm	,5Q . =	society	Pofni.	(no text)
Binarization	(no text)	(no text)	(no text)	(no text)	(no text)	(no text)
VideoText	luwfithansa com,	(no text)	Service	society	Pomt	FULL
Proposed	Iu\ a??thansa.oom	out	Service	society	Pomt	FULL

$$DIS = \sum_i \frac{dis_i}{len_i}, \quad (8.10)$$

where the term  $DIS$  denotes the total edit distance, the term  $dis_i$  denotes edit distance on image  $i$ , the term  $len_i$  denotes the number of characters in ground truth text of image  $i$ .

We also use recognized rate to evaluate the recognition performance on word level, because in many cases, the textual information is still lost or ambiguous with only part of the texts are recognized. Only the word is recognized completely the same as the ground truth text, the corresponding image will be labeled as successfully recognized. And the word recognition rate is defined as follows:

$$precision = \frac{N_{correct}}{N_{total}}, \quad (8.11)$$

where the term  $N_{correct}$  denotes the number of successfully recognized images, the term  $N_{total}$  denotes the total image number.

We compared our results with different methods, including baseline method, recognition after smoothing with Bilateral filters(Bilateral) [57], recognition after binarized with the well-known binarization method(Binarization) [14], and

---

Table 8.4: Recognition results of the web images in Figure 8.1 using Abbyy

Methods	(a)	(b)	(c)	(d)	(e)	(f)
GT	lufthansa.com	out	Service	society	Point	FULL
Baseline	lufthansa	rm	Is	(no text)	Lti??U	1
Bilateral	lufthansa	m	(no text)	society	Otilia	i
Binarization	lufthansa	(no text)	(no text)	(no text)	(no text)	(no text)
VideoText	luffhansa	(no text)	Service	society	Point	FULL
Proposed	lufthansa.com	out	Service	society	Point	FULL

the recognition method for video text(VideoText) [46]. The open source Google Tesseract-OCR and commercial ABBYY OCR <sup>1</sup> are used to obtain the recognition results. The recognition results on the testing dataset of these methods are shown in Table 8.1 and 8.2. The baseline results are obtained by directly applying OCR on the images in testing dataset using Google Tesseract and Abbyy. The results of Abbyy are much better than Google Tesseract, that might be due to some preprocessing work that is done by Abbyy before recognition. And the recognition results after binarization are really bad, those classic binarization techniques cannot be applied on web images directly, most of the textual information has been lost after binarization. Our proposed method performs best among all the methods. Compared with the bilateral filtering, our proposed smoothing schema works better on edge preserving than bilateral filtering [57], hence more textual information is retained after smoothing.

The recognition results of the images in Figure 8.1 is shown in Table 8.3 and 8.4, each column denotes one image from Figure 8.1. Full recognition results are also released. As shown in Table 8.3 and 8.4, our proposed technique successfully recognizes almost all of those challenging text, which outperforms other

<sup>1</sup>Google Tesseract:<http://code.google.com/p/tesseract-ocr/>  
 ABBYY FineReader:<http://finereader.abbyy.com/>



Figure 8.6: Some web image examples that cannot be recognized methods.



# Chapter 9

## Document Image Deblurring

Blur is a form of bandwidth reduction in the image formation process, which may arise from different sources and often causes imperfect vision [6]. Sometimes blur may be produced by the photographer to strengthen photo's expressiveness, but unintentional blur will decrease the image quality, which is caused by incorrect focus, object motion, hand shaking and so on. A mathematical blur model is usually required to recover an sharper image from blurry image.

### 9.1 Mathematical Model of Image Blur

Mathematically, the blurring operation can be linear or non-linear. In this paper, we just assume that the operation from sharp image to blur image is linear, because the blur is indeed linear or well approximated by a linear model in most situations [58]. Given a sharp image  $i$  and corresponding blurry image  $i_0$ , A general linear blur  $i_0 = K[i]$  is defined by a linear operator  $K$ . And there may be some noise  $n$  introduced during the blurring procedure. So the linear blur model

---

is defined as follows:

$$i_0 = K[i] + n; \tag{9.1}$$

where the linear operator  $K$  is usually assumed as shift-invariance, which is defined as:

$$\begin{aligned} &\text{for any shift } a \in \mathfrak{R}^2 \\ &i_0(x) = K[i(x)] \Rightarrow i_0(x - a) = K[i(x - a)] \end{aligned} \tag{9.2}$$

From the signal process theorem [59], The linear shift-invariance operator  $K$  must be in the form of convolution:

$$K[i(x)] = \int_{\mathfrak{R}^2} k(x - y)i(y)dy, \tag{9.3}$$

where the term  $K[i(x)]$  denotes the values of convoluted result at position  $x$ , the term  $k$  refers to the convolution operator, and the pair  $x, y$  refers to the image position index.

So from the matrix point of view, the blurry image is constructed by convoluting the sharp image with a two-dimensional matrix with additive noise, which is represented by Equation 9.4. And the two-dimensional matrix is called Point-Spread-Function(PSF) [58], which causes the single bright pixel spread over its

---

neighborhood pixels.

$$g = h * f + n, \tag{9.4}$$

where the variable  $g$  denotes the blurry image, the variable  $f$  denotes the original image, the variable  $h$  denotes the point spread function (PSF), the variable  $n$  denotes the noise, and the term  $*$  denotes the convolution operator.

While transferring the image from time domain to frequency domain using fourier transformation, The convolution operator becomes multiplication as shown below:

$$G = H \cdot F + N, \tag{9.5}$$

where the terms  $G, H, F, N$  denotes the fourier transform pairs of  $g, h, f, n$  respectively. From the perspective of frequency domain, the high frequency information will be discarded, because  $h$  tries to smooth the variation of neighborhood pixel values. So  $H$  can be viewed as a low-pass filter.

By rearranging the elements of  $g, f, h, n$  in Equation 9.4, a general linear model of blurring is given as follows [58]:

$$b = Ax + \hat{n}, \tag{9.6}$$

where the terms  $b, x, \hat{n}$  are the column vector representations of  $g, f, n$ , and the term  $h$  is manipulated to obtain  $A$ . The obtained matrix  $A$  has a little difference under different boundary estimation [58]. Figure 9.1 visualizes the process of blurring.

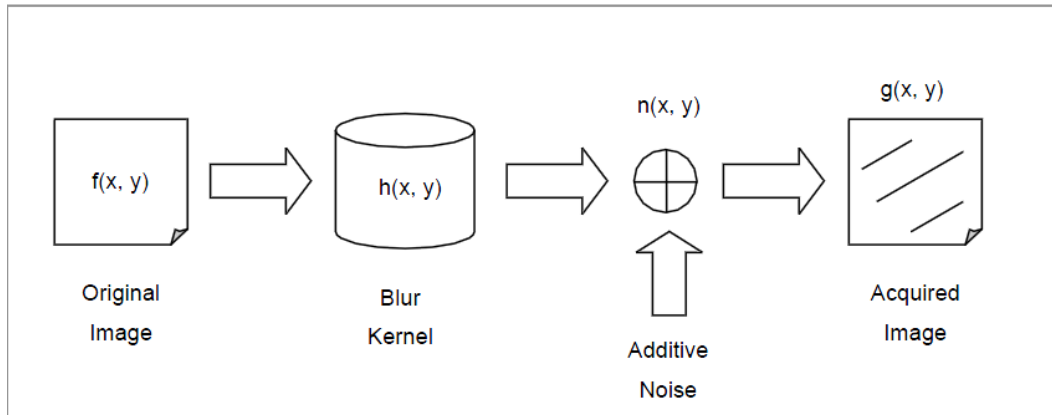


Figure 9.1: The model of Image Blurring, which is adopted from [6]

## 9.2 Image Deblurring as an Ill-posed Problem

The deblurring problem is an ill-posed problem, which can be understood from the following four viewpoints:

- Deblurring is inverting low-pass filtering, which is unstable to reconstruct the high frequency information of the image.
- Deblurring is Backward Diffusion, which is reversing an irreversible random spread process.
- Deblurring is Entropy Decreasing, which never occurs naturally without extra information, according to the second law of statistical mechanics.
- Deblurring is Inverting Compact Operators, which is reconstructing the formerly suppressed dimensions of features and information.

If we look into the blurring model, there are two unknown variables in Equation 9.4: The PSF  $h$  and the original image  $f$ . The PSF  $h$  needs to be estimated

---

first before we can determine the original image. Even if the PSF  $h$  is assumed to be known, the restoration task is still difficult because it is an ill-posed problem, and small noises in the blurred images can be amplified significantly in the restored images. Furthermore, the denoising tasks need to be carried on with deblurring processing synchronously to suppress the noise  $n$ , which makes the image deblurring an even more challenging problem. To solve the ill-posed problem of deblurring, any unique and meaningful solution has to be defined in some proper way with some constraint or prior knowledge. We will discuss some well-known algorithms in the following section.

### 9.3 Related Work

Many works have been reported for image deblurring and applied on different domains, such as medical image, astronomy, and digital image enhancement. The state-of-the-art techniques can be classified in different manners, such as non-iterative/iterative deblurring algorithm, global/spatial deblurring algorithm, and spatial/frequency domain deblurring algorithm [60]. In this thesis, we will divide the deblurring techniques into two categories:

- The non-blind deblurring algorithms [61, 62, 63, 64, 65] that assume the PSF  $h$  is known, and only estimate the original image. Many of the non-blind deblurring algorithms inverse the blurring matrix in frequency domain using Wiener filter [64], wavelet transform [65] and Local Polynomial Approximation [61] and so on. Another type of non-blind deblurring algorithms directly works on spatial domain by employing a regularized term [63, 66] and Bayesian inferences [67, 68].

- 
- The blind deblurring algorithms [69, 70, 71, 72, 73, 74, 75, 76] that try to estimate the PSF and original image together. These algorithms usually use an iterative scheme and employ some additional prior assumption. Molina et. al [72] model the original image and PSF using simultaneous auto-regressive models and Gaussian distributions. Raskar et. al. [77] manipulate the shuttering procedure of camera to make the PSF more suitable for deconvolution. Sparsity-based regularization [71], Natural Image statistics obtained from user-assistant [70] and boundary transparency [69] are also investigated for blind motion deblurring. Reeves and Mersereau [74] make use of the Generalized Cross Validation method. Dai and Wu [75] propose an alpha-channel blur constraint, and Savakis and Trussell [76] employ residual spectral matching technique.

Other approaches try to improve the deconvolution process by incorporating more than one image. A noisy/blurry [78] or flash/non-flash [79] image pairs are combined to produce a better noise-free and sharp image. Cai et. al. [80] use the sparsity of blur kernels to recover a clean image from multiple blurry images. An robust deconvolution algorithm [81] is also developed by using a robust cost function to estimate the two blur kernel of dual blurry images.

Up to now most reported techniques focus on image deblurring under the assumption that images are degraded by certain known type of image blurs. On the other hand, not many techniques have been reported for blur detection and classification that usually need to be performed before the image deblurring. Automatic image blurred region detection and classification without deblurring are useful for learning the image information, which can be used in different multimedia analysis applications such as image segmentation, depth recovery and image

---

retrieval. Most blur detection techniques are based on edge sharpness information [82, 83]. However, these edge sharpness based methods cannot distinguish either the blurred/non-blurred image regions or the type of image blurs. And Kovacs and Sziranyi’s method [84] extracts the clear region using blind deconvolution. Based on the observation that blurred images usually lack high frequency information, some techniques [8, 85] detect image blurs using low pass filtering, without applying the deconvolution.

In particular, we will pay attention on the deblurring for document image domain. To the best of our knowledge, little work has been reported to deal with the restoration of blurred camera images of documents where the target is to extract the text information from blurred document images. However, as described in Chen et al.’s paper [86], the heavy-tailed distribution prior to natural-scene images may not be consistent for document images, the natural-scene image deblurring method based on gradient distribution cannot be directly applied. There are strong edges between the background and text in document images, which may cause strong ringing artifacts after deblurring. So the PSF need to be estimated very accurate. Qi [6] use cepstrum analysis technique for motion blur parameters estimation, but it can only deal with motion blur with a constant acceleration.

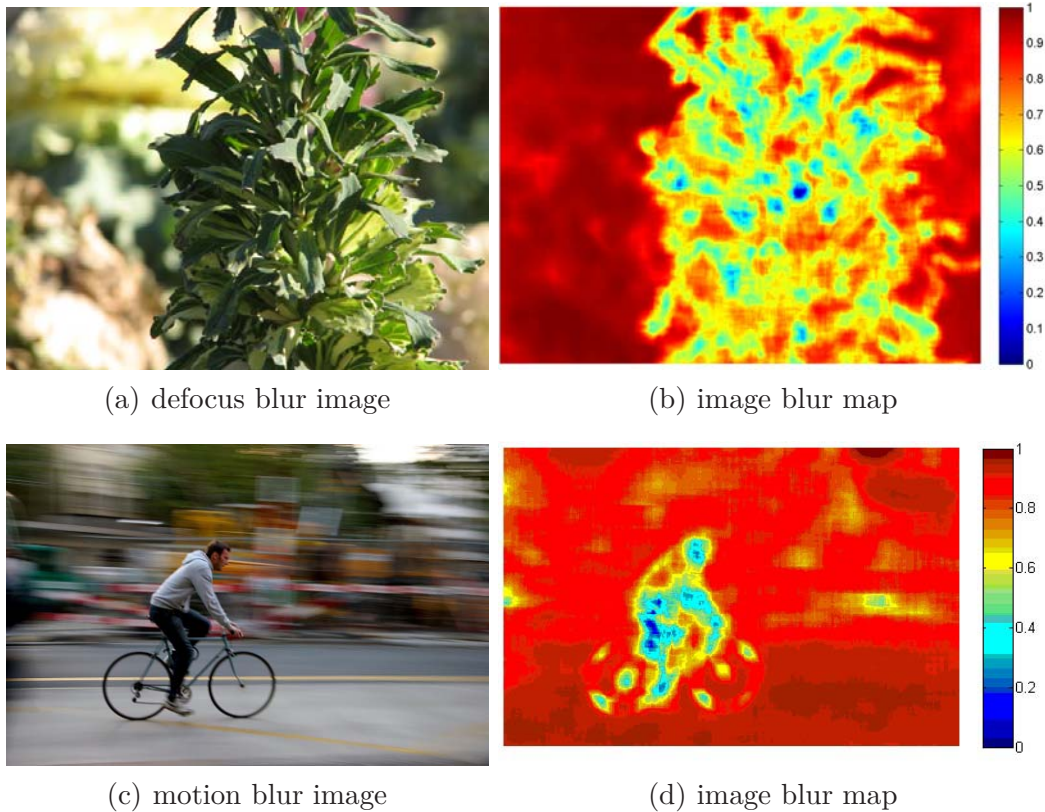


Figure 9.2: Illustration of the blur map constructed by a singular value feature: (a,c) show a pair of example images that suffer from defocus blur and motion blur; (b,d) show the corresponding blur maps that are constructed based on the proposed singular value feature.

## 9.4 Blurred Image Region Detection and Classification

As described in previous section. There exists blur in many digital images due to defocus or motion, which is illustrated in Figure 9.2(a) and (c). In this section, we discuss a technique that can automatic detect image blurred region and identify blur type, which is useful for learning the image information, and can be used in different multimedia analysis applications such as image segmentation, depth



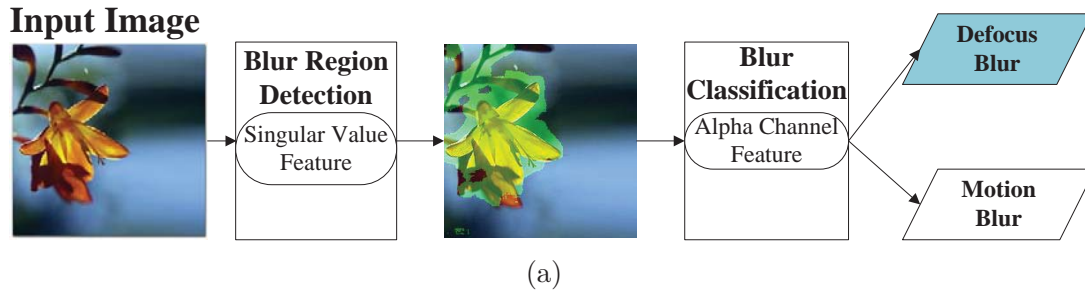


Figure 9.3: Framework of the proposed image blurred region detection and classification technique.

recovery and image retrieval. The overall framework is illustrated in Figure 9.3. The non blurred region is first extracted from the input image, which is marked by green color, and then the blurred region is classified into defocus blur as highlighted in the blue box.

### 9.4.1 Image Blur Features

There are two image features that are used in our proposed blur detection and classification technique. One feature is a singular value feature that can be used as a blur metric to detect image blur effectively and accurately. The other feature is an alpha channel feature that can be used for blur type classification.

#### Singular Value Feature

Singular value decomposition (SVD) is one of the most useful techniques in linear algebra, and has been applied to different areas of computer science. Given an image  $I$ , its SVD can be represented by  $I = U\Lambda V^T$  where the terms  $U, V$  are orthogonal matrices and  $\Lambda$  is a diagonal matrix that is composed of multiple singular values arranged in decreasing order. The image can therefore be decomposed into multiple rank 1 matrices (which are also called eigen-images) [87] as

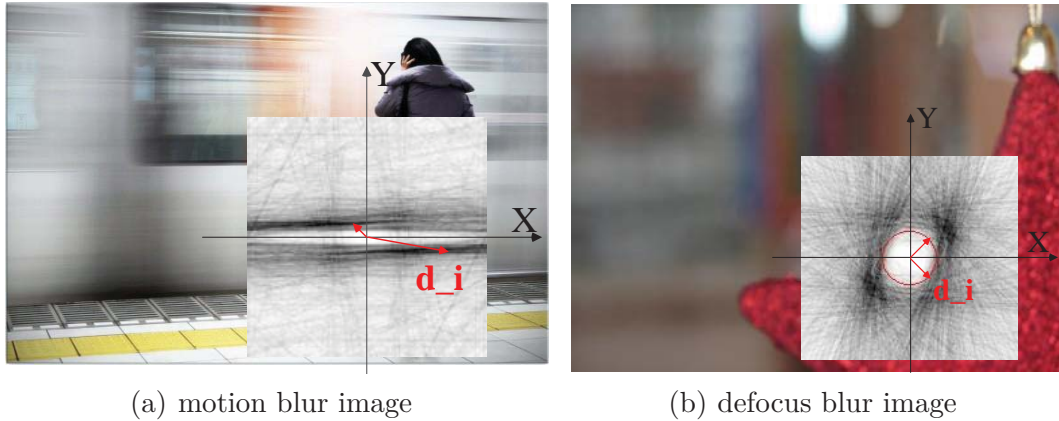


Figure 9.4: A pair of example images suffering from motion blur image and defocus blur and their corresponding  $\nabla\alpha$  distributions in Hough space (a clear white circle region appears in  $\nabla\alpha$  distribution of the defocus blur image as highlighted by a red color circle in (b)).

follows:

$$I = \sum_{i=1}^n \lambda_i \mathbf{u}_i \mathbf{v}_i^T, \quad (9.7)$$

where the terms  $\mathbf{u}_i$ ,  $\mathbf{v}_i$ , and  $\lambda_i$  are the column vectors of  $U$ ,  $V$  and diagonal terms of  $\Lambda$ . Under the framework of digital image compression, the image  $I$  can be approximated by  $I_k$  that sums up the first  $k$  components.

As Equation 9.7 shows, the singular value decomposition actually decomposes an image into a weighted summation of a number of eigen-images where the weights are exactly the singular values themselves. Therefore, the compressed image which omits the small singular value in tail actually replaces the original image by a coarse approximation. Those eigen-images with a small singular value which often capture detailed information are instead discarded.

Such situation is similar to the image blurring that keeps shape structures at

---

large scales but discards image details at small scales. From another viewpoint, those eigen-images in Equation 9.7 provide different scale-space analysis of the image [87], i.e., the first few most significant eigen-images work on large scales that provide rough shapes of the image while those latter less significant eigen-images encode the image details. Suppose that an image  $I$  is convoluted with a Point Spread Function (PSF)  $H$  as follows:

$$I * H = \sum_{i=1}^n \lambda_i (\mathbf{u}_i \mathbf{v}_i^T) * H, \quad (9.8)$$

where the convolution operator  $(\mathbf{u}_i \mathbf{v}_i^T) * H$  tends to increase the scale-space of eigen-images and accordingly causes a loss of high-frequency details. In other word, those small singular values that match to small scale space eigen-images correspond to larger scale-space eigen-images after convolution. As a result, the image details are weakened and those large scale-space eigen-images get higher weights. For a blurred image, the first few most significant eigen-images therefore usually have much higher weights (i.e. singular values) compared with that of a clear image. We thus propose a singular value feature that measures the blurry degree of an image as follows:

$$\beta_1 = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i}, \quad (9.9)$$

where the term  $\lambda_i$  denotes the singular value that is evaluated within a local image patch for each image pixel. As Equation 9.9 shows, the singular feature is actually the ratio between the first  $k$  most significant singular value and all singular values. In our experiments, we set  $k$  to 5.

Generally, blurred image regions have a higher blur degree compared with

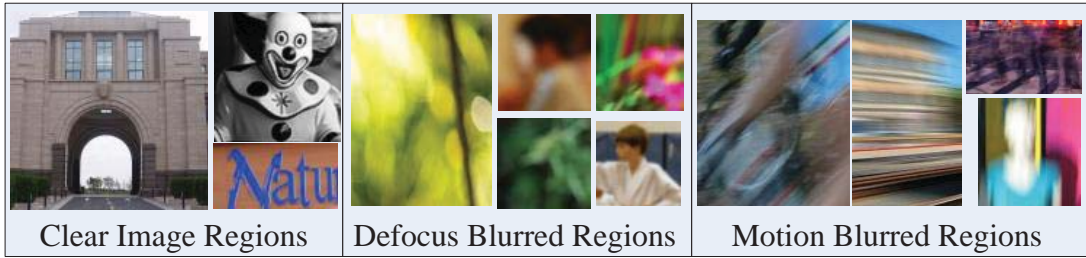


Figure 9.5: Selected samples of blurred/non-blurred image regions from our dataset.

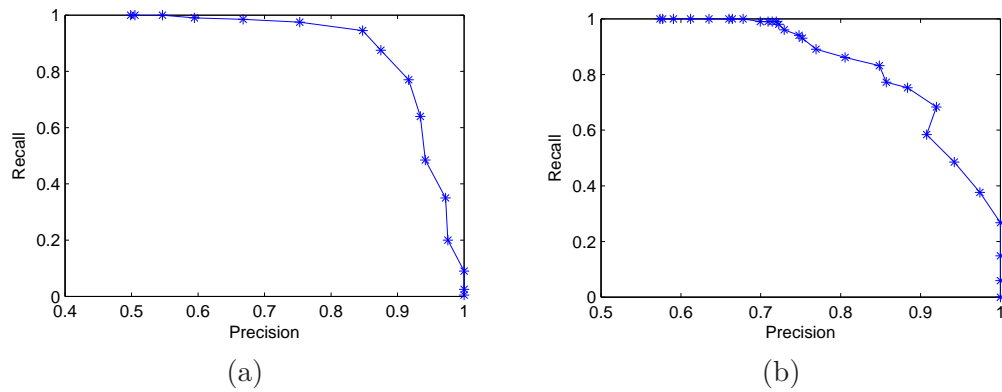


Figure 9.6: Illustration of the Recall-Precision curve of our classification method. (a) the recall-precision curve of 'blur' in blur/non-blur classification using singular value feature. (b) the recall-precision curve of 'defocus blur' in motion/defocus blur classification using alpha channel feature.

clear image regions with no blurs, which is shown in Figure 9.2(b) and (d). So an image pixel will be classified into blurred region if its  $\beta_1$  is larger than a threshold, otherwise, it will be categorized as non blurred region. The selection of threshold need to be selected manually based on the image statistics.

### Alpha Channel Feature

Once the image blur is detected, the image blur type needs to be identified as either motion blur or defocus blur. We use the two-layer image composition

---

model [7], in which an image  $I$  is viewed as a combination of an image foreground  $F$  and an image background  $B$  as follows:

$$I = \alpha F + (1 - \alpha)B, \quad (9.10)$$

where the variable  $\alpha$  lies between 0 and 1. In a clear image, most of the values of  $\alpha$  are either 1 or 0. But in a blurred image, the foreground and background tend to mix together, most of the value of  $\alpha$  lie at the boundary between foreground and background become pure decimal.

An alpha channel model constraint [75] has been reported for motion blur images, which is defined as  $\nabla\alpha \cdot \mathbf{b} = \pm 1$ . The term  $\mathbf{b}$  is a  $2 \times 1$  vector, denotes the blur extension in horizontal and vertical direction. The blur kernels  $\mathbf{b}$  of motion blurred images are usually directional, so the distribution of  $\nabla\alpha$  will be lines, which is shown in Figure 9.4(a). But the image pixel intensities will spread out with constant steps at every directions after blurred for defocus blurred images, so the elements of  $\nabla\alpha$  will have similar magnitude values but different angles. And then the  $\nabla\alpha$  distribution will look like a circle, as shown in Figure 9.4(b).

So we evaluate the circularity of shape pattern of  $\nabla\alpha$  distribution. We calculate the distances from the center to nearest salient points(dark spot) at different directions on the  $\nabla\alpha$  distribution to obtain an array  $d_1, d_2, \dots, d_n$ , where the term  $d_i$  denotes the estimated distance at one direction, which is shown in Figure 9.4. We define the alpha channel feature  $\beta 2$  as the variation of the distance array, which is shown in Equation 9.11.

$$\beta 2 = Var\{d_1, d_2, \dots, d_n\} \quad (9.11)$$

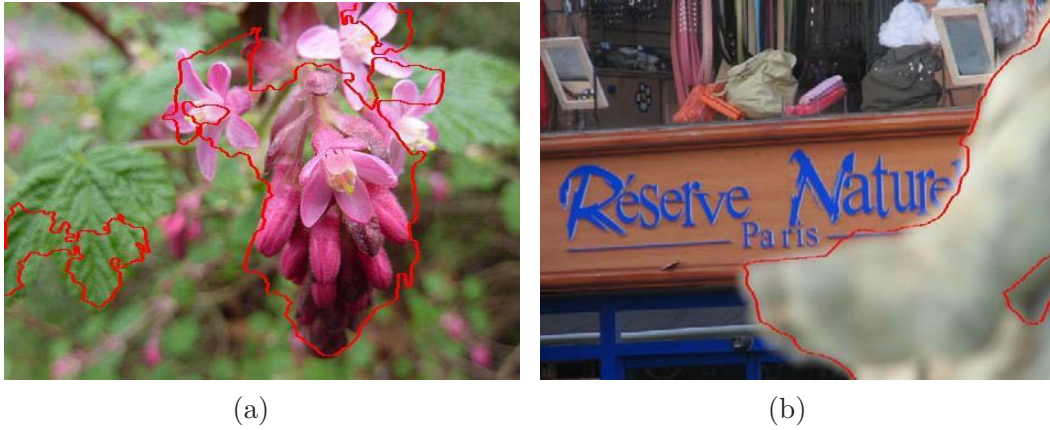


Figure 9.7: Illustration of blurred and non-blurred image region extractions by several example images: the red curves separate the blurred and non-blurred images regions where the image in (a) has a blurred background and the image in (b) has a blurred foreground.

The motion blurred image regions will have much larger  $\beta_2$  values compared with defocus blurred image regions. So a threshold is used on  $\beta_2$  to classify the type of blurred regions into either motion or defocus.

## 9.4.2 Experiments and Applications

### Blur Region and Type Classification

Our blur detection and classification features are tested on different images. Those images are first cropped into smaller regions so that they contain only blur or clear region. We generate 200 clear image regions, 100 defocus blur image regions, and 100 motion blur image regions in total from 100 digital images. Some examples of our dataset are shown in Figure 9.5.

Figure 9.6 shows the recall-precision curve of our method. The threshold for

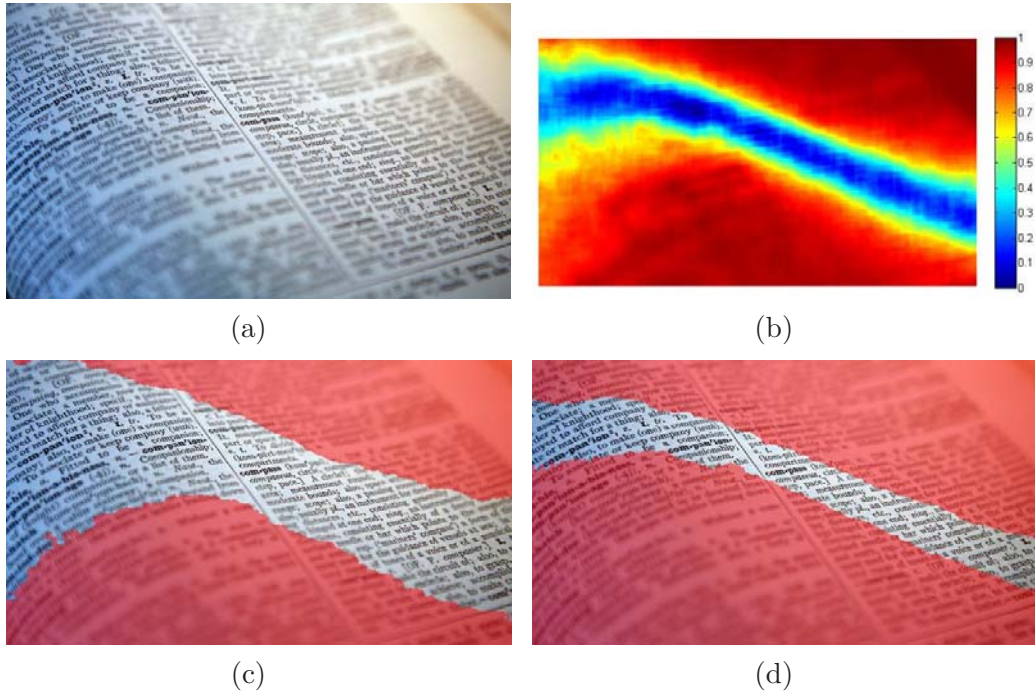


Figure 9.8: Blurred region extraction using different thresholds. The document image in (a) contains defocus blur of different extents. Its corresponding singular value map is shown in (b), those regions with different blur degrees are highlighted in different color. (c) and (d) show the two extracted blurred image regions of (a) when the threshold is set at 0.91 and 0.76, respectively.





Figure 9.9: Comparison of blurred image region extraction: (a-c) show the blurred image regions that are extracted by using Levin's method [7], Liu et al.'s method [8], and our proposed method, respectively. The images in (b) and (c) are adopted from [8]

singular value feature varies from 0 to 1, with a step 0.05, while the threshold of alpha channel feature changes in a range  $[0,0.4]$ , the interval is 0.01. And the method produce best accuracy for blur/nonblur classification when singular value threshold is 0.75, the accuracy is 88.78%. The best accuracy for motion/defocus classification is 80% when the alpha channel threshold is 0.12. Compared with Liu et.al.'s method [8], which reports maximum 76.98% accuracy for blur/nonblur classification and 78.84% for motion/defocus blur classification, our method is simpler yet performs great.

### Blur Region Segmentation

We then use our technique to extract blurred regions of images. A blurred region is extracted based on the constructed singular value blur map. And a blur mask is built based on the threshold obtained in the previous subsection to



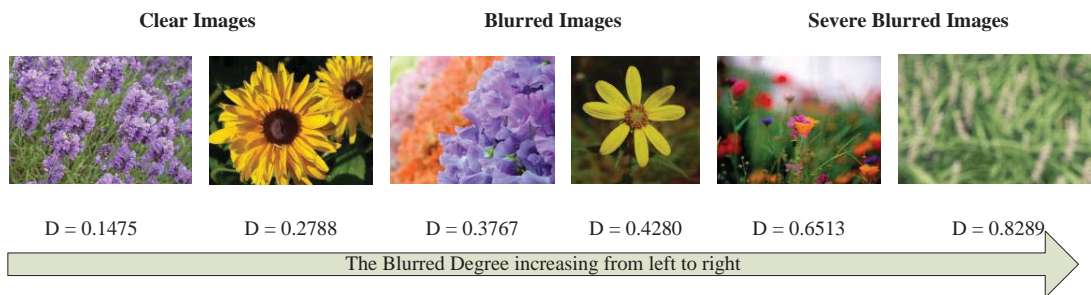


Figure 9.10: Images ranked based on the estimated blurry degree  $D$ : the proposed  $D$  in Equation 9.12 captures the image blurry degree properly, i.e., images are blurred more severely with the increase of  $D$ .

divide blurred/non blurred regions. To evaluate the accuracy, we first manually extracted the blurred regions of 10 partially blurred images as ground truth. The blurred image regions extracted by our proposed method is then evaluated based on their comparison with the ground truth image regions. The accuracy measurement of our region extraction is defined as the ratio of the correctly segmented pixels and the total image pixels. Our experiments show that the blurred region extraction accuracy (by our proposed method) is 83% on averaged over the 10 images. Our method can successfully extracts the blurred image regions, as shown in Figure 9.7.

On the other hand, the size of extracted blurred image regions depends on the singular value threshold because there is no clear boundary between blurred and non-blurred image regions. Figure 9.8(a) shows a blurred document image example and Figure 9.8(b) shows the singular value blur map of the example image that illustrates blur degree variation. Different blurred image extraction results can be obtained when different thresholds are set. As shown in Figure 9.8, two different blurred image regions in (c) and (d) can be extracted from (a) based

---

on the thresholds of 0.91 and 0.76, respectively. We also compare our proposed technique with Liu et al. [8] and Levin’s [88] based on a public image shown in Figure 9.9, which also produces good segmentation result.

### Blur Degree Estimation

Lastly, we can use our blur detection features to rank the blur level of images. The blur degree of an image is affected by two aspects: one is the blur extent of the whole image, which can be evaluated by the singular value feature  $\beta_1$ ; the other is the ratio of the size of blur area to the whole image size. The blur degree of an image is then given as follows:

$$D = k\beta_1 + (1 - k)\left(\frac{\Omega_b}{\Omega}\right), \quad (9.12)$$

where the variable  $k$  is a weight that is set at 0.5 to give equal weights to the two aspects, the term  $\Omega_b$  denotes the size of blurred image regions, the term  $\Omega$  denotes the whole image size. Those severely blurred images should have large values on  $\beta_1$  and  $\frac{\Omega_b}{\Omega}$ , and then the degree  $D$ . Figure 9.10 shows an array of example images that are ranked by the blur degree estimated by our proposed technique, which are sorted properly.

## 9.5 Restoration of Motion Blurred Document Images

The restoration of blurry images is a difficult problem. In this section, we focus on restoring the blurred image caused by motion. As the motion is usually linear in practice, we model the motion blur as a spatially linear invariant system. A

---

novel document image deblur technique is proposed to automatically enhance the document visual quality and restore the lost text information. The proposed technique first builds an alpha channel map for the input blurred document. Then the blur parameters are calculated using the constructed alpha channel map. The  $\alpha$ -motion blur constraint [75] is applied to obtain the blur direction and extent for linear motion blur. Finally, we use the non-blind deblurring method for recovery of blurred documents.

### 9.5.1 Alpha Channel Map

The digital image can be considered as a two-layer image composition model [7], an image  $I$  is viewed as a combination of an image foreground  $F$  and an image background  $B$  as follows:

$$I = \alpha F + (1 - \alpha)B, \quad (9.13)$$

where the term  $\alpha$  is between 0 and 1. Most of the values of  $\alpha$  are either 0 or 1 in a clear image, because there are sharp boundaries between foreground and background. In a blurred image, foreground and background are mixed together at the boundary areas, so the values of  $\alpha$  usually lies between 0 and 1.

The spectral matting [7] can be used for automatic extraction of alpha channel, but it is very time consuming. Hence we propose a much faster and simpler way to extract alpha channel for document images. By experiments, our method runs 5 to 10 times faster than spectral matting method to obtain the alpha channel map. We notice that there is already a foreground-background distribution in document images, compared with other digital images. It would be useful to

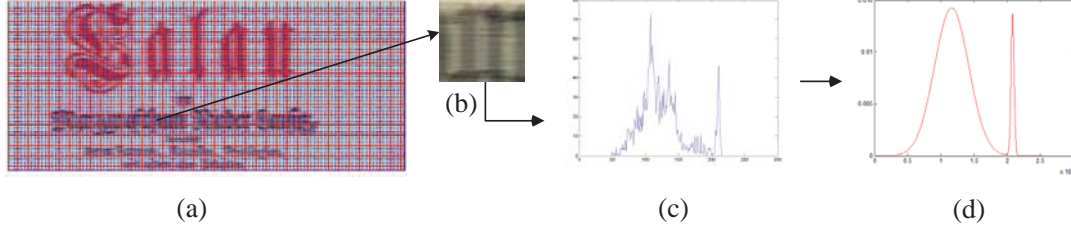


Figure 9.11: Procedure of constructing the alpha channel map. (a) is the input blurry document, and is divided into blocks, which is illustrated by red solid lines, and the overlapped border is illustrated by blue dot lines, (b) shows one block taken from (a), (c) and (d) illustrate the histogram distribution and corresponding Gaussian mixture distribution of (b), respectively.

employ this characteristic of document images for alpha channel extraction.

The overall extraction procedure is shown in Figure 9.11. The input blurry document is first divided into blocks with border overlapped. The block size is  $50 \times 50$ , and border size is 10. Then the histogram distribution within a block is clearer to analyze. If there is mostly background or text in the block, the histogram should only have one peak. Otherwise, more than one peak will appear within the histogram distribution with one denoting the text stroke intensity and the other denoting the background intensity.

From Equation 9.13, since the pixel intensity  $I$  is known, the alpha value of each pixel can be easily derived given the foreground and background boundary intensity  $F$  and  $B$ . This is shown in Equation 9.14.

$$\alpha_i = \begin{cases} 1 & I_i \leq I_{fp} \\ 0 & I_i \geq I_{bp} \\ \frac{I_i - I_{fp}}{I_{bp} - I_{fp}} & \text{else} \end{cases}, \quad (9.14)$$

where the term  $\alpha_i$  denotes the alpha value of a given pixel, the term  $I_i$  denotes the

---

intensity of the given pixel, and the terms  $I_{fp}$ ,  $I_{bp}$  denote the boundary intensity of foreground and background, respectively.

So the remaining issue is to determine the value of foreground and background. In ideal cases, the foreground and background boundary intensity  $I_{fp}$  and  $I_{bp}$  in Equation 9.14 can be directly set as the intensity of the two peaks in a document image block. However, in practice, there is intensity variation within text region and background region, and the background and foreground intensity may shift due to blurry effect. The pixel intensity will expand to both side of the peak intensity, as shown in Figure 9.11(c). So the foreground boundary intensity should be smaller than the foreground peak value, while the background boundary intensity should be larger than the background peak value.

We therefore use Gaussian mixture model to fit the histogram distribution, each peak will be aligned to one Gaussian distribution, as represented in Figure 9.11(d). If only one Gaussian distribution is derived from the histogram, the region is denoted as pure foreground or background. The alpha value of such block can be set to 0 or 1 by the binarization result of the testing document. If two Gaussian distribution models are obtained, then one is aligned to the background peak, the other is aligned to the foreground peak. Then the Gaussian model is used to control how many image pixels will lie between the foreground and background boundaries, which are determined as follows:

$$\begin{aligned}
 I_{fp} &= \{I_i | P(I \leq I_i) == \mu\} \\
 I_{bp} &= \{I_j | P(I \geq I_j) == \mu\}
 \end{aligned}
 \tag{9.15}$$

---

where the terms  $I_{fp}, I_{bp}$  denote the background and foreground boundary intensities, respectively. The term  $P$  denotes the possibility of the Gaussian mixture model corresponding to the background and foreground peak, the term  $P(I \leq I_i)$  is the possibility of a pixel intensity smaller than a given intensity  $I_i$ , the term  $P(I \geq I_j)$  is the possibility of a pixel intensity larger than a given intensity  $I_j$ , and the term  $\mu$  is a threshold lying between 0 and 1, which controls the number of image pixels between the foreground and background boundaries and is set between 0.005 and 0.1 empirically.

The overall flowchart of construction of alpha map within a block is illustrated as follows. The histogram of the testing image block is first extracted. Then the Gaussian mixture model is applied to fit the histogram distribution. If only one Gaussian distribution is obtained, the alpha value of this image block is set to 0 or 1 based on its corresponding binary image. If two Gaussian distributions are generated, the boundary intensity is determined using Equation 9.15 and the alpha value of this image block is calculated using Equation 9.14.

### 9.5.2 Restoration of Motion blur image

The linear motion blur kernel  $h$  can be represented by its direction  $\theta$  and motion length  $l$  in pixels. So we can parametrize  $h$  as a vector  $\mathbf{b} = (u, v)^T$ , where  $u = l \cos \theta, v = l \sin \theta$ . Dai and Wu [75] proved that the following  $\alpha$ -motion blur constraint holds for those  $\|\nabla\alpha\| \neq 0$ :

$$\nabla\alpha \cdot \mathbf{b} = \pm 1, \tag{9.16}$$

where the term  $\nabla\alpha = (\nabla\alpha_x, \nabla\alpha_y)$  denotes the gradient of the alpha channel of

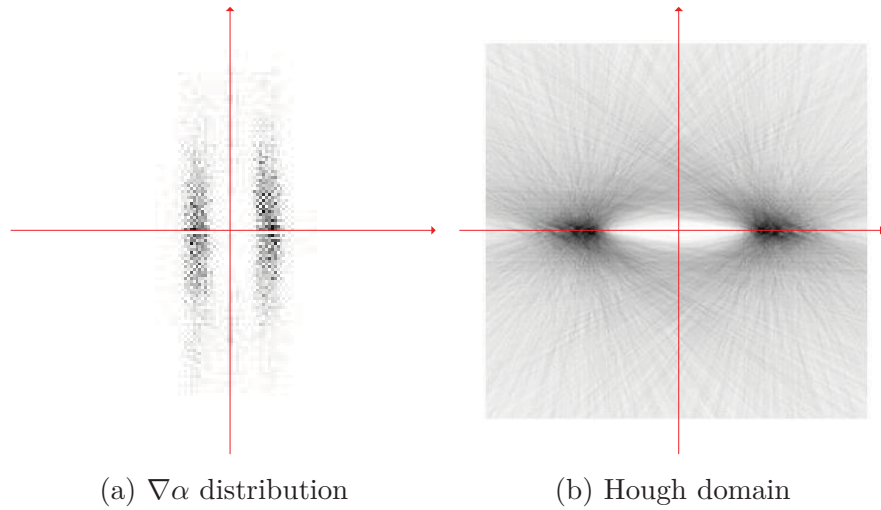


Figure 9.12: Distribution of  $\nabla\alpha$  on 2D  $(\nabla\alpha_x, \nabla\alpha_y)$  coordinate and Hough domain, the origin is in the center.

the input blurry document, the terms  $\nabla\alpha_x, \nabla\alpha_y$  are the gradient value in  $x$  and  $y$  direction, respectively, and the term  $\mathbf{b}$  is a  $2 \times 1$  blur vector, denoting the blur extend in  $x$  and  $y$  directions, as described before.

Equation 9.16 can be viewed as a representation of two symmetry lines with respect to the origin on a 2D Axis, where the term  $\mathbf{b}$  is the coefficient of the two lines. So as described in Dai and Wu's paper [75], we can project the  $\nabla\alpha$  values to the 2D  $(\nabla\alpha_x, \nabla\alpha_y)$  coordinate to form two parallel straight lines on the plane, which is shown in Figure 9.12(a). The  $\nabla\alpha$  values can be further projected to Hough space as shown in Figure 9.12(b). Since there are two possible blur directions, the two salient points correspond to the blur parameters  $\pm\mathbf{b}$ .

So to obtain the blur parameter  $\mathbf{b}$  from Equation 9.16, we need to optimize

---

the following objective function:

$$\mathbf{b}_{est} = \arg \min_{\mathbf{b}} \sum_p \min_{z=\pm 1} G(\nabla \alpha_p \cdot \mathbf{b} - z), \quad (9.17)$$

where the term  $\mathbf{b}_{est}$  denotes the estimated blur parameters, the variable  $p$  denotes one image pixel, the variable  $z$  is either  $+1$  or  $-1$ , the term  $G(*)$  is the penalty function which is proportional to estimation error. We adapted Dai and Wu’s method [75] to obtain the blur parameters. For the blurry image in Figure 7.5(a), we estimated the blur parameter as  $[\pm 0.6798, \mp 20.4030]$ , and the blur parameter is estimated as  $[\pm 0.0300, \mp 17.3603]$  using the alpha channel created by spectral matting [7]. Compared with Levin et. al.’s method [7], our proposed estimation is closer to the truth blur parameters, which is  $[0, 20]$ . Then we use Shan et al.’s non-blind deconvolution method [9] to obtain the restored image.

### 9.5.3 Experiments

We collect 30 document images from different sources, including name cards, book covers, posters, sign board and so on. These images are motion blurred under different direction and extent, half of them are taken naturally through digital camera and the others are blurred synthetically.

First we compare the blur identification accuracy of our proposed method with cepstrum analysis method [6] using the synthetically blurred images. The average least square error of our proposed method is 0.1990, which is more precise than 2.7080 given by Qi’s method [6]. Figure 9.13 shows two restored image examples obtained these two methods. Compared with Qi’s method, our proposed technique produces much better results, which restore most details of the original





Figure 9.13: Restoration Results of motion blurred document images using different methods. The first column is the blurred images, the second column is the corresponding recovered images by cepstrum method, the third column is the corresponding recovered images by proposed method, the last column is the origin clear images.

images.

We also compare our method with Li et al.’s method [89]. The results are shown in Figure 9.14. As Figure 9.14 shows, the visual qualities of our proposed technique are much better than the other two methods. We then use the free google OCR engine to recognize those testing examples to verify our proposed method. The overall recognition recall and precision of the testing images are around 10% and 15% before restoration, and increase to around 60% and 70% after restoration, respectively. There are some ringing artifacts appeared near the strong edges in the recovered images, which decrease the visual quality of the document images. In the future, we will work on this issue and try to reduce the ringing artifacts in our recovered images.



Figure 9.14: Four motion blurred document image examples in the first column and corresponding recovered images by our proposed method in the second column, Shan et al.'s method [9] in the third column and Qi's method [6] in the fourth column, respectively.

# Chapter 10

## Conclusions and Future Work

Several aspects of document image enhancement were introduced in this thesis. The aim of document image enhancement is to improve the accessibility of the textual information embedded in digital images. We have reviewed the state-of-the-art techniques in Document image binarization, Document image deblurring, Web image text recognition.

### 10.1 Conclusions

Document Image Binarization is performed in the preprocessing stage for document analysis and it aims to segment the foreground text from the document background. A fast and accurate document image binarization technique is important for the ensuing document image processing tasks such as optical character recognition (OCR) and Document Image Retrieval(DIR). This research area has been studied for decades, many techniques have been reported and applied on different commercial document analysis applications. However, there are still some unsolved problems need to be addressed due to the high inter/intra-variation between the text stroke and the document background across different document images.

---

Images with text are frequently used on Internet for different purposes. The textual information embedded in web images is useful for different applications, such as web page understanding, filtering and retrieval. Automatic recognition of text from web images plays an important role on extraction and retrieval of web information. However, the texts in web images are usually of low resolution and contain different kinds of degradation including computer-generated-character artifacts, and special effects on images for attractiveness purpose. It makes word recognition a challenge task even after the text has been localized.

Document image deblurring is a mathematically ill-posed problem, because it tries to invert a low-pass filter, and wants to reconstruct the formerly suppressed dimensions of information. To find an unique and meaningful solution, some constrains and prior knowledge have to be incorporated in the deconvolution process. We explore some of research effort in literature, many restoration methods have been proposed and gain successful achievements in many fields. And some techniques are proposed for document image domain. However, there still exists many unsolved problems needed to be investigated.

In summary, although there have been many achievements in the research area of document image enhancement, the enhancement tasks are still difficult and need to be further explored. Our major contribution and future research directions are concluded in the following part.

## **10.2 Contributions of my thesis work**

In this thesis, we have proposed several document enhancement techniques that have been tested on some public datasets and shown superior performance. We would like to see the following major benefits of our proposed methods for these

---

research areas.

### **Document Image Binarization**

- We proposed three document binarization techniques for degraded document images that make use of the local maximum and minimum [5, 90] and background estimation [91] in Chapter 3 to 5, respectively. The proposed techniques have been tested on the recent document image binarization contests [1, 30, 31] and achieved good performance. These methods have been cited by different researchers and used in different applications [92, 93, 94].
- Besides inventing new binarization methods, we also try to adaptively increase the performance of existing document image binarization methods by employing domain knowledge and image statistic. And two frameworks have been developed to re-classified binary results produced by existing methods [95, 96] and combine the binary results of different binarization methods [97] in Chapter 7.

### **Text Recognition on Web Images**

We try to present a text recognition technique that is tolerant to different types of document degradation that widely exist within web images [98] in Chapter 8. As the textual information within web images is contributing more and more due to the future network development, it is important to convert the web images into text format.

### **Document Image Deblurring**

- We proposed a blurred region detection and classification technique [99] in Chapter 9 that can be used in different multimedia analysis applications,

---

such as image segmentation, depth recovery and image retrieval. The connection between image blurs and singular value distributions was observed and a blur metric is designed based on it to detect blurred regions. The distribution pattern of the alpha channel gradient was also captured to differentiate motion and de-focus blur.

- We developed a restoration method for motion blurred document images [100] in Chapter 10. The proposed technique makes use of the bi-modal pattern of document images to construct the alpha channel map, and applied alpha motion blur constraint to obtain the sharp image.

## 10.3 Future Research Direction

### Document Image Binarization

There are still some limitations of our proposed methods as described in previous sections. The background estimation method might not work well on some document images with complex background or with various text objects, such as maps, drawings. The local maximum and minimum method can deal with the ink-bleeding when the back-side text strokes are much weaker compared with the front-side text. But when the back-side text strokes are as dark as or even darker than the front-side text strokes, the LMM method cannot classify the two types of character strokes correctly. In addition, the LMM method depends heavily on the high contrast image pixels. As a result, it may introduce error if the background of the degraded document images contain a certain amount of pixels that are dense and at the same time have a fairly high image contrast. We will study these issues in our future works.

---

Instead of designing a new binarization method, we try to apply a set of learning frameworks on existing binarization methods, which improves not only the performance of existing binarization methods, but also the robustness on different kinds of degraded document images. Better performance may be achieved by more sophisticated learning and classification methods. This issue will be investigated in our future work.

### **Enhancement of Web Images**

Our proposed technique has some limitations. First, there are some heuristic steps in our proposed technique. It will make our technique more robust by employing some learning techniques. Second, the proposed technique cannot handle some special fonts, it could be solved by incorporated a list of potential words. We will explore this issue in future study.

### **Document Image Deblurring**

We propose a document image deblurring technique that automatically recovers the blurry document caused by motion. But the proposed method can only deal with linear motion blur. In future, we will try to extend our method to deal with different kinds of blur and reduce the ringing artifacts in our recovered images.

# Chapter 11

## Publications arising from this work

For citation counts and other details of my publications, please visit Google Citation:<http://scholar.google.com/citations?hl=en&user=ymlKC0EAAAAJ>

1. **Bolan Su**, Shijian Lu, Trung Quy Phan, Chew Lim Tan. A Robust Document Image Binarization Technique for Severely Degraded Document Images. IEEE Transaction on Image Processing. **submitted**
2. **Bolan Su**, Shijian Lu, Trung Quy Phan, Chew Lim Tan. Character Extraction in Web Image for Text Recognition. International Conference on Pattern Recognition, 2012.
3. **Bolan Su**, Shijian Lu, Chew Lim Tan. A Learning Framework for Degraded Document Image Binarization using Markov Random Field. International Conference on Pattern Recognition, 2012.[**Oral**]
4. **Bolan Su**, Shijian Lu, Umapada Pal, Chew Lim Tan. An Effective Staff Detection and Removal Technique for Musical Documents. International Workshop on Document Analysis Systems, 2012.[**Full paper, Oral**]



- 
5. **Bolan Su**, Shijian Lu, Chew Lim Tan: Restoration of Motion Blurred Document Images. In Proceedings of 27th ACM Symposium on Applied Computing, 2012. [**Oral**]
  6. **Bolan Su**, Shijian Lu, Chew Lim Tan: Blurred Image Region Detection and Classification. In Proceedings of 19th ACM international conference on Multimedia (ACMMM), 2011.
  7. **Bolan Su**, Shijian Lu, Chew Lim Tan: Combination of Document Image Binarization Techniques. International Conference on Document Analysis and Recognition (ICDAR), 2011. [**Oral**]
  8. **Bolan Su**, Shijian Lu, Chew Lim Tan. A Self-training Learning Document Binarization Framework. International Conference on Pattern Recognition (ICPR), 2010.
  9. **Bolan Su**, Shijian Lu, Chew Lim Tan. Binarization of Historical Document Images Using the Local Maximum and Minimum. International Workshop on Document Analysis Systems (DAS), 2010.[**Full paper, Oral**]
  10. Shijian Lu, **Bolan Su**, Chew Lim Tan. Document Image Binarization Using Background Estimation and Stroke Edges. International Journal on Document Analysis and Recognition (IJ DAR). 2010.
  11. P. Shivakumara, S. Bhowmick, **Bolan Su**, Chew Lim Tan, U. Pal: A New Gradient based Character Segmentation Method for Video Text Recognition. International Conference on Document Analysis and Recognition (ICDAR), 2011.

- 
12. D. Rajendran, P. Shivakumara, **Bolan Su**, Shijian Lu, Chew Lim Tan: A New Fourier-Moments based Video Word and Character Extraction Method for Recognition. International Conference on Document Analysis and Recognition (ICDAR), 2011.
  13. Trung Quy Phan, P. Shivakumara, **Bolan Su**, Chew Lim Tan: A Gradient Vector Flow-Based Method for Video Character Segmentation. International Conference on Document Analysis and Recognition (ICDAR), 2011.

# References

- [1] B. Gatos, K. Ntirogiannis, and I. Pratikakis, “ICDAR 2009 document image binarization contest(DIBCO 2009),” *International Conference on Document Analysis and Recognition*, pp. 1375–1382, July 2009. [viii](#), [6](#), [11](#), [38](#), [39](#), [43](#), [54](#), [59](#), [61](#), [64](#), [69](#), [116](#)
- [2] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, January 1986. [viii](#), [14](#), [15](#), [16](#), [32](#), [34](#), [66](#)
- [3] M. van Herk, “A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels,” *Pattern Recognition Letters*, vol. 21, pp. 517–521, July 1992. [viii](#), [15](#), [16](#), [57](#)
- [4] D. Ziou and S. Tabbone, “Edge detection techniques - an overview,” *International Journal of Pattern Recognition and Image Analysis*, vol. 8, no. 4, pp. 537–559, 1998. [ix](#), [30](#), [31](#)
- [5] B. Su, S. Lu, and C. L. Tan, “Binarization of historical document images using the local maximum and minimum,” *International Workshop on Document Analysis Systems*, pp. 159–166, 2010. [ix](#), [31](#), [32](#), [33](#), [79](#), [116](#)

- [6] X. Y. Qi, *Motion Deblurring for Optical Character Recognition*. Master Thesis, SoC, NUS, 2004. [x](#), [xiii](#), [88](#), [91](#), [94](#), [111](#), [113](#)
- [7] A. Levin, A. Rav-Acha, and D. Lischinski, “Spectral matting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1699–1712, 2008. [xii](#), [100](#), [103](#), [106](#), [111](#)
- [8] R. Liu, Z. Li, and J. Jia, “Image partial blur detection and classification,” *IEEE conference on Computer Vision and Pattern Recognition*, 2008. [xii](#), [94](#), [103](#), [105](#)
- [9] Q. Shan, J. Jia, and A. Agarwala, “High-quality motion deblurring from a single image,” *ACM Transactions on Graphics*, vol. 27, no. 3, p. 73, 2008. [xiii](#), [111](#), [113](#)
- [10] Z. L, Y. Zhang, and C. L. Tan, “An improved physically-based method for geometrical restoration of distorted document images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 728–734, April 2008. [3](#)
- [11] D. Capel and A. Zisserman, “Super-resolution enhancement of text image sequences,” *International Conference on Pattern Recognition*, pp. 600–605, Spetember 2000. [3](#)
- [12] N. Otsu, “A threshold selection method from gray level histogram,” *IEEE Transactions on System, Man, Cybernetics*, vol. 19, no. 1, pp. 62–66, January 1978. [7](#), [9](#), [27](#), [33](#), [42](#), [44](#), [58](#), [64](#), [65](#), [69](#)
- [13] W. Niblack, *An Introduction to Digital Image Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1986. [8](#), [9](#), [42](#), [44](#), [58](#), [69](#)

- [14] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, January 2000. [8](#), [9](#), [42](#), [44](#), [58](#), [64](#), [65](#), [69](#), [85](#)
- [15] G. Leedham, C. Yan, K. Takru, J. H. N. Tan, and L. Mian, "Comparison of some thresholding algorithms for text/background segmentation in difficult document images," *International Conference on Document Analysis and Recognition*, vol. 2, pp. 859–864, September 2003. [9](#)
- [16] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 312–315, March 1995. [9](#)
- [17] A. D. Brink, "Thresholding of digital images using two dimensional entropies," *Pattern Recognition*, vol. 25, no. 8, pp. 803–808, 1992. [9](#)
- [18] J. Kittler and J. Illingworth, "On threshold selection using clustering criteria," *IEEE transactions on systems, man, and cybernetics*, vol. 15, pp. 652–655, 1985. [9](#)
- [19] L. Eikvil, T. Taxt, and K. Moen, "A fast adaptive method for binarization of document images," *International Conference on Document Analysis and Recognition*, pp. 435–443, September 1991. [9](#)
- [20] I. K. Kim, D. W. Jung, and R. H. Park, "Document image binarization based on topographic analysis using a water flow model," *Pattern Recognition*, vol. 35, pp. 141–150, 2002. [9](#)
- [21] J. R. Parker, C. Jennings, and A. G. Salkauskas, "Thresholding using an

- illumination model,” *International Conference on Document Analysis and Recognition*, pp. 270–273, September 1993. [9](#)
- [22] J. Yang, Y. Chen, and W. Hsu, “Adaptive thresholding algorithm and its hardware implementation,” *Pattern Recognition Letter*, vol. 15, no. 2, pp. 141–150, 1994. [9](#)
- [23] S. Lu and C. L. Tan, “Binarization of badly illuminated document images through shading estimation and compensation,” *International Conference on Document Analysis and Recognition*, vol. 1, pp. 312–316, September 2007. [10](#)
- [24] B. Gatos, I. Pratikakis, and S. J. Perantonis, “Adaptive degraded document image binarization,” *Pattern Recognition*, vol. 39, no. 3, pp. 317–327, March 2006. [10](#), [42](#), [44](#), [64](#)
- [25] J. Bernsen, “Dynamic thresholding of gray-level images,” *International Conference on Pattern Recognition*, pp. 1251–1255, October 1986. [10](#), [42](#), [44](#)
- [26] Y. Liu and S. N. Srihari, “Document image binarization based on texture features,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 540–533, April 1997. [10](#)
- [27] A. Dawoud, “Iterative cross section sequence graph for handwritten character segmentation,” *IEEE Transaction on Image Processing*, vol. 16, no. 8, pp. 2150–2154, August 2007. [10](#)
- [28] Y. Chen and G. Leedham, “Decompose algorithm for thresholding degraded

- 
- historical document images,” *Vision, Image and Signal Processing, IEEE Proceedings*, vol. 152, no. 6, pp. 702–714, December 2005. [10](#)
- [29] N. Howe, “A laplacian energy for document binarization,” *International Conference on Document Analysis and Recognition*, pp. 6–10, September 2011. [10](#), [43](#), [44](#)
- [30] I. Pratikakis, B. Gatos, and K. Ntirogiannis, “ICDAR 2011 document image binarization contest (DIBCO 2011),” *International Conference on Document Analysis and Recognition*, September 2011. [11](#), [38](#), [39](#), [69](#), [116](#)
- [31] Pratikakis, Gatos, and Ntirogiannis, “H-DIBCO 2010 handwritten document image binarization competition,” *International Conference on Frontiers in Handwriting Recognition*, pp. 727–732, November 2010. [11](#), [38](#), [39](#), [43](#), [61](#), [64](#), [69](#), [116](#)
- [32] F. Deng, Z. Wu, Z. Lu, and M. S. Brown, “Binarizationshop: A user-assisted software suite for converting old documents to black-and-white,” *Annual Joint Conference on Digital Libraries*, 2010. [39](#)
- [33] K. Ntirogiannis, B. Gatos, and I. Pratikakis, “An objective evaluation methodology for document image binarization techniques,” in *International Workshop on Document Analysis Systems*, 2008, pp. 217–224. [40](#)
- [34] H. Lu, K. A.C., and S. Y.Q., “Distance-reciprocal distortion measure for binary document images,” in *IEEE Signal Processing Letters*, vol. 11, 2004, pp. 228–231. [42](#)
- [35] T. Lelore and F. Bouchara, “Super-resolved binarization of text based on

- the fair algorithm.” in *International Conference on Document Analysis and Recognition*, September 2011, pp. 839–843. [43](#), [44](#)
- [36] B. Su, S. Lu, and C. L. Tan, “A robust document image binarization technique for degraded document images,” *IEEE Transaction on Image Processing (Accepted)*, 2012. [44](#)
- [37] E. Saund, J. Lin, and P. Sarkar, “Pixlabeler: user interface for pixel-level labeling of elements in document images,” *International Conference on Document Analysis and Recognition*, pp. 646–650, July 2009. [45](#)
- [38] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967. [57](#)
- [39] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, “Lazy snapping,” *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 303–308, August 2004. [67](#)
- [40] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, “A comparative study of energy minimization methods for markov random fields.” In *9th European Conference on Computer Vision*, vol. 2, pp. 16–29, 2006. [69](#)
- [41] T. Kanungo and C. Lee, “What fraction of images on the web contain text?” *International Workshop on Web Document Analysis(WDA)*, pp. 43–46, 2001. [71](#)
- [42] H. Petrie, C. Harrison, and S. Dev, “Describing images on the web: a survey



- of current practice and prospects for the future,” *In Proceedings of Human Computer Interaction International (HCII)*, July 2005. [71](#)
- [43] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy, “ICDAR 2011 robust reading competition - challenge 1: Reading text in born-digital images (web and email),” *International Conference on Document Analysis and Recognition*, pp. 1485–1490, September 2011. [71](#), [72](#), [73](#), [83](#)
- [44] D. L. Smith, J. Field, and E. Learned-Miller, “Enforcing similarity constraints with integer programming for better scene text recognition,” *IEEE Conference on Computer Vision and Pattern Recognition(CVPR 2011)*, pp. 73–80, June 2011. [72](#)
- [45] K. Wang, B. Babenko, and S. Belongie, “End-to-end scene text recognition,” *International Conference on Computer Vision(ICCV 2011)*, 2011. [72](#)
- [46] T. Q. Phan, P. Shivakumara, B. Su, and C. L. Tan, “A gradient vector flow-based method for video character segmentation,” *International Conference on Document Analysis and Recognition*, pp. 1024–1028, September 2011. [72](#), [80](#), [86](#)
- [47] S. Perantonis, B. Gatos, and V. Maragos, “A novel web image processing algorithm for text area identification that helps commercial ocr engines to improve their web image recognition efficiency,” *International Workshop on Web Document Analysis(WDA 2003)*, pp. 61–64, August 2003. [72](#)
- [48] D. Lopresti and J. Zhou, “Locating and recognizing text in WWW images,” *Information Retrieval 2*, pp. 177–206, 2000. [72](#)

- [49] D. Karatzas and A. Antonacopoulos, “Colour text segmentation in web images based on human perception,” *Image and Vision Computing*, vol. 25, no. 5, pp. 564–577, 2007. [72](#)
- [50] N. Leavit, “Vendors fight spam’s sudden rise,” *IEEE Computer*, vol. 40, no. 3, pp. 16–19, 2007. [72](#)
- [51] H. Aradhye, G. Meyers, and J. Herson, “Image analysis for efficient categorization of image-based spam e-mail,” *International Conference on Document Analysis and Recognition*, pp. 914–918, 2005. [72](#)
- [52] G. Fumera, I. Pillai, and F. Roli, “Spam filtering based on the analysis of text information embedded into images,” *International Symposium of text information embedded into images*, pp. 291–296, 2003. [72](#)
- [53] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via L0 gradient minimization,” *ACM Transactions on Graphics (SIGGRAPH Asia 2011)*, vol. 30, no. 6, 2011. [74](#), [77](#), [78](#)
- [54] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing*, vol. 29, pp. 1153–1160, December 1981. [74](#)
- [55] Y. Wang, J. Yang, W. Yin, and Y. Zhang, “Spam filtering based on the analysis of text information embedded into images,” *SIAM Journal on Imaging Sciences*, pp. 248–272, 2008. [77](#)
- [56] I. Jolliffe, *Principal Component Analysis, Series: Springer Series in Statistics, 2nd Edition*. Springer, 2002. [83](#)

- [57] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” *International Conference on Computer Vision (ICCV 1998)*, pp. 839–846, 1998. [85](#), [86](#)
- [58] P. C. Hansen, *Deblurring Images: Matrices, Spectra, and Filtering*. Society for Industrial and Applied Mathematic, 2006. [88](#), [89](#), [90](#)
- [59] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing (2nd Edition)*. New Jersey: Prentice Hall Inc., 1989. [89](#)
- [60] R. C. Puetter, T. R. Gosnell, and A. Yahil, “Digital image reconstruction: Deblurring and denoising,” *Annual Review of Astronomy and Astrophysics*, pp. 139–194, 2005. [92](#)
- [61] V. Katkovnik, K. Egiazarian, and J. Astola, “A spatially adaptive nonparametric regression image deblurring,” *IEEE Transactions on Image Processing*, pp. 1469–1478, 2005. [92](#)
- [62] J. M. Bioucas-dias, “Bayesian wavelet-based image deconvolution: A gem algorithm exploiting a class of heavy-tailed priors,” *IEEE Transactions on Image Processing*, pp. 937–951, 2006. [92](#)
- [63] P. L. Combettes, J. christophe Pesquet, S. Member, and S. Member, “Image restoration subject to a total variation constraint,” *IEEE Transactions on Image Processing*, vol. 13, pp. 1213–1222, 2004. [92](#)
- [64] A. M. Tekalp, H. Kaufman, and J. W. Woods, “Edge-adaptive kalman filtering for image restoration with ringing suppression,” *IEEE Transactions*

- on Acoustics Speech and Signal Processing*, vol. 37, no. 6, pp. 892–899, 1989. [92](#)
- [65] R. Neelamani, H. Choi, and R. Baraniuk, “Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 418–433, 2002. [92](#)
- [66] M. Mignotte, “A segmentation-based regularization term for image deconvolution,” *IEEE Transactions on Image Processing*, pp. 1973–1984, 2006. [92](#)
- [67] R. Narayan and R. Nityananda, “Maximum entropy image restoration in astronomy,” *Annual review of astronomy and astrophysics*, vol. 24, pp. 127–170, 1986. [92](#)
- [68] W. H. Richardson, “Bayesian-based iterative method of image restoration,” *Journal of the Optical Society of America*, vol. 62, no. 1, 1972. [92](#)
- [69] J. Jia, “Single image motion deblurring using transparency,” *IEEE Computer Vision and Pattern Recognition*, 2007. [93](#)
- [70] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” *ACM Transactions on Graphics*, vol. 25, pp. 787–794, July 2006. [93](#)
- [71] J.-f. Cai, H. Ji, C. Liu, and Z. Shen, “Blind motion deblurring from a single image using sparse approximation,” *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, vol. 1, no. 1, pp. 104–111, 2009. [93](#)

- [72] R. Molina, J. Mateos, and A. Katsaggelos, “Blind deconvolution using a variational approach to parameter, image, and blur estimation,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3715–3727, December 2006. [93](#)
- [73] L. Chen and K.-H. Yap, “Efficient discrete spatial techniques for blur support identification in blind image deconvolution,” *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1557–1562, 2006. [93](#)
- [74] S. J. Reeves and R. M. Mersereau, “Blur identification by the method of generalized cross-validation,” *IEEE Trans. Image Processing*, vol. 1, pp. 301–311, 1991. [93](#)
- [75] S. Dai and Y. Wu, “Motion from blur,” *IEEE conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008. [93](#), [100](#), [106](#), [109](#), [110](#), [111](#)
- [76] A. E. Savakis and H. J. Trussell, “Blur identification by residual spectral matching,” *IEEE Transactions on Image Processing*, pp. 141–151, 1993. [93](#)
- [77] R. Raskar, A. Agrawal, and J. Tumblin, “Coded exposure photography: motion deblurring using fluttered shutter,” *ACM Transactions on Graphics*, vol. 25, pp. 795–804, July 2006. [93](#)
- [78] L. Yuan, J. Sun, , Q. Long, and H.-Y. Shum, “Image deblurring with blurred/noisy image pairs,” *ACM Transactions on Graphics*, vol. 26, no. 3, 2007. [93](#)
- [79] S. Zhuo, D. Guo, and T. Sim, “Robust flash deblurring,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. [93](#)

- [80] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, “Blind motion deblurring using multiple images,” *Journal of Computational Physics*, vol. 228, pp. 5057–5071, August 2009. [93](#)
- [81] J. Chen, C.-K. Tang, and L. Quan, “Robust dual motion deblurring,” *IEEE Computer Vision and Pattern Recognition*, 2008. [93](#)
- [82] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, “A non-reference perceptual blur metric.” *International Conference on Image Processing*, vol. 3, pp. 57–60, 2002. [94](#)
- [83] H. Tong, M. Li, H. Zhang, and C. Zhang, “Blur detection for digital images using wavelet transform,” *In Proceedings of IEEE International Conference on Multimedia&Expo*, pp. 17–20, 2004. [94](#)
- [84] L. Kovacs and T. T. Sziranyi, “Focus area extraction by blind deconvolution for defining regions of interest,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1080–1085, 2007. [94](#)
- [85] J. D. Rugna and H. Konik, “Automatic blur detection for metadata extraction in content-based retrieval context,” *SPIE*, vol. 5304, pp. 285–294, 2003. [94](#)
- [86] X. Chen, X. He, J. Yang, and Q. Wu, “An effective document image deblurring algorithm,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 369–376. [94](#)
- [87] H. Andrews and C. Patterson, “Singular value decompositions and digi-

- tal image processing,” *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 24, pp. 26–53, 1976. [96](#), [98](#)
- [88] A. Levin, “Blind motion deblurring using image statistics,” *Advances in Neural Information Processing Systems (NIPS)*, 2007. [105](#)
- [89] L. Xu and J. Jia, “Two-phase kernel estimation for robust motion deblurring,” *European Conference on Computer Vision*, pp. 157–170, 2010. [112](#)
- [90] B. Su, S. Lu, and C. L. Tan, “A robust document binarization technique for severely degraded document images,” *IEEE Transaction on Image Processing*, submitted. [116](#)
- [91] S. Lu, B. Su, and C. L. Tan, “Document image binarization using background estimation and stroke edges,” *International Journal on Document Analysis and Recognition*, vol. 13, pp. 303–314, December 2010. [116](#)
- [92] P. Shivakumara, T. Q. Phan, S. Lu, and C. L. Tan, “Video character recognition through hierarchical classification,” *International Conference on Document Analysis and Recognition*, pp. 131 – 135, September 2011. [116](#)
- [93] A. Alaei, U. Pal, and P. Nagabhushan, “A new scheme for unconstrained handwritten text-line segmentation,” *Pattern Recognition*, vol. 44, no. 4, pp. 917 – 928, 2011. [116](#)
- [94] V. P. Truyen, Z. Bilan, and N. Masaki, “Development of nom character segmentation for collecting patterns from historical document pages,” *Work-*

- shop on Historical Document Imaging and Processing*, pp. 133–139, 2011. [116](#)
- [95] B. Su, S. Lu, and C. L. Tan, “A self-training learning document binarization framework,” *20th International Conference on Pattern Recognition*, pp. 3187–3190, August 2010. [116](#)
- [96] —, “A learning framework for degraded document image binarization using markov random field,” *International Conference on Pattern Recognition*, 2012. [116](#)
- [97] —, “Combination of document image binarization techniques,” *International Conference on Document Analysis and Recognition*, pp. 22–26, September 2011. [116](#)
- [98] B. Su, S. Lu, T. Q. Phan, and C. L. Tan, “Character extraction in web image for text recognition,” *International Conference on Pattern Recognition*, 2012. [116](#)
- [99] B. Su, S. Lu, and C. L. Tan, “Blurred image region detection and classification,” *19th ACM international conference on Multimedia*, pp. 1397–1400, 2011. [116](#)
- [100] —, “Restoration of motion blurred document images,” *27th Annual ACM Symposium on Applied Computing*, pp. 767–770, 2012. [117](#)