Generic Methods for Document Layout Analysis and Preprocessing

Dissertation

submitted to the

Department of Computer Science

Technical University of Kaiserslautern

for the fulfillment of the requirements for the doctoral degree

Doctor of Engineering

(Dr.-Ing.)

by

Syed Saqib Bukhari

Thesis supervisors:
Prof. Dr. Thomas M. Breuel, TU Kaiserslautern
Prof. Dr. Andreas Dengel, TU Kaiserslautern

Chair of supervisory committee:
Prof. Dr. Karsten Berns, TU Kaiserslautern

D 386

Abstract

Generic layout analysis–process of decomposing document image into homogeneous regions for a collection of diverse document images–has many important applications in document image analysis and understanding such as preprocessing of degraded warped, camera-captured document images, high performance layout analysis of document images containing complex cursive scripts, and word spotting in historical document images at page level. Many areas in this field like generic text line extraction method are considered as elusive goals so far, still beyond the reach of the state-of-the-art methods [NJ07, LSZT07, KB06]. This thesis addresses this problem in such a way that it presents generic, domain-independent, text line extraction and text and non-text segmentation methods, and then describes some important applications, that were developed based on these methods. An overview of the key contributions of this thesis is as follows.

The first part of this thesis presents a generic text line extraction method using a combination of matched filtering and ridge detection techniques, which are commonly used in computer vision. Unlike the state-of-the-art text line extraction methods in the literature, the generic text line extraction method can be equally and robustly applied to a large variety of document image classes including scanned and camera-captured documents, binary and grayscale documents, typed-text and handwritten documents, historical and contemporary documents, and documents containing different scripts. Different standard datasets are selected for performance evaluation that belong to different categories of document images such as the UW-III [GHHP97] dataset of scanned documents, the ICDAR 2007 [GAS07] and the UMD [LZDJ08] datasets of handwritten documents, the DFKI-I [SB07] dataset of camera-captured documents, Arabic/Urdu script documents dataset, and German calligraphic (Fraktur) script historical documents dataset. The generic text line extraction method achieves 86% ($n = 23,763$ text lines in 650 documents) text line detection accuracy which is better than the aggregate accuracy of 73% of the best performing domain-specific state-of-the-art methods. To the best of the author's knowledge,

it is the first general-purpose text line extraction method that can be equally used for a diverse collection of documents.

This thesis also presents an active contour (snake) based curled text line extraction method for warped, camera-captured document images. The presented approach is applied to DFKI-I [SB07] dataset of camera-captured, Latin script document images for curled text line extraction. It achieves above 95% ($n = 3,091$ text lines in 102 documents) text line detection accuracy, which is significantly better than the competing state-of-the-art curled text line extraction methods. The presented text line extraction method can also be applied to document images containing different scripts like Chinese, Devanagari, and Arabic after small modifications.

The second part of this thesis presents an improved version of the state-of-the-art multiresolution morphology (Leptonica) based text and non-text segmentation method [Blo91], which is a domain-independent page segmentation approach and can be equally applied to a diverse collection of binarized document images. It is demonstrated that the presented improvements result in an increase in segmentation accuracy from 93% to 99% ($n = 113$ documents).

This thesis also introduces a discriminative learning based approach for page segmentation, where a self-tunable multi-layer perceptron (MLP) classifier [BS10] is trained for distinguishing between text and non-text connected components. Unlike other classification based page segmentation approaches in the literature, the connected components based discriminative learning based approach is faster than pixel based classification methods and does not require a block segmentation method beforehand. A segmentation accuracy of 96% ($n = 113$ documents) is achieved in comparison to the state-of-the-art multiresolution morphology (Leptonica) based page segmentation method [Blo91] that achieves a segmentation accuracy of 93%. In addition to text and non-text segmentation of Latin script documents, the presented approach can also be adapted for document images containing other scripts as well as for other specialized layout analysis tasks such as digit and non-digit segmentation [HBSB12], orientation detection [RBSB09], and body-text and side-note segmentation [BAESB12].

Finally, this thesis presents important applications of the two generic layout analysis techniques, ridge-based text line extraction method and the multi-resolution morphology based text and non-text segmentation method, discussed above. First, a complete preprocessing pipeline is described for removing different types of degradations from grayscale warped, camera-captured document images that includes removal of grayscale degradations such as non-uniform shadows and blurring through binarization, noise cleanup

applying page frame detection, and document rectification using monocular dewarping. Each of these preprocessing steps shows significant improvement in comparison to the analyzed state-of-the-art methods in the literature. Second, a high performance layout analysis method is described for complex Arabic script document images written in different languages such as Arabic, Urdu, and Persian and different styles for example Naskh and Nastaliq. The presented layout analysis system is robust against different types of document image degradations and shows better performance for text and non-text segmentation, text line extraction, and reading order determination on a variety of Arabic and Urdu document images as compared to the state-of-the-art methods. It can be used for large scale Arabic and Urdu documents' digitization processes. These applications demonstrate that the layout analysis methods, ridge-based text line extraction and the multi-resolution morphology based text and non-text segmentation, are generic and can be applied easily to a large collection of diverse document images.

Acknowledgements

First of all, I would like to thank my advisor Prof. Thomas Breuel for introducing me to the exciting field of document image analysis and for his support, advice, and guidance throughout my Ph.D. work. I would also like to thank Prof. Andreas Dengel for reviewing this thesis as a referee. His comments were very helpful and have improved the quality of this work significantly.

I am grateful to Dr. Faisal Shafait for many interesting discussions. I would also like to thank graduate students, researchers and other staff members of Image Understanding and Pattern Recognition group at the University of Kaiserslautern and Multimedia Analysis & Data Mining group at the German Research Center for Artificial Intelligence (DFKI, Kaiserslautern) for providing a creative and friendly environment.

Finally, I would like to thank my parents and my wife for their continuous support in all aspects of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Layout analysis is the process of decomposing a document image into homogeneous regions such as text, graphics, and line drawings. It is a major performance limiting step in document analysis and understanding. A large number of layout analysis methods have been proposed for typed-text, handwritten and historical document images in the literature over the last four decades. Many of them have got community recognition and are being used in a variety of applications. However, one of the common limitations of most of these methods is that they are domain-specific or difficult to apply robustly on a variety of document images. Layout analysis of a collection of diverse document images, *generic layout analysis*, is an open challenging problem. This thesis presents generic text line extraction and text and non-text segmentation approaches that can be equally applied to a diverse collection of document images such as books, magazines, newspapers, handwritten and historical documents. Moreover, these documents may be scanned or camera-captured, binary or grayscale, typed-text or handwritten, and containing different scripts like Latin, Chinese, and Arabic.

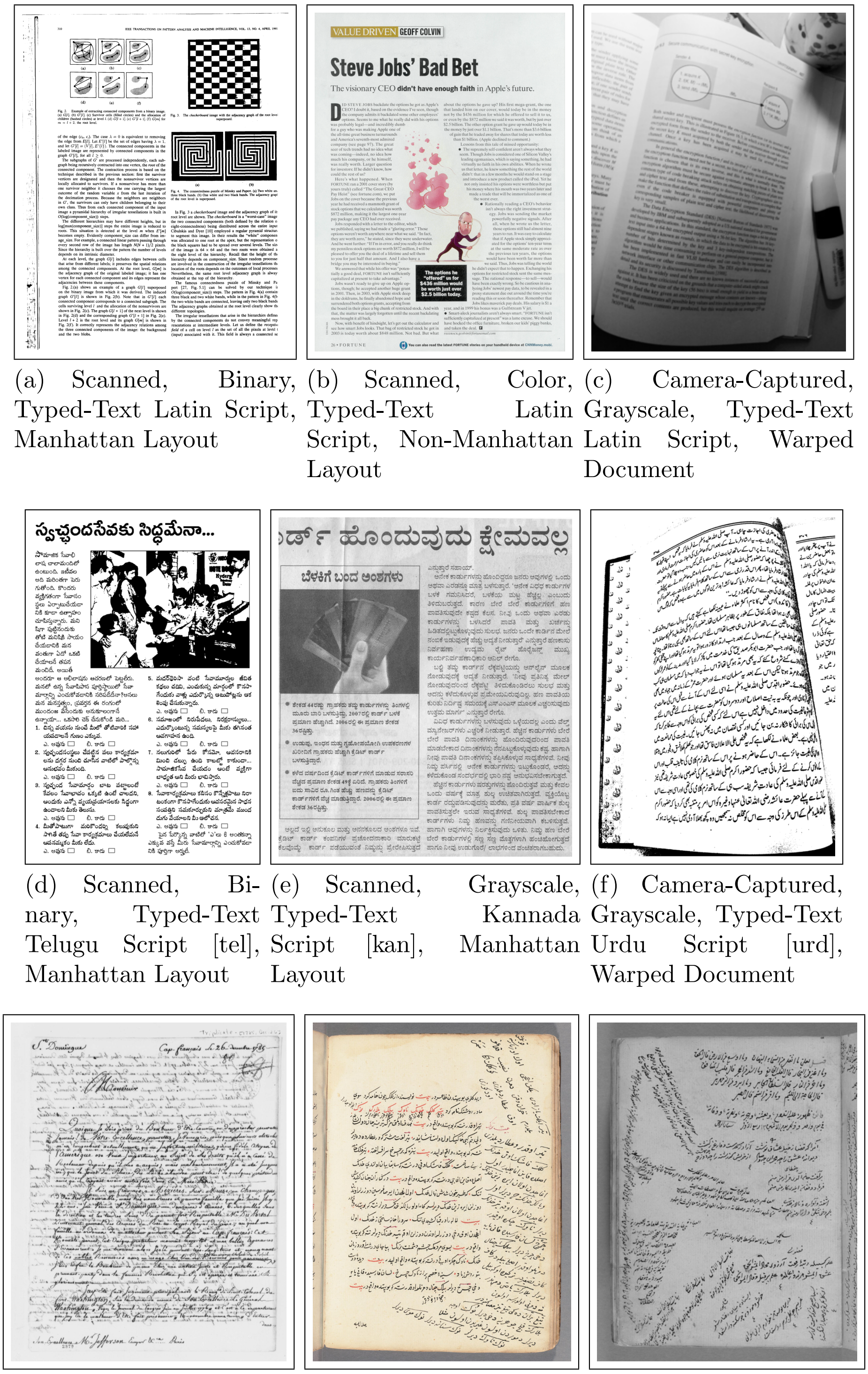
The rest of this chapter is organized as follows. Section 1.1 briefly introduces the reader to the field of document image processing. Section 1.2 outlines the scope of this thesis by defining diverse collection of document images. Section 1.3 briefly describes the problem of text line extraction, and discusses various text line extraction methods and associated challenges in the sub-domains of camera-captured, complex scripts, and handwritten/historical documents. Section 1.4 highlights the need of a generic text line extraction method and finally Section 1.5 enlists the main contributions of this thesis.

1.1 Document Image Processing

Document image processing is a subfield of digital image processing. It mainly deals with the transformation of digitized document images into electronic/symbolic form for storage, transmission, reuse, and modification. There are many different types of documents, such as books, magazines, newspapers, archives, forms, invoices, envelops, engineering drawings, maps, and music sheets. Together with some common preprocessing steps, different documents may have different document image processing goals. For example, document image processing goal for a book may be to recognize text, while the document image processing objective for maps may be to extract GIS (Geographic Information Systems) representation. Traditionally, the field of document image processing is distinguished between *mostly text image processing* and *mostly graphics image processing* [Nag00]. Categorization of mostly text and mostly graphics document images and their corresponding document image processing goals is as follows. Mostly text document images are composed of plain/illustrated text (books, magazines, newspaper, archives) and structured text (forms, invoices, envelop), where plain/illustrated texts are usually processed for OCR in reading order and link to illustrations [MR05,APBP09] and structured texts are processed for generating executable form together with OCR [YJ96, NBO09]. Mostly graphics document images are composed of maps, engineering drawings, and music sheets; their correspoding document image processing goals are GIS (Geographic Information Systems) representation for maps [CA09], CAD (Computer Aided Design) format for engineering drawing [Tom98,WZY07], and MIDI (Musical Instrument Digital Interface) representation for music scores [BB01]. This thesis deals with document image processing (layout analysis) of diverse collection of *mostly (plain/illustrated) text document images* including books, magazines, newspaper, archives, handwritten and historical documents; hereafter they are referred to as *document images*.

1.2 Diverse Collection of Document Images

Traditionally, scanners are used for document digitization. They produce planar surface document images with slight variations in skew angle. Sample scanned document images are shown in Figure 1.1(a) and 1.1(b), where the document layout in Figure 1.1(a) is called Manhattan (rectangular) layout and the document layout in Figure 1.1(b) is referred to as non-Manhattan layout. With the advent of less expensive and widely used digital camcorders, digital cameras, cell-phone cameras and PC-cams, consumer-end digital cameras



(a) Scanned, Binary, Typed-Text Latin Script, Manhattan Layout

(b) Scanned, Color, Typed-Text Latin Script, Non-Manhattan Layout

(c) Camera-Captured, Grayscale, Typed-Text Latin Script, Warped Document

(d) Scanned, Binary, Typed-Text Telugu Script [tel], Manhattan Layout

(e) Scanned, Grayscale, Typed-Text Kannada Script [kan], Manhattan Layout

(f) Camera-Captured, Grayscale, Typed-Text Urdu Script [urd], Warped Document

(g) Scanned, Grayscale, Handwritten/Historical English Script, Background Noise [lib]

(h) Scanned, Color, Handwritten/Historical Arabic Script, Irregular Layout [har]

(i) Scanned, Grayscale, Handwritten/Historical Arabic Script, Irregular Layout [har]

Figure 1.1: A diverse collection of document images.

are also being used for document digitization nowadays [TZND99, Bre05]. Cameras have several advantages over scanners with respect to document digitization, such as they support fast, flexible, and non-contact document imaging which is especially helpful for the digitization of historical documents in libraries. In general, camera based document image processing systems are more flexible than scanner based systems [DLL03]. A sample grayscale, camera captured document image is shown in Figure 1.1(c), where the surface of document image is curvilinear because of non-planar surface of the page. All of the document images which are shown in Figures 1.1(a), 1.1(b), and 1.1(c) are composed of typed-text Latin (English) script. Latin script is one of the simplest scripts with respect to document image processing [Nag00] as compared to complex scripts like Arabic, Persian, African, Urdu, and Indic scripts (like Kannada and Telugu) [KSK06]. Sample document images of typed-text complex scripts like Telugu, Kannada, and Urdu scripts are shown in Figures 1.1(d), 1.1(e), and 1.1(f), respectively. Besides typed-text documents, there exits a huge amount of handwritten document images especially in the form of historical manuscripts in libraries all over the world [LSZT07, KB06]. Sample handwritten/historical document images are shown in Figure 1.1(g), 1.1(h), and 1.1(i). Document images with typed-text scripts generally follow some regular/structured layout as shown in the top two rows of Figure 1.1. On the other side, handwritten document images usually consist of irregular layouts as shown in the bottom row of Figure 1.1.

Based on the discussion above it can be concluded that, document images can be categorized into different classes with respect to the following features: i) writing style (typed-text or handwritten), ii) scripts and languages (like Latin, Chinese, and Arabic), iii) digitization method (scanner or camera), and iv) intensity values (binary, grayscale, color). In this dissertation, a corpus of document images consisting of a variety of these features, as shown in Figure 1.1, is referred to as *diverse collection of document images.*

1.3 Text Line Extraction

A document image usually consists of text and non-text regions. The first step of document image processing in the pipeline of information retrieval from a document image is *geometric layout analysis*. Geometric layout analysis is the process of document image decomposition into homogeneous regions and determination of their reading order. Document images as shown in the Figure 1.1(a) consist of rectangular blocks/zones, whereas non-Manhattan layout (Figure 1.1(b)) and camera-captured document images (Figure 1.1(c)) do not contain rectangular blocks, but contain uniform segments. Handwritten/historical document images, on the other hand, mostly contain irregular layout as shown in the bottom row of Figure 1.1. Therefore, instead of blocks, *text line* is the dominant geometrical layout structure in the context of diverse collection of document images [LSZT07].

Text line extraction is considered as one of the most important document image analysis steps. In general, extracted text lines [BB08], or further segmented words [Vin02], or characters [AYV01], either machine printed or handwritten, are fed into recognition engine which converts them into appropriate format (like ASCII and UTF-8). Besides that, extracted text lines also provide important information for implementing other specialized tasks for different categories of document images. Text line based skew and orientation detection [KD05, AKR07, vBSB10], zone segmentation [O'G93], and document cleanup using page frame detection [SvBKB08] are common preprocessing steps in scanned typed-text document images. For camera-captured warped document images, text line based monocular dewarping methods are applied for rectification [ZT01, DLL03, LT06a, GPN07, FWL$^+$07, SB07], so that the dewarped document images can be processed by traditional scanner based OCR systems. For handwritten document images, text line information helps in writer's identification [MMB01]. For historical document images, where automatic reading of a complete image is still a long-term objective because of a large vocabulary and unusual fonts, text line extraction is an important step for indexing and word spotting [LSZT07,SSG05,KB06,ASS07,MR05]. Additionally, before word spotting, the geometric rectifications in historical document images can also be corrected by using text lines information [BYHKD09]. In short, regardless of the categories of document images, text line extraction is the first and the most critical step for document image processing.

Over the last four decades, several text line extraction methods have been proposed. Comprehensive overview of the state-of-the-art page segmentation / text line extraction methods has been provided in [CCMM98b, Nag00, LSZT07]. Some of these algorithms have come to widespread use for analyzing documents in different scripts and languages [SKB08b], such as smearing [WCW82], whitespace analysis [BJF90], X-Y cut [NSV92], Docstrum [O'G93], Voronoi [KSI98], and constrained text line extraction [Bre02b]. These algorithms perform well for scanned, binarized, cleaned document images with simple layouts and scripts, but usually fail on warped camera-captured document images [BSB10b], complex scripts document images [KKJ07], and handwritten/historical document images [LZDJ08, LSZT07] because of the specific challenges in these types of documents images. Some of these documents and their associated challenges are described as follows. Hand-held camera-captured document images (Figure 1.1(c) and 1.1(f)) usually consist of degradations like geometric and perspective distortions, low resolution, uneven lighting, wide-angle lens distortion, complex backgrounds (noise), zooming and focusing. Complex scripts document images (Figure 1.1(d), 1.1(e)), and 1.1(f)) may contain variations in spatial distribution of the components within a line structure, positional variation of a modifier component, variation of size of connected component within a line, lack of four-line rule standard, and overlapping of connected components from neighboring lines. Historical/handwritten document images (Figure 1.1(g), 1.1(h), and 1.1(i)) usually consist of irregular page layouts, curvilinear lines, touched and/or overlapped components, character size variations, smudges, bleed-through, faded print, as well as hand-held camera-captured document imaging artifacts (noise, skew, geometric/perspective distortions, and uneven lighting).

Text line extraction for complex document images, like camera-captured, complex script, and handwritten/historical document images, is an open challenging field. Every year, different text line extraction methods are being proposed for solving specific problems in each of these categories of complex document images. Detailed overview of the state-of-the-art text line extraction methods for warped camera-captured document images has been provided in [BSB10b] and for handwritten and historical document images in [LSZT07], and some of the specialized script-specific text line extraction approaches for complex script document images can be found in [KKJ07, SHKB06]. However, most of these algorithms rely on certain assumptions about the structure of target document images for which they are designed and fail on other types of document images, where the underlying assumptions are not hold.

1.4 A Generic Text Line Extraction Method and Applications

There is no universal or generic text line extraction method that can be robustly applied to a diverse collection of document images. However, a generic text line extraction method can solve a variety of document image processing problems/tasks such as it can be used: i) to overcome the requirement of specific text line extraction methods for different categories of document images, ii) to improve the performance of existing OCR software for complex typed-text document layouts (like non-Manhattan), iii) to remove geometric and perspective distortions of warped camera-captured document images, so that dewarped document images can be directly processed by traditional/existing scanner based OCR software, iv) to solve layout analysis problems for document images containing complex script (like Urdu, Telugu, and Kannada), v) to upgrades/promotes the document image processing pipeline for handwritten/historical document images to page level, which traditionally focuses on the recognition at line, word, or character levels because of complex (irregular) page layouts. A universal/generic text line extraction method has remained an elusive goal so far [NJ07,LSZT07,KB06] and still beyond the reach of the state-of-the-art in the field.

1.5 Contributions of this Dissertation

The main contributions of this dissertation are:

- A generic text line extraction method is presented (Chapter 3). This method is based on two standard computer vision algorithms: matched filtering and ridge detection. To the best of the author's knowledge, the generic text line extraction method is the first general purpose text line extraction technique that can be directly applied on a large variety of document image classes (typed-text, handwritten and historical documents, scanned and camera-captured documents, binary and grayscale documents, and document images containing different scripts such as Latin, Chinese, Arabic, and Indic). Different standard datasets are selected for performance evaluation that belong to different categories of document images such as the UW-III [GHHP97] dataset of scanned documents, the ICDAR 2007 [GAS07] and the UMD [LZDJ08] datasets handwritten documents, the DFKI-I [SB07] dataset camera-captured documents, Arabic/Urdu script documents dataset, and German calligraphic (Fraktur) script historical documents dataset. A text line detection accuracy of 86% ($n = 23,763$ text lines in 650 documents) is achieved, which is significantly better than the aggregate accuracy of 73% of the best performing domain-specific state-of-the-art methods.

- A novel coupled snakelets model is introduced for document image segmentation (Chapter 2). The coupled snakelets model is an extension of a traditional active contour (snake) model. It introduces an automatic initialization of multiple weighted-coupled pairs of snakes on discrete points and their deformation in evolving fashion. In this thesis, the coupled snakelets model is adapted for extracting curled text lines from typed-text warped camera-captured documents through estimating their x-lines and baselines pairs. Above 95% ($n = 3,091$ text lines in 102 documents) text line extraction accuracy is achieved on the DFKI-I [SB07] dataset, which is significantly better than previously reported results on this dataset.

- The modifications to the Leptonica's text and halftone segmentation algorithm [Blo91] are presented in Chapter 5 for making it a generic text and non-text image segmentation algorithm, where non-text elements can be halftones, drawings, maps, and graphics. For performance evaluation, standard datasets like the UW-III [GHHP97] dataset, the ICDAR 2009 page segmentation competition test dataset [APBP09] and the circuit diagrams dataset are selected that contain texts, halftones, line drawings, maps and graphics elements. The presented improvements result in a significant performance gain from 93% to 99% ($n = 113$ documents).

- A novel discriminative learning based text and non-text segmentation method is introduced in Chapter 4, where a self-tunable multi-layer perceptron (MLP) classifier [BS10] is trained for distinguishing between text and non-text connected components using a combination of shape- and context-based features. A segmentation accuracy of 96% ($n = 113$ documents) is achieved in comparison to the state-of-the-art multiresolution morphology based Leptonica's text and halftone segmentation method [Blo91] that achieves a segmentation accuracy of 93%.

- A novel foreground-background guided local adaptive binarization method is introduced for degraded camera-captured document images (Chapter 6). Unlike using the same values of free parameters in a traditional local adaptive binarization method for both foreground and background regions , the concept of applying different values of free parameters for prior-estimated-foreground and prior-estimated-background regions is introduced in this work. A better OCR-based and image-based performance is achieved as compared to the state-of-the-art local thresholding (Sauvola [SP00]) and global thresholding (Otsu [Ots79]) binarization methods on a subset of the DFKI-I [SB07] dataset.

- A document cleanup using page frame detection method is developed using text line and text/non-text information for camera-captured document images (Chapter 7). A better performance is achieved on the DFKI-I [SB07] dataset as compared to the state-of-the-art Shafait et. al [SvBKB08] and Stamatopoulos et. al [SGK07] page frame detection methods for camera-captured document images.

- A text line based monocular dewarping method is presented for hand-held camera-captured document images (Chapter 8). A better performance is achieved on the DFKI-I [SB07] dataset as compared to the participants' methods in CBDAR 2007 page dewarping contest [SvBKB08].

- An image based performance evaluation method is presented in Chapter 9 for page dewarping algorithms using SIFT features. The performance evaluation metrics measure how well a dewarping algorithm works on both text and non-text regions in warped documents. Unlike other image based performance evaluation methods for page dewarping algorithms, the presented performance evaluation method requires less manual assistance in ground-truth generation process and measures vectorial performance evaluation score that identifies important classes of dewarping errors.

- A high performance layout analysis method is introduced for Arabic document images in Chapter 10. The layout analysis system is developed for segmenting text and non-text elements and extracting text lines in reading order from scanned Arabic script document images written in different languages (like Arabic, Urdu, and Persian) and different styles (like Naskh and Nastaliq). A significantly better text line detection accuracy of 95% ($n = 3,595$ text lines in 45 documents) is achieved as compared to 85% accuracy of the RAST based text line detection method [SHKB06] and 75% accuracy of the X-Y cut method [NSV92].

1.6 Dissertation Overview

An overview of the organization of this thesis that highlights the relationship between different chapters of the thesis is shown in Figure 1.2.

Part I (Chapter 2 and Chapter 3) describes two novel text line extraction methods. The coupled snakelets based text line extraction method, which is described in Chapter 2, is designed for extracting curled text lines from typed-text camera-captured document images. It performs better than previously reported curled text line extraction methods for Latin script document images. In addition to Latin script, the coupled snakelets based text line extraction method can be applied to different scripts like Chinese, Devanagari, and Arabic with specialized script-specific adaptations. In contrast to that, the matched filtering and ridge detection based text line extraction method, which is described in Chapter 3, is a domain-independent, generic text line extraction algorithm that can be equally applied to a large variety of document image classes (typed-text, handwritten and historical documents; document images containing complex scripts like Latin, Chinese, and Arabic; scanned and camera-captured documents, binary and grayscale documents). It achieves better performance as compared to the aggregate results of domain-specific state-of-the-art methods on a large collection of diverse document images.

Part II (Chapter 4 and Chapter 5) describes two different text and non-text segmentation methods. In Chapter 4, a text/non-text segmentation method is presented using connected component based classification, which is faster than pixel based classification methods and does not require a block segmentation beforehand unlike other classification based page segmentation approaches in the literature. In this thesis, this method is tested with Latin script document images, however, it can be adapted for document images containing others scripts as well as for other specialized layout analysis tasks, such as digit and non-digit segmentation [HBSB12], orientation detection [RBSB09], and body-text and side-note segmentation [BAESB12]. In Chapter 5, an improved version of multiresolution morphology based text/non-text segmentation method is presented, which is a domain-independent page segmentation method and it can be equally applied to a diverse collection of document images.

Part III (Chapter 6 - Chapter 9) presents a complete preprocessing of grayscale, warped camera-captured document images for removing their inherent degradations (like non-uniform shadows, noise, perspective and geometric distortions) through binarization, noise cleanup and dewarping steps. All of these preprocessing steps are mainly designed using the generic, domain-independent, text line extraction method (Chapter 3) and/or text and non-text segmentation method (Chapter 5). Part IV (Chapter 10) presents a high performance layout analysis system for complex Arabic and Urdu script document images which is a combination of the presented domain-independent text line extraction (Chapter 3) and text/non-text segmentation (Chapter 5) methods.

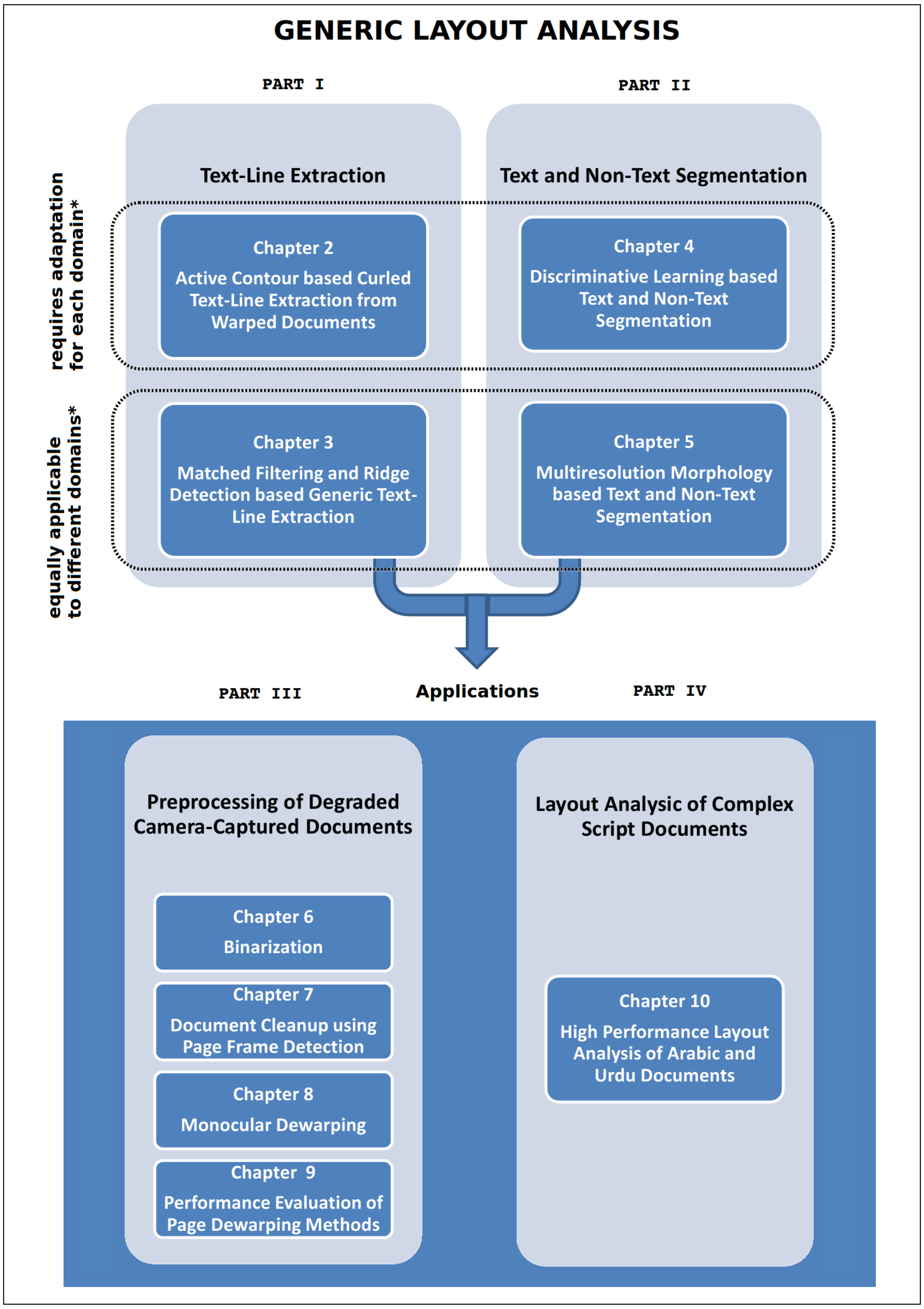

GENERIC LAYOUT ANALYSIS
PART I
PART II
Text-Line Extraction
Text and Non-Text Segmentation
requires adaptation for each domain*
equally applicable to different domains*
Chapter 2 Active Contour based Curled Text-Line Extraction from Warped Documents
Chapter 4 Discriminative Learning based Text and Non-Text Segmentation
Chapter 3 Matched Filtering and Ridge Detection based Generic Text-Line Extraction
Chapter 5 Multiresolution Morphology based Text and Non-Text Segmentation
PART III
Applications
PART IV
Preprocessing of Degraded Camera-Captured Documents
Chapter 6 Binarization
Chapter 7 Document Cleanup using Page Frame Detection
Chapter 8 Monocular Dewarping
Chapter 9 Performance Evaluation of Page Dewarping Methods
Layout Analysic of Complex Script Documents
Chapter 10 High Performance Layout Analysis of Arabic and Urdu Documents

Figure 1.2: A visualization of the structure of this thesis illustrating the relationship between different chapters and their contribution to different areas of generic layout analysis.[Note*: here, domain refers to a specific document class that is defined by a combination of following attributes: digitization method (scanner or camera-captured), intensity value (color, grayscale, or binary), writing style (printed text or handwritten), type of script (like Latin, Chinese, and Arabic). For example a domain of scanned, binarized, printed text, Latin script document images, and a domain of camera-captured, grayscale, handwritten, Arabic script document images.]

Part I

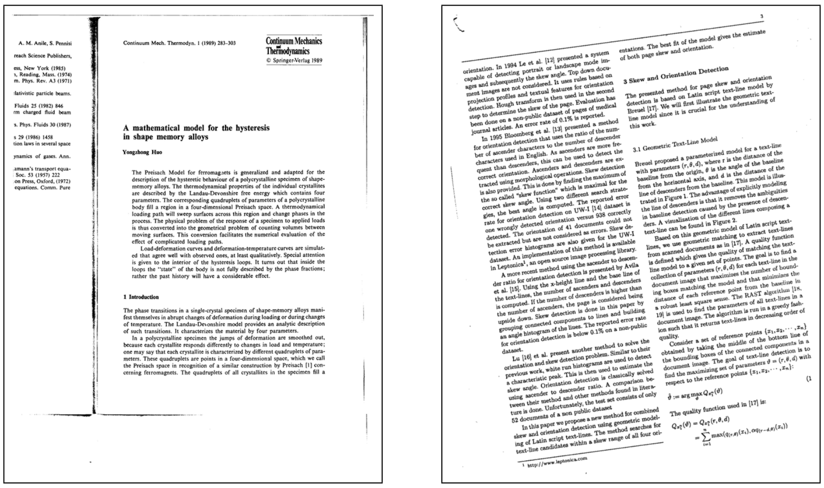
Text Line Extraction

Chapter 2

Coupled Snakelets for Curled Text Line Extraction from Warped Document Images[1]

Summary: *camera-captured, warped document images usually contain curled text lines because of distortions caused by camera perspective view and page curl. Warped document images can be transformed into planar document images for improving optical character recognition (OCR) accuracy and human readability using monocular dewarping techniques. Curled text lines segmentation is a crucial initial step for most of the monocular dewarping techniques. Most of the existing curled text line segmentation approaches are sensitive to geometric and perspective distortions. In this chapter, a novel curled text line segmentation algorithm is presented by adapting active contour (snake). The presented method performs text line segmentation by estimating pairs of x-line and baseline. It estimates a local pair of x-line and baseline on each connected component by jointly tracing top and bottom points of neighboring connected components, and finally each group of overlapping pairs is considered as a segmented text line. The presented algorithm has achieved curled text line segmentation accuracy of above 95% on the DFKI-I (the CBDAR 2007 dewarping contest) dataset, which is significantly better than previously reported results on this dataset.*

[1] This work was published in Bukhari et al. [BSB11e] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. Coupled snakelets for curled text line segmentation from warped document images. International Journal on Document Analysis and Recognition, pages 1-21, October 2011. Copyright ©Springer-Verlag Berlin Heidelberg 2012"*. This chapter is an adapted version of the published work.



A. M. Anile, S. Pennisi
reach Science Publishers,
ess, New York (1985)
ı, Reading, Mass. (1974)
m. Phys. Rev. A3 (1971)
:lativistic particle beams.
Fluids 25 (1982) 846
rm charged fluid beam
s. Phys. Fluids 30 (1987)
s 29 (1986) 1458
tion laws in several space
ynamics of gases. Ann.
zmann's transport equa-
Soc. 53 (1957) 222
on Press, Oxford, (1972)
equations. Comm. Pure

Continuum Mech. Thermodyn. 1 (1989) 283–303

Continuum Mechanics and Thermodynamics
© Springer-Verlag 1989

A mathematical model for the hysteresis in shape memory alloys

Yongzhong Huo

The Preisach Model for ferromagnets is generalized and adapted for the description of the hysteretic behaviour of a polycrystalline specimen of shape-memory alloys. The thermodynamical properties of the individual crystallites are described by the Landau-Devonshire free energy which contains four parameters. The corresponding quadruplets of parameters of a polycrystalline body fill a region in a four-dimensional Preisach space. A thermodynamical loading path will sweep surfaces across this region and change phases in the process. The physical problem of the response of a specimen to applied loads is thus converted into the geometrical problem of counting volumes between moving surfaces. This conversion facilitates the numerical evaluation of the effect of complicated loading paths.

Load-deformation curves and deformation-temperature curves are simulated that agree well with observed ones, at least qualitatively. Special attention is given to the interior of the hysteresis loops. It turns out that inside the loops the "state" of the body is not fully described by the phase fractions; rather the past history will have a considerable effect.

1 Introduction

The phase transitions in a single-crystal specimen of shape-memory alloys manifest themselves in abrupt changes of deformation during loading or during changes of temperature. The Landau-Devonshire model provides an analytic description of such transitions. It characterizes the material by four parameters.

In a polycrystalline specimen the jumps of deformation are smoothed out, because each crystallite responds differently to changes in load and temperature; one may say that each crystallite is characterized by different quadruplets of parameters. These quadruplets are points in a four-dimensional space, which we call the Preisach space in recognition of a similar construction by Preisach [1] concerning ferromagnets. The quadruplets of all crystallites in the specimen fill a

3

orientation. In 1994 Le et al. [12] presented a system capable of detecting portrait or landscape mode images and subsequently the skew angle. Top down document images are not considered. It uses rules based on projection profiles and textual features for orientation detection. Hough transform is then used in the second step to determine the skew of the page. Evaluation has been done on a non-public dataset of pages of medical journal articles. An error rate of 0.1% is reported.

In 1995 Bloomberg et al. [13] presented a method for orientation detection that uses the ratio of the number of ascender characters to the number of descender characters used in English. As ascenders are more frequent than descenders, this can be used to detect the correct orientation. Ascenders and descenders are extracted using morphological operations. Skew detection is also provided. This is done by finding the maximum of the so called "skew function" which is maximal for the correct skew angle. Using two different search strategies, the best angle is computed. The reported error rate for orientation detection on UW-I [14] dataset is one wrongly detected orientation versus 938 correctly detected. The orientation of 41 documents could not be extracted but are not considered as errors. Skew detection error histograms are also given for the UW-I dataset. An implementation of this method is available in Leptonica[1], an open source image processing library.

A more recent method using the ascender to descender ratio for orientation detection is presented by Avila et al. [15]. Using the x-height line and the base line of the text-lines, the number of ascenders and descenders is computed. If the number of descenders is higher than the number of ascenders, the page is considered being upside down. Skew detection is done in this paper by grouping connected components to lines and building an angle histogram of the lines. The reported error rate for orientation detection is below 0.1% on a non-public dataset.

Lu [16] et al. present another method to solve the orientation and skew detection problem. Similar to their previous work, white run histograms are used to detect a characteristic peak. This is then used to estimate the skew angle. Orientation detection is classically solved using ascender to descender ratio. A comparison between their method and other methods found in literature is done. Unfortunately, the test set consists of only 52 documents of a non public dataset

In this paper we propose a new method for combined skew and orientation detection using geometric modeling of Latin script text-lines. The method searches for text-line candidates within a skew range of all four orientations. The best fit of the model gives the estimate of both page skew and orientation.

3 Skew and Orientation Detection

The presented method for page skew and orientation detection is based on Latin script text-line model by Breuel [17]. We will first illustrate the geometric text-line model since it is crucial for the understanding of this work.

3.1 Geometric Text-Line Model

Breuel proposed a parameterized model for a text-line with parameters (r, θ, d), where r is the distance of the baseline from the origin, θ is the angle of the baseline from the horizontal axis, and d is the distance of the line of descenders from the baseline. This model is illustrated in Figure 1. The advantage of explicitly modeling the line of descenders is that it removes the ambiguities in baseline detection caused by the presence of descenders. A visualization of the different lines composing a text-line can be found in Figure 2.

Based on this geometric model of Latin script text-lines, we use geometric matching to extract text-lines from scanned documents as in [17]. A quality function is defined which gives the quality of matching the text-line model to a given set of points. The goal is to find a collection of parameters (r, θ, d) for each text-line in the document image that maximizes the number of bounding boxes matching the model and that minimizes the distance of each reference point from the baseline in a robust least square sense. The RAST algorithm [18, 19] is used to find the parameters of all text-lines in a document image. The algorithm is run in a greedy fashion such that it returns text-lines in decreasing order of quality.

Consider a set of reference points $\{x_1, x_2, \cdots, x_n\}$ obtained by taking the middle of the bottom line of the bounding boxes of the connected components in a document image. The goal of text-line detection is to find the maximizing set of parameters $\vartheta = (r, \theta, d)$ with respect to the reference points $\{x_1, x_2, \cdots, x_n\}$:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{x_1^n}(\vartheta) \qquad (1$$

The quality function used in [17] is:

$$Q_{x_1^n}(\vartheta) = Q_{x_1^n}(r, \theta, d) = \sum_{i=1}^{n} \max(q_{(r,\theta)}(x_i), \alpha q_{(r-d,\theta)}(x_i))$$

[1] http://www.leptonica.com

(a) straight, horizontal text lines

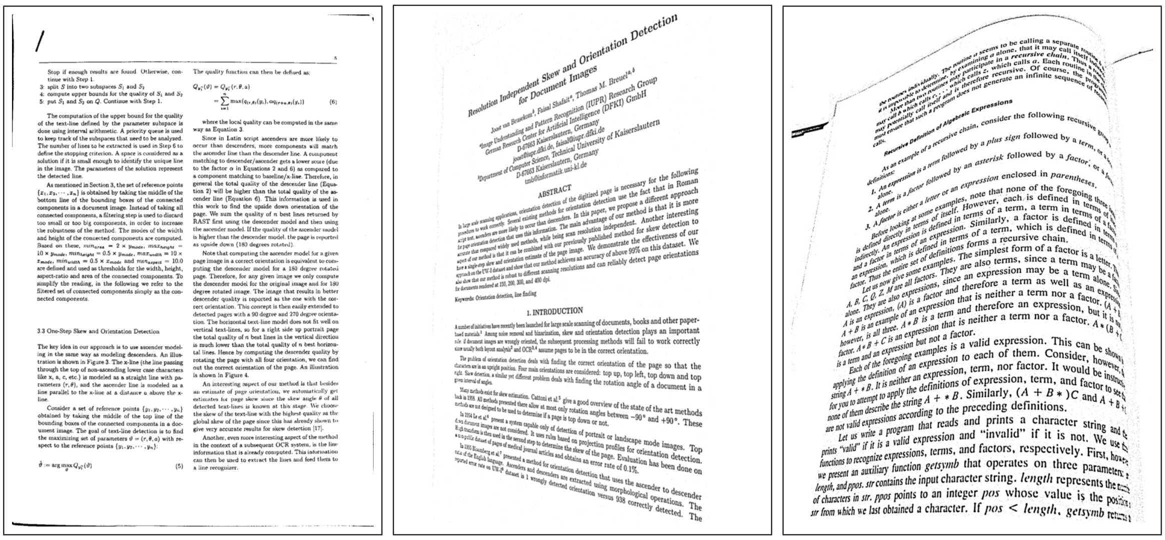
(b) skewed text lines (8° skew angle)

Figure 2.1: Examples of scanned document images.

2.1 Introduction

Text line extraction is one of the important layout analysis steps in document image understanding systems. It is usually applied before feeding text to an optical character recognition (OCR) system. Text lines information can also be used for implementing most of the other document image processing tasks such as binarization [BSB09c], document cleanup [SvBKB08, BSB11a], skew correction [KD05, AKR07, vBSB10], zone segmentation [O'G93], indexing/retrieval based on word and character recognition [MB01], dewarping of camera-captured warped document images [BSB09a] etc. Dewarping is relatively a new document image pre-processing step as compared to others which are mentioned here. It is a process of rectifying camera-captured document images that suffer from perspective and geometric distortions. It can be done either by applying stereo vision techniques [TZZX06] or by using monocular dewarping techniques [BSB09a] – a dewarping technique that is developed for images which are captured by single camera is called a *monocular dewarping* technique. Most of the stat-of-the-art monocular dewarping methods are based on text line extraction.

Documents are traditionally digitized using scanners. When a page containing straight, horizontal text lines is scanned, the resulting scanned image may have horizontal or skewed text lines owing to the paper positioning distortions introduced by the scanning process, as shown in Figure 2.1. These types of document images are referred to as planar document images. There is a large number of state-of-the-art techniques for planar doc-



(a) straight, horizontal text lines (b) skewed text lines due to perspective distortion (c) curled text lines due to perspective and geometric distortions

Figure 2.2: Examples of camera-captured document images.

ument image segmentation [SKB08b], such as projection profile [Gla56, NSV92], Hough transform [FK88], run-length smearing [WCW82], Docstrum [O'G93], branch and bound method [Bre02a]. Most of the commercial and open-source OCR systems work on the assumption that input document images are planar in nature.

Nowadays cameras are available widely at low cost and offer fast, flexible, and non-contact document imaging. These advantages make cameras a potential substitute of scanners for document digitization. Liang et al. [LDL05] presented a brief comparison between scanners and cameras and concluded that camera-based document analysis systems are more flexible than scanner-based systems. The camera-captured document image of a planar document surface is shown in Figure 2.2(a), where the captured image looks like a scanned image.

However, some image degradations come along with the flexibility of using digital cameras for document imaging. For a planar document surface, a digital camera may produce a distorted image due to perspective distortion that arises from the perspective viewpoint of the camera, as shown in Figure 2.2(b). Furthermore, for a thick book page, a digital camera can produce a distorted image because of geometric distortion that is caused by the curled document surface. In such a case, the distorted image is composed of curled text lines with multiple skew angles as shown in Figure 2.2(c). Therefore, the quality of camera-captured document images generally declines due to perspective and/or

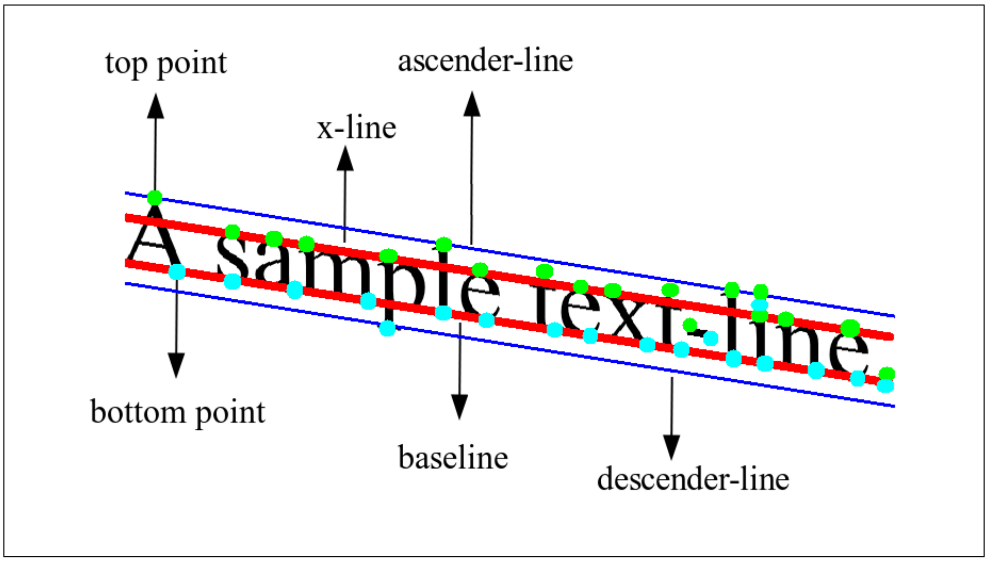


Figure 2.3: The typographic lines and points of a sample text line.

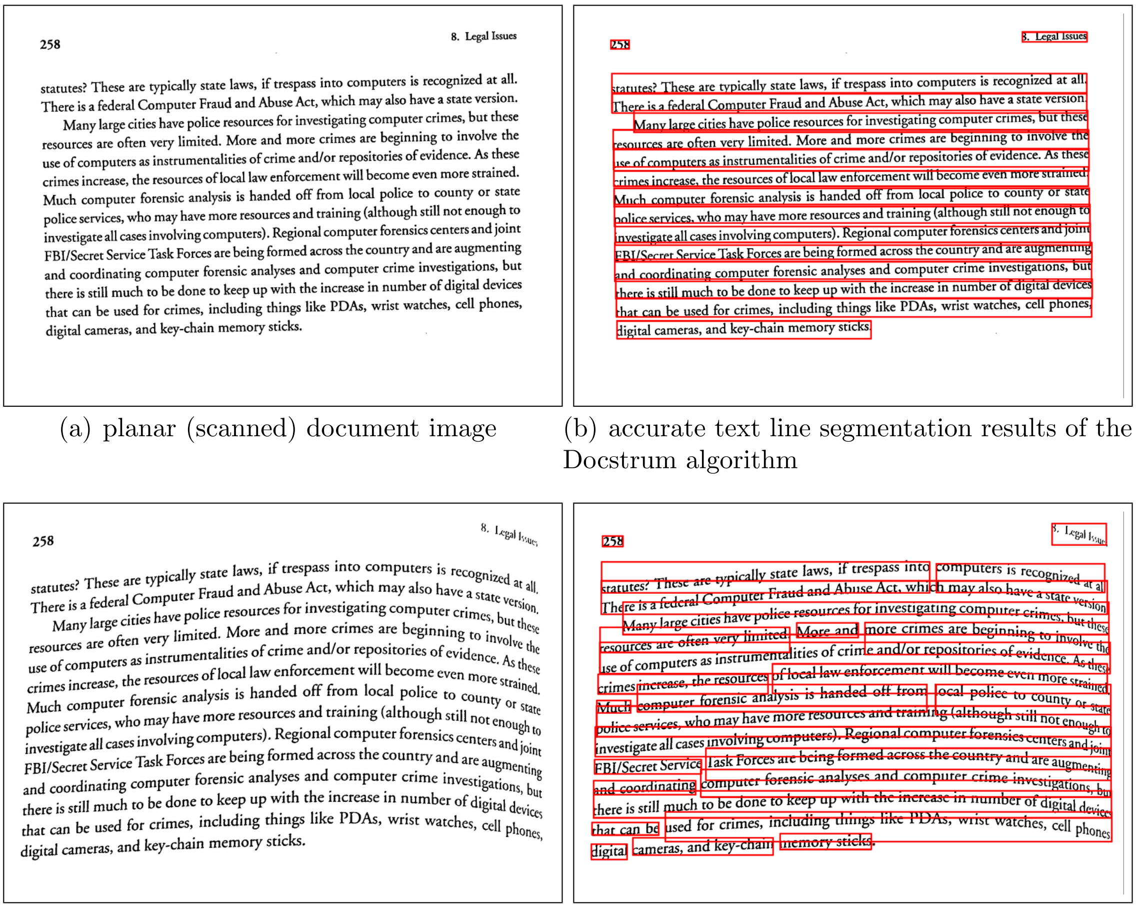
geometric distortions.

Camera-captured document images that contain perspective and/or geometric distortions are usually called *warped document images*. The main problem with a warped document image is that, it reduces not only human readability, but also causes problems for document image processing, like layout analysis and character recognition. Consequently, dewarping is a necessary step in camera-captured document image processing. Most of the monocular dewarping techniques are based on text line information [ZT01, LT06a, GPN07, FWL$^+$07, SB07, BSB09a, OLT$^+$10]. Therefore, text line segmentation is an important step in camera-captured document image processing.

A text line is composed of different typographic lines, i.e. ascender-line, x-line, baseline and descender-line. For each connected component in a text line, its top point is defined here as the coordinate of its top most pixel and bottom point as the coordinate of its bottom most pixel. These terms, being frequently used in the rest of this chapter, are illustrated in Figure 2.3 for a sample text line.

Curled text line detection in warped, camera-captured document images (Figure 2.2(c)) is a challenging problem. Planar document image segmentation techniques, like Docstrum [O'G93], X-Y cut [NSV92] etc., can not be robustly applied for curled text line segmentation [LDD08]. For example, Docstrum is one of the state-of-the-art and widely used planar/straight document image segmentation algorithms, but it performs poorly on warped document image segmentation as shown in Figure 2.4.

In recent years, several curled text line finding methods are proposed in the literature



258 8. Legal Issues

statutes? These are typically state laws, if trespass into computers is recognized at all. There is a federal Computer Fraud and Abuse Act, which may also have a state version.

Many large cities have police resources for investigating computer crimes, but these resources are often very limited. More and more crimes are beginning to involve the use of computers as instrumentalities of crime and/or repositories of evidence. As these crimes increase, the resources of local law enforcement will become even more strained. Much computer forensic analysis is handed off from local police to county or state police services, who may have more resources and training (although still not enough to investigate all cases involving computers). Regional computer forensics centers and joint FBI/Secret Service Task Forces are being formed across the country and are augmenting and coordinating computer forensic analyses and computer crime investigations, but there is still much to be done to keep up with the increase in number of digital devices that can be used for crimes, including things like PDAs, wrist watches, cell phones, digital cameras, and key-chain memory sticks.

258 8. Legal Issues

statutes? These are typically state laws, if trespass into computers is recognized at all. There is a federal Computer Fraud and Abuse Act, which may also have a state version.

Many large cities have police resources for investigating computer crimes, but these resources are often very limited. More and more crimes are beginning to involve the use of computers as instrumentalities of crime and/or repositories of evidence. As these crimes increase, the resources of local law enforcement will become even more strained. Much computer forensic analysis is handed off from local police to county or state police services, who may have more resources and training (although still not enough to investigate all cases involving computers). Regional computer forensics centers and joint FBI/Secret Service Task Forces are being formed across the country and are augmenting and coordinating computer forensic analyses and computer crime investigations, but there is still much to be done to keep up with the increase in number of digital devices that can be used for crimes, including things like PDAs, wrist watches, cell phones, digital cameras, and key-chain memory sticks.

(a) planar (scanned) document image

(b) accurate text line segmentation results of the Docstrum algorithm

258 8. Legal Issues

statutes? These are typically state laws, if trespass into computers is recognized at all. There is a federal Computer Fraud and Abuse Act, which may also have a state version.

Many large cities have police resources for investigating computer crimes, but these resources are often very limited. More and more crimes are beginning to involve the use of computers as instrumentalities of crime and/or repositories of evidence. As these crimes increase, the resources of local law enforcement will become even more strained. Much computer forensic analysis is handed off from local police to county or state police services, who may have more resources and training (although still not enough to investigate all cases involving computers). Regional computer forensics centers and joint FBI/Secret Service Task Forces are being formed across the country and are augmenting and coordinating computer forensic analyses and computer crime investigations, but there is still much to be done to keep up with the increase in number of digital devices that can be used for crimes, including things like PDAs, wrist watches, cell phones, digital cameras, and key-chain memory sticks.

258 8. Legal Issues

statutes? These are typically state laws, if trespass into computers is recognized at all. There is a federal Computer Fraud and Abuse Act, which may also have a state version.

Many large cities have police resources for investigating computer crimes, but these resources are often very limited. More and more crimes are beginning to involve the use of computers as instrumentalities of crime and/or repositories of evidence. As these crimes increase, the resources of local law enforcement will become even more strained. Much computer forensic analysis is handed off from local police to county or state police services, who may have more resources and training (although still not enough to investigate all cases involving computers). Regional computer forensics centers and joint FBI/Secret Service Task Forces are being formed across the country and are augmenting and coordinating computer forensic analyses and computer crime investigations, but there is still much to be done to keep up with the increase in number of digital devices that can be used for crimes, including things like PDAs, wrist watches, cell phones, digital cameras, and key-chain memory sticks.

(c) curled (camera-captured) document image

(d) text line segmentation failures of the Docstrum algorithm

Figure 2.4: Text line extraction results of the Docstrum [O'G93], a state-of-the-art planar document image segmentation technique, for a planar document image and a curled document image. For the curled document image, the Docstrum produced a lot of segmentation errors and failed to extract text lines.

[GA99, ZT01, LT02, LT06a, GPN07, FWL$^+$07, BSB08, BSB09b, BBS09, OLT$^+$10] mainly in the context of monocular dewarping approaches. Most of these methods use nearest neighbor based grouping of connected components for detecting text lines, but these methods usually produce under-segmentation failures in the presence of high degree of curl/skew in document images. Another general observation about the existing approaches is that, they estimate x-line and baseline pairs after segmenting text lines by using regression over top and bottom points of segmented text lines, respectively, that may result in inaccurate estimation. A brief overview of these methods is given in Section 2.2.

In this chapter, a curled text line segmentation method applying active contours (snakes) [KWT88] is presented. Here, snakes are adapted for estimating a local pair of x-line and baseline at each connected component in a document image, where each connected component may represent a character, a broken piece of a character or a bunch of joined characters. Afterwards, each group of overlapping pairs is considered as a segmented text line that also provides x-line and baseline information of the segmented text line. The presented curled text line segmentation method is less sensitive to high degree of curl and skew in document images and produces better segmentation results than existing curled text line segmentation approaches as shown in the performance evaluation section (Section 2.5). Furthermore, unlike other approaches, the presented algorithm performs segmentation of text lines and estimation of their x-lines and baselines together at the same time, which also gives more precise x-lines and baselines information than regression based methods.

The presented text line detection algorithm is designed for hand-held camera-captured images of isolated or bound pages that contain straight text lines of typed-text Latin script. As mentioned earlier, hand-held camera-captured images usually suffer from perspective distortion (due to camera view angle) and/or geometric distortion (due to curled document surface). Therefore, straight text lines in documents (as shown in Figure 2.2(a)) are transformed into skewed and/or curled text lines in camera-captured images (as shown in Figure 2.2(b) and 2.2(c), respectively). The presented algorithm can handle skew and/or curl angle up to ±45°. It can also deal with variable character sizes within a document image with a minimum (average) character size (length/height) of 10 pixels. The presented algorithm can also work in the presence of figures, tables, equations, and noise.

The rest of this chapter is organized as follows. A brief description of existing curled text line segmentation approaches is presented in Section 2.2. The presented coupled snakelets model is described in Section 2.3. Implementation details of the reported curled text line segmentation algorithm applying coupled snakelets model are presented in Section 2.4. Performance evaluation and experimental results are given in Section 2.5, followed by a conclusion in Section 2.7.

2.2 Related Work

Several curled text line segmentation approaches are proposed in the literature [GA99, ZT01,LT02,LT06a,GPN07,FWL+07,BSB08,BSB09b,BBS09,OLT+10] for camera-captured warped/curled document images. Most of these curled text lines extraction approaches are mainly proposed as a pre-processing step of monocular dewarping of camera-captured document images. Some of these approaches are briefly discussed here.

Goto and Aso [GA99] proposed a text line segmentation method for a document image that may contain curved text lines with arbitrary orientations. Their algorithm is based on linking of locally linear components. First, the *primitive rectangles* are estimated from the connected components of a document image. Then, these rectangles are grouped together on the basis of a predefined criteria to achieve segmented text lines.

Zhang et al. [ZT01] introduced a curled text line finding algorithm using *box-hand* [SPC97] approach. In this algorithm, connected components are first combined to form words using nearest neighbor analysis. Then, a pair of left and right rectangular box-hands are attached with each word. Each chain of overlapping words is considered as a segmented text line.

Loo and Tan [LT02] proposed a word and sentence extraction method for a document image that may contain a wide variety of text line orientations and layouts. Their algorithm is based on the *irregular pyramid structure* that help in merging characters into words and then words into sentences.

Lu and Tan [LT06a] proposed a curled text line segmentation approach, where top and bottom points of connected components are first estimated by using morphological operations. Then, text line detection is performed by tracking either top or bottom points. For a point, left and right nearest neighbors are searched and this process is repeated for neighbors until no more neighbor is found. The same process is repeated for remaining points. Each group of connected components is considered as a segmented text line.

Gatos et al. [GPN07] proposed a smearing based curled text line detection algorithm. In this approach, horizontal run-length smearing is used to combine characters into words. The height corresponding to the maximum peak of connected components' height histogram (H) is used as a threshold for smearing. After smearing, left and right neighboring words are searched for each word within a limited distance (D) such that $D < 5H$, and the search is repeated until no more neighbors are found. The same process of word grouping is repeated for the remaining words. Each group of words is referred to as a segmented text line. It has been observed that, the algorithm works well on clean document images where the parameter H can be reliably estimated. However, in the presence of salt-and-pepper noise or a large number of broken characters, the estimated value of H is usually too small. This badly affects the performance of their algorithm. Here, a slight modification is proposed in this algorithm such that if H is less than a predefined threshold (T), all values less than T are removed from the height histogram and the height corresponding to the maximum peak of the remaining histogram is selected as H. The value of T can be set equal to the mean height of a character in a targeted dataset of document images.

Fu et al. [FWL$^+$07] proposed a curled text line segmentation technique using nearest-neighbor analysis over text lines portions. In this approach, portions of text lines are first estimated using wavelet based enhancement technique [HLW03]. These portions are then grouped together using nearest-neighbor approach, where each group is considered as a segmented text line.

Oliveria et al. [OLT$^+$10] proposed a rule-based method for warped text line segmentation. In this algorithm, a *same-size* nearest-neighbor is found for each connected component. All pairs are added into a *priority-queue.* Then for each pair, nearest-neighbors are iteratively searched in both right and left directions using *moving-window* analysis which holds the following conditions: *same-size*, *smaller than window*, in-between *parallel line with offset* and distance is less than *maximum distance between letters.* Each group of connected components is referred to as a text line. Afterwards, detected text lines are further improved by using the following steps. Each text line is selected one by one in a decreasing text line's length priority order, and *upper and lower text lines* are searched for its each component. Two upper and/or lower text lines are merged together if they satisfy some predefined thresholding criteria. The final step is the removal of those text lines that contain connected components less than some predefined threshold or contain connected component on 10% of image border. Together with some predefined threshold, all of the above italicized terms are defined using some empirically selected values.

Most of the above curled text lines segmentation methods (like [GA99, ZT01, LT02, LT06a, GPN07]) are based on grouping of connected components using some predefined nearest neighbor criteria. The main limitation of a nearest neighbor based curled text line finding method is that, it can only handle a moderate skew/curl angle, and it produces a number of over- and under-segmentation errors under a high degree of skew/curl. In contrast to nearest neighbor based text line finding methods, the proposed method comparatively produces less number of undersegmentation errors. The curled text line segmentation method of Oliveria et al. [OLT$^+$10] performs well even in the presence of high degree of skew/curl, but it contains a large number of free parameters. The proposed method contains around six free parameters, where most of them are non-sensitive. In section 2.5, the performance of the presented coupled snakelets based curled text line segmentation method is compared with: i) nearest-neighbors (Gatos et al. [GPN07]), ii) baby-snakes (Bukhari et al. [BSB08]), and iii) rule-based (Oliveria et al. [OLT$^+$10]), and iv) Docstrum [O'G93]. The main reason of selecting these method for comparison is to show the performance of different categories of curled text line detection techniques on a common dataset.

2.3 Coupled Snakelets for Curled Text Line Segmentation

Coupled snakelets model for curled text line segmentation is based on active contour (snake) [KWT88], which is one of the state-of-the-art image segmentation techniques in computer vision. First, a brief description of basic active contour (snake) model is presented in Section 2.3.1 and then salient features of the presented coupled snakelets model are explained in Section 2.3.2.

2.3.1 Review of Active Contours (Snake) Model

Active contour (snake) was introduced by Kass et al. [KWT88] for image segmentation. A snake is a closed curve of points $S(s) = [x(s), y(s)]$, where $s \in [0, 1]$, that moves through the spatial domain of an image to minimize the energy function (E):

$$E = \int_0^1 E_{int}\{S(s)\} + E_{ext}\{(S(s)\}ds \tag{2.1}$$

$$E = \int_0^1 \frac{1}{2}[\alpha\{S'(s)\} + \beta\{S''(s)\}] + E_{ext}\{S(s)\}ds \tag{2.2}$$

The snake slithers towards a targeted object under the influence of internal energy (E_{int}) and external energy (E_{ext}), where the internal energy is estimated from the snake points and the external energy is computed from image contents. The internal energy tries to keep the snake's points close to each other and the external energy tries to move the snake towards the boundary of a targeted object. These internal and external energies are defined in such a way that, the snake deforms iteratively towards a targeted object and finally wraps around the object's boundary. Internal energy is further decomposed into two factors: i) $S'(s)$ (first order derivative of $S(s)$) represents tension within snake's points, ii) $S''(s)$ (second order derivative of $S(s)$) represents rigidity within snake's points. The weighted parameters α and β are used for controlling snake's tension and rigidity, respectively. The snake remains more rigid for a big value of β than a small value.

The weight of the external energy can also be controlled by a free parameter that can take a value in between 1 to 0. In this chapter, this parameter is defined as γ. The Equation 2.1 can be rewritten as:

$$E = \int_0^1 E_{int}\{S(s)\} + \gamma E_{ext}\{(S(s)\}ds \tag{2.3}$$

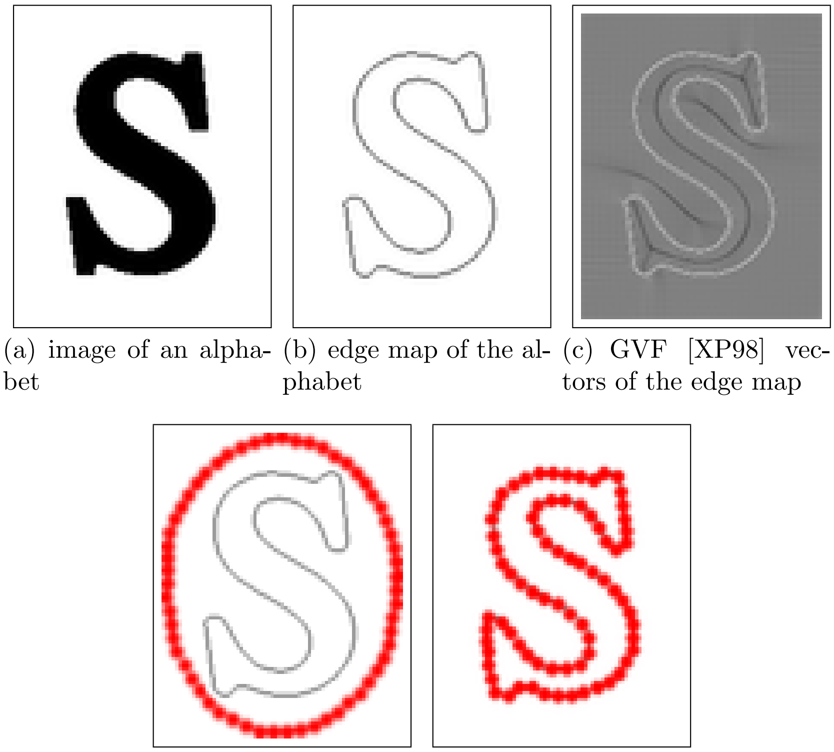
In general, external energy can be calculated from the edge map of an image by using gradient, Gaussian of gradient or Gradient Vector Flow (GVF) [XP98]. The gradient vectors or Gaussian of gradient vectors have large magnitudes only in the immediate vicinity of the edges; but these vectors are zero in homogeneous regions where image data is nearly constant. Therefore, the range of gradient or Gaussian of gradient based external energy is limited and it only exists near the edges. In such a case, manual assistance is required for initializing snake near a targeted object. In contrast to these types of external energies, GVF is calculated by using the computational diffusion of gradient vectors iteratively, where it maintains the gradient vectors near the edges and at the same time extends these vectors farther away from the edges into homogeneous regions. Therefore, GVF covers a large range of energies (gradient vectors) around edges that helps to diverge the snake towards the boundary of a targeted object even if it is initialized far away from the object. In such a case, manual assistance is not required for snake initialization.

A simple example to illustrate the basic concept of object's boundary detection using active contour (snake) is illustrated in Figure 2.5. Traditional active contour mechanism of image segmentation, which is illustrated in Figure 2.5, can not be directly applied for text lines segmentation in document images as shown in the Figure 2.6. In this chapter, active contour (snake) is adapted for text line segmentation. For this purpose, a *coupled snakelets* model is introduced that is derived from active contour (snake) model. Detailed discussion about coupled snakelets model is given in the next section (Section 2.3.2).

2.3.2 Coupled Snakelets Model

In this section, some relevant features are introduced and added in the basic active contour (snake) model [KWT88] for making it applicable for text line segmentation problem. Here, the adapted active contour (snake) model is referred to as *coupled snakelets* model. Some salient features of coupled snakelets model are explained below.

- **Open-Curve Snake:** A text line can be represented by a close-curve boundary around it or simply by typographic lines, for example x-line, baseline, ascender-line



(a) image of an alphabet

(b) edge map of the alphabet

(c) GVF [XP98] vectors of the edge map

(d) initial close-curve snake

(e) deformed snake

Figure 2.5: An example of image boundary detection using active contour (snake) model.

or descender-line. Traditional close-curve snakes, as show in Figure 2.5(d), can not be used to find close-curve boundary of a text line due to the close proximity of a text line to its neighboring text lines. A concept of open-curve snakes is introduced for text line segmentation. In contrast to a close-curve snake, an open-curve snake is a straight line snake. For example, a group of open-curve snakes are shown in Figure 2.7(b).

- **Multiple Snakes:** Each text line of a document image consists of several connected components. Furthermore, there are many text lines in a single document image. Coupled snakelets model uses multiple snakes to cope with this problem. These snakes are deformed independently with respect to one another.

- **Automatic Initialization of Pair of Snakes:** Coupled snakelets model uses automatic initialization of snakes over connected components in a document image.

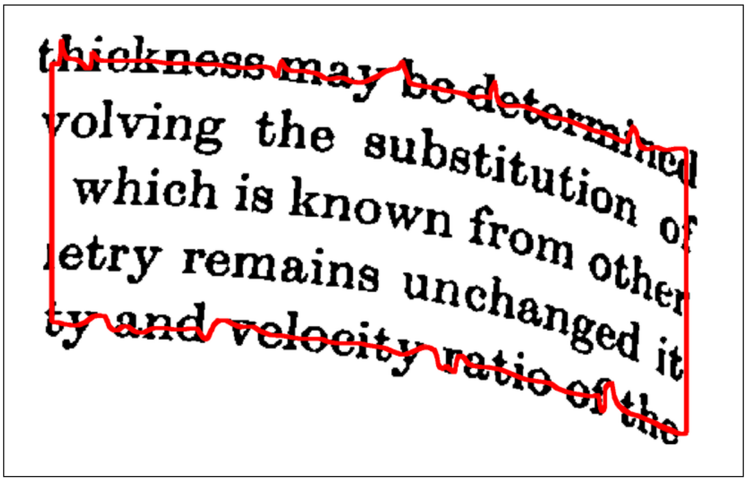

thickness may be determined
volving the substitution
which is known from of
etry remains unchanged other
ty and velocity ratio of the it

Figure 2.6: A traditional process of active contour (snake) based image segmentation, as shown in Figure 2.5, can not be applied directly for text line segmentation in document images.

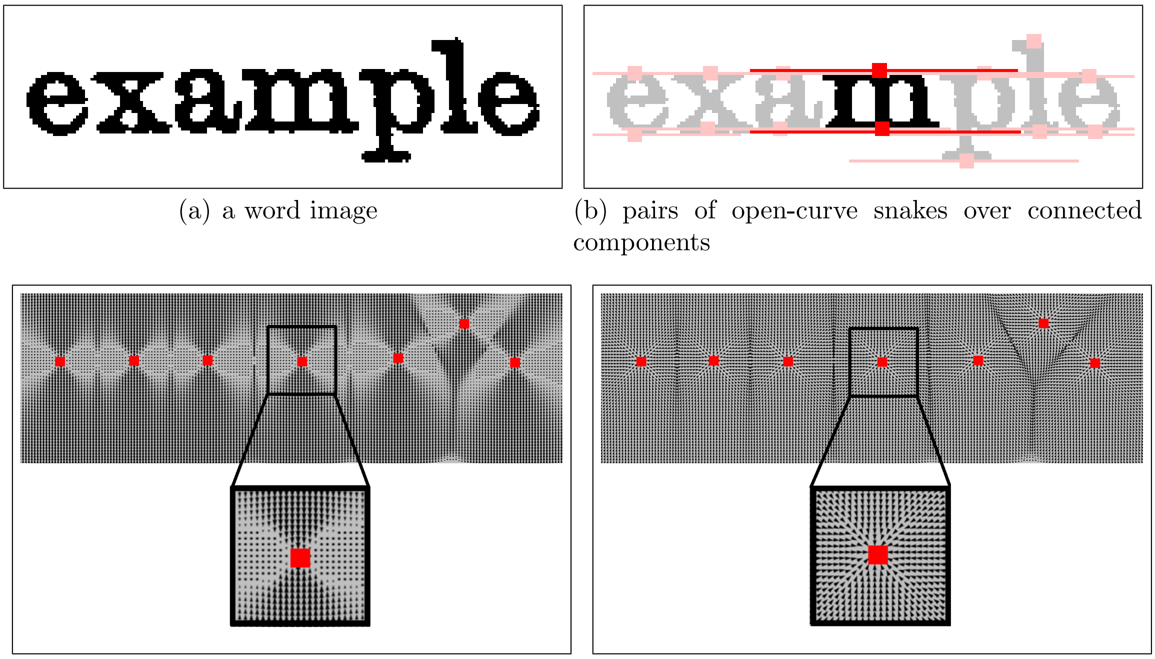
For each connected component, a pair of open-curve snakes is initialized over it such that one snake is initialized at its top point and another one at its bottom point.

Multiple open-curve pairs of snakes that are initialized automatically over connected components of Figure 2.7(a) are shown in Figure 2.7(b).

- **External Energy Calculation from Discrete Points:** Instead of using an edge map of connected components, coupled snakelets model uses discrete top or bottom points of connected components for GVF (external energy) calculation.

- **Deformation of Snakes in Targeted Direction:** In Latin scripts, text lines are usually horizontal in nature. Neighboring snakes can be joined together for segmenting text lines by deforming them in vertical direction only. Coupled snakelets model deforms a snake only in vertical direction such that x-coordinates of the snake points are kept static and y-coordinates of the snake points are deformed with respect to the vertical components of GVF of discrete points.

For comparison, GVF vectors that represent only vertical components and both vertical and horizontal components are shown in Figure 2.7(c) and Figure 2.7(d), respectively. In coupled snakelets, an open-curve snake is deformed using only ver-



(a) a word image

(b) pairs of open-curve snakes over connected components

(c) vectors corresponding to vertical components of GVF of top points

(d) vectors corresponding to vertical and horizontal components of GVF of top points

Figure 2.7: Coupled Snakelets Features: a) an example image, b) multiple open-curve snakes are initialized automatically over the top and bottom points of connected components, which are shown here by square symbol (red color), c) vertical components of GVF vectors that were calculated using top points (shown in red color, square symbol) of connected components; it is visible in the enlarged portion that each of theses vectors either points downwards or upwards with some amplitude. d) both vertical and horizontal components of GVF vectors for top points, it is visible in the enlarged portion that theses vectors are pointing towards all directions. Both of these images (c and d) are shown here for illustration.

tical components of GVF vectors.

- **Evolving Snakes:** Coupled snakelets model introduces a concept of *evolving snake*. As mentioned earlier, a pair of snakes is initialized at each connected component. For a connected component, both of its top and bottom snakes are deformed independently with respect to the top and bottom points, respectively, in evolving fashion which is described as follows. First, a small rectangular region that centered around the connected component is selected. Then, the top (bottom) snake is deformed with respect to the vertical components of GVF that is calculated from the top (bottom) points of connected components inside the selected area. After the first cycle of deformation, a second cycle is started such that the top (bottom) snake's length and the selected area are increased before deformation. The same process is repeated for a few number of deformation cycles. The main motivation behind this approach is that there exist a number of left, right, top and bottom neighboring connected components around a particular connected component in a document image. For a small rectangular region around the connected component, almost all of these points belong to the same text lines to which the connected component belongs. For a big rectangular region, some of these points belong to the same text line to which the connected component belongs, and others belong to the neighboring top and/or bottom text lines. If all of these top (bottom) points within a big rectangular region are initially used for external energy calculation, the top (bottom) snake may deform in a wrong upward or downward direction and may cause segmentation failures. In contrast to that, evolving snake criteria expands the top (bottom) snake in a corresponding text line direction even in the presence of a high degree of skew and/or curl and prevents segmentation failures.

- **Weighted-Coupled Pair of Snakes:** Two or more snakes can also be simultaneously used for a image segmentation such that each of them is deformed independently and then all of them are adjusted before further deformation steps [GN97, HH03, WRS$^+$09]. This type of snakes are referred to as coupled snakes. The presented coupled snakelets model adapts this idea for curled text line segmentation. Here two general observations of Latin script document images are exploited for defining the presented coupled snakelets idea:

  - **Observation #1:** for a text line in Latin scripts, where ascenders are more frequent than descenders [vBSB10], majority of the bottom points of the connected components lie over its baseline as compared to the top points of connected components over its x-line.
  - **Observation #2:** within a text line, the same distance exists between the pair of its x-line and baseline, as long as the complete text line has the same font.

For a connected component, a pair of evolving snakes is first initialized over it. Then, on the basis of observation #1, the top snake is deformed using a small weighted percentage of the external energy of top points within a initial selected region, and the bottom snake is deformed using a comparatively large weighted percentage of bottom points. After each deformation step, the top and bottom snakes in the pair are coupled such that, first the average distance is calculated from the distances between corresponding pairs of points in the top and bottom snakes and then each corresponding pair of points in these snakes is updated to make its distance equal to the average distance. This type of coupling between the top and bottom evolving snakes is done on the basis of observation #2. In this way, the top and bottom snakes estimates a local pair of x-line and baseline at the connected component after a few number of deformation cycles. The same process is repeated for other connected components as well, and each group of overlapping pairs of snakes represents a segmented text line.

2.4 Curled Text Line Segmentation Algorithm

The steps of the presented curled text line segmentation algorithm on the basis of above described coupled snakelets model are shown in Figure 2.8. Each of these steps is described here in detail.

Binarization and Noise Cleanup

An input grayscale camera-captured document image is first binarized by using adaptive thresholding technique. The binarization method is defined as follows: "for each pixel, the background intensity $B(p)$ is defined as the 0.8-quantile in a window shaped surrounding; the pixel is then classified as background if its intensity is above this constant fraction of $B(p)$". Note that, this binarization scheme is the same as used in [ULB05]. An example binarized image is shown in Figure 2.9(a). The binarized document image may contain marginal as well as salt-and-pepper noise. A heuristic based noise cleanup process is applied as follows. Let H_{doc} and W_{doc} represent the height and width of the document image, respectively; H_{avg} and W_{avg} represent the mean height and width of connected components, respectively; σ_H and σ_W represent standard deviation of heights and widths of connected components, respectively. A connected component, whose height and width are represented by H_{cc} and W_{cc}, respectively, is removed as a large noisy component if any of the conditions specified in Equation 2.4 is true or as a small noisy component if the

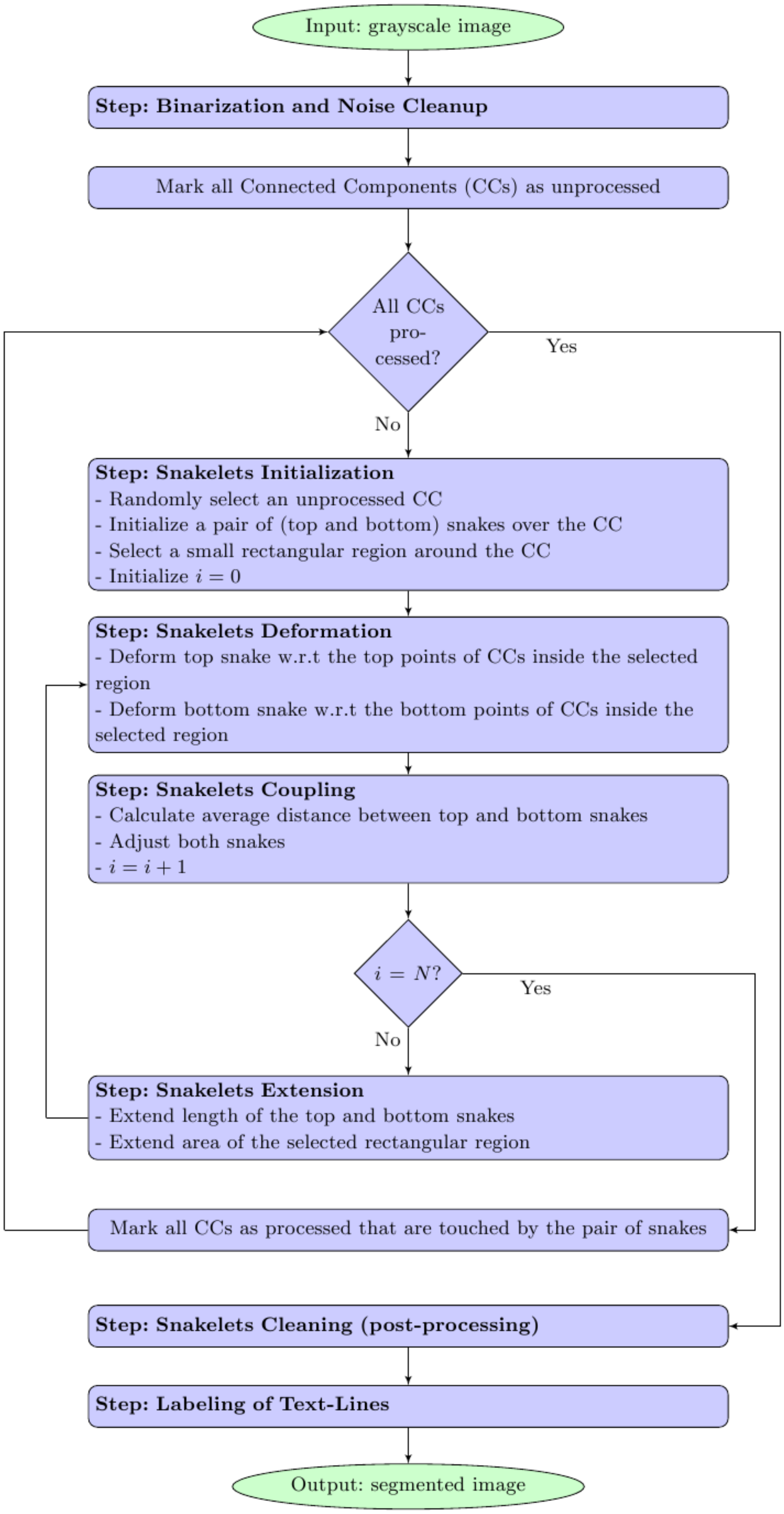
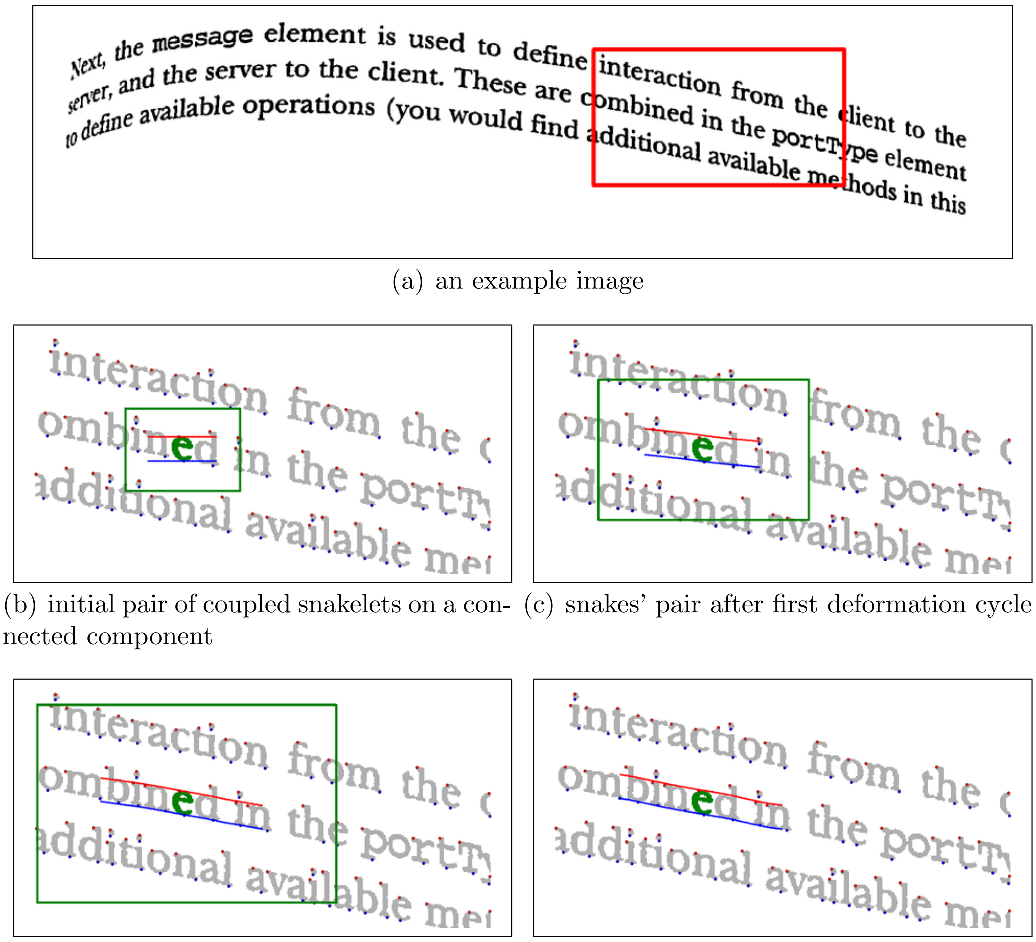

Input: grayscale image
Step: Binarization and Noise Cleanup
Mark all Connected Components (CCs) as unprocessed
All CCs processed?
Yes
No
Step: Snakelets Initialization
- Randomly select an unprocessed CC
- Initialize a pair of (top and bottom) snakes over the CC
- Select a small rectangular region around the CC
- Initialize $i = 0$
Step: Snakelets Deformation
- Deform top snake w.r.t the top points of CCs inside the selected region
- Deform bottom snake w.r.t the bottom points of CCs inside the selected region
Step: Snakelets Coupling
- Calculate average distance between top and bottom snakes
- Adjust both snakes
- $i = i + 1$
$i = N$?
Yes
No
Step: Snakelets Extension
- Extend length of the top and bottom snakes
- Extend area of the selected rectangular region
Mark all CCs as processed that are touched by the pair of snakes
Step: Snakelets Cleaning (post-processing)
Step: Labeling of Text-Lines
Output: segmented image

Figure 2.8: The coupled Snakelets based curled text line segmentation algorithm.


Next, the message element is used to define interaction from the client to the
server, and the server to the client. These are combined in the portType element
to define available operations (you would find additional available methods in this

(a) an example image

(b) initial pair of coupled snakelets on a connected component

(c) snakes' pair after first deformation cycle

(d) snakes' pair after second deformation cycle

(e) snakes' pair after the third (last) deformation cycle

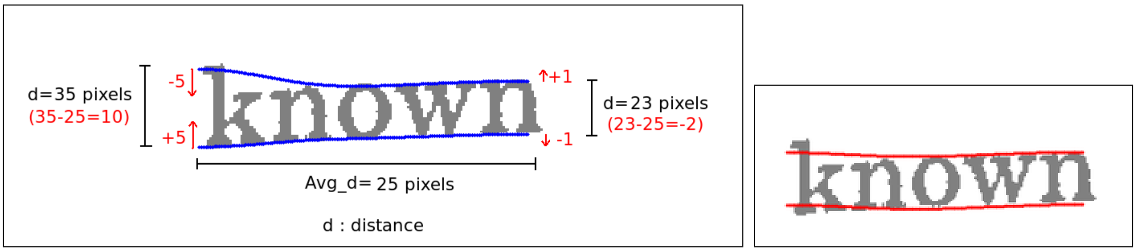
Figure 2.9: Coupled Snakelets Features: illustration of evolving and weighted coupling nature of a pair of snakes ($N = 3$). Note that the snakelets were not misled by the ascenders "b" and "d" due to their coupling.

condition specified in Equation 2.5 is true:

$$\begin{array}{ccc} H_{cc} > (0.1 \times H_{doc}) & \text{or} & H_{cc} > (7 \times \sigma_H) \\ W_{cc} > (0.1 \times W_{doc}) & \text{or} & W_{cc} > (7 \times \sigma_W) \end{array} \tag{2.4}$$

$$(H_{cc} \times W_{cc}) < (\frac{1}{3} \times H_{avg} \times W_{avg}) \tag{2.5}$$

After removing noisy connected components, let mean width and mean height of all the remaining connected components be represented by W and H, respectively.



(a) before snakelets coupling

(b) after snakelets coupling

Figure 2.10: An illustration of snakelets coupling procedure. a) a pair of snakelets, b) the points of the top and bottom snakelets are adjusted with respect to the average distance between them.

Snakelets Initialization

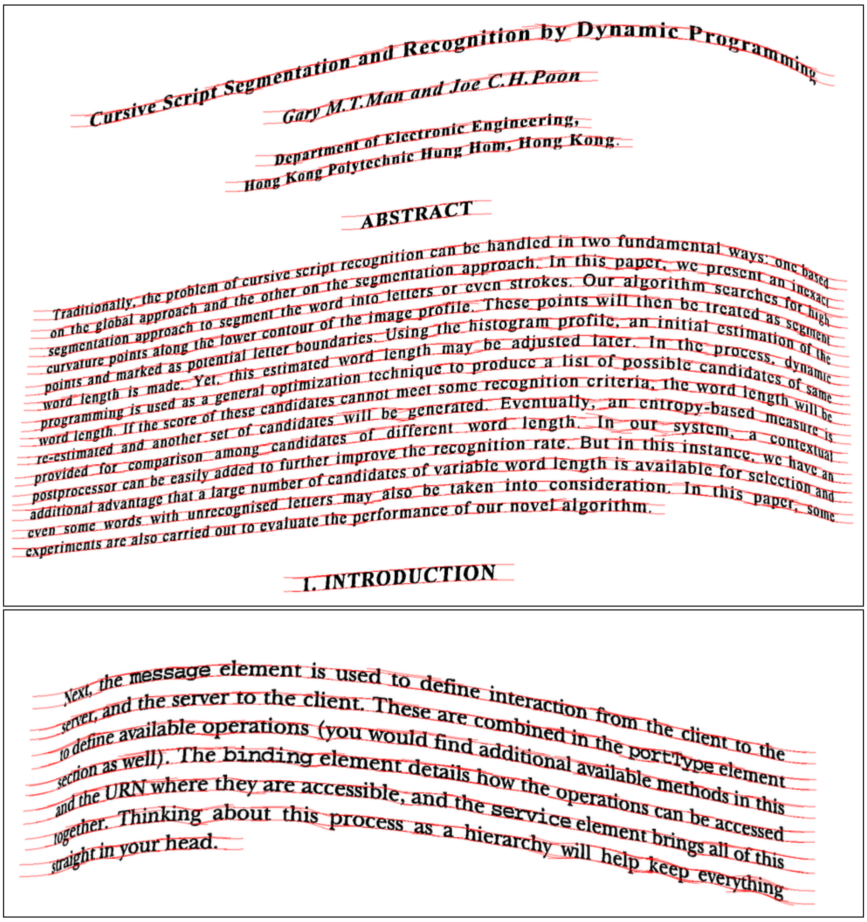
All the connected components of the binarized document image are marked as unprocessed. Then, a connected component is selected randomly. A pair of horizontal open-curve snakes is initialized over the selected connected component, such that one snake is initialized at its top point and another one at its bottom point. By keeping the connected component at center, a small rectangular region is selected around it. The initial length of the snakes (L) and the size of the rectangular region ($W_R \times H_R$) are selected in such a way that both of them cover a few neighboring connected components around the selected connected component. Let H_{cc} and W_{cc} represent the height and width of the connected component, respectively. L, W_R, and H_R are defined as:

$$\begin{aligned} L &:= W_{cc} + 2 \times W \\ W_R &:= W_{cc} + 4 \times W \\ H_R &:= H_{cc} + 2 \times H \end{aligned} \tag{2.6}$$

The main reason of selecting the width greater than the height of the rectangular region is that, text lines in a document are usually horizontal in nature. An example of an initial pair of snakes and a selected rectangular region for a connected component is shown in Figure 2.9(b).

Snakelets Deformation

Gradient vector flow (GVF) is calculated by using the top points of all connected components inside the selected region around the connected component. Then the top snake is deformed by using the vertical components of the GVF with $\gamma/2$ (Equa-



Cursive Script Segmentation and Recognition by Dynamic Programming

Gary M.T.Man and Joe C.H.Poon

Department of Electronic Engineering,
Hong Kong Polytechnic Hung Hom, Hong Kong.

ABSTRACT

Traditionally, the problem of cursive script recognition can be handled in two fundamental ways: one based on the global approach and the other on the segmentation approach. In this paper, we present an inexact segmentation approach to segment the word into letters or even strokes. Our algorithm searches for high curvature points along the lower contour of the image profile. These points will then be treated as segment points and marked as potential letter boundaries. Using the histogram profile, an initial estimation of the word length is made. Yet, this estimated word length may be adjusted later. In the process, dynamic programming is used as a general optimization technique to produce a list of possible candidates of same word length. If the score of these candidates cannot meet some recognition criteria, the word length will be re-estimated and another set of candidates will be generated. Eventually, an entropy-based measure is provided for comparison among candidates of different word length. In our system, a contextual postprocessor can be easily added to further improve the recognition rate. But in this instance, we have an additional advantage that a large number of candidates of variable word length is available for selection and even some words with unrecognised letters may also be taken into consideration. In this paper, some experiments are also carried out to evaluate the performance of our novel algorithm.

1. INTRODUCTION

Next, the message element is used to define interaction from the client to the server, and the server to the client. These are combined in the portType element to define available operations (you would find additional available methods in this section as well). The binding element details how the operations can be accessed and the URN where they are accessible, and the service element brings all of this together. Thinking about this process as a hierarchy will help keep everything straight in your head.

Figure 2.11: A few example images of curled text line segmentation using coupled snakelets algorithm where a pair of snakes estimates a local pair of x-line and baseline and a group of overlapping pairs of snakes represents a segmented text line with its x-line and baseline information. The presented method can handle different fonts sizes within a document image (top figure) and high degree of different directions of curls/skews.

tion 2.3). Similarly bottom snake is deformed by using the vertical component of the GVF with γ, where the GVF is calculated form the bottom points of all connected components within the selected region.

Snakelets Coupling

The top and bottom snakes are composed of the same number of points with similar values of x-coordinates. For each common value of x-coordinate of the top and bottom snakes, absolute distance is calculated from the corresponding values of y-coordinates. Then, average distance is computed. Now, for each common value of x-coordinate of both snakes, the corresponding values of y-coordinates are increased or decreased proportionally such that the distance between them becomes equal to the average distance. Snakelets coupling procedure is illustrated in Figure 2.10.

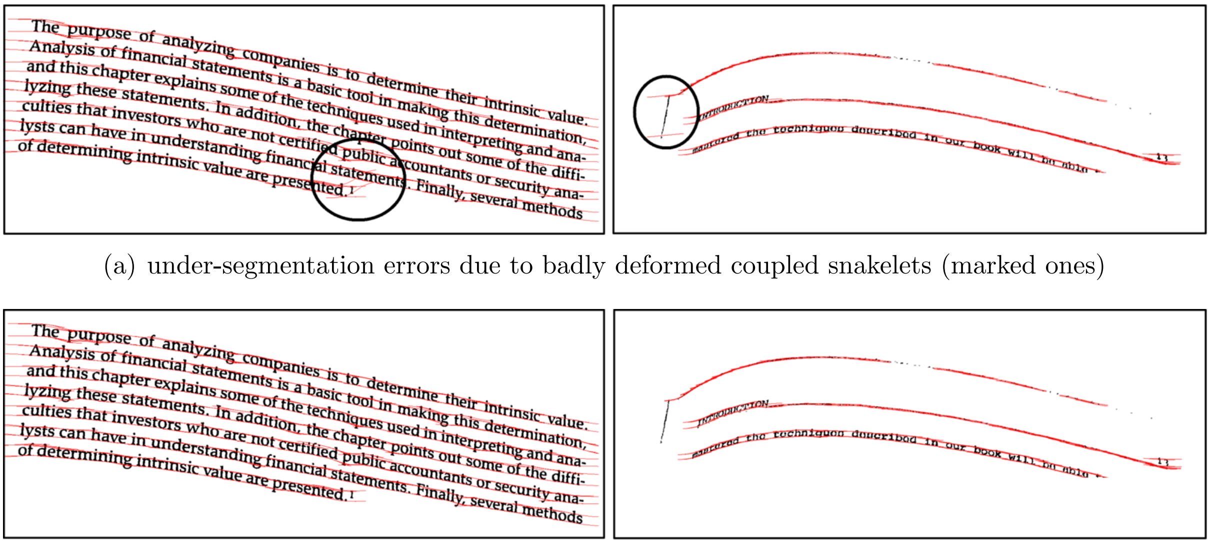
Snakelets Extension

First, the average slope of the pair of snakes is calculated. Then, each of the top and bottom snake is extended by a length equal to the average width (W) from both left and right sides and the slope of these extended lengths is kept the same as the average slope. Similarly, the rectangular region around the connected component is extended, such that its width and height become twice as big compared to its previous width and height after extension.

After snakelets deformation, coupling and extension steps, the first deformation cycle of the snakes' pair is completed, which is shown in Figure 2.9(c). The pair of snakes is further processed by a few number (N) of deformation cycles. We empirically found that three iterations of snakelets extension are sufficient for printed Latin script documents. The results of coupled snakelets for two more deformation cycles are shown in Figure 2.9(d) and Figure 2.9(e), respectively.

The pair of snakes approximates a local pair of x-line and baseline on the connected component, as shown in Figure 2.9(e). Now all the connected components that are overlapped/touched by the pair of snakes are marked as processed. Afterwards, the same process is repeated for another unprocessed connected component and is continued until no more unprocessed connected components are left. Some example images with all computed pairs of coupled snakelets are shown in Figure 2.11. In these example images, each group of overlapping/touching pairs of coupled snakes can be considered as a segmented text line. In these example images, it is also visible that the presented method can handle a high degree of curl/skew and different font sizes within a document image.

Coupled snakelets based text line segmentation algorithm may also cause undersegmentation failures. Some examples of undersegmentation failures of the presented algorithm are shown in Figure 2.12(a). Such type of segmentation failures occur because of some *badly deformed* pairs of snakes, as marked in Figure 2.12(a). Here, a badly deformed pair of snakes is defined as follows: a pair of snakes that is not correct with respect to the estimation of its corresponding local x-line and baseline pairs and is not uniform with



(a) under-segmentation errors due to badly deformed coupled snakelets (marked ones)

(b) improved text line segmentation results after post-processing

Figure 2.12: Example of under-segmentation failures of the presented algorithm due to badly deformed coupled snakelets and their improvement through snakelets cleaning (post-processing) a) [Left] a badly deformed coupled snakelet due to the presence of a superscript letter in between text lines that have different ending positions, a) [Right] a badly deformed coupled snakelet due to a big connected component near to comparatively small connected components. b) improved text line segmentation results after removing badly deformed coupled snakelets through snakelets cleaning (post-processing).

respect to its neighboring pairs of snakes. Such type of badly deformed pairs of snakes mainly occur because of some inherent properties of a document image:

- a document image may contain text lines with different starting and ending positions with respect to each other, as shown in the left image of Figure 2.12(a).
- a document image may contain slightly big connected component(s) near to comparatively small connected component(s) as shown in the right image of Figure 2.12(a).

A post-processing step is introduced here for cleaning/filtering badly deformed pairs of snakes and for achieving better segmentation results.

Snakelets Cleaning (post-processing)

The following observations are developed from a close examination of the coupled snakelets (pairs of snakes) in Figure 2.12(a): i) the slope of each pair, except the marked ones, is approximately the same as that of the neighboring pairs, ii) the thickness (average distance) of each pair, except the marked ones, is approximately the same as that of other neighboring pairs. Both or either of these observations do not hold for badly deformed coupled snakelets as shown in Figure 2.12(a). Therefore, badly deformed coupled snakelets can be removed by using slope and thickness based statistical analysis. Each coupled snakelet is removed as a badly deformed pair if the difference between its slope and the mean slope of neighboring coupled snakelets is greater than a predefined threshold, or if the difference between its thickness value and the mean thickness value of neighboring coupled snakelets is greater than a predefined threshold. The slope threshold can be set equal to a small value such as 10° or 15° and can be represented by T_S. The thickness threshold can be relatively selected with respect to the average height of connected components (H) and can be represented by $T_T \times H$. The height and length of the neighboring window for estimating the mean thickness and mean slope can also be relatively selected with respect to H and can be represented by $R_{pp} \times H$. Altogether, the snakelets cleaning (post-processing) step contains three free parameters: T_S, T_T, and R_{pp}. The snakelets cleaning process is applied to the examples in Figure 2.12(a) for predefined values of these free parameters, and the remaining coupled snakelets are shown in Figure 2.12(b). It is clearly visible from these examples that, the post-processing cleaning step removed badly deformed coupled snakelets and overcame undersegmentation failures. Furthermore, the performance of the presented text line segmentation algorithm is evaluated for different possible values of T_S, T_T, and R_{pp} in Section 2.5.

Text Lines Labeling

As shown in Figures 2.11 and 2.12(b), each group of overlapping or touching pairs of snakes represents a group of connected components that belong to a particular text line. Each group of connected components is assigned a unique text line label. Each small noisy components that was removed in the noise cleanup step is assigned the label of its nearest text line. A few example results of curled text lines segmentation using coupled snakelets algorithm followed by post-processing and text lines labeling are shown in Figure 2.13.

All steps of the coupled snakelets based curled text line extraction algorithm is shown in Figure 2.8.

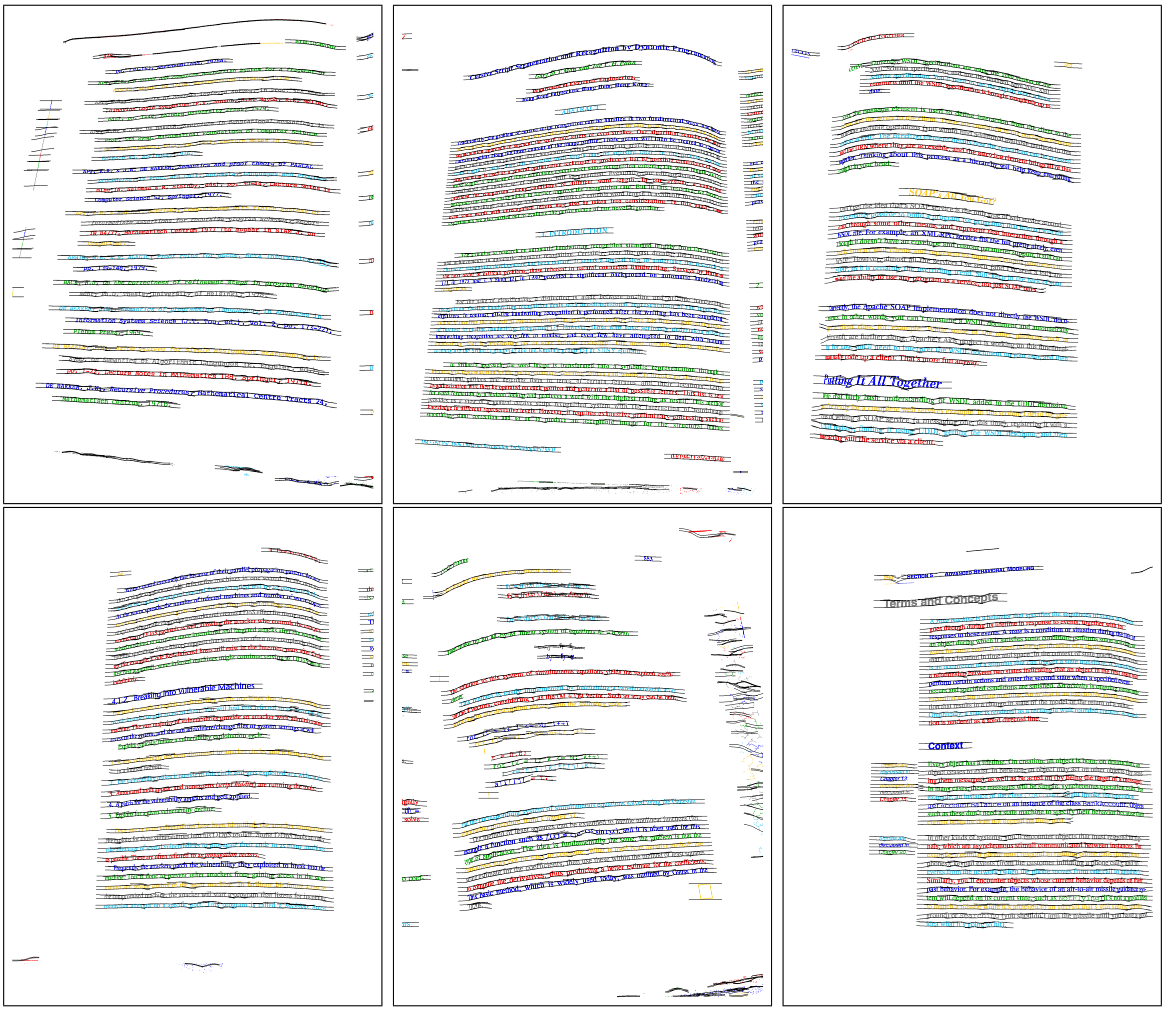


Figure 2.13: Accurate curled text line segmentation results of coupled snakelets algorithm for camera-captured document images of the DFKI-I dataset. [Note: text line segmentation results are shown in color coded form using repetition of six different colors. Two or more text lines with same color do not necessarily mean undersegmentation error. In order to avoid this confusion, coupled snakelets are also drawn here to mention the boundary of each segmented text line.]

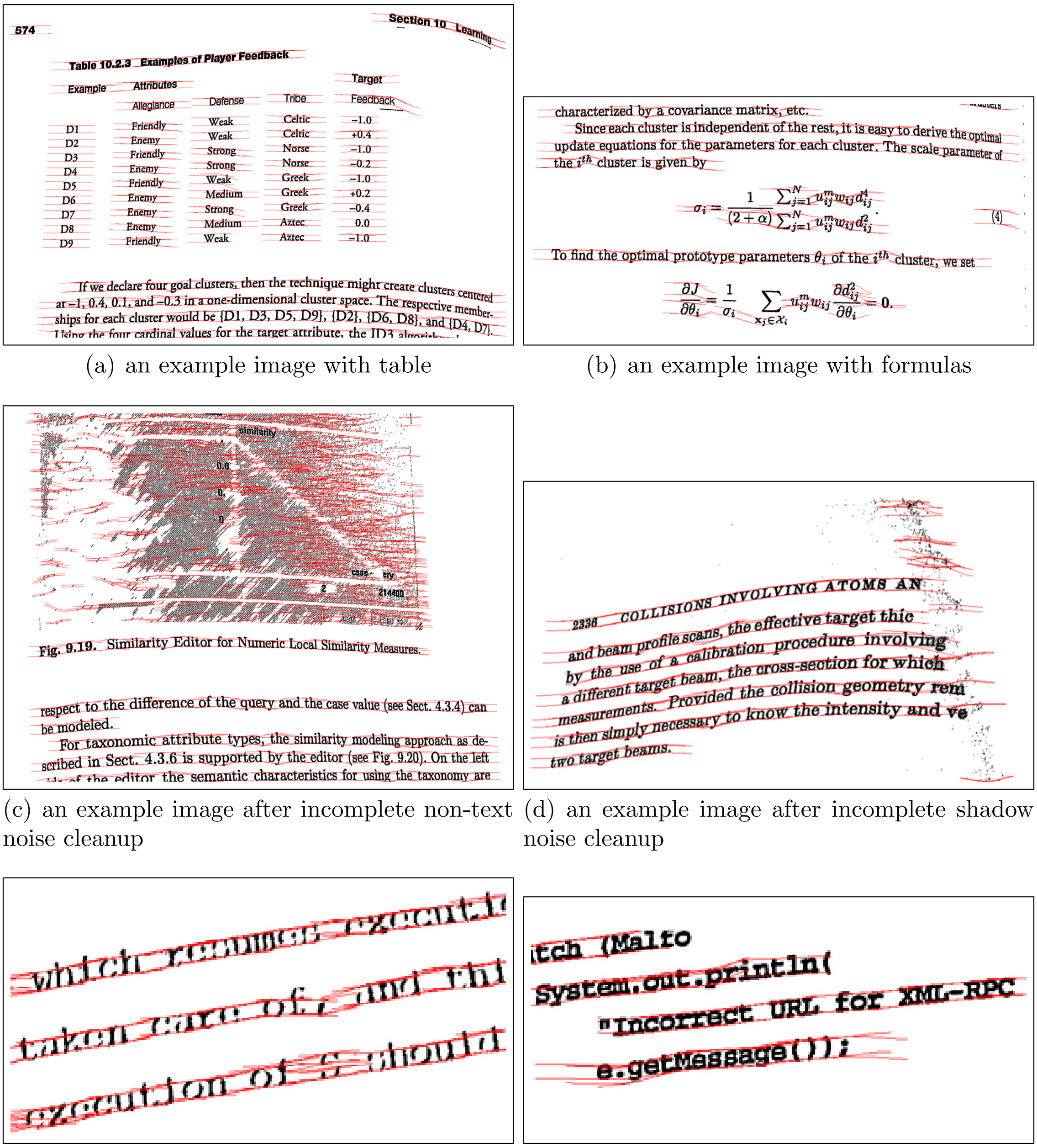
Behavior of Coupled Snakelets under Challenging Conditions:

The presented algorithm is designed for text lines segmentation which can not handle tables and/or formulas segmentation, but it detects text lines correctly even in the presence of tables and/or formulas as shown in Figure 2.14(a) and Figure 2.14(b), respectively. Grayscale camera-captured document images are usually composed of shadows that are captured using a hand-held camera in an unconstrained environment. A local adaptive thresholding produces a small amount of noise from shadows, but a global thresholding technique produces a large amount of shadow based noise. The presented text line detection algorithm starts with binarization step using a local adaptive thresholding technique, and then, it performs cleanup step in order to remove noise (that are originated from shadows, borders, etc.) and other non-text components (like graphics, drawings, etc.). It is also possible that the cleanup process is unable to remove noise and/or non-text components completely. The presented algorithm works well even in the presence of the remaining amount of non-text noise and/or shadow noise as shown in Figure 2.14(c) and Figure 2.14(d), respectively. A binarized camera-captured document image may also contain broken or joined characters. Coupled snakelets algorithm can give satisfactory text lines segmentation results under these conditions as shown in Figure 2.14(e) and Figure 2.14(f), respectively, until characters are broken into a large number of pieces and/or a large number of characters are joined together. In the presence of big connected components of joined characters, text line segmentation results can further be improved by using a character segmentation algorithm like dynamic programing based curved-cut segmentation [Bre01].

2.5 Performance Evaluation

The performance of coupled snakelets based curled text line segmentation algorithm is evaluated on the publicly available DFKI-I (the CBDAR 2007 dewarping contest) dataset [SB07] by using Shafait et al. [SKB08b] performance evaluation metrics.

The DFKI-I dataset [SB07] consists of 102 document images. These images are captured from several technical books by using hand-held camera in an office environment and are composed of curled text lines due to geometric and perspective distortions. This dataset contains ASCII-text ground-truth and pixel-based ground-truth for zones, text



574 Section 10 Learning

Table 10.2.3 Examples of Player Feedback

Example	Attributes			Target
	Allegiance	Defense	Tribe	Feedback
D1	Friendly	Weak	Celtic	–1.0
D2	Enemy	Weak	Celtic	+0.4
D3	Friendly	Strong	Norse	–1.0
D4	Enemy	Strong	Norse	–0.2
D5	Friendly	Weak	Greek	–1.0
D6	Enemy	Medium	Greek	+0.2
D7	Enemy	Strong	Greek	–0.4
D8	Enemy	Medium	Aztec	0.0
D9	Friendly	Weak	Aztec	–1.0

If we declare four goal clusters, then the technique might create clusters centered at –1, 0.4, 0.1, and –0.3 in a one-dimensional cluster space. The respective memberships for each cluster would be {D1, D3, D5, D9}, {D2}, {D6, D8}, and {D4, D7}. Using the four cardinal values for the target attribute, the ID3 algorith

(a) an example image with table

characterized by a covariance matrix, etc.

Since each cluster is independent of the rest, it is easy to derive the optimal update equations for the parameters for each cluster. The scale parameter of the i^{th} cluster is given by

$$\sigma_i = \frac{1}{(2+\alpha)} \frac{\sum_{j=1}^{N} u_{ij}^m w_{ij} d_{ij}^4}{\sum_{j=1}^{N} u_{ij}^m w_{ij} d_{ij}^2}. \quad (4)$$

To find the optimal prototype parameters θ_i of the i^{th} cluster, we set

$$\frac{\partial J}{\partial \theta_i} = \frac{1}{\sigma_i} \sum_{\mathbf{x}_j \in \mathcal{X}_i} u_{ij}^m w_{ij} \frac{\partial d_{ij}^2}{\partial \theta_i} = \mathbf{0}.$$

(b) an example image with formulas

Fig. 9.19. Similarity Editor for Numeric Local Similarity Measures.

respect to the difference of the query and the case value (see Sect. 4.3.4) can be modeled.

For taxonomic attribute types, the similarity modeling approach as described in Sect. 4.3.6 is supported by the editor (see Fig. 9.20). On the left side of the editor the semantic characteristics for using the taxonomy are

(c) an example image after incomplete non-text noise cleanup

2336 COLLISIONS INVOLVING ATOMS AN

and beam profile scans, the effective target thic
by the use of a calibration procedure involving
a different target beam, the cross-section for which
measurements. Provided the collision geometry rem
is then simply necessary to know the intensity and ve
two target beams.

(d) an example image after incomplete shadow noise cleanup

which requires executio
taken care of, and thi
execution of S should

(e) an example image with broken characters

```
tch (Malto
System.out.println(
    "Incorrect URL for XML-RPC
    e.getMessage());
```

(f) an example image with joined characters

Figure 2.14: Behavior of coupled snakelets text line detection method under challenging conditions. Coupled snakelets based text line segmentation algorithm segments text lines correctly in the presence of tables, formulas, remaining portions of non-text noise, or remaining portion of shadow noise as shown in Figures (a) to (d). Here, It is also important to note that, the snakelets over tables, formulas, or remaining amount of noise produce false alarms. Coupled snakelets based text line segmentation algorithm also work well in the presence of broken and/or joined characters as shown in Figures (e) and (f), respectively.

lines, formulas, tables and figures. Pixel-based, color-coded ground-truth is defined as follows: i) red channel contains zone class information, ii) blue channel contains zone number (in reading order) information, iii) green channel contains textline number information which is equal to zero for formulas, tables and figures and iv) marginal noise and foreground objects outside page boundary are marked with black color (all three color channels are set equal to zero). For curled text line segmentation performance evaluation, text line based ground-truth images are generated automatically using color-coded information. A text line based ground-truth image contains labeling only for text lines such that all other foreground objects within page boundary where green channel equals to zero, like formulas, tables and figures, are marked as noisy pixels with black color. An example image from the DFKI-I dataset and its corresponding text lines based ground-truth image are shown in Figure 2.15.

Performance evaluation is based on vectorial performance evaluation metrics that were presented in Shafait et al. [SKB08b], which are described as follows. Consider two segmented images, the ground truth G and hypothesized segmentation H. A weighted bipartite graph is computed called “pixel-correspondence graph” between G and H for evaluating the quality of the segmentation algorithm. Each node in G represents a text line (*ground-truth component*), and each node in H represents a segmented text line (*segmented component*). An edge is constructed between two nodes such that the weight of the edge equals the number of foreground pixels in the intersection of the regions covered by the two segments represented by the nodes. The matching between G and H is considered perfect if there is only one edge incident to each component of G or H, otherwise it is not perfect, i.e. each node in G or H may have multiple edges. The edge incident to a node is significant if the value of $w_i/P \geq t_r$ and $w_i \geq t_a$, where w_i is the edge-weight, P is the number of pixels corresponding to a node (segment), t_r is a relative threshold and t_a is an absolute threshold. In practice, $t_r = 0.1$ and $t_a = 100$ are good choices for text lines based performance evaluation for typed-text document images [SKB08b]. The same parameter values are used here for the performance evaluation of the coupled snakelets and other text line segmentation algorithms. However, for handwritten document images $t_r = 0.15$ and $t_a = 500$ are good choices.

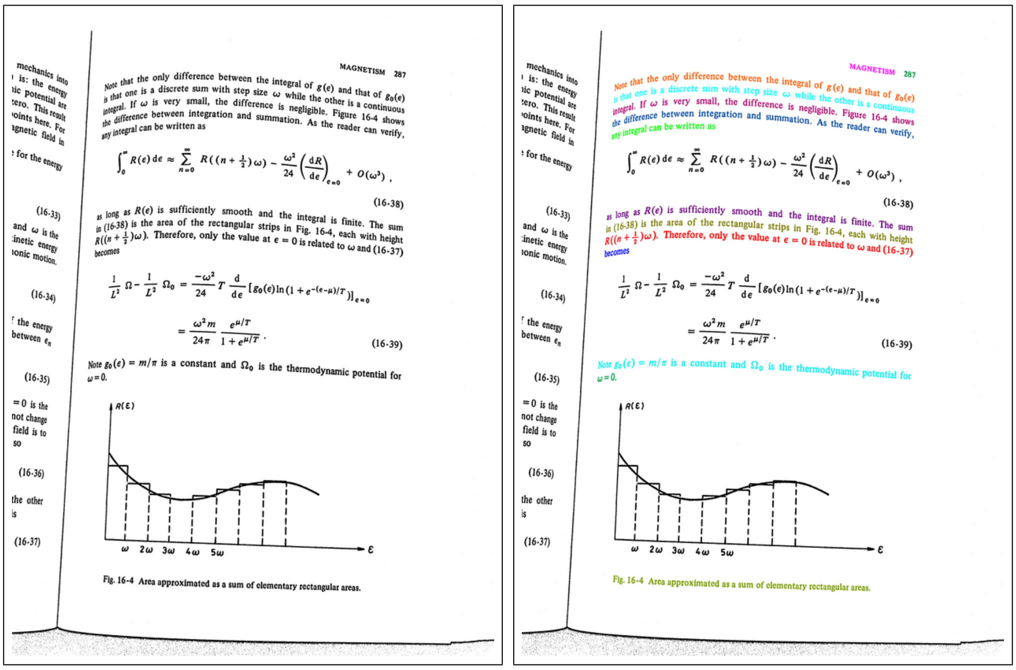
Let N_g represents total number of ground-truth components and N_s represents total number of segmented components. Based on the above description, the performance evaluation metrics are:

- **Total correct segmentation (N_{o2o}):** the number of one-to-one matches between the ground-truth components and the segmented components. The one-to-one match accuracy is calculated by $P_{o2o} = N_{o2o}/N_g$.

- **Over-segmented components (N_{ocomp}):** the number of ground truth lines having more than one significant edge. The percentage of over-segmented components is calculated by $P_{ocomp} = N_{ocomp}/N_g$.
- **Under-segmented components (N_{ucomp}):** the number of segmented lines having more than one significant edge. The percentage of under-segmented components is calculated by $P_{ucomp} = N_{ucomp}/N_g$.
- **Missed components (N_{mcomp}):** the number of ground truth components that match the background in the hypothesized segmentation. The percentage of missed components is calculated by $P_{mcomp} = N_{mcomp}/N_g$.
- **Total over-segmentations (N_{oseg}):** the number of significant edges that ground truth lines have, minus the number of ground truth lines.
- **Total under-segmentations (N_{useg}):** the number of significant edges that segmented lines have, minus the number of segmented lines.
- **False alarms (N_{falarm}):** the number of components in the hypothesized segmentation that did not match any foreground component in the ground-truth segmentation.

One of the main advantages of this vectorial metric is that, it represents not only one-to-one segmentation accuracy (P_{o2o}), but also most important classes of segmentation errors, such as over-, under-, and miss-segmentation.

The presented coupled snakelets based curled text line segmentation algorithm contains three free/tunable parameters (α, β, and γ) for the coupled snakelets estimation, and three parameters (T_S, T_T, and R_{pp}) for the post-processing step. All of these parameters have been explained in detail in Section 2.3. A brief description of these parameters are as follows. Parameters α and β are used to control snake's internal energy during deformation, and γ is used to control snake's external energy. The parameter α is usually set to a value like 0.05, 0.5, 5, etc. The parameter β is set to a small value when no stiffness is required and to a large value when high snake's stiffness is required during deformation steps (like the presented coupled snakelets model). The possible range of values for parameter γ is in between 0 to 1. The parameter T_S is the slope threshold, T_T is the relative thickness threshold (with respect to the mean height of connected components in a document image H), and R_{pp} is the relative window size with respect to H.



MAGNETISM 287

Note that the only difference between the integral of $g(\epsilon)$ and that of $g_0(\epsilon)$ is that one is a discrete sum with step size ω while the other is a continuous integral. If ω is very small, the difference is negligible. Figure 16-4 shows the difference between integration and summation. As the reader can verify, any integral can be written as

$$\int_0^\infty R(\epsilon)\,\mathrm{d}\epsilon \approx \sum_{n=0}^{\infty} R((n+\tfrac{1}{2})\omega) - \frac{\omega^2}{24}\left(\frac{\mathrm{d}R}{\mathrm{d}\epsilon}\right)_{\epsilon=0} + O(\omega^3)\,, \tag{16-38}$$

as long as $R(\epsilon)$ is sufficiently smooth and the integral is finite. The sum in (16-38) is the area of the rectangular strips in Fig. 16-4, each with height $R((n+\frac{1}{2})\omega)$. Therefore, only the value at $\epsilon = 0$ is related to ω and (16-37) becomes

$$\frac{1}{L^2}\Omega - \frac{1}{L^2}\Omega_0 = \frac{-\omega^2}{24}T\frac{\mathrm{d}}{\mathrm{d}\epsilon}[g_0(\epsilon)\ln(1+e^{-(\epsilon-\mu)/T})]_{\epsilon=0}$$

$$= \frac{\omega^2 m}{24\pi}\frac{e^{\mu/T}}{1+e^{\mu/T}}\,. \tag{16-39}$$

Note $g_0(\epsilon) = m/\pi$ is a constant and Ω_0 is the thermodynamic potential for $\omega = 0$.

Fig. 16-4 Area approximated as a sum of elementary rectangular areas.

(a) original Image

MAGNETISM 287

Note that the only difference between the integral of $g(\epsilon)$ and that of $g_0(\epsilon)$ is that one is a discrete sum with step size ω while the other is a continuous integral. If ω is very small, the difference is negligible. Figure 16-4 shows the difference between integration and summation. As the reader can verify, any integral can be written as

$$\int_0^\infty R(\epsilon)\,\mathrm{d}\epsilon \approx \sum_{n=0}^{\infty} R((n+\tfrac{1}{2})\omega) - \frac{\omega^2}{24}\left(\frac{\mathrm{d}R}{\mathrm{d}\epsilon}\right)_{\epsilon=0} + O(\omega^3)\,, \tag{16-38}$$

as long as $R(\epsilon)$ is sufficiently smooth and the integral is finite. The sum in (16-38) is the area of the rectangular strips in Fig. 16-4, each with height $R((n+\frac{1}{2})\omega)$. Therefore, only the value at $\epsilon = 0$ is related to ω and (16-37) becomes

$$\frac{1}{L^2}\Omega - \frac{1}{L^2}\Omega_0 = \frac{-\omega^2}{24}T\frac{\mathrm{d}}{\mathrm{d}\epsilon}[g_0(\epsilon)\ln(1+e^{-(\epsilon-\mu)/T})]_{\epsilon=0}$$

$$= \frac{\omega^2 m}{24\pi}\frac{e^{\mu/T}}{1+e^{\mu/T}}\,. \tag{16-39}$$

Note $g_0(\epsilon) = m/\pi$ is a constant and Ω_0 is the thermodynamic potential for $\omega = 0$.

Fig. 16-4 Area approximated as a sum of elementary rectangular areas.

(b) labeled text lines

Figure 2.15: An example image of the DFKI-I dataset [SB07] and its corresponding text line based ground-truth image. Note that non-text elements (equation, graphics) as well as partial text lines from the neighboring page have been considered as "noise" in the ground-truth.

Experimental results show that the post-processing step does not require a very small value (like 0°) or a comparatively large value (like 45°) for parameter T_S. Similarly, the relative values for T_T and R_{pp} can be set in between 1 to 10.

For optimization of these parameters and showing their effects on text line detection accuracy, the presented coupled snakelets based curled text line segmentation algorithm is evaluated on a subset of 11 images from the DFKI-I dataset (that start with name *dsc00*) for different values of these free parameters. The one-to-one text line segmentation accuracy (P_{o2o}) of the presented algorithm for the different values of these free parameters is shown in Figure 2.16. Here a sequential procedure is adopted for evaluating and optimizing the performance of the presented text line detection method with respect to the different values of these parameters. In Figure 2.16(a) the text line detection accuracy is shown for different values of α and β with empirically chosen values for other parameters ($\gamma = 1$, $T_S = 10°$, $T_T = 30$ pixels (absolute value), $R_{pp} = 150$ pixels (absolute value)). Similarly, in Figure 2.16(b) the text line detection accuracy is represented for different values of γ with optimized values for $\alpha = 0.05$ and $\beta = 10000$ (from Figure 2.16(a)), and chosen values for other parameters ($T_S = 10°$, $T_T = 30$ pixels, $R_{pp} = 150$ pixels). Likewise, in Figure 2.16(c) the text line detection accuracy is shown for different values of T_T and T_S with optimized values for $\alpha = 0.05$, $\beta = 10000$ and $\gamma = 1$(from Figure 2.16(a) and Figure 2.16(b)), and chosen value for $R_{pp} = 150$ pixels. Finally, in Figure 2.16(d) the text line detection accuracy is shown for different values of R_{pp} with optimized values for others ($\alpha = 0.05$, $\beta = 10000$ and $\gamma = 1$, $T_T = 1$ and $T_S = 10°$; from Figure 2.16(a) to 2.16(c)).

From Figure 2.16, it can be concluded that the performance of the presented text line segmentation method is not sensitive to the values of most of the free parameters (like α, γ, T_T and R_{pp}), except β and T_S. The optimized values of these parameters for a subset of 11 images from the DFKI-I dataset are as follows: $\alpha = 0.05$, $\beta = 10000$, $\gamma = 1$, $T_S = 10°$, $T_T = 1$, and $R_{pp} = 5$.

On the complete DFKI-I dataset, the performance of the presented coupled snakelets based curled text line segmentation algorithm is also compared with other competing curled text line segmentation algorithms: i) nearest-neighbors (Gatos et al. [GPN07]), ii) baby-snakes (Bukhari et al. [BSB08]), iii) rule-based (Oliveria et al. [OLT$^+$10]), iv) Docstrum [O'G93]. In the literature review (Section 2.2), a minor modification for the nearest neighbor based algorithm [GPN07] is also proposed by introducing the free parameter T. The average height of a connected components in the DFKI-I dataset is approximately equal to 20, therefore we set $T = 20$ for the modified version of nearest neighbor based algorithm [GPN07]. The performance of a state-of-the-art straight text line segmentation algorithm (Docstrum [O'G93]) is also evaluated for curled text line segmentation from camera-captured document images. Docstrum is one of the state-of-the-art page segmentation algorithm for scanned document images with straight text lines. The main reason of including it here is to show that how challenging the dataset is, and that straight text lines segmentation algorithms can not be directly applied for curled document images. Performance evaluation results of all of these algorithms for the DFKI-I dataset are shown in Table 2.1.

Among all curled text line segmentation algorithms that are shown in Table 2.1, the presented coupled snakelets algorithm achieved the highest percentage of one-to-one segmentation accuracy and the lowest percentages of oversegmentation and missed text line errors. The presented algorithm also achieved the second lowest percentage of undersegmentation errors. Almost all of the algorithms have produced large numbers of false-alarm errors. In general, a large number of false-alarms can be reduced by using an appropriate pre-processing or post-processing step, for example a page boundary de-

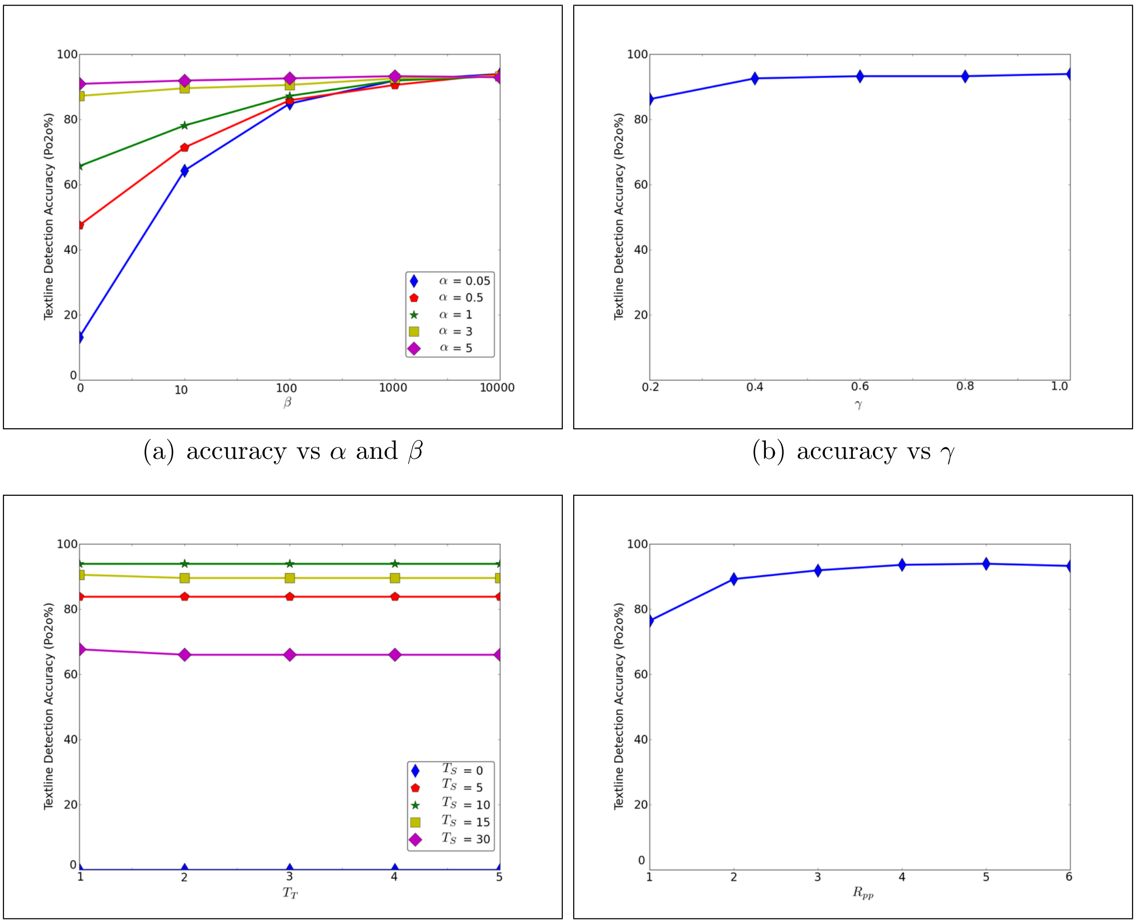
Table 2.1: Performance evaluation results of the presented coupled snakelets based and competing state-of-the-art curled text line segmentation algorithms as well as a straight text line segmentation algorithm (Docstrum [O'G93]) on binary camera-captured document images of the DFKI-I dataset [SB07] by using performance evaluation metrics [SKB08b]. This dataset contains 102 document images captured using hand-held camera in an uncontrolled environment. The coupled snakelets method achieved the highest one-to-one text line segmentation accuracy as compared to other methods.

Algorithm	**Performance Evaluation Metrics** [a]									
	N_g	N_s	N_{o2o}	N_{falarm}	N_{useg}	N_{oseg}	P_{ucomp}	P_{ocomp}	P_{mcomp}	P_{o2o}
Docstrum [O'G93] [b]	3091	6852	657	6066	2096	4383	51.05%	66.90%	**0%**	21.26%
Nearest-Neighbor [GPN07]	3091	6983	898	307601	**16**	1365	**0.49%**	22.94%	44.74%	29.05%
Neighbor modified [GPN07] [c]	3091	3256	2780	4215	102	208	3.17%	6.05%	0.03%	89.93%
Rule-Based [OLT$^+$10]	3091	2924	2816	**785**	57	682	1.81%	21.71%	4.43%	91.10%
Baby-Snakes [BSB08]	3091	3371	2707	13199	117	294	2.91%	5.79%	**0%**	87.58%
Coupled Snakelets	*3091*	*3106*	***2940***	*3328*	*51*	***61***	*1.58%*	***1.84%***	**0%**	***95.12%***

[a] N_g:ground-truth components; N_s:segmented components; N_{o2o}:one-to-one matched components; N_{falarm}: false alarms; N_{useg}: undersegmentations; N_{oseg}: oversegmentations; N_{ucomp}: undersegmented components; $P_{ucomp} = N_{ucomp}/N_g$; N_{ocomp}:oversegmented components; $P_{ocomp} = N_{ocomp}/N_g$; N_{mcomp}: missed components; $P_{mcomp} = N_{mcomp}/N_g$; $P_{o2o} = N_{o2o}/N_g$;

[b] Docstrum [O'G93] is used for scanned document image segmentation with straight text lines. Here it is used for curled text lines segmentation to show that: i) straight text lines algorithm are not directly application on camera-captured documents and ii) the the DFKI-I dataset is challenging with respect to curled text lines.

[c] The original version of Gatos et al. [GPN07] nearest-neighbor based curled text line detection algorithm and the proposed modification are described in Section 2.2.



(a) accuracy vs α and β

(b) accuracy vs γ

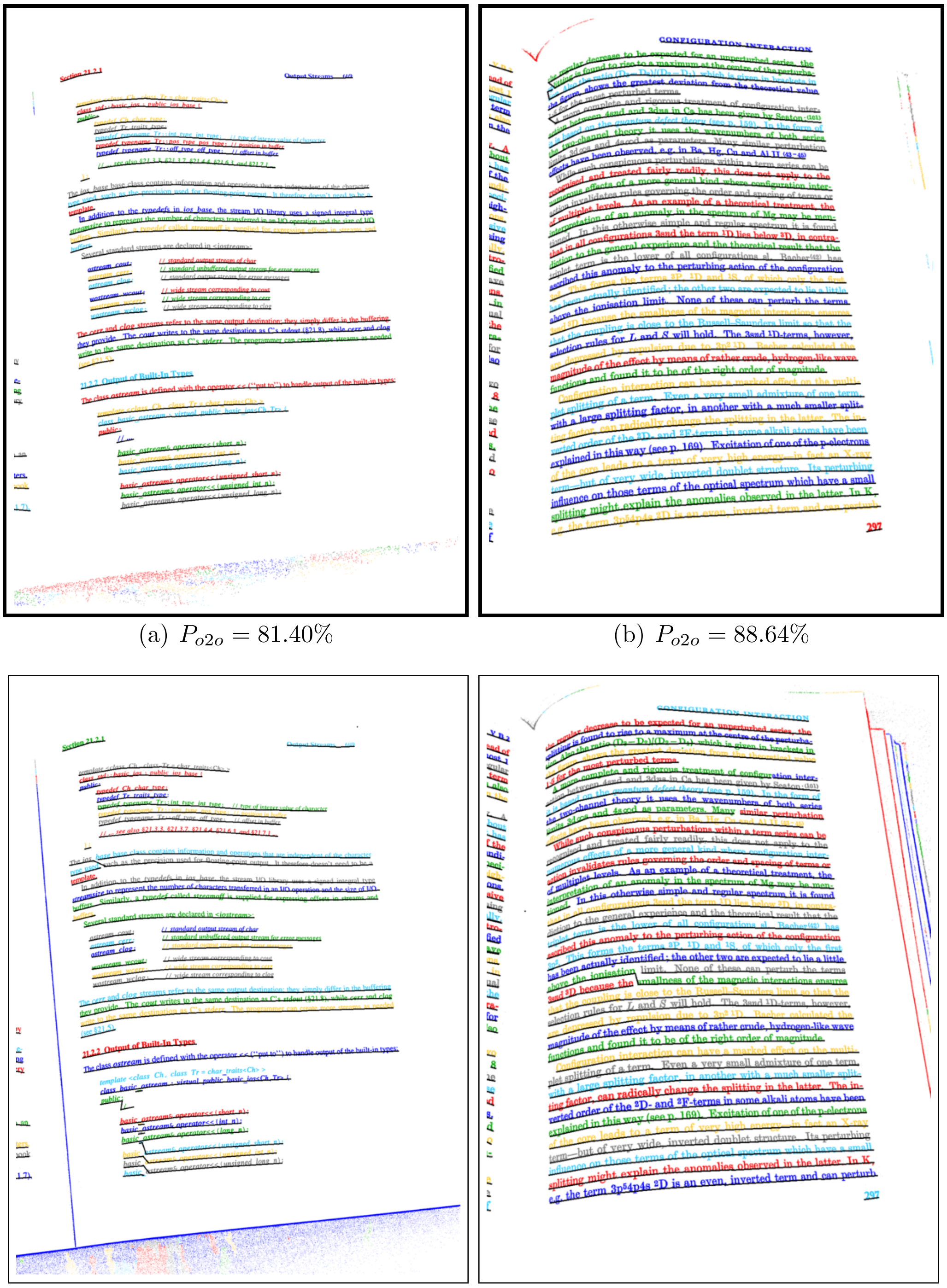
(c) accuracy vs T_T and T_S

(d) accuracy vs Rpp

Figure 2.16: Curled text line segmentation accuracy (P_{o2o}) of the presented coupled snakelets based algorithm for different values of free parameters, α, β, γ, T_S, T_T, and R_{pp}, on a subset of 11 images from the DFKI-I dataset.

tection method [SGK07, SvBKB08] can help in removing textual and non-textual border noise.

A few sample documents from the DFKI-I dataset having the largest number of text line segmentation errors for the presented coupled snakelets algorithm are shown in the top row of Figure 2.17, and for comparison the corresponding results of the modified nearest-neighbor based algorithm [GPN07] are shown in the bottom row of Figure 2.17. Even in these examples, the presented algorithm has overall performed better than modified nearest-neighbor based algorithm [GPN07]. The document image in Figure 2.17(a) contains total 43 text lines, and the presented method detected 35 of them correctly. In this example, most of the errors belong to oversegmentation category. These oversegmentation errors mainly occur because of big gaps between words within some text lines,



(a) $P_{o2o} = 81.40\%$ (b) $P_{o2o} = 88.64\%$

(c) $P_{o2o} = 69.77\%$ (d) $P_{o2o} = 86.36\%$

Figure 2.17: Text Line Segmentation Failures: **Top Row:** the largest number of text line segmentation errors on some sample documents from the the DFKI-I dataset for the presented coupled snakelets algorithm. **Bottom Row:** the corresponding results of modified version of nearest-neighbor based curled text line segmentation algorithm [GPN07] for comparison. [Note: in order to highlight the segmentation results, the color coded segmented text lines are also underlined manually by black line.]

which can be seen in the middle of the page. The document image of Figure 2.17(b) contains very small gaps between text lines resulting in undersegmentation errors. For this image, the presented coupled snakelets method detected 39 text lines correctly out of 44 text lines, and produced some undersegmentation errors, which can be seen in the top area of the document image in Figure2.17(b).

The average size of a document image in the DFKI-I dataset is around 8 Mega-pixels, and the average size of a character is this dataset is 19 pixels wide (with a standard deviation of 11 pixels) and 25 pixels tall (with a standard deviation of 9 pixels). The execution time of the presented coupled snakelets text line detection method is directly proportional to the size and the number of connected components in a document image. Since the coupled snakelets method takes a lot of deformations, it is slow. On weighted average with respect to the number of connected components, the presented algorithm takes around 38 minutes per page with text line detection accuracy of around 95%. The code is implemented using Python programming language without using any Python specific and/or active contour specific optimization techniques. The execution time can be reduced by using these types of optimization techniques, which is one of a future research goals. Here, a basic strategy is tested for reducing the execution time of the presented method such that, a document image is downscaled for coupled snakelets calculation and then the estimated snakelets are upscaled proportionally for text lines labeling step. Figure 2.18 shows the text line detection accuracy of the presented coupled snakelets method and the corresponding average execution time for different size of downscaled images for the DFKI-I dataset. This process also demonstrates how well the presented text line segmentation algorithm can perform for small character sizes. Around three 3-fold speed gain (i.e. 13 minutes per page) is achieved with around 1% reduction in text line detection accuracy (i.e. 94%) for around 2-fold image downscaling factor. It also shows that, the presented algorithm can work gracefully up to a minimum character size of around 10 pixels wide and around 12 pixels tall.

2.6 Adaptation to Other Scripts

In addition to Latin scripts, the presented text line extraction method can be applied to other scripts. Here, the adaptation of the coupled snakelets based text line extraction method to different scripts like Chinese, Devanagari and Arabic is described.

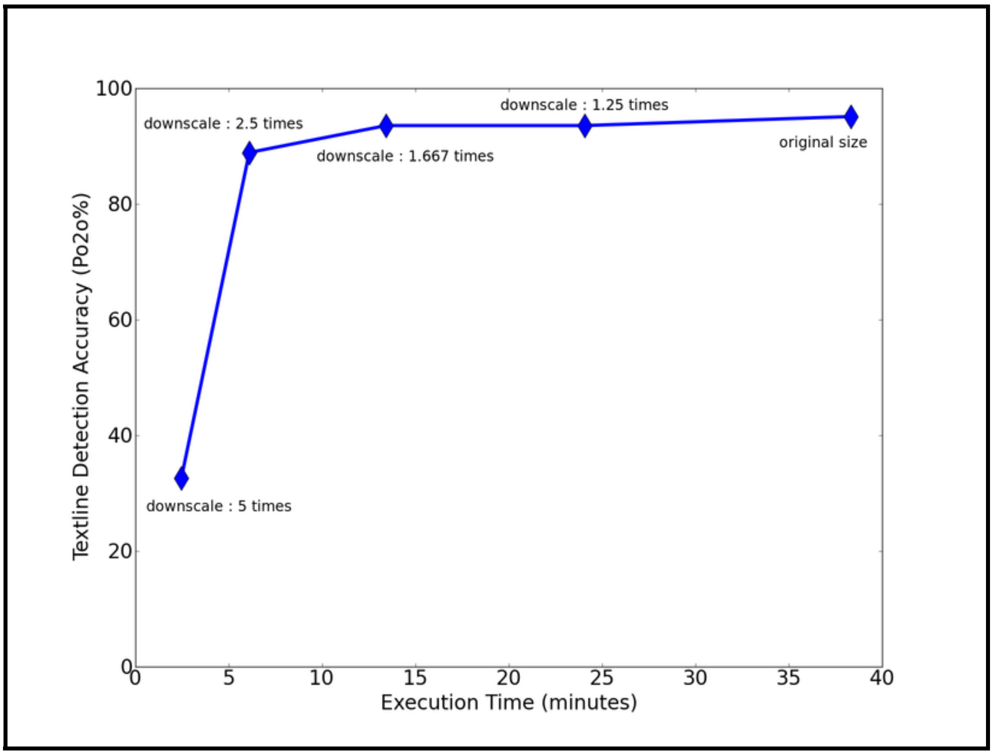


Figure 2.18: The execution time and the corresponding text line detection accuracy of the presented coupled snakelets method for different downscaling factors on the DFKI-I dataset.

2.6.1 Chinese

Chinese characters are composed of small strokes and geometric shapes, as shown in Figure 2.19(a). Unlike Latin script, a collection of top and bottom points of connected components of a text line can not represent the uniform top and bottom line of the text line, respectively, which are required for properly estimating baseline and x-line pairs through coupled snakelets method. In order to solve this problem, a Chinese script document image is smeared before top and bottom points estimation. For this purpose, a smearing method, which is a sequence of dilation, fill-holes and erosion operations, is applied. For a sample document image (Figure 2.19(a)), the smeared document image is shown in Figure 2.19(b). Now the top and bottom points of the smeared connected components can give better information about top and bottom lines of text lines, respectively. Two further modifications are made for adapting coupled snakelets method to Chinese script: (i) all top and bottom points of each connected component are used instead of using only one top and one bottom point of each connected component; (ii) in first deformation cycle, each pair of snakes is deformed without extra length and neighborhood area. In this way, the initial pair of snakes for each connected components is aligned to its slope direction. The remaining deformation cycles follow the same additional length and growing neighborhood area criteria as mentioned in the algorithm (Figure 2.8). For the smeared document image (Figure 2.19(b)), the deformed pairs of coupled snakelets are shown in Figure 2.19(c) and the corresponding extracted text lines result is shown in Figure 2.19(d).

2.6.2 Devanagari

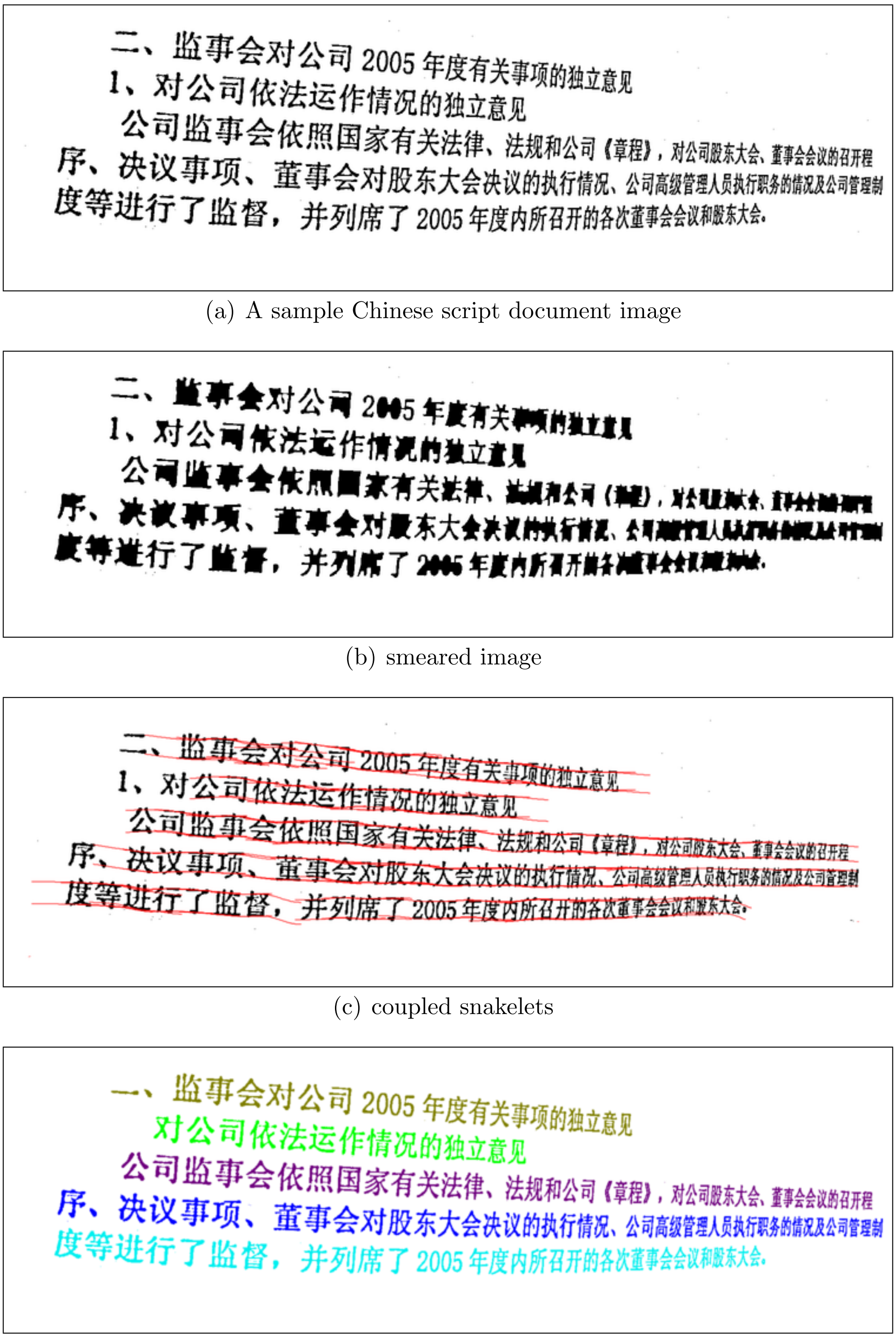
Devanagari script contains big characters as compared to Latin and Chinese scripts as shown in Figure 2.20(a). Here the same approach as mentioned for Chinese document images is followed except smearing, because Devanagari connected components do not have multiple non-touching components. For a sample document image, the deformed pairs of coupled snakelets are shown in Figure 2.20(b) and the corresponding extracted text lines are shown in Figure 2.20(c).

2.6.3 Arabic

The Coupled snakelets method is mainly designed to follow the uniform top and bottom points of connected components. Here, the potential of coupled snakelets based text line extraction method is shown for a case where the characters within a text line do not contain smooth top and bottom points. Unlike Latin, Chinese and Devanagari scripts, Arabic script document images contains non-uniform top and bottom lines within a text line, as shown in Figure 2.21(a). Additionally, Arabic script document images contain big connected components like Devanagari script document images. Here, the same approach is applied for Arabic script curled text line extraction as described for Devanagari script. For a sample document image (Figure 2.21(a)), the deformed pairs of coupled snakelets are shown in Figure 2.21(b). Extracted text lines are shown in Figure 2.21(c).

2.7 Conclusion

Hand-held camera-captured document images usually contain warped/curled text lines because of geometric and/or perspective distortions. In this chapter, a novel curled text line segmentation algorithm is introduced by adapting active contour (snake) [KWT88]. Here, the adapted active contour (snake) model for text line segmentation is referred to as *coupled snakelets*. The presented algorithm uses only top and bottom points of connected components within a document image for detecting text lines. It jointly estimates a local pair of x-line and baseline on each connected component using top and bottom points and then each group of overlapping and/or touching pairs of x-line and baseline is considered as a segmented text line. The DFKI-I (the CBDAR 2007 dewarping contest) dataset [SB07] is used for performance evaluation of the presented method and comparison with other state-of-the-art approaches: i) nearest-neighbors - original(Gatos et al. [GPN07]) and the proposed modified version, ii) baby-snakes (Bukhari et al [BSB08]), iii) rule-based (Oliveria et al. [OLT$^+$10]), and iv) Docstrum [O'G93]. The presented method is less sensitive to high degree of curl and skew and produces a less number of over- and under-segmentation errors as compared to other state-of-the-art curled text line segmentation methods. Unlike existing approaches, the presented algorithm performs text lines segmentation and their x-line and baseline pairs estimation simultaneously that results in improved segmentation with better estimation of x-lines and baseline than other approaches. The performance evaluation results are shown in Table 2.1. The presented coupled snakelets algorithm achieved the highest one-to-one text line segmentation accuracy as compared to other methods. It also yields the lowest oversegmentation and missed text lines errors, and a small number of undersegmentation errors. The presented method contains six free/tunable parameters. Most of these parameters are non-sensitive with respect to the performance the presented method.



二、监事会对公司2005年度有关事项的独立意见

1、对公司依法运作情况的独立意见

公司监事会依照国家有关法律、法规和公司《章程》，对公司股东大会、董事会会议的召开程序、决议事项、董事会对股东大会决议的执行情况、公司高级管理人员执行职务的情况及公司管理制度等进行了监督，并列席了2005年度内所召开的各次董事会会议和股东大会。

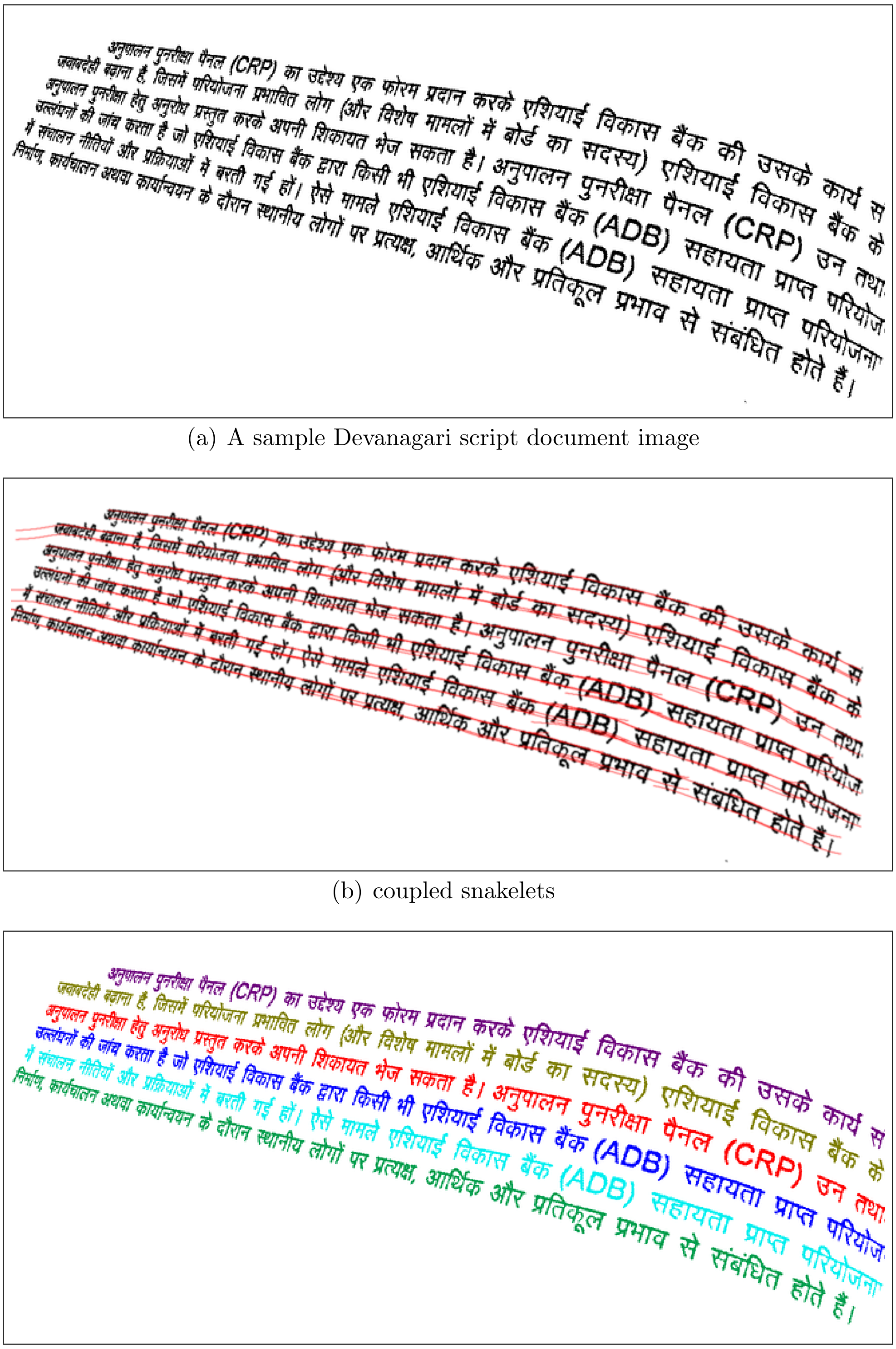
(a) A sample Chinese script document image

(b) smeared image

(c) coupled snakelets

(d) extracted text lines

Figure 2.19: Example of curled text lines extraction of camera-captured document image of Chinese script using coupled snakelets.



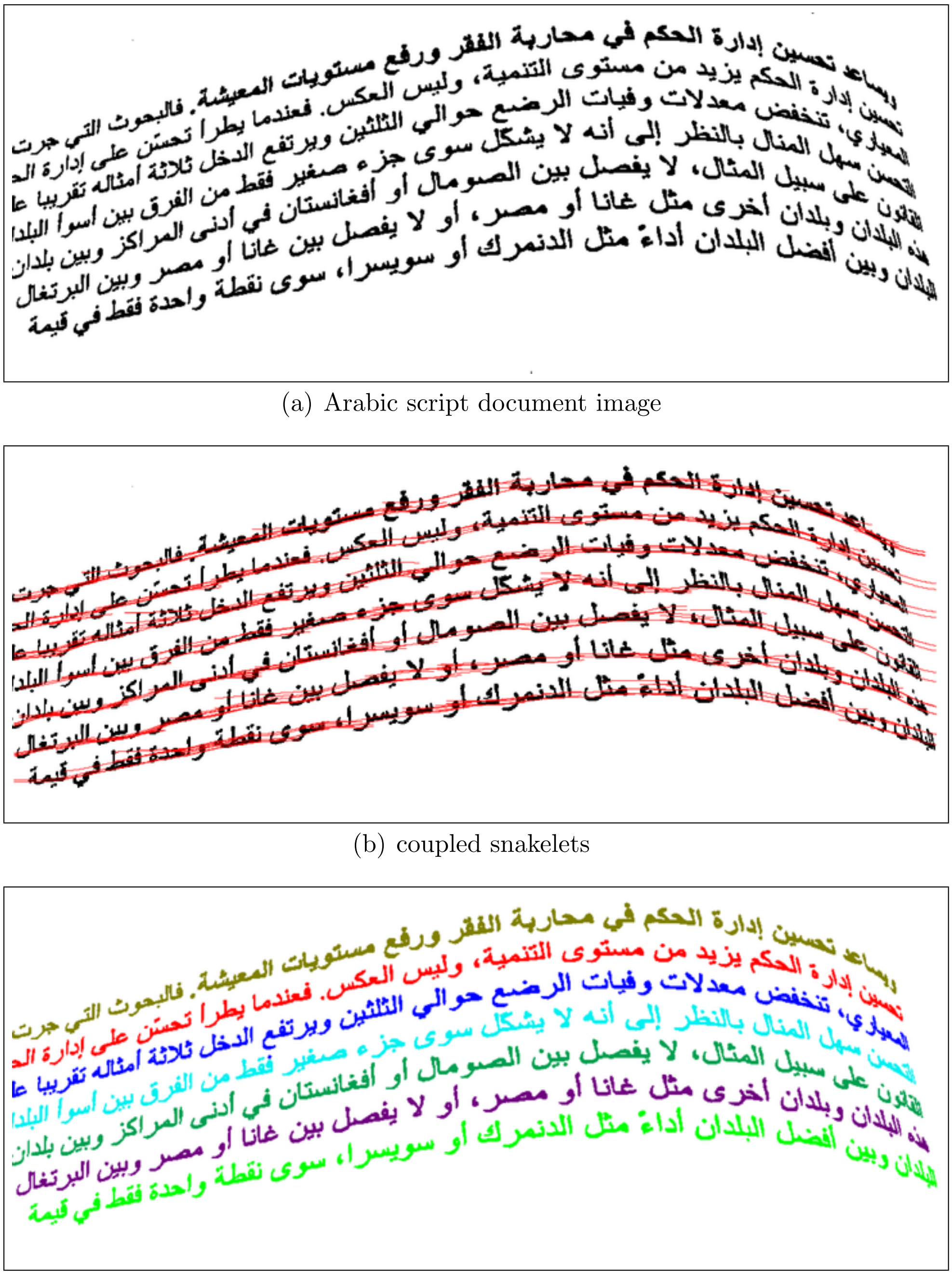
अनुपालन पुनरीक्षा पैनल (CRP) का उद्देश्य एक फोरम प्रदान करके एशियाई विकास बैंक की उसके कार्य स
जवाबदेही बढ़ाना है, जिसमें परियोजना प्रभावित लोग (और विशेष मामलों में बोर्ड का सदस्य) एशियाई विकास बैंक के
अनुपालन पुनरीक्षा हेतु अनुरोध प्रस्तुत करके अपनी शिकायत भेज सकता है। अनुपालन पुनरीक्षा पैनल (CRP) उन तथा
उल्लंघनों की जांच करता है जो एशियाई विकास बैंक द्वारा किसी भी एशियाई विकास बैंक (ADB) सहायता प्राप्त परियोज
में संचालन नीतियों और प्रक्रियाओं में बरती गई हों। ऐसे मामले एशियाई विकास बैंक (ADB) सहायता प्राप्त परियोजना
निर्माण, कार्यचालन अथवा कार्यान्वयन के दौरान स्थानीय लोगों पर प्रत्यक्ष, आर्थिक और प्रतिकूल प्रभाव से संबंधित होते हैं।

(a) A sample Devanagari script document image

(b) coupled snakelets

(c) extracted text lines

Figure 2.20: Example of curled text lines extraction of camera-captured document image of Devanagari script using coupled snakelets.



(a) Arabic script document image

(b) coupled snakelets

(c) extracted text lines

Figure 2.21: Example of curled text lines extraction of camera-captured document image of Arabic script using coupled snakelets.

Chapter 3

A Generic Text Line Extraction Method

Summary: *text line extraction is an essential step in many OCR systems. Failures in text line extraction can result in strong increase in overall system error rates. For decades, a large number of text line finding approaches have been proposed in the literature. However, there is no general-purpose text line finding method that can be robustly applied to a large variety of document classes. This chapter presents a novel text line extraction technique using matched filtering and ridge detection, that can be equally applied to a diverse collection of documents, and therefore, can be considered as a generic text line extraction technique. For the performance evaluation of the presented text line extraction method, different standard datasets are selected that belong to different categories of document images: the UW-III dataset of scanned documents, the ICDAR 2007 and the UMD datasets handwritten documents, the DFKI-I dataset camera-captured documents, Arabic/Urdu script documents dataset, and German calligraphic (Fraktur) script documents dataset. In total, there are 650 document images that collectively contain 23763 text lines. For each of these domain-specific datasets, the text line detection accuracy of different domain-specific state-of-the-art methods are analyzed. Experiments show that the presented method achieves significantly better text line detection accuracy (*86%*) than aggregate accuracy of the best performing domain-specific state-of-the-art methods (*73%*).*

3.1 Introduction

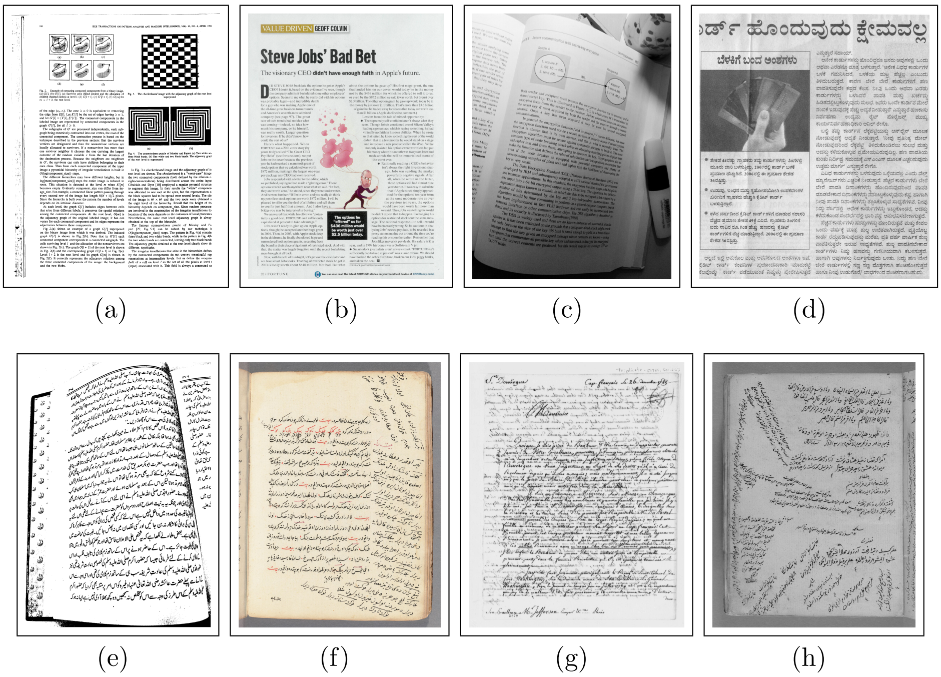
Text line is the most dominant geometrical layout structure in the context of *a diverse collection of document images* [LSZT07]–a corpus of document images consisting of a large variety of binary and grayscale, scanned and camera-captured, typed-text, handwritten and historical document images of various different scripts. A sample collection of a diverse document images is shown in Figure 3.1. Text line extraction is considered as one of the most important document image analysis steps. A large number of text line extraction methods are available in the literature. Almost all of the traditional text line extraction algorithms rely on certain assumptions about target class of documents with respect to writing styles (typed-text or handwritten), digitization methods (scanner or camera), intensity values (binary, gray or color), or scripts (Latin, Chinese, Arabic, ...). These traditional algorithms usually fail where the underlying assumptions are not satisfied. Therefore, they are referred to as domain or class specific text line extraction methods. In general, generic text line methods are especially important for both camera-based and historical document analysis, both of which are increasingly becoming important with the advent of less expensive and widely used digital cameras and digitization of a huge amount of historical documents in libraries all over the world.

This chapter introduces a generic text line extraction method using matched filtering and ridge detection approaches, which are very popular in computer vision but so far have not been applied in document image processing.

Over the last four decades, several text line extraction methods have been proposed. Comprehensive overview of the traditional state-of-the-art text line extraction methods has been provided in [CCMM98b, Nag00]. Some of these algorithms have come to widespread use for analyzing documents in different scrips and languages [SKB08b], such as X-Y cut [NSV92], smearing [WCW82], whitespace analysis [BJF90], constrained text line extraction [Bre02b], Docstrum [O'G93], and Voronoi [KSI98]. These algorithms perform well for scanned, binarized, cleaned document images with simple script, but fail on warped camera-captured document images [BSB10b], complex scripts document images [KKJ07], and handwritten/historical document images [LZDJ08, LSZT07] because of their specific challenging problems as shown in Figure 3.1. Hand-held camera-captured document images usually consist of high degree of curl with a variety of curl directions, as shown in Figure 3.1(c) and 3.1(e), which is one of the most difficult problem for curled text line finding [BSB10b]. Bukhari et. al [BSB10b] shows that the Docstrum [O'G93], which is one of the state-of-the-art text line finding methods, does not work for warped, camera-captured document images. Free-style handwritten text lines finding can be considered as the most difficult task because of the following significant challenges [LGPH08, KAAKD10]: irregular layout, lack of a well-defined baseline, variability in skew angle between different text lines and along a single text line, interline overlap and touching, noise and distortion, such as smudges, smears, faded print, and bleed-through. Some of these challenging problems for handwritten text line segmentation are shown in Figure 3.1(f), 3.1(g), and 3.1(h). Li et al. [LZDJ08] described that the X-Y cut [NSV92] and Docstrum [O'G93], which are well-known layout analysis algorithms for machine-printed document images, do not perform well for text line segmentation from freestyle handwritten document images. Latin script is considered as one of the most simplest script in the world with respect to document image processing [Nag00] as compared to other complex scripts like Arabic, Persian, African, Urdu, and Indic (Figure 3.1(d) and 3.1(e)), Kumar et al. [KKJ07] also demonstrated that the successful page segmentation algorithms for Latin Scripts [SKB08b] such as X-Y cut [NSV92], smearing [WCW82], Docstrum [O'G93], Voronoi [KSI98], constrained text line detection [Bre02b] and whitespace analysis [BJF90] do not give good results for complex Indic scripts.

Text line extraction for complex document images, like camera-captured, complex script, and handwritten/historical document images, is an open challenging problem. Every year, several researchers proposed different text line extraction methods for solving specific problems in each of these categories of complex document images. Detailed overview of the state-of-the-art text line extraction methods for warped camera-captured document images has been provided in [BSB10b] and for handwritten and historical document images in [LSZT07, GAS07, LGPH09, ASES11], and some of the specialized script-specific text line extraction approaches for complex script document images can be found in [KKJ07, SHKB06]. Beside the sophisticated nature of most of these domain specific methods, some of them also require preprocessing steps (such as binarization, document cleanup, zone segmentation, skew correction etc.) before moving towards text line extraction step and/or some post-processing steps after applying text line extraction. Most of these approaches also consist of a few or many, but ambiguous or ill-defined, free-parameters and their tunning/optimization is a difficult task. Even most of these specialized domain/class specific methods rely on certain assumptions about the structure of target document images for which they are designed, and fail on other types of document images within the same domain where the underlying assumptions are not satisfied. For example, even if an algorithm is designed for Arabic handwritten text line segmentation problem (domain specific method), there is no guarantee that it can work robustly for different varieties of Arabic handwritten document images.

Currently, there is no universal or generic text line finding method that can be robustly applied without modifications to a diverse collection of document images. Figure 3.1 shows a diverse collection of documents. Many researchers [NJ07, LSZT07] highlighted



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.1: A collection of diverse document images. a) Scanned, Binary, Printed-Text Latin Script, Manhattan Layout. b) Scanned, Color, Printed-Text Latin Script, Non-Manhattan Layout. c) Camera-Captured, Grayscale, Printed-Text Latin Script, Warped Document. d) Scanned, Grayscale, Printed-Text Kannada Script, Manhattan Layout. e) Camera-Captured, Grayscale, Printed-Text Urdu Script, Warped Document. f) Scanned, Color, Handwritten/Historical Arabic Script, Irregular Layout. g) Scanned, Grayscale, Handwritten/Historical English Script, Background Noise. h) Scanned, Grayscale, Handwritten/Historical Arabic Script, Irregular Layout.

the major challenging problem of a universal/generic text line extraction method as an elusive goal so far, which is still beyond the reach of the state-of-the-art in the field.

This chapter presents a novel text line extraction method using standard computer vision approaches: matched filtering and ridge detection. The presented method is a generic text line extraction algorithms that can be equally applied to a large variety of document image classes (as shown in Figure 3.1), which include binary and grayscale intensity values, scanned and camera-captured documents, different scripts, and different text line structures, such as typed-text straight, skewed and curled text lines, and free-style handwritten text lines (that may contain interline touching, inter-line overlapping, irregular layout, and noise). Unlike the sophisticated nature of most of the domain specific text line finding methods, the generic text line finding technique consists of two standard, simple, and easy to understand and implement image processing algorithms:

(i) matched filtering and (ii) ridge detection. Because of that, it contains well-defined free-parameters, which will be discussed in Section 3.2. The tuning/optimization of these free-parameters is a simple task. In this chapter, performance of the ridge-based text line extraction method is evaluated on a collection of diverse document images and compared with several domain-specific state-of-the-art methods. This chapter shows that the ridge-based text line extraction method achieves significantly better text line detection accuracy than different state-of-the-art methods.

The rest of this chapter is organized as follows. The presented generic text line finding method is described in Section 3.2. Performance evaluation and experimental results are discussed in Section 3.3, followed by a conclusion in Section 3.4.

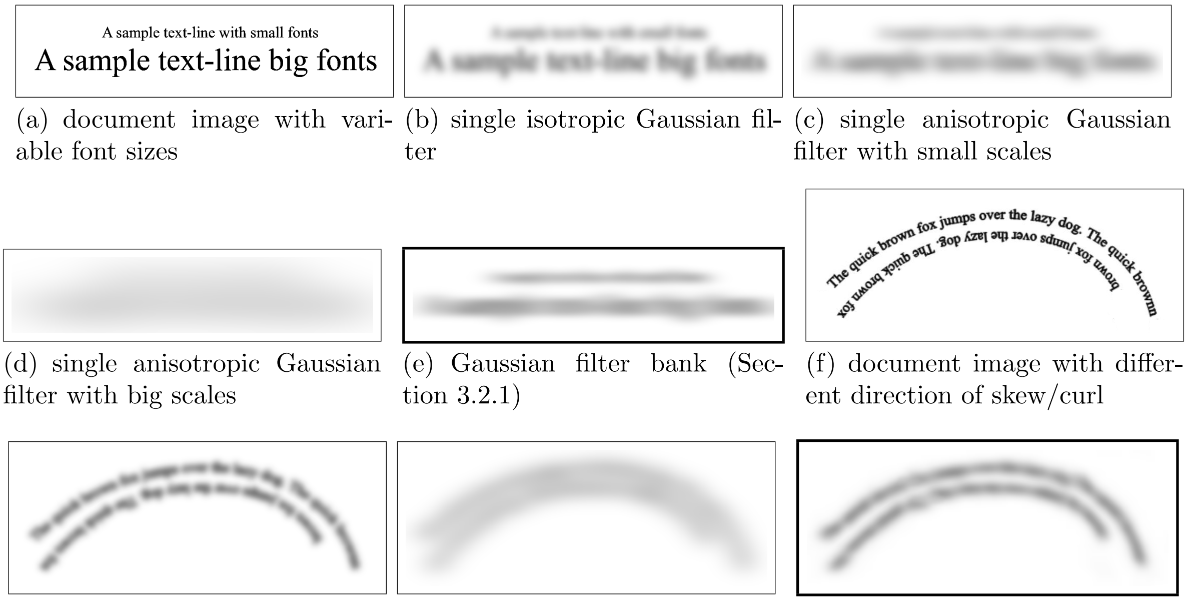
3.2 The Generic Text Line Finding Algorithm

The generic text line finding algorithm consists of two main steps: (i) text lines structure enhancement using matched filtering approach, and (ii) text line extraction using ridge detection over the enhanced image. Section 3.2.1 describes the motivation behind using matched filtering for text lines structure enhancement and its technical details. Similarly, section 3.2.2 describes the reason for using ridge detection method for the first time in document image processing for finding text line regions on the enhanced/smoothed image along with technical details. Section 3.2.3 presents approach of labeling text lines using detected ridges.

3.2.1 Step 1: Text line Structure Enhancement

Matched filtering is a very popular approach in computer vision especially for blood vessel detection in retinal images [CCK$^+$89, HKG00, ARQA07], fingerprint image enhancement [GN88,SAJ10], and image smoothing and segmentation [OCDK04,BKMA10,LW06]. Matched filtering is the concept of using a bank/set of filters, instead of one, for a given data processing task, such that a set of filters is applied to each pixel of an input image, and the maximum filter response at each pixel is selected for the output image. Here, Gaussian filter bank smoothing, which is motivated by the concept of matched filtering approach, is used for enhancing text lines structure in document images. The main motivation behind using the Gaussian filter bank smoothing instead of a single Gaussian filter is described below.

Document images contain lots of fine-scale details like character strokes, punctuation


A sample text-line with small fonts
A sample text-line big fonts
The quick brown fox jumps over the lazy dog. The quick brown
brown fox jumps over the lazy dog. The quick brown fox

(a) document image with variable font sizes (b) single isotropic Gaussian filter (c) single anisotropic Gaussian filter with small scales

(d) single anisotropic Gaussian filter with big scales (e) Gaussian filter bank (Section 3.2.1) (f) document image with different direction of skew/curl

(g) single anisotropic Gaussian filter with small scale (h) single anisotropic Gaussian filter with big scale (i) Gaussian filter bank (Section 3.2.1)

Figure 3.2: Different possible ways of text lines structure enhancement/smoothing in document images and their effects. Single isotropic Gaussian filter smoothing, where $\sigma_x = \sigma_y$ in Equation 3.2; the structure of text line is completely lost (Figure 3.2(b)), because text lines are horizontal/vertical in nature. Single anisotropic Gaussian filter smoothing, where $\sigma_x > \sigma_y$ in Equation 3.2; also the details of text lines structure is lost (Figure 3.2(c), 3.2(d), 3.2(g), and 3.2(h)) mainly because document images consists of multiple font sizes and/or multiple skew/curl. Multi-scale, multi-orientation Gaussian filter bank smoothing (Section 3.2.1) enhanced text lines structure well without mixing them with their neighboring text lines (Figure 3.2(e) and 3.2(i)).

marks, dots etc. Smoothing of a document image can transform its text lines structure into a high level coarse lines structure by blending their inter-line white spaces with fine-scale details. This type of transformation is referred to as *text lines structure enhancement*.

In different scripts, text lines are either arranged horizontally (like English, Arabic, etc.) or vertically (like Japanese). Therefore, a single isotropic Gaussian filter neither with a small nor with a big value of standard deviation (σ) is suitable for enhancing text line structure (as shown in Figure 3.2(b)). For text line structure enhancement, a better choice is to use an anisotropic Gaussian filter, but with different pair of values of standard deviations depending upon the targeted document's class, such as $\sigma_x > \sigma_y$ for horizontally aligned text lines and $\sigma_x < \sigma_x$ for vertically aligned text lines. Based on this consideration, even a single pair of σ_x and σ_y values are not sufficient for a collection of both horizontally and vertically aligned scripts.

A document image may also consist of a diversity of font sizes (as shown in Figure 3.2(a)), different text line orientations (as shown in Figure 3.2(f)), and variable spaces between characters, words and text lines. In such cases, each foreground pixel may have specific scales for x- and y-axis standard deviations and orientation with respect to neighboring white spaces. In such a case, a pair of small values of σ_x and σ_y can only enhance text lines structure with small fonts but not with big fonts (as shown in Figure 3.2(c)), or a pair of big values of σ_x and σ_y can enhance text lines structure of big fonts; for the case of text lines with small fonts, the details are mixed with neighboring text lines (as shown in Figure 3.2(d)). Therefore, even if all text lines are either aligned horizontally or vertically, a single pair of σ_x and σ_y values is not sufficient for text lines structure enhancement in the presence of multiple font sizes. Furthermore, as mentioned above, document images may also contain multiple skew/curl (such cases mainly occur in camera-captured, warped document images as shown in Figure 3.2(f)), and therefore a single anisotropic Gaussian filter with horizontal orientation can not enhance the structure of multi-orientation curled text lines (as shown in Figure 3.2(g) and 3.2(h)). These problems of text line structure enhancement under the cases of multiple font sizes and/or multiple text orientations can be solved by applying the matched filtering concept of multi-scale, multi-orientation anisotropic Gaussian filter bank smoothing (described in the following section). The results of text line structure enhancement using Gaussian filter bank are shown in Figure 3.2(e) and Figure 3.2(i). It is clearly visible in these figures that Gaussian filter bank properly enhances the structure of text lines in the presence of different font sizes and different directions of curl, respectively. It is the main motivation of using Gaussian filter bank smoothing approach for text lines structure enhancement in document images for text line detection. The technical details of Gaussian filter bank smoothing techniques is explained in the following section.

Anisotropic Gaussian Filter Bank Smoothing

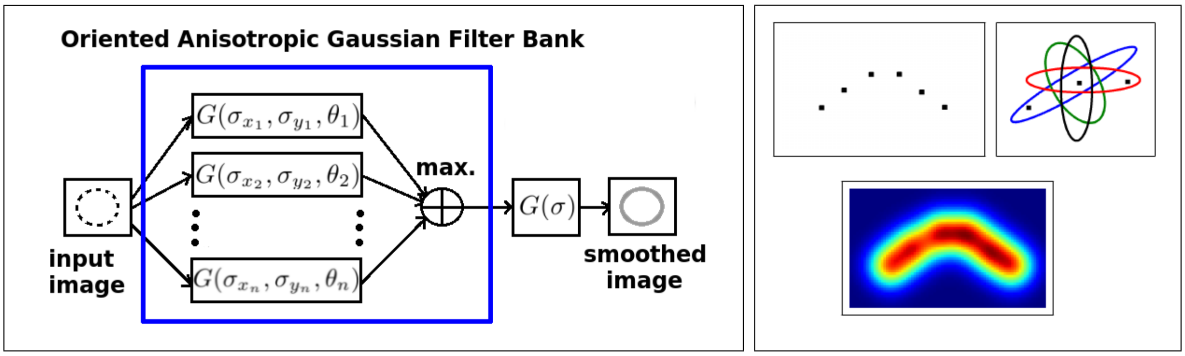
The formula of isotropic Gaussian filter and oriented anisotropic Gaussian filter is given in Equation 3.1 and 3.2, respectively.

$$G(x, y;\, \sigma) = \frac{1}{2\pi\sigma^2} exp\{-\frac{1}{2}\frac{(x^2+y^2)}{\sigma^2}\} \tag{3.1}$$

$$G(x,y,\sigma_x,\sigma_y,\theta)=\frac{1}{2\pi\sigma_x\sigma_y}exp\{-\frac{1}{2}(\frac{(x\cos\theta+y\sin\theta)^2}{{\sigma_x}^2}+\frac{(-x\sin\theta+y\cos\theta)^2}{{\sigma_y}^2})\} \quad (3.2)$$

Isotropic Gaussian filer consist of a single well-defined parameter for defining standard deviation (σ). Oriented anisotropic Gaussian filter consists of three well-defined parameters; σ_x: x-axis standard deviation, σ_y: y-axis standard deviation and θ: orientation angle. In contrast to isotropic Gaussian filter, oriented anisotropic Gaussian filter is a more general form and it can also be transformed into isotropic Gaussian filter by defining $\sigma_x = \sigma_y$ and $\theta = 0$.

For generating a bank of oriented anisotropic Gaussian filters, the ranges are first defined for these parameters: σ_x, σ_y and θ. A suitable range for θ can be $-45°$ to $45°$, that can cover all possible line orientations. The ranges for σ_x ($w_{start} \rightarrow w_{end}$) and σ_y ($h_{start} \rightarrow h_{end}$) can be selected relatively and automatically for a binary document image using its connected components statistics such as median width (W_{med_cc}) and median height (H_{med_cc}), like: $w_{start} = w_{weight} \times W_{med\_cc}$, $w_{end} = (w_{weight} + w_{weight\_offset}) \times W_{med\_cc}$, $h_{start} = h_{weight} \times H_{med\_cc}$, and $h_{end} = (h_{weight} + h_{height\_offset}) \times H_{med\_cc}$. The values of the following free parameters w_{weight}, h_{weight}, w_{weight_offset}, and h_{height_offset} can be chosen empirically. For grayscale documents, the absolute values can be selected empirically for the ranges of σ_x ($w_{start} \rightarrow w_{end}$) and σ_y ($h_{start} \rightarrow h_{end}$). After defining these ranges, a set of filters is generated for different possible combinations of σ_x, σ_y and θ from their predefined ranges. This set of oriented anisotropic Gaussian smoothing filters is then applied to each pixel of an input image. The maximum filter response for a particular pixel corresponds to the pixel's local orientation and scales of x- and y-axis standard deviations. Therefore, the maximum filter response at each pixel is selected for the smoothed or enhanced text lines image. For blending the fine discontinuities in the smoothed image, the smoothed image can be finally processed by an isotropic Gaussian filter with a small value of standard deviation. The block diagram of Gaussian filter bank smoothing is shown in Figure 3.3(a). A simple example image to illustrate the concept of Gaussian filter bank smoothing is also shown in Figure 3.3(b). The algorithm of text lines structure enhancement method using oriented anisotropic Gaussian filter bank smoothing is shown in Figure 3.4(a). For a sample document image (Figure 3.5(a)), the result of anisotropic Gaussian filter bank smoothing is shown in Figure 3.5(b).


Oriented Anisotropic Gaussian Filter Bank
$G(\sigma_{x_1}, \sigma_{y_1}, \theta_1)$
$G(\sigma_{x_2}, \sigma_{y_2}, \theta_2)$
$G(\sigma_{x_n}, \sigma_{y_n}, \theta_n)$
max.
$G(\sigma)$
input image
smoothed image

(a) Block diagram of oriented anisotropic Gaussian filter bank smoothing

(b) Illustration

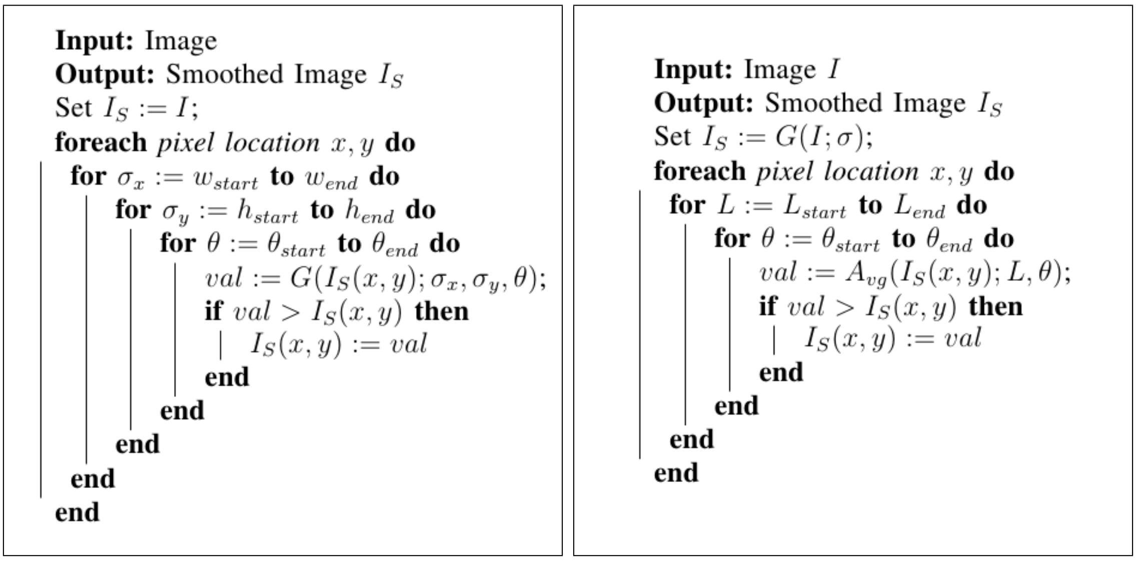
Figure 3.3: a) Processing flow of image smoothing using oriented anisotropic Gaussian filter bank approach. b) Example image for illustrating the basic principle of multi-orientation, multi-scale Gaussian filter bank smoothing; a filter bank over a pixel; result of filter bank smoothing.

A New Line Filter Bank Smoothing Technique

As mentioned above, anisotropic Gaussian filter bank smoothing is suitable for text line structure enhancement. However, it takes a large number of computational operations for a large number of filters. In order to overcome this problem, a novel concept of *line averaging filter bank smoothing* is introduced here for enhancing text line structure. The line averaging filter bank smoothing requires fewer computational operations for a large number filters as compared to anisotropic Gaussian filter bank smoothing.

A novel principle is presented here which states that "an oriented anisotropic Gaussian filter (Equation 3.2) can be approximated by a convolution of an isotropic Gaussian filter (Equation 3.1) and a simple *line averaging filter*". The main concept of line averaging filter bank smoothing is based on that novel principal. Here, the line averaging filter is defined as a function of length (L pixels) and orientation (θ degrees as slope) of a line, and it works as follows. For a pixel (x, y) in the smoothed image, the center point of a line filter is first placed over the pixel. By doing that, the line filter coincides with that pixel and its neighboring pixels. This way, the line averaging filter simply returns an average intensity value corresponding to all of these pixels. The line averaging filter is represented as $A_{vg}(x, y;\ L, \theta)$.

According to the presented principle, for approximating the smoothing affect of an oriented anisotropic Gaussian filter ($G(x, y;\ \sigma_x, \sigma_y, \theta)$, as show in Equation 3.2) for an image, the image is first smoothed by an appropriate isotropic Gaussian filter ($G(x, y;\ \sigma)$), and then the smoothed image is processed by an appropriate line averaging filter ($A_{vg}(x, y;\ L, \theta)$).



```
Input: Image
Output: Smoothed Image $I_S$
Set $I_S := I$;
foreach pixel location $x, y$ do
    for $\sigma_x := w_{start}$ to $w_{end}$ do
        for $\sigma_y := h_{start}$ to $h_{end}$ do
            for $\theta := \theta_{start}$ to $\theta_{end}$ do
                $val := G(I_S(x, y); \sigma_x, \sigma_y, \theta)$;
                if $val > I_S(x, y)$ then
                    $I_S(x, y) := val$
                end
            end
        end
    end
end
```

(a) Anisotropic Gaussian filter bank smoothing

```
Input: Image $I$
Output: Smoothed Image $I_S$
Set $I_S := G(I; \sigma)$;
foreach pixel location $x, y$ do
    for $L := L_{start}$ to $L_{end}$ do
        for $\theta := \theta_{start}$ to $\theta_{end}$ do
            $val := A_{vg}(I_S(x, y); L, \theta)$;
            if $val > I_S(x, y)$ then
                $I_S(x, y) := val$
            end
        end
    end
end
```

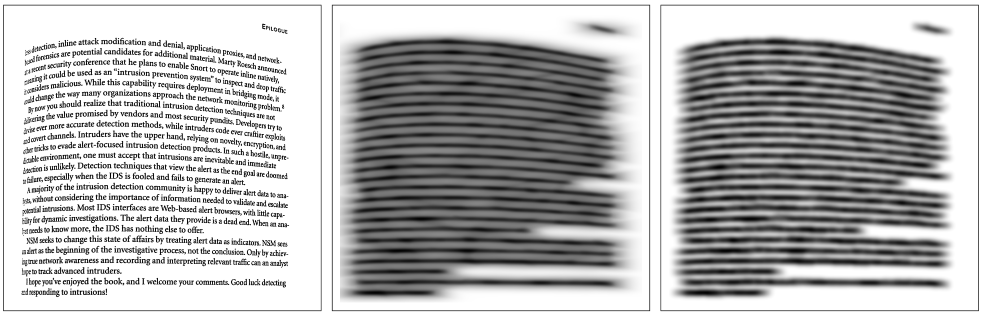
(b) Line averaging filter bank smoothing

Figure 3.4: Algorithms of text lines structure enhancement using i) multi-orientation, multi-scale anisotropic Gaussian filter bank smoothing. and ii) line averaging filter bank smoothing.

Some approximation examples are shown in Figure 3.6.

Using the presented convolution of isotropic Gaussian filter with line averaging filter, the main concept of the line filter bank smoothing technique is described as follows: an input image is first smoothed by an isotropic Gaussian filter with a predefined value of standard deviation (σ). For grayscale document image the value of σ can be chosen empirically. For binary image, the value of σ can be selected relatively with respect to median height of connected components (H_{med_cc}) such that $\sigma = \sigma_{weight} \times H_{med\_cc}$, where the value of σ_{weight} can be chosen empirically. After smoothing image by applying isotropic Gaussian filter, a set of line averaging filters is generated with varying lengths (L) and slopes (θ). Similar to anisotropic Gaussian filter bank smoothing, a suitable range of θ can be $-45°$ to $45°$ for generating a bank of line averaging filter. For grayscale documents, the absolute values can be selected empirically for defining the range of L ($l_{start} \rightarrow l_{end}$). It can be selected relatively and automatically for a binary document image using median width of connected components (W_{med_cc}), like: $l_{start} = l_{weight} \times W_{med\_cc}$, $l_{end} = (l_{weight} + l_{weight_{offset}}) \times W_{med\_cc}$, where the values of l_{weight} and $l_{weight_{offset}}$ can be set empirically.

After generating the set of line averaging filters from predefined ranges of θ and L, it is applied to the smoothed image, and for each pixel the maximum filter response is



EPILOGUE

less detection, inline attack modification and denial, application proxies, and network-
based forensics are potential candidates for additional material. Marty Roesch announced
at a recent security conference that he plans to enable Snort to operate inline natively,
meaning it could be used as an "intrusion prevention system" to inspect and drop traffic
it considers malicious. While this capability requires deployment in bridging mode, it
could change the way many organizations approach the network monitoring problem.[8]
By now you should realize that traditional intrusion detection techniques are not
delivering the value promised by vendors and most security pundits. Developers try to
devise ever more accurate detection methods, while intruders code ever craftier exploits
and covert channels. Intruders have the upper hand, relying on novelty, encryption, and
other tricks to evade alert-focused intrusion detection products. In such a hostile, unpre-
dictable environment, one must accept that intrusions are inevitable and immediate
detection is unlikely. Detection techniques that view the alert as the end goal are doomed
to failure, especially when the IDS is fooled and fails to generate an alert.
A majority of the intrusion detection community is happy to deliver alert data to ana-
lysts, without considering the importance of information needed to validate and escalate
potential intrusions. Most IDS interfaces are Web-based alert browsers, with little capa-
bility for dynamic investigations. The alert data they provide is a dead end. When an ana-
lyst needs to know more, the IDS has nothing else to offer.
NSM seeks to change this state of affairs by treating alert data as indicators. NSM sees
an alert as the beginning of the investigative process, not the conclusion. Only by achiev-
ing true network awareness and recording and interpreting relevant traffic can an analyst
hope to track advanced intruders.
I hope you've enjoyed the book, and I welcome your comments. Good luck detecting
and responding to intrusions!

(a) Camera-captured document (b) Anisotropic Gaussian filter bank smoothing (c) Line averaging filter bank smoothing

Figure 3.5: A sample camera-captured document image and its smoothed text lines results using both anisotropic Gaussian filter bank and line filter bank smoothing methods and their corresponding detected ridges. (a) sample camera-captured image, (b) result of anisotropic Gaussian filter bank smoothing, (c) result of line filter bank smoothing.

selected for the resulting smoothed image. The resulting image is further smoothed by an isotropic Gaussian filter with a small value of standard deviation. The block diagram of the filter bank smoothing technique is shown in Figure 3.7(a). A simple example image to illustrate the concept of line averaging filter bank smoothing is also shown in Figure 3.7(b). In contrast to anisotropic Gaussian filter bank smoothing technique, it is interesting to note that the line filter bank smoothing method just requires a single isotropic Gaussian filter followed by a set of line averaging filters. The algorithm of text lines structure enhancement method using line filter bank smoothing is shown in Figure 3.4(b).

A sample camera-captured document and its corresponding smoothed text lines results for anisotropic Gaussian filter bank smoothing and the new line filter bank smoothing are shown in Figure 3.5. It is clearly visible in Figure 3.5 that both of the smoothing techniques enhances the text lines structure well. It is, however, important to note that the outputs of both smoothing methods may not be exactly the same (at pixel level) because of their different smoothing techniques. The theoretical aspects of their differences is out of scope of this chapter. In performance evaluation section (Section 3.3), the performance of text line detection for both smoothing techniques (with different possible values of their free/tunable parameters) are evaluated.

In order to compare the computational complexities of the filter bank smoothing approaches (oriented anisotropic Gaussian filter bank and the newly presented oriented line averaging filter bank), their basic implementations are considered without any opti-

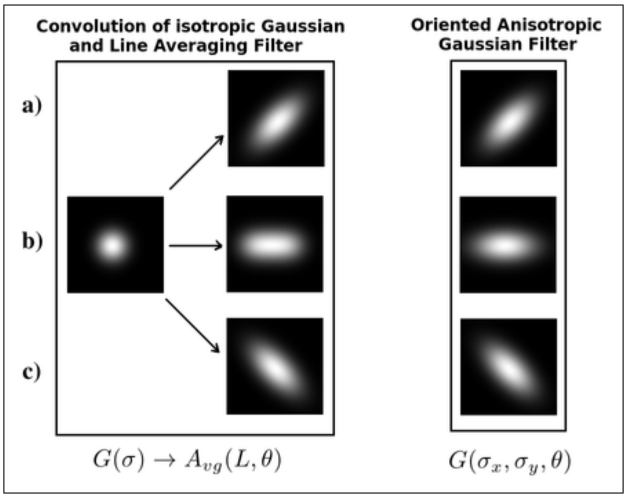

Convolution of isotropic Gaussian and Line Averaging Filter
Oriented Anisotropic Gaussian Filter
a)
b)
c)
$G(\sigma) \rightarrow A_{vg}(L,\theta)$
$G(\sigma_x,\sigma_y,\theta)$

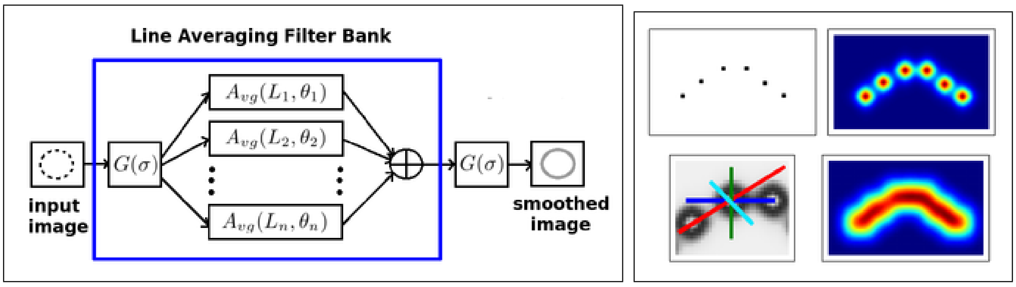
Figure 3.6: Sample results for the approximation of an oriented anisotropic Gaussian filter ($G(\sigma_x, \sigma_y, \theta)$) using the presented linear combination of an isotropic Gaussian filter ($G(\sigma)$) and a line averaging filter ($A(L, \theta)$). a) $\theta = 45°$, b) $\theta = 0°$, c) $\theta = -45°$.

mization. Computational complexity of anisotropic Gaussian filter bank, with F number of oriented anisotropic Gaussian filters of $W \times W$ window size, for an $N \times N$ image is equal to $O(F \times W^2 \times N^2)$. Similarly, computational complexity of line averaging filter bank, with an isotropic Gaussian filter of $W \times W$ window size and a set of F number of oriented line averaging filters of L pixels, for an $N \times N$ image is equal to $O(W^2 \times N^2 + F \times L \times N^2)$. Therefore, a large number of computational operations are required for oriented anisotropic Gaussian filter bank than the new line averaging filter bank.

Sample document images and their corresponding smoothed text lines images are shown in Figure 3.8. In contrast to a single isotropic or a single anisotropic Gaussian smoothing, the filter bank smoothing, either anisotropic Gaussian filter bank smoothing or line filter bank smoothing, enhances text lines structure well. After text line enhancement, text lines regions in the smoothed image are detected by applying ridge detection method. The following section describes the motivation behind using ridge detection method for text line detection and its technical details.

3.2.2 Step 2: Text Line Detection

Ridge detection approach has been widely used in computer vision for representing shapes of objects and producing symbolic descriptions of significant features [Ril87b, EGM+94,



(a) Block diagram of isotropic Gaussian filter with line averaging filter bank

(b) Illustration

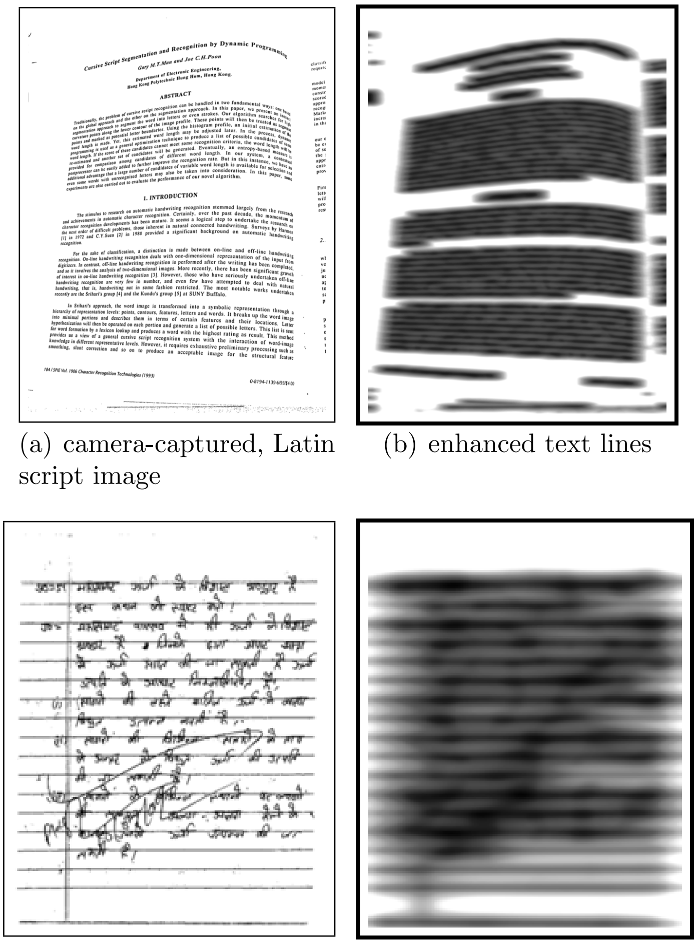
Figure 3.7: a) Processing flow of image smoothing using the presented line filter bank smoothing technique. b) Example image for illustrating the basic principle of lien averaging filter bank smoothing, image smoothing using an isotropic Gaussian filter, illustration of multi-orientation, multi-scale line averaging filters over a point, result of the line filter bank smoothing technique.

Dam99]. A basic ridge detection function can be defined as follows: consider a 3D image, such as x-coordinate, y-coordinate and intensity value for each pixel. The ridge detection is a function of these three variables which results in a set of curves, where the points on the curves are the local maximum of the ridge detection function in one dimension, i.e. intensity.

In case of document images, each enhanced text lines in the smoothed document image as shown in Figure 3.8 can be interpreted as ridge like structure, where the imaginary line that passes through the center of each enhanced text line represents ridge. Therefore, text line regions (these are the lines passing through the center of text lines) can be detected in the smoothed image using ridge detection technique, where each continuous/connected ridge can represent a segmented/detected text line. This is the main motivation of using ridge detection technique for the first time in document image processing for text line extraction. Here, a ridge detection method is used that is based on standard ridge detection techniques [Ril87b, EGM+94, Dam99]. The ridge detection method is described in detail below.

Consider an image I for which intensity value at the location (x, y) is represented as $I(x, y)$. The ridge detection method is based on the analysis of gradient vectors (∇I) and greatest downward curvatures (symbol $\check{I}$ is chosen for representing the greatest downward curvatures). For a pixel $I(x, y)$, gradient vector $(\nabla I(x, y))$ is defined in Equation 3.3:

$$\nabla I(x, y) = \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right) \tag{3.3}$$



(a) camera-captured, Latin script image

(b) enhanced text lines

(c) scanned, handwritten, Hindi script image

(d) enhanced text lines

Figure 3.8: Example images of typed-text, camera-captured, Latin script and scanned, handwritten, Hindi script document images and their corresponding enhanced/smoothed text lines images.

For a pixel $I(x, y)$, greatest downward curvature $(\check{I}(x, y))$ is defined as the local direction in which surface curves the most downward from the tangent plane. Precisely stated, a local greatest downward curvature is defined as the direction vector of the minimum second directional derivative at a given point, which can be estimated by using Hessian matrix (H). The definition of Hessian matrix at point $I(x, y)$ is given in Equation 3.4.

$$H(x,y) = \begin{pmatrix} \frac{\partial^2 I(x,y)}{\partial x^2} & \frac{\partial^2 I(x,y)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y)}{\partial x \partial y} & \frac{\partial^2 I(x,y)}{\partial y^2} \end{pmatrix} \tag{3.4}$$

Let $\lambda_1(x, y)$ and $\lambda_2(x, y)$ be the two Eignevalues of $H(x, y)$. Define $\lambda_s(x, y)$ as the minimum Eignevalue i.e. $\lambda_s(x, y) = \min(\lambda_1(x, y), \lambda_2(x, y))$ and $\lambda_b(x, y)$ as the maximum Eignevalue i.e. $\lambda_b(x, y) = \max(\lambda_1(x, y), \lambda_2(x, y))$. Let $\mathbf{e}_s(x, y)$ be the Eignevector corresponding to $\lambda_s(x, y)$ and $\mathbf{e}_b(x, y)$ be the Eignevector corresponding to $\lambda_b(x, y)$. The greatest downward curvature at that point $(\check{I}(x, y))$ is defined as the Eignevector corresponding to the smaller Eignevalue as shown in Equation 3.5.

$$\check{I}(x, y) = \frac{\mathbf{e}_s(x, y)}{\mid \mathbf{e}_s(x, y) \mid} \tag{3.5}$$

A simple example image to illustrate the concept of ridge detection is shown in Figure 3.10(a), where a black pixel corresponds to the largest intensity value and a white pixel corresponds to the smallest intensity value. This image is composed of two objects: one object contains horizontally aligned ridge points in the middle of that object and another one contains an isolated point. The gradient vectors ∇I, greatest downward curvatures $\check{I}$, small negative Eignevalues and small positive Eignevalues are shown in Figure 3.10(b) with red, blue, yellow, and green colors, respectively. A close examination of gradient vectors, greatest downward curvature vectors, and the signs of small Eignevalues at ridge and non-ridge points in Figure 3.10(b) describes the following facts. The gradient vectors move away from the ridge's points and the greatest downward curvatures point perpendicularly to the gradient vectors in the overall neighborhood of the ridges. Therefore, if both gradient and greatest downward curvature vectors at a point are perpendicular to each other then the point is considered as a candidate ridge's point. If the small Eignevalue at that point is negative i.e. $e_s(x, y) < 0$, the point is selected as ridge's point. Based on the above description, the ridge's points are the set of points which satisfy the following conditions: $\nabla I.\check{I} = 0$ and $e_s < 0$. The dot product of ∇I and $\check{I}$ is zero where these vectors are perpendicular to each other and $e_s < 0$ ensures that the points are ridges points, not trough points. The ridge detection using only these conditions can also produce ridges points around an isolated point (like salt-and-pepper noise), which is shown in Figure 3.10(c). Such type of noisy ridges is removed by adding another condition such that a point is a ridge's point if at least one of its 4-neighboring points $((x+dx, y+dy)$, i.e. either $(x+1, y+0)$, $(x-1, y+0)$, $(x+0, y+1)$, or $(x+0, y-1))$ is also a ridge's point. For checking either a point is a ridge's point or not, all of these three conditions are combined together that results into a set of rules that are shown in Equation 3.6. For detecting ridge points in the smoothed image, each point $I(x, y)$ is analyzed for a ridge's point at a time with one of its 4-neighboring point $(I(x + dx, y + dy))$ based on the rules mentioned in Equation 3.6. The ridge detection algorithm is shown in

$$R(x,y,dx,dy) = \begin{cases} 1 \quad \text{if} & \begin{cases} 1.\ e_s(x,y) < 0 \ \text{ and } \ \|e_s(x,y)\| > \|e_b(x,y)\| \\ 2.\ e_s(x+dx,y+dy) < 0 \ \text{ and} \\ \quad \|e_s(x+dx,y+dy)\| > \|e_b(x+dx,y+dy)\| \\ 3.\ \nabla I(x,y).\nabla I(x+dx,y+dy) < \check{I}(x,y).\check{I}(x+dx,y+dy) \\ 4.\ \nabla I(x,y).\check{I}(x,y) * \nabla I(x+dx,y+dy).\check{I}(x+dx,y+dy)* \\ \quad \check{I}(x,y).\check{I}(x+dx,y+dy) < 0 \end{cases} \\ 0 \quad \text{else} & \end{cases} \tag{3.6}$$

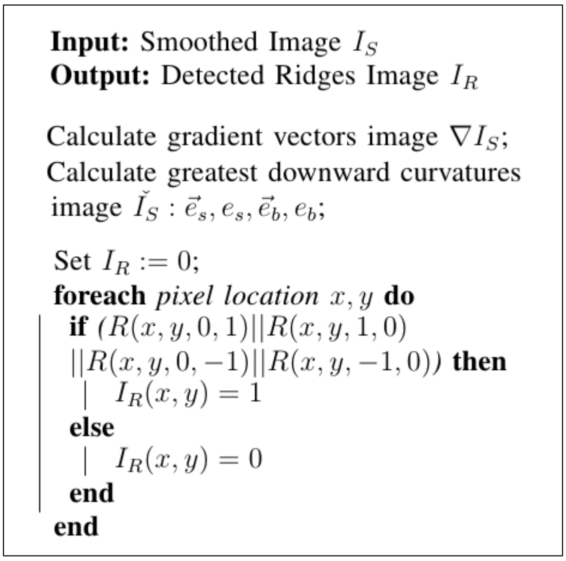
Figure 3.9.

For the example image in Figure 3.10(a), detected ridges are shown in Figure 3.10(d). The detected ridges over a sample smoothed document image (Figure3.10(f)) is also shown in Figure 3.10(g). It i interesting to note in Figure 3.10(g) that, each connected ridge line represents the complete region of a particular text line. The detected ridges of the enhanced/smoothed text line images of Figure 3.8(b) and 3.8(d) are also shown in Figures 3.11(a) and 3.11(c), respectively, where the detected ridges are mapped over the input images for viewing clarity. Detected ridges for some of the challenging cases, especially in grayscale document images, like smudge (Figure 3.12(a)) and bleed-through (Figure 3.12(e)) are also shown in the Figure 3.12(b) and 3.12(f), respectively. Even in such complex cases, most of the detected ridge represent segmented text lines regions, except some small noise ridges which mostly lie on document background. These noisy ridges can be removed by using size based filtering. It is visible in Figures 3.11(a), 3.11(c), 3.12(b), and 3.12(f) that, each ridge covers the complete region of a particular text line. The ridges based text line regions detection method can be applied equally on both binary and grayscale document images.

After detecting text lines regions, i.e. ridge detection over the smoothed image, the final labeling step, which assign a unique label to text associated with each particular text line using detected text lines regions (ridges), is described in the following section.

3.2.3 Step 3: Text Line Labeling

The text line labeling procedure using detected ridges is defined as follows. First of all, each connected ridge line is assigned a unique label. Some examples of labeled ridges with different colors are shown in Figure 3.11(a), 3.11(c), 3.12(b), and 3.12(f). It is important to note that, each ridge covers a complete region of a particular text line. Labeling of characters (connected components) in a document image with respect to their text lines association can be considered as a trivial task, such that each connected component in the



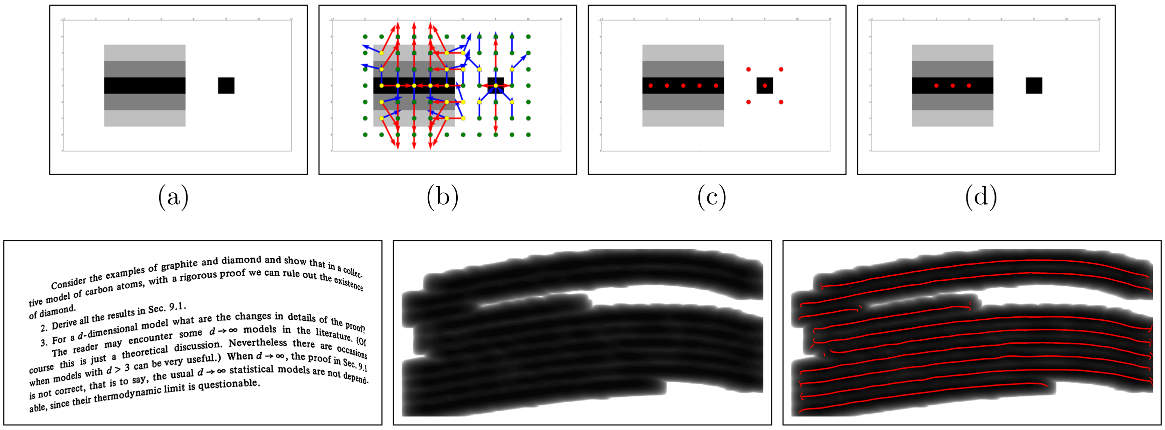
```
Input: Smoothed Image $I_S$
Output: Detected Ridges Image $I_R$

Calculate gradient vectors image $\nabla I_S$;
Calculate greatest downward curvatures
image $\check{I}_S : \vec{e}_s, e_s, \vec{e}_b, e_b$;

Set $I_R := 0$;
foreach pixel location $x, y$ do
    if $(R(x, y, 0, 1)||R(x, y, 1, 0)$
    $||R(x, y, 0, -1)||R(x, y, -1, 0))$ then
        $I_R(x, y) = 1$
    else
        $I_R(x, y) = 0$
    end
end
```

Figure 3.9: Algorithm of text line detection using ridge detection method. The formula for $R(x, y, dx, dy)$ is given in Equation 3.6.

document image is assigned the label of its overlapping ridge, and then each unlabeled connected component (like dots, punctuation marks that could not overlap with any ridge) are assigned the label of nearest neighbor (labeled) component. This type of trivial labeling is only possible for document images which do not contain overlapping and/or touching characters in-between text lines. However, free-style handwritten document images usually contain a lot of overlapping and/or touching components in-between text lines, as well as other challenging problems like smudge, bleed-through, etc. In such cases, more than one ridges can lie over a single connected component. For example, the whole text in the document image of Figure 3.11(c) is considered as single connected component (because of the page lining and inter-line touching text). Therefore, text line labeling under these challenging conditions is still an open problem. Here, a simple and effective text lines labeling algorithm is introduced for document images with overlapping and/or touching connected components, smudge, etc., which is defined as follows. If a single ridge lies over a single connected component then the connected component is assigned the label of that particular ridge. If more than one ridges lie over a single connected component, then a connected component is cut from the center between each of the two consecutive ridges in the orientation approximately similar to the orientation of consecutive ridges. The illustration of this cutting procedure for sample document image with inter-line touching is shown in Figure 3.13. By doing that, each segment of a particular connected component overlaps with a single ridge (as shown in Figure 3.13(b)),



Consider the examples of graphite and diamond and show that in a collec-
tive model of carbon atoms, with a rigorous proof we can rule out the existence
of diamond.
2. Derive all the results in Sec. 9.1.
3. For a d-dimensional model what are the changes in details of the proof?
The reader may encounter some $d \rightarrow \infty$ models in the literature. (Of
course this is just a theoretical discussion. Nevertheless there are occasions
when models with $d > 3$ can be very useful.) When $d \rightarrow \infty$, the proof in Sec. 9.1
is not correct, that is to say, the usual $d \rightarrow \infty$ statistical models are not depend-
able, since their thermodynamic limit is questionable.

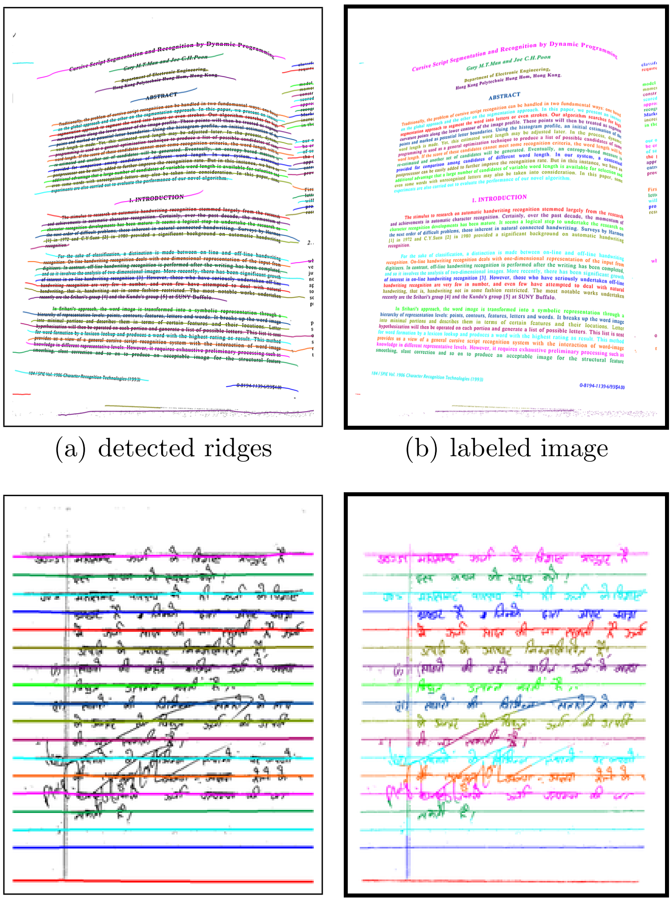
(a) (b) (c) (d)

(e) sample document image (f) smoothed document image (g) detected ridges

Figure 3.10: Illustration of ridge detection principle using an example image and its result on a sample document image. a) grayscale image where black pixels represent the maximum intensity and white pixel represents the minimum intensity value. b) ∇I (red arrows), $\check{I}$ (blue arrows), and signs of small Eigenvalues (yellow dots represent -ve sign and green dots represents +ve sign). c) detected ridges: each pixel is examined for ridge point in conjunction with its four neighboring pixels and a pixel is reported as ridge's point if the pixel itself and at least one of its neighboring pixel hold ridge criteria. d) curled text lines in camera-captured document image. e) smoothed/enhanced text line structure. f) detected ridge; it is important to note that, each connected ridge represent a complete region of a particular text line.

and therefore is assigned the label of that particular ridge as shown in Figure 3.13(c). Some examples of labeled text lines result using the above mentioned procedure are shown in Figures 3.11(b) and 3.11(d). Additionally, text line finding results of the ridge-based text line detection algorithm in the presence of some other challenging problems are also shown in Figure 3.14. As shown in these results, the ridge-based text lines extraction algorithm is robust to different problems such as small inter-line gaps, large amount of border noise, inter-line touching and overlapping components, curled text lines, etc.

As mentioned above, the presented smoothing and ridge detection method can also be directly applied to grayscale documents (as shown in Figure 3.12(b) and 3.12(f)), where each detected ridge corresponds to the region of a particular text line. However, so far no grayscale text line labeling method has been developed that is also an open challenging problem in grayscale document image processing. In order to cope with this problem, after detecting ridges, the grayscale input image is first converted into binarized image and then the detected ridges of the grayscale image are overlapped into its

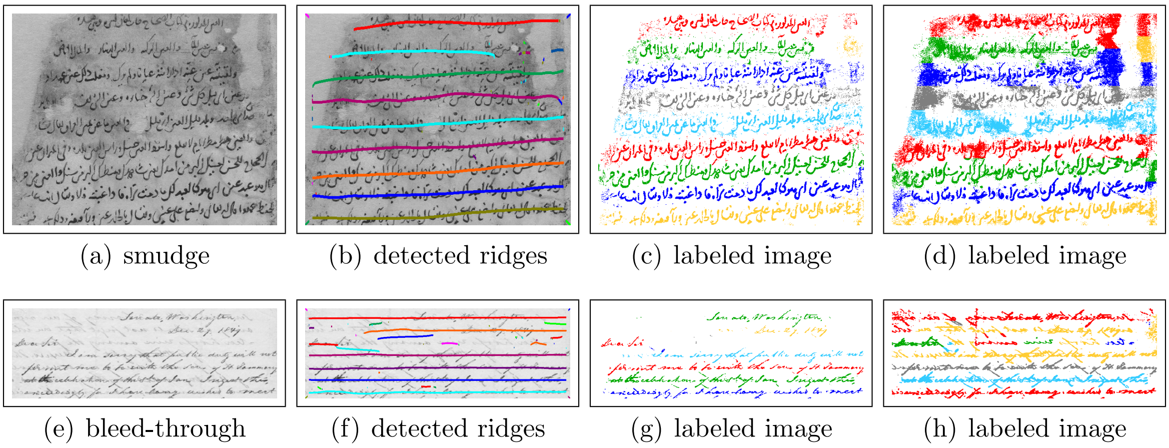


(a) detected ridges (b) labeled image

(c) detected ridges (d) labeled image

Figure 3.11: Text line labeling using detected ridges. a) and c): Detected ridges from enhanced text lines document images of Figure 3.8 are mapped over their corresponding original images. b) and d): Text line detection results where text line labeling is performed by using detected ridges.

corresponding binarized document image for labeling purpose. Sample results are shown in Figure 3.12(d) and 3.12(h). It is also important to note that, the same ridges, that are detected from a grayscale document image, may produce different text line labeling results for different qualities of the corresponding binarized images. For example, text line labeling under the challenging cases of smudge and bleed-through for two different qualities of their corresponding binarized images are shown in Figures 3.12(c) and 3.12(d), and Figures 3.12(g) and 3.12(h), respectively.


(a) smudge (b) detected ridges (c) labeled image (d) labeled image

(e) bleed-through (f) detected ridges (g) labeled image (h) labeled image

Figure 3.12: Ridge detection results over grayscale document images for challenging cases. These detected ridges are mapped over the two different qualities of corresponding binarized images for text line labeling.

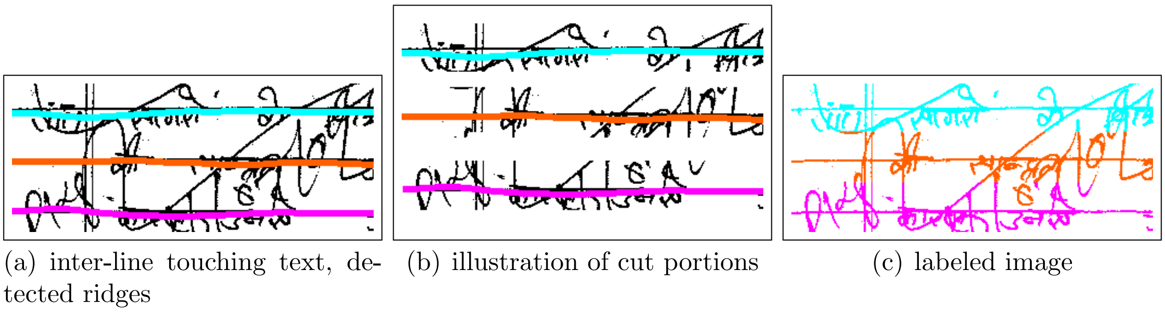
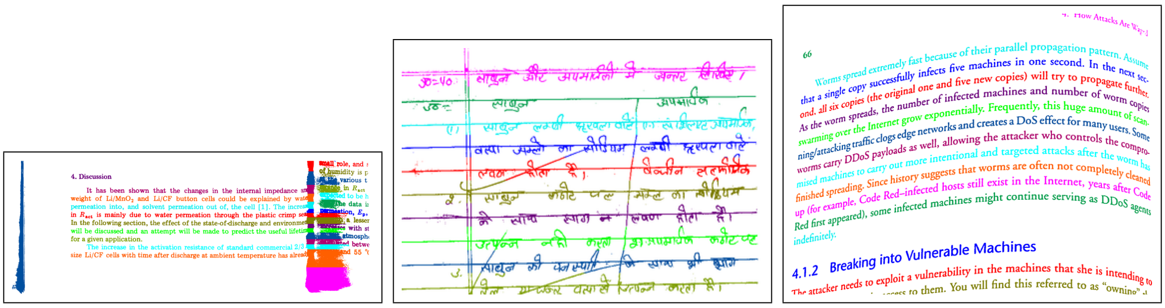

(a) inter-line touching text, detected ridges (b) illustration of cut portions (c) labeled image

Figure 3.13: Illustration of text line labeling method under the challenging cases of inter-line touching text.

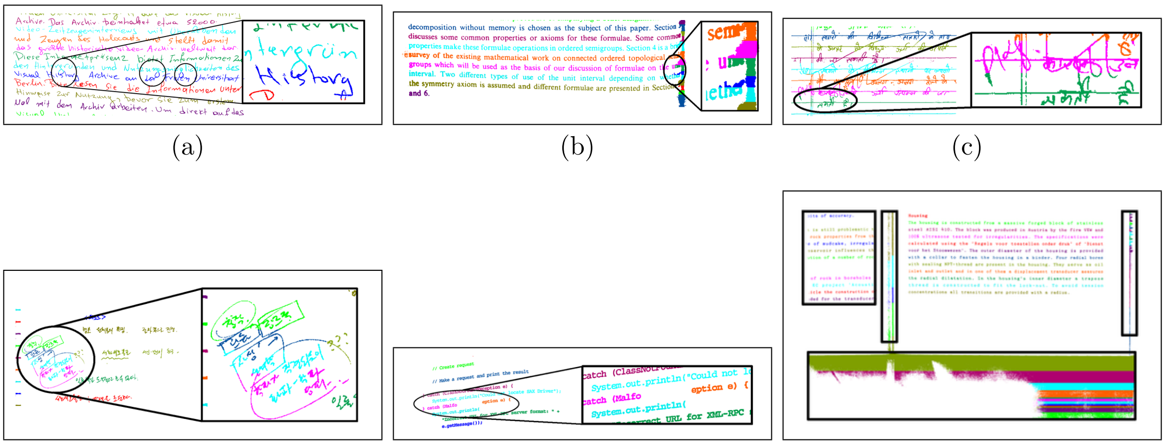
3.3 Performance Evaluation

The performance of the presented generic text line finding algorithm, that is also referred to as ridge-based method, is evaluated for different categories of standard datasets. These datasets include typed-text scanned document images (the UW-III dataset [GHHP97]), binary and grayscale camera-captured document images (the DFKI-I dewarping contest dataset [SB07]), free-style handwritten scanned document images (the ICDAR 2007 handwritten segmentation contest dataset [GAS07] and the UMD handwritten documents dataset [LZDJ08]), Arabic/Urdu scripts typed-text dataset [buk], and calligraphic German script (Fraktur) dataset [buk]. In these datasets, ground-truth is presented in color-coded image format, such that text lines in a ground-truth image are assigned unique labels and all the other components, like graphics, rulings, tables, text outside



(a) (b) (c)

Figure 3.14: Accurate text line finding results of the presented algorithm in the presence of challenging problems: free-style handwritten text lines with interline touching, noise and distortions, straight text lines merged with border noise, and curled text lines.



(a) (b) (c)

(d) (e) (f)

Figure 3.15: Conditions that are considered as errors based on the metrics defined in [SKB08b]. a-c) over-segmentations and under-segmentations errors because of wrong labels. d) over-segmentation and under-segmentation errors due to a large number of strokes. e) over-segmentation error that is a result of using small size x- and y-axis standard deviation ranges for Gaussian filter bank smoothing, and can be overcome by using appropriate large size ranges. f) False alarms: text outside page border and noise are also detected as text lines. These type of errors can be removed by using some preprocessing or post-processing step, like text line labeling within page border or document image cleaning.

page border etc., are treated as noise. Here, the performance of the ridge-based text line extraction method is compared with other state-of-the-art text line extraction methods.

As described in Section 3.2.1,the ridge-based text line detection method has two different forms: i) ridge detection with anisotropic Gaussian smoothing (having free parameters w_{weight}, h_{weight}, w_{weight_offset}, and h_{weight_offset}), ii) ridge detection with line filter bank smoothing (having free parameters σ_{weight}, l_{weight}, and l_{weight_offset}). Some experiments have been conducted for both of these different types of smoothing methods on a large variety of document images, and found that, for anisotropic Gaussian smoothing w_{weight} and h_{weight} are more critical than w_{weight_offset} and h_{weight_offset}, and for line filter smoothing σ_{weight} and l_{weight} are more critical than l_{weight_offset}. Based on these experiments, w_{weight_offset} and h_{weight_offset} are set equal to values 2 and 0.4, respectively, for Gaussian filter bank smoothing for evaluating its performance with respect to different values of w_{weight} and h_{weight} on each dataset separately as well as all documents together. Here, the w_{weight} can be set equal to any of the following values: 1, 2, 3, 4, 5 or 6, and h_{weight} can be set equal to any of the following values: 0.1, 0.2, 0.3, 0.4, 0.5 or 0.6. Similarly, for line filter smoothing, l_{weight_offset} is empirically set equal to 2 for evaluating its performance with respect to different values of σ_{weight} (0.25, 0.3 or 0.5) and l_{weight} (1, 2, 3, 4, 5 or 6) on each dataset separately as well as all documents together. These range of values are chosen empirically in order to balance a trade-off between text line finding accuracy and execution time speed. The range of θ for both smoothing methods can be set to 0 for straight text line documents, and for curled text line document images the range of θ can be set in-between ± 10.

This section is further divided into two subsections. Performance evaluation on different standard datasets and comparison with the state-of-the-art algorithms is presented in Section 3.3.1. In Section 3.3.2, execution time of the ridge-based text line detection algorithm with both different types of smoothing methods is analyzed.

3.3.1 Performance Evaluation and Comparison on Standard Datasets

Performance evaluation of the ridge-based text line finding algorithm is based on vectorial performance evaluation metrics that were presented in Shafait et al. [SKB08b], which are described in Section 2.5 of Chapter 2.

Here, the following standard datasets are selected that belong to different categories of document image classes for the evaluation of the ridge-based text line finding method and its comparison with other state-of-the-art methods: i) the UMD handwritten documents dataset [LZDJ08], ii) the ICDAR 2007 handwritten segmentation contest dataset [GAS07], iii) the UW-III dataset [GHHP97], iv) Arabic/Urdu script dataset [buk], v) Fraktur dataset [buk], vi) the DFKI-I 2007 dewarping contest dataset [SB07].

For the performance evaluation of the ridge-based method on a collection of diverse documents, all of the document images in these different datasets are combined into a single dataset, which contains 650 document images with 23763 text lines. This single dataset represents a collection of diverse document images. For finding the default values of free parameters of the ridge-based method that can be applied equally to different types of document images, the text line detection accuracy of the ridge-based method is evaluated with both anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different possible values of their free parameters (w_{weight} and h_{weight} for anisotropic Gaussian filter bank smoothing, and σ_{weight} and l_{weight} for line filter bank smoothing). Figure 3.16 shows the one-to-one text line finding accuracy ($P_{o2o}\%$) of the ridge-based text line finding algorithm for different values of their free parameters. According to Figure 3.16, the best text line detection accuracy (84.89%) for anisotropic Gaussian filter bank smoothing is achieved with $w_{weight} = 3$ and $h_{weight} = 0.4$, which can be considered as dataset-independent/default values of free parameters of anisotropic Gaussian filter bank smoothing. Similarly, the best text line detection accuracy (86.10%) for line filter bank smoothing is achieved with $\sigma_{weight} = 0.3$ and $l_{weight} = 5$, which can be considered as dataset-independent/default values of free parameters of line filter bank smoothing. It is also important to note that, both of the filter bank smoothing techniques achieved almost similar text line detection results. Therefore, it can also be conclude that the line filter bank smoothing technique approximates the affect of oriented Gaussian filter bank smoothing technique but with less number of computational operations, which was one of the main motivations of introducing the line filter bank smoothing.

Here, for each of the domain-specific dataset, the performance of the ridge-based method (with its dataset-dependent/optimized values of free parameters) is compared with different domain-specific stat-of-the-art methods.

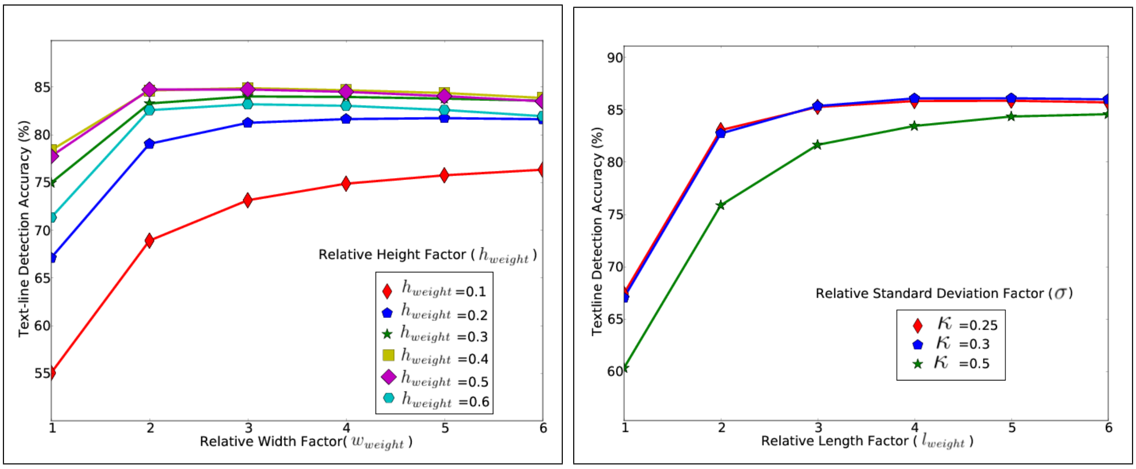
The UMD Handwritten Documents Dataset

The UMD handwritten document images dataset [LZDJ08] was developed by the Language and Media Processing Laboratory at the University of Maryland, which consists of around 300 document images of Chinese, Korean and Hindi scripts. Document images in this dataset consist of a lot of touching and overlapping characters, background

Table 3.1: Text line extraction accuracy of the ridge-based method (with both anisotropic Gaussian filter bank smoothing and line filter bank smoothing) on a large number of standard dataset that belongs to a diverse collection of documents and its comparison with a variety of domain-specific state-of-the-art methods by using performance evaluation metrics that are defined in [SKB08b]. For each dataset, the text line detection accuracy of the ridge-based method is better than the text line detection accuracies of the domain-specific state-of-the-art methods.

Dataset	**Method**	Performance Evaluation Metrics[a]								
		N_s	N_{o2o}	N_{falarm}	N_{useg}	N_{oseg}	$P_{ucomp}\%$	$P_{ocomp}\%$	$P_{mcomp}\%$	$P_{o2o}\%$
UMD [LZDJ08]	Adapted Levelset [LZDJ08]	6242	4595	527	757	864	8.04	7.97	0.0	52.85
(Docs: 300)	**Ridge-based (Aniso/Line)**	**7981**	**6063**	**759**	**680**	**915**	**7.04**	**9.44**	**1.29**	**69.74**
(N_g: 8694)		**8408**	**6461**	**1229**	**635**	**1000**	**6.6**	**9.97**	**0.32**	**74.32**
ICDAR07 [GAS07]	ILSP-LWSeg [GAS07]	1773	1713	-	-	-	-	-	-	96.73
(Docs: 80)	**Ridge-based (Aniso/Line)**	**1767**	**1719**	**1**	**31**	**25**	**1.13**	**1.41**	**0.0**	**97.06**
(N_g: 1771)		**1807**	**1731**	**3**	**22**	**27**	**1.19**	**1.41**	**0.0**	**97.74**
	X-Y cut [NSV92]	3836	2611	375	322	2204	5.06	53.24	7.76	72.63
Arabic/Urdu [buk]	RAST [SHKB06]	3564	3058	190	198	1202	4.7	30.13	3.23	85.06
(Docs: 45)	**Ridge-based (Aniso/Line)**	**3648**	**3373**	**437**	**72**	**129**	**1.97**	**2.25**	**0.95**	**93.83**
(N_g: 3595)		**3782**	**3377**	**516**	**59**	**158**	**1.64**	**2.42**	**1.06**	**93.94**
Fraktur [buk]	RAST [Bre02b]	2827	1545	0	133	193	2.95	5.97	1.28	90.38
(Docs: 22)	**Ridge-based (Aniso/Line)**	**2857**	**2760**	**6**	**6**	**56**	**0.21**	**1.46**	**0.32**	**98.01**
(N_g: 2816)		**2858**	**2761**	**11**	**4**	**53**	**0.14**	**1.46**	**0.28**	**98.05**
	Smearing [WCW82]	3281	2952	7637	69	25	1.05	0.66	19.60	77.77
UW-III [GHHP97]	RAST [Bre02b]	3812	3618	2398	97	60	2.24	1.58	0.58	95.31
(Docs: 100)	**Ridge-based (Ansio/Line)**	**3725**	**3566**	**1649**	**94**	**71**	**2.40**	**1.53**	**1.24**	**93.94**
(N_g: 3796)		**3879**	**3609**	**4641**	**128**	**184**	**3.29**	**3.45**	**0.08**	**95.07**
	Nearest-Neighbor [GPN07]	3293	2753	4264	103	241	3.20	7.02	0.03	89.07
DFKI-I [SB07]	Rule-Based [OLT$^+$10]	2924	2816	785	57	682	1.81	21.71	4.43	91.10
(Docs: 102)	**Ridge-based (Aniso/Line)**	**3032**	**2805**	**1762**	**125**	**86**	**3.66**	**2.39**	**0.65**	**90.75**
(N_g: 3091)		**3296**	**2882**	**2830**	**72**	**277**	**2.04**	**3.75**	**0.03**	**93.24**

[a] N_g:ground-truth components; N_s:segmented components; N_{o2o}:one-to-one matched components; $P_{o2o}\% = N_{o2o}/N_g$;N_{oseg}: over-segmentations;N_{useg}: under-segmentations;N_{ocomp}:over-segmented components; $P_{ocomp}\% = N_{ocomp}/N_g$; N_{ucomp}: under-segmented components; $P_{ucomp}\% = N_{ucomp}/N_g$; N_{mcomp}: missed components; $P_{mcomp}\% = N_{mcomp}/N_g$; N_{falarm}: false alarms;



(a) Ridge with anisotropic Gaussian filter bank smoothing

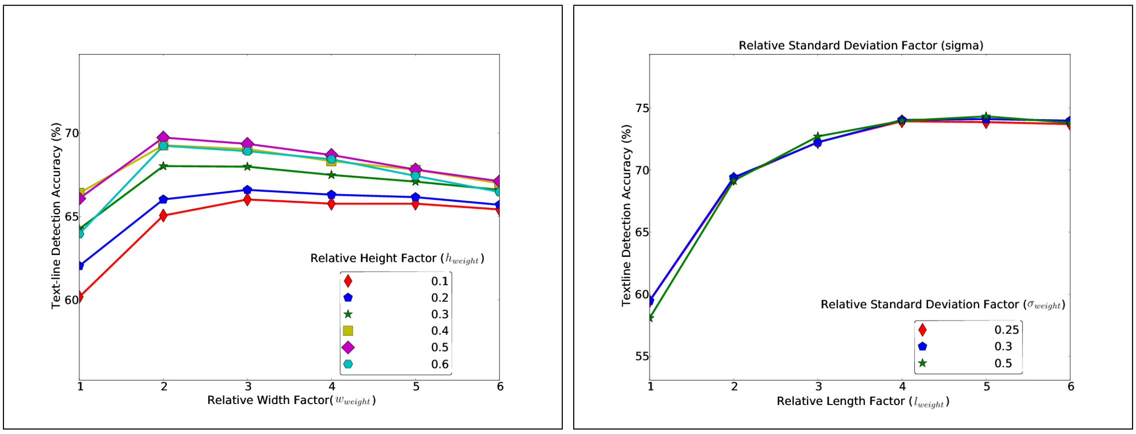
(b) Ridge with line filter bank smoothing

Figure 3.16: Collection of Diverse Documents: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on a diverse collection of document images dataset (650 document images with 23763 text lines).

noise (signatures, check marks, drawings, lined paper etc.), irregular layout, and multi-orientation text lines. Example images are shown in Figures 3.8(c), 3.14(b), 3.15(c) and 3.15(d). From Figure 3.17, the optimized values of free parameters for this dataset are: $w_{weight} = 2$ and $h_{weight} = 0.5$ for ridge detection with anisotropic Gaussian smoothing, and $\sigma_{weight} = 0.5$ and $l_{weight} = 5$ for ridge detection with line filter bank smoothing. The performance evaluation results of the ridge-based text line finding algorithm and adapted levelset algorithm of Li et al. [LZDJ08] are shown in Table 3.1, where the ridge-based method performs much better than the levelset method. A few examples of over-segmentation and under-segmentation errors for this dataset are shown in Figures 3.15(c) and 3.15(d).

The ICDAR 2007 Handwritten Segmentation Contest Dataset

The ICDAR 2007 handwritten segmentation contest dataset [GAS07] consists of 80 handwritten document images of English, French, German and Greek scripts. Document images in this dataset consist of irregular layout, multi-orientation text lines and overlapping and touching characters. Sample image is shown in Figures 3.15(a). From Figure 3.18, the optimized values of free parameters for this dataset are: $w_{weight} = 5$ and $h_{weight} = 0.4$ for



(a) Ridge with anisotropic Gaussian filter bank smoothing

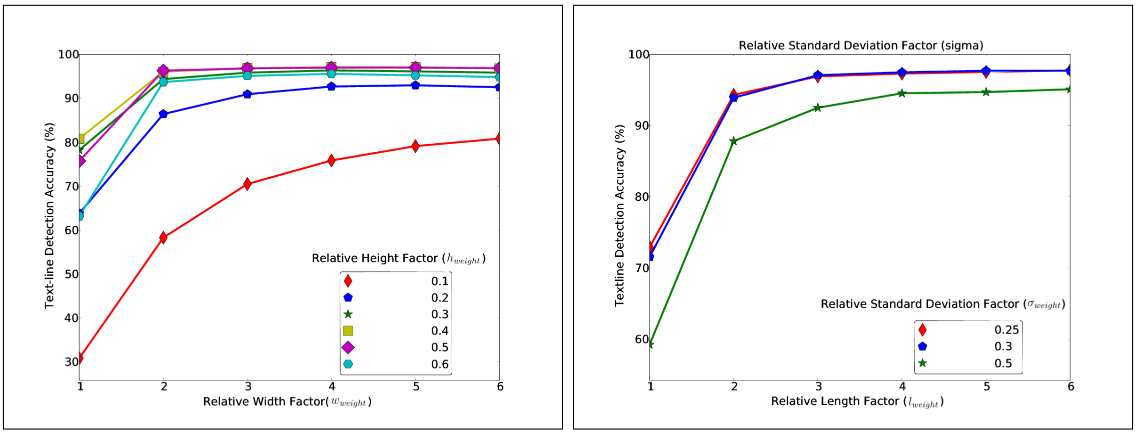
(b) Ridge with line filter bank smoothing

Figure 3.17: the UMD handwritten documents dataset: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on the UMD handwritten documents dataset [LZDJ08].

ridge detection with anisotropic Gaussian smoothing, and $\sigma_{weight} = 0.25$ and $l_{weight} = 6$ for ridge detection with line filter bank smoothing. This dataset was used in the ICDAR 2007 handwritten segmentation contest [GAS07] where 5 methods had participated. This contest used different performance evaluation metrics, but one of the main performance evaluation metrics is exactly similar to the P_{o2o}. The performance of the ridge-based algorithm is better than the winner of the ICDAR 2007 contest participants methods, as shown in Table 3.1. An over-segmentation error for an example image of this dataset is shown in Figure 3.15(a).

The UW-III Dataset

The University of Washington III (UW-III) typed-text scanned document images dataset [GHHP97] consists of scanned English script document images. A subset of 100 single column document images is selected for the performance evaluation. Document images in the UW-III dataset consist of straight text lines with interline touching and border noise. Example images are shown in Figures 3.14(a), 3.15(b) and 3.15(f). From Figure 3.19, the optimized values of free parameters for this dataset are: $w_{weight} = 2$ and $h_{weight} = 0.5$ for ridge detection with anisotropic Gaussian smoothing, and $\sigma_{weight} = 0.25$ and $l_{weight} = 2$ for ridge detection with line filter bank smoothing. The performance



(a) Ridge with anisotropic Gaussian filter bank smoothing

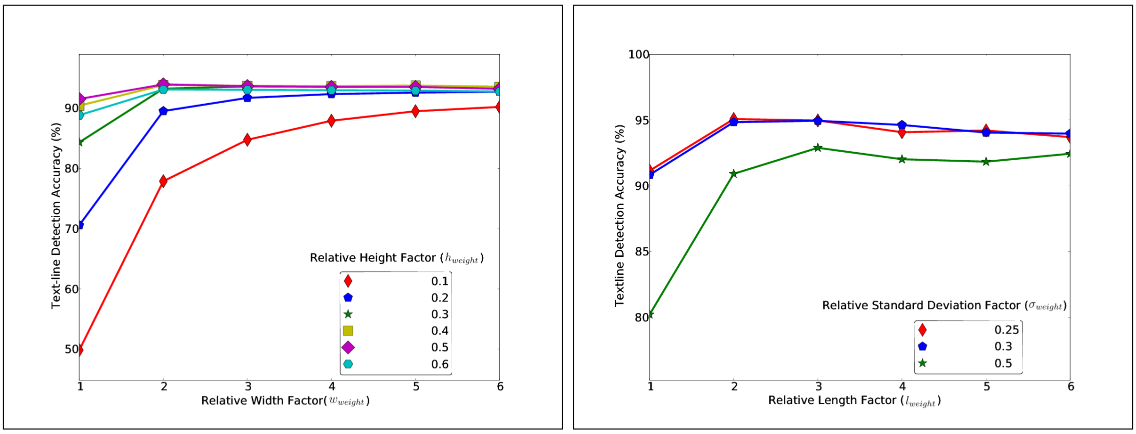
(b) Ridge with line filter bank smoothing

Figure 3.18: the ICDAR 2007 handwritten segmentation contest dataset: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on the ICDAR 2007 handwritten segmentation contest dataset [GAS07].

evaluation results of RAST [Bre02b], smearing [WCW82] and the ridge-based text line finding algorithms are shown in Table 3.1. As shown in the Table 3.1, the ridge-based algorithm achieves better accuracy than smearing [WCW82] and close to RAST [Bre02b].

Arabic/Urdu Script Dataset

Total 25 images of Arabic documents mostly in Naskh script is collected from books, newspapers, and multi-script (English and Arabic) documents. Together with that, 20 images from Urdu documents dataset [SHKB06] (Nastaliq script) are also selected, which belong to the following categories: books, poetries, digests, and magazines. All together, the Arabic and Urdu dataset consists of 45 Arabic and Urdu scripts document images with a variety of multi-column layouts. This dataset can be downloaded from [buk]. For better performance of ridge-based method, column segmentation is done by using the state-of-the-art whitespace cover [Bre02b] before applying text line finding methods. From Figure 3.20, the optimized values of free parameters for this dataset are $w_{weight} = 4$ and $h_{weight} = 0.4$ for ridge detection with anisotropic Gaussian smoothing, and $\sigma_{weight} = 0.3$ and $l_{weight} = 2$ for ridge detection with line filter bank smoothing. The performance evaluation results of the X-Y cut [NSV92], the RAST [SHKB06], and the ridge-based text line finding algorithms are shown in Table 3.1. As shown in the Table 3.1, the



(a) Ridge with anisotropic Gaussian filter bank smoothing

(b) Ridge with line filter bank smoothing

Figure 3.19: the UW-III dataset: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on the UW-III dataset [GHHP97].

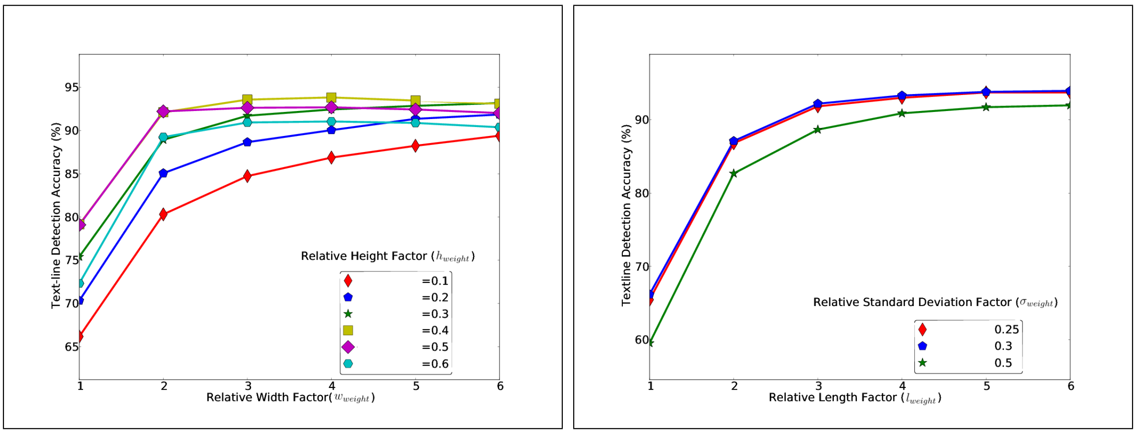
ridge-based method achieved significantly better text line detection accuracy other two state-of-the-art methods.

Fraktur Dataset

This dataset consists of 22 document images with multi-column layouts of Fraktur script, which is one of the famous German calligraphic scripts, and it can be downloaded from [buk]. From Figure 3.21, the optimized values of free parameters for this dataset are $w_{weight} = 2$ and $h_{weight} = 0.3$ for ridge detection with anisotropic Gaussian smoothing, and $\sigma_{weight} = 0.25$ and $l_{weight} = 6$ for ridge detection with line filter bank smoothing. The performance evaluation results of the RAST [Bre02b] and the ridge-based text line finding algorithms, with column segmentation as a preprocessing step using whitespace cover [Bre02b], are shown in Table 3.1. As shown in the Table 3.1, shown in the Table 3.1, the ridge-based method achieved better text line detection accuracy than the state-of-the-art method.

The DFKI-I 2007 Dewarping Contest Dataset

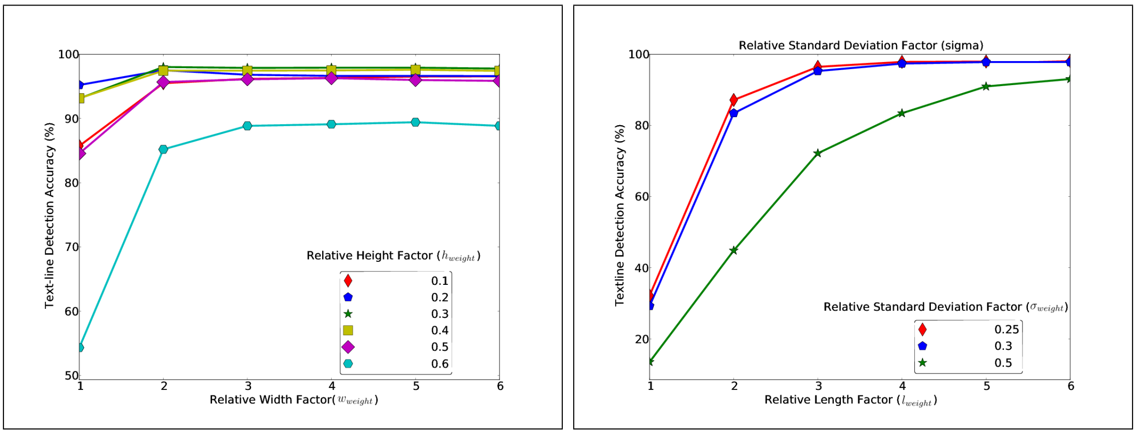
The DFKI-I dataset [SB07] contains 102 grayscale and binarized document images of pages from several technical books captured by an off-the-shelf hand-held digital camera in a normal office environment. Document images in this dataset consist of warped



(a) Ridge with anisotropic Gaussian filter bank smoothing

(b) Ridge with line filter bank smoothing

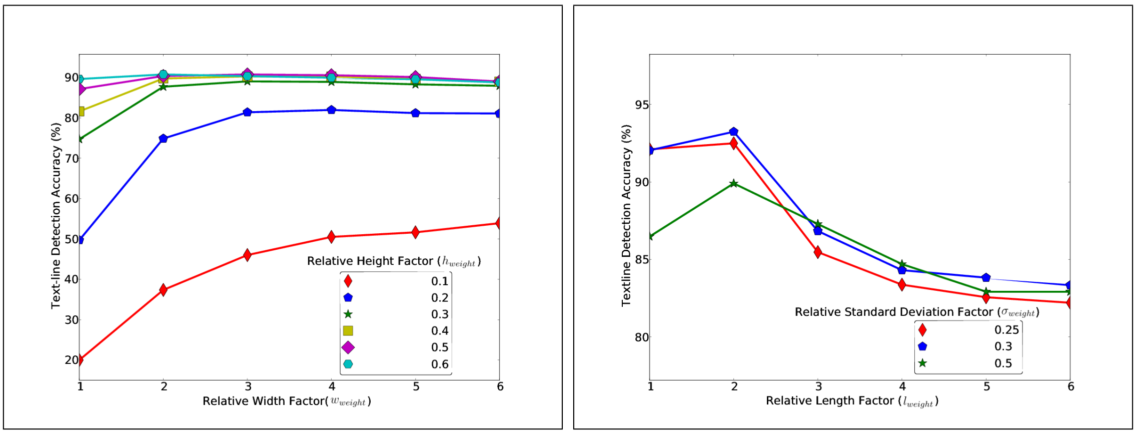
Figure 3.20: Arabic/Urdu Script Dataset: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on Arabic/Urdu script dataset.



(a) Ridge with anisotropic Gaussian filter bank smoothing

(b) Ridge with line filter bank smoothing

Figure 3.21: Fraktur Dataset: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on Fraktur dataset.



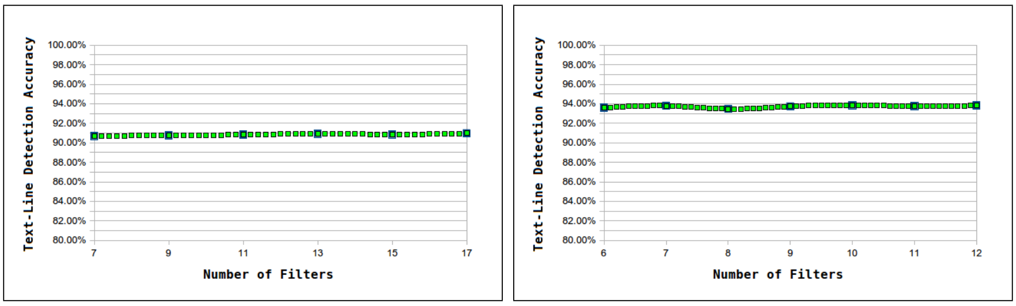
(a) Ridge with anisotropic Gaussian filter bank smoothing

(b) Ridge with line filter bank smoothing

Figure 3.22: the DFKI-I 2007 dewarping contest dataset: plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different values of their free parameters on the DFKI-I 2007 dewarping contest dataset [SB07].

text lines with high degree of curl, different directions of curl within an image, non-text (graphics, halftone, etc.) components and a lot of textual and non-textual border noise. Example images of the dataset are shown in Figures 3.8(a), 3.14(c), and 3.15(e). From Figure 3.22, the optimized values of free parameters for this dataset are: $w_{weight} = 3$ and $h_{weight} = 0.5$ for ridge detection with anisotropic Gaussian smoothing, and $\sigma_{weight} = 0.3$ and $l_{weight} = 2$ for ridge detection with line filter bank smoothing. The performance evaluation results of the ridge-based text line finding algorithm and other two curled text line finding algorithms (rule-based method [OLT^{+}10] and nearest neighbor (NN) based method) are shown in Table 3.1. As shown in the Table 3.1, the ridge-based algorithm achieved better one-to-one text line finding accuracy than nearest-neighbor based [GPN07] and rule-based [OLT^{+}10] algorithms.

An additional experiment is also conducted on the DFKI-I dataset for showing the affect of number of filters within predefined fixed range of values of free parameters on text line detection accuracy. As mentioned above, the optimized values of free parameters for anisotropic Gaussian smoothing are $w_{weight} = 3$ and $h_{weight} = 0.5$, and therefore the corresponding ranges of standard deviations for Gaussian filter bank are: $w_{start} = 3$ to $w_{end} = 5$, and $h_{start} = 0.5$ to $h_{end} = 0.9$. Different number of filters are generated within these selected ranges and the corresponding text line detection accuracy of the ridge-based method is evaluated for each different number of filters. Similarly, for the



(a) ridge detection with anisotropic Gaussian filter bank smoothing

(b) ridge detection with line filter bank smoothing

Figure 3.23: Plot against one-to-one segmentation accuracy of the ridge-based text line finding algorithm with anisotropic Gaussian filter bank smoothing and line filter bank smoothing for different number of filters within the predefined ranges of their values of free parameters on the DFKI-I dataset (102 document images with 3091 text lines). Here, there is no significant change in the text line detection accuracies with respect to the increased/decreased number of filters within the fixed ranges. The main reason is that, the DFKI-I dataset is only compose of few different types of font sizes.

optimized range of $l_{start} = 2$ to $l_{end} = 4$ for line filter bank smoothing, different number of filters are generated and their corresponding affect on text line detection accuracy are evaluated. The results for both of these cases are shown in Figure 3.23. It is important to note that, the text line detection accuracies do change significantly with respect to the increased/decreased number of filters within the fixed ranges. The main reason is that, the DFKI-I dataset does not compose of many different types of font sizes.

The ridge-based text line finding method is also tested on grayscale camera-captured document images from the DFKI-I dewarping contest dataset. For grayscale images of this dataset, the values of free parameters are selected empirically. There is no text line based ground-truth generation and performance evaluation techniques for grayscale document images, which are still the open challenging areas in grayscale document image processing. Therefore, the detected text line regions (ridges) from grayscale camera-captured document images are mapped over their corresponding binary images for text line labeling and performance evaluation purpose. The text line detection accuracy of 91.17% is achieved for ridge-based method with anisotropic Gaussian filter bank smoothing, and 92.75% for ridge-based method with line filter bank smoothing. For comparison, no state-of-the-art text line detection method is found in the literature for grayscale camera-captured document images.

A Collection of Diverse Documents Images

The text line detection accuracy of the ridge-based method (with their default values of free parameters) is already evaluate in the beginning of this section on the combined collection of diverse documents ($n = 23,763$ text lines in 650 documents): 84.89% (Gaussian filter bank smoothing) and 86.10% (line filter bank smoothing). From Table 3.1, the aggregate accuracy of the ridge-based method with dataset-specific optimized values of free parameters can also be calculated: 85.37% (Gaussian filter bank smoothing) and 87.62% (line filter bank smoothing). One important thing to notice here is that, the text line detection accuracy of the ridge-based method with its default values of free parameters are nearly same as the text line detection accuracy with dataset-specific optimized values of free parameters. Additionally, from Table 3.1, the aggregate text line detection result of the best performing dataset-specific state-of-the-art methods of different datasets is also calculated, i.e. 72.99%. From these results, two important conclusions can be made: i) there is no major difference between the text line detection accuracies of ridge-based method with dataset-independent/default and dataset-specific/optimized values of free parameters, and ii) the text line detection accuracy of ridge-based method is significantly better than the aggregate results of the best performing state-of-the-art methods (as shown in Figure 3.24).

3.3.2 Execution Time Analysis

A controlled experiment is also conducted for analyzing the execution times of the main steps of the ridge-based text line finding methods, these are ridge detection with anisotropic Gaussian filter bank smoothing and ridge detection with line filter bank smoothing methods, on the DFKI-I dataset. Here, the reason of selecting only the DFKI-I dataset is that a document image in this dataset contains a biggest size (i.e. around 8 mega pixels) as compared to other datasets. In this experiment, the same number of filters are used with similar range of values of parameters for both smoothing techniques. The computational complexities of both smoothing techniques have been already explained in Section 3.2.1. The anisotropic Gaussian filter bank smoothing step (without any computational optimization) takes around 17 min. per image on the DFKI-I dataset. Whereas, the line filter bank smoothing technique (also without any computational optimization) takes around 3.24 min. per image. However, the ridge detection method is common in both ridge-based text line detection versions, and its running time is also independent on the number of filters. It takes around 5.78 sec. per document image. The machine,

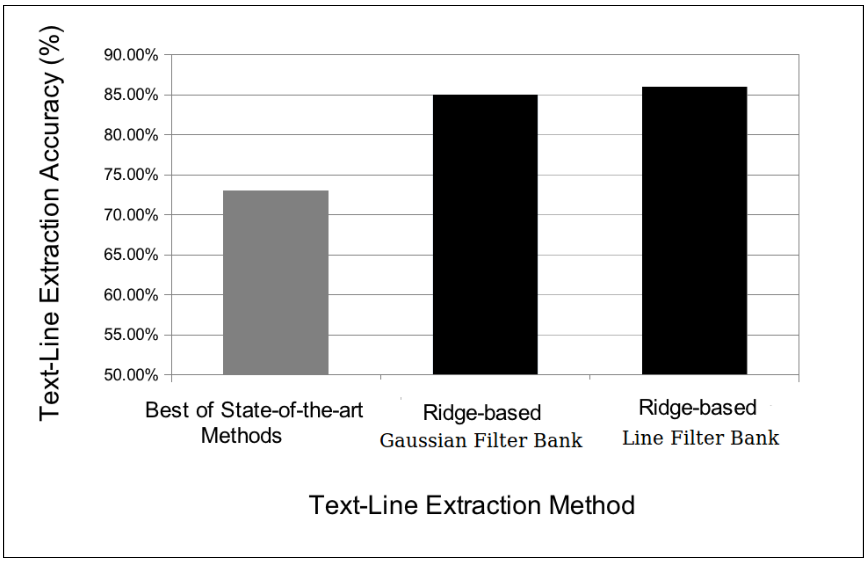


Figure 3.24: Aggregate text line detection accuracy of the state-of-the-art methods (those who performed the best for each datasets), as shown in Table 3.1. Similarly, aggregate text line detection accuracies of the ridge-based method for both, anisotropic Gaussian and line based, filter bank smoothing techniques with their default/dataset-independent values of free parameters. Firstly, it is important to note that, the text line detection accuracies of both versions of the ridge-based method are nearly same (i.e. 86%). Secondly, the text line detection accuracy of the ridge-based method is much better than the aggregate accuracy of best performing domain-specific methods (i.e. 73%). [Note: out of total 23763 text lines, the number of correctly detected text lines for the state-of-the-art methods are 17347 (73%) and for the ridge-based method are 20436 (86%), which are statistically significant than the number of correctly detected text lines for the state-of-the-art methods.]

that was used for the execution time analysis, had following specifications: 2.53 GHz processor and 40 GB RAM with Linux (Ubuntu) operating system. This means, line filter bank smoothing technique is competitively faster than oriented anisotropic Gaussian filter bank smoothing. The execution time of oriented anisotropic Gaussian smoothing can be reduced by using fast implementations of anisotropic Gaussian filter [GSW03, LW06], and the execution time of line averaging filter bank smoothing can also be reduced by using integral images [SKB08a].

3.4 Conclusion

A large number of text line finding algorithms have been introduced in the literature. Each one of them is designed for document images that hold certain assumptions about writing styles, scripts, digitization methods, intensity values and text line structures, and fails when these assumptions are not satisfied. In this chapter, a generic text line finding algorithm is introduced and experimentally tested that can be robustly and equally applied on a large variety of document image categories with respect to writing styles (typed-text or handwritten), digitization methods (scanner or camera), intensity values (binary and grayscale), scripts (Latin, Chinese, Arabic, . . .) and text lines structures (straight, skewed, curled and freestyle handwritten text lines). To the best of author's knowledge, the ridge-based text line extraction method is the first general-purpose text line finding algorithm for document image analysis that can handle the following challenging problems; skewed and/or curled text lines, touching and/or overlapping text lines, free style handwritten text lines, irregular layout, noise and distortions. The ridge-based text line finding algorithm consists of two standard, easy to understand and easy to implement computer vision algorithms: (i) matched filtering and (ii) ridge detection. Unlike most of the state-of-the-art text line finding approaches, the ridge-based method does not require any preprocessing, like zone-segmentation, skew-correction etc., and post-processing steps. The ridge-based text line detection method is so far tested on handwritten, typed-text scanned binary images, and typed-text camera-captured binary and grayscale images having different scripts, and compared its performance with other domain-specific state-of-the art algorithms for different document image classes. The performance evaluation and comparison results are mentioned in Figure 3.1 and Figure 3.24. Either for each separate dataset or a diverse collection of document images in all datasets, the ridge-based text line finding algorithm performs better than the domain-specific best performing state-of-the-art methods as shown in Figure 3.1 and Figure 3.24, respectively. In general, most of the over-segmentation and under-segmentation errors of the ridge-based text line finding algorithm, mainly under the challenging condition of interline touching, can be reduced by investigating a more efficient text line labeling technique that is an open research field. Similarly, most of the false alarm errors in typed-text camera-captured and scanned document images can be reduced by using an effective document cleanup step. The execution time of the ridge-based text line finding algorithm can be reduced by using optimization techniques for filter bank smoothing.

Part II

Text and Non-Text Segmentation

Chapter 4

Text and Non-Text Segmentation using Discriminative Learning[1]

Summary: *segmentation of a document image into text and non-text regions is an important preprocessing step for a variety of document image analysis tasks, like improving OCR, document compression etc. Most of the state-of-the-art document image segmentation approaches perform segmentation using pixel-based or zone(block)-based classification. Pixel-based classification approaches are time consuming, whereas block-based methods heavily depend on the accuracy of block segmentation step. In contrast to the state-of-the-art document image segmentation approaches, the presented segmentation approach introduces a connected component based classification, thereby not requiring a block segmentation beforehand. Here a self-tunable multi-layer perceptron (MLP) classifier is trained for distinguishing between text and non-text connected components using shape and context information as a feature vector. Experimental results prove the effectiveness of the presented method. It has been evaluated on subset of the UW-III, the ICDAR 2009 page segmentation competition test images and circuit diagrams datasets and compared with the state-of-the-art Leptonica's* [2] *page segmentation method.*

[1] This work was published in Bukhari et al. [BSB10a] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. Document Image Segmentation using Discriminative Learning over Connected Components. In Proceedings 9th IAPR Workshop on Document Analysis Systems, pages 183-190, Boston, Massachusetts, USA, 2010. Copyright ©2012 ACM, Inc"*. This chapter is an adapted version of the published work.

[2] http://code.google.com/p/Leptonica/

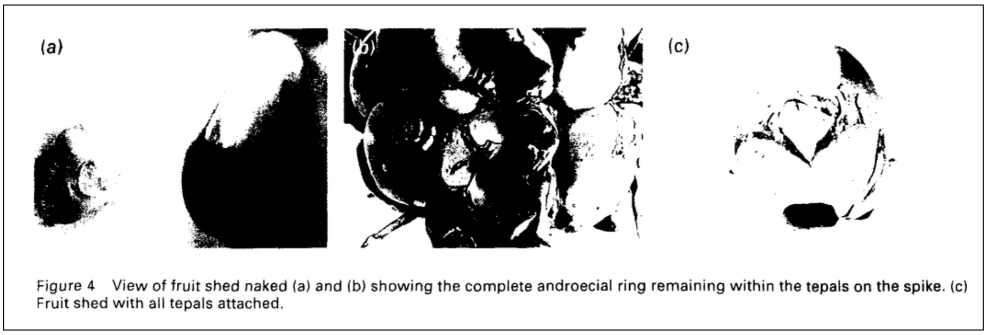
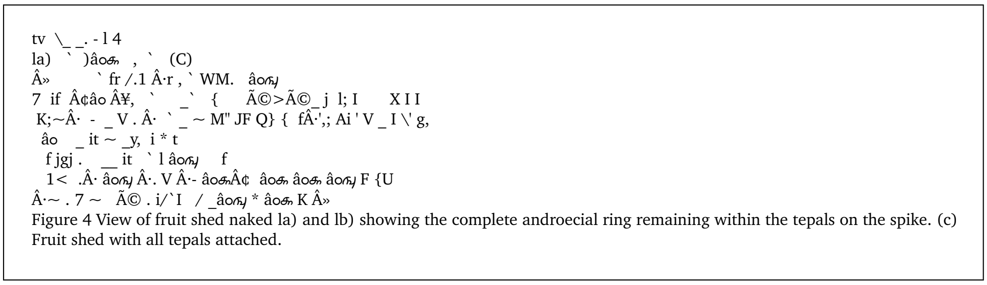


Figure 4 View of fruit shed naked (a) and (b) showing the complete androecial ring remaining within the tepals on the spike. (c) Fruit shed with all tepals attached.

(a) Input page segment



```
tv \_ _. - l 4
la)  `  )âoண  ,  `   (C)
Â»        ` fr /.1 Â·r , ` WM.   âoஜ
7  if  Â¢âo Â¥,   `     _`   {     Ã©>Ã©_j  l; I      X I I
 K;~Â·  -  _V . Â·  `  _ ~ M" JF Q} {  fÂ·',; Ai ' V _ I \' g,
  âo    _ it ~ _y,  i * t
   f jgj .    __ it   ` l âoஜ     f
    1<  .Â· âoஜ Â·. V Â·- âoணÂ¢  âoண âoண âoஜ F {U
Â·~ . 7 ~   Ã© . i/`I   / _âoஜ * âoண K Â»
Figure 4 View of fruit shed naked la) and lb) showing the complete androecial ring remaining within the tepals on the spike. (c)
Fruit shed with all tepals attached.
```

(b) OCR result

Figure 4.1: The OCR result of an in-correctly segmented zone containing both images and text. The OCR system generates many garbage symbols from the non-text parts of the input page segment.

4.1 Introduction

Document image segmentation is the problem of classifying the contents of a document image into a set of text and non-text classes. Non-text class consists of following categories: halftone, drawing, maths, logos, tables, etc. Document image segmentation is one of the most important preprocessing steps before feeding the specific contents to an optical character recognition (OCR) system otherwise OCR engine produces lot of garbage characters originated from non-text components, as shown in Figure 4.1.

Document image segmentation approaches in the literature can generally be classified into two groups: (i) block or zone based classification and (ii) pixels based classification. Block based segmentation approaches apply page segmentation [SvBKB08] on the document image and then classify the obtained blocks into a set of determined classes [KSB07]. On the other hand pixel based approaches attempt to classify individual pixels [MBA08, MB08] according to predefined classes.

Several block classification algorithms have been proposed over the years. For a more detailed overview of related work in the field of document block classification please refer to Okun [ODP99] and Wang [WPH06]. Okun et al. [ODP99] proposed an approach for document block classification based on connected components and run-length statistics. Wang et al. [WPH06] presented the block classification system, each block with a 25 dimensional feature vector and use an optimized decision tree classifier to classify each block into one of different target classes. The most recent and detailed block classification approach is introduced by Keysers et. al [KSB07] which showed that a document block classification system can be constructed using run-length histogram feature vector alone. That work includes several classes of blocks (math, logo, text, table, drawing, halftone, ruling and speckles). In general, the approaches that classify blocks depend heavily on the result of page segmentation into blocks. The blocks may be segmented in a wrong way leading to miss-classification.

Moll et al. [MBA08, MB08] classify individual pixels instead of regions, to avoid the constriction of the limited classes of region shapes. The approach is applied on handwritten, machine printed and photographed document images. Pixel based classification approaches are slow with respect to execution time. The approach by Won [Won08] focuses on a combination of a block based algorithm and a pixel based algorithm to segment a document image into text and image area.

Together with block based and pixel based image segmentation approaches, there is another state-of-the-art text and halftone segmentation approach reported by Bloomberg et al. [Blo91] based on multi-resolution morphological operations. This approach comprises three steps: 1) at first step, seed image is generated by sub-sampling input image such that the resulting seed image mainly contains halftone pixels. 2) Then mask image is produced by using morphological operations such that together with all image pixels there is a sufficient connectivity of halftone seed pixels with other pixels covering halftone regions. 3) In last step binary filling operation is used to transform seed image with the help of mask image into final halftone mask image. The open-source version of this algorithm is presented in Leptonica library developed by Dan Bloomberg. This method produces promising results for halftone objects but is unable to recognize thin halftone and drawing like objects as non-text objects.

In this chapter, the main aim is to perform text and non-text classification based on connected components, instead of pixels or blocks. For this purpose, a simple and easy to compute feature vector is used. For training, a multi-layer perception (MLP) classifier is applied which has already been used in different document image pre-processing tasks [MGS05], like binarization [CW01], deskewing [RB95]. Classifier tuning is considered as one of the hard problem with respect to the optimization of parameters. In order to get rid of this problem a self-tunable MLP classifier [BS10] is used. the presented method is independent of block segmentation and equally applicable to different categories of non-text objects if they were included in the training data. One can analyze the ease of implementation and accuracy of the presented method in the algorithm description and the experimental evaluation sections, respectively.

The rest of this chapter is organized as follows. In Section 4.2 the presented text and non-text document image segmentation method is described. Section 4.3 deals with the experimental results. Section 4.4 describes conclusion.

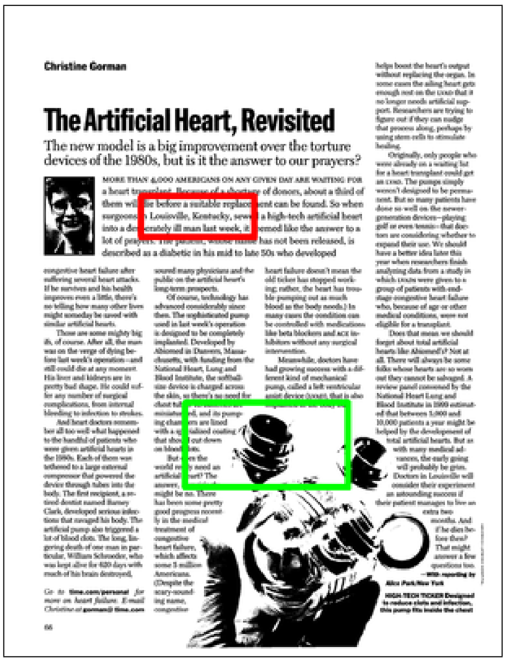
4.2 The Text and Non-Text Segmentation Method

Here the document image segmentation algorithm is described which segments document image into text and non-text regions. The main target is to classify each connected component as either text or non-text component. In Section 4.2.1 the feature extraction process is described. In Section 4.2.2 the training of extracted features using a self-tunable multi-layer perceptron (AutoMLP) classifier is described.

4.2.1 Feature Extraction

Instead of extracting complex features from a connected component, the raw shape of a connected component itself is an important distinguishable feature for classifying structured text and random/irregular non-text components, as shown in Figure 4.2. Together with the shape of connected component, the surrounding area (context) of a connected component can also play an important role for text and non-text classification, similarly because of the structured text and non-structured non-text surrounding areas, respectively. Figure 4.2 shows neighborhood surrounding areas (context) for text and non-text regions. Based on the above mentioned hypothesis, the feature vector of connected component is composed of shape and context information. Detail description of the feature vector is presented below.

- **shape of connected component**: In document images, most of the text components are smaller than non-text components. Therefore size information can play an important role in the text and non-text components classification. But only size information is not enough for classifying the big text and the small non-text components. Therefore, together with size information some other discriminative features are required. As already mentioned, the shapes of non-text connected components



Christine Gorman

The Artificial Heart, Revisited

The new model is a big improvement over the torture devices of the 1980s, but is it the answer to our prayers?

Figure 4.2: Sample image from the ICDAR 2009 page segmentation competition. This image shows the structured shapes of text components and random shapes of non-text components.

are irregular, random and vary a lot form one image to another and on other hand the shapes of text components are uniformly structured in document images. The structured and random shapes of text and non-text components respectively can be learned by the MLP classifier. For generating feature vector, each connected component is rescaled to a 40 × 40 pixel window size. This rescaling performs only downscaling, such that a connected component is downscaled if either length or height of component is greater than 40 pixels otherwise it is fit into the center of a 40 × 40 window. The advantage of doing this type of rescaling is to distinguish the shape of small components from large components. This type of rescaling can produce different feature vectors for a same components, for example small and big font ‘a’. Here, the target is not to classify each characters but to classify the text and non-text components. Therefore, this type of rescaling works better than normal rescaling for text and non-text classification because of incorporating implicit size information of text and non-text components. Samples of rescaled text and non-text connected components are shown in Figure 4.3(a) and Figure 4.3(b), respectively. Together with raw rescaled connected component, the shape based feature vector is also composed of four other size based features, mentioned below.

So all together the size of the shape-based feature vector is 1604.

1. normalized length (length of a component divided by the length of an input image).
2. normalized height (height of a component divided by the height of an input image).
3. aspect ratio of a component (length divided by height).
4. number of foreground pixels in a rescaled area divided by total rescaled area.

- **surrounding context of connected component**: Usually the text components are aligned horizontally in the document images which results in structured surrounding area for a text component as compared to the non-structured surrounding area for non-text components. Therefore, the surrounding context of a connected component can play an important role in classifying the text and the non-text components. Each connected component with its surrounding context area is rescaled to a 40×40 window size for generating context-based feature vector. Here the surrounding context area is not fixed for all of the connected components for calculating feature vectors, but it is a function of component's length(l) and height(h). Such that, for each connected component the area of dimensions $5 \times l$ by $2 \times h$ is chosen empirically by keeping a connected component at center for rescaling. The rescaled text and non-text context components are shown in Figure 4.3(c) and Figure 4.3(d), respectively. The size of the context-based feature vector is 1600.

In this way, the size of a complete feature vector is 3204 which consist of raw rescaled shape (dimension 1600), raw rescaled context (dimension 1600) and four size based features.

4.2.2 Classification

In general, classifier tuning is a hard problem with respect to the optimization of their sensitive parameters, for example learning rate of MLP classifier, 'C' and gamma of SVM classifier, confidence of decision tree classifier, maximum depth and number of attributes of random forest classifier, 'k' of K nearest neighbor classifier etc. Here, MLP classifier is used for text and non-text classification. Performance of MLP classifier is sensitive to the chosen parameters values. The optimal parameters values depend upon the dataset. The parameters optimization problem can be solved by using grid search for classifier

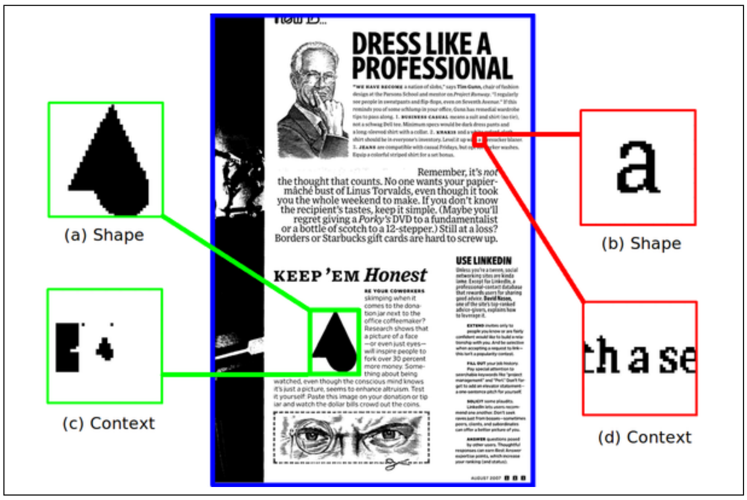


Figure 4.3: Text and non-text connected components shape and context features, (a) and (b) show rescaled (no upscaling, either downscale or fit into the center to preserve size) connected component's shape features. (c) and (d) show rescaled connected component's context features.

training, but grid search is a slow process. Therefore in order to overcome this problem AutoMLP [BS10] classifier (a self-tuning classifier) is used.

AutoMLP combines ideas from genetic algorithms and stochastic optimization. It trains a small number of networks in parallel with different learning rates and different numbers of hidden layers. After a small number of training cycles the error rate of each network is determined with respect to a validation dataset according to an internal validation process. Based on validation errors, the networks with bad performance are replaced by the modified copies of networks with good performance. The modified copies are generated with different learning rates and different numbers of hidden layers using probability distributions derived from successful rates and sizes. The whole process is repeated a few number of times, and finally the best network is selected as an optimally trained MLP classifier.

Feature vectors for training AutoMLP classifier have been extracted from the UW-III dataset. The UW-III dataset contains zone-level ground truth for text, halftone, ruling, drawing and logo. From this zone-level ground-truth information, the text and the non-text (halftone, drawing and logo) regions are extracted form document images. Non-text regions were small in number, which have been increased up to four times by rotating each non-text region in four different orientations. Around 0.7 million text samples and 0.1 million non-text samples are used for training AutoMLP classifier.

For testing and evaluation purpose, the feature vector for each connected component

Table 4.1: Performance evaluation of the presented discriminative learning based method and the Leptonica's page segmentation method on the UW-III dataset (95 document images), the ICDAR 2009 page segmentation competition test dataset (8 document images) and the combined UW-III and ICDAR 2009 datasets (103 document images). The document images in these datasets contains only text and halftone elements. Both of the methods achieved nearly same segmentation accuracy of these dataset.

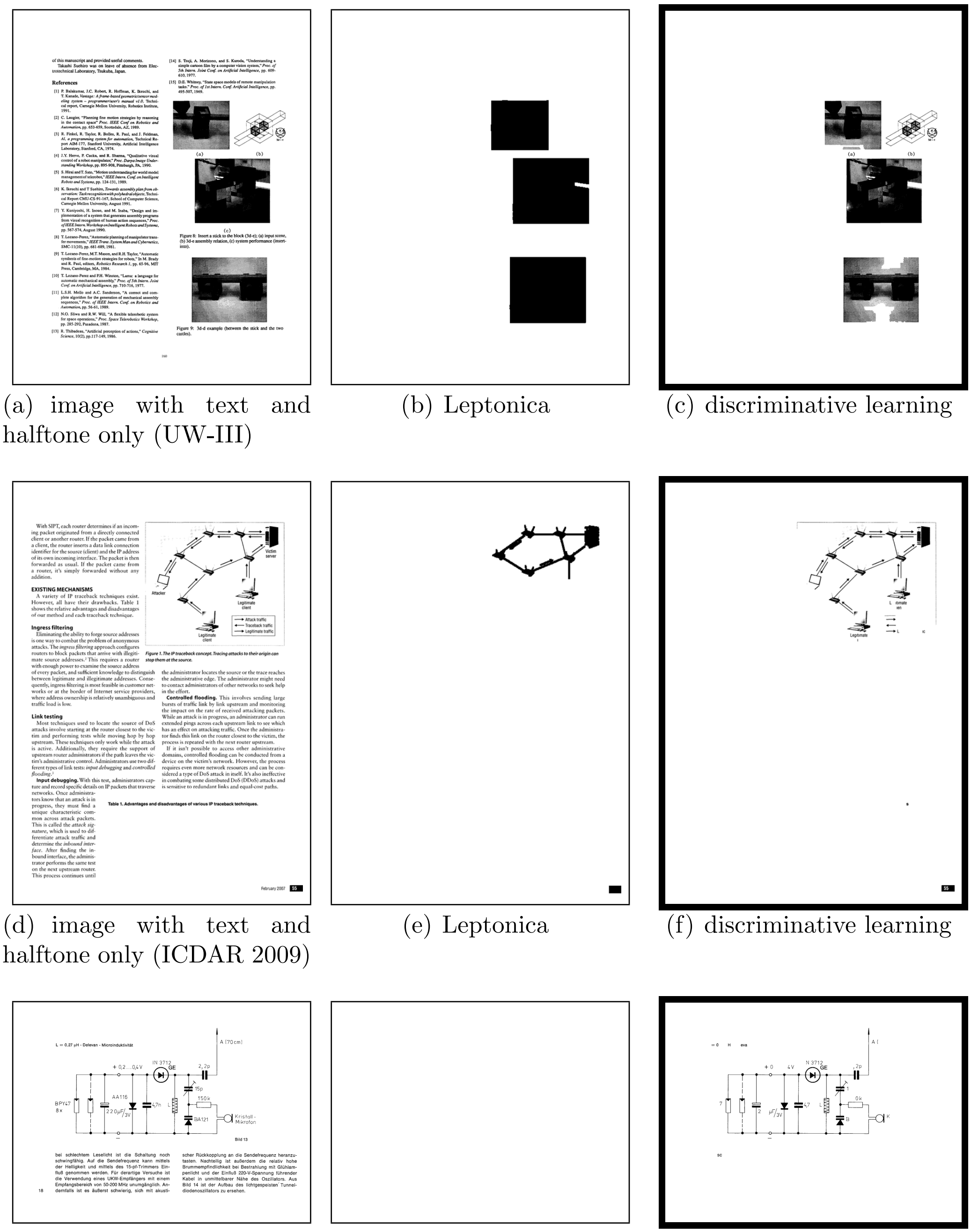
	UW-III		ICDAR-2009		*Combined*	
	Leptonica	discriminative learning	Leptonica	discriminative learning	*Leptonica*	*discriminative learning*
non-text classified as non-text	95.36%	98.91%	84.91%	96.70%	*94.77%*	*98.79%*
non-text classified as text	4.64%	1.09%	15.09%	3.30%	*5.23%*	*1.21%*
text classified as text	99.79%	95.93%	99.87%	93.31%	*99.79%*	*95.72%*
text classified as non-text	0.21%	4.07%	0.13%	6.69%	*0.21%*	*4.28%*
segmentation accuracy	97.57%	97.42%	92.39%	95.01%	*97.28%*	*97.25%*

of a test document image is extracted in the same way as described in Section 4.2.1. Then a class label is assigned to each connected component based on classification probabilities of text and non-text.

In order to improve the segmentation results, a nearest neighbor analysis by using class probabilities is also performed for refining the class label of each connected component. For this purpose, a region of 70×70 (empirically chosen) is selected from document image by keeping targeted connected component at center. The probabilities of connected components within the selected regions are already computed during classification. Already assigned class labels of the connected components are updated using the average text and non-text probabilities of connected components within selected region. Some of segmented results are shown in Figure 4.4.

4.3 Experimental Results

The presented text and non-text segmentation method is evaluated on the publicly available UW-III [GHHP97] and the ICDAR-2009 page segmentation competition test dataset [APBP09] and the private circuit diagrams dataset. The main reason for using different datasets is to check the accuracy of the presented approach on different types of images which have not been used in training as well as to have a variety of text and non-text components. For example, majority of the document images in the UW-III dataset contain Manhattan-layout but the ICDAR 2009 dataset also contains documents with



(a) image with text and halftone only (UW-III)

(b) Leptonica

(c) discriminative learning

(d) image with text and halftone only (ICDAR 2009)

(e) Leptonica

(f) discriminative learning

(g) image with text and halftone only (Circuit Diagram)

(h) Leptonica

(i) discriminative learning

Figure 4.4: Document image segmentation results of the presented discriminative learning based method and Leptonica's method in non-text mask format.

non-Manhattan layout. All non-text components, except halftone, have been removed from the UW-III and the ICDAR-2009 test datasets. In contrast to this, the circuit diagrams dataset mainly composed of text and drawing components having no other types of non-text components. The UW-III dataset consists of zone level ground-truth information. Total 95 documents have been selected from the UW-III dataset. The publicly available ICDAR 2009 dataset contains 8 test images with zone level ground truth information. The circuit diagrams dataset composed of 10 selected images from a circuit diagram book; zone level ground-truth information is also generated for these document images.

For each dataset, pixel-level ground truth has been generated using zone-level ground truth information. Each pixel in ground-truth images contains either text or non-text label. Different types of metrics have been used for the performance evaluation of document image segmentation method which are defined below:

1. **non-text classified as non-text:** percentage of intersection of non-text pixels in both segmented and ground truth image with respect to the total number of non-text pixels in ground truth image.

2. **non-text classified as text:** percentage of intersection of text pixels in segmented image and non-text pixels in ground truth image with respect to the total number of non-text pixels in ground truth image.

3. **text classified as text:** percentage of intersection of text pixels in both segmented and ground truth image with respect to the total number of text pixels in ground truth image.

4. **text classified as non-text:** percentage of intersection of non-text pixels in segmented image and text pixels in ground truth image with respect to the total number of text pixels in ground truth image.

5. **segmentation accuracy:** average percentage of text classified as text accuracy and non-text classified as non-text accuracy.

Based on the matrices defined above, the presented method is compared with Leptonica's page-segmentation method. Leptonica algorithm is exclusively designed for segmenting text and halftone components. Performance comparison results of the presented method and the Leptonica methods on the UW-III and the ICDAR 2009 test datasets (which contain only text and halftone components) are shown in Table 4.1. The presented

Table 4.2: Performance evaluation results of the presented discriminative learning based method and Leptonica's page segmentation algorithms on circuit diagrams dataset (10 document images). Note: Leptonica method is designed for text and halftone segmentation. Here it has been used for evaluation on circuit diagrams dataset to show that usually text and halftone based segmentation methods can not be directly applied on other types of non-text components segmentation. The discriminative learning based method achieved significantly better segmentation accuracy as compared to the Leptonica's method on circuit diagram dataset.

	Leptonica	discriminative learning
non-text classified as non-text	0%	89.79%
non-text classified as text	100%	10.21%
text classified as text	100%	89.29%
text classified as non-text	0%	10.72%
segmentation accuracy	50%	89.54%

method has also been evaluated on circuit diagrams dataset in order to show its potential as compared to Leptonica; results are shown in Table 4.2.

4.4 Conclusion

A new method for document image segmentation into text and non-text regions based on discriminative learning over connected components is described and experimentally evaluated here. The self-tuning MLP classifier (AutoMLP) [BS10] is applied for classification which automatically optimized learning parameters. The presented method is independent of preprocessing step of zone segmentation which is usually the case in zone based classification approaches. The presented method is evaluated on the UW-III, the ICDAR 2009 page segmentation competition test dataset and circuit diagrams and also compared with the state-of-the-art Leptonica's page segmentation method [Blo91]. In general, both the text and non-text components are equally important in document image analysis operations. For example, OCR exclusively requires text components and the document image compression or symbol recognition approaches exclusively require non-text components. The performance evaluation results of the presented method and Leptonica method are shown in Table 4.1 and Table 4.2. It is obvious from the results that Leptonica method has better text classification accuracy than non-text classification. Leptonica method missclassifies the small non-text components as the text components, as shown in Figure 4.4(b) and Figure 4.4(e). On the other hand, the presented method gives equal importance to both the text and non-text components during the classification. Unlike Leptonica method, the presented method can also classify between the small non-text and text components, as shown in Figure 4.4(c) and Figure 4.4(f). Leptonica method is designed for the text and halftone segmentation and is not specifically designed for the drawing objects segmentation. Therefore, it is unable to recognize drawing images in circuit diagram dataset, as shown in Table 4.2 and Figure 4.4(h). Together with halftone components segmentation, the presented method also has a potential of segmenting drawing components (for example circuit diagrams), as shown in Table 4.2 and Figure 4.4(i). It achieved the aggregated segmentation accuracy of 96% (n = 113 documents), which is better than the 93% segmentation accuracy of the state-of-the-art multiresolution morphology (Leptonica) based page segmentation method. The segmentation results of the presented method can be improved by increasing training samples and/or by using some post-processing operations.

Chapter 5

Text and Non-Text Segmentation using Multiresolution Morphology[1]

Summary: *Bloomberg's page segmentation algorithm [Blo91] is a text and halftone image separation approach that is unable to segment other type of non-text elements such as line drawings, graphs, and maps. This chapter describes improvements to the Bloomberg's text/halftone segmentation algorithm to make it a general text and non-text image segmentation approach, where non-text components can be halftones, line drawing, maps, and graphs. The modifications result in significant improvements over Bloomberg's algorithm on the UW-III, the ICDAR 2009 page segmentation competition test images and the circuit diagrams datasets.*

5.1 Introduction

Bloomberg [Blo91] described an approach to page segmentation based on multiresolution morphology. Bloomberg's approach is simple and performs well for separating halftone images from text. Furthermore, an open source implementation is available as part of the Leptonica library [Blo]. Bloomberg's text/image segmentation approach was specifically designed for separating text and halftone components. It is often unable to differentiate between text and non-text components other than halftones, like drawings, graphs, maps etc. In this chapter, improvements in Bloomberg's text/image segmentation algorithm

[1] This work was published in Bukhari et al. [BSB11c] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In Proceedings SPIE Document Recognition and Retrieval XVIII, San Jose, CA, USA, Jan. 2011. Copyright ©2012 SPIE"*. This chapter is an adapted version of the published work.

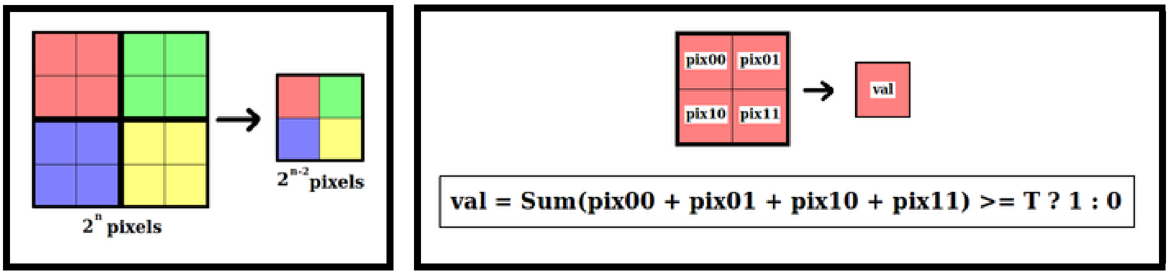
are introduced to generalize it for separating text and non-text components including halftones, drawings, graphs, maps, etc.

The rest of the chapter is organized as follows. In Section 5.2, Bloomberg's text/image segmentation algorithm is described in detail. In Section 5.3, the improvements to Bloomberg's algorithm are explained. Section 5.4 deals with the experimental results and Section 5.5 discusses conclusions.

5.2 Bloomberg's Text and Halftone Image Segmentation

Multiresolution morphology is the main technique used in Bloomberg's text/image segmentation algorithm. Bloomberg [Blo91] first defined the outline of the text/image segmentation algorithm using basic morphological operations before introducing his multiresolution morphology based algorithm, such that: i) an image can be morphologically closed with a sufficiently large structuring element intending to solidify halftone components, ii) then the image can be morphologically opened with an even larger structuring element intending to remove the text blobs and to preserve some portions of halftone components, iii) the residual portions or seeds of the halftone image can be used for generating the halftone mask from the original image. Bloomberg [Blo91] has highlighted the importance of the multi-scale image representation by emphasizing that it can be used for efficient analysis of image contents as well as speeding up image processing operations (like morphology). He updated the aforementioned basic outline of the text and halftone segmentation algorithm using multi-scale image representation such that: i) an image can be closed or dilated before subsampling, in order to coalesce halftone components, ii) the image can be opened or eroded before further subsampling to intend to preserve only halftone portions. As it is expensive to use large structuring element at full or high image resolution, he introduced the key concept of "threshold reduction" for implementing the subsampling based text/image segmentation algorithm. The threshold reduction is defined as follows.

Threshold Reduction: consider a binary image where each foreground pixel is represented by '1' and each background pixel is represented by '0'. The image is tiled into 2 × 2 pixel blocks. Each 2 × 2 block of four pixels is replaced by a single pixel in subsampled image. The value of each subsampled pixel is either '1' or '0' depending on the chosen threshold, that ranges between one and four. The subsampled pixel value is '1'



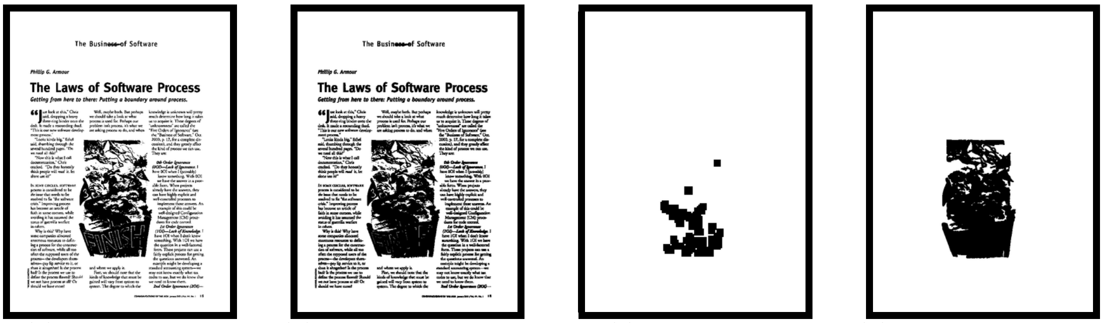
(a) 4 × 1 Threshold Reduction (b) Formula of 4 × 1 Threshold Reduction

Figure 5.1: Definition of multiresolution morphology based threshold reduction operation: (a) each 2 × 2 block of four pixels is subsampled to one pixel (4 × 1 Reduction). (b) the value of subsampled or reduced pixel is '1' if the sum of the values of four pixel within 2 × 2 block is greater than or equal to the threshold (T), otherwise '0'. The threshold can be set between one and four.

if the sum of the values of four pixels is greater than or equal to the threshold, otherwise '0'. The subsampling operation of each 2 × 2 block into single pixel with the threshold equal to one mimics the dilation of image with 2 × 2 structuring element followed by subsampling of upper-left pixel of each 2 × 2 pixel block. Similarly, subsampling with the threshold equal to four mimics the erosion of image with 2 × 2 structuring element followed by subsampling of upper-left pixel of each 2 × 2 pixel block. Besides thresholds of one for dilation and four for erosion, the threshold can be set equal to two or three as well. This type of threshold selection is referred as threshold convolution or rank order filter. Bloomberg [Blo91] referred the combination of threshold convolution followed by subsampling as "threshold reduction". After a single threshold reduction (also called 4×1 reduction) operation, the number of image pixels is reduced from 2^n to 2^{n-2}. The concept of threshold reduction is illustrated in Figure 5.1. Bloomberg's text/image segmentation algorithm is described below.

5.2.1 The Original Bloomberg's Algorithm

As mentioned before, Bloomberg's algorithm is based on the aforementioned threshold reduction (multiresolution morphology) concept and basic morphological operations. It also uses the trivial 1 × 4 expansion operation in which each pixel value is copied into 2×2 pixel block of four pixels. Bloomberg's halftone mask image generation algorithm is described as follows. Consider a binary image in which the foreground pixel value is '1' and the background pixel value is '0'. At first an input image is processed by two threshold reduction operations with thresholds equal to one. This operation subsamples the input



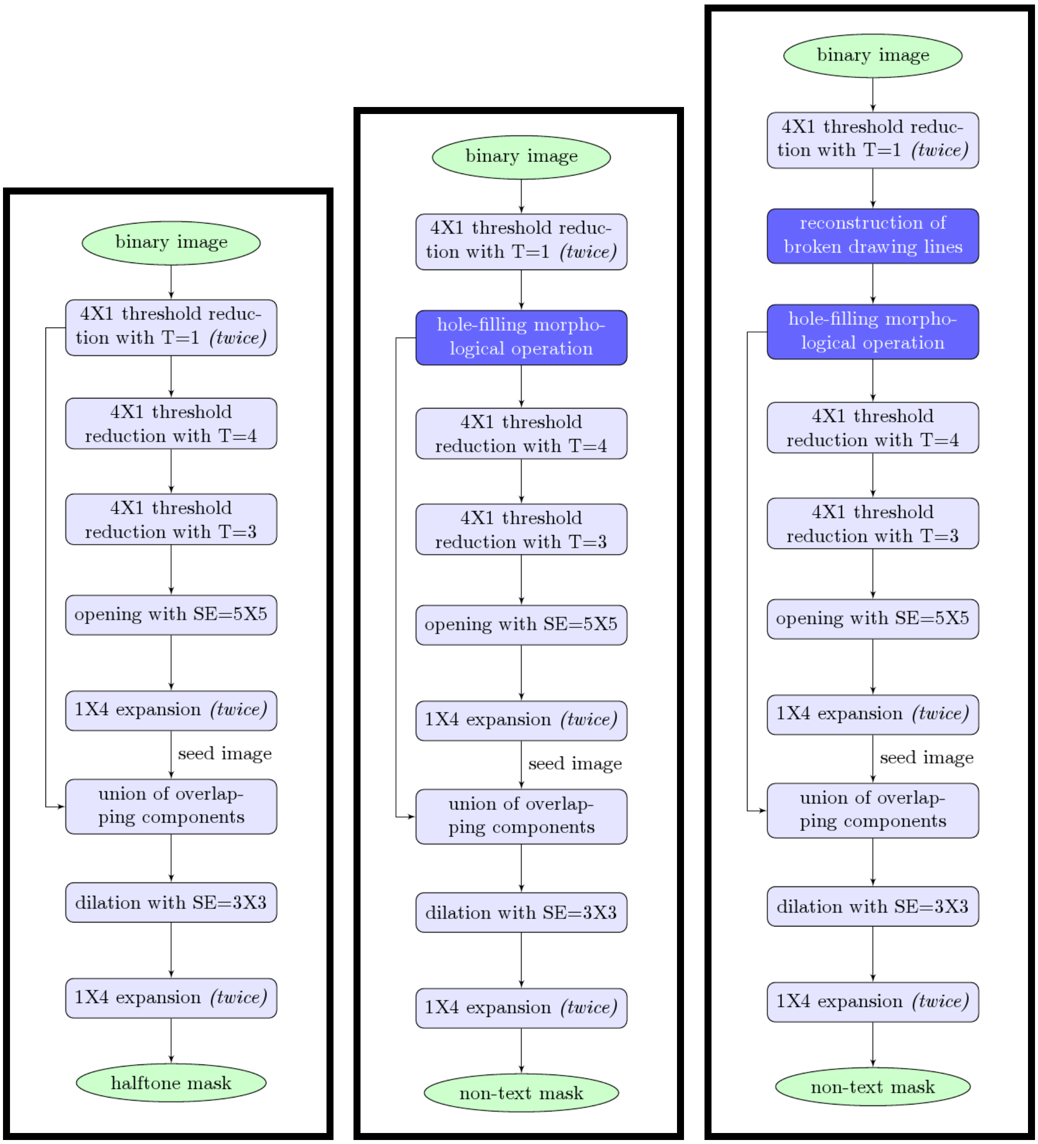
(a) input image (b) 16 × 1 subsampled image (c) seed image (d) halftone mask image

Figure 5.2: Snapshots of Bloomberg's text and halftone image segmentation algorithm.

image from 2^n to 2^{n-4} pixels by preserving the density of low as well as high frequency components within document image. This image can then be referred to as a 16 × 1 subsampled image, as shown in Figure 5.2(b). The subsampled image is further reduced by two threshold reduction operations with thresholds equal to four and three respectively and then followed by morphological opening by using a 5 × 5 structuring element. These further threshold reductions of the 16 × 1 subsampled image and morphological opening are intended to remove the text components and preserve some portions of halftone components, as shown in Figure 5.2(c). The image in Figure 5.2(c) is referred to as the seed image. The seed image is expended by using two 1 × 4 expansions to become equal in size to the 16 × 1 subsampled image of Figure 5.2(b). Finally, the halftone mask image is generated by comparing the 16 × 1 subsampled image (Figure 5.2(b)) with the seed image (Figure 5.2(c)) and selecting only fully or partially overlapping components between them. After morphological dilation (structuring element 3 × 3), the halftone mask image is expended by two 1 × 4 expansions to become equal to the dimension of the input image. The halftone mask image is shown in Figure 5.2(d). The data flow diagram of Bloomberg's text/image segmentation algorithm is shown in Figure 5.3(a).

5.3 The Modifications to Bloomberg's Algorithm

Bloomberg's text/image segmentation algorithm is specifically designed for separating text and halftone image from a document image. It is unable to discriminate between text and drawing type non-text components and therefore fails to separate both of them from each other. Here, it is first described why Bloomberg's algorithm is unable to distinguish between text and non-text components except text and halftone components.

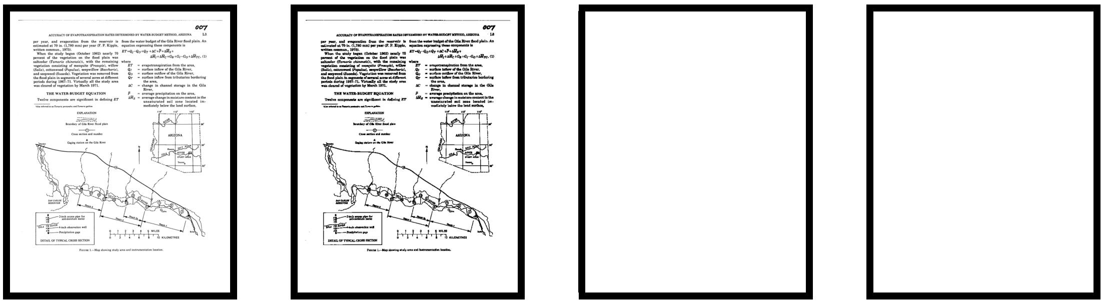

binary image
4X1 threshold reduction with T=1 *(twice)*
4X1 threshold reduction with T=4
4X1 threshold reduction with T=3
opening with SE=5X5
1X4 expansion *(twice)*
seed image
union of overlapping components
dilation with SE=3X3
1X4 expansion *(twice)*
halftone mask
binary image
4X1 threshold reduction with T=1 *(twice)*
hole-filling morphological operation
4X1 threshold reduction with T=4
4X1 threshold reduction with T=3
opening with SE=5X5
1X4 expansion *(twice)*
seed image
union of overlapping components
dilation with SE=3X3
1X4 expansion *(twice)*
non-text mask
binary image
4X1 threshold reduction with T=1 *(twice)*
reconstruction of broken drawing lines
hole-filling morphological operation
4X1 threshold reduction with T=4
4X1 threshold reduction with T=3
opening with SE=5X5
1X4 expansion *(twice)*
seed image
union of overlapping components
dilation with SE=3X3
1X4 expansion *(twice)*
non-text mask

(a) The original Bloomberg's algorithm

(b) First modified version

(c) Second modified version

Figure 5.3: Data flow diagrams of Bloomberg's text/image segmentation algorithm and the modified versions ('T': threshold; 'SE': structuring element).



(a) input image

(b) subsampled (16× 1) image

(c) empty seed image (no seed for drawing type elements)

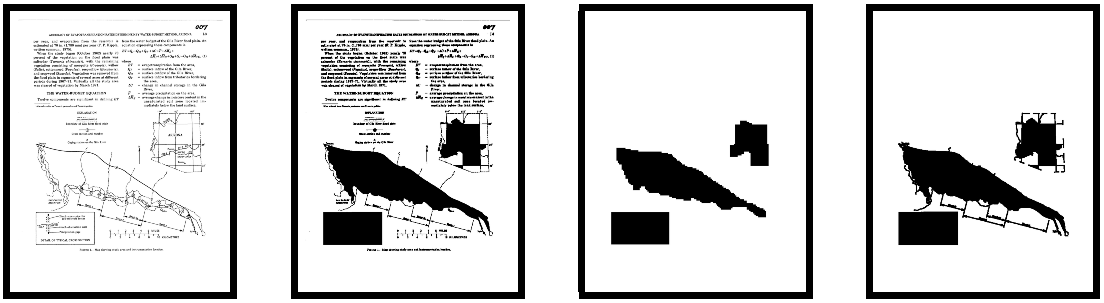
(d) empty seed image fails to produce correct halftone mask image

Figure 5.4: Bloomberg's text/image segmentation algorithm often fails to separate drawing type non-text components from text components.

Then, the proposed modifications to Bloomberg's text/image segmentation algorithm are described mainly to improve it for efficiently separating text and non-text components, including halftones, drawings, maps, graphs, etc.

Bloomberg's algorithm is intended to preserve some portion(s) of the halftone image components in the seed image, which is later used for generating the halftone image mask. The seed image is generated by using four consecutive threshold reduction operations with thresholds equal to one, one, four and three, respectively. The threshold reduction with the threshold equals to one preserves low as well as high frequency details of an image while a threshold greater than one drops fine or minor image details. On one hand, if a non-text component in an original image contains some solid bunch of pixels, then it will be preserved in the seed image, even after the threshold reductions with thresholds greater than one. On another hand, if a non-text component only composed of thin drawing lines, then it will vanish after the high value threshold reductions in the seed image. Therefore, Bloomberg's algorithm often fails to separate text and non-text components where non-text components do not contain any solid bunch of pixels, as shown in Figure 5.4.

First Modification: Hole-filling Morphological Operation: it has been observed that non-text components, such as drawings, maps, graphs and even halftones, are often composed of hollow contours of geometric and irregular shapes, as shown in Figure 5.4(a). The threshold reduction operations with high thresholds remove these hollow contours. But if these hollow contours can be filled before the high value threshold reduction operations, then they will remain present in the seed image. Another consequence of



(a) input image

(b) subsampled (16 × 1) image followed by hole-filling operation)

(c) seed image

(d) non-text mask image

Figure 5.5: First modified Bloomberg's text/image segmentation algorithm: hole-filling based improved Bloomberg's algorithm produces accurate non-text mask for drawing type components as compared to the result of the original Bloomberg's algorithm as shown in Figure 5.4.

using the image filling operation is that it only fills hollow shape image components and preserve other text and non-text components as before. For this purpose, the well known "hole-filling" morphological operation is used, which is briefly described here as, (i) an input image with foreground pixels '1' and background pixels '0' is used as mask image, ii) the filled-image is initialized with all '0' pixels except the top-left pixel with '1', iii) the filled-image is dilated using a 3 × 3 structuring element, iv) after dilation, all of the pixels that are '0' in the mask image are set to '0' in the filled-image, v) dilation followed by resetting of the filled-image's pixels is repeated until no more changes are made to the filled-image. The hole-filling based modified Bloomberg's algorithm is briefly illustrated in Figure 5.3(b). The text/image segmentation results of the original Bloomberg's' algorithm and modified version are shown in Figures 5.4 and 5.5 respectively for comparison. Unlike the original Bloomberg's algorithm, the improved version can accurately separate text and non-text images including halftones, drawings, logos, graphs, maps etc.

Second Modification: Reconstruction of Broken Drawing Lines: It has also been observed that sometimes non-text components consist of broken drawing lines, by choice or because of document digitization errors (like low resolution, bad binarization etc.). Hole-filling morphological operation only fills those hollow drawing components which are composed of unbroken contour lines. Non-text components with broken drawing lines remain unfilled, even after hole-filling operation. Therefore, even the first modified version of Bloomberg's algorithm misclassifies them as text components, which is shown in the top row of Figure 5.6. The Bloomberg's algorithm can further be improved by reconstructing broken drawing lines before the hole-filling operation. At first, one might consider using a morphological closing operation with oriented structuring elements for drawing lines reconstruction. However, a morphological closing operation can not handle drawing line reconstruction and produces worse effect on the final non-text mask, as shown in the middle row of Figure 5.6. Here an efficient and easy technique is introduced for horizontal and vertical drawing line reconstruction, which can be generalized for a variety of line orientations. The presented drawing line reconstruction algorithm is described as follows: i) horizontal and vertical lines from the morphologically thinned 16×1 subsampled image are identified using a morphological hit-miss transform using horizontal and vertical structuring elements respectively, ii) the broken horizontal lines are blended together through anisotropic Gaussian smoothing with $\sigma_x > \sigma_y$, iii) the smoothed image is converted into the binarized image using global thresholding, which produces connected horizontal lines, iv) these line are labeled using connected components analysis, v) the broken horizontal lines are labeled with respect to the labeling of connected horizontal lines. vi) finally, all of the broken lines with the same label are joined together, resulting in reconstructed horizontal drawing lines, vii) the same procedure is repeated for reconstructing vertical drawing lines by smoothing with $\sigma_x < \sigma_y$. The modified Bloomberg's algorithm with reconstruction of broken drawing lines before hole-filling operation is shown in Figure 5.3(c). The results of reconstructed drawing lines using the aforementioned approach and its positive effect on final non-text mask separation are shown in the bottom row of Figure 5.6.

5.4 Experiments and Results

The performance of Bloomberg's original text/image segmentation algorithm and the modified versions are compared using standard datasets like the UW-III [GHHP97], the ICDAR-2009 page segmentation competition test images [APBP09] and the private circuit diagrams images. The main reason for using different datasets is to compare the text/image segmentation accuracy of these algorithms on different types of document images with a variety of text and non-text components. A total of 95 documents, mainly composed of text and halftone components, were selected from the UW-III dataset. The circuit diagrams dataset composed of 10 images having text and drawing components. The ICDAR 2009 dataset contains 8 test images with non-Manhattan layout, unlike the other selected datasets. For each dataset, pixel-level ground-truth images were generated using zone-level ground truth information. Each pixel in a ground-truth image contains

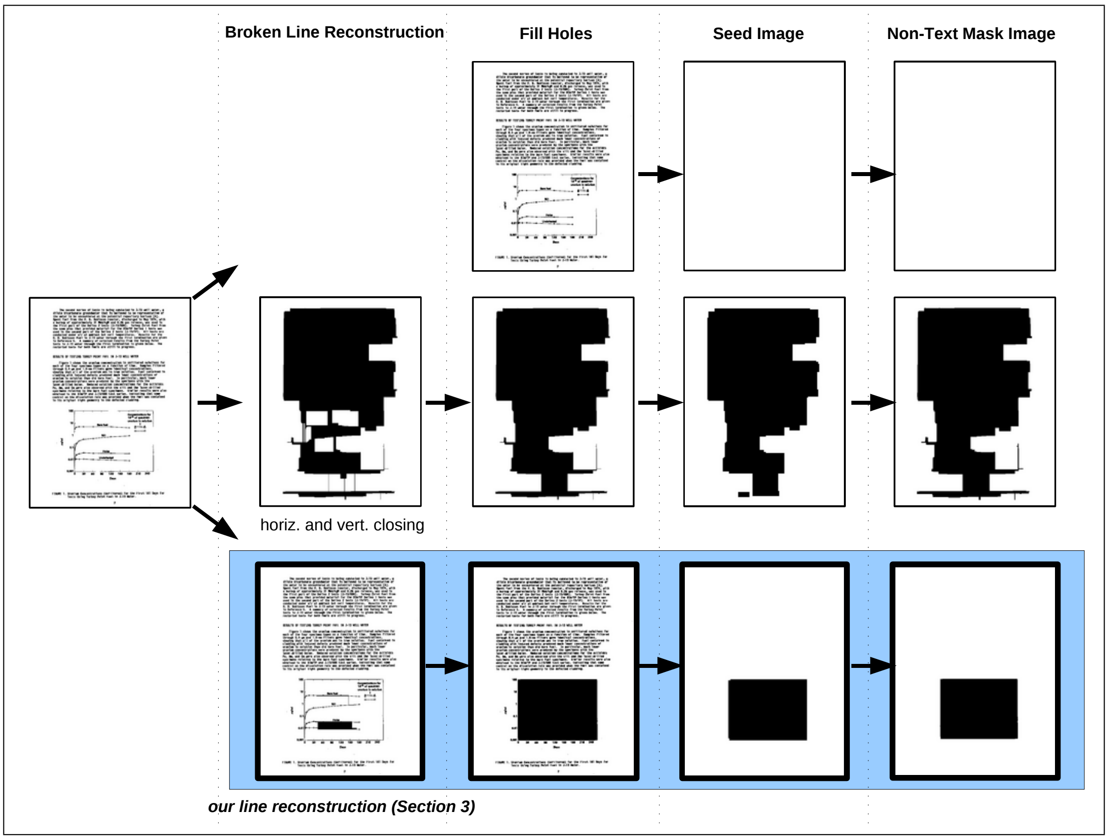


Figure 5.6: Second modified Bloomberg's text/image segmentation algorithm: reconstruction of broken drawing lines followed by hole-filling based modified Bloomberg's algorithm. **Top Row:** hole-filling operation on broken drawing lines image does not fill the hollow non-text components and therefore misclassifies non-text components as text. **Middle Row:** horizontal and vertical closing based line reconstruction produces a garbage image and a garbage non-text mask. **Bottom Row:** the presented broken line reconstruction method (described in Section 3) generates closed contour drawing shapes, which help in producing an accurate non-text mask.

Table 5.1: Performance evaluation of the original Bloomberg's text/halftone segmentation algorithm and the improved versions on the UW-III dataset (95 document images) and the ICDAR 2009 page segmentation competition test dataset (8 document images). The improved versions achieved better segmentation accuracy than the original version.

	UW-III			ICDAR-2009		
	Original	1st version	2nd version	Original	1st version	2nd version
non-text classified as non-text	95.36%	99.39%	99.51%	85.62%	91.44%	98.41%
text classified as text	99.79%	99.28%	99.19%	100%	99.11%	99.42%
segmentation accuracy	97.58%	**99.34%**	**99.35%**	92.81%	**95.28%**	**98.92%**

either a text or non-text label.

Different types of metrics were used for the performance evaluation of text/image segmentation algorithm, as defined below:

1. **non-text classified as non-text:** percentage of intersection of non-text pixels in both segmented image and ground-truth image with respect to the total number of non-text pixels in the ground-truth image.

2. **text classified as text:** percentage of intersection of text pixels in both segmented image and ground-truth image with respect to the total number of text pixels in the ground-truth image.

3. **segmentation accuracy:** average percentage of non-text classified as non-text and text classified as text accuracy.

Based on the metrics defined above, the comparison among the original Bloomberg's text/image segmentation algorithm and the modified versions are shown in Table 5.1 and 5.2. It is clearly visible in Table 5.1 and 5.2 that the modified versions achieved better segmentation accuracy as compared to the original Bloomberg's algorithm.

5.5 Conclusion

Bloomberg's text/image segmentation algorithm [Blo91] is specifically designed for text and halftone image separation. It is simple and fast approach and performs well on text and halftone image segmentation, but it is unable to segment text and non-text components other than halftones, such as drawings, graphs, maps, etc. In this chapter, the modifications to the original Bloomberg's algorithm are presented for making it a general text and non-text image segmentation approach, where non-text components can be

Table 5.2: Performance evaluation of the original Bloomberg's text/halftone segmentation algorithm and the improved versions on circuit diagram dataset (10 document images). The improved versions achieved significantly better segmentation accuracy than the original version.

	Circuit Diagrams		
	Original	1st version	2nd version
non-text classified as non-text	0%	89.11%	90.31%
text classified as text	100%	100%	96.67%
segmentation accuracy	50%	**94.56%**	**93.49%**

halftones, drawings, maps, graphs, etc. The original Bloomberg's approach and the modified versions are evaluated on standard datasets like the UW-III and the ICDAR 2009 page segmentation competition test images as well as the circuit diagram dataset. The modifications result in significant improvements over the original Bloomberg's text/image segmentation algorithm, as shown in Table 5.1 and Table 5.2. The presented improvements results in a increase in the aggregated segmentation accuracy of 93% (n = 113 documents) to 99%.

Part III

Preprocessing of Degraded Camera-Captured Document Images

Chapter 6

Binarization[1]

Summary: *this chapter presents a new adaptive binarization technique for degraded hand-held camera-captured document images. The state-of-the-art locally adaptive binarization methods are sensitive to the values of free parameters. This problem is more critical when binarizing degraded camera-captured document images because of distortions like non-uniform illumination, bad shading, blurring, smearing and low resolution. In this chapter, it is demonstrated that local binarization methods are not only sensitive to the selection of free parameters values, but also sensitive to the constant values of free parameters for all pixels of a document image. However, regarding binarization of degraded document, it has been observed that some range of values of free parameters are better for foreground regions and some other range of values are better for background regions. For improving binarization of degraded camera-captured documents, this chapter presents an adaptation of a state-of-the-art local binarization method such that two different set of free parameters values are used for foreground and background regions, respectively. In a document image, foreground regions can be categorized into two main classes, text and non-text. Text elements are more important than non-text elements with respect to Optical Character Recognition (OCR) accuracy, which is one of the main focuses of this chapter. The ridge-based text line detection method, which is presented in Chapter 3, can be applied directly to grayscale documents for detecting text regions.This information is then used to calculate appropriate threshold using different set of free parameters values for the foreground and background regions, respectively. Evaluation of the method using an OCR-based measure and a pixel-based measure show that the presented*

[1] This work was published in Bukhari et al. [BSB09c] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. Adaptive Binarization of Unconstrained Hand-Held Camera-Captured Document Images. Journal of Universal Computer Science (JUCS), 15(18):3343-3363, 2009. Copyright ©J.UCS"*. This chapter is an adapted version of the published work.

method achieves better performance as compared to the state-of-the-art global and local binarization methods.

6.1 Introduction

Scanners are traditionally and widely used in document image capturing for document analysis systems like OCR. Scanners produce planar document images with a high resolution. For decades, many novel approaches have been proposed for planar document image segmentation [SKB08b] and OCR [MSY92]. Nowadays cameras are available widely at low cost and embedded with around all mobile devices, that offer fast, flexible and non-contact document imaging. On one hand, these advantages make camera a potential substitute of scanner for document capturing and they also open doors for many new applications such as mobile OCR, digitizing thick books, digitizing fragile historical documents, finding text-in-scene-images, etc. On the other hand, the quality of unconstrained hand-held camera-captured document images is lower than the quality of scanned document images because of degradations which are not very common in scanned images such as perspective distortions, non-uniform shading, image blurring, character smearing (due to low resolution) and lighting variations.

In the case of scanned document images, most of the state-of-the-art document analysis systems have been designed to work on binary document images [CCMM98a]. Therefore document image binarization is an important initial step in most of the scanned document image processing tasks such as OCR [MSY92], page segmentation [SKB08b], layout analysis [SBKB08], etc.

In the case of camera-captured document images, current OCR systems which are designed for scanner based planar document images do not have a capability to deal with geometric and perspective distortions. Therefore, current OCR systems give poor performance when applied directly to warped camera-captured document images. Designing dewarping techniques for flattening warped document images is a possible solution for improving the performance of OCR systems on camera-captured document images. Over last decade, different approaches have been proposed for document image dewarping [LDL05, SB07]. These approaches can be divided into two main categories based on document capturing methodology: (i) approaches in which specialized hardware arrangement like stereo-camera is required for 3D shape reconstruction of warped documents [CDL03,BS04,TZZX06] and (ii) approaches in which dewarping method is designed for images which are captured using a single hand-held camera in an uncontrolled environment [ZT03,LT06a,LCK05,FWL+07,ULB05,GPN07,BSB09a]; also referred to as monocular dewarping techniques. Most of these monocular dewarping techniques work on binarized images.

This discussion concludes that binarization is one of the most important initial steps for both scanned and camera-based document image analysis. Binarization of handheld camera-captured document images is a more challenging task than binarization of scanned document images because of one or more of the following common distortions in camera-captured document images: bad shading, blurring, non-uniform illumination and low resolution.

6.1.1 Related Work

In the literature, many different approaches have been proposed for binarization of grayscale document images [Ots79, WR83, Ber86, Nib86, O'G94, SP00, Kim04, GPP06, LT07, SKB08a] as well as color document images [SKPB00, TL02, BNP06]. Grayscale binarization techniques are used more frequently than color binarization, because grayscale binarization techniques can be applied to color documents by first converting them into grayscale. Grayscale binarization approaches can be classified into two main groups: i) global binarization methods and ii) local binarization methods.

Global binarization methods (like Otsu [Ots79]) estimate a single threshold value for binarization of a document such that each pixel in the document is assigned either to foreground or background based on the estimated threshold. Some researchers [SS04, BP05] have evaluated different state-of-the-art global binarization methods and reported that Otsu binarization method [Ots79] is better than other types of global binarization techniques. Global binarization methods are computationally inexpensive and perform better for typical scanned document images. However, they produce marginal noise artifacts [SvBKB08] if grayscale documents contains non-uniform illumination, which is usually present in the case of scanned thick book, scanned historical documents and degraded camera-captured documents.

Local binarization methods [Ber86, Nib86, O'G94, WR83, SP00] try to overcome these problems by calculating threshold values for each pixel differently using local neighborhood information. Sauvola binarization method [SP00] is one the widely used local binarization methods. Generally, local binarization methods perform better than global binarization methods on degraded document images but are computationally slow, sensitive to the selection of free parameter values [RSB09] and do not work well for degraded camera-captured document images.

In recent years, some special global binarization and local binarization techniques [GPP06, LT07] have been proposed for improving the binarization of degraded historical and camera-captured document images. Gatos et. al [GPP06] proposed local binarization method for scanned degraded historical document images. This technique has not yet been tested on blurred and low-resolution camera-captured document images. Lu and Tan [LT07] proposed global binarization method for camera-captured document images. Their method is based on the assumption that document image contains uniform illumination and uniform background, which is not usually the case.

In the literature, researchers have also presented some guided binarization methods. Kim [Kim04] proposed multi-window based local binarization method for camera-captured document images, which is a modification of Sauvola binarization method. He highlighted that a single window size, with respect to a local adaptive binarization method, is not good for a complete document image. Instead of that, he proposed that the appropriate window size for each pixel is estimated using a preliminary binarization result. This approach contains more free parameters than Sauvola binarization method. A similar binarization approach is presented by Yang et. al [YKJ10] for barcode images, where the window size for each pixel is guided by edge structure and other statistical information.

Most of the state-of-the-art local binarization methods apply same values of free parameters for all pixels in a document image. It has been observed in this chapter that the quality of binarization of degraded camera-captured documents can be improved by applying different values of free parameters for foreground and background regions, respectively. This chapter presents a local guided binarization technique for degraded grayscale camera-captured document images that selects different values of free parameters for pixels that belong to roughly estimated foreground regions and pixels that belong to roughly estimated background regions, respectively. The presented method can handle grayscale distortions like bad shading, blurring, low resolution and non-uniform illumination, and it is less sensitive to the selection of free parameter values as compared to other well know existing local binarization methods. In this chapter, ridges based text line regions detection method (as described in Chapter 3) is used for estimating foreground (text and drawing-line) regions.

The rest of this chapter is organized as follows: Section 6.2 explains the sensitivity of binarization over the selection of free parameters values. Section 6.3 describes the technical details of the presented binarization method. Section 6.4 deals with experimental results and Section 6.5 describes the conclusion.

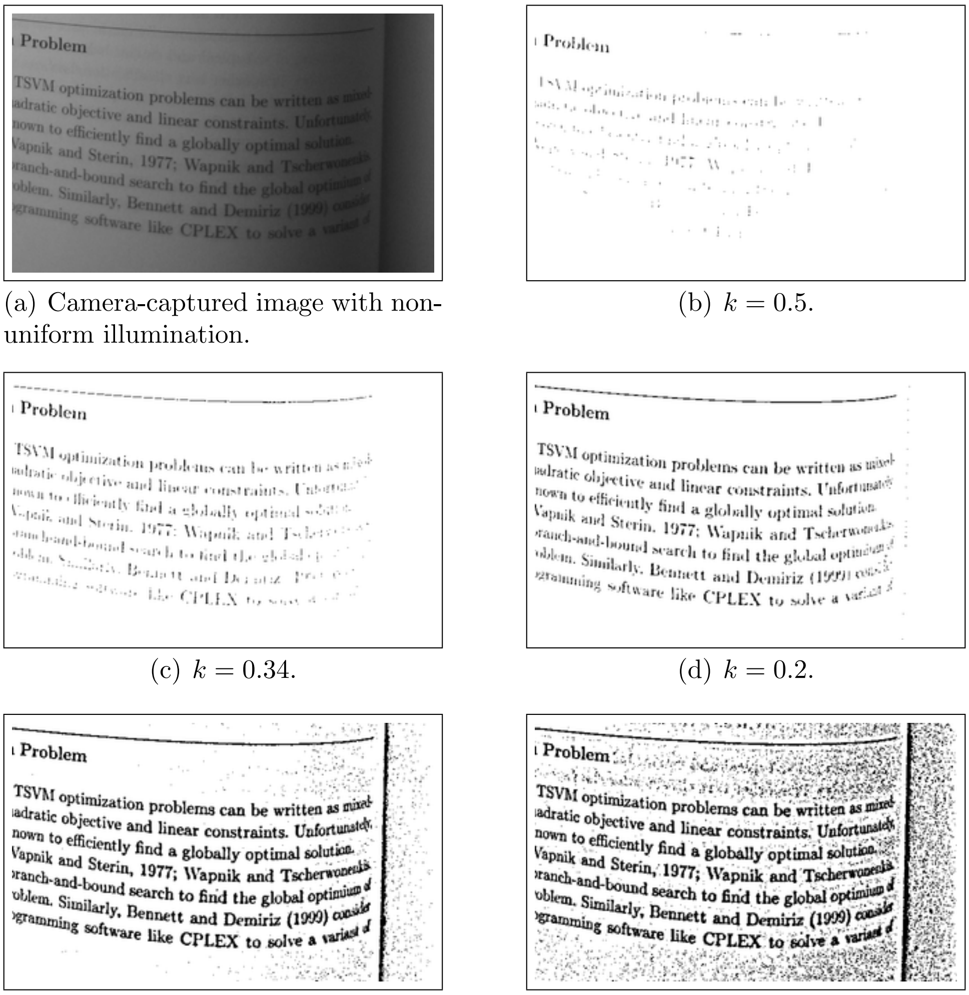
6.2 Sensitivity of Local Binarization Methods to Selection of Free Parameters Values

Each local binarization method contains some free or tunable parameters. Usually, the suitable values of the free parameters of any local binarization method highly depend on the context of targeted applications and type of documents. For achieving high performance on heterogeneous documents, manual procedures for estimating free parameters values can be used, though is not a suitable way. Some techniques have already been proposed in the literature for automatic estimation of free parameters values [RSB09, BP05], but the concern of their work is to estimate best parameter values which can be fixed for all pixels in a document image.

This section demonstrates some issues related to local binarization methods due to the use of fixed values of free parameters, found manually or automatically, for all the pixels in a document image. For demonstration purpose, Sauvola binarization method is used, which is one of the widely used local binarization methods [SS04, BP05]. For a pixel (x, y), the threshold $t(x, y)$ in Sauvola binarization method is computed using mean $\mu(x, y)$ and standard deviation $\sigma(x, y)$ of the pixel intensities in a $w \times w$ window centered around the pixel (x, y). The formula for computing the threshold $t(x, y)$ is shown in Equation 6.1.

$$t(x, y) = \mu(x, y) \left[1 + k \left(\frac{\sigma(x, y)}{R} - 1 \right) \right], \tag{6.1}$$

where R is the maximum value of the standard deviation ($R = 128$ for a grayscale document), and k is a parameter which takes positive values. The formula (Equation 6.1) has been designed in such a way that, the value of the threshold is adapted according to contrast in the local neighborhood of the pixel using local mean $\mu(x, y)$ and local standard deviation $\sigma(x, y)$. Because of that, it tries to estimate an appropriate threshold $t(x, y)$ for each pixel under both possible conditions: high and low contrast. In the case of high contrast region ($\sigma(x, y) \approx R$), the threshold $t(x, y)$ is nearly equal to $\mu(x, y)$. In a quite low contrast region ($\sigma << R$), the threshold goes below the mean value thereby successfully removing the relatively dark regions of the background. The parameter k controls the value of the threshold in the local window such that the higher the value of k the lower the threshold from the local mean $m(x, y)$.



(a) Camera-captured image with non-uniform illumination.

(b) $k = 0.5$.

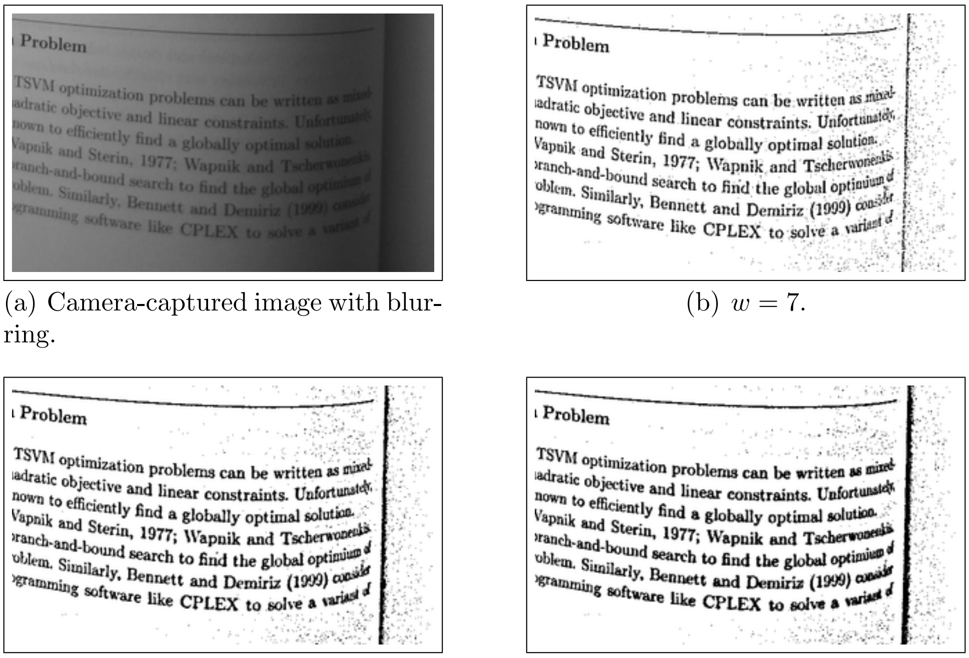
(c) $k = 0.34$.

(d) $k = 0.2$.

(e) $k = 0.05$.

(f) $k = 0.02$.

Figure 6.1: Sauvola binarization results for different values of k, with fixed $w = 15$. Some of the best reported values of k are: $k = 0.5$ (Sauvola [SP00] and Sezgin [SS04]) and $k = 0.34$ (Badekas et al. [BP05]). Some more values (like $k = 0.2$, $k = 0.05$ and $k = 0.02$) are also added for this experiment. Observations: for $k >= 0.2$ binarized results contain clean background regions and broken foreground characters; for $k <= 0.05$ results contain unclean background regions and unbroken/joined foreground characters.



(a) Camera-captured image with blurring.

(b) $w = 7$.

(c) $w = 15$.

(d) $w = 21$.

Figure 6.2: Sauvola binarization results for different values of w with fixed $k = 0.05$.

The statistical constraint in Equation 6.1 gives acceptable results even for degraded documents. But there is no consensus regarding the appropriate value of k in research community. Badekas et al. [BP05] experimented with different values and found that $k = 0.34$ gives the best results, but Sauvola [SP00] and Sezgin [SS04] proposed $k = 0.5$. This indicate that a suitable value of parameter k should be found experimentally for a targeted collection of documents.

Here, Sauvola binarization method is analyzed for different values of k (with fixed w) and different values of w (with fixed k) for degraded camera-captured document images. Some experimental results are shown in Figure 6.1 and Figure 6.2 for different values of k and w, respectively. As shown in Figure 6.1, Sauvola binarization method is sensitive to the selection of appropriate value of k. But, Sauvola binarization method does not very much sensitive to the value of w, as shown in Figure 6.2. Therefore in this chapter, the sensitivity of k for Sauvola binarization method is further analyzed. Additionally, already reported values of k, i.e $k = 0.5$ [SS04, SP00] and $k = 0.34$ [BP05], do not give acceptable results under blurring or non-uniform illuminations in degraded camera-captured document images as shown in Figure 6.1.

From these experiments (Figure 6.1), it has been noticed that, on one hand small values of k (like $k <= 0.05$) yield low noise in the background regions but they produce broken characters. On the other hand, relatively large values of k (like $k >= 0.2$) give good results for foreground regions with unbroken characters but they produce noise in background regions (Figure 6.1).

6.3 Foreground-Background Guided Binarization

The experiments in the previous section clarify that Sauvola binarization method can perform better on degraded camera-captured document images if two different sets of values of free parameters are used for binarization. For example, in case of Sauvola binarization, a small value of k is used for those pixels which are roughly estimated as foreground pixels and a large value of k is used otherwise. In this chapter, Sauvola binarization method is modified to incorporate the stated logic. The presented approach can also be applied to other types of local binarization methods. Technical details of the presented method is described here.

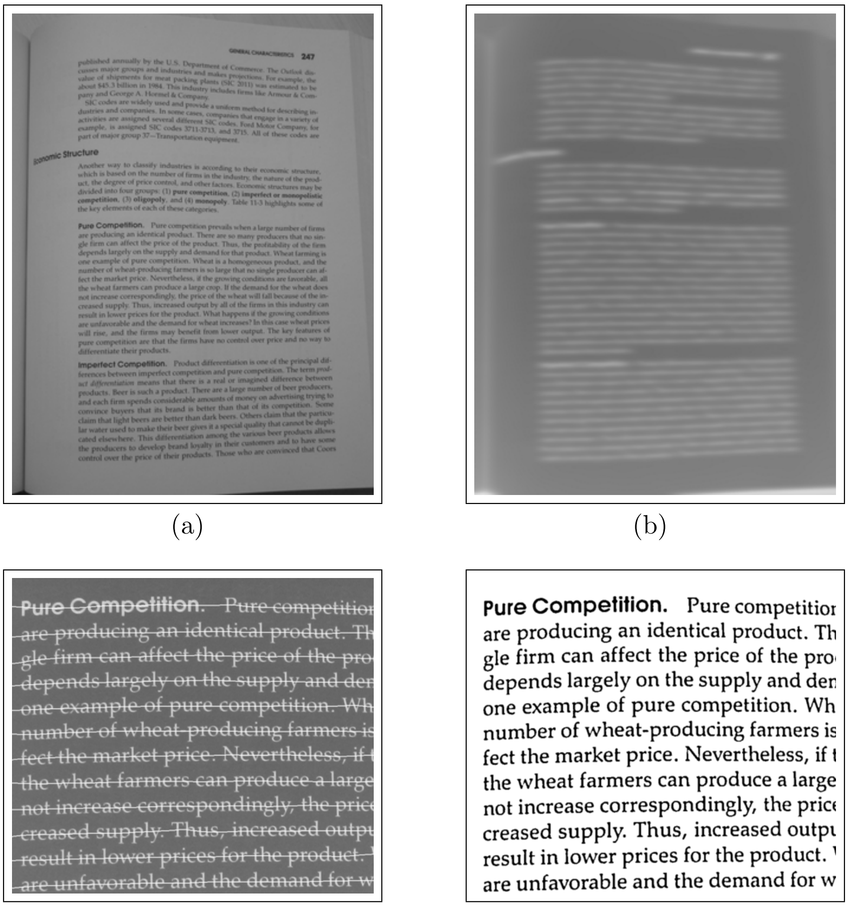
6.3.1 Foreground Regions Detection

As a first step of the presented binarization method, foreground regions are roughly estimated in a grayscale camera-captured document image. With respect to OCR performance, text elements can be considered more important than non-text elements in a document image. For detecting foreground text regions in the grayscale document image, the ridge-based text line region extraction method is used, which is described in Chapter 3.

Detected ridges over the smoothed image of Figure 6.3(b) are shown in Figure 6.3(c). It is clearly visible in Figure 6.3(c) that, each ridge covers the central line structure of a foreground object.

6.3.2 Foreground-Background Guided Local Binarization

It has already been discussed in Section 6.2 that no single value of parameter k in Sauvola binarization method is suitable for different types of degraded camera-captured documents. However, according to the experiment in Figure 6.1, a small value of k (like $k <= 0.05$) gives better results for foreground regions but with background noise and a comparatively large value of k (like $k >= 0.2$) give noise free background regions but with broken characters. As shown in Figure 6.3(c), the detected ridges are present near foreground regions. Therefore, instead of using a fixed value of k for all pixels, different



(a)

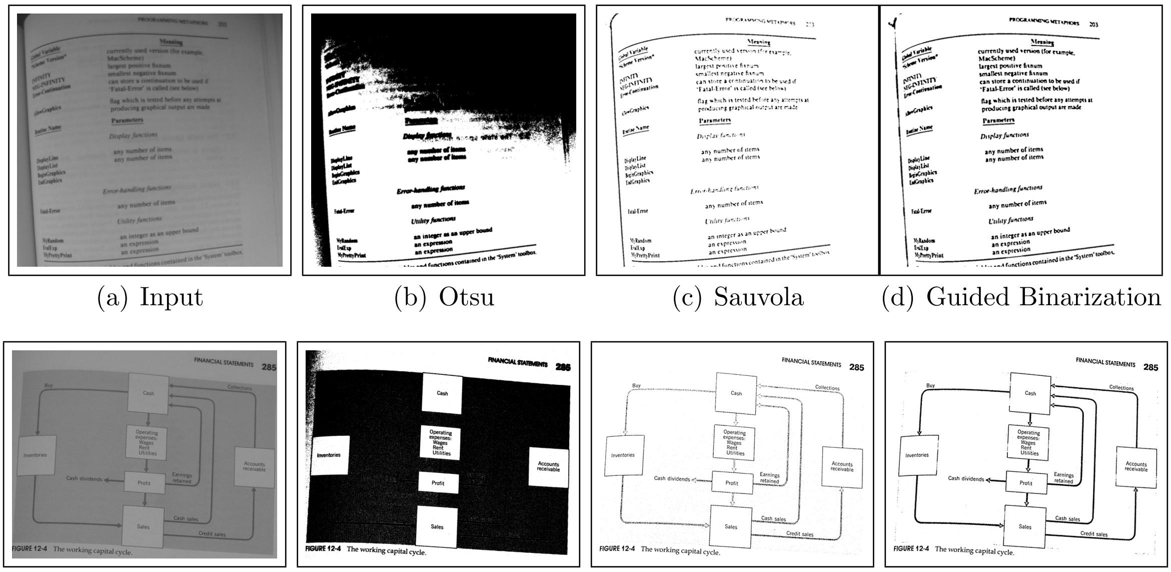
(b)

Pure Competition. Pure competitior
are producing an identical product. Th
gle firm can affect the price of the pro
depends largely on the supply and der
one example of pure competition. Wh
number of wheat-producing farmers is
fect the market price. Nevertheless, if t
the wheat farmers can produce a large
not increase correspondingly, the price
creased supply. Thus, increased outpu
result in lower prices for the product. '
are unfavorable and the demand for w

(c)

(d)

Figure 6.3: Snapshots of the presented binarization method. (a) Input Image, (b) Smoothed Image generated by using match filter bank approach, (c) Horn-Riley method [Hor70, Ril87b] is used for detecting ridges, which are visible in the zoom area of document image, (d) Result of foreground-background guided Sauvola binarization (zoom area of document image).



(a) Input (b) Otsu (c) Sauvola (d) Guided Binarization

(e) Input (f) Otsu (g) Sauvola (h) Guided Binarization

Figure 6.4: Binarization results of Otsu, Sauvola and the presented guided binarization methods. It is important to note that Otsu binarization results contain large amount of noise. For Sauvola binarization, manually selected parameter values are used; these are $w = 15$ and $k = 0.15$. It is also important to note that Sauvola binarization results contain broken characters for blurred documents. The presented guided binarization method shows better results in the presence of document image degradations like blurring.

values of k can be used for foreground and background regions, respectively, to improve binarization results. For this purpose, Sauvola binarization method is redefined in Equation 6.2.

$$t(x,y) = \mu(x,y)\left[1 + k(x,y)\left(\frac{\sigma(x,y)}{R} - 1\right)\right], \tag{6.2}$$

where $k(x,y) = 0.05$ if a ridge found in the local neighborhood window, otherwise $k(x,y) = 0.2$. After thresholding, median filter can also be applied to further remove the salt and pepper noise. A sample binarization results of the presented foreground-background guided Sauvola method are shown in Figures 6.3(d). Binarization results of Otsu, Sauvola and the presented foreground-background guided Sauvola binarization methods on sample document images are shown in Figure 6.4 and Figure 6.7.

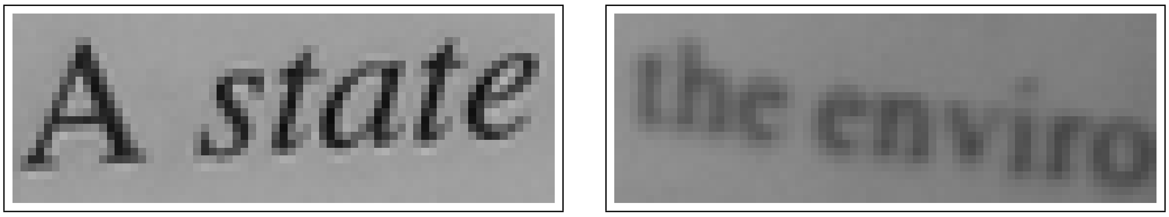


Figure 6.5: High vs low resolution image comparison: (left) 6 mega-pixels high resolution camera-captured image and (right) 2 mega-pixels low resolution camera-captured image.

6.4 Experiments and Results

The performance of the presented binarization method is evaluated on both low and high resolution degraded camera-captured documents. Here, two different types of experiments have been conducted for performance evaluation.

- **OCR-based evaluation** for high resolution degraded camera-captured documents
- **pixel-based evaluation** for low resolution degraded camera-captured documents

One can distinguish between the quality of high and low resolution of grayscale camera-captured document images, which are used here, in Figure 6.5. For performance evaluation, $k = 0.05$ is used for pixels near roughly estimated foreground regions and $k = 0.2$ otherwise. The robustness of the presented method is also demonstrated over different sets of values of k for foreground and background regions, respectively, for pixel-based evaluation (Section 6.4.2).

6.4.1 OCR-based Evaluation

OCR-based evaluation is important for comparing the presented method with different state-of-the-art binarization methods. OCR-based evaluation can also be considered as goal-oriented evaluation, because OCR is usually the results in most of the document analysis tasks.

Here, the presented binarization approach is evaluated on a subset of the hand-held camera-captured document images dataset (the DFKI-I dataset) which were used in the CBDAR 2007 document image dewarping contest [SB07]. The resolution of each document image is around 6 mega-pixels. ASCII text ground-truth is provided with the dataset. For performance evaluation, a subset of 10 degraded documents are selected from the DFKI-I dataset. The state-of-the-art Otsu and Sauvola binarization methods are used for OCR-based comparative evaluation.

The OCR error rates of all three binarization methods are compared after dewarping. As mentioned earlier in the introduction, so far commercially available OCR softwares are designed for planar document images and may produce bad results for warped camera-captured documents. Therefore, the binarization results are first converted into planar form by applying a dewarping method. For this purpose, the dewarping method that is introduced in Chapter 8 is used here. Then, the dewarped documents are processed through the commercial OCR system ABBYY Fine Reader 9.0. Finally, the OCR error rates of all three binarization methods (Otsu, Sauvola and the presented guided binarization method) are evaluated using the block edit distance[2]. Table 6.1 shows the comparative results of all three methods with respect to mean edit distance and the number of documents for each algorithm on which it has the lowest edit distance (in case of tie, all algorithms having the lowest edit distance are scored for that document). It is visible in Table 6.1 that the presented binarization method achieves lowest mean edit distance as well as performed better than other methods on a large number of document images.

Table 6.1: OCR error rates of different binarization algorithms on a subset of the DFKI-I dataset using ABBYY Fine Reader 9.0 as preferred OCR system. The guided binarization method achieved the lowest error rate on the majority of document images as compared to the state-of-the art binarization methods (Otsu and Sauvola).

Algorithm	**Mean Edit Distance %**	**Number of documents**[a]
Otsu Binarization	6.96	2
Sauvola Binarization[b]	4.92	3
Guided-Binarization	**4.62**	**5**

[a] Number of documents for each algorithm on which it has the lowest edit distance.

[b] manually selected: ($w = 15$, $k = 0.15$); tested different values for k in between 0.1 to 0.5 and found 0.15 is the best for the given dataset.

6.4.2 Pixel-based Evaluation

Pixel-based evaluation has been inspired from Document Image Binarization COntest (DIBCO-2009) [GNP09] in which the binarization result of an algorithm is compared with

[2] http://sites.google.com/site/ocropus/release-notes

semi-automatically generated ground-truth binary image. DIBCO dataset consists of 10 scanned images with distortions like smudge, bleed-through, show-through and shadows. As compared to degraded scanned document images, camera-captured document images contain different types of degradations like non-uniform illumination, blurring, smearing of characters at low resolution and bad-shading. Therefore, a custom dataset is used here for pixel-based evaluation which is a representative of above mentioned degradations.

Similar to OCR-based evaluation, here also the presented binarization method is compared with different state-of-the-art global (Otsu [Ots79]) and local (Sauvola [SP00]) binarization methods. The custom dataset, ground-truth generation process and evaluation measure are presented below.

Dataset

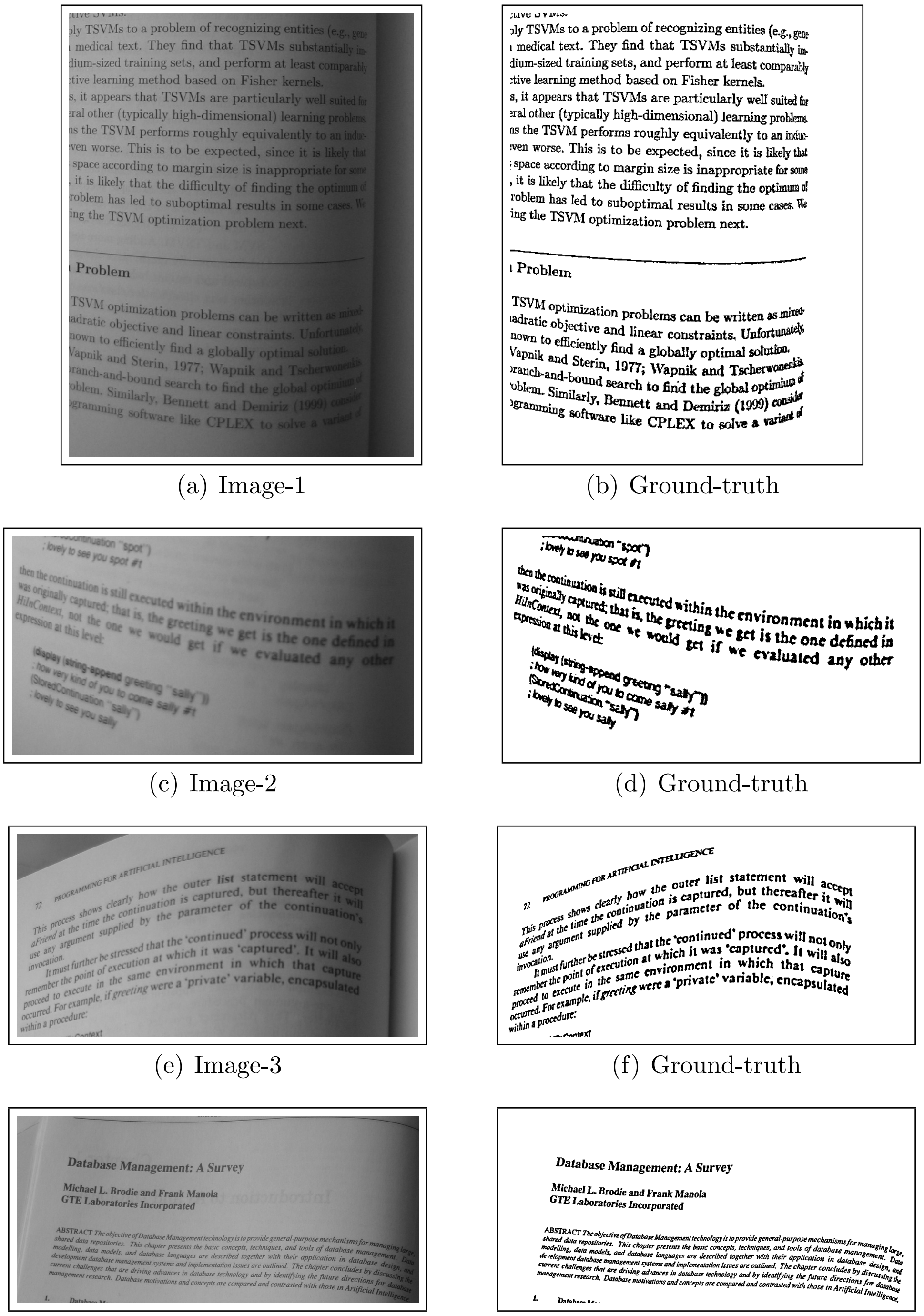
A small number of camera-captured texts are selected that contain distortions like bad-shading, non-uniform illumination, blurring and smearing of characters at low resolution from four different camera-captured document images. Furthermore, these document images have been captured at low resolution of two mega-pixels as compared to document images captured at high resolution of six mega-pixels for OCR-based evaluation (Section 6.4.1). The dataset and corresponding ground-truth images are shown in Figure 6.6.

Ground-Truth Generation

Ground-truth binarized images are generated through a semi-automatic process. In this process, different binarized results, that were generated using Sauvola binarization method with different combinations of parameter values of k and w, are manually examined. These results show that for the given dataset $k = 0.02$ and $w = 15$ preserve character strokes at foreground regions better than other values of k and w. However, this combination of k and w produces noise in background regions. Therefore, binary ground-truth images are generated in two stages: first, Sauvola binarization method is applied with $k = 0.02$ and $w = 15$. Then, noise is manually removed from background regions. The semi-automatically generated binary ground-truth images are shown in Figure 6.6 with their corresponding grayscale images.

Evaluation Measure

One of the evaluation measures, ‘F-measure’, mentioned in [GNP09] is used here for comparison among different binarization methods. The F-measure is described below



ctive SVMs.
ply TSVMs to a problem of recognizing entities (e.g., gene
medical text. They find that TSVMs substantially im-
dium-sized training sets, and perform at least comparably
ctive learning method based on Fisher kernels.
s, it appears that TSVMs are particularly well suited for
eral other (typically high-dimensional) learning problems.
as the TSVM performs roughly equivalently to an induc-
ven worse. This is to be expected, since it is likely that
space according to margin size is inappropriate for some
it is likely that the difficulty of finding the optimum of
roblem has led to suboptimal results in some cases. We
ing the TSVM optimization problem next.

Problem

TSVM optimization problems can be written as mixed-
adratic objective and linear constraints. Unfortunately,
nown to efficiently find a globally optimal solution.
Vapnik and Sterin, 1977; Wapnik and Tscherwonenkis
ranch-and-bound search to find the global optimum of
oblem. Similarly, Bennett and Demiriz (1999) conside
gramming software like CPLEX to solve a variant of

(a) Image-1 (b) Ground-truth

continuation "spot")
; lovely to see you spot #1

then the continuation is still executed within the environment in which it was originally captured; that is, the greeting we get is the one defined in HilnContext, not the one we would get if we evaluated any other expression at this level:

(display (string-append greeting "sally"))
; how very kind of you to come sally #1
(StoredContinuation "sally")
; lovely to see you sally

(c) Image-2 (d) Ground-truth

72 PROGRAMMING FOR ARTIFICIAL INTELLIGENCE

This process shows clearly how the outer list statement will accept aFriend at the time the continuation is captured, but thereafter it will use any argument supplied by the parameter of the continuation's invocation.

It must further be stressed that the 'continued' process will not only remember the point of execution at which it was 'captured'. It will also proceed to execute in the same environment in which that capture occurred. For example, if greeting were a 'private' variable, encapsulated within a procedure:

Context

(e) Image-3 (f) Ground-truth

Database Management: A Survey

Michael L. Brodie and Frank Manola
GTE Laboratories Incorporated

ABSTRACT *The objective of Database Management technology is to provide general-purpose mechanisms for managing large, shared data repositories. This chapter presents the basic concepts, techniques, and tools of database management. Data modelling, data models, and database languages are described together with their application in database design, development database management systems and implementation issues are outlined. The chapter concludes by discussing the current challenges that are driving advances in database technology and by identifying the future directions for database management research. Database motivations and concepts are compared and contrasted with those in Artificial Intelligence.*

1. Database Man

(g) Image-4 (h) Ground-truth

Figure 6.6: Dataset and Ground-Truth: 4 image portions have been selected from low resolution (2 mega-pixels) camera-captured document images, which contain degradations like, blurring, non-uniform illumination, bad-shading and smearing. Binary ground-truth images have been generated by using semi-automatic process described in Section 6.4.2

Table 6.2: Pixel-based performance evaluation of different binarization methods using low resolution dataset mentioned in Figure 6.6. The guided binarization method achieved better FMeasure (%) than the state-of-the-art binarization methods (Otsu and Sauvola).

	FMeasure (%)				
	Image-1	Image-2	Image-3	Image-4	Average
Otsu Binarization	32.71	22.41	27.64	54.79	**34.39**
Sauvola Binarization[a]	90.33	89.77	87.82	93.55	**90.37**
Guided-Binarization[b]	90.74	93.10	90.66	92.19	**91.67**

[a] manually selected: ($w = 15$ and $k = 0.05$); tested different values for widow-size and k and found ($w = 15$ and $k = 0.05$) is the best for the given dataset.

[b] manually selected: ($w = 15$ and $k = 0.02$ in the presence of ridge(s) otherwise $k = 0.2$)

in Equations 6.3, 6.4 and 6.5, where TP, FP, and FN represent the true-positive (total number of matched foreground pixels), false-positive (total number of misclassified foreground pixels in binarization result as compared to ground-truth) and false-negative (total number of misclassified background pixels in binarization result as compared to ground-truth) values, respectively.

$$\text{FMeasure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{6.3}$$

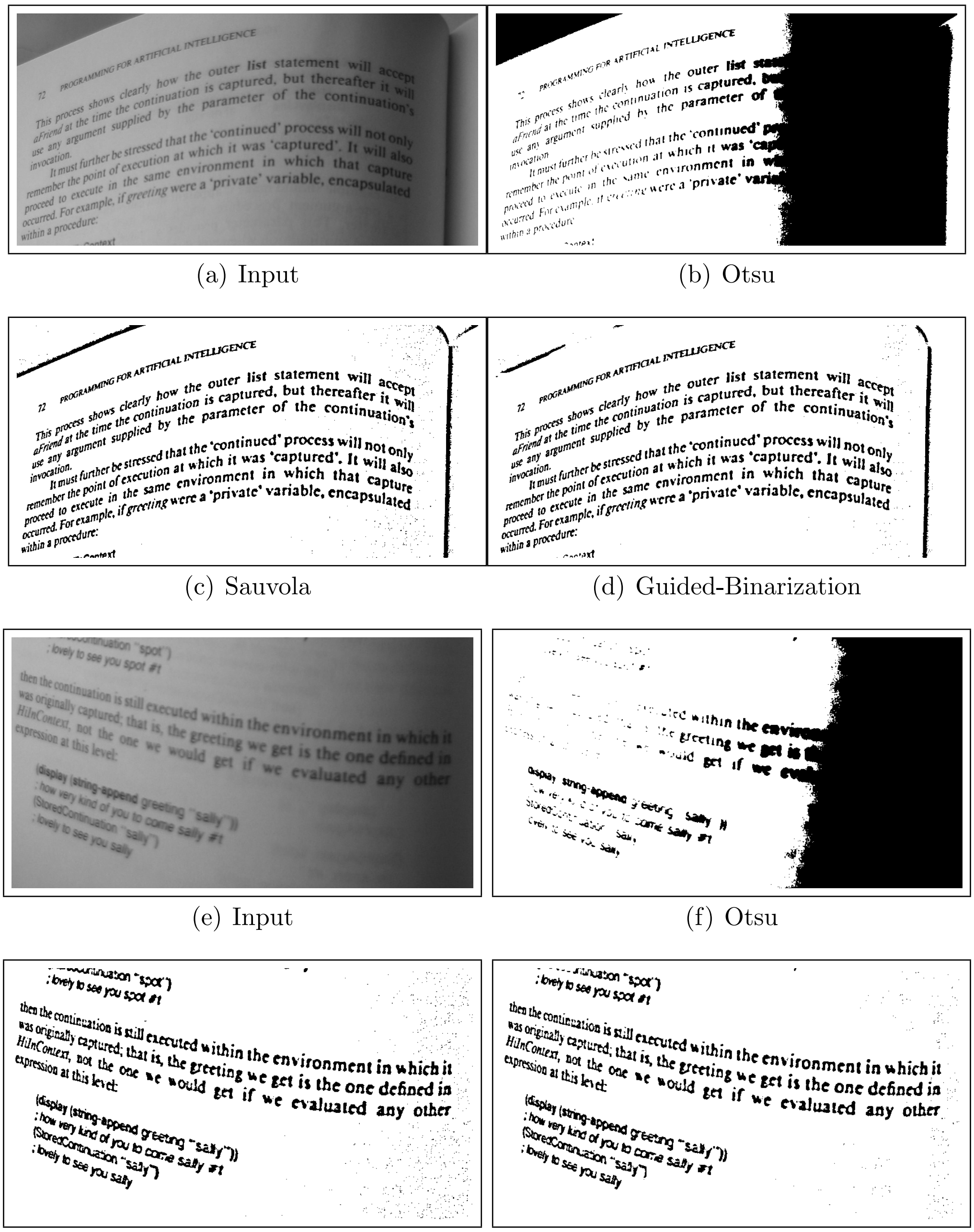
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6.4}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6.5}$$

Analysis

Based on the above mentioned setup for pixel-based evaluation, comparative results of Otsu binarization, Sauvola binarization and the presented guided-binarization methods are shown in Table 6.2. For Sauvola binarization method, $k = 0.05$ and $w = 15$ are the best empirically selected values for given dataset. It is mentioned in the algorithm description (Section 6.3.2) that median-filter can also be applied after binarization, but for pixel-based evaluation, raw results are used to do a fair comparison. Binarization results on some of the images in the dataset (Figure 6.6) are shown in Figure 6.7.

The sensitivity of Sauvola binarization method and robustness of the presented guided-



(a) Input

(b) Otsu

(c) Sauvola

(d) Guided-Binarization

(e) Input

(f) Otsu

(g) Sauvola

(h) Guided-Binarization

Figure 6.7: Binarization results of different algorithms on the low resolution dataset mentioned in Figure 6.6. For the results of Sauvola and the presented guided-binarization method in this figure, best parameter values for k have been selected manually, which have given good compromise between character-strokes and noise.

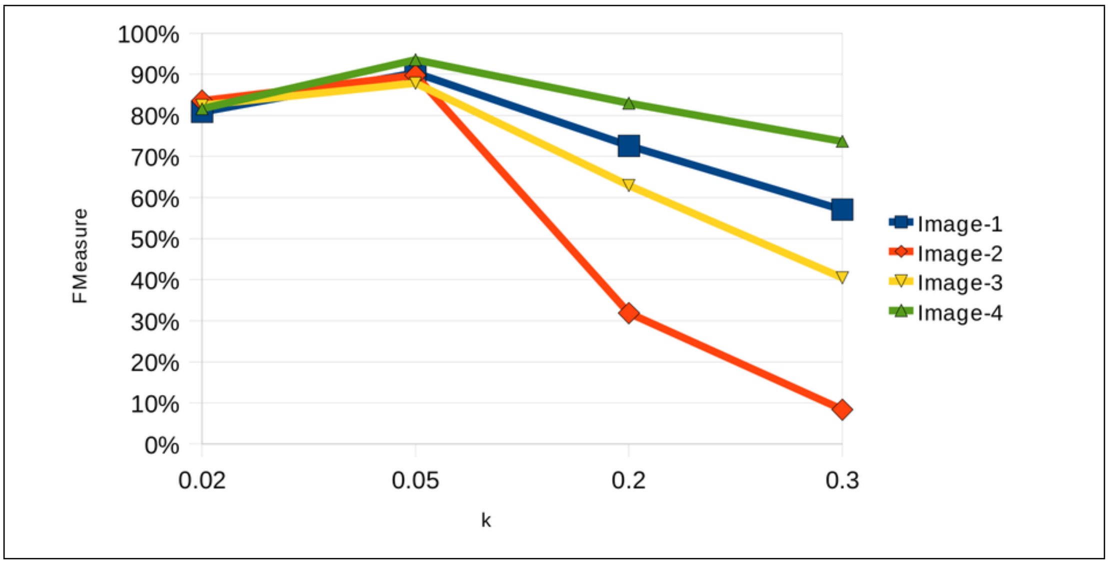


Figure 6.8: Analysis of the sensitivity of Sauvola binarization method with respect to the value of k over degraded low resolution camera-captured document images shown in Figure 6.6.

binarization method are also analyzed with respect to the different values of k. These experiment are also conducted on the same dataset shown in Figure 6.6. Figure 6.8 shows the pixel-based accuracy of Sauvola binarization method for different values of k. Similarly Figure 6.9 shows the pixel-based accuracy of the presented guided-binarization method for different values of pair of k. It is important to note that, Sauvola uses a single value of k while the presented guided method uses two values of k i.e. (k_r, k_nr). Therefore, the range of appropriate values of k is much larger for Sauvola's method than for the presented method. It can be concluded from Figure 6.8 that Sauvola binarization method is sensitive to the parameter selection for k in the presence of degradations in camera-captured document images, whereas the presented guided binarization method is robust against parameter selection for the pair (k_r, k_nr).

6.5 Conclusion

In this chapter, the sensitivity of fixing free parameters values of local binarization methods for all pixels in a camera-captured document image is explored. It has been demonstrated that no matter how to find the free parameters values (either manually or automatically), some range of values of free parameters gives better binarization results for foreground regions and some other range of values gives better binarization result

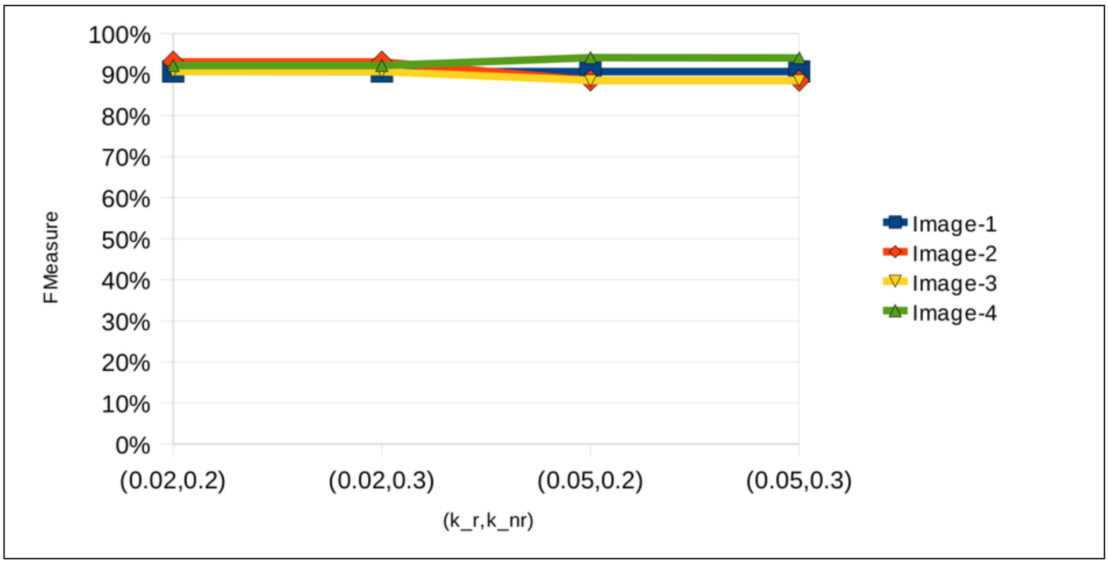


Figure 6.9: Analysis of the robustness of the presented guided binarization method with respect to the value of pair of k over degraded low resolution camera-captured document images shown in Figure 6.6 (Note: k_r: value of k in the presence of ridges; k_nr: value of k in the absence of ridges).

for background regions. This sensitivity is overcome by introducing the idea of not using the constant values of free parameters for all pixels, but use different values of free parameters for pixels belong to roughly estimated foreground and background regions. For this purpose, the idea of applying ridge-based text line extraction method (Chapter 3) for roughly estimating foreground text regions from grayscale documents. OCR-based and pixel-based performance evaluation techniques are used for comparing the presented foreground-background guided binarization method with other state-of-the-art binarization methods (Otsu and Sauvola). Here, an improvement over Sauvola binarization method is shown by selecting different values of parameter k for foreground and background regions, respectively. The presented idea of foreground-background guided binarization is also adaptable with other types of local binarization methods.

Chapter 7

Page Frame Detection[1]

Summary: *camera-captured document images usually contain two main types of marginal noise: textual noise (coming from neighboring pages) and non-textual noise (resulting from the page surrounding and/or binarization process). These types of marginal noise degrade the performance of the preprocessing (dewarping) of camera-captured document images and subsequent document digitization/recognition processes. Page frame detection is one of the newly investigated areas in document image processing, which is used to remove border noise and to identify the actual content area of document images. In this chapter, a new technique is presented for page frame detection of camera-captured document images by utilizing text and non-text contents information. The presented method is evaluated on the DFKI-I (the CBDAR 2007 Dewarping Contest) dataset. Experimental results show the effectiveness of the presented method in comparison to other state-of-the-art page frame detection approaches.*

7.1 Introduction

When a page of a book is photographed, the captured image usually contains undesired parts of text from the neighboring page. Besides, some regions of background (table surface etc.) also appear in the image. These undesired regions of the image are usually referred to as border noise [SB11]. These types of border noise are called textual noise

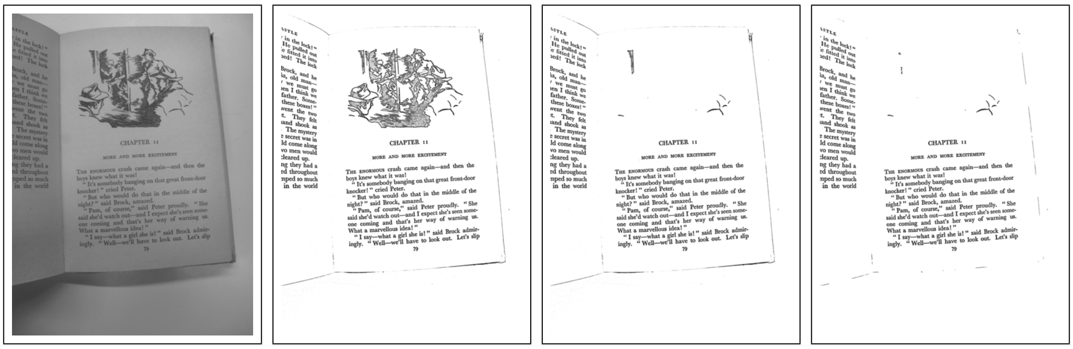
[1] This work was published in Bukhari et al. [BSB12a] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. An image based performance evaluation method for page dewarping algorithms using sift features. In Masakazu Iwamura and Faisal Shafait, editors, 4th International Workshop, CBDAR 2011 Beijing, China, September 22, 2011 Revised Selected Papers, volume 7139 of Lecture Notes in Computer Science, Image Processing, Computer Vision, Pattern Recognition, and Graphics, pages 138-149. Springer Berlin / Heidelberg, 2012. Copyright ©Springer-Verlag Berlin Heidelberg 2012".* This chapter is an adapted version of the published work.

and non-textual noise, respectively. When textual noise regions are fed to a character recognition engine, extra characters appear in the output of the OCR system along with the actual contents of the document. These extra characters in the OCR output result in inaccurate retrieval results, since the keywords given by the user might match some text from the textual noise instead of the actual document contents. Non-textual noise, on the other hand, makes further processing of document like text line extraction or dewarping a difficult task.

The problem of border noise is also well-known in the domain of scanned document analysis. Many approaches have been reported in literature to deal with border noise of scanned images. Most of the these approaches (e.g. [LTW96,AL04,FWL02]) focus only on removal of non-textual noise. Cinque et al. [CLLT02] proposed an algorithm for removing both textual and non-textual noise from grayscale images based on image statistics like horizontal/vertical difference vectors and row luminosities. The method presented in [SB09] detects border noise using different black/white filters. These approaches rely on certain assumption about scanned documents like an axis-aligned pattern of noise or presence of thick black non-textual noise regions. However, these assumptions do not hold for camera-captured documents since the document can be captured from any perspective (hence page border is not axis-aligned any more). Besides, the captured document is binarized using a local thresholding method like [SKB08a] (hence no thick black regions appear in the binarized image).

Instead of identifying and removing noisy components themselves, some methods focus on identifying the actual content area or the page frame of the documents [SvBKB08, SGG10]. The page frame of a scanned document is defined as the smallest region (rectangle or polygonal) that encloses all the foreground elements of the document image. The method presented in [SvBKB08] finds the page frame of structured documents (journal articles, books, magazines) by exploiting their text alignment property. The method by Fan et. al [FZT10] estimates page frame using a rectangular active contour. This method is not directly applicable to page frame detection of camera-captured documents due to the presence of perspective distortions. Stamatopoulos et al. [SGG10] proposed a method for splitting double-page scanned document images into two pages without noisy borders. Their method is based on vertical and horizontal white runs projections.

So far very few approaches are developed for marginal noise removal using page frame detection for camera-captured document images. Shafait et al. [SvBKB08] applied their page frame detection approach to camera-captured document images. When applied to camera-captured document images, the method focuses on finding the left and right page



(a) grayscale image (b) binarized image (c) segmented text parts (d) after noise cleanup

Figure 7.1: Preprocessing: (a) a sample grayscale camera-captured document image, (b) binarized document image, (c) segmented text parts of the binarized image, (d) cleaned image after noise removal.

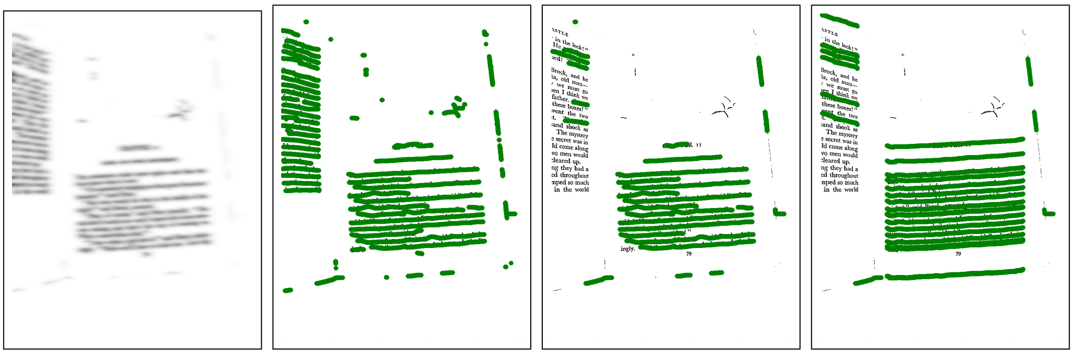
border lines only using a geometric matching method. The method gives good results for camera-captured document images, but does not remove border noise on the upper and lower sides of the document images. Stamatopoulos et al. [SGK07] proposed an algorithm for detecting borders of camera-captured document images based on projection profile. This method works well for a small degree of skew/curl in document images, but can not handle document images with a large degree of skew/curl, which is usually present in hand-held camera-captured documents.

In this chapter, a page frame detection method is presented for camera-captured document images. The method starts with preprocessing steps which includes binarization and text and non-text segmentation (Chapter 5). Then, text lines are detected by applying the ridge-based text line finding method that is described in Chapter 3. Finally, page frame is detected by using text lines and text and non-text information. The presented method can detect the upper and lower borders together with the left and right borders, and is robust to a large degree of skew/curl in camera-captured document images.

The rest of the Chapter is organized as follows. The proposed page frame detection method is described in Section 7.2. Experiments and results are discussed in Section 7.3. Section 7.4 presents conclusions.

7.2 Page Frame Detection Method

The proposed page frame detection method consists of three main steps: i) preprocessing, ii) text line detection, iii) page frame detection. Preprocessing (binarization and text and



(a) smoothed image (b) detected ridges (c) cleaned/filtered ridges (d) aligned ridges

Figure 7.2: Text Line Detection: (a) the smoothed image is generated using Gaussian filter bank smoothing, (b) ridges are detected from the smoothed image; most of the ridges represent text lines, (c) small ridges and ridges near corners are removed using heuristically applied rules, (d) ridges have been aligned (with respect to their staring and ending positions) by projecting neighboring ridges over each of them.

non-text segmentation) of camera-captured document images is discussed in Section 7.2.1. Text line detection method is described in Section 7.2.2. Page frame detection method using text line and text and non-text contents information is explained in Section 7.2.3.

7.2.1 Preprocessing

Here, the preprocessing approach mainly consists of binarization and text segmentation steps. An input grayscale camera-captured document image is first binarized using the adaptive thresholding technique mentioned in [ULB05], which is described as follows: "for each pixel, the background intensity $B(p)$ is defined as the 0.8-quantile in a window shaped surrounding; the pixel is then classified as background if its intensity is above a constant fraction of $B(p)$". An example grayscale document and its corresponding binarized document images are shown in Figure 7.1(a) and Figure 7.1(b), respectively.

The multiresolution morphology based text and non-text segmentation method, which is presented in Chapter 5, is applied here for text and non-text segmentation. The segmented text from the binarized document image (Figure 7.1(b)) is shown in Figure 7.1(c).

After text and non-text segmentation, a heuristic size based noise cleanup process is applied for removing comparatively large marginal noise and small salt-and-pepper noise elements as follows. A connected component is considered as a large noisy component if its height/width is greater than 10% of document height/width or greater than 7 standard deviation above mean height/width. Similarly, a connected component is removed as a small noisy component if its area is smaller than $\frac{1}{3}^{rd}$ of the mean area of all connected components in the document image. The document image in Figure 7.1(c) after noise cleanup is shown in Figure 7.1(d).

7.2.2 Text Line Detection

A generic text line extraction method is presented in Chapter 3. This method can be equally applied to a diverse collection of document images, including scanned, camera-captures, binarized, grayscale, typed-text, handwritten, etc. The text line finding method consists of two standard and easy to understand image processing algorithms: (i) matched filtering and (ii) ridge detection. Figure 7.2(a) shows the result of matched filtering for the document image in Figure 7.1(d). After smoothing, text lines are extracted by detecting ridges from the smoothed image. Most of the detected ridges, that are shown in Figure 7.2(c), are situated over text lines.

Some of the detected ridged are very small in size as compared to others, and some on them also lie over marginal textual noise. These types of ridges are not useful for further processing. A heuristic based method is applied for filtering/cleaning such types of ridges as follows. A ridge is considered as a small-size ridge if its length is smaller than $\frac{1}{10}^{th}$ of document width. Textual-noise is usually present in the left and right corners of the document. Therefore, a ridge is considered as a ridge over textual-noise if its starting/ending point exists very close (within ±25 pixels) to the left/right corner of document image and its length is smaller than $\frac{1}{5}^{th}$ of document width. After filtering small-size ridges and ridges over textual-noise, the starting and ending points of the remaining ridges can be used for approximating left and right borders, respectively. The remaining ridges are shown in Figure 7.2(c). Most of these remaining ridges are present over the actual content area of the page.

Another major problem in using these remaining ridges/text lines for left and right borders approximation is that, their starting and ending positions are not aligned with respect to each other. A ridges alignment method is presented in Chapter 8 for solving this issue, which is described here as follows. For each ridge, the neighboring top and bottom ridges are projected over it, and then are combined together to produce a new (aligned) ridge. The aligned ridges are shown in Figure 7.2(d). Some more results of ridges after alignment for document images in the DFKI-I dataset [SB07] are shown in Figure 7.3.

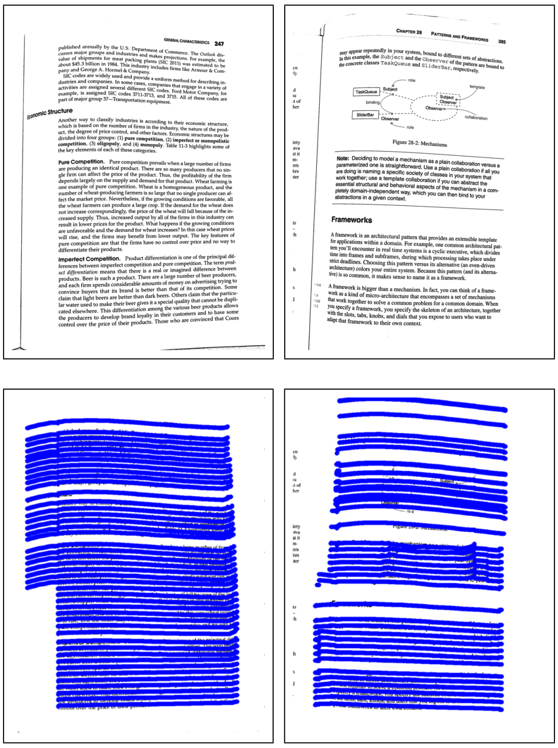
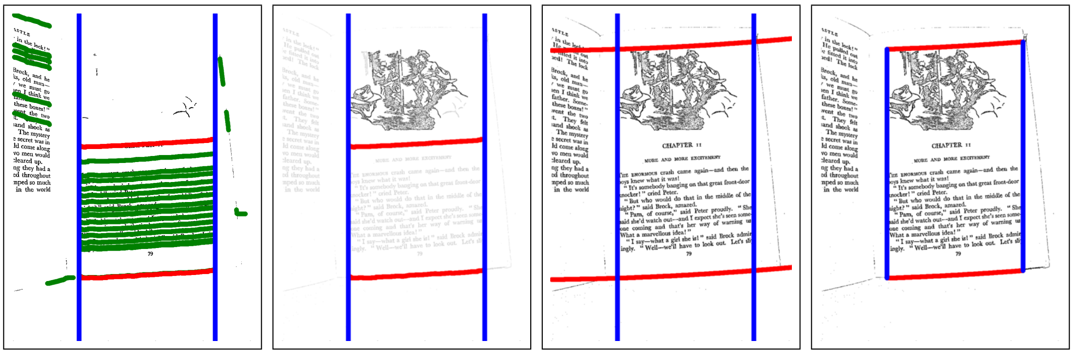


Figure 7.3: Sample results of aligned ridges for documents in the DFKI-I dataset.

7.2.3 Page Frame Detection

The left and right vertical borders are calculated by applying a straight-line approximation algorithm over the starting and ending points of the ridges, respectively. For this purpose, RANdom SAmple Consensus (RANSAC) method is chosen, which approximates slope and intercept parameters. The left and right borders are shown in Figure 7.4(a) in blue color. The initial estimation of upper and lower borders are done by selecting the top and bottom most ridges within the left and right borders, respectively. The upper and lower horizontal borders are also shown in Figure 7.4(b) in red color, where the lower border is correct, but upper border is incorrect with respect to the non-text content area of the page.

The initial page frame possesses only non-text elements which lie between text lines and misses others, as shown in Figure 7.4(b). The page frame is corrected by dragging the upper and/or lower borders according to the non-text elements which were segmented in the preprocessing step such that: i) if the top most pixel of non-text elements is above the top most pixel of the upper border, the upper border is dragged up to the top most pixel of non-text element, ii) similarly, if the bottom most pixel of non-text elements is



(a) left and right borders (blue color)

(b) initial upper and lower borders (red color)

(c) dragged and extended upper and lower borders

(d) page frame

Figure 7.4: Snapshots of the presented page frame detection method. (a) left and right borders (blue colors) are detected using starting and ending points of ridges (green color), (b) the top most and the bottom most ridges inside vertical borders are selected as upper and lower borders; non-text parts (black color), that were deleted in preprocessing, are pasted back into the document image, (c) the upper and lower borders are dragged up to the top most pixel and bottom most pixel of the non-text elements, and finally both of them are extended down to the page width, (d) page frame

below the bottom most pixel of the lower border, the lower border is dragged up to the bottom most pixel of non-text element. Finally, the upper and lower borders are extended across the document width using polynomial fitting. The upper and lower borders after dragging and extending are shown in Figure 7.4(c). The final page frame is shown in Figure 7.4(d). Some of example results of the presented page frame detection method on the DFKI-I dataset are also shown in Figure 7.5.

Generally, the starting points of some of the text lines in a document image coincide with the document's left border line and similarly the ending points of some of the text lines coincide with the right border line. The ridge alignment step helps in propagating this information to neighboring text lines. Therefore, left and right borders estimation using starting and ending points of text lines gives correct results. In a special case where a document image contains only short or centered text lines with non-text elements spanning throughout the page width, the left and right borders can not estimate the actual page contents area. In order to solve this issue, the left and/or right borders can also be dragged with respect to non-text elements, same as it is done in case of upper and/or lower border dragging.

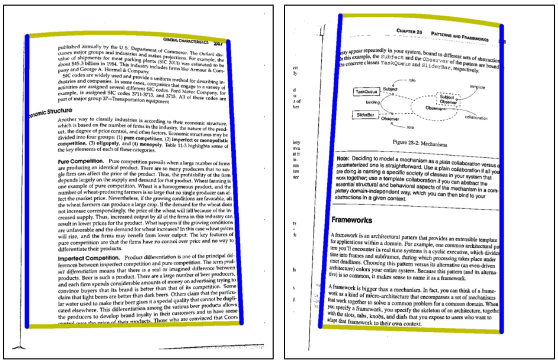


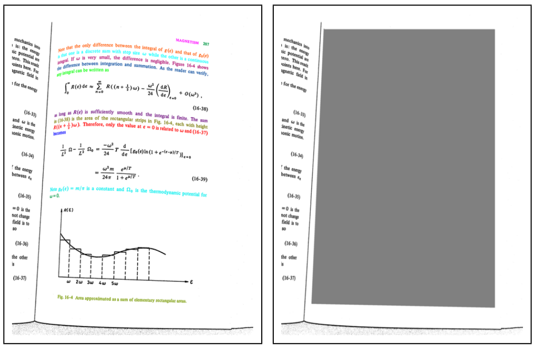
Figure 7.5: Sample results of the presented page frame detection method for the DFKI-I dataset.

7.3 Experiments and Results

The presented page frame detection method is compared with the state-of-the-art methods [SvBKB08, SGK07] by evaluating them on the publicly available DFKI-I (the CBDAR 2007 dewarping contest) dataset [SB07]. Two different experiments are conducted for performance evaluation: i) text line based evaluation, ii) pixel based evaluation.

The DFKI-I dataset contains 102 grayscale and binarized document images of pages from several technical books captured using an off-the-shelf hand-held digital camera in a normal office environment. Document images in this dataset consist of warped text lines with a high degree of curl, different directions of curl within an image, non-text (graphics, halftone, etc.) components, and a lot of textual and non-textual border noise. Together with ASCII-text ground-truth, this dataset also contains pixel based ground-truth for zones, text lines, formulas, tables and figures. For text line based performance evaluation method, text line based ground-truth images are generated from the original ground-truth images. A text lines based ground-truth image contains labeling only for text lines and all the other foreground objects, like formulas, tables and figures, are marked as noise with black color. For pixel based performance evaluation method, ground-truth images are generated by masking the actual page contents only. An example image and its corresponding text lines and pixel based ground-truth images are shown in Figure 7.6.

In document images, text lines are the main source of information from optical character recognition (OCR) point of view. For each text line in a text line based ground-truth image, the pixel-correspondence (P) is defined as the ratio of the number of overlapping pixels between the ground-truth image and the corresponding cleaned image and total number of pixels of a particular text line. Text line based performance evaluation metrics



(a) labeled text lines (b) masked page contents

Figure 7.6: Text line based and page contents based ground-truth images for an example image in the DFKI-I dataset [SB07].

using pixel-correspondence (P) were defined in [SB07]. In this work, the same metrics are used. These metrics are defined as follows: i) TI: totally-in text line ($P \geq 90\%$), ii) TO: totally-out text line ($P = 0\%$), and iii) PI: partially-in text line ($P < 90\%$). These metrics measure the percentage of totally-in, partially-in, and totally-out text lines within page contents with respect to the page frame. The text line based performance evaluation of the presented page frame detection method, Shafait et. al [SvBKB08], and Stamatopoulos et. al [SGK07] page frame detection methods are shown in Table 7.1. The results show that the presented method outperforms the other two methods.

Table 7.1: Text line based performance evaluation of the presented page frame detection method and the state-of-the-art methods [SvBKB08, SGK07] on the DFKI-I dataset. The results of the Shafait et. al method [SvBKB08] have been obtained from their paper. TI: totally-in text lines; TO: totally-out text lines; PI: partially-in text lines. The presented method achieved the larger number of TI and the smaller number of TO as compare to the state-of-the-art methods. (Note: total number of document images = 102; total number of text lines = 3097.)

Method	**TI**	**PI**	**TO**
Shafait et. al [SvBKB08]	95.6%	2.3%	2.1%
Stamatopoulos et. al [SGK07]	96.48%	0.71%	2.81%
the presented method	**98.10%**	**1.13%**	**0.78%**

Tex-line based performance metrics only measure the performance of a page frame detection method for text within actual page content area. They report nothing about the performance of a page frame detection method for marginal noise as well as non-text elements within page content area. Furthermore, text line based performance evaluation is not a useful measure for the case where the boundary of a complete document image, which contains both textual and non-textual noise, is marked as the page frame. In such a case, text line based performance evaluation reports 100% totally-in text lines with no partial-in or totally-out text lines. Therefore, text line based performance evaluation alone is not enough for comparing the performance of different page frame detection algorithms. In order to measure how well a page frame detection method works with respect to both marginal noise and actual page contents, a pixel based performance evaluation is used. The pixel based performance evaluation method, which is used here, measures the pixel-correspondence (P) for both actual page contents and marginal noise between a ground-truth image and the corresponding cleaned image. Pixel correspondence for page content is defined as the ratio of the number of overlapping pixels between the page contents of ground-truth image and the corresponding cleaned image and total number of page contents pixels in ground-truth image. Likewise, the pixel correspondence is defined for marginal noise. The pixel based performance evaluation results of the presented method and Stamatopoulos et. al [SGK07] method are shown in Table 7.2. It shows that both methods give good performance for actual page contents, but the presented method performs better for marginal noise cleanup.

Table 7.2: Pixel based performance evaluation of the presented page frame detection method and the state-of-the-art method [SGK07] on the DFKI-I dataset. 'Page Contents' represents the percentage of page contents inside detected page frame. 'Marginal Noise' represent the percentage of noise outside detected page frame. The presented method achieved better results than the Stamatopoulos et. al [SGK07] method. (Note: total number of page contents pixels = 48188808 (88.52%); total number marginal noise pixels = 6247054 (11.48%)).

method	**Page Contents**	**Marginal Noise**
Stamatopoulos et. al [SGK07]	99.11%	36.04%
the presented method	98.96%	74.81%

7.4 Conclusion

In this chapter, a page frame detection method is presented for warped camera-captured document images. The method uses text lines and non-text contents information for detecting page frame (left, right, upper, and lower borders). The ridge-based text line finding method (Chapter 3) and the multiresolution based text/non-text segmentation method (Chapter 5) are applied here for detecting text lines and non-text elements, respectively. For the performance evaluation of the presented method and its comparison with state-of-the-art methods, two different methodologies, text line based and pixel based, have been used. For both performance evaluation methodologies, the presented method has achieved better results than Shafait et. al [SvBKB08] and Stamatopoulos et. al [SGK07] page frame detection methods, as shown in Table 7.1 and Table 7.2.

Chapter 8

Monocular Dewarping[1]

Summary: *traditional OCR systems are designed for planar (dewarped) document images and the accuracy is reduced when applied on warped document images. Therefore, developing new OCR techniques for warped images or developing dewarping techniques are the possible solutions for improving OCR accuracy of camera-captured document images. Among different types of dewarping techniques, curled text lines information based dewarping techniques are the most popular ones, but are sensitive to high degree of curl and variable line spacing. In this chapter, a novel monocular dewarping approach is presented, which is based on curled text lines information. The presented dewarping approach is less sensitive to different direction of curl and variable line spacing. Experimental results show that OCR error rate, from warped to dewarped documents, has been reduced from* 5.15% *to* 1.92% *on the DFKI-I (the CBDAR 2007 document image dewarping contest) dataset. The performance of the presented method in also compared with other state-of-the-art monocular dewarping methods.*

8.1 Introduction

For document analysis and recognition, flat-bed scanners are traditionally and widely used for document capturing. They produce planar images with high resolution. From decades, many approaches have been proposed for planar document image segmentation [SKB08b] and Optical Character Recognition (OCR) [MSY92]. Nowadays good quality cameras are available at low cost, that offer fast, easy, flexible and non-contact

[1] This work was published in Bukhari et al. [BSB09a] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. Dewarping of document images using coupled-snakes. In Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pages 34-41, Barcelona, Spain, 2009"*. This chapter is an adapted version of the published work.

imaging. These advantages make cameras a potential substitute of scanner for document capturing and at the same time open doors for many new applications, like mobile OCR, digitizing thick books, digitizing fragile historical documents, finding text in scene images, etc. However, camera-captured document images suffer from various types of distortions, like non-planar (warped) shape, uneven light shading, motion blur, perspective distortion, under- and over-exposure. Therefore, current OCR systems, which are designed for planar document images, are not capable to deal with these distortions and give poor performance when applied directly to warped camera-captured document images. There could be two possible solutions for improving the OCR performance of warped document images: (i) design new camera-based document analysis techniques or (ii) design dewarping techniques for flattening document images such that current OCR systems can be directly applied to them.

So far, much attention has not been given to developing special document image analysis techniques for camera-captured document images. But over last decade, different approaches have been proposed for document image dewarping [LDL05, SB07]. These approaches can be divided into two main categories based on the document capturing methodology: (i) approaches in which specialized hardware arrangement, like stereo-camera, is required for 3D shape reconstruction of warped document [CDL03, BS04, TZZX06] and (ii) approaches in which dewarping method is designed for images which are captured using single hand-held camera in an uncontrolled environment, also referred to as monocular dewarping approaches [LDD05,LT06b,MM07,CM03,ZT03,ULB05, LCK05, LT05, LT06a, GPN07, GN07, SGPP08]. Hand-held camera-based dewarping approaches can be further classified into two groups: (i) approaches based on document geometry [LDD05, LT06b, MM07, CM03] and (ii) approaches based on curled text lines information [ZT03, ULB05, LCK05, LT05, LT06a, GPN07, GN07, SGPP08].

The dewarping approach presented in this chapter falls under the category of curled text line based monocular dewarping method. Literature review on curled text line based dewarping approaches is given below.

Zhang and Tan [ZT03] proposed dewarping method for scanned document from thick bound volumes. They consider that the major portion of the image is straight. They estimate neighboring curved portions of each straight text line by clustering connected components and then move these connected components parallel to their straight line portion.

Ulges et al. [ULB05] proposed dewarping technique based on priori layout information and local text line approximation using RAST [Bre02b]. After local text lines finding, they estimate quadrilateral cell for each letter and then map to a rectangle of appropriate size and position in the dewarped image.

Lu et al. [LCK05] introduced a rectification technique for restoring documents with perspective distortions. Their algorithm is based on tip-points and vertical stroke boundaries estimation using morphological operations. They use top points and bottom points for estimating top and bottom text lines, respectively. They estimate source quadrilaterals using vertical stroke boundaries and text lines pairs and then construct rectification homography using each pair of source and target quadrilaterals.

Lu and Tan [LT05, LT06a] proposed dewarping algorithms which are the extension of work presented in [LCK05]. These methods can remove skew, perspective and geometric distortions.

Gatos et al. [GPN07] introduced dewarping techniques using text lines information. They perform horizontal smoothing [WWC06] to combine characters into words and then find lines by grouping neighboring words. They rotate each word of a line individually based on its slope and then align all words of a line with respect to the left most word.

Gatos and Ntirogiannis [GN07] proposed dewarping approach based on the estimation of words and text line by using the modified "box hand" [ZT01,SPC97] approach. Similar to [GPN07], they rotate each word of a line and then align all words of a line with respect to the left most word.

Dewarping technique by Fu et al. [FWL+07] starts by estimating sub lines using characters combination method [HLW03].They cluster mid points of sub lines based on a proximity criteria, which results in text lines. From these text lines, they estimate left and right borders and top and bottom curves. They stretch cylinder surface area into planar surface area based on the model presented in [CDL03].

Stamatopoulos et al. [SGPP08] introduced a two-step dewarping algorithm. They estimate text lines using method introduced in [GPN07]. In coarse dewarping step, they estimate left and right borders and top and bottom curves using text lines information and then transform curve area into 2D rectangular area. In fine dewarping step, they perform dewarping algorithm, presented in [GPN07], over coarse dewarped image.

In this chapter, a monocular dewarping approach based on curled text line information is presented. The method starts by extracting text line using ridge-based text line detection method (Chapter 3). Afterwards, for each detected text line, the x-line and baseline pair is estimated by adapting coupled-snakes model (Chapter 2). For each curled text line in a warped image, the starting position of its corresponding straight text line for a dewarped image is calculated using neighboring curled text lines in the warped image. Then, geometric distortion or page curl is removed by mapping characters over each curled x-line and baseline pair to its corresponding straight x-line and baseline pair. Finally, perspective distortion is removed by using four point homography algorithm.

The rest of the chapter is organized as follows: Section 8.2 describes the technical and implementation details of the presented dewarping algorithm. Section 8.3 comprises the performance evaluation and experimental results. Section 8.4 discusses the results and conclusion.

8.2 The Dewarping Algorithm

The presented dewarping algorithm comprises of three steps: (1) curled text line information extraction from binarized document images, (2) geometric distortion handling using curled text line information and (3) perspective distortion handling using four point homography algorithm. All of these steps are described below.

8.2.1 Curled Text line Information Extraction

For extracting curled text line information from camera-captured document images, the ridge-based text line detection method is used that is presented in Chapter 3. Detected ridges over the smoothed image of Figure 8.1(b) are shown in Figure 8.1(c). It is clearly visible in the Figure 8.1(c) that each ridge covers the complete central line structure of a text line, which results in text lines detection.

After detecting text lines, the task is to determine the x-line and baseline pair information for each detected text line using the coupled-snakes model (Chapter 2). For this purpose, the gradient of binary image is computed by using Sobel filter. Then, gradient image is divided into two images: one contains positive magnitudes (top-image) and another one contains absolute values of negative magnitudes (bottom-image). Top-image is dominated by the top parts of curled text lines and similarly bottom-image is dominated by the bottom parts of curled text lines. Then, gradient vector flow (GVF) [XP98] of both images are calculated. Active contour (snake) [KWT88] is adapted for finding x-line and baseline pairs information that is presented in Chapter 2. The process is briefly describe here. First of all, the duplicated ridges are used as initial open-curve snakes pairs for curled text lines. Each pair is deformed using GVFs of top- and bottom-image in the weighted-coupled snakes fashion, describe as follows. For each pair, one snake is deformed with respect to the vertical components of GVF of top-image and another one with respect to the vertical components of GVF of bottom-image. A large percentage of GVF of bottom gradient image and a comparatively small percentage of GVF of top gradient image are used during coupling, because of the assumption that more characters lie on baseline than on x-line. After deformation, the distances between each pair of snakes are adjusted to make them equal to average distance. The same deformation process is repeated for few more iterations, where the length of snakes remain fixed during each iterations. Figures 8.1(d) and 8.1(e) show the properly estimated pairs of x-line and baseline pairs for curled text lines.

8.2.2 Handling of Geometric Distortions

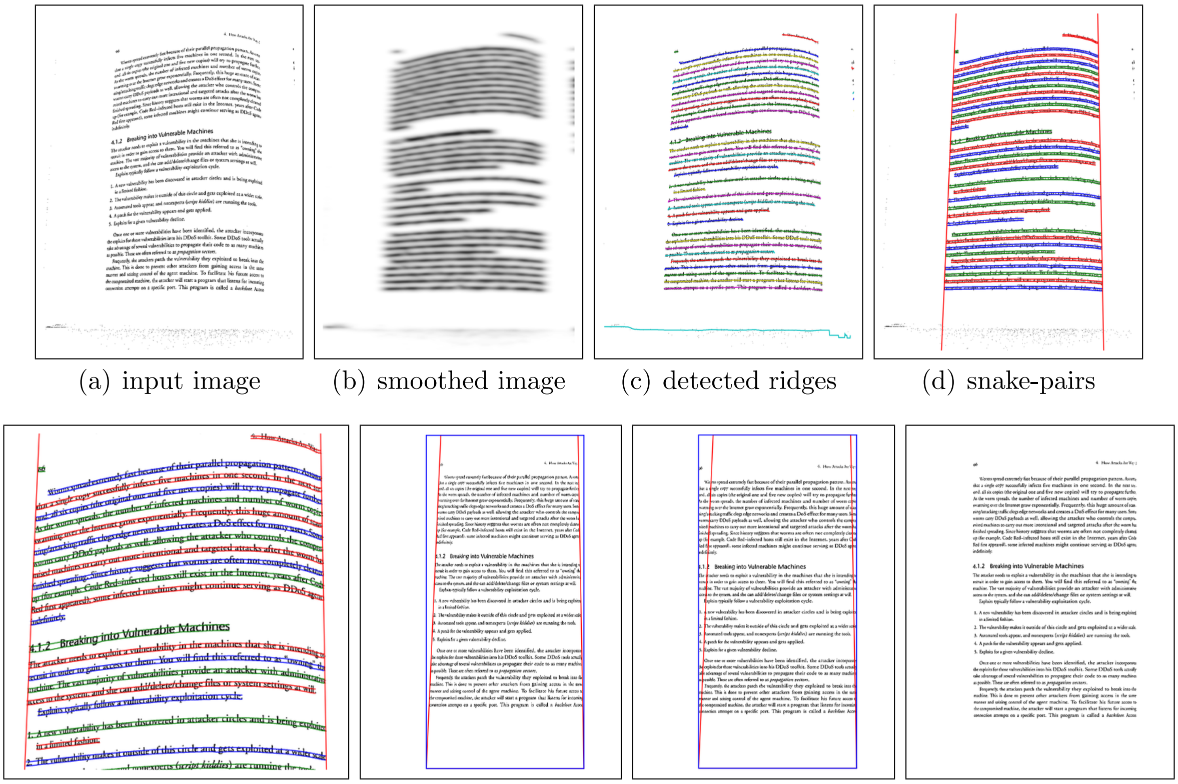
One easy and efficient way of handling geometric (curl) distortion by using curled text line pairs is to estimate corresponding straight text line pairs and then map all pixel values from curled text line pairs to corresponding straight text line pairs. For each text line, the starting and ending x-coordinate values of straight text line pair is set similar to curled text line pair. Now the more critical task is the approximation of y-coordinate values for straight text line pairs. One way of calculating y-coordinates for each straight text line pair is to find the average y-coordinate values of top and bottom curled text lines within a pair. However, in document image some text lines are small and some are large and large text lines contain more information than smaller ones. Due to this fact, estimated y-coordinates of small text lines are not accurate and results in overlapping of text lines in dewarped image. Therefore, neighboring text lines information are used here for estimating y-coordinate values for straight-line pairs. For each text line, top and bottom curled lines from neighboring text lines are aligned or projected over the top curled line and bottom curled line of targeted text line, respectively. Then, all these top curled lines are combined together and bottom curled lines are combined together, which results in an approximated curled text line pair for targeted text line. Approximated curled text line pairs contain more curled information than actual curled text line pairs, especially for small text lines. These approximated pairs are used only for calculating y-coordinates for straight text line pairs. For each text line, the top and bottom y-coordinates for corresponding straight text line pair are calculated from the approximated curled text line pair through averaging y-coordinates of its top curled line and bottom curled line. After estimating straight text line pairs, all pixels over curled-line pairs are mapped to the corresponding straight-line pairs. Resulting dewarped image is shown in Figure 8.1(f), in which text lines are straight as compared to curled text lines of input image (Figure 8.1(a)).

8.2.3 Handling of Perspective Distortions

After handling geometric (curl) distortion, the next step is to remove perspective distortion in the image, as shown in Figure 8.1(f). For handling perspective distortion, the four point homography algorithm [HZ04] is used here, in which homography matrix is calculated from source and target quadrilaterals. Here, quadrilateral with perspective distortion is used as source and quadrilateral without perspective distortion is used as target. The process of removing perspective distortion starts by finding left and right vertical borders of warped image. Left and right border are calculated by applying RANSAC on staring and ending points of curled text line pairs. Resulting borders are shown in Figure 8.1(d). For perspective distortion free rectangle, left border perpendicular to page width is calculated by finding minimum x-coordinate value from left border shown in Figure 8.1(d). Similarly right border perpendicular to page width, is calculated by finding maximum x-coordinate value of right border shown in Figure 8.1(d). Resulting source and target quadrilaterals are shown in Figures 8.1(f) and 8.1(g), in blue and red colors, respectively. Rectifying homography matrix is calculated by using source and target quadrilateral. Then, x-and y-coordinates of source quadrilateral are transformed into target quadrilateral using homography matrix and bilinear-interpolation is applied for calculating intensity values for dewarped image. Final dewarped image is represented in Figures 8.1(g) and 8.1(h). One can compare the good quality of final dewarped image (Figure 8.1(h)) with input warped image (Figure 8.1(a)). The presented method also gives good dewarping results for two column document images as shown in Figure 8.2.

8.3 Experiments and Results

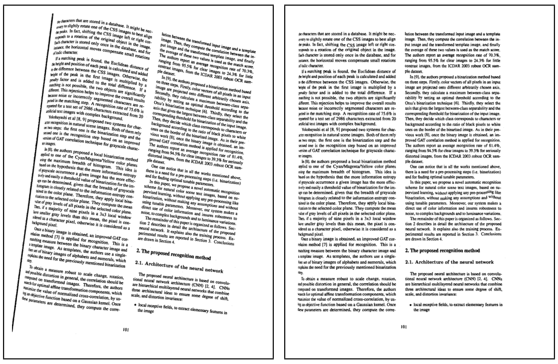
To demonstrate the performance of the presented algorithm on real world documents, the DFKI-I (the CBDAR 2007 document image dewarping contest) dataset [SB07] is used. The dataset consists of 102 documents, captured with hand-held camera. This dataset is freely available with ASCII text ground-truth. Three methods participated in this contest: SEG [GPN07], SKEL [MM07] and CTM (un-cleaned results) and CTM2 (cleaned up results) [FWL$^+$07]. The presented dewarping method is referred to as "Ridges-Snakes". The results of all methods on some example documents from the dataset are shown in Figure 8.3.



(a) input image (b) smoothed image (c) detected ridges (d) snake-pairs

(e) closeup portion (f) rectification of geometric distortion (g) rectification of perspective distortion (h) dewarped image

Figure 8.1: Different stages of the presented dewarping algorithm. i) a sample camera-captured warped document image, ii) the smoothed image is generated by using mutli-oriented anisotropic Gaussian smoothing, iii) ridges have been detected from smoothed image, iv) snake-pairs have been estimated using ridges based coupled-snakes model. Left and right vertical borders have been calculated using RANSAC on starting and ending points of each pair, v) closeup portion of 8.1(d), vi) geometric distortion has been handled by straightening curled text lines snake-pairs, shown in Figure 8.1(d). Red borders show that document image still have perspective distortion. Blue rectangle has been calculated using red borders, vii) perspective distortion has been handled by using four point homography algorithm, viii) final dewarped image.



(a) (b)

Figure 8.2: A sample two column document image and its dewarped results. a) Two column document image. b) Dewarped document image.

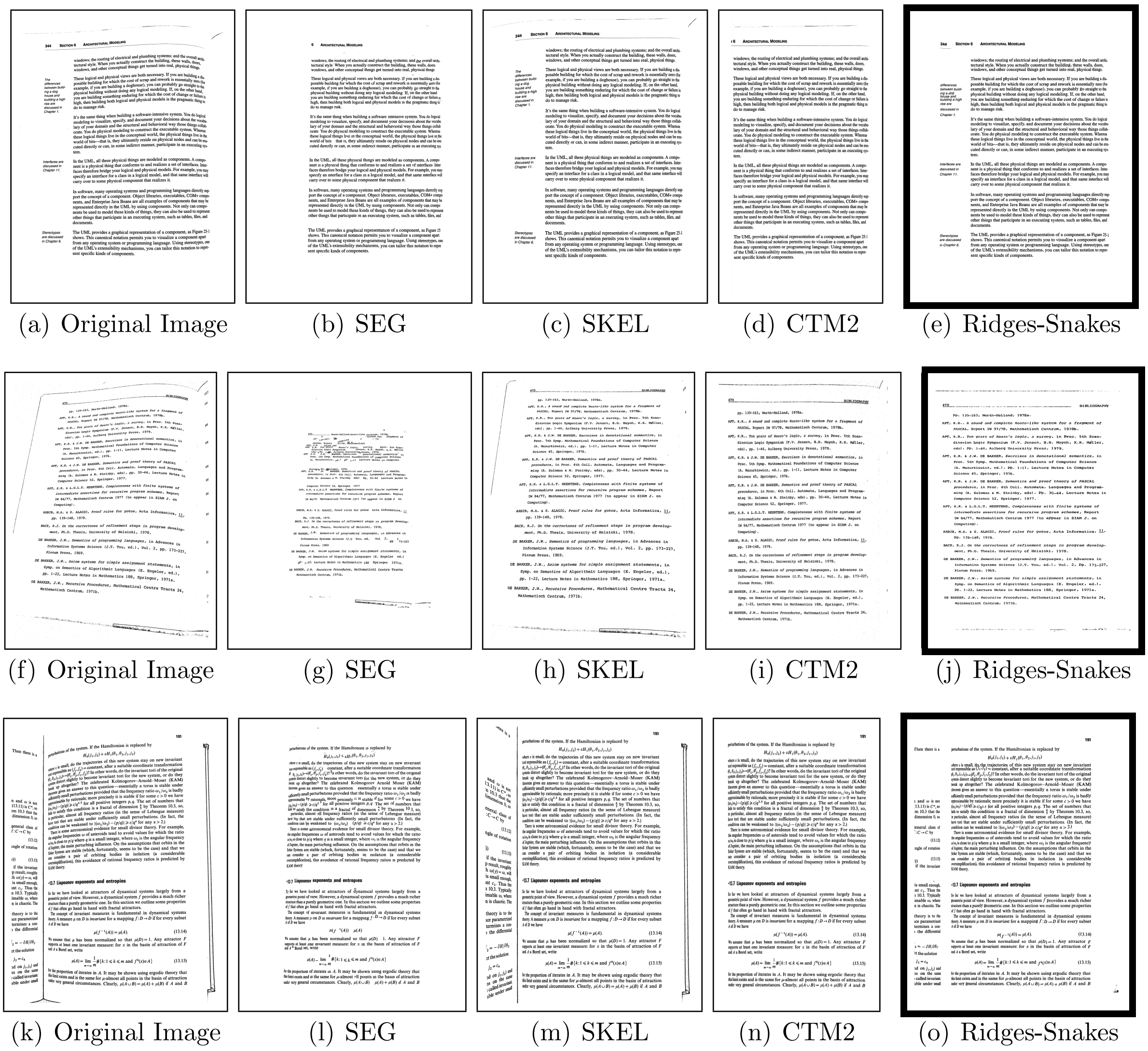
The dewarped documents of all methods are processed through a commercial OCR system **ABBYY Fine Reader 9.0**. After obtaining text from the OCR software, the block edit distance[2] with the ASCII ground-truth has been used as the error measure. Table 8.1 shows the comparative results of all methods with respect to error rate by mean edit distance, error rate by median edit distance and the number of documents for each algorithm on which it has the lowest error rate by edit distance (in case of tie, all algorithms having the lowest error rate by edit distance are scored for that document).

Together with comparative performance evaluation, mean edit distance error rates, before and after dewarping, are also compared. Before dewarping, error rate is 5.153%. After dewarping using the presented method, error rate is 1.917%, as mentioned in Table 8.1. This demonstrates that, after dewarping average edit distance error rate is reduced by 3.24%.

8.4 Summary

In this chapter, a new monocular dewarping approach is presented using curled text lines information for warped, camera-captured document images. Unlike some other dewarping approaches like [FWL+07], the presented dewarping method does not use any type of post-processing step for cleaning resulting dewarped documents. After applying the reported dewarping method on the DFKI-I (the CBDAR 2007 dewarping contest) dataset [SB07], OCR error rate is reduced by 3.24%. Additionally, a fair comparison of the presented dewarping method with other three participants of the dewarping contest [SB07] has been

[2] http://sites.google.com/site/ocropus/release-notes



(a) Original Image (b) SEG (c) SKEL (d) CTM2 (e) Ridges-Snakes

(f) Original Image (g) SEG (h) SKEL (i) CTM2 (j) Ridges-Snakes

(k) Original Image (l) SEG (m) SKEL (n) CTM2 (o) Ridges-Snakes

Figure 8.3: Comparative results of different dewarping methods (SEG [GPN07], SKEL [MM07], CTM [FWL+07] and Ridges-Snakes: For image 8.3(a) SEG and CTM2 methods removed text-note, among all SKEL and Ridges-Snakes have done proper dewarping. For image 8.3(f) SEG method failed to remove geometric and perspective distortions, SKEL method removed only geometric distortion; among all CTM2 and Ridges-Snakes have done proper dewarping by removing both geometric and perspective distortions.

Table 8.1: Comparative OCR error rate (block edit distance) results on the DFKI-I dataset based on ABBYY Fine Reader 9.0. The presented approach achieved the lowest error rate in the majority of document images as compared to the state-of-the-art methods.

Algorithm	**Error rate**	**Error rate**	**Number of documents**
	by mean edit distance	**by median edit distance**	**with the lowest error rate**
SEG [GPN07]	4.088	2.122	02
SKEL [MM07]	2.162	0.972	29
CTM [FWL+07]	2.113	0.893	30
CTM2 [FWL+07]	**1.758**	0.827	38
Ridges-Snakes	1.917	**0.733**	**41**

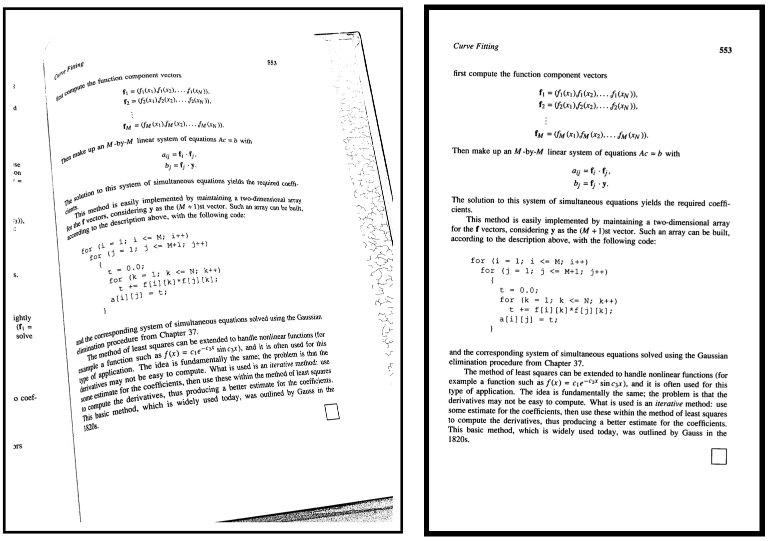
done. The winning method [FWL+07] of that contest had submitted two different results: (i) dewarped results without post-processing (CTM) and (ii) dewarped results with post-processing for removing graphics and images (CTM2). According to the statistics presented in the Table 8.1, the performance of the presented method is nearly similar to the cleaned dewarped results of winning method of the dewarping contest, i.e. CTM2, but it is better than the un-cleaned dewarped results of winning method of the dewarping contest, i.e. CTM. Furthermore, as shown in Figure 8.3, dewarped results of the presented method look more planar than other three methods and the presented dewarping method performs better than the three other methods in the presence of margin-notes, two-column documents and high degrees of perspective distortions.

Chapter 9

Image based Performance Evaluation of Dewarping Methods[1]

Summary: *dewarping of camera-captured document images is one the important preprocessing steps before feeding them to a document analysis system. Over the last few years, many approaches have been proposed for document image dewarping. Usually optical character recognition (OCR) based and/or feature based approaches are used for the evaluation of dewarping algorithms. OCR based evaluation is a good measure for the performance of a dewarping method on text regions, but it does not measure how well the dewarping algorithm works on the non-text regions like mathematical equations, graphics, or tables. Feature based evaluation methods, on the other hand, do not have this problem, however, they have following limitations: i) a lot of manual assistance is required for ground-truth generation, and ii) evaluation metrics are not sufficient to get meaningful information about dewarping quality. In this chapter, an image based methodology is presented for the performance evaluation of dewarping methods using Scale Invariant Feature Transform (SIFT) features. For ground-truths, the presented method only requires scanned images of pages which have been captured by a camera. This chapter introduces a vectorial performance evaluation score which gives comprehensive information for determining the performance of different dewarping methods. The presented performance evaluation methodology has been tested on the participating methods of the CBDAR 2007*

[1] This work was published in Bukhari et al. [BSB12a] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. An Image Based Performance Evaluation Method for Page Dewarping Algorithms using SIFT Features. In Masakazu Iwamura and Faisal Shafait, editors, 4th International Workshop, CBDAR 2011 Beijing, China, September 22, 2011 Revised Selected Papers, volume 7139 of Lecture Notes in Computer Science, Image Processing, Computer Vision, Pattern Recognition, and Graphics, pages 138-149. Springer Berlin / Heidelberg, 2012. Copyright ©Springer-Verlag Berlin Heidelberg 2012".* This chapter is an adapted version of the published work.



Curve Fitting **553**

first compute the function component vectors

$$\mathbf{f}_1 = (f_1(x_1), f_1(x_2), \ldots f_1(x_N)),$$
$$\mathbf{f}_2 = (f_2(x_1), f_2(x_2), \ldots f_2(x_N)),$$
$$\vdots$$
$$\mathbf{f}_M = (f_M(x_1), f_M(x_2), \ldots f_M(x_N)).$$

Then make up an M-by-M linear system of equations $Ac = b$ with

$$a_{ij} = \mathbf{f}_i \cdot \mathbf{f}_j,$$
$$b_j = \mathbf{f}_j \cdot \mathbf{y}.$$

The solution to this system of simultaneous equations yields the required coefficients.

This method is easily implemented by maintaining a two-dimensional array for the **f** vectors, considering **y** as the $(M+1)$st vector. Such an array can be built, according to the description above, with the following code:

```
for (i = 1; i <= M; i++)
  for (j = 1; j <= M+1; j++)
    {
      t = 0.0;
      for (k = 1; k <= N; k++)
        t += f[i][k]*f[j][k];
      a[i][j] = t;
    }
```

and the corresponding system of simultaneous equations solved using the Gaussian elimination procedure from Chapter 37.

The method of least squares can be extended to handle nonlinear functions (for example a function such as $f(x) = c_1 e^{-c_2 x} \sin c_3 x$), and it is often used for this type of application. The idea is fundamentally the same; the problem is that the derivatives may not be easy to compute. What is used is an *iterative* method: use some estimate for the coefficients, then use these within the method of least squares to compute the derivatives, thus producing a better estimate for the coefficients. This basic method, which is widely used today, was outlined by Gauss in the 1820s.

□

(a) camera-captured document

(b) scanned document (ground-truth dewarped image)

Figure 9.1: A sample camera-captured document and its corresponding scanned image from the DFKI-I dataset. The scanned images in the DFKI-I dataset are used here as ground-truth dewarped images.

document image dewarping contest, which illustrates its correctness.

9.1 Introduction

The goal of page dewarping is to flatten a camera-captured document such that it becomes readable by current OCR systems. Page dewarping has triggered a lot of interest in the scientific community over the last few years and many approaches have been proposed. These dewarping approaches can be broadly divide into two main categories: i) 3-D document shape reconstruction [CDL03, BS04, TZZX06] and ii) 2-D image processing (monocular dewarping) [CM03, ZT03, ULB05, LDD05, GAS07, BSB09a].

Despite a large number of dewarping techniques, performance evaluation of page dewarping methods is still an unsolved problem. Most of the time it has been done on the basis of visual quality of dewarped images [GAS07,BT06], but it is a subjective evaluation and gives no quantitative measure. In order to objectively compare dewarping methods, OCR based [SB07, BSB09a] and feature based [SGP09] performance evaluation methods have been proposed. OCR based performance evaluation is an indirect method which can only measure the performance of a dewarping method on text regions. Nowadays commercial OCR software can handle degradations in documents to some extend, therefore, OCR based evaluation can not measure how well text elements have been dewarped with respect to their shapes. On the other hand, feature based performance evaluation do not have these problems and can measure the performance of a dewarping method for both text and non-text regions. However, existing feature based performance evaluation methods have following limitations: i) a cumbersome manual marking is required for generating ground-truth data, and ii) a single performance evaluation metric is used which may not be sufficient to compare the performance of different dewarping methods.

In this chapter, an image based performance evaluation methodology using SIFT features [Low04] is presented for comparing dewarping methods. The presented technique overcomes the limitations of the existing feature based performance evaluation methods. The scanned images of pages, that were captured by camera, are used as ground-truth dewarped images. In this way, no manual efforts are required for generating ground-truth data for publicly available datasets that contain corresponding scanned documents as well (like the DFKI-I [SB07]), or less manual efforts are required for preparing ground-truths for new datasets. For performance evaluation, instead of a single performance evaluation metric, a vectorial score is presented, which is particularly useful in analyzing the behavior of different page dewarping algorithms. On the basis of the SIFT features matching between a dewarped image and its corresponding ground-truth scanned/dewarped image, the matching percentage and the matching error are estimated.

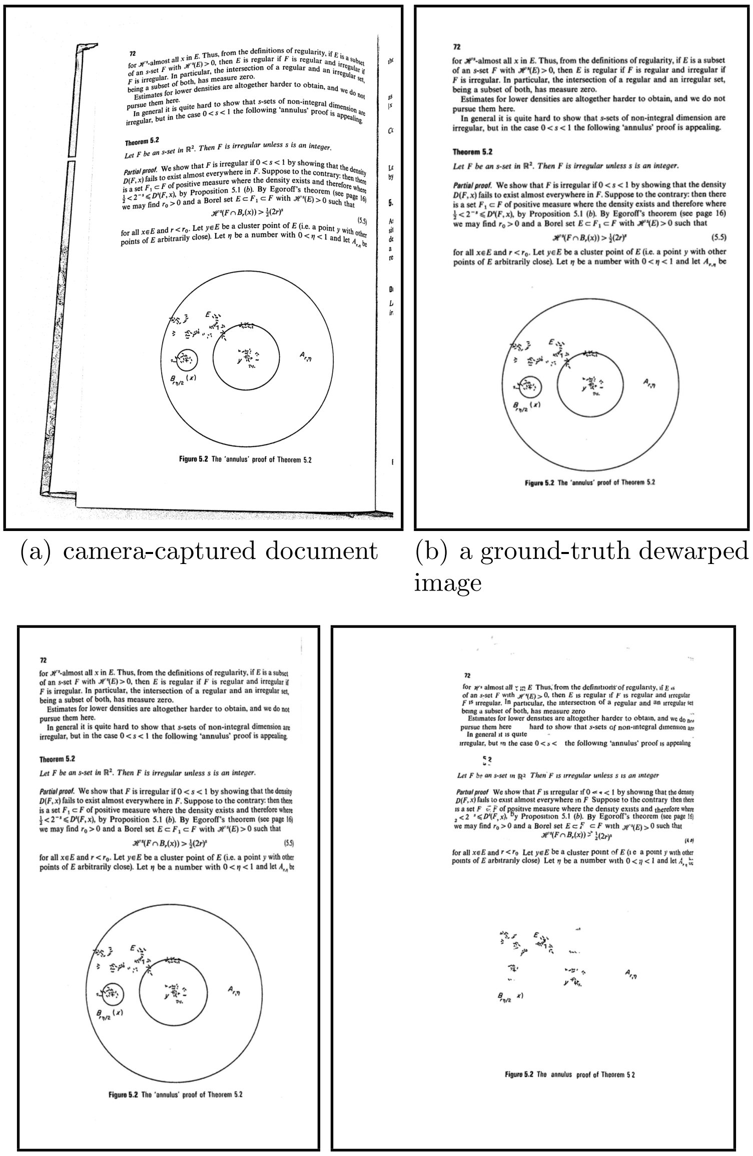
The rest of the chapter is organized as follows. The proposed image based performance evaluation is described in Sections 9.2. Experiments and results are discussed in Section 9.3. Section 9.4 presents conclusions.

9.2 Image based Performance Evaluation

The proposed performance evaluation metrics are described here in detail along with the requirement of ground-truth dewarped images. This section is organized as follows. Section 9.2.1 discusses about the ground-truth dewarped images. The performance evaluation metrics using the SIFT based matches are explained in Section 9.2.2.

9.2.1 Ground-Truth Dewarped Images

The presented image based performance evaluation method requires ground-truth dewarped images. So far, the DFKI-I [SB07] and the IUPR [BSB11d] are the only two



72

for $\mathcal{H}^s$-almost all x in E. Thus, from the definitions of regularity, if E is a subset of an s-set F with $\mathcal{H}^s(E) > 0$, then E is regular if F is regular and irregular if F is irregular. In particular, the intersection of a regular and an irregular set, being a subset of both, has measure zero.

Estimates for lower densities are altogether harder to obtain, and we do not pursue them here.

In general it is quite hard to show that s-sets of non-integral dimension are irregular, but in the case $0 < s < 1$ the following 'annulus' proof is appealing.

Theorem 5.2

Let F be an s-set in $\mathbb{R}^2$. Then F is irregular unless s is an integer.

Partial proof. We show that F is irregular if $0 < s < 1$ by showing that the density $D(F, x)$ fails to exist almost everywhere in F. Suppose to the contrary: then there is a set $F_1 \subset F$ of positive measure where the density exists and therefore where $\frac{1}{2} < 2^{-s} \leqslant D^s(F, x)$, by Proposition 5.1 (*b*). By Egoroff's theorem (see page 16) we may find $r_0 > 0$ and a Borel set $E \subset F_1 \subset F$ with $\mathcal{H}^s(E) > 0$ such that

$$\mathcal{H}^s(F \cap B_r(x)) > \tfrac{1}{2}(2r)^s \tag{5.5}$$

for all $x \in E$ and $r < r_0$. Let $y \in E$ be a cluster point of E (i.e. a point y with other points of E arbitrarily close). Let η be a number with $0 < \eta < 1$ and let $A_{r,\eta}$ be

Figure 5.2 The 'annulus' proof of Theorem 5.2

(a) camera-captured document

(b) a ground-truth dewarped image

(c) a good dewarped output

(d) a relatively bad dewarped output

Figure 9.2: A sample camera-captured document image and its corresponding ground-truth dewarped image and a good and a bad dewarped images.

publicly available datasets of camera-captured document images. These datasets were prepared to compare different layout analysis approaches for camera-based document image analysis. The following types of ground-truth were provided with these dataset: i) ground-truth ASCII text in plain text format, ii) ground-truth page segments (text lines and zones and their types) in color coded form, iii) scanned images of pages which have been captured by a camera.

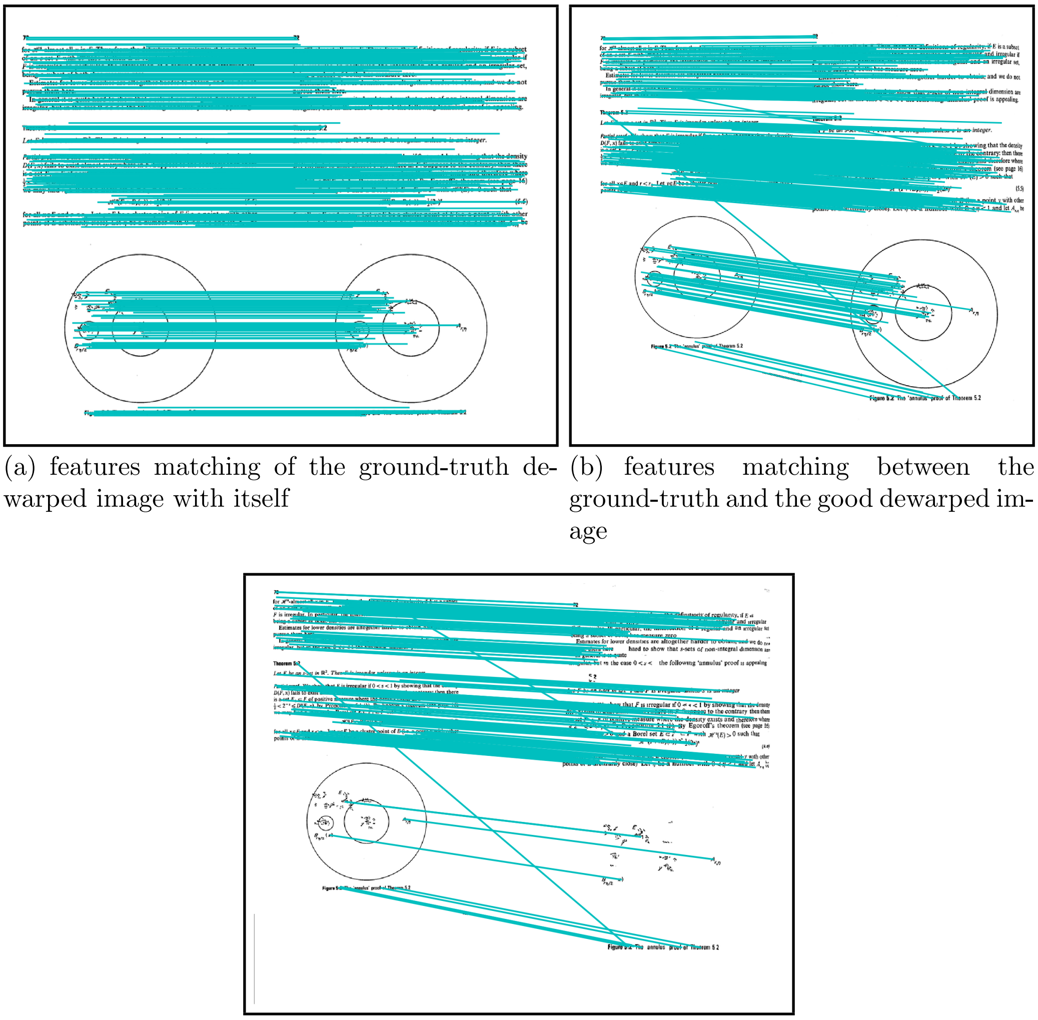
The DFKI-I dataset was used in a Document Image Dewarping Contest that was held at the CBDAR 2007 workshop [SB07]. In that contest, several different dewarping methods participated, and they were compared through OCR based performance evaluation methodology using ASCII text ground-truth. The DFKI-I dataset is also used here for comparing these participated methods through the presented image based performance evaluation methodology.

A sample camera-captured document and its corresponding scanned image from the DFKI-I dataset are shown in Figure 9.1. The scanned document images in this dataset, as shown in Figure 9.1(b), are flat and straight. Therefore, they can be used as ground-truth dewarped images. For the purpose of performance evaluation, scanning of pages together with capturing them through camera requires very less manual effort as compared to marking images manually [SGP09] or to generate ASCII text ground-truth [SB07].

9.2.2 Performance Evaluation Methodology

To compare the quality of a dewarped document against a ground-truth dewarped document, image based features are calculated using SIFT [Low04]. For an image, the SIFT estimates key features and returns their corresponding locations and descriptors. Matching between the features of two different images is done by calculating cosine inverse of the dot product of their normalized descriptors. The bad matches are removed by applying a thresholding criteria such as, a match is considered bad if the angle ratio between first and second nearest neighbors is greater than a predefined threshold. Here, this threshold is set equal to 0.6. It has also been noticed that there could be more wrong SIFT matches between two similar document images at high image resolutions than at low image resolutions. Therefore, document images are downscaled by the factor of 4 before the SIFT features based comparison.

A sample camera-captured, warped document image and its corresponding ground-truth dewarped image are shown in Figure 9.2(a) and Figure 9.2(b), respectively. For the camera-captured image (Figure 9.2(a)), two different, a good one and a bad one,



(a) features matching of the ground-truth dewarped image with itself

(b) features matching between the ground-truth and the good dewarped image

(c) features matching between the ground-truth and the bad dewarped image

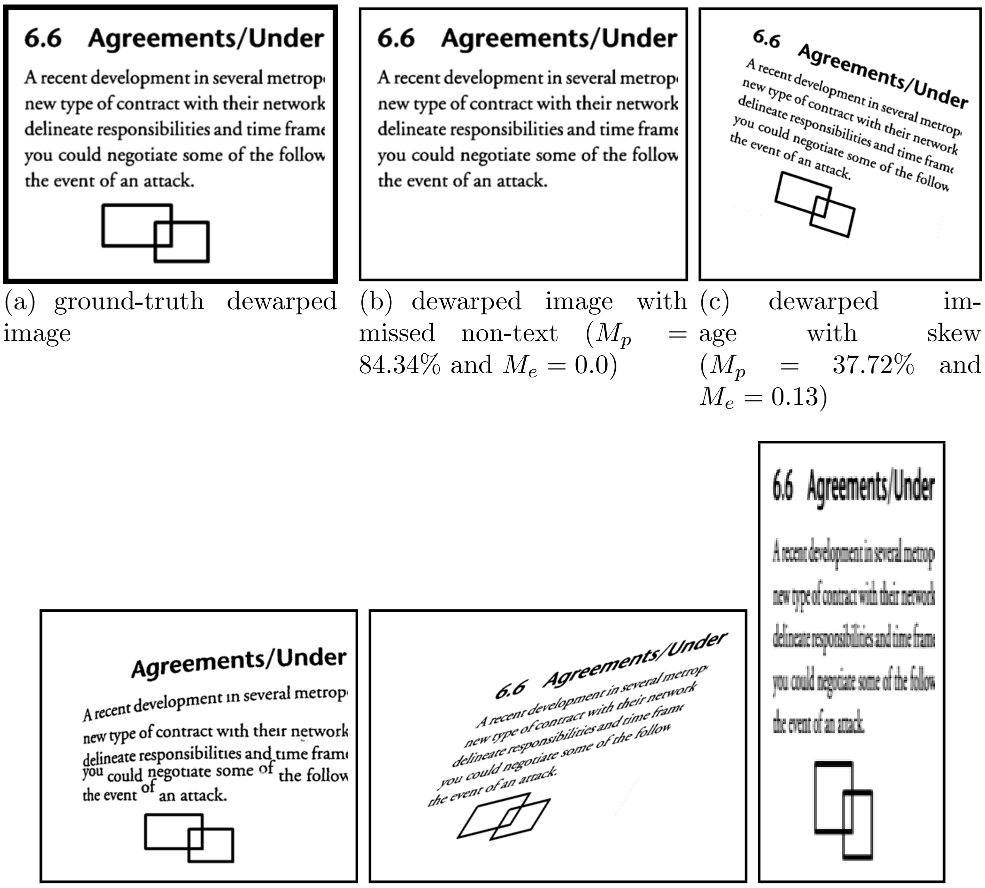
Figure 9.3: Matching between the SIFT features of: a) the ground-truth image (Figure 9.2(b)) with itself, b) the ground-truth image and the good dewarped image (Figure 9.2(c)), c) the ground-truth image and the bad dewarped image (Figure 9.2(d)).

dewarped images are also shown in Figure 9.2(c) and Figure 9.2(d), respectively. Here, it can be noticed that the good dewarped image visually looks similar to the ground-truth image and contains both text and non-text elements, except slight non-linearity in text lines and different aspect ratio. The bad dewarped image, on the other hand, missed most of the non-text elements and some of the text elements along with irregularity/non-linearity in text lines. The SIFT based matching between: i) the ground-truth image with itself is shown in Figure 9.3(a), ii) the ground-truth image and the good dewarped image is shown in Figure 9.3(b), and iii) the ground-truth image and the bad dewarped is shown in Figure 9.3(c). The ground-truth image matches perfectly with itself as shown in the Figure 9.3(a). Most of the matches in Figure 9.3(b) and Figure 9.3(c) are correct with respect to the corresponding descriptors and their locations, and some of them are only correct with respect to the corresponding descriptors, but not with the corresponding locations. In order to remove these types of wrong matches, a filtering criteria is used, according to which, all those matches that have distances greater than $T\%$ of document diagonal are removed. The value of T can be set in-between 0% to 100%, where $T = 0\%$ means that the matched descriptors should be at the perfectly same locations otherwise discarded, and $T = 100\%$ means that the locations of matched descriptors can be far apart. Both of these extreme values are not suitable for the case here. A reasonable value can be set in-between 10% to 30%. It is also important to note that, the number of matches between the ground-truth image and the good dewarped image are more than the number of matches between the the ground-truth image and the bad dewarped image. Therefore, the number of matches and other related metrics can be used for the performance evaluation of page dewarping methods, which are discussed below.

Consider that two dewarped images are given, the dewarped image I, and the ground-truth dewarped image G. Let, L_I and D_I represent the locations and normalized descriptors of the SIFT features for the dewarped image I, and L_g and D_g represent the SIFT features for the ground-truth dewarped image G. If the dewarped image I agrees perfectly with the ground-truth dewarped image G, there will be a perfect matching between their corresponding SIFT features as shown in Figure 9.3(a). If there are differences between the two dewarped images, then there will not be a perfect matching as shown in Figure 9.3(b) and Figure 9.3(c).

Here, two different performance measures are defined to evaluate different aspects of the behavior of a page dewarping algorithm using the SIFT based features matching. These measures are defined as follows:

1. **Matching Percentage M_p:** let total number of matches between G and I is


6.6 Agreements/Under
A recent development in several metrop
new type of contract with their network
delineate responsibilities and time fram
you could negotiate some of the follow
the event of an attack.

(a) ground-truth dewarped image

(b) dewarped image with missed non-text ($M_p = 84.34\%$ and $M_e = 0.0$)

(c) dewarped image with skew ($M_p = 37.72\%$ and $M_e = 0.13$)

(d) dewarped image with warped, missed and irregular text ($M_p = 14.59\%$ and $M_e = 0.19$)

(e) dewarped image with perspective distortions ($M_p = 0\%$)

(f) dewarped image with incorrect aspect ratio ($M_p = 0\%$)

Figure 9.4: Behavior of the proposed performance evaluation metrics (matching percentage (M_p) and matching error (M_e) in the presence of typical errors produced by dewarping methods. A dewarped image with warped text, perspectively distorted text, and/or incorrect aspect ratio can be considered as the much more erroneous than missed non-text or global skew with respect to OCR performance.

represented by N, and total number of features in G is represented by N_G. The matching percentage (M_p) is defined as:

$$M_p = \frac{N}{N_G} \tag{9.1}$$

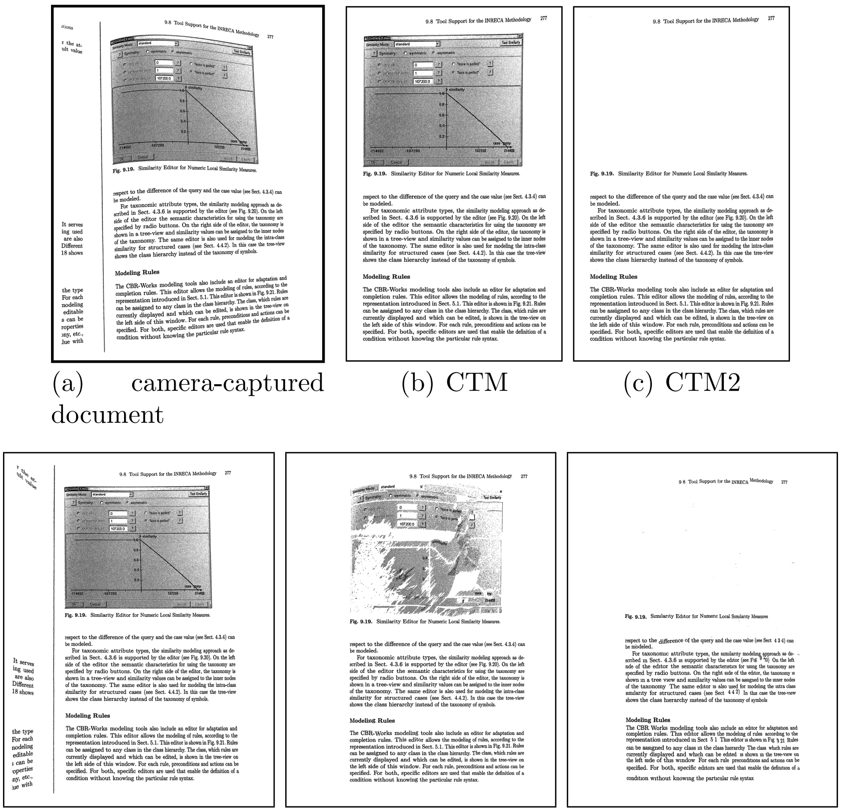
2. **Matching Error M_e:** for a pair of matched descriptors p, let $D_G(p)$ represents a descriptor in G, and $D_I(p)$ represent a corresponding matched descriptor in I. The mean error of all matching pairs is calculated as follows:

$$M_e = \frac{\sum_{p=1}^{N} \arccos(D_G(p) \cdot D_I(p))}{N} \tag{9.2}$$

The effectiveness and correctness of the presented metrics can be analyzed by comparing a ground-truth dewarped image with both a good and a bad dewarped images, such an example is shown in the Figure 9.2. For the good dewarped image (Figure 9.2(c)), the values of these metrics are as follows: $M_p = 44.57\%$ and $M_e = 0.15$. Similarly, these values for the bad dewarped image (Figure 9.2(d)) are as follows: $M_p = 11.73\%$ and $M_e = 0.19$. As shown in the Figure 9.2(c), the qualities of the good and the bad dewarped images are consistent with their corresponding values of matching percentage (M_p) and matching error (M_e).

The proposed metrics are also effective in terms of indicating typical errors produced by dewarping methods such as i) missed non-text parts as shown in Figure 9.4(b) where $M_p = 84.34\%$ and $E_m = 0.0$, ii) global skew as shown in Figure 9.4(c) where $M_p = 37.72\%$ and $M_e = 0.13$, iii) warped, missed, and irregular text as shown in Figure 9.4(d) where $M_p = 14.59\%$ and $M_e = 0.19$, iv) perspective distortion as shown in Figure 9.4(e) where $M_p = 0\%$, and v) incorrect aspect ratio as shown in Figure 9.4(f) where $M_p = 0\%$. The main purpose of dewarping is to transform warped, non-planar documents into planar images so that traditional scanner based OCR software can also process them equally like scanned documents. These results are consistent with the visual (planar) quality of dewarped images as well as with respect to OCR accuracy.

In order to analyze some additional visual quality aspects of a dewarping method that do not directly influence OCR accuracy, standard deviation of matching locations can be estimated between a ground-truth image and its corresponding dewarped image. For example, the standard deviations of plane, skewed and irregular document images as shown in Figure 9.4 with respect to the ground-truth image are equal to 0, 8, and 4.65,



9.8 Tool Support for the INRECA Methodology 277

Fig. 9.19. Similarity Editor for Numeric Local Similarity Measures.

respect to the difference of the query and the case value (see Sect. 4.3.4) can be modeled.

For taxonomic attribute types, the similarity modeling approach as described in Sect. 4.3.6 is supported by the editor (see Fig. 9.20). On the left side of the editor the semantic characteristics for using the taxonomy are specified by radio buttons. On the right side of the editor, the taxonomy is shown in a tree-view and similarity values can be assigned to the inner nodes of the taxonomy. The same editor is also used for modeling the intra-class similarity for structured cases (see Sect. 4.4.2). In this case the tree-view shows the class hierarchy instead of the taxonomy of symbols.

Modeling Rules

The CBR-Works modeling tools also include an editor for adaptation and completion rules. This editor allows the modeling of rules, according to the representation introduced in Sect. 5.1. This editor is shown in Fig. 9.21. Rules can be assigned to any class in the class hierarchy. The class, which rules are currently displayed and which can be edited, is shown in the tree-view on the left side of this window. For each rule, preconditions and actions can be specified. For both, specific editors are used that enable the definition of a condition without knowing the particular rule syntax.

(a) camera-captured document (b) CTM (c) CTM2

(d) SKEL (e) Ridge-Snake (f) SEG

Figure 9.5: Example results of different methods for a sample camera-captured document of the DFKI-I dataset: b) CTM [FWL+07], c) CTM2 [FWL+07], d) SKEL [MM07], e) Ridge-Snake (Chapter 8), f) SEG [GAS07].

respectively. It is important to note that, the skewed image (Figure 9.4(c)) has bigger standard deviation as compared to the irregular text (Figure 9.4(d)), but the skewed image may produce less number of OCR errors than the irregular text, mainly because a skew correction step is a part of standard OCR pipeline.

9.3 Experiment and Results

As a first step towards comparative evaluation of page dewarping techniques, a page dewarping contest using the DFKI-I camera-captured documents dataset was organized along with the CBDAR 2007 workshop [SB07]. Three groups participated in the contest. These three method are referred as CTM [FWL+07], SKEL [MM07], and SEG [GAS07]. The CTM method also used their programs to remove graphics and images from the processed pages. The results thus produced are referred to as CTM2. For the description

Table 9.1: OCR based error rate (by edit distance) of different dewarping methods on the DFKI-I dataset.

Algorithm	**Error Rate (by edit distance)**
CTM2 [FWL$^+$07]	1.758
Ridge-Snake (Chapter 8)	1.917
CTM [FWL$^+$07]	2.113
SKEL [MM07]	2.162
SEG [GAS07]	4.088

of the participating methods please refer to [SB07]. The text line based dewarping method is also presented in Chapter 8, referred to as Ridge-Snake, and compared its performance with those of contest participants. For a sample camera-captured document image of the DFKI-I dataset, the dewarped images of all these methods are shown in Figure 9.5.

These different methods have been compared with each other through OCR based error rate by edit distance using ASCII text ground-truth (Chapter 8). The OCR based performance evaluation results are shown in Table 9.1. The CTM2 method performs the best on the DFKI-I dataset, and its results are better than CTM, i.e. after post-processing to remove graphics and images. This is because the ground-truth ASCII text contains text coming only from the textual parts of the documents, so the text that is present in graphics or images is ignored. Hence, the dewarped documents that contain text inside graphics regions get higher error rate by edit distance. On the basis of OCR based performance evaluation, CTM, CTM2, SKEL and Ridge-Snake have similar performance, and SEG has relatively inferior performance.

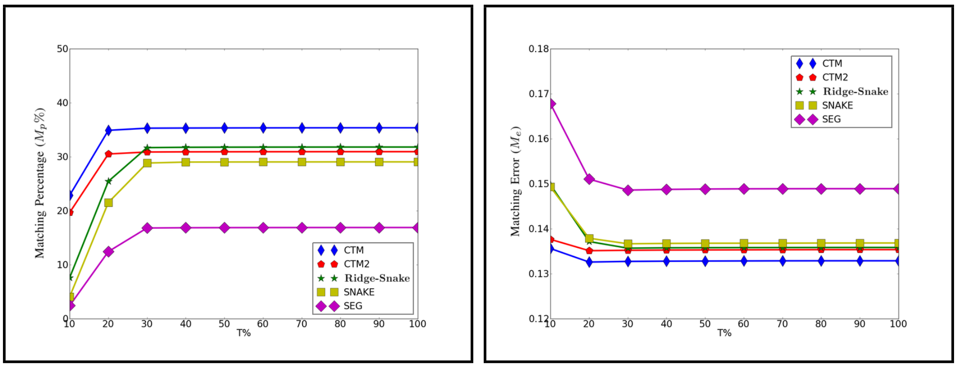
From the descriptions of the dewarping methods, it has been determined that both CTM and SKEL handle non-text elements together with text elements, but SEG and Ridge-Snake methods mainly perform dewarping for text elements and do not handle non-text elements. One of such examples for the DFKI-I dataset can be seen in Figure 9.5.

In this chapter, these dewarping methods are compared using the presented performance evaluation metrics (matching percentage (M_p), matching error (M_e)) on the DFKI-I dataset. The SIFT features based performance evaluation results of the dewarping methods for different values of T (10% to 100%) are shown in Figure 9.6. For an empirically selected optimal value of T (i.e. T = 20%), the SIFT features based performance evaluation results are shown in Table 9.2. CTM method has achieved the best matching percentage (M_p) among all other methods. The matching percentage and

Table 9.2: The SIFT features based performance evaluation results of different dewarping methods on the DFKI-I dataset using proposed vectorial performance evaluation metrics (matching percentage (M_p) and matching error (M_e)). It is also interesting to note that the SIFT features based performance evaluation results are also closely consistent with the OCR based results.

Algorithm	$M_p\%$	M_e
CTM [FWL$^+$07]	34.90%	0.13
CTM2 [FWL$^+$07]	30.51%	0.14
SKEL [MM07]	25.45%	0.14
Ridge-Snake (Chapter 8)	21.52%	0.14
SEG [GAS07]	12.44%	0.15

matching error of CTM are better than the CTM2, which is also perfectly consistent with the definition of CTM2 (i.e. removed graphics and images). CTM method has also achieved the lowest matching error (M_e) as compared to other methods. On the other hand, SEG has comparatively achieved the lowest matching percentage and highest matching error in comparison to other methods. It is very interesting to note that these SIFT features based performance evaluation results are also closely consistent with the OCR based results. However, the SIFT features based results give more details about the quality of dewarped images with respect to both text and non-text elements.



(a) Matching Percentage ($M_p\%$)

(b) Matching Error (M_e)

Figure 9.6: Comparative performance evaluation of different methods for the DFKI-I dataset by using the presented SIFT features based performance evaluation metrics (matching percentage (M_p) and matching error (M_e)) for different values of T.

9.4 Conclusion

This chapter has described a SIFT features based method for evaluating the performance of dewarping algorithms. Unlike OCR based performance evaluation techniques [SB07, BSB09a], a feature based technique indicates how well a dewarping method performs on both text and non-text elements in warped images. Unlike previous feature based performance evaluation techniques [SGP09], the presented feature based technique does not require manual labeling for generating ground-truth images, and calculate vectorial performance evaluation metrics (matching percentage (M_p) and matching error (M_e)), instead of a single score that may not be sufficient to compare the performance of different dewarping methods. It has also been demonstrated that the SIFT features based performance evaluation results are consistent with the OCR based performance evaluation results.

Part IV

Layout Analysis of Complex Script Documents

Chapter 10

High Performance Layout Analysis of Arabic and Urdu Documents [1]

Summary: *layout analysis–extraction of text lines from a document image and identification of their reading order–is an important step in converting the document into a searchable electronic representation. Projection methods are typically employed for extraction of text lines in Arabic script documents. Although projection methods achieve good accuracy on clean, skew-free documents; their performance drops under challenging situations (border noise, skew, complex layouts, etc.). This chapter presents a layout analysis system for extracting text lines in reading order from scanned Arabic script document images written in different languages (Arabic, Urdu, Persian, etc.) and different styles (Naskh, Nastaliq, etc.). The presented system is based on a suitable combination of different well-established techniques that have proven to be robust against different types of document image degradations. The main contribution of this chapter is to show the effectiveness of these techniques on a variety of Arabic script document images.*

10.1 Introduction

Layout analysis deals with text lines detection and their reading order determination in document images. A wide variety of layouts in large scale document digitization

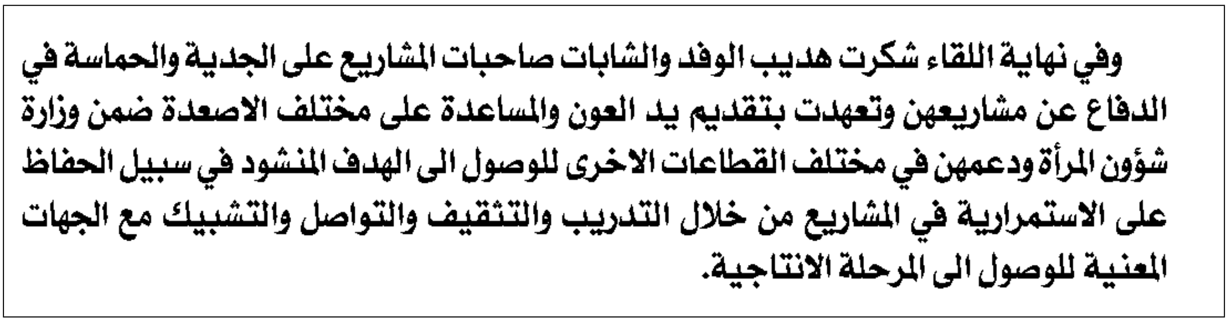
[1] This work was published in Bukhari et al. [BSB11b] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. High performance layout analysis of Arabic and Urdu document images. In Proceedings 11th International Conference on Document Analysis and Recognition, pages 1275-1279, Beijing, China, 2011. Copyright ©2012 IEEE"*, and Bukhari et al. [BSB12b] *"S. S. Bukhari, F. Shafait, and T. M. Breuel. Layout analysis of arabic script documents. In Guide to OCR for Arabic Scripts. Springer-Verlag, 2012. Copyright ©Springer-Verlag Berlin Heidelberg 2012"*. This chapter is an adapted version of the published work.

projects poses stern challenges to document image analysis. A document image may contain different types of contents like text, graphics, halftones, etc. The goal of optical character recognition (OCR) is to extract text from a document image. This is achieved in two steps. The first step, geometric layout analysis, locates text lines in the image and identifies their reading order. In the second step, text lines identified by the layout analysis step are fed to a character recognition engine which converts them into text in an appropriate format (ASCII, UTF-8, etc.).

The Arabic script is used for writing several languages of Asia and Africa like Arabic, Urdu, Persian, Pashto, Kurdi, Jawi, etc. After Latin script, it is the second most widely used script in the world. It is a cursive script, i.e. individual characters are usually combined to form ligatures. Although there are many styles for writing Arabic script, the most widely used styles are Naskh and Nastaliq. Naskh writing style is dominant in Arabic and Pashto languages, whereas Nastaliq is the standard style adopted for writing Urdu and Persian. Examples of printed Arabic text written in Naskh script and Urdu text written in Nastaliq script are shown in Figure 10.1. From layout analysis point of view, the main differences of Nastaliq script as compared to Naskh script are: (i) very small inter-line and inter-word spacing, (ii) tall ascenders and descenders that overlap into adjacent text lines.

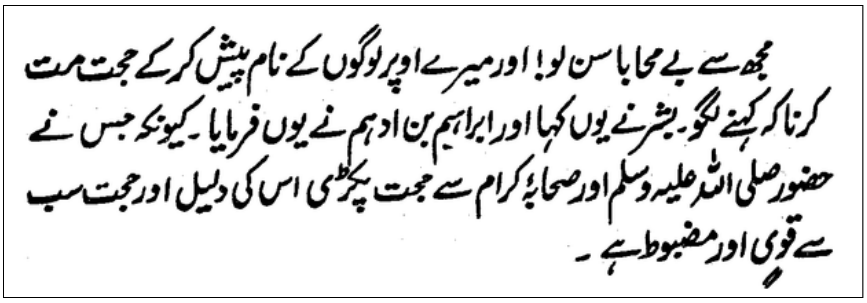
One of the primary goals of Arabic script OCR is word recognition [AM11]. Since Arabic is generally written in Naskh script, text line segmentation using horizontal projections works quite well for machine printed documents mainly because of large inter-line spacing [Kho02]. Therefore, very few approaches have been proposed for text line extraction from machine-printed Arabic script document images. However, horizontal projection is a basic approach that works only for clean, single-column documents with large inter-line spacing, but does not handle multi-column documents. For handling multi-column documents, some advanced horizontal projection based approaches are used like X-Y cut [SSMSSS06], morphology [SSMSSS06], etc.. More sophisticated approaches for text line extraction have been presented in the domain of segmenting handwritten and historical Arabic documents [ZTMR01, BZA$^+$10]. However, the key problem addressed in these approaches is to handle local non-linearity of text lines.

Several layout analysis algorithms have been presented in the literature [CCMM98a, Nag00] over the last two decades. Some of these approaches are quite robust to the presence of noise and work for different document layouts, which come to widespread use for analyzing document images in different scripts. The performance of six algorithms for page segmentation on Nastaliq script was evaluated by Kumar et al. [KKJ07] that



وفي نهاية اللقاء شكرت هديب الوفد والشابات صاحبات المشاريع على الجدية والحماسة في
الدفاع عن مشاريعهن وتعهدت بتقديم يد العون والمساعدة على مختلف الاصعدة ضمن وزارة
شؤون المرأة ودعمهن في مختلف القطاعات الاخرى للوصول الى الهدف المنشود في سبيل الحفاظ
على الاستمرارية في المشاريع من خلال التدريب والتثقيف والتواصل والتشبيك مع الجهات
المعنية للوصول الى المرحلة الانتاجية.

(a) Sample Arabic document written in Naskh script.



مجھ سے بے محابا سن لو! اور میرے اوپر لوگوں کے نام پیش کر کے حجت مت
کرنا کہ کہنے لگو۔ بشر نے یوں کہا اور ابراہیم بن ادہم نے یوں فرمایا۔ کیونکہ جس نے
حضور صلی اللہ علیہ وسلم اور صحابۂ کرام سے حجت پکڑی اس کی دلیل اور حجت سب
سے قوی اور مضبوط ہے۔

(b) Sample Urdu document written in Nastaliq script.

Figure 10.1: An example of printed Arabic text in Naskh script and Urdu text in Nastaliq script. Text lines in Nastaliq have very little spacing between them as compared to Naskh script.

include X-Y cut [NSV92], smearing [WCW82], whitespace analysis [Bai94], constrained text line finding algorithm [Bre02b], Docstrum [O'G93], and Voronoi-diagram based approach [KSI98]. As shown by Shafait et. al [SKB08b], these algorithms perform very well for segmenting documents in Latin script, but as shown by Kumar et al. [KKJ07] none of these algorithms was able to achieve an accuracy of more than 70% on Nastaliq script documents having simple book layouts with no font size variations within each page.

Urdu, a national language of Pakistan, is mostly written in Nastaliq script using more than 20,000 ligatures [Wik]. There has been very little work in the area of Urdu or Persian document analysis as compared to Arabic OCR. An Urdu character recognition method is proposed by Husain et al. [HA02] for the Nastaliq script, where the layout analysis step was skipped to concentrate more on the OCR part. Pal et al. [PS03] also presented an approach for recognizing printed Urdu documents. In their system, first of all skew correction is done using Hough transform. Then, text lines are segmented by horizontal projection. For extracting text lines from printed Persian documents, a similar approach is used by Jelodar et al. [JFMF05].

Shafait et al. [SHKB06] presented an adaptation of the layout system described

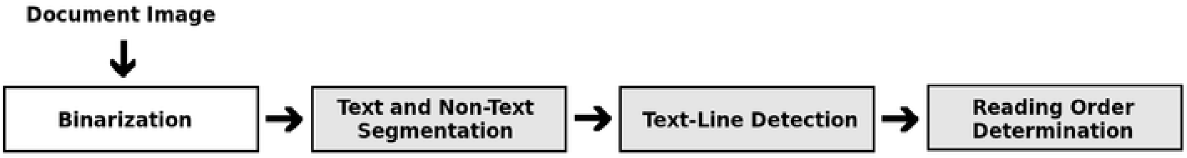
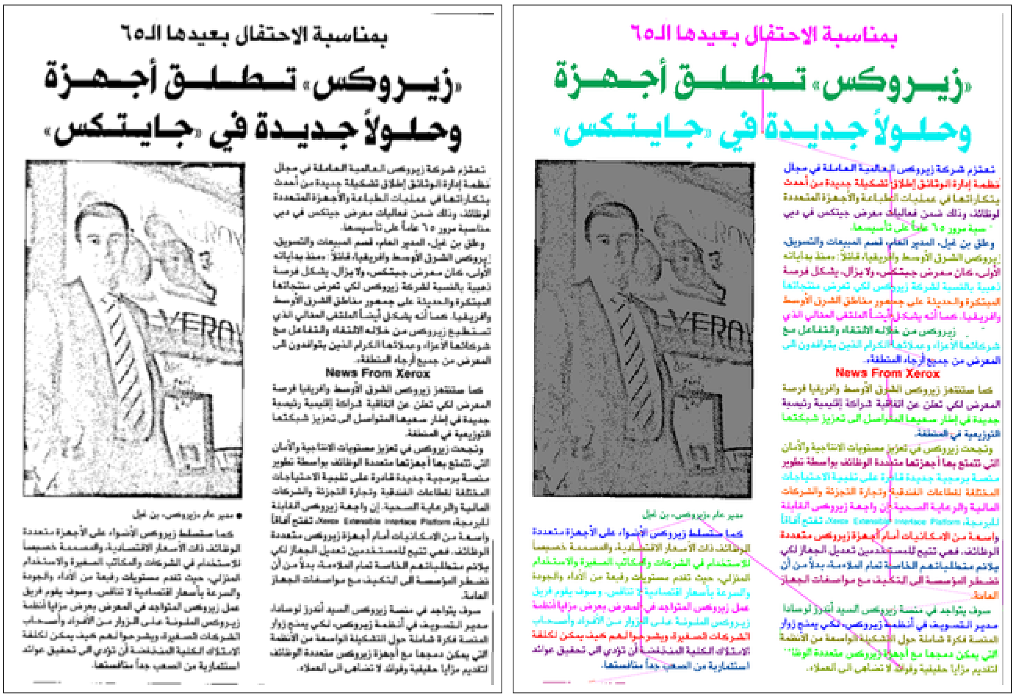


Figure 10.2: Processing flow of a high performance generic layout analysis system. Filled blocks show the areas to which this chapter discussed in detail.

in [Bre03] to Urdu script documents. First, they evaluated empty whitespace rectangles as candidates for column separators or gutters. Text lines are then detected by modifying the RAST based text line finding algorithm [Bre02b] where column separators are introduced as “obstacles”. Finally, text lines were analyzed for determining the reading order using constraints on the geometric arrangement of text line segments on the page. Particular advantages of their system are that it is nearly a parameter-free approach and robust to the presence of noise in document images.

In this chapter, a layout analysis system is presented that is a combination of the morphology based text/non-text segmentation method (Chapter 5), the ridge-based text line extraction method (Chapter 3), and the reading order determination method [SHKB06]. The presented layout analysis system is applicable to a wide variety of Arabic scripts binary document images. A grayscale document image can be first converted into binary form using an appropriate binarization approach such as Otsu [Ots79] and Sauvola [SP00], which are commonly used state-of-the-art binarization approaches. A possible flow of generic layout analysis system is shown in Figure 10.2. Here, the following steps are discussed: text and non-text segmentation, text lines detection and their reading order determination. For a sample Arabic script document image, output of the layout analysis system is shown in Figure 10.11.

The rest of the chapter is organized as follows. In Section 10.2, the multiresolution morphology based text and non-text segmentation algorithm (Chapter 5) is briefly described. The state-of-the-art X-Y cut [NSV92] and ridge-based (Chapter 3) text line finding methods are explained in Section 10.3. A topological sorting based reading order determination algorithm [SHKB06] is discussed in Section 10.4 followed by experimental results and a conclusion in Section 10.5 and Section 10.6, respectively.



بمناسبة الاحتفال بعيدها الـ٦٥

«زيـــروكس» تـــطـــلـــق أجــهــزة
وحـــلـــولاً جـديـــدة في «جـايـتـكس»

تعتزم شركة زيروكس العالمية العاملة في مجال
نظمة إدارة الوثائق إطلاق تشكيلة جديدة من أحدث
بتكاراتها في عمليات الطباعة والأجهزة المتعددة
لوظائف وذلك ضمن فعاليات معرض جيتكس في دبي
مناسبة مرور ٦٥ عاماً على تأسيسها.

وعلق بن غيل، المدير العام، قسم المبيعات والتسويق،
زيروكس الشرق الأوسط وافريقيا، قائلاً: «منذ بداياته
الأولى، كان معرض جيتكس، ولا يزال، يشكل فرصة
ذهبية بالنسبة لشركة زيروكس لكي تعرض منتجاتها
المبتكرة والحديثة على جمهور مناطق الشرق الأوسط
وافريقيا. كما أنه يشكل أيضاً الملتقى المثالي الذي
تستطيع زيروكس من خلاله الالتقاء والتفاعل مع
شركائها الأعزاء وعملائها الكرام الذين يتوافدون الى
المعرض من جميع أرجاء المنطقة».

News From Xerox

كما ستنتهز زيروكس الشرق الأوسط وافريقيا فرصة
المعرض لكي تعلن عن اتفاقية شراكة إقليمية رئيسية
جديدة في إطار سعيها المتواصل الى تعزيز شبكتها
التوزيعية في المنطقة.

ونجحت زيروكس في تعزيز مستويات الانتاجية والأمان
التي تتمتع بها أجهزتها متعددة الوظائف بواسطة تطوير
منصة برمجية جديدة قادرة على تلبية الاحتياجات
المختلفة لقطاعات الفندقية وتجارة التجزئة والشركات
المالية والرعاية الصحية. إن واجهة زيروكس القابلة
للبرمجة Xerox Extensible Interface Platform تفتح آفاقاً
واسعة من الامكانيات أمام أجهزة زيروكس متعددة
الوظائف، فهي تتيح للمستخدمين تعديل الجهاز لكي
يلائم متطلباتهم الخاصة تمام الملاءمة بدلاً من أن
تضطر المؤسسة الى التكيف مع مواصفات الجهاز
العامة.

سوف يتواجد في منصة زيروكس السيد أندرو لوساند،
مدير التسويق في أنظمة زيروكس، لكي يمنح زوار
المنصة فكرة شاملة حول التشكيلة الواسعة من الأنظمة
التي يمكن دمجها مع أجهزة زيروكس متعددة الوظائف
لتقديم مزايا حقيقية وفوائد لا تتناهى الى العملاء.

• مدير عام «زيروكس» بن غيل

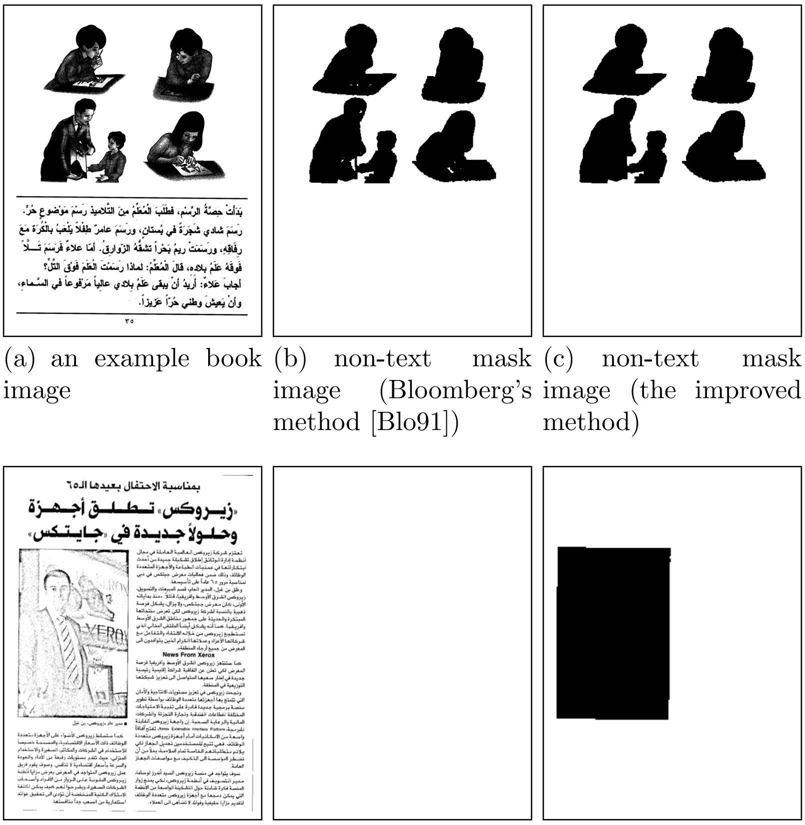
كما ستسلط زيروكس الأضواء على الأجهزة متعددة
الوظائف ذات الأسعار الاقتصادية والمصممة خصيصاً
للاستخدام في الشركات والمكاتب الصغيرة والاستخدام
المنزلي، حيث تقدم مستويات رفيعة من الأداء والجودة
والسرعة بأسعار اقتصادية لا تنافس. وسوف يقوم فريق
عمل زيروكس المتواجد في المعرض بعرض مزايا أنظمة
زيروكس الملونة على الزوار من الأفراد وأصحاب
الشركات الصغيرة، ويشرحوا لهم كيف يمكن لتكلفة
الامتلاك الكلية المنخفضة أن تؤدي الى تحقيق عوائد
استثمارية من الصعب جداً منافستها.

Figure 10.3: A sample newspaper image printed in Arabic Naskh script and its corresponding layout analysis result: gray-region represents non-text components, color-coded labeling represent segmented text lines and magenta-line shows text lines reading order. [Note: left image has been taken from the website [IJM].]

10.2 Text and Non-Text Segmentation

As mentioned in Chapter 5, Bloomberg [Blo91] presented a multiresolution morphology based text and non-text segmentation method. It is a simple and script independent text and non-text segmentation method. It is based on the assumption that the size of non-text elements is larger than text elements in document images. It performs well for halftone mask segmentation, for which it was designed, but most of the time fails to accurately segment drawing type non-text elements such as line art, maps etc. An improved multiresolution morphology based text and non-text segmentation algorithm is presented in Chapter 5, that can handle halftones as well as drawing type non-text elements.

Figure 10.4 shows sample Arabic script document images and their text and non-text segmentations for the original version of multiresolution morphology based text and non-text segmentation method [Blo91] and its improved version (Chapter 5). The improved version performs well for these examples as compared to the original version.



(a) an example book image

(b) non-text mask image (Bloomberg's method [Blo91])

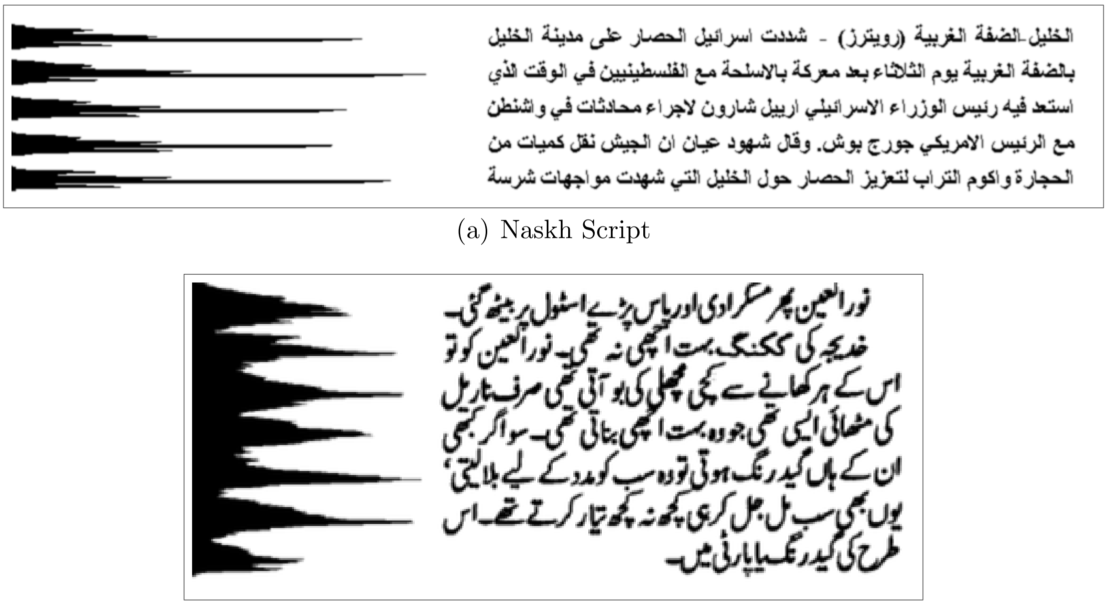
(c) non-text mask image (the improved method)

(d) an example newspaper image

(e) non-text mask image (Bloomberg's method [Blo91])

(f) non-text mask image (the improved method)

Figure 10.4: Results of non-text masks using Bloomberg's multiresolution morphology based text and non-text segmentation method [Blo91] and the improved version (Chapter 5). Top figure shows a simple case where non-text components are larger than text components and can also be separated by using median size of connected components analysis. Bottom figure shows a challenging condition where size of non-text components are comparable or even smaller than text components. In contrast to [Blo91], improved multiresolution morphology based text and non-text segmentation algorithm gives correct result for both simple and challenging conditions.



(a) Naskh Script

(b) Naskh Script

Figure 10.5: Horizontal projection of Arabic scripts. The top figure shows the case of larger, well-defined inter-line spacing in Naskh script and there are between-line zero-valleys in the projection profile. The bottom figure shows the case of small inter-line spacing in Nastaliq script (Urdu) and there are no between-line zero-valleys in the projection profile.

In Arabic script document images, like Latin script images, the size of text elements is usually smaller than non-text elements and this situation fits well to the assumption of multiresolution morphology based text and non-text segmentation. Therefore, this approach also works well for Arabic script document images.

10.3 Text line Detection

Text line detection is an important layout analysis step in document image processing. It is often used before feeding a page to character recognition engine. The performance of text lines detection operation directly influences the accuracy of recognition engine.

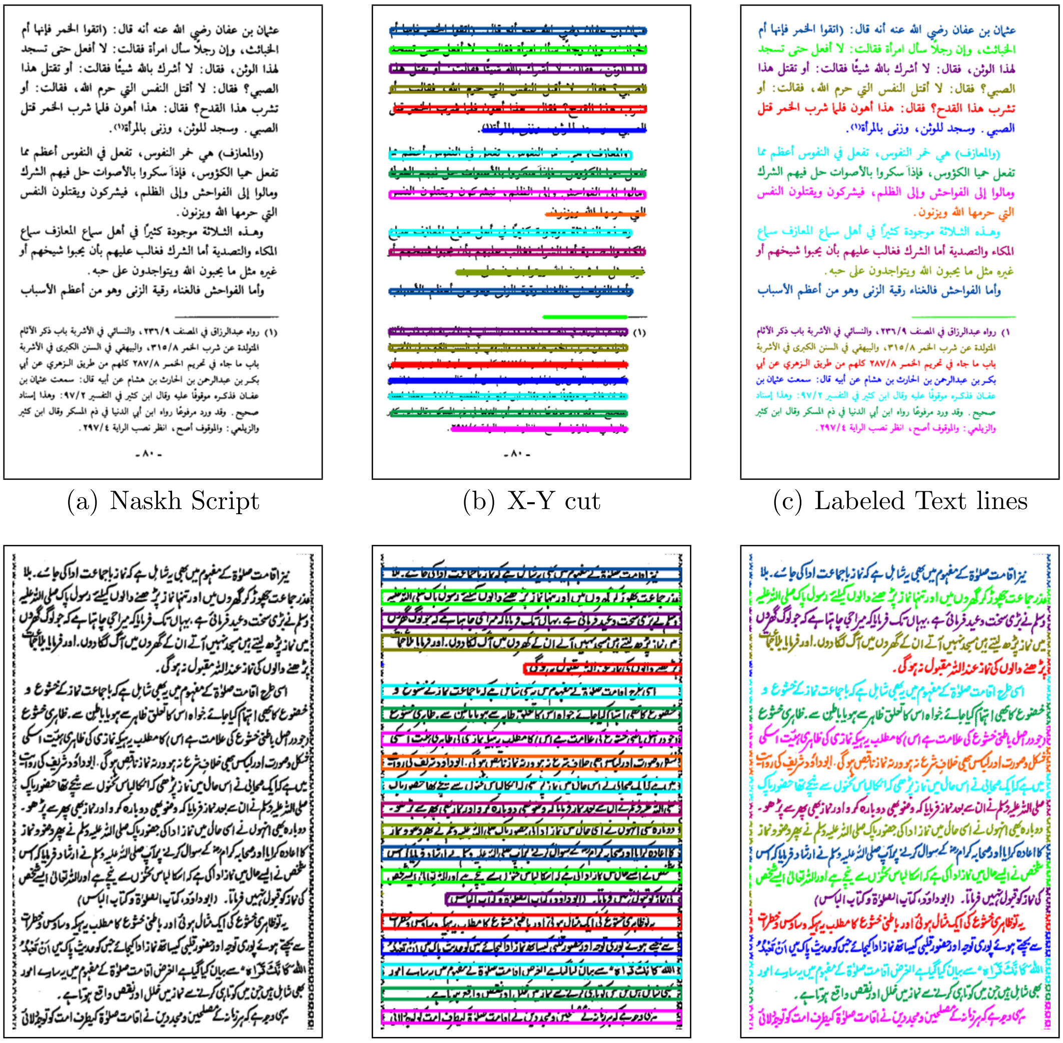
In the literature, a large number of text line detection approaches are proposed for Arabic document images. Among them, projection profile analysis is a widely used algorithm for detecting text lines in Arabic script document images [Kho02]. It works well for clean document images with large inter-line spacing (for example Figure 10.5(a)), but fails on document images which contain noise, multi-column and small inter-line spacing (for example Figure 10.5(b)). The X-Y cut [NSV92] is one of the state-of-the-art page segmentation approaches. It is based on project profile analysis. It can handle multi-column documents with small inter-line spacing. However, it fails on skewed document images as well as images with large amount of noise. A ridge-based generic text lines detection method is presented in Chapter 3. The ridge-based text line finding method is robust to the presence of noise, skew and small inter-line spacing. It can be used equally for text line detection in different types document images. Both the X-Y cut and ridge-based text lines detection methods are briefly described below.

10.3.1 The X-Y Cut Text Line Detection Method

The X-Y cut page segmentation method [NSV92] is a tree-based algorithm. An input document image is considered as a rectangular block. The X-Y cut algorithm recursively cuts a block into smaller blocks, until no block can be cut further. For splitting a block, first, its horizontal and vertical projection profiles are computed. The noise removal thresholds t_n^x and t_n^y are then used for computing valleys in the projection profiles. The bins of horizontal and vertical projection profiles are set to zero which contain values less than linearly scaled threshold t_n^x and t_n^y, respectively, with respect to the width and height of the block. The valleys of horizontal (v_x) and vertical (v_y) projection profiles are compared with the predefined thresholds t_x and t_y, respectively. The block is split into two blocks at the mid-point of wider of v_x and v_y which are larger than t_x and t_y respectively.

The horizontal projection profiles of sample paragraphs of Naskh and Natsaliq scripts are shown in Figure 10.5. There are clear zero-valleys in the projection profile of Naskh script corresponding to inter-line gaps between text lines. In contrast, there is no zero-valley in the projection profile of Nastaliq script. The X-Y cut method can be used to segment Nastaliq script documents. In such cases, the noise thresholds are set to a high value for finding the main body of text lines. Afterwards, the remaining portions of text lines are assigned to them through simple post-processing step. Sample document images of Nastaliq script and their correctly segmented text lines are shown in Figure 10.6. The following values of thresholds are used for generating these results: $t_n^x = 100$, $t_n^y = 100$, $t_x = 100$ and $t_y = 10$.

The X-Y cut algorithm usually fails on documents with a large amount of border noise and reports the whole page as one segment. It also produces wrong text line segmentations for skewed document images. Failed cases of the X-Y cut text line segmentation method are shown in Figure 10.7. The ridge-based text line finding algorithm (Chapter 3)



(a) Naskh Script (b) X-Y cut (c) Labeled Text lines

(d) Nastaliq Script (e) X-Y cut (f) Labeled Text lines

Figure 10.6: The X-Y cut algorithm produces correct text line segmentation results for sample Arabic script document images.

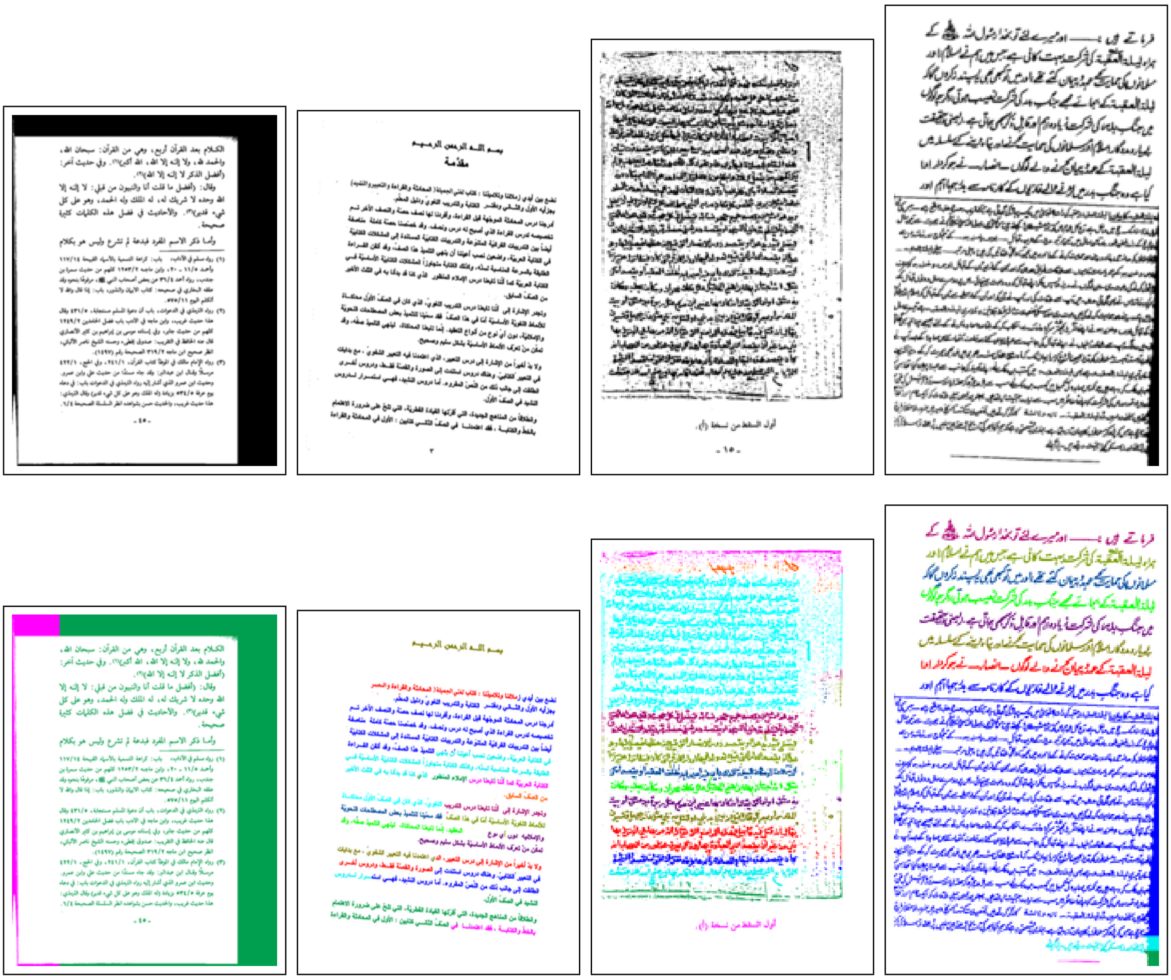


Figure 10.7: The X-Y cut method produces text lines segmentation failures for sample document images which contain challenging conditions, like border noise, skew and a large number of joined characters.

presented in the following can be used in such cases.

10.3.2 The Ridge-based Text line Detection Method

A ridge-based generic text line finding algorithm is presented in Chapter 3. The ridge-based text line finding method can be equally applied on different types of document images with respect to digitization methods (scanned or camera-captured), intensity values (binary or grayscale), scripting languages (like Latin, Chinese, Arabic, ...), and writing styles (typed-text or handwritten). It is composed of two standard image processing techniques: (i) matched filtering and (ii) ridge detection. A sample document image, its corresponding smoothed image, and the detected ridges from the smoothed image is shown in Figure 10.8(a)(a), 10.8(b)(b), and Figures 10.8(c)(c), respectively.

A ridge covers a complete region of a particular text line for single-column document images. For multi-column documents, the filter bank may fill small gaps between text lines in-between different columns. Therefore, a single ridge may cover either a single text line or multiple text lines at same height (Figure 10.8(c)). This situation causes over-segmentation errors. This type of over-segmentation errors can be corrected by *whitespace analysis* as described here.

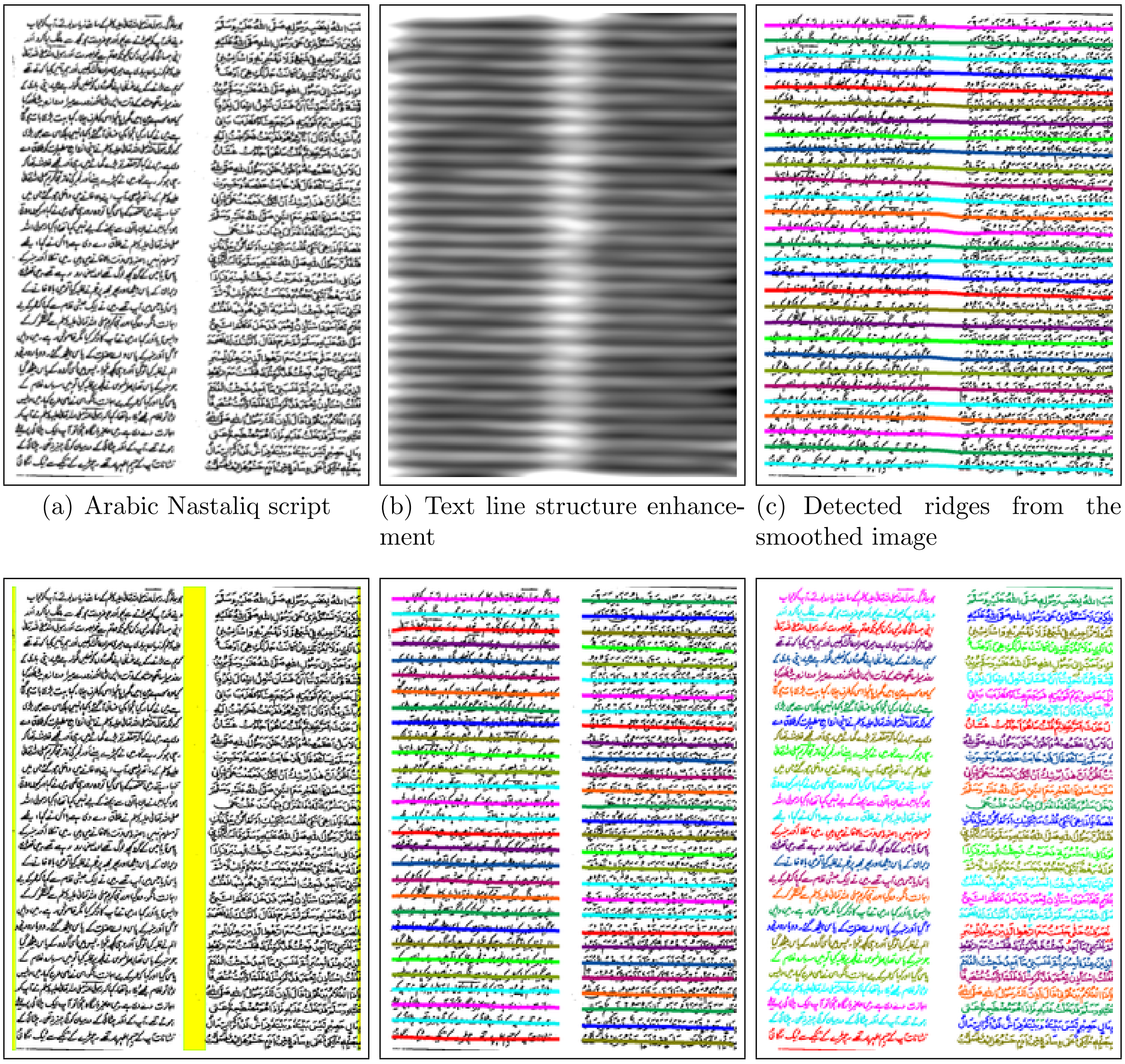
Whitespace analysis aims to find a set of maximal white rectangles in a document image such that the union of these rectangles completely covers the document's background. It is usually used for page segmentation [Bai94] or multi-column separation [Bre02b]. An algorithm for finding maximal whitespace rectangles is presented in [Bre02b]. The main idea behind that algorithm is similar to quick-sort or branch-and-bound methods. The whitespace rectangles are evaluated as candidates for column separators or gutters based on their statistics, like aspect ratio, width, etc. Whitespace cuts that correspond to column separators are shown in Figure 10.8(d). Now, under-segmentation errors (as shown in Figure 10.8(c)) can be corrected by cutting detected ridges at those points which lie over whitespace rectangles. The output ridges are shown in Figure 10.8(e), where a single ridge covers a single text line. These ridges are considered as detected text lines. Result of labeled text lines in color-coded form is shown in Figures 10.8(f).

For some of the challenging problems like document skew and noise, text line detection results of ridge-based text line extraction method are shown in Figure 10.9. As can be seen in the figure, the ridge-based text lines detection method is robust to document skew, small inter-line gaps, border noise and inter-line touching and/or overlapping as compared to the X-Y cut method (whose results are shown in Figure 10.7).

10.4 Text line Reading Order Determination

A reading order determination method tries to find the order of text lines with respect to their corresponding reading flow. Reading order can be determined by applying some ordering criteria over the positioning of text lines. In contrast to Latin script, reading order of Arabic script is from right to left. A reading order determination method is presented in [Bre03] for Latin script, which is modified for Nastaliq Arabic script in [SHKB06]. The ordering criteria that was presented in [SHKB06] is stated as follows:

- *"Text Line 'a' comes before text line 'b' if their ranges of x-coordinates overlap and if text line 'a' is above text line 'b' on the page".*



(a) Arabic Nastaliq script

(b) Text line structure enhancement

(c) Detected ridges from the smoothed image

(d) Whitespace cuts

(e) Detected text lines regions

(f) Color-coded text lines labeling

Figure 10.8: Steps of the ridge-based text line finding algorithm. a) An example image of Arabic Nastaliq script. b) Smoothed text lines (text lines enhanced) image, which is generated by using oriented anisotropic Gaussian smoothing filter bank approach. (c) Detected ridges from smoothed image using Riley based ridge detection [Ril87b, Ril87a] algorithm. There are over-segmentation errors because of multi-column. (d) Column separators that were detected through whitespace analysis; these separators help in correcting over-segmentation errors. (e) Processed ridges using whitespace separators; each ridge covers a complete region of a particular text lines. (f) Color-coded labeled text lines result using detected ridges.

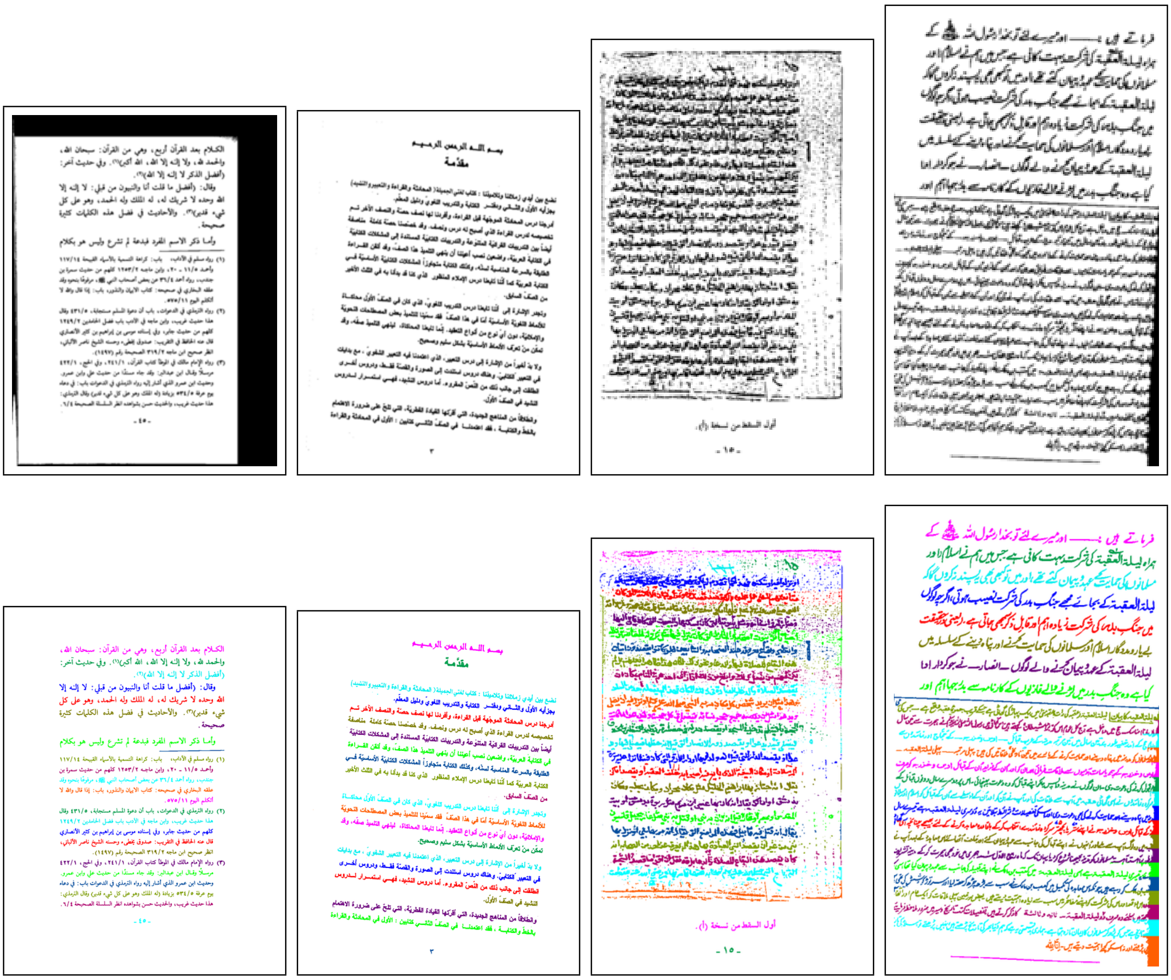
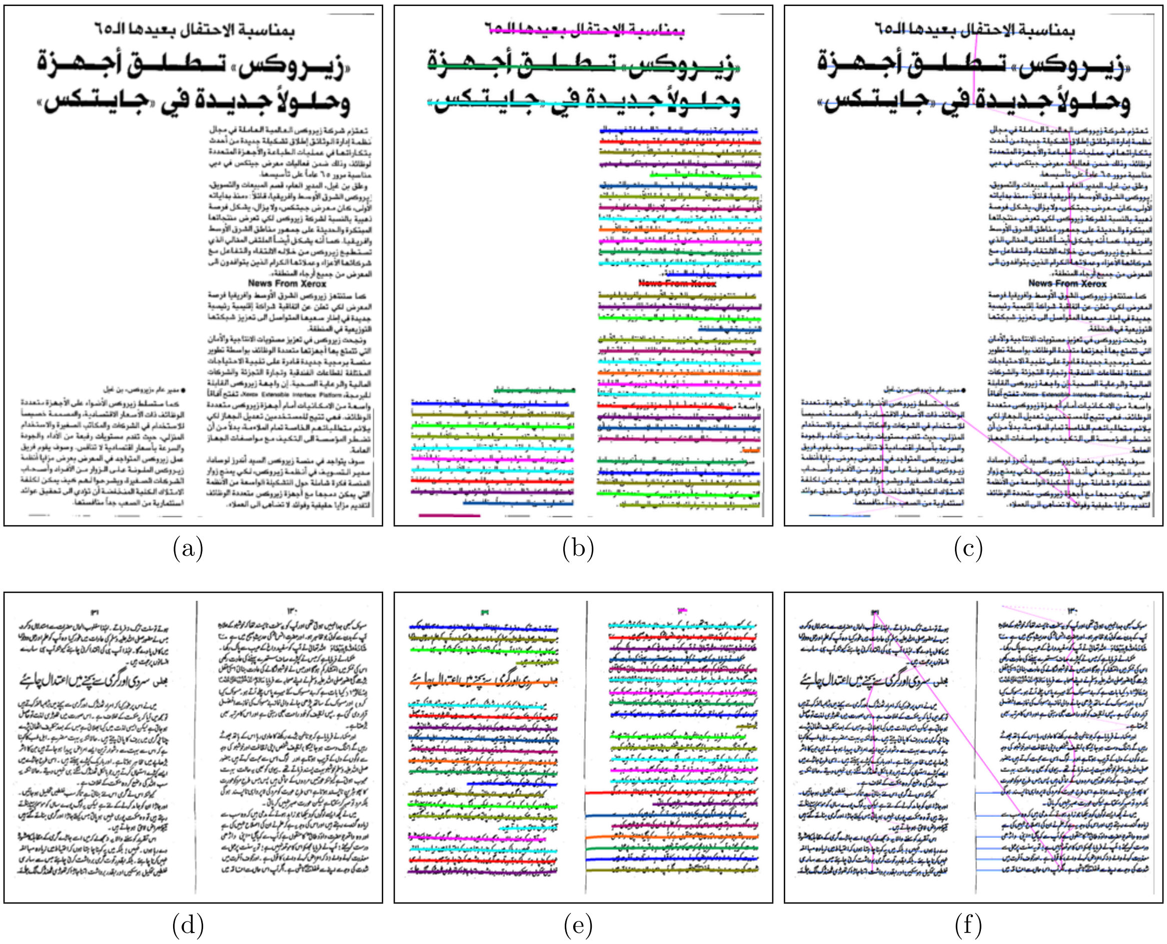


Figure 10.9: The ridge-based text line finding method produces correct text line segmentation results for sample document images with border noise, skew and a large number of joined characters.

- *"Text Line 'a' comes before text line 'b' if a is entirely to the right of 'b' and if there does not exist a text line 'c' whose y-coordinates are between 'a' and 'b' and whose range of x-coordinates overlaps both 'a' and 'b' ".*

The reading order determination method [Bre03, SHKB06] finds the partial ordering of text lines through the above defined ordering criteria, and then finds a complete order using a topological sorting algorithm [CLR90].

Examples of reading order determination on sample document images are shown in Figure 10.10. The performance of reading order determination method decreases with the decrease in text line detection accuracy and/or the presence of noise in document images.


(a) (b) (c)

(d) (e) (f)

Figure 10.10: Example images illustrating results of reading order for a newspaper and a book page. Thin horizontal lines with different colors indicate detected text line segments, and the magenta lines running down and diagonally across the image indicate reading order.

For sample Arabic and Urdu documents, result of the presented layout analysis system (text and non-text segmentation, text line extraction, and their reading order direction) are shown in Figure 10.11.

10.5 Performance Evaluation

For the performance evaluation of the presented layout analysis system, 25 images of Arabic documents are collected, mostly Naskh script, from books, newspapers, and multi-script (English and Arabic) documents. These images contain both text and non-text elements. For this dataset, text and non-text, text line, and reading order level ground-

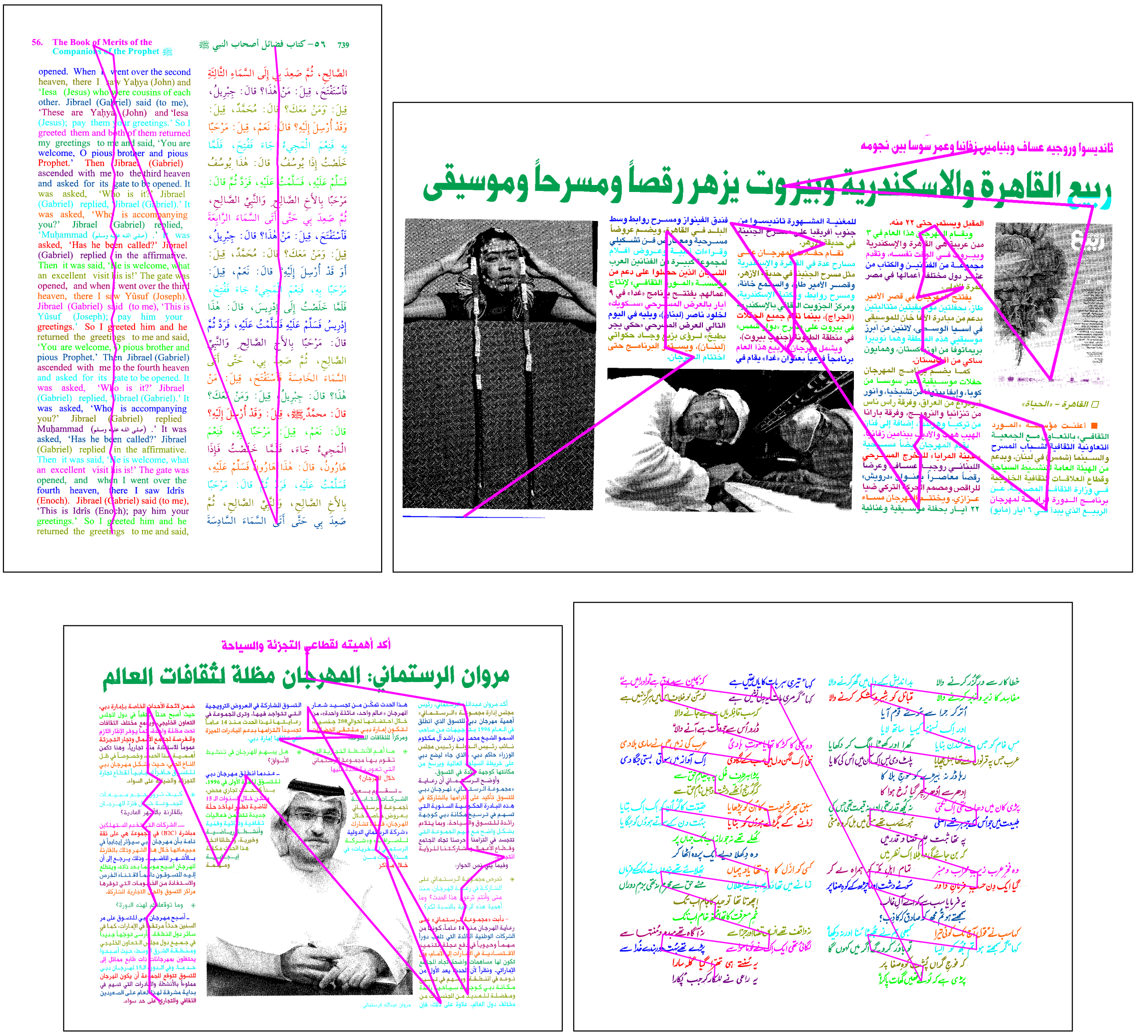


Figure 10.11: Sample images from Arabic and Urdu documents datasets and their corresponding layout analysis results: the black-pixels represents non-text components, the color coded labeling represent extracted text lines and the magenta-line shows reading order of segmented text lines. [top-left] Arabic-English book page; [top-right and bottom-left] Arabic newspapers (these images have been taken from the websites: *http://www.alrostamanigroup.ae* and http://www.mawred.org, respectively; [bottom-right] Urdu poetry image.

Table 10.1: Performance evaluation of the original multiresolution morphology based text and non-text segmentation method [Blo91] and its improved version [BSB11c] for Arabic dataset (25 documents). Both the original and improved versions achieved nearly similar segmentation accuracy because the document images in Arabic dataset are composed of only text and halftone elements.

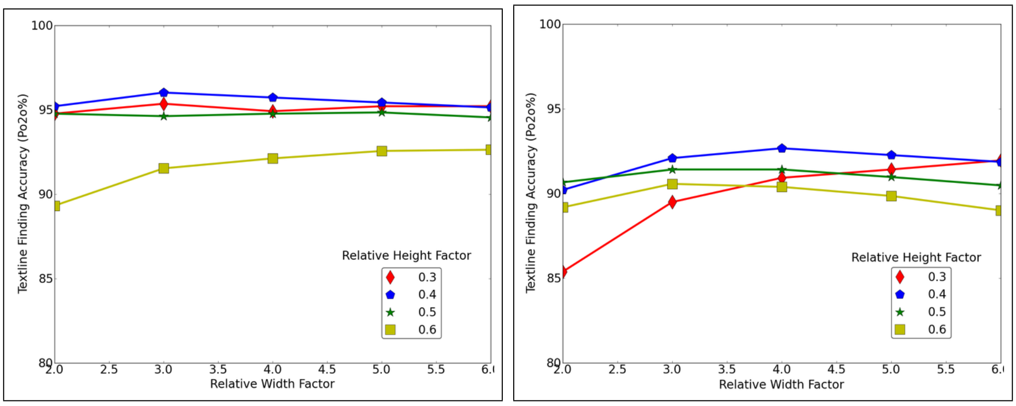
	Original [Blo91]	Improved [BSB11c]
text classified as text	99.82%	99.80%
non-text classified as non-text	99.15%	99.60%
segmentation accuracy	99.49%	99.70%

truths are prepared in color coded pixel form. From Urdu (Nastaliq script) documents dataset [SHKB06], 20 document images are also selected that belong to the categories of books, poetries, digests, and magazines. These selected documents contain only text elements. Like Arabic dataset, the text line and reading order level ground-truths are also provided in color coded form in Urdu dataset. Both datasets contain a variety of single- and multi-column layouts as shown in Figure 10.11, and hence they can be used to evaluate the performance of a layout analysis algorithm for Arabic document images.

Here, the performance evaluation of the presented layout analysis systems is done in three parts. The first part evaluates the performance of text and non-text segmentation (Section 10.5.1), the second part analyzes the errors made in text line detection (Section 10.5.2), and the third part evaluates the accuracy of reading order (Section 10.5.3).

10.5.1 Text and Non-Text Segmentation Accuracy

The performance evaluation metrics for text and non-text segmentation accuracy are described in [BSB11c]. These metrics evaluates the percentage of non-text pixels classified as non-text, text pixels classified as text, and the average of both is considered as segmentation accuracy. Here, the same metrics are applied for evaluating the performance of multiresolution morphology based text and non-text segmentation method [Blo91] and its improved version [BSB11c] on Arabic documents dataset. Performance evaluation results are shown in Table 10.1. Arabic dataset contains only text and halftone elements, and no drawing or any other type of non-text elements. Therefore, both original and improved versions achieved nearly similar and good segmentation accuracy.



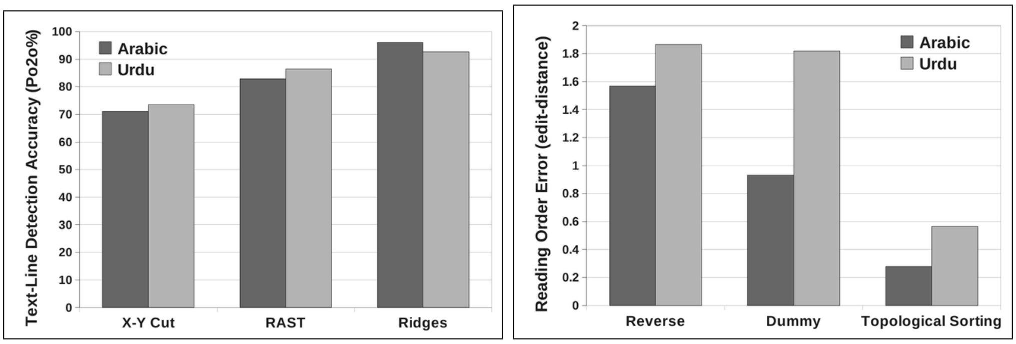
(a) Arabic Dataset (25 documents; 1358 text lines)

(b) Urdu Dataset (20 documents; 2237 text lines)

Figure 10.12: Plot against one-to-one segmentation accuracy of the ridge-based text line finding method for different values of its free parameters on (a) Arabic documents and (b) Urdu documents.

10.5.2 Text Line Extraction Accuracy

The performance evaluation metrics for text line detection accuracy are defined in [SB11], where a text line is said to be correctly detected if it does not fall into any of the following types of errors: over-segmentation, under-segmentation, missed text lines, and false-alarms. From these metrics, one-to-one correctly detected text lines accuracy ($P_{o2o}\%$) is used here. For ridge-based text line finding method on Arabic dataset, a performance gain, from 93.89% to 96.02%, is achieved after text and non-text segmentation. Figure 10.12 shows the one-to-one text line finding accuracy of the ridge-based text line finding algorithm for different values of its free parameters for both Arabic documents (after text and non-text segmentation) and Urdu documents datasets. The relative flatness of the curves in Figure 10.12 indicates that the ridge-based text line detection method is reasonably stable with respect to its free parameters. The performance evaluation results on both Arabic and Urdu datasets of ridge-based, adapted RAST [SHKB06] and X-Y cut [NSV92] text line finding methods, with optimized values of their free parameters, are shown in Figure 10.13(a). The ridge-based method has achieved above 96% text line finding accuracy for Arabic dataset and above 92% for Urdu dataset, which are better than the performance of the adapted RAST [SHKB06] and the X-Y cut [NSV92] methods on these datasets. The X-Y cut method usually fails due to small inter-line gaps and presence of multiple columns. Under these conditions, the RAST works better than the X-Y cut but gives errors for very small inter-line gaps and page curl. The ridge-based



(a) text lines accuracy

(b) reading order accuracy

Figure 10.13: (a) Performance evaluation results of one-to-one text line extraction accuracy of the X-Y cut [NSV92], the adapted RAST [SHKB06], and the ridge-based text line finding methods on Arabic and Urdu datasets. (b) Performance evaluation results of reading order determination of topological sorting based [SHKB06], the dummy and the reverse reading order determination methods.

method performs better than both the X-Y cut and RAST under small inter-line gaps, multiple columns and page curl, but fails for very small inter-line gaps.

10.5.3 Reading Order Accuracy

A reading order determination algorithm heavily depends on text line detection accuracy. Here, an edit distance based reading order performance evaluation strategy is applied, such that edit distance is calculated between a detected and the corresponding ground-truth reading orders. Here, a dummy and a reverse reading order determination methods are also defined for comparison with the topological sorting based reading order determination method [SHKB06]. The dummy method simply returns the sorted order of text lines with respect to their baseline positions. The reverse-order method returns a complete reverse reading order with respect to a given ground-truth information. For simple document layouts, dummy method gives better reading order than reverse-order. However, for complex document layouts (like Urdu poetry as shown in Figure 10.11), both give bad result. The performance evaluation results of topological sorting based [SHKB06], dummy, and reverse reading order determination methods are shown in Figure 10.13(b). Dummy method performs better than reverse-order method for Arabic dataset, but for Urdu dataset both give almost same error, which is also comparatively larger than the errors for Arabic dataset. This also demonstrates that, layouts in Urdu dataset are more challenging than Arabic dataset with respect to reading order. The topological sorting based reading order determination method performs better than both dummy and reverse-order methods. It gives an incorrect reading order if two text lines from different text columns are merged, because in such a case they are interpreted as a separator. It gives larger error for Urdu dataset than Arabic dataset, because it cannot handle Urdu poetry written in two column format or other likewise layouts, as it is misinterpreted as a two-column text.

10.6 Conclusion

In this chapter, a high performance layout analysis system is presented for machine printed, scanned Arabic and Urdu document images, which are composed of a variety of single- and multi-column layouts. The layout analysis system is composed of a suitable combination of well-established and robust text and non-text segmentation, text line extraction, and reading order determination techniques that have proven to be robust against different types of document image degradations. The main contribution of this chapter is to show the effectiveness of these techniques on a variety of Arabic script document images. The presented layout analysis system is evaluated on 25 Arabic and 20 Urdu documents, which are composed of a variety of layouts as shown in Figure 10.11. For text and non-text segmentation, multiresolution morphology based method [BSB11c] is used. Above 99% text and non-text segmentation accuracy is achieved on Arabic dataset. For text line extraction, ridge-based method is used (Chapter 3). For ridge-based method, above 96% text line detection accuracy is archived for Arabic dataset and above 92% for Urdu dataset, which are better than the performance of both the X-Y cut [NSV92] and adapted RAST [SHKB06] based text line detection methods on these datasets. For determining the reading order of extracted text lines, topological sorting based reading order determination method [SHKB06] is used. A better reading order accuracy is achieved as compared to the dummy and the reverse reading order determination methods. Altogether, the layout analysis system showed good performance for text and non-text segmentation, text line extraction, and reading order determination on a variety of Arabic and Urdu document images, and it can be used for large scale Arabic and Urdu documents digitization processes.

Chapter 11

Conclusions and Future Work

The work presented in this thesis addresses the problem of generic layout analysis of a diverse collection of document images, which is considered as an elusive goal so far, still beyond the reach of the state-of-the-art in the field [NJ07,LSZT07,KB06]. One of the main goals of this thesis was to develop generic layout analysis methods that can be equally applied to a large variety of document images including scanned and camera-captured documents, binary and grayscale documents, typed-text and handwritten documents, historical and contemporary documents, and documents containing different scripts. Experimental results in this thesis has shown that the goal of generic layout analysis is successfully achieved to a great extent. Moreover, this thesis has also shown some important document image analysis applications of the presented generic layout analysis methods. The work presented in this thesis has already caught the attention of the community which is evident by over 70 external citations. This thesis has made several key contributions in the field of generic layout analysis, the most important of which are outlined here.

This thesis has presented a generic text line extraction method that can be equally used for a large variety of document image categories with respect to writing styles (typed-text or handwritten), digitization methods (scanner or camera), intensity values (binary and grayscale), scripts (like Latin, Chinese, and Arabic) and text lines structures (straight, skewed, curled and freestyle handwritten text lines). The generic text-line extraction method is based on two standard computer vision algorithms: matched filtering and ridge detection. For the performance evaluation of the presented text line extraction method, different standard and publicly available datasets have been selected that belong to various categories of document images including typed-text and handwritten document images, grayscale camera-captured document images, and historical document images with a large variety of scripts. Experimental results have shown that the presented generic text line extraction method achieves significantly better text line detection accuracy than aggregate accuracy of the best performing domain-specific state-of-the-art methods. To the best of the author's knowledge, it is the first general-purpose text line extraction method that can be equally used for a diverse collection of document images.

This thesis has also presented a novel curled text line extraction method, referred to as the coupled snakelets model, that is based on the state-of-the-art active contour (snake) image segmentation model. Unlike existing curled text-line extraction approaches, the coupled snakelets method extracts text lines and estimates their x-line and baseline pairs simultaneously; resulting in better segmentation with more accurate estimation of x-lines and baseline than competing approaches. The DFKI-I dataset [SB07], which is a publicly available collection of camera-captured warped, Latin script document images, is used for performance evaluation of the coupled snakelets method and comparison with the competing state-of-the-art curled text-line extraction approaches. Experimental results have shown that the coupled snakelets algorithm achieves the highest one-to-one text line extraction accuracy. It also yields the lowest oversegmentation and missed text lines errors, and a small number of undersegmentation errors. In addition to Latin script, the coupled snakelets method can also be applied to different scripts like Chinese, Devanagari, and Arabic. with specialized script-specific adaptations.

The modifications to the original Bloomberg's text and halftone segmentation algorithm [Blo91] have been presented in this thesis for transforming it into a general purpose text and non-text image segmentation method, where non-text components can be halftones, line drawings, maps, and graphics. The original Bloomberg's method and the modified method have been evaluated on standard datasets including the UW-III dataset, the ICDAR 2009 page segmentation competition test dataset and the circuit diagrams dataset. These datasets represent a collection of varied Latin script document images that contain texts, halftones, line drawings, maps and graphics elements. Experiments have shown that the modifications result in significant improvements over the original Bloomberg's page segmentation algorithm. The modified method, like the original one, is a domain-independent page segmentation algorithm and it can be directly applied to a diverse collection of document images.

A novel page segmentation method using discriminative learning over connected components has been also presented in this thesis for segmenting a document image into text and non-text regions. The discriminative learning based method has been evaluated on the standard datasets (UW-III, ICDAR 2009, and circuit diagrams) that contain a large variety of text and non-text elements. It has achieved a better segmentation accuracy as compared to the state-of-the-art Bloomberg's page segmentation method [Blo91], which is a text and halftone segmentation approach. In addition to Latin script, this method can also be adapted for other scripts as well as for other specialized layout analysis tasks such as digit and non-digit segmentation [HBSB12], orientation detection [RBSB09], and body-text and side-note segmentation [BAESB12].

Finally, this thesis has presented two important document image analysis applications using the generic layout analysis methods: the ridge-based text line extraction method and the improved multi-resolution morphology based text and non-text segmentation method. First, a preprocessing of hand-held camera-captured document images has been presented for grayscale warped document images, which includes binarization, page frame detection, and monocular dewarping steps. The presented binarization approach has been able to handle grayscale degradations like non-uniform shading and blurring that are common in hand-held camera-captured document images. The presented page frame detection method has been used to remove border noise and to identify the actual content area of document images, and the presented monocular dewarping method has been developed to rectify warped document images. Experimental results have shown that all the proposed modifications significantly improve the preprocessing steps. Second, a high performance layout analysis system has been presented for typed-text, scanned Arabic and Urdu document images, which are composed of a variety of single- and multi-column layouts. The presented layout analysis system is robust against different types of document image degradations. It has shown better performance for text and non-text segmentation, text line extraction, and reading order determination on a variety of Arabic and Urdu document images as compared to the state-of-the-art methods. The presented layout analysis system can be used for large scale Arabic and Urdu documents' digitization processes. These applications demonstrate that the two layout analysis methods are generic and can be applied easily to a large collection of diverse document images.

Overall, this thesis has presented high performance generic layout analysis methods that can be equally applied to a collection of document images such as books, magazines, newspaper, handwritten and historical document images. The work presented in this thesis has been used for removing geometric and perspective distortions of warped camera-captured document images so that dewarped document images can be directly processed by existing scanner based OCR software, and solving the layout analysis problems of complex script (like Urdu, Telugu, and Kannada) document images and degraded handwritten and historical document images.

Moreover, the presented methods can potentially be adapted for other categories of document images such as structured text document images (like forms, invoices, and envelop) and mostly graphics document images (like maps, engineering drawings, and music sheets) as well as other types of images like medical images and satellite images for performing various image analysis tasks. The ridge-based generic text-line extraction method can be applied for text line extraction in unstructured invoice images, line detection in map and engineering drawing images, and music score segmentation in music sheet images. The coupled snakelets based curled text line segmentation method can be adapted for road segmentation in satellite images, structural line (like nerves and bones) segmentation in medical images, and edge detection and recovery in natural images. The multi-resolution morphology based text and non-text segmentation method can be adapted for text and non-text separation in engineering drawing images like floor plans and circuit diagrams. The discriminative learning based segmentation method can be trained for separating multiple objects in document images such as text and line drawing separation in map images.

Bibliography

[AKR07] K. Arvind, Jayant Kumar, and A. Ramakrishnan. Entropy based skew correction of document images. In *Pattern Recognition and Machine Intelligence*, volume 4815 of *Lecture Notes in Computer Science*, pages 495–502. 2007.

[AL04] B. T. Avila and R. D. Lins. Efficient removal of noisy borders from monochromatic documents. In *Int. Conf. on Image Analysis and Recognition*, pages 249–256, Porto, Portugal, Sep. 2004.

[AM11] H. E. Abed and V. Mrgner. ICDAR 2009-Arabic handwriting recognition competition. *International Journal on Document Analysis and Recognition*, 14:3–13, 2011.

[APBP09] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. ICDAR 2009 page segmentation competition. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1370 –1374, july 2009.

[ARQA07] Mohammed Al-Rawi, Munib Qutaishat, and Mohammed Arrar. An improved matched filter for blood vessel detection of digital retinal images. *Comput. Biol. Med.*, 37:262–267, February 2007.

[ASES11] A. Asi, R. Saabni, and J. El-Sana. Text line segmentation for gray scale historical document images. In *International Workshop on Historical Document Imaging and Processing, HIP'11*, Beijing, China, 2011.

[ASS07] Manivannan Arivazhagan, Harish Srinivasan, and Sargur Srihari. A statistical approach to line segmentation in handwritten documents. volume 6500. SPIE, 2007.

[AYV01] N. Arica and F.T. Yarman-Vural. An overview of character recognition focused on off-line handwriting. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 31(2):216 –233, may 2001.

[BAESB12] S. S. Bukhari, A. Asi, J. El-Sana, and T. M. Breuel. Layout analysis for Arabic historical document images using machine learning. In *13th International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, 2012.

[Bai94] H. S. Baird. Background structure in document images. In H. Bunke, P. Wang, and H. S. Baird, editors, *Document Image Analysis*, pages 17–34. World Scientific, Singapore, 1994.

[BB01] David Bainbridge and Tim Bell. The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95 – 121, 2001.

[BB08] Roman Bertolami and Horst Bunke. Hidden markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, 41(11):3452 – 3460, 2008.

[BBS09] Syed Saqib Bukhari, Thomas M. Breuel, and Faisal Shafait. Textline information extraction from grayscale camera-captured document images. In *Proceedings of the 16th IEEE international conference on Image processing*, ICIP'09, pages 1993–1996, 2009.

[Ber86] J. Bernsen. Dynamic thresholding of gray level images. In *Proceedings 8th International Conference on Pattern Recognition*, pages 1251–1255, 1986.

[BJF90] H. S. Baird, S. E. Jones, and S. J. Fortune. Image segmentation by shape-directed covers. In *Proceedings 10th International Conference on Pattern Recognition*, pages 820–825, jun 1990.

[BKMA10] M. Benjelil, S. Kanoun, R. Mullot, and A. Alimi. Complex documents images segmentation based on steerable pyramid features. *International Journal on Document Analysis and Recognition*, 13:209–228, 2010.

[Blo] D. S. Bloomberg. Leptonica: An open source c library for efficient image processing and image analysis operations. http://code.google.com/p/leptonica/.

[Blo91] D. S. Bloomberg. Multiresolution morphological approach to document image analysis. In *Proceedings 1st International Conference on Document Analysis and Recognition*, pages 963–971, St. Malo, France, 1991.

[BNP06] E. Badekas, N. Nikolaou, and N. Papamarkos. Text binarization in color documents. *International Journal of Imaging Systems and Technology*, 16(6):262–274, 2006.

[BP05] E. Badekas and N. Papamarkos. Automatic evaluation of document binarization results. In *Proceedings 10th Iberoamerican Congress on Pattern Recognition*, pages 1005–1014, Havana, Cuba, 2005.

[Bre01] T. M. Breuel. Segmentation of handprinted letter strings using a dynamic programming algorithm. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 821–826, 2001.

[Bre02a] T. M. Breuel. Robust least square baseline finding using a branch and bound algorithm. In *9th Conference on Document Recognition and Retrieval*, pages 20–27, 2002.

[Bre02b] T. M. Breuel. Two geometric algorithms for layout analysis. In *Proceedings of the 5th International Workshop on Document Analysis Systems*, pages 188–199, London, UK, 2002. Springer-Verlag.

[Bre03] T. M. Breuel. High performance document layout analysis. In *Symposium on Document Image Understanding Technology*, Greenbelt, MD, USA, April 2003.

[Bre05] T.M. Breuel. The future of document imaging in the era of electronic documents. In *Int. Workshop on Document Analysis*, Kolkata, India, Mar. 2005.

[BS04] M. S. Brown and W. B. Seales. Image restoration of arbitrarily warped documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1295–1306, 2004.

[BS10] T. M. Breuel and F. Shafait. Automlp: Simple, effective, fully automated learning rate and size adjustment. In *The Learning Workshop, Snowbird, Utah*, 2010.

[BSB08] S. S. Bukhari, F. Shafait, and T. M. Breuel. Segmentation of curled textlines using active contours. In *Proceedings 8th IAPR Workshop on Document Analysis Systems*, pages 270–277, Nara, Japan, 2008.

[BSB09a] S. S. Bukhari, F. Shafait, and T. M. Breuel. Dewarping of document images using coupled-snakes. In *Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition*, pages 34–41, Barcelona, Spain, 2009.

[BSB09b] S. S. Bukhari, F. Shafait, and T. M. Breuel. Ridges based curled textline region detection from grayscale camera-captured document images. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*, CAIP '09, pages 173–180, Berlin, Heidelberg, 2009. Springer-Verlag.

[BSB09c] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Adaptive binarization of unconstrained hand-held camera-captured document images. *Journal of Universal Computer Science*, 15(18):3343–3363, dec 2009.

[BSB10a] S. S. Bukhari, F. Shafait, and T. M. Breuel. Document image segmentation using discriminative learning over connected components. In *Proceedings 9th IAPR Workshop on Document Analysis Systems*, pages 183–190, Boston, Massachusetts, USA, 2010.

[BSB10b] S. S. Bukhari, F. Shafait, and T. M. Breuel. Performance evaluation of curled textline segmentation algorithms on CBDAR 2007 dewarping contest dataset. In *Int. Conf. on Image Processing, 2010 17th*, pages 2161 –2164, Cairo, Egypt, sept. 2010.

[BSB11a] S. S. Bukhari, F. Shafait, and T. M. Breuel. Border noise removal of camera-captured document images using page frame detection. In *Fourth International Workshop on Camera-Based Document Analysis and Recognition*, Beijing, China, 2011.

[BSB11b] S. S. Bukhari, F. Shafait, and T. M. Breuel. High performance layout analysis of Arabic and Urdu document images. In *Proceedings 11th International Conference on Document Analysis and Recognition*, pages 1275–1279, Beijing, China, 2011.

[BSB11c] S. S. Bukhari, F. Shafait, and T. M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In *Proceedings SPIE Document Recognition and Retrieval XVIII*, San Jose, CA, USA, Jan. 2011.

[BSB11d] S. S. Bukhari, F. Shafait, and T. M. Breuel. The iupr dataset of camera-captured document images. In *Fourth International Workshop on Camera-Based Document Analysis and Recognition*, Beijing, China, 2011.

[BSB11e] Syed Saqib Bukhari, Faisal Shafait, and Thomas Breuel. Coupled snakelets for curled text-line segmentation from warped document images. *International Journal on Document Analysis and Recognition*, pages 1–21, October 2011.

[BSB12a] S. S. Bukhari, F. Shafait, and T. M. Breuel. Border noise removal of camera-captured document images using page frame detection. In Masakazu Iwamura and Faisal Shafait, editors, *4th International Workshop, CBDAR 2011 Beijing, China, September 22, 2011 Revised Selected Papers*, volume 7139 of *Lecture Notes in Computer Science, Image Processing, Computer Vision, Pattern Recognition, and Graphics*, pages 126–137. Springer Berlin / Heidelberg, 2012.

[BSB12b] S. S. Bukhari, F. Shafait, and T. M. Breuel. Layout analysis of Arabic script documents. In *Guide to OCR for Arabic Scripts*. Springer-Verlag, 2012.

[BT06] M. S. Brown and Y. C. Tsoi. Geometric and shading correction for images of printed materials using boundary. *Image Processing, IEEE Transactions on*, 15(6):1544 –1554, june 2006.

[buk] Arabic, Urdu, and Fraktur datasets online.

[BYHKD09] Itay Bar-Yosef, Nate Hagbi, Klara Kedem, and Itshak Dinstein. Line segmentation for degraded handwritten historical documents. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 1161–1165, Washington, DC, USA, 2009. IEEE Computer Society.

[BZA$^+$10] W. Boussellaa, A. Zahour, H. E. Abed, A. Benabdelhafid, and A. M. Alimi. Unsupervised block covering analysis for text-line segmentation of

Arabic ancient handwritten document images. In *Proceedings 20th International Conference on Pattern Recognition*, pages 1929–1932, Istanbul, Turkey, 2010.

[CA09] P. Chias and T. Abad. Geolocating and Georeferencing: GIS tools for ancient maps visualisation. In *Information Visualisation, 2009 13th International Conference*, pages 529 –538, july 2009.

[CCK+89] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two dimensional matched lters. *IEEE Trans. Med. Imaging*, 8:263–269, 1989.

[CCMM98a] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical report, IRST, Trento, Italy, 1998.

[CCMM98b] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena. Layout analysis techniques for document image understanding: a review. In *available from http://citeseer.nj.nec.com/, IRST, Trento, Italy, Tech. Rep. 9703-09*, 1998.

[CDL03] H. Cao, X. Ding, and C. Liu. Rectifying the bound document image captured by the camera: a model based approach. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 71–75, Edinburgh, Scotland, 2003.

[CLLT02] L. Cinque, S. Levialdi, L. Lombardi, and S. Tanimoto. Segmentation of page images having artifacts of photocopying and scanning. *Pattern Recognition*, 35(5):1167–1177, 2002.

[CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*, chapter 23, pages 485–488. MIT Press, Cambridge, MA, 1990.

[CM03] P. Clark and M. Mirmehdi. Rectifying perspective views of text in 3D scenes using vanishing points. *Pattern Recognition*, 36(11):2673–2686, 2003.

[CW01] Z. Chi and K. W. Wong. A two-stage binarization approach for document images. Proc. Int. Symp. Intelligent Multimedia, Video and Speech Processing (ISIMP'01), pages 275–278, 2001.

[Dam99] James Damon. Properties of ridges and cores for two-dimensional images. *Journal of Mathematical Imaging and Vision*, 10:163–174, 1999. 10.1023/A:1008379107611.

[DLL03] David Doermann, Jian Liang, and Huiping Li. Progress in camera-based document image analysis. In *Proc. ICDAR03*, pages 606–616, 2003.

[EGM$^+$94] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *J. Math. Imaging Vis.*, 4(4):353–373, 1994.

[FK88] Lloyd A. Fletcher and Rangachar Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions Pattern Analysis Machine Intelligence*, 10(6):910–918, 1988.

[FWL02] K. C. Fan, Y. K. Wang, and T. R. Lay. Marginal noise removal of document images. *Pattern Recognition*, 35(11):2593–2611, 2002.

[FWL$^+$07] B. Fu, M. Wu, R. Li, W. Li, and Z. Xu. A model-based book dewarping method using text line detection. In *2nd International Workshop on Camera Based Document Analysis and Recognition*, September 2007.

[FZT10] H. Fan, L. Zhu, and Y. Tang. Skew detection in document images based on rectangular active contour. *International Journal on Document Analysis and Recognition*, 13(4):261–269, 2010.

[GA99] H. Goto and H. Aso. Extracting curved text lines using local linearity of the text line. *International Journal on Document Analysis and Recognition*, 2:111–119, 1999.

[GAS07] B. Gatos, A. Antonacopoulos, and N. Stamatopoulos. Handwriting segmentation contest. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 1284–1288, Curitiba, Brazil, 2007.

[GHHP97] I. Guyon, R. M. Haralick, J. J. Hull, and I. T. Phillips. Data sets for OCR and document image understanding research. In H. Bunke and P. Wang, editors, *Handbook of character recognition and document image analysis*, pages 779–799. World Scientific, Singapore, 1997.

[Gla56] M. H. Glauberman. Character recognition for business machines. *Electronics*, 29:132–136, 1956.

[GN88] L. O. Gorman and J. V. Nickerson. Matched filter design for fingerprint image enhancement. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-88.*, pages 916–919, New York, NY , USA, 1988.

[GN97] S. R. Gunn and M. S. Nixon. A robust snake implementation; a dual active contour. *IEEE Trans. Pattern Anal. Mach. Intell*, 19(1):63–68, 1997.

[GN07] B. Gatos and K. Ntirogiannis. Restoration of arbitrarily warped document images based on text line and word detection. In *Proceedings 4th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pages 203–208, Innsbruck, Austria, 2007.

[GNP09] B. Gatos, K. Ntirogiannis, and I. Pratikakis. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009). In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1375 –1382, july 2009.

[GPN07] B. Gatos, I. Pratikakis, and K. Ntirogiannis. Segmentation based recovery of arbitrarily warped document images. In *Proceedings 9th International Conference on Document Analysis and Recognition*, pages 989–993, Curitiba, Brazi, 2007.

[GPP06] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3):317–327, 2006.

[GSW03] Jan-Mark Geusebroek, Arnold W. M. Smeulders, and Joost Van De Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003.

[HA02] S. A. Husain and S. H. Amin. A multi-tier holistic approach for Urdu Nastaliq recognition. In *IEEE International Multi-topic Conference*, Karachi, Pakistan, Dec. 2002.

[har] http://ocp.hul.harvard.edu/ihp/.

[HBSB12] A. Ul Hasan, S. S. Bukhari, F. Shafait, and T. M. Breuel. OCR-free table of contents detection in urdu books. In *Proceedings 10th IAPR Workshop on Document Analysis Systems*, pages 404–408, Gold Coast, Australia, 2012.

[HH03] B. Hohnhaeuser and G. Hommel. 3D pose estimation using coupled snakes. *Journal of WSCG*, 12(1-3):1213–6972, Feb 2003.

[HKG00] A. Hoover, V. Kouzntesova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched lter responses. *IEEE Trans. Med. Imaging*, 19:263–269, 2000.

[HLW03] C. T. Hsieh, E. Lai, and Y. C. Wang. An effective algorithm for fingerprint image enhancement based on wavelet transform. *Pattern Recognition*, 36(2):302–312, 2003.

[Hor70] B. K. P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. *PhD Thesis, MIT*, 1970.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[IJM] http://www.ijma3.org/Templates/InsideTemplate.aspx?PostingId=427.

[JFMF05] M. S. Jelodar, M. J. Fadaeieslam, N. Mozayani, and M. Fazeli. A Persian OCR system using morphological operators. *Proceedings of World Academy of Science, Engineering and Technology*, 4:137–140, 2005.

[KAAKD10] J. Kumar, W. Abd-Almageed, L. Kang, and D. Doermann. Handwritten Arabic text line segmentation using affinity propagation. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 135–142, New York, NY, USA, 2010.

[kan] http://en.wikipedia.org/wiki/Kannada_alphabet.

[KB06] D. J. Kennard and W. A. Barrett. Separating lines of text in free-form handwritten historical documents. In *2nd Int. Conf. on Document Image Analysis for Libraries.*, pages 12 – 23, Los Alamitos, CA, USA, april 2006.

[KD05] Chih-Hong Kao and Hon-Son Don. Skew detection of document images using line structural information. In *International Conference on Information Technology and Applications*, volume 1, pages 704–715. Los Alamitos, CA, USA, 2005.

[Kho02] M. S. Khorsheed. Off-Line Arabic character recognition - a review. *Pattern Analysis & Applications*, 5(1):31–45, 2002.

[Kim04] In-Jung Kim. Multi-window binarization of camera image for document recognition. In *Proceedings 9th International Workshop on Frontiers in Handwriting Recognition*, pages 323–327, Washington, DC, USA, 2004.

[KKJ07] K. S. Kumar, S. Kumar, and C. Jawahar. On segmentation of documents in complex scripts. In *9th Int. Conf. on Document Analysis and Recognition*, pages 1243–1247, Washington, DC, USA, 2007.

[KSB07] D. Keysers, F. Shafait, and T. M. Breuel. Document image zone classification- a simple high-performance approach. In *Proc. 2nd Int. Conf. Computer Vision Theory and Applications*, pages 44–51, Barcelona, Spain, Mar. 2007.

[KSI98] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision Image Understanding*, 70:370–382, June 1998.

[KSK06] C.V. Jawahar K.S. Sesh Kumar, A. M. Namboodiri. Learning segmentation of documents with complex scripts. In *Fifth Indian Conference on Computer Vision, Graphics and Image Processing, Madurai, India, LNCS 4338*, pages 749–760, 2006.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):1162–1173, 1988.

[LCK05] S. J. Lu, B. M. Chen, and C. C. Ko. Perspective rectification of document images using fuzzy set and morphological operations. *Image and Vision Computing*, 23:541–553, 2005.

[LDD05] J. Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *Proceedings 18th International Conference on Computer Vision and Pattern Recognition*, pages 338–345, San Diego, CA, USA, 2005.

[LDD08] Jian Liang, Daniel DeMenthon, and David Doermann. Geometric rectification of camera-captured document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:591–605, 2008.

[LDL05] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *International Journal of Document Analysis and Recognition*, 7(2-3):84–104, 2005.

[LGPH08] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line detection in handwritten documents. *Journal Pattern Recogn.*, 41(12):3758–3772, 2008.

[LGPH09] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line and word segmentation of handwritten documents. *Pattern Recognition*, 42(12):3169 – 3183, 2009.

[lib] http://www.loc.gov/index.html.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[LSZT07] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents: a survey. *Int. Journal on Document Analysis and Recognition*, 9:123–138, 2007.

[LT02] P. K. Loo and C. L. Tan. Word and sentence extraction using irregular pyramid. In *Document Analysis Systems V*, volume 2423 of *Lecture Notes in Computer Science*, pages 307–318. Springer Berlin / Heidelberg, 2002.

[LT05] S. J. Lu and C. L. Tan. Camera document restoration for OCR. In *Proceedings of First International Workshop on Camera-Based Document Analysis and Recognition*, pages 17–24, Seoul, Korea, 2005.

[LT06a] S. Lu and C. L. Tan. The restoration of camera documents through image segmentation. In *Proceedings 7th IAPR workshop on Document Analysis Systems*, pages 484–495, 2006.

[LT06b] S. Lu and C.L. Tan. Document flattening through grid modeling and regularization. In *Proceedings 18th International Conference on Pattern Recognition*, pages 971–974, 2006.

[LT07] S. Lu and C. L. Tan. Thresholding of badly illuminated document images through photometric correction. In *Proceedings 2007 ACM symposium on Document engineering*, pages 3–8, Winnipeg, Manitoba, Canada, 2007.

[LTW96] D. X. Le, G. R. Thoma, and H. Wechsler. Automated borders detection and adaptive segmentation for binary document images. In *13th Int. Conf. on Pattern Recognition*, pages 737–741, Vienna, Austria, Aug. 1996.

[LW06] C.H. Lampert and O. Wirjadi. An optimal nonorthogonal separation of the anisotropic gaussian convolution filter. *IEEE Transactions on Image Processing*, 15(11):3501–3513, 2006.

[LZDJ08] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger. Script-independent text line segmentation in freestyle handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1313 –1329, aug. 2008.

[MB01] U.-V. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In *Proceedings 6th International Conference on Document Analysis and Recognition*, pages 159 –163, Los Alamitos, CA, USA, 2001.

[MB08] M. A. Moll and H. S. Baird. Segmentation-based retrieval of document images from diverse collections. In *Document Recognition and Retrieval XV, Proceeding of the SPIE*, volume 6815, pages 68150L–68150L, 2008.

[MBA08] M. A. Moll, H. S. Baird, and C. An. Truthing for pixel-accurate segmentation. In *Document Analysis Systems, the Eighth IAPR Int. Workshop*, pages 379–385, Sep. 2008.

[MGS05] S. Marinai, M. Gori, and G. Soda. Artificial neural networks for document analysis and recognition. volume 27(1) of *IEEE Transaction on Pattren Analysis and Machine Intelligence*, Jan. 2005.

[MM07] A. Masalovitch and L. Mestetskiy. Usage of continuous skeletal image representation for document images de-warping. In *Proceedings 2nd International Workshop on Camera-Based Document Analysis and Recognition*, pages 45–52, Curitiba, Brazil, 2007.

[MMB01] U.-V. Marti, R. Messerli, and H. Bunke. Writer identification using text line based features. In *Proceedings 6th International Conference on Document Analysis and Recognition*, pages 101 –105, Los Alamitos, CA, USA, 2001.

[MR05] R. Manmatha and Jamie L. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1212–1225, August 2005.

[MSY92] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.

[Nag00] G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38 –62, jan 2000.

[NBO09] Y. Navon, E. Barkan, and B. Ophir. A generic form processing approach for large variant templates. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 311 –315, july 2009.

[Nib86] W. Niblack. *An Introduction to Image Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1986.

[NJ07] A. M. Namboodiri and A. Jain. Document structure and layout analysis. pages 29–48, London, UK, 2007. Springer-Verlag.

[NSV92] George Nagy, Sharad Seth, and Mahesh Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, 1992.

[OCDK04] Kazunori Okada, Dorin Comaniciu, Navneet Dalal, and Arun Krishnan. A robust algorithm for characterizing anisotropic local structures. In *in European Conference on Computer Vision*, volume 3021 of *Lecture Notes in Computer Science*, pages 549–561. 2004.

[ODP99] O. Okun, D. Doermann, and M. Pietikainen. Page segmentation and zone calssification: the state of art. In *Technical Report LAM-TR-036, CAR-TR-927, CS-TR-4079*, University of Maryland, College Park, Nov. 1999.

[O'G93] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162 –1173, nov. 1993.

[O'G94] L. O'Gorman. Binarization and multithresholding of document images using connectivity. *Graphical Model and Image Processing*, 56(6):494–506, Nov. 1994.

[OLT+10] D. M. Oliveira, R. D. Lins, G. Torreo, J. Fan, and M. Thielo. A new method for text-line segmentation for warped document. In *Proceedings of International Conference on Image Analysis and Recognition*, pages 398–408, Povoa de Varzim, Portugal, 2010.

[Ots79] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[PS03] U. Pal and A. Sarkar. Recognition of printed Urdu script. In *7th International Conference on Document Analysis and Recognition*, pages 1183–1187, Edinburgh, UK, Aug. 2003.

[RB95] N. Rondel and G. Breuel. Coorperation of multilayer perceptrons for the estimation of skew angle in text document images. Proc. Int'l Conf. Documnet Analysis and Recognition (ICDAR'95), pages 1141–1144, 1995.

[RBSB09] S. F. Rashid, S. S. Bukhari, F. Shafait, and T. M. Breuel. A discriminative learning approach for orientation detection of Urdu document images. In *13th IEEE International Multitopic Conference, INMIC '09,*, pages 1–5, Islamabad, Pakistan, 2009.

[Ril87a] M. D. Riley. Beyond quasi-stationarity: Designing time-frequency representations for speech signals. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 657 – 660, Apr. 1987.

[Ril87b] Michael D. Riley. Time-frequency representation for speech signals. *PhD Thesis, MIT*, 1987.

[RSB09] Y. Rangoni, F. Shafait, and T. M. Breuel. OCR based thresholding. In *Proceedings IAPR Conference on Machin Vision Applications*, Yokohama, Japan, 2009.

[SAJ10] Prawit Sutthiwichaiporn, Vutipong Areekul, and Suksan Jirachaweng. Iterative fingerprint enhancement with matched filtering and quality diffusion in spatial-frequency domain. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 1257–1260, Washington, DC, USA, 2010. IEEE Computer Society.

[SB07] F. Shafait and T. M. Breuel. Document image dewarping contest. In *2nd International Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, Sep 2007.

[SB09] F. Shafait and T. M. Breuel. A simple and effective approach for border noise removal from document images. In *13th IEEE Int. Multi-topic Conference*, Islamabad, Pakistan, Dec 2009.

[SB11] F. Shafait and T. M. Breuel. The effect of border noise on the performance of projection based page segmentation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):846–851, 2011.

[SBKB08] F. Shafait, J. V. Beusekom, D. Keysers, and T. M. Breuel. Structural mixtures for statistical layout analysis. In *Proceedings 8th International Workshop on Document Analysis Systems*, pages 415–422, Nara, Japan, 2008.

[SGG10] N. Stamatopoulos, B. Gatos, and T. Georgiou. Page frame detection for double page document images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 401– 408, Boston, MA, USA, 2010.

[SGK07] N. Stamatopoulos, B. Gatos, and A. Kesidis. Automatic borders detection of camera document images. In *Proceedings of Second International Workshop on Camera-Based Document Analysis and Recognition*, pages 71–78, Curitiba, Brazil, 2007.

[SGP09] N. Stamatopoulos, B. Gatos, and I. Pratikakis. A methodology for document image dewarping techniques performance evaluation. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 956–960, Barcelona, Spain, 2009.

[SGPP08] N. Stamatopoulos, B. Gatos, I. Pratikakis, and S. J. Perantonis. A two-step dewarping of camera document images. In *Proceedings 8th IAPR Workshop on Document Analysis Systems*, pages 209–216, Nara, Japan, 2008.

[SHKB06] F. Shafait, A. U. Hasan, D. Keysers, and T. M. Breuel. Layout analysis of Urdu document images. In *IEEE Int. Multitopic Conference, INMIC '06*, pages 293–298, Islamabad, Pakistan, 2006.

[SKB08a] F. Shafait, D. Keysers, and T. M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proc. SPIE Document Recognition and Retrieval XV*, pages 101–106, San Jose, CA, USA, Jan. 2008.

[SKB08b] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):941 –954, june 2008.

[SKPB00] K. Sobottka, H. Kronenberg, T. Perroud, and H. Bunke. Text extraction from colored book and journal covers. *International Journal on Document Analysis and Recognition*, 2(4):163–176, June 2000.

[SP00] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[SPC97] C. Strouthopoulos, N. Papamarkos, and C. Chamzas. Identification of text-only areas in mixed-type documents. *Engineering Applications of Artificial Intelligence*, 10(4):387–401, 1997.

[SS04] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, 2004.

[SSG05] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. Text extraction from gray scale historical document images using adaptive local connectivity map. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, pages 794–798, Washington, DC, USA, 2005. IEEE Computer Society.

[SSMSSS06] S. Shirali-Shahreza, M. T. Manzuri-Shalmani, and M. H. Shirali-Shahreza. Page segmentation of Persian/Arabic printed text using ink spread effect. In *SICE-ICASE International Joint Conference*, pages 259–262, Busan, Korea, Oct. 2006.

[SvBKB08] Faisal Shafait, Joost van Beusekom, Daniel Keysers, and Thomas Breuel. Document cleanup using page frame detection. *International Journal on Document Analysis and Recognition*, 11:81–96, 2008.

[tel] http://en.wikipedia.org/wiki/Telugu_alphabet.

[TL02] C.M. Tsai and H.J. Lee. Binarization of color document images via luminance and saturation color features. *IEEE Transactions on Image Processing*, 11(4):434–451, April 2002.

[Tom98] Karl Tombre. Analysis of engineering drawings: State of the art and challenges. In Karl Tombre and Atul Chhabra, editors, *Graphics Recognition Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 257–264. Springer Berlin / Heidelberg, 1998.

[TZND99] M. J. Taylor, A. Zappala, W. M. Newman, and C. R. Dance. Documents through cameras. In *Image and Vision Computing 17*, volume 11, pages 831–844, September 1999.

[TZZX06] C. L. Tan, L. Zhang, Z. Zhang, and T. Xia. Restoring warped document images through 3D shape modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):195–208, 2006.

[ULB05] A. Ulges, C. H. Lampert, and T. M. Breuel. Document image dewarping using robust estimation of curled text lines. In *Proceedings 8th International Conference on Document Analysis and Recognition*, pages 1001–1005, Seoul, Korea, 2005.

[urd] http://en.wikipedia.org/wiki/Urdu_alphabet.

[vBSB10] Joost van Beusekom, Faisal Shafait, and T. M. Breuel. Combined orientation and skew detection using geometric text-line modeling. *International Journal of Document Analysis and Recognition*, 13(2):79–92, 2010.

[Vin02] Alessandro Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1433 – 1446, 2002.

[WCW82] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, 1982.

[Wik] http://en.wikipedia.org/wiki/Nastaliq_script.

[Won08] Ch. S. Won. Image extraction in digital documents. In *Journal of Electronic Imaging*, volume 17, page 033016, 2008.

[WPH06] Y. Wang, I. Phillips, and R. Haralick. Document zone content classification and its performance evaluation. In *Pattern Recognition*, volume 39, pages 57–73, 2006.

[WR83] J. M. White and G. D. Rohrer. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM Journal of Research and Development*, 27(4):400–411, July 1983.

[WRS$^+$09] Q. Wang, O. Ronneberger, E. Schulze, R. Baumeister, and H. Burkhardt. Using lateral coupled snakes for modeling the contours of worms. *Pattern Recognition, Lecture Notes in Computer Science*, 5748:542–551, 2009.

[WWC06] F. M. Wahl, K. Y. Wong, and R. G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20:375–390, 2006.

[WZY07] Liu Wenyin, Wan Zhang, and Luo Yan. An interactive example-driven approach to graphics recognition in engineering drawings. *International Journal on Document Analysis and Recognition*, 9:13–29, 2007. 10.1007/s10032-006-0025-x.

[XP98] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. In *IEEE Transaction of Image Processing*, volume 7, pages 359–369, 1998.

[YJ96] Bin Yu and Anil K. Jain. A generic system for form dropout. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 18:1127–1134, November 1996.

[YKJ10] Huijuan Yang, Alex C. Kot, and Xudong Jiang. Knowledge guided adaptive binarization for 2d barcode images captured by mobile phones. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1046 –1049, march 2010.

[ZT01] Z. Zhang and C. L. Tan. Recovery of distorted document images from bound volumes. In *Proceedings International Conference on Document Analysis and Recognition*, pages 429–433, Los Alamitos, CA, USA, 2001.

[ZT03] Z. Zhang and C. L. Tan. Correcting document image warping based on regression of curved text lines. In *Proceedings 7th International Conference on*

Document Analysis and Recognition, pages 589–593, Edinburgh, Scotland, 2003.

[ZTMR01] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane. Arabic hand-written text-line extraction. *Document Analysis and Recognition, International Conference on*, pages 281–285, 2001.

Syed Saqib Bukhari

Image Understanding and Pattern Recognition Group
Technical University of Kaiserslautern, Germany
E-mail: bukhari@iupr.com
Web: https://sites.google.com/a/iupr.com/bukhari/
Linkedin: http://www.linkedin.com/in/syedsaqibbukhari

Research Interests

- Computer Vision, Image Analysis, Pattern Recognition, and their Applications
- Efficient and Reliable Algorithms for Document Image Analysis

Education

PhD in Computer System Engineering, April 2008 - November 2012

Technical University of Kaiserslautern, Kaiserslautern, Germany

- Thesis Topic: ***Generic Methods for Document Layout Analysis and Preprocessing***
- Thesis Advisor: *Prof. Dr. Thomas M. Breuel*

PhD-Admission Qualification Studies, April 2007 - April-2008

Technical University of Kaiserslautern, Kaiserslautern, Germany

Masters of Engineering in Computer Systems, Jan 2004 - July 2006

NED University of Engineering and Technology, Karachi, Pakistan

Bachelor of Engineering in Computer Systems, Jan 1999 - Feb 2003

NED University of Engineering and Technology, Karachi, Pakistan

Honors and Awards

- The best student paper award in International Conference on Frontiers in Handwriting Recognition (ICFHR), Bari, Italy, 2012.
- Selected for the "International Computer Vision Summer School, ICVSS'09", Sicily, Italy, 2009
- Outstanding performance in Masters of engineering in computer systems (Full GPA: 4.0/4.0), 2006
- Position in B.E. computer systems engineering (5th/76), 2003

Publications

I have published two journal papers, one book chapter, and twenty-four conference papers in well-know and prestigious journals, publishers, and conferences, respectively, and other papers are still under review.

Publications in the area of document image analysis:
International Journal of Document Analysis and Recognition, IJDAR
International Conference on Document Analysis and Recognition, ICDAR
Workshop on Document Analysis Systems, DAS
International Conference on Frontiers in Handwriting Recognition, ICFHR
Document Recognition and Retrieval, DRR
Workshop on Camera-Based Document Analysis and Recognition, CBDAR

Publications in the area of image analysis and pattern recognition:
Journal of Universal Computer Science, JUCS

International Conference on Pattern Recognition, ICPR
International Conference on Image Processing, ICIP
International Conference on Computer Analysis of Images and Patterns, CAIP

Reviewer in Journals and Conferences

I have been working as a reviewer for the following journals and conferences.

- Electronic Letters on Computer Vision and Image Analysis (ELCVIA)
- International Journal on Document Analysis and Recognition (IJDAR)
- International Conference on Document Analysis and Recognition (ICDAR)

Invited Talks

I have given invited talks about my research work at the folloiwng places.

- Intelligent Media Processing Lab at Osaka Prefecture University, Osaka, Japan, December 2011.
- Symposium on Document Image Analysis at Lahore University of Management Sciences (LUMS), Lahore, Pakistan, March 2012.

Internship

Completed internship at C & C Innovation Research Laboratories, NEC Corporation, Nara, Japan, in-between August 2011 to January 2012. Project: "Behavior Based Transport Guidance System", related to the area of Geographic Information System (GIS).

Academic and Professional Experience

Image Understanding and Pattern Recognition (IUPR), Kaiserslautern, Germany
http://sites.google.com/a/iupr.com/home/
Researcher: **October 2009 - To Date**
Working as a researcher scholar in the area of *Document Image Analysis*

Insiders Technologies, Kaiserslautern, Germany
http://www.insiders-technologies.de/
Core Researcher and Developer: **May 2012 - To Date**
Working as a core researcher on commercial products related to the field of document processing for optimizing business processes

German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, Germany
http://www.dfki.de
Researcher: **April 2007 - September 2009**
Worked as a research scholar in the area of *Document Image Analysis*

Fraunhofer-Gesellschaft, Kaiserslautern, Germany
http://www.itwm.fraunhofer.de/
Research Assistant and Developer: **August 2008 - September 2009**
Worked as a research assistant in *Industrial Mathematics, Image Processing Department*

NED University of Engineering and Technology, Karachi, Pakistan
http://www.neduet.edu.pk/
Lecturer: **Feb 2003 - Jan 2007**
Worked in Computer and Information System Engineering Department of NED University of Engineering and Technology

Projects

I have been working on the following research projects:

- **OCRopus OCR System**: The OCRopus project is an on-going effort to create a high-performance OCR system for both printed and handwritten text, and to develop novel and robust algorithms for document image preprocessing, page segmentation, text recognition, and statistical language modeling.
  (http://code.google.com/p/ocropus/)
- **DECAPOD**: Decapod is a project focused on building a low-cost digitization solution that will allow for rare materials, materials held in collections without large budgets, and other scholarly content to be digitized into a high-quality PDF format. This project works to incorporate the hardware and software necessary to accomplish this goal.
  (http://sites.google.com/site/decapodproject/)
- **SICURA**: The SICURA project aims to develop object recognition and image database retrieval techniques for automated X-ray image analysis.
  (https://sites.google.com/a/iupr.com/sicura/)
- **MapDigitizer**: This project aims to digitized handwritten maps.
  (https://www.sites.google.com/a/iupr.com/map-digitization/)

References

Prof. Dr. Thomas M. Breuel
Image Understanding and Pattern Recognition
Technical University of Kaiserslautern
Gottlieb-Daimler-Str., 67663 Kaiserslautern, Germany
Phone: +49 (0)631 205-3456 Email: tmb@iupr.com
http://sites.google.com/a/iupr.com/home/

Prof. Dr. Andreas Dengel
Member of the Management Board
Scientific Director
German Research Center for Artificial Intelligence (DFKI)
Trippstadter Strasse 122, D-67663 Kaiserslautern
Phone: +49 (0)631 205-75100 Email: Andreas.Dengel@dfki.de
http://www.dfki.de

Prof. Dr. Koichi Kise
Department of Computer Science and Intelligent Systems
Osaka Prefecture University, Japan
Phone: +81-72-254-9276 E-mail: kise@cs.osakafu-u.ac.jp
http://imlab.jp/~kise/index_e.html

Ms. Satoko Itaya
Assistant Manager
C & C Innovation Research Laboratories, NEC Corporation, Nara, Japan
Email: s-itaya@bp.jp.nec.com
http://www.nec.com/en/global/rd/labs/ccii/index.html

Publications

Syed Saqib Bukhari

Journal Papers

1. S. S. Bukhari, F. Shafait, T. M. Breuel, “Coupled Snakelets for Curled Text-Line Segmentation from Warped Document Images”, International Journal on Document Analysis and Recognition, IJDAR, 2011.
2. S. S. Bukhari, F. Shafait, T. M. Breuel, “Adaptive Binarization of Unconstrained Hand-Held Camera-Captured Document Images”, Special Issue of JUCS - Journal of Universal Computer Science, 2009.

Book Chapters

1. S. S. Bukhari, F. Shafait, T. M. Breuel, “Layout Analysis of Arabic Script Documents” for Book Chapter ‘Guide to OCR for Arabic Scripts’, Springer 2012.

Conference Papers

1. S. S. Bukhari, F. Shafait, T. M. Breuel, “Layout Analysis for Arabic Historical Document Images Using Machine Learning”, 13th International Conference on Frontiers in Handwriting Recognition, ICFHR ’12, Bari, Italy, 2012 **(The Best Student Paper Award)**.
2. A. Ul-Hasan, S. S. Bukhari, S. F. Rashid, F. Shafait, T. M. Breuel, “Semi-Automated OCR Database Generation for Complex Scripts”, 21st International Conference on Pattern Recognition, ICPR’12, Japan, November 2012.
3. M. Z. Afzal, S. S. Bukhari, M.Krmer, F. Shafait, T. M. Breuel, “Robust Stereo Matching for Document Images Using Parameter Selection of Text-Line Extraction”, 21st International Conference on Pattern Recognition, ICPR’12, Japan, November 2012.
4. M. Krmer, M. Z. Afzal, S. S. Bukhari, F. Shafait, T. M. Breuel, “Robust Stereo Correspondence for Documents by Matching Connected Components of Text-Lines with Dynamic Programming”, 21st International Conference on Pattern Recognition, ICPR’12, Japan, November 2012.
5. A. Ul-Hasan, S. S. Bukhari, F. Shafait, T. M. Breuel, “OCR-Free Table of Contents Detection in Urdu Books”, 10th IAPR Workshop on Document Analysis Systems, DAS12, Gold Coast, Australia, Mar. 2012.
6. M. Z. Afzal, M. Krmer, S. S. Bukhari, F. Shafait, T. M. Breuel, “Improvements to Uncalibrated Feature-based Stereo Matching for Document Images by using Text-Line Segmentation”, 10th IAPR Workshop on Document Analysis Systems, DAS12, Gold Coast, Australia, Mar. 2012.
7. S. S. Bukhari, F. Shafait, T. M. Breuel, “Border Noise Removal of Camera-Captured Document Images using Page-Frame Detection”, Camera-Based Document Analysis and Recognition - 4th International Workshop, CBDAR 2011, Beijing, China, September 22, 2011, Revised Selected Papers Springer 2012.
8. S. S. Bukhari, F. Shafait, T. M. Breuel, “A Pixel-Based Performance Evaluation Method for Page Dewarping Algorithms using SIFT Features”, Camera-Based Document Analysis and Recognition - 4th International Workshop, CBDAR 2011, Beijing, China, September 22, 2011, Revised Selected Papers Springer 2012.
9. S. S. Bukhari, F. Shafait, T. M. Breuel, “The IUPR Dataset of Camera-Captured Document Images”, Camera-Based Document Analysis and Recognition - 4th International Workshop, CBDAR 2011, Beijing, China, September 22, 2011, Revised Selected Papers Springer 2012.

10. F. Shafait, M. P. Cutter, J. V. Beusekom, S. S. Bukhari, T. M. Breuel, “Decapod: A flexible, low cost digitization solution for small and medium archives”, Camera-Based Document Analysis and Recognition - 4th International Workshop, CBDAR 2011, Beijing, China, September 22, 2011, Revised Selected Papers Springer 2012.

11. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Text-Line Extraction using a Convolution of Isotropic Gaussian Filter with a Set of Line Filter”, 11th International Conference on Document Analysis and Recognition, ICDAR’11, Beijing, China, September 2011.

12. S. S. Bukhari, F. Shafait, and T. M. Breuel, “High Performance Layout Analysis of Arabic and Urdu Scripts Document Image”, 11th International Conference on Document Analysis and Recognition, ICDAR’11, Beijing, China, September 2011.

13. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Improved Document Image Segmentation Algorithm using Multiresolution Morphology”, Document Recognition and Retrieval XVIII, SPIE 2011, San Francisco, CA USA, 2011.

14. S. S. Bukhari, M. Al Azawi, F. Shafait, and T. M. Breuel, “Document Image Segmentation using Discriminative Learning over Connected Components”, 9th IAPR Workshop on Document Analysis Systems, DAS’10, Boston, MA, USA, 2010.

15. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Performance Evaluation of Curled Textlines Segmentation Algorithms on CBDAR 2007 Dewarping Contest Dataset”, International Conference on Image Processing, ICIP’10, Hong kong, 2010.

16. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Performance Evaluation of Curled Textlines Segmentation Algorithms”, Camera-Based Document Analysis and Recognition - 4th International Workshop, CBDAR 2011, Beijing, China, September 22, 2011, Revised Selected Papers Springer 2012mas M. Breuel, (short paper) 9th IAPR Workshop on Document Analysis Systems, DAS’10, Boston, MA, USA, 2010.

17. ‘S. S. Bukhari, F. Shafait, and T. M. Breuel, “Coupled Snakelet Model for Curled Textline Segmentation of Camera-Captured Document Images”, 10th International Conference on Document Analysis and Recognition, ICDAR’09, Barcelona, Spain, 2009.

18. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Script-Independent Handwritten Textlines Segmentation using Active Contours”, 10th International Conference on Document Analysis and Recognition, ICDAR’09, Barcelona, Spain, 2009.

19. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Ridges based Curled Textline Region Detection from Grayscale Camera-Captured Document Images”, 13th International Conference on Computer Analysis of Images and Patterns, CAIP’09, Muenster, Germany, 2009.

20. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Curled Textline Information Extraction from Grayscale Camera-Captured Document Images”, International Conference on Image Processing, ICIP’09, Cairo, Egypt, 2009.

21. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Foreground-Background Regions Guided Binarization of Camera-Captured Document Images”, 3rd International Workshop on Camera Based Document Analysis and Recognition, CBDAR’09, Barcelona, Spain, 2009.

22. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Dewarping of Camera-Captured Document Images”, 3rd International Workshop on Camera Based Document Analysis and Recognition, CBDAR’09, Barcelona, Spain, 2009.

23. S. F. Rashid, S. S. Bukhari, F. Shafait, and T. M. Breuel, “A Discriminative Learning Approach for Orientation Detection of Urdu Document Images”, 13th IEEE International Multitopic Conference, INMIC’09, Islamabad, Pakistan, 2009.

24. S. S. Bukhari, F. Shafait, and T. M. Breuel, “Segmentation of Curled Text Lines using Active Contours”, 8th IAPR Workshop on Document Analysis Systems, DAS’08, Nara, Japan, 2008.