Camera-Captured Document Image Analysis

A Thesis
Submitted for the Degree of
Doctor of Philosophy
in the Faculty of Engineering

by

Thotreingam Kasar

Department of Electrical Engineering
Indian Institute of Science
Bangalore – 560 012

NOVEMBER 2011

Abstract

Text is no longer confined to scanned pages and often appears in camera-based images originating from text on real world objects. Unlike the images from conventional flatbed scanners, which have a controlled acquisition environment, camera-based images pose new challenges such as uneven illumination, blur, poor resolution, perspective distortion and 3D deformations that can severely affect the performance of any optical character recognition (OCR) system. Due to the variations in the imaging condition as well as the target document type, traditional OCR systems, designed for scanned images, cannot be directly applied to camera-captured images and a new level of processing needs to be addressed. In this thesis, we study some of the issues commonly encountered in camera-based image analysis and propose novel methods to overcome them. All the methods make use of color connected components.

1. Connected component descriptor for document image mosaicing

Document image analysis often requires mosaicing when it is not possible to capture a large document at a reasonable resolution in a single exposure. Such a document is captured in parts and mosaicing stitches them into a single image. Since connected components (CCs) in a document image can easily be extracted regardless of the image rotation, scale and perspective distortion, we design a robust feature named connected component descriptor that is tailored for mosaicing camera-captured document images. The method involves extraction of a circular measurement region around each CC and its description using the angular radial transform (ART). To ensure geometric consistency during feature matching, the ART coefficients of a CC are augmented with those of its 2 nearest neighbors. Our method addresses two critical issues often encountered in correspondence matching: (*i*) the stability of features and (*ii*) robustness against false matches due to multiple instances of many characters in a document image. We illustrate the effectiveness of the proposed method on camera-captured document images exhibiting large variations in viewpoint, illumination and scale.

2. Font and background color independent text binarization

The first step in an OCR system, after document acquisition, is binarization, which

converts a gray-scale/color image into a two-level image - the foreground text and the background. We propose two methods for binarization of color documents whereby the foreground text is output as black and the background as white regardless of the polarity of foreground-background shades.

(a) *Hierarchical CC Analysis*: The method employs an edge-based connected component approach and automatically determines a threshold for each component. It overcomes several limitations of existing locally-adaptive thresholding techniques. Firstly, it can handle documents with multi-colored texts with different background shades. Secondly, the method is applicable to documents having text of widely varying sizes, usually not handled by local binarization methods. Thirdly, the method automatically computes the threshold for binarization and the logic for inverting the output from the image data and does not require any input parameter. However, the method is sensitive to complex backgrounds since it relies on the edge information to identify CCs. It also uses script-specific characteristics to filter out edge components before binarization and currently works well for Roman script only.

(b) *Contour-based color clustering (COCOCLUST)*: To overcome the above limitations, we introduce a novel unsupervised color clustering approach that operates on a 'small' representative set of color pixels identified using the contour information. Based on the assumption that every character is of a uniform color, we analyze each color layer individually and identify potential text regions for binarization. Experiments on several complex images having large variations in font, size, color, orientation and script illustrate the robustness of the method.

3. Multi-script and multi-oriented text extraction from scene images

Scene text understanding normally involves a pre-processing step of text detection and extraction before subjecting the acquired image for character recognition task. The subsequent recognition task is performed only on the detected text regions so as to mitigate the effect of background complexity. We propose a color-based CC labeling for robust text segmentation from natural scene images. Text CCs are identified using a combination of support vector machine and neural network classifiers trained on a set of low-level features derived from the boundary, stroke and gradient information. We develop a semi-automatic annotation toolkit to generate pixel-accurate groundtruth of 100 scenic images containing text in various layout styles and multiple scripts. The overall precision, recall and f-measure obtained on our dataset are 0.8, 0.86 and 0.83, respectively. The proposed method is also compared with others in the literature using the ICDAR 2003 robust reading competition dataset, which, however, has only horizontal English text. The overall precision, recall and f-measure obtained are 0.63, 0.59 and 0.61 respectively, which is comparable to the best performing methods in the ICDAR 2005 text locating competition. A recent method proposed by Epshtein *et al.* [1] achieves better results but it cannot handle arbitrarily oriented text. Our method, however, works well for generic scene images having arbitrary text orientations.

4. Alignment of curved text lines

Conventional OCR systems perform poorly on document images that contain multi-oriented text lines. We propose a technique that first identifies individual text lines by grouping adjacent CCs based on their proximity and regularity. For each identified text string, a B-spline curve is fitted to the centroids of the constituent characters and normal vectors are computed along the fitted curve. Each character is then individually rotated such that the corresponding normal vector is aligned with the vertical axis. The method has been tested on a data set consisting of 50 images with text laid out in various ways namely along arcs, waves, triangles and a combination of these with linearly skewed text lines. It yields 95.9% recognition accuracy on text strings, where, before alignment, state-of-the-art OCRs fail to recognize any text.

The CC-based pre-processing algorithms developed are well-suited for processing camera-captured images. We demonstrate the feasibility of the algorithms on the publicly-available ICDAR 2003 robust reading competition dataset and our own database comprising camera-captured document images that contain multiple scripts and arbitrary text layouts.

Acknowledgements

It is a great pleasure for me to thank Prof. A. G. Ramakrishnan, my doctoral advisor, for giving me the privilege to work with him. His kind inspiration, invaluable guidance and encouragement from time to time had kept me going in spite of the adversities confronted and have culminated in the successful completion of this thesis.

I would like to acknowledge the faculty of this institute for excellent courses they have offered which has equipped me with the background required to carry out my research work. I would also like to thank the chairman of the department, Dr. P.S. Sastry and other faculty members for their kind help and encouragement. I am grateful to all the staffs of the department for their co-operation and friendly moral support throughout.

I have benefitted a lot from my colleagues at MILE lab. Thanks to Pati, Neelam, Farshad, Ananth, Lakshmish, Bhavna, Sri Lakhmi, Shanthi, Swetha, Jayant, Arvind, Amrik, Ranjith, Suresh, Mahadev Prasad, Ranjani, Satyanarayana, Shiva, Deepak, Vikram, Anil and Girish. I will always cherish the good times we have had. I have also learnt a lot working with Varun, Apoorva, Varsha, Sonam, Sharanya, Musfira Jilani, Amey and Abhishek. Thanks to all my friends Robindro, Jugeshwar, Sunanda, Sadananda, Lun, Suhesh, Raghu, Tomba, Bisheswar, Nganba, Ranjita, Johnson, Donald, Kiran, Sidhartha, Samir, Biswajit, Subhajit, Chandrashekhar, Supratim, Diyanshu, Aswin, Senti, Dapme, Bari, Divya Anand, Mahader and Smriti. You have made my stay at I S-110, IISc Bangalore, a memorable period of my life.

I would also like to thank all of my family and my fiancee Crassina for their unfailing support and love all through. Above all, I thank the Almighty God for bringing me thus far.

Contents

List of Figures

List of Tables

Chapter 1

Camera-based document image analysis

"A journey of a thousand miles begins with a single step." - *Confucius*

Abstract: *Over the last decade, there has been a rapid spurt of research in camera-based document analysis and recognition. This strong interest is mainly driven by the availability of inexpensive digital cameras that are compact, easy to use, fast, portable, and are more suitable for capturing documents in less-constrained environments. In addition to imaging hard copy paper manuscripts, they are now used to image text in natural scenes which normally would be inaccessible. However, this flexibility and ease of use come at the cost of much more challenging images to process. Camera-captured images inherently suffer from various geometric and photometric distortions that are not present in a scanned image. Traditional scanner-based document analysis systems fail against this new and promising acquisition mode. Specialized techniques are required for processing such images so as to tap the potential advantages of camera-based systems. This chapter gives an overview of the challenges in camera-based document processing and touches upon the key research issues dealt with in this work.*

1.1 Introduction

Documents organize and present information for human understanding and our daily life is governed by the efficiency with which we process information. Technology to automate the extraction and dissemination of information from documents can help us achieve significant productivity enhancement. The birth of commercial OCR systems in the 1950s allowed for an automated reading and interpretation of hard-copy documents by eliminating the need to input information manually and enabling accurate and efficient information processing in less time. Since then, document image analysis has been extensively studied, which has pushed the character recognition accuracy for Roman script beyond 99% for high quality documents with clean backgrounds. It has found numerous successful applications in tasks with large continuous data-processing requirements like postal address recognition, form processing, bank cheque and invoice reading.

Historically, papyrus, palm leaves, clay/wood/stone tablets, metal plates and paper have been used as media for documentation. Though present day documents exist both in printed and electronic format, paper is still the preferred medium of communication, since it allows easy navigation through documents, cross-referencing of several documents at the same time, annotations and interweaving of reading with writing.

Traditional document image analysis systems use flatbed, sheet-fed, or mounted imaging devices for imaging hard copy paper manuscripts. However, in recent years, it has become hard to define the term document due to the blurring in the distinction between documents and user-interfaces. Text is no longer confined to scanned pages and often appears in camera-based images originating from text on real world objects. Hand-held imaging devices such as digital still cameras, camcorders, cameras in personal digital assistants and mobile phones are now being used for supplementing the traditional method of document image acquisition [2]. Small, light, and handy, these devices enable document acquisition with minimal constraints on the imaging environment. In addition to imaging hard copy documents, they capture text present on 3-D real world objects such as buildings, billboards, road signs, license plates, black/white boards or even on a T-shirt which otherwise would be inaccessible.

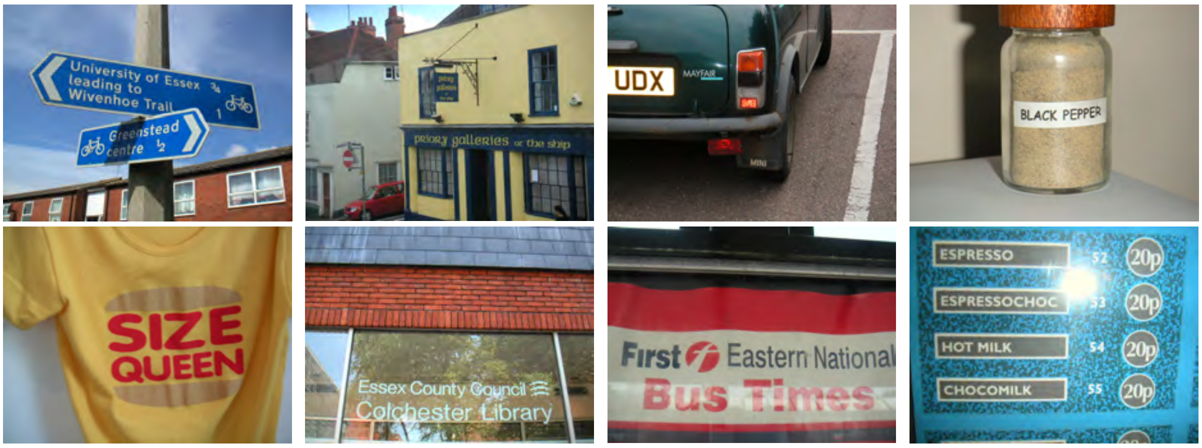


Figure 1.1: Some sample images from the ICDAR 2003 robust reading competition data set containing text on 3-D real world objects such as road signs, buildings, vehicles, cans and T-shirts. Clearly, these scene images require specialized pre-processing techniques to detect, locate and extract text regions before attempting to recognize its content.

This pervasive use of hand-held digital cameras has immense potential for newer applications that go far beyond what traditional OCR has to offer. However, due to the variations in the imaging condition as well as the target document type, the technology of traditional scanner-based OCR systems is insufficient for camera-captured images and a new level of processing needs to be done. This has led to the new sub-field of research on analysis and recognition of camera-captured documents. Addressing these new challenges is important for the progress in the field. In 2003, a robust reading competition was held at the 7^{th} international conference of document analysis and recognition (ICDAR) to find the system that best reads complete words in camera-captured scene images. Some sample images from the data set are shown in Fig. 1.1. These images are beyond the capabilities of current commercial OCR packages and therefore require specialized techniques. In 2005, the first international workshop on camera-based document analysis and recognition (CBDAR) was held in conjunction with ICDAR with a special focus on camera-captured documents to boost research in this relatively new area.

1.2 Potential applications of CBDAR

The camera provides a great opportunity for input from the physical world. Recognizing text in real-world scenes can be considered as an extension of current OCR technology widely available in the market. While the market of imaging devices is exhibiting a gradual decline in the sale of scanners, there is a rapid rise in the sale of digital cameras, especially that of camera phones. Based on a Gartners report [3], mobile phones will overtake the PC as the most common web-access device worldwide by 2013. Powered by increasing computational capabilities, these cameras are now being used as ubiquitous input devices. The combined capabilities of image acquisition, processing, storage and communication in a compact, portable device make it an ideal platform for embedding computer vision and image processing capabilities [4]. Mobile phones embed many other technologies such as text-to-speech, voice recognition, global positioning system and network connection to provide a synergy with camera-based document processing. Applications of CBDAR are numerous and limited only by the imagination. We illustrate a few such applications below.

Augmented reality: An exciting area for camera-based OCR is the field of augmented reality. Camera-based white-board capture systems are an integral part of computer-aided collaborative working environments that bridge the worlds of electronic and physical documents together and facilitate the interaction of humans with different kinds of documents. The Digital Office project [5] at Xerox involves camera systems that can interpret human gestures and scan paper documents, white boards, books, desktops and the human office workers themselves. It has three different systems namely (i) The ZombieBoard that provides a test-bed for exploring new forms of human computer interaction based on an understanding of hand-drawn diagrams and gesture recognition, (ii) desktop scanning using a video camera with/without projection onto the desktop, and (iii) a perceptual browser that allows users to interact with electronic documents in a way that is analogous to their interactions with paper documents.

Wearable computing: Another promising application is in the field of wearable computing. A camera embedded in a pair of glasses could make use of OCR technology to augment the user's view of the world with an endless range of information. Visually impaired people directly benefit from such research. Recognizing text in real-world scenes, coupled with text-to-speech technology, can make machines 'read' street signs, book covers, bank notes, name tags and labels on office doors, medicine labels and LCD/LED displays of digital devices. For persons with visual disability, such devices hold great promise and will be a reality, strengthened by the recent launch of automated book readers.

Robotic navigation: In general, way-finding in man-made environments is helped considerably by the ability to read signs. Text phrases on door plates, posters and labeled installations can be used to indirectly identify the user's current position [6]. Scene text recognition also finds application in high-level robot navigation, such as path planning or goal-driven navigation [7].

Portable travel aids: Cameras with translation capability have been proposed to acquire and automatically decipher snapshots of text written in foreign languages. Such technology will help people traveling abroad in understanding signs, names of streets and destinations of buses and trains. Several such prototype systems [8, 9, 10] have been reported in the literature.

Digital libraries: Cameras are the ideal choice for digitization of books in digital libraries owing to the fast and non-contact mechanism of image acquisition. High resolution cameras are often used to image documents of bound volumes or to image fragile historical documents that cannot or should not be handled. While manual scanning has a throughput of 5-8 pages/minute, the KABIS III book imaging system [11] of Kirtas Technologies allows fast acquisition at the rate of 50 pages/minute at 400 dpi in full 24-bit color.

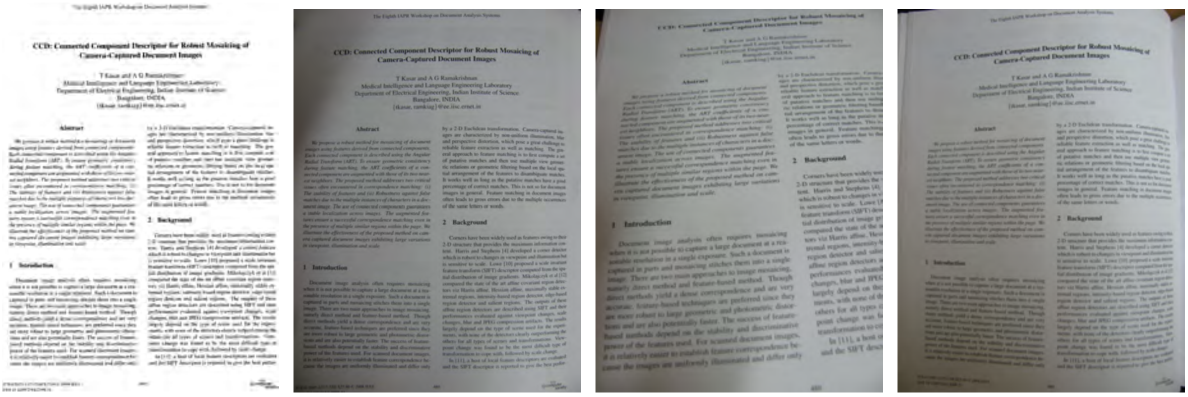
Content-based video indexing and retrieval: Text detection in video key frames has received a great deal of attention as it provides a supplementary way of indexing the video [12]. Text occurring in video, if extracted, naturally gives clues about its content since words have well-defined, unambiguous meanings. Text detection and recognition in images and video frames, which aims at integrating advanced OCR and text based searching technologies, is now recognized as a key component in the development of advanced image and video annotation and retrieval systems.

1.3 Challenges in camera-based document analysis

State-of-the-art OCR systems can produce very good results for high resolution, high quality document images with a clean background. In practice, many real-life documents seldom satisfy these optimal conditions. The presence of complex backgrounds, arbitrary layouts, noisy data, merged/broken characters, variations in font styles, size and color can drastically affect the performance of any OCR system. In addition, camera-based images, due to the less-constrained mode of acquisition, are often characterized by non-uniform illumination, poor resolution, focus misalignment, perspective distortion and 3-D deformation.

Uneven illumination: Unlike scanners, digital cameras have far less control of the lighting conditions of the acquisition environment. Uneven illumination is common, due to both the physical environment such as shadows or reflective surfaces and lack of controlled lighting. In addition, camera images inherently suffer from vignetting, where the center of the view is bright and the intensity gradually decays towards the image boundary. Extraction of characters from images with varying background illumination is not a trivial task. It is highly desirable to have good character segmentation, since it affects all the subsequent processing steps involved in recognition. This calls for robust thresholding and character segmentation techniques.

Perspective distortion: Perspective distortion occurs when the text plane is not parallel to the imaging plane. It usually occurs while capturing text on buildings, road signs, billboards or charts where it is physically infeasible to have an orthographic view. The effect is that characters situated farther away appear smaller and the parallel line assumption no longer holds in the image. This results in increased difficulty in performing layout analysis and decomposition of the document image into text lines, words and characters. In addition, the characters get deformed leading to difficulty in recognition. The conventional process of skew detection and correction cannot handle perspective distortion. Current document analysis systems, tailored for scanner-based input, fail when the captured image



(a) (b)

Figure 1.2: Issues with camera-captured images. (a) A scanned document image (b) The same document captured from a hand-held camera exhibiting non-uniform illumination, perspective distortion and 3D deformation.

exhibits perspective distortion.

Warped text: Pages of an opened book are rarely flat and are more often curled. Just like the case of perspective distortion, page layout analysis and character segmentation become a difficult task. Current document analysis systems fail even under a moderate warping. Fig. 1.2 shows the difference between the scanned image of a document and the image of the same captured by a camera. While the effects of non-uniform illumination can be handled by local adaptive thresholding techniques, image warping and rendering techniques are required to undo geometric distortions. In any camera-based document analysis system, it is first required to normalize the photometric and geometric distortions present in the acquired image and render it into a form similar to that of a scanned image.

Background clutter and arbitrary layouts: Scene text can appear on any surface, not necessarily on a plane. Such text is the primal target of camera-based document analysis systems. Unlike that of processing conventional document images, scene text understanding normally involves a pre-processing step of text detection and extraction before subjecting the acquired image for character recognition task. The subsequent recognition task is performed only on the detected text regions so as to mitigate the effect of background complexity. However, the available technology is still far from being able to reliably separate text from the background clutter. It is difficult to precisely define the

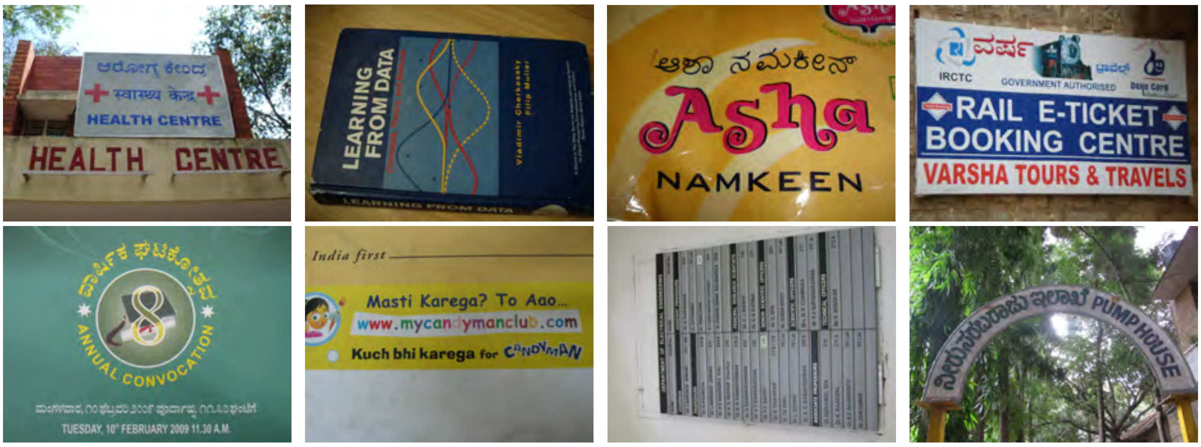


Figure 1.3: Example scene images exhibiting non-uniform illumination, presence of arbitrary text orientation and layout, perspective distortion, multi-script content, artistic font styles multiple colors and complex backgrounds.

features of text in a scene image since they are often characterized by artistic fonts, variable size and color, arbitrary layouts, complex backgrounds and clutter. Figure 1.3 highlights these issues that arise in natural scene images. Arbitrary text layouts and background clutter pose the biggest challenge in layout analysis and character segmentation. Current OCR systems cannot handle scene images mainly due to their inability to segment the foreground text from the background. In fact, this weakness of the current OCR systems is exploited in designing CAPTCHA (Completely Automated Public Turing Test To Tell Computers and Humans Apart) that requires the user to type letters or digits from a distorted image on the screen; this prevents automated software from performing actions that degrade the quality of service of a given system, whether due to abuse or resource expenditure. While computers are unable to solve it accurately, humans can easily read distorted text and any user entering the correct text is presumed to be a human being.

Multiple scripts: The presence of multiple scripts requires special treatment too. In a multi-lingual country like India, many documents, forms and signboards are generally bi-lingual or tri-lingual in nature. It is common to find English words interspersed within sentences in Indic-script documents. Every script has certain characteristics that distinguish it from other scripts and may require script-specific processing methods. Identification of script plays a vital role for the development of next generation OCRs. Besides enabling automated processing and utilization of documents, it allows us to exploit the language and/or script specific rules to enhance the OCR output.

Text detection in video: Currently, most research on camera-based document analysis is focussed on still imagery but video text processing is gaining attention because it could be one of the sources of documents in future. Current digital video cameras have low resolution, typically 640 × 480 or 320 × 240 pixels per frame, because they are designed primarily for low bandwidth presentation and are often highly compressed. The fact that they are not designed specifically for document image capture presents many challenges in text recognition from videos. The temporal nature of video introduces a new dimension into the text extraction problem; while some text could be static, others like movie credits may scroll in a linear fashion. It is difficult to detect text with varying size, embedded in real world scenarios, and captured using an unconstrained video camera.

Current consumer grade digital cameras have a resolution of 8-14 mega pixels which should be sufficient for capturing standard documents. The Nokia camera phone N8 has a 12 mega pixel camera but mainstream phones still have resolutions in the range of 1-3 mega pixels. With the availability of hand-held imaging devices with increased memory and computing capabilities, the whole chain of processing including image acquisition, recognition, storage, and communication can soon be performed on the device itself.

1.4 Research issues addressed in this thesis

Robust pre-processing techniques can enable the use of off-the-shelf commercial OCR packages on camera-captured images, obviating the need for any modification of the OCR. In this work, we study some of the issues commonly encountered in camera-based image analysis and propose novel methods to overcome them.

Document image mosaicing: The first problem considered is acquisition of ‘large’ documents that often require mosaicing since it may not be possible to acquire the document at a good resolution in a single exposure. Such a document is captured in parts and image mosaicing technique is used to stitch them into a single composite image.

We exploit the nature of the type of image in deriving a robust feature descriptor suitable for mosaicing document images. Our method addresses two critical issues often encountered in correspondence matching: (*i*) the stability of features and (*ii*) robustness against false matches due to multiple instances of the same characters in a document image. We illustrate the effectiveness of the proposed method on camera-captured document images exhibiting large variations in viewpoint, illumination and scale.

Binarization of color documents: The first step in an OCR system, after document acquisition, is binarization, which converts a gray-scale/color image into a two-level image - the foreground text and the background. We propose two novel methods for binarization of color documents whereby the foreground text is output as black and the background as white regardless of the polarity of foreground-background shades. Our approach overcomes several limitations of existing locally-adaptive thresholding techniques. Firstly, it can handle documents with multi-colored texts with different background shades. Secondly, the method is applicable to documents having text of widely varying sizes, usually not handled by local binarization methods. Thirdly, the method automatically computes the threshold for binarization and the logic for inverting the output from the image data and does not require any input parameter. The proposed methods have been applied to a broad domain of target document types and are found to have a good adaptability.

Multi-script and multi-oriented text localization from scenic images: Text in natural scenes requires special treatment since it is often designed with artistic fonts, colors and complex backgrounds and layouts. The standard approach in natural scene understanding is to first detect the presence of text and localize the text regions. The subsequent processes involved in recognition are then performed only on the detected text regions so that the effect of complex backgrounds is minimized. We use color edge detection followed by a post-processing step of linking open edges. The edge information is used to obtain a set of colors which in turn is used to initiate an unsupervised clustering algorithm that yields accurate and robust identification of CCs from complex scenic images. Text CCs are identified using a combination of a support vector machine and a neural network classifier trained on a set of ‘intrinsic’ text features derived from the geometric properties, boundary, stroke and gradient information. Experiments on camera-captured images that contain variable font, size, color, irregular layout, non-uniform illumination and multiple scripts illustrate the robustness of the method. A comparison of the proposed method with others in the literature is also performed using the ICDAR 2003 robust reading competition dataset.

Alignment of curved character strings: Conventional OCR systems perform poorly on document images that contain multi-oriented text lines. While the horizontally aligned text is easily detected and recognized, curved text not only poses a challenge to recognition but also makes the text segmentation process difficult. In most existing OCR systems, a skew correction process is often performed prior to recognition, should a need arise. Most skew estimation techniques assume the presence of long and straight text lines which is seldom valid for scene images. We propose a 2-step technique that first identifies individual text strings by grouping adjacent CCs based on their proximity and regularity. Then, for each identified text string, a B-spline curve is fitted to the centroids of the constituent characters and normal vectors are computed along the fitted curve. Each character is individually rotated such that the corresponding normal vector is aligned with the vertical axis. The method has been successfully tested on text layouts where state-of-the-art OCRs fail to recognize any text.

1.5 Organization of the thesis

This thesis is organized as follows. Chapter 2 starts with the theory of image mosaicing and an overview of existing techniques for mosaicing document images. An image mosaicing technique is presented next, based on a novel feature specialized for document images. The main focus is on the design of a feature invariant to translation, rotation and scale variations and in tailoring it to handle multiple occurrences of a letter at different places in the document image. Experimental results on camera captured document images exhibiting large variations in viewpoint, illumination and scale and the conclusions drawn are given in the end. In chapter 3, we address the issue of extracting the foreground text from the background via thresholding. This chapter describes two novel CC-based binarization methods that overcome the limitations of existing local thresholding techniques.

The proposed methods are tested on a broad domain of target document types and the results obtained are discussed. Chapter 4 discusses the problem of locating multi-script and multi-oriented text from natural scenic images. It describes a novel method for robust extraction of CCs. Text CCs are identified using a classifier trained on a set of 'intrinsic' features of text. We illustrate the versatility of the proposed method by the results on arbitrary layouts and multi-script documents. For comparison with others in the literature, the proposed method is also tested on the ICDAR 2003 robust reading competition dataset. In chapter 5, we present a new technique to detect and extract text laid out in a curvilinear fashion, where each character in the text string is skewed differently and to align them horizontally. The core modules of the method are the text string extraction module that identifies all the text strings present in the image and the alignment module that transforms the identified curved/skewed text strings into horizontal text lines. The effectiveness of the proposed method is validated by experiments on images with various text layouts on which state-of-the-art OCRs fail to recognize any text. Chapter 6 concludes the thesis with a summary of the experiments carried out, lists the main contributions and highlights the avenues for further work.

Chapter 2

Connected component descriptor for document image mosaicing

"The whole is greater than the sum of its parts." - Aristotle

Abstract: *This chapter describes a new region-based descriptor, well suited for document image processing. Connected components are natural candidates for localization of features in document images since they can easily be extracted, are highly stable and largely invariant to geometric and photometric distortions. We propose a feature descriptor called connected component descriptor (CCD) where each CC is described using the angular radial transform (ART). To ensure geometric consistency during feature matching, the ART coefficients of a connected component are augmented with those of its two nearest neighbors. The main focus is on the design of a feature invariant to translation, rotation and scale changes and how it is tailored to handle multiple occurrences of a character at different places in the document image. The robustness of CCD is amply illustrated by our experiments on camera-captured images exhibiting large variations in viewpoint, illumination and scale.*

2.1 Introduction

Image mosaicing refers to the alignment of multiple, partially-overlapping images representing portions of a scene into a single composite image. Digital cameras bundled with image mosaicing capability can act as a synthetic wide-angle lens camera in creating high-resolution mosaics from a set of lower resolution images. Mosaics of images have a number of interesting applications such as creating virtual environments [13], panoramic images [14], representing and indexing video information [15]. Document image analysis often requires mosaicing when it is not possible to capture a large document at a reasonable resolution in a single exposure. Such a document is captured in parts and mosaicing stitches them into a single image. There are two main approaches to image mosaicing, namely the direct method and the feature-based method. Though direct methods yield a dense correspondence and are very accurate, feature-based techniques are preferred since they are more robust to large geometric and photometric distortions and are also potentially faster. They can automatically discover the adjacency relationships among an unordered set of images, which makes them ideally suited for fully-automated image mosaicing. Hence, feature-based methods are used throughout this thesis.

The success of feature-based methods depends on the stability and discriminative power of the features used. For scanned document images, it is relatively easier to establish feature correspondence because the images are uniformly illuminated and differ only by a 2-D similarity transformation. Camera-captured images are characterized by non-uniform illumination, and have blur and perspective distortion, which pose challenges to reliable feature extraction as well as matching. The general approach to feature matching is to first compute a set of putative matches and then use multiple view geometric relations based on the local spatial arrangement of the features to disambiguate matches. It works well as long as the putative matches have a good percentage of correct matches. This is not so for document images in general. Feature matching in document images often leads to gross errors due to the multiple occurrences of the same letters or words. We propose a new region-based descriptor that addresses the above issue.

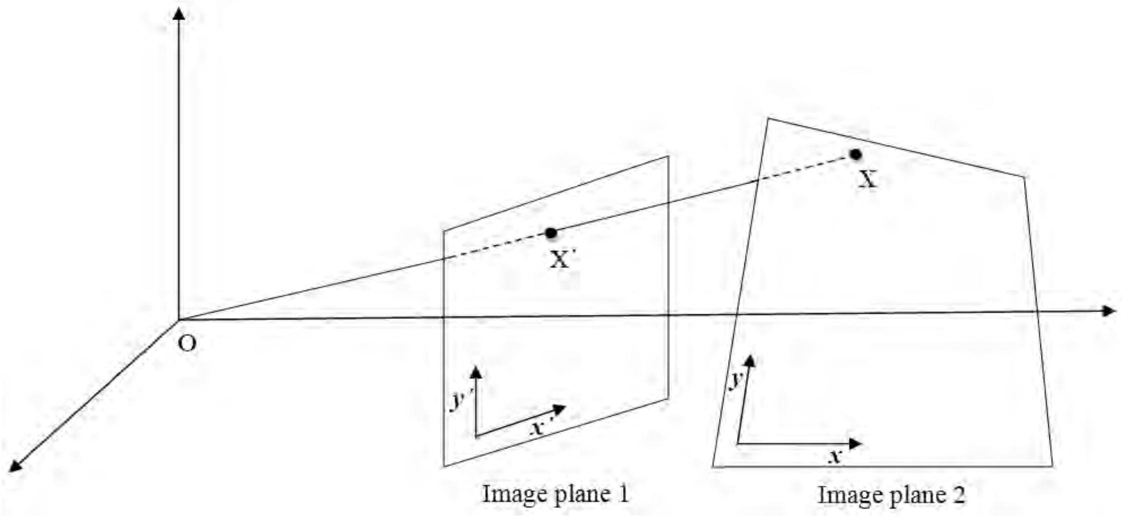


Figure 2.1: Planar projective transformation.

2.2 Theory of image mosaicing

Image mosaicing involves warping multiple images onto a common coordinate frame by establishing point-to-point correspondences accross the input images. The correspondence problem can be stated as follows: *Given two different views of a scene, for each image point in one view, find the image point in the second view that corresponds to the same point in the scene.* For imaging planar surfaces under general camera motion, the images are related by a planar projective transformation, also called a homography.

The general projective transformation of one image plane to another (refer Fig. 2.1) is given by [16]:

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{2.1}$$

which can be written as $\mathbf{X}'_i = \mathbf{H}\mathbf{X}_i$ where $\mathbf{X}_i = (x_i, y_i, z_i)^T$ and $\mathbf{X}'_i = (x'_i, y'_i, z'_i)^T$ represent the homogeneous coordinates of corresponding image points $(\frac{x_i}{z_i}, \frac{y_i}{z_i})$ and $(\frac{x'_i}{z'_i}, \frac{y'_i}{z'_i})$ respectively in the Euclidean space. The 3×3 matrix $\mathbf{H}$ is the homography that relates the two image planes. Since there are 8 independent ratios amongst the 9 elements of $\mathbf{H}$, a projective transformation has 8 degrees of freedom. The equality in Eqn. 2.1 holds up to a scale factor. The projective transformation can be written in the inhomogeneous form as follows:

$$\begin{aligned} \frac{x'_i}{z'_i} &= \frac{h_{11}x_i + h_{12}y_i + h_{13}z_i}{h_{31}x_i + h_{32}y_i + h_{33}z_i} \\ \frac{y'_i}{z'_i} &= \frac{h_{21}x_i + h_{22}y_i + h_{23}z_i}{h_{31}x_i + h_{32}y_i + h_{33}z_i} \end{aligned} \tag{2.2}$$

Thus, each point correspondence generates two equations which are linear in the elements of **H**:

$$\begin{aligned} x'_i(h_{31}x_i + h_{32}y_i + h_{33}z_i) &= (h_{11}x_i + h_{12}y_i + h_{13}z_i)z'_i \\ y'_i(h_{31}x_i + h_{32}y_i + h_{33}z_i) &= (h_{21}x_i + h_{22}y_i + h_{23}z_i)z'_i \end{aligned} \tag{2.3}$$

Four point correspondences lead to eight such linear equations in the entries of H which are sufficient to solve for H. Since scale is insignificant in projective geometry, we can normalise the third homogeneous coordinate z'_i to 1. Similarly, the elements of the homography **H** can be uniformly scaled by $1/h_{33}$ so that its last element equals 1. With these substitutions, Eqn. 2.3 can be expressed as follows:

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_ix_i & -x'_iy_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_ix_i & -y'_iy_i \end{bmatrix} \begin{bmatrix} h11 \\ h12 \\ h13 \\ h21 \\ h22 \\ h23 \\ h31 \\ h32 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \tag{2.4}$$

For four such point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{X}'_i, i = 1, 2, .., 4$, we have

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 \end{bmatrix}}\underbrace{\begin{bmatrix} h11 \\ h12 \\ h13 \\ h21 \\ h22 \\ h23 \\ h31 \\ h32 \end{bmatrix}} = \underbrace{\begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix}} \tag{2.5}$$

$$\text{or,} \qquad \mathbf{A} \qquad \mathbf{h} = \mathbf{b} \tag{2.6}$$

The required homography is obtained by solving for **h**. If the positions of point correspondences are exact, there is an exact solution for **h**. In practice, the image coordinates of these corresponding points are usually obtained by matching image features such as interest points. Though there are interest point detectors that are accurate up to sub-pixel resolutions, they inevitably have small localization errors and the estimated point correspondences may not be fully compatible with any projective transformation. Instead of demanding an exact solution, a large number of point correspondences are usually computed in practice and the over-determined system of equations is solved using least squares.

$$\mathbf{h} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{2.7}$$

The next step is to warp one of the images (target image) onto the other (reference image) so that they share a common coordinate system. This involves copying each pixel in the target image onto the reference image at positions determined by the homography. Since we have multiple intensity values at each pixel location in the overlapping region, blending is performed to obtain a single value using a suitable blending function. In this thesis, we adopt the feathered blending approach [16], where the weighting function for each component image has a unit value at its center and progressively diminishes to zero at the image boundaries. Thus, the pixel intensities are assigned gradually diminishing weightage as we move away from the respective image center. This blending process eliminates the intensity discontinuities that may occur at the image boundaries resulting in a seamless mosaic.

The main challenge of mosaicing lies in establishing the point-to-point correspondence between the input image pairs, from which the homography can be computed. Robust and accurate detection of image features and matching of the detected features across different views form the key to the success of these algorithms.

2.3 Review of featured-based document image mosaicing

Corners have been widely used as features owing to their 2-D structure that provides the maximum information content. Harris and Stephens [17] developed a corner detector which is robust to changes in viewpoint and illumination but is sensitive to scale. Lowe [18] proposed a scale invariant feature transform (SIFT) descriptor computed from the spatial distribution of image gradients. In [19], a host of local feature descriptors are evaluated and the SIFT descriptor is reported to give the best performance. The SIFT descriptor has been successfully used by many researchers in a number of applications such as object recognition, generating panoramic images and image retrieval. However, being a local feature descriptor, the SIFT descriptor does not work well when there are repetitive structures in the image [20, 21]. It does not make any distinction between different instances of the same letter in the document image; this can lead to a large number of outliers in correspondence matching.

Mikolajczyk et al [22] compared the state-of-the-art affine covariant region detectors namely Harris affine, Hessian affine, maximally stable extremal regions, intensity-based region detector, edge-based region detector and salient regions. The outputs of these affine region detectors are described using SIFT and their performances evaluated against viewpoint changes, scale changes, blur and JPEG compression artifacts. The results largely depend on the type of scene used for the experiments, with none of the detectors clearly outperforming others for all the types of scenes and transformations. Viewpoint change was found to be the most difficult type of transformation to cope with, followed by scale change.

Several methods have been designed specially for document images. Wichello and Yan [23] proposed a simple method for mosaicing binary documents using a cross-correlation match. It was assumed that apriori knowledge of image placement and overlap are available. It was also assumed that there is no warping thus limiting its use to scanned documents only. Pilu and Isgrò [24] introduced a two-stage approach for mosaicing scanned documents using a corner detector described in [25] called SUSAN (Smallest Univalue Segment Assimilating Nucleus). They used an intensity-based cross-correlation technique to compute an initial transformation hypothesis, which is then used to gather more supporting matches.

Zappala et al [26] proposed a mosaicing technique where the user slides the paper to be mosaiced under a stationary, over-the-desk camera until the whole document have passed through the field of view of the camera. In their method, first the skew is corrected and then the image is segmented into a hierarchy of columns, lines and words. Point correspondences are then established by matching the lower right hand corners in pairs of overlapping images. Lian et al [20] have proposed a 2-step approach for mosaicing without restricting the motion of the camera, thus allowing greater flexibility than scanner-based or fixed-camera-based approaches. Firstly, perspective distortion and relative rotation are removed by mapping the vanishing points of text line direction and vertical character stroke directions to points at infinity. Then, principal component analysis-SIFT is employed to establish feature correspondence. Finally, accurate registration is obtained by a cross-correlation block matching. The above methods are suitable only for documents which are predominantly text and have well-behaved page layouts so that the document may be easily segmented into lines, columns and words.

2.4 CCD: Connected component descriptor

Unlike the above methods, we use features derived from CCs that are found in abundance in any document image and can easily be computed irrespective of the page layout. The use of CCs is a natural choice for localization of 'interest points/regions' in document images since they are highly stable, unaffected by rotation, scale and other deformations. We use an augmented feature for resolving ambiguities in feature matching that arise due to multiple similar regions in the document image. In this work, no knowledge of the camera parameters is assumed and the success of the algorithm solely depends on the robustness of the features.

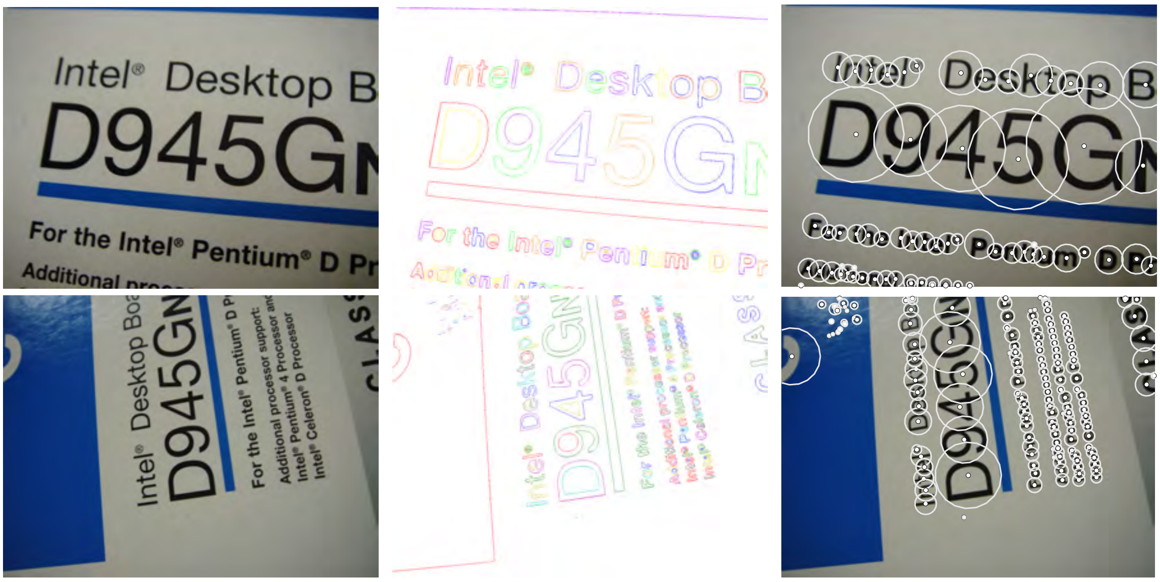
We introduce a new region descriptor derived from connected components. The key advantage of the new descriptor is that the same CC can be detected across different images of a document captured under different viewing conditions. Thus, the new descriptor inherently has an excellent repeatability rate, which is highly a desirable requirement for establishing point-to-point image correspondence.

2.4.1 Localization of measurement regions

We use a robust method of obtaining CCs from the edge image so that characters having different polarity of foreground-background intensities can be handled [27]. Canny edge detection [28] is performed individually on each channel of the color image and the overall edge map $\mathbf{E}$ is obtained by combining the three edge images as follows:

$$\mathbf{E} = \mathbf{E}_R \vee \mathbf{E}_G \vee \mathbf{E}_B \tag{2.8}$$

Here, $\mathbf{E}_R$, $\mathbf{E}_G$ and $\mathbf{E}_B$ are the edge images corresponding to the three color channels and $\vee$ denotes the logical OR operation. This simple method yields the boundaries of all the characters present in the document image irrespective of its color, size or orientation. A 8-connected component labeling follows the edge detection step and the associated bounding box information is computed. Edge detection yields both the inner and outer boundaries of the characters. However, it suffices to describe only the region bounded by



(a) (b) (c)

Figure 2.2: (a) Input images (b) Connected components (color-coded) derived from the edge image (c) Measurement regions computed from each connected component. It may be observed that the measurement regions exhibit a high repeatability rate, thanks to the stability of CCs against different imaging conditions.

the outer boundary. Hence, we filter out all the inner boundaries while retaining the outer boundaries of the characters. The centroid of each CC is used for localizing the region of interest for feature extraction. The position of these centroids are very stable and remain invariant under 2-D affine transformations. The measurement region is identified as the smallest circle, centered at the computed centroid, that just encloses the CC as shown in Fig. 2.2(c). All the measurement regions are then normalized to a standard size for feature extraction. It may be observed that all the measurement regions identified in the first image are re-detected in the second image in spite of the large variations in illumination, image rotation, perspective distortion and scale changes. The measurement regions exhibit a high repeatability rate, thanks to the stability of CCs against different imaging conditions.

2.4.2 Extraction of angular radial transform features

Though the feature may be any invariant descriptor, we have chosen angular radial transform (ART) because of its desirable properties like compact size, robustness to noise, scaling and deformation, invariance to rotation and ability to describe complex objects [29, 30]. The ART is the complex orthogonal unitary transform defined on a unit disk that consists of the complete orthonormal sinusoidal basis functions in polar coordinates. The ART descriptor, proposed by Kim and Kim [29] in 1999, is adopted in the MPEG-7 standard for shape coding. The ART coefficients of order m and n are defined as follows:

$$\begin{aligned} \mathbf{F}_{m,n} &= \langle \mathbf{B}_{m,n}(\rho,\theta)\mathbf{I}(\rho,\theta)\rangle \\ &= \int_0^{2\pi}\int_0^1 \mathbf{B}_{m,n}(\rho,\theta)\,\mathbf{I}(\rho,\theta)\,\rho\,d\rho\,d\theta \end{aligned} \tag{2.9}$$

where $\mathbf{I}(\rho,\theta)$ is the image function in polar coordinates, $\mathbf{B}_{m,n}(\rho,\theta)$ is the ART basis function of order m and n. These ART basis functions are defined in polar coordinates and are separable along the radial and angular directions.

$$\mathbf{B}_{m,n}(\rho,\theta) = \mathbf{R}_m(\rho)\,\mathbf{A}_n(\theta) \tag{2.10}$$

where m and n are non-negative integers. The radial basis is defined by a cosine function as follows:

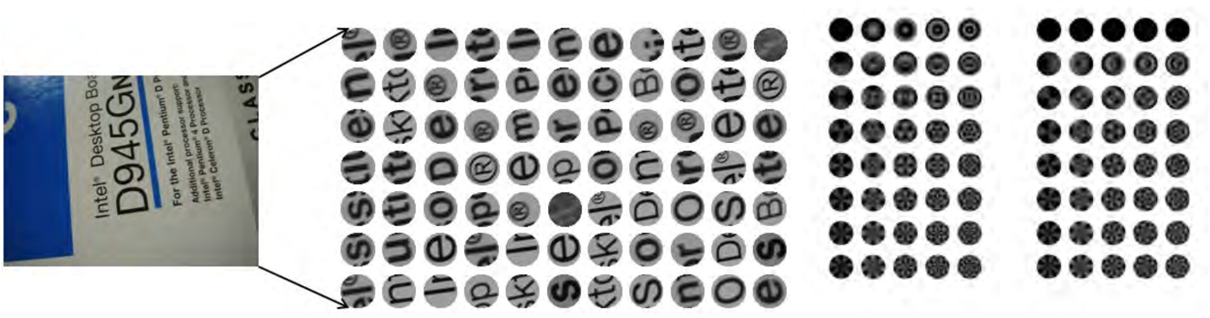
$$\mathbf{R}_m(\rho) = \begin{cases} 1 & m = 0 \\ 2\cos(\pi m\rho) & \text{otherwise} \end{cases} \tag{2.11}$$

To achieve invariance to rotation, an exponential function is used as the angular basis function.

$$\mathbf{A}_n(\theta) = \frac{1}{2\pi}\exp(jn\theta) \tag{2.12}$$

Let $\mathbf{I}^{\phi}(\rho,\theta)$ denote the image $\mathbf{I}(\rho,\theta)$ rotated by angle ϕ about its origin; then we have the following relation:

$$\mathbf{I}^{\phi}(\rho,\theta) = \mathbf{I}(\rho,\theta-\phi) \tag{2.13}$$



(a) (b) (c)

Figure 2.3: Extraction of invariant features. (a) Measurement regions computed from each connected component. The (b) real and (c) imaginary parts of the ART basis functions of order m = 0,1,. . . , 4 and n = 0,1, . . . , 7. Each measurement region is described by 39 normalized ART coefficients.

The ART coefficients of the rotated image are given by:

$$\begin{aligned}\mathbf{F}^{\phi}_{m,n} &= \int_0^{2\pi}\int_0^1 \mathbf{B}_{m,n}(\rho,\theta)\,\mathbf{I}^{\phi}(\rho,\theta)\,\rho\,d\rho\,d\theta \\ &= \mathbf{F}_{m,n}\exp(-jn\phi) \end{aligned} \tag{2.14}$$

Thus, the magnitudes of the ART coefficients of the rotated image and that of the reference image are the same.

$$\|\mathbf{F}^{\phi}_{m,n}\| = \|\mathbf{F}_{m,n}\| \tag{2.15}$$

In our implementation, we compute the ART coefficients of order $m = 0, 1, \ldots, 4$ and $n = 0, 1, \ldots, 7$ for each of the identified measurement regions. While the number of radial and angular orders are set to be 3 and 12 respectively in [29], our experiments show a better performance with 5 and 8 for characters. To achieve rotational invariance, the magnitude of the ART coefficients are used as the feature vector. The coefficient $\|\mathbf{F}_{0,0}\|$ is used as a normalization factor yielding a 39-dimensional vector for each CC.

2.4.3 Augmented feature for matching

Feature matching across different views is one of the difficult problems in computer vision. There are always some false matches, mainly due to similar and repeated structures in the images. Because of the local region of support employed in feature-based methods, there is no discrimination between multiple similar regions. Many a times, the conventional method of matching a feature to the 'best one' is found to be wanting, when applied to document images, due to the presence of multiple occurrences of the same character. To address this problem, we augment the ART coefficients of a CC with those of its 2 nearest neighbors (NN). This augmented feature, which we call connected component descriptor (CCD), has the all robust characteristics of ART as well as additional local geometric constraints that enhances the success of feature matching. Then, correspondence is established using these augmented features.

The distance between two ART feature vectors $\mathbf{F}_1$ and $\mathbf{F}_2$ is calculated using L_1 norm as follows:

$$d_{12} = Dist(\mathbf{F}_1, \mathbf{F}_2) = \sum_{d=1}^{39} |\mathbf{F}_1(d) - \mathbf{F}_2(d)| \tag{2.16}$$

We employ a 2-way feature matching strategy. In the first step, for every feature vector $\mathbf{F}_i$ in the reference image, the 'best' match amongst all the feature vectors extracted from the target image is given by the nearest neighbor in the feature space.

$$NN_{ij} = \text{argmin}_j \{Dist(\mathbf{F}_i, \mathbf{F}_j)\} \tag{2.17}$$

These matched features from the target image are then subjected to another round of feature matching in the reverse direction; i.e. the corresponding 'best' matches in the reference image are determined. We declare two feature vectors as a matched pair only if they have a mutual nearest neighbor relationship.

2.5 Estimation of homography

In practice, there are inevitably some measurement regions which do not have a match in the other image. More importantly, the set of matched features, identified as mentioned above, often contain a lot of incorrect ones. These outliers can drastically affect the homography estimate and consequently should be filtered out. The RANdom SAmple Consensus (RANSAC) algorithm proposed by Fischler and Bolles [31] is a simple and powerful method for estimating the correct model parameters, given a data set highly contaminated with outliers. The method considers many random subsets, each containing the minimum number of samples required to compute the model parameters, and selects the parameter set that yields the largest consensus i.e the largest number of inliers. The consensus set $|\mathbf{S}_k|$ for an estimated homography $\mathbf{H}_k$ is the number of data points that satisfy $\mathbf{X}'_i = \mathbf{H}_k\mathbf{X}_i$. This is obtained by comparing the distance between a data point and its projected point against a threshold i.e $|\mathbf{X}'_i - \mathbf{H}_k\mathbf{X}_i| < T$. The threshold value T is set to 4 pixels in our experiments. If the process is repeated a sufficiently large number of times, it can so happen that one of the random selections is free from outliers and the resulting model therefore has a large consensus. The RANSAC algorithm for estimating homography given m point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{X}'_i, i = 1, 2, .., m$ is as follows:

```
1.  Set iteration number $k = 1$, consensus set $|S_0| = 0$
2.  Randomly select 4 point matches
3.  Estimate homography $\mathrm{H}_k$ using Eqn. 2.5
4.  Count the consensus set $|\mathrm{S}_k|$
5.  If $|\mathrm{S}_k| > |\mathrm{S}_{k-1}|$, optimal homography $\mathrm{H}^* = \mathrm{H}_k$
6.  $k = k + 1$
7.  While $k < n$, repeat Steps 2 to 6
```

At the end of n iterations, the above procedure returns the optimal homography $\mathbf{H}^*$ which corresponds to the model that has the highest consensus set. All the points from the largest consensus set are then used to compute the least squares estimate of the homography using Eqn. 2.7.

Using the estimated homography, we perform guided feature matching by looking for the best match in the neighborhood of a projected point. We accept the match to be correct if the distance of the best match is less than $(\mu_d + \sigma_d)$, where μ_d and σ_d are the mean and standard deviation of the NN distances of the matched pairs in the highest consensus set. Finally, a refined estimate of $\mathbf{H}$ is computed using all the consistent matched pairs.

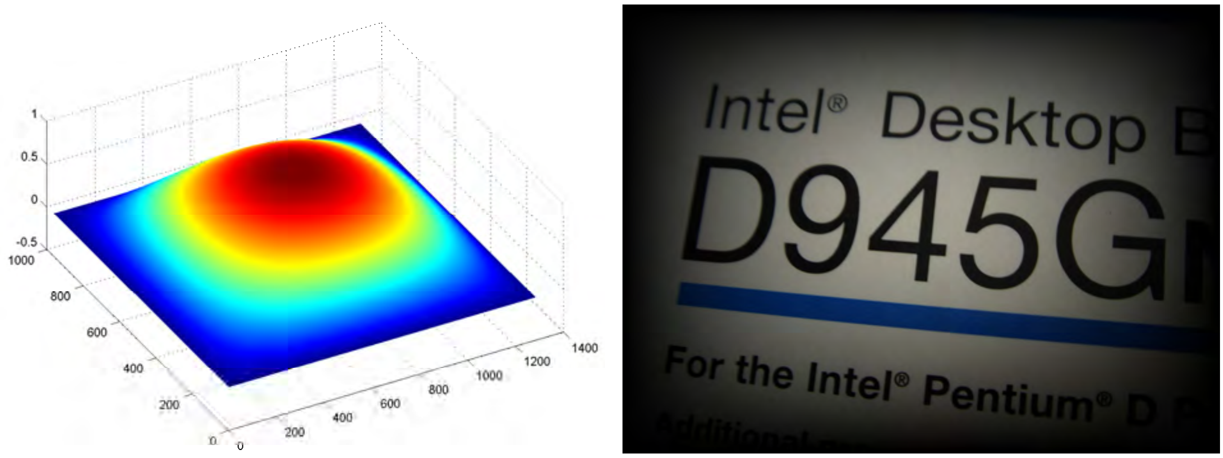
The number of trials n is chosen sufficiently large to ensure with a probability p that at least one of the random trial is free from outliers. If w denotes the probability that any selected point-pair is an inlier, then $e = (1 - w)$ is the probability that the selected data pair is an outlier. Thus, we need at least n random number of selections, each of s points ($s = 4$ in the case of fitting a homography), where

$$\begin{aligned} (1 - w^s)^n &= 1 - p \\ \text{or, } n &= \frac{\log(1 - p)}{\log(1 - (1 - e)^s)} \end{aligned} \tag{2.18}$$

2.6 Warping and blending

The final step of mosaic generation is to project the input images onto a common coordinate system (chosen as that of the reference image). The homography maps integer pixel locations to real-valued positions. Since the pixels are defined to lie on a discrete integer lattice, this is an issue while compositing discrete images. The inverse mapping technique is commonly used which projects each pixel location in the reference frame onto the target frame via the inverse mapping $\mathbf{H}^{-1}$. An interpolation technique such as bilinear interpolation is used to obtain the image attributes (gray scale intensity/color) at the non-integer mapped position in the target image. The interpolated value is then copied onto the reference frame. Bilinear interpolation takes a weighted average of the values of the four neighboring pixels from the mapped point to arrive at its interpolated value. Denoting the mapped point by (x_m, y_m) and the image intensities at its four neighbors by $I(x, y), I(x, y + 1), I(x + 1, y)$ and $I(x + 1, y + 1)$, the interpolated value is obtained using the following relation [32].

$$\begin{aligned} I(x_m, y_m) &= [1 - \Delta x][1 - \Delta y]I(x, y) + [1 - \Delta x]\Delta y I(x, y + 1) \\ &\quad + \Delta x[1 - \Delta y]I(x + 1, y) + \Delta x \Delta y I(x + 1, y + 1) \end{aligned} \tag{2.19}$$



(a) (b)

Figure 2.4: (a) Bi-quadratic weighting function used for blending the registered images (b) Each pixel in an image weighted by the chosen function.

where $\Delta x = (x_m - x)$ and $\Delta y = (y_m - y)$. Finally, the registered images are composited using biquadratic blending functions $W_k(x, y)$ that have unit value at the respective (k^{th}) image centers and progressively diminish to zero towards the image boundaries.

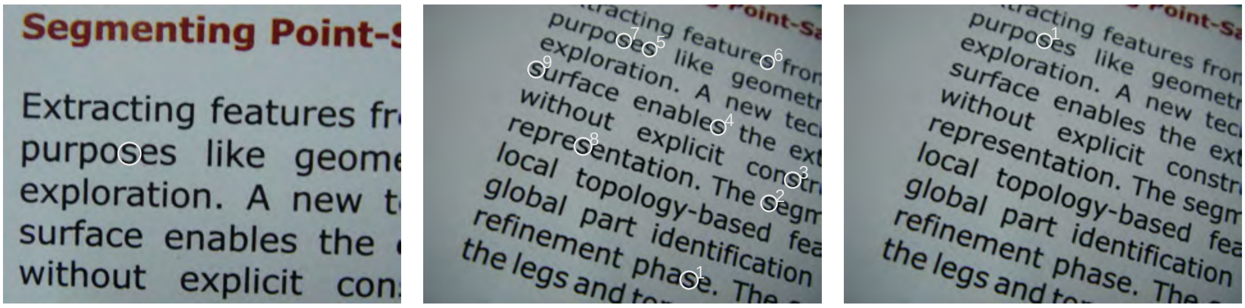
$$W_k(x,y) = \left[1 - \left(\frac{x - x_{c,k}}{x_{c,k}}\right)^2\right]\left[1 - \left(\frac{y - y_{c,k}}{y_{c,k}}\right)^2\right] \tag{2.20}$$

where the index k refers to the k^{th} image, $1 \leq x \leq \mathrm{M}, 1 \leq y \leq \mathrm{N}$ with M and N denoting the dimensions of the image and $(x_{c,k}, y_{c,k})$ is the location of the center of k^{th} component image.

The mosaic (of n images) is obtained by weighting each pixel of the input images at the location (x, y) by the respective blending functions as follows:

$$I_{out}(x,y) = \frac{\sum_{i=1}^{n} W_i(x,y) I_i(x,y)}{\sum_{i=1}^{n} W_i(x,y)} \tag{2.21}$$

This blending process eliminates the intensity discontinuities at the image boundaries, resulting in a seamless mosaic.



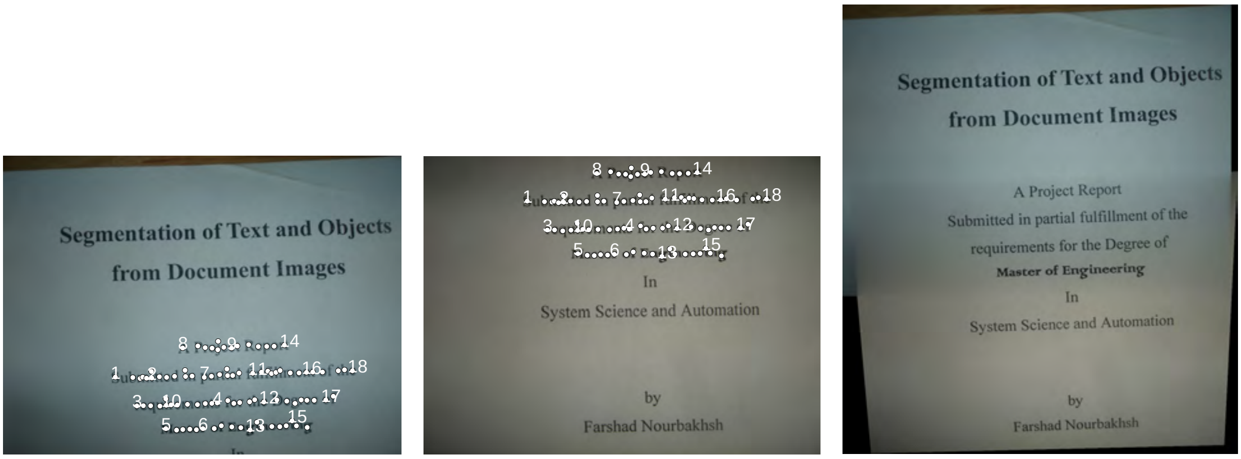
(a) (b) (c)

Figure 2.5: Illustration of matching with and without augmented features. The best 9 matches of a component ‘s’ indicated in (a) are shown along with the orders of their ‘similarity’ scores in (b). In the presence of multiple similar regions, the best match is seldom the correct one. However, using the augmented features, the best match, as indicated in (c), is the correct one. The use of augmented features significantly increases the number of correct matches.

2.7 Experiments and results

The proposed method is tested on a number of document images acquired using a hand-held camera at a resolution of 1280 × 960. Performing a CC labelling on the edge image, all the characters in the document, irrespective of the polarity of their foreground and background intensities, are identified for description. We have used the values of 0.2 and 0.3 as thresholds for the hysteresis thresholding step of Canny edge detection. The CCs and their associated measurement regions are identified and normalized to a standard size of 33 × 33. Feature vectors are matched in pairs across images. A component from the reference image is declared a correct match to one from the target image if they have mutual nearest neighbor relationship.

Fig. 2.5 illustrates the effectiveness of augmented ART features over plain ART features. Fig. 2.5(b) shows the best 9 matches of a component ‘s’ indicated in Fig. 2.5(a) along with their ranks of ‘similarity’; all of them correspond to the same character. This situation is highly probable while matching document images. Clearly, choosing the ‘best’ match would lead to a lot of false matches, that render even robust algorithms like RANSAC ineffective. On the other hand, using the augmented feature, correspondence can be successfully established (see Fig. 2.5(c)) even when there are several instances of the same letter in the document. The 2-NN constraint, imposed by the augmented

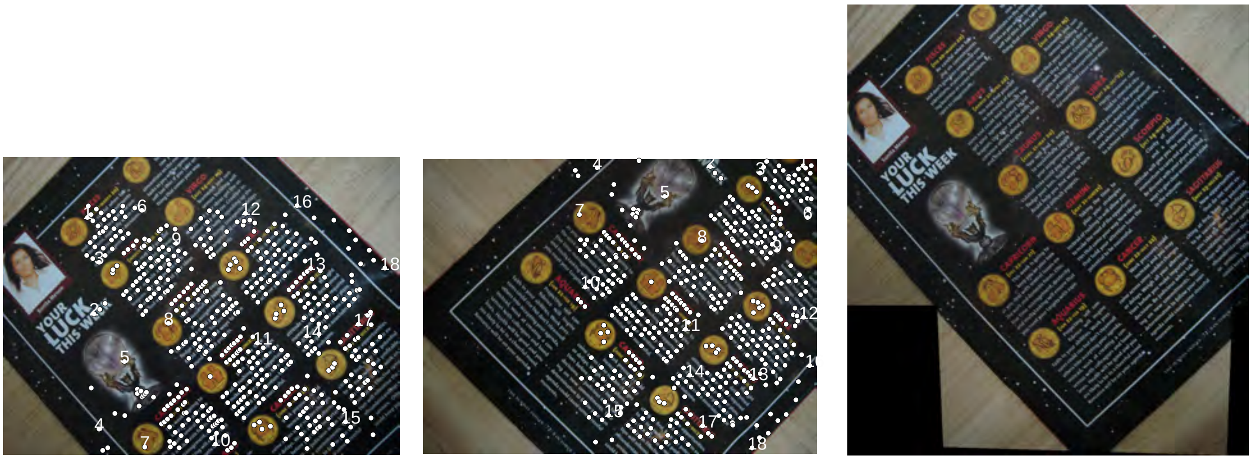

Segmentation of Text and Objects
from Document Images
A Project Report
Submitted in partial fulfillment of the
requirements for the Degree of
Master of Engineering
In
System Science and Automation
by
Farshad Nourbakhsh

(a) (b) (c)

Figure 2.6: Result of feature matching across a pair of images (a) and (b) exhibiting translation with different exposures. Of the 94 correctly matched components, only a few of them are indicated for the sake of clarity. The corresponding mosaic is shown in (c).

features, resolves the ambiguity due to the presence of multiple similar components. This significantly increases the proportion of inliers in the putative matches. The few cases of wrong matches that may arise due to repeated words are effectively handled by RANSAC algorithm.

Since CCD is computed for every CC irrespective of its location, it is invariant to translation. In addition, it has all the robust characteristics of ART. Size normalization of the measurement regions effectively handles large changes in scale. Fig. 2.6 shows the result of feature matching for an image pair exhibiting translation and different exposure. The matched pairs obtained using CCD are indicated by overlaying same numbers on the corresponding points on the respective input images. Fig. 2.7 demonstrates the rotation invariant property of CCD. Feature correspondence is successfully established across images with a large rotation difference of about 90°. In Fig. 2.8, we consider images taken under general viewing conditions that can have rotation, scale and perspective distortion. A large number of feature matches is obtained, as desired, from the regions common to both the input images. Using the computed feature correspondences, the input images are registered. The final mosaiced output image, obtained after blending, is shown along with the corresponding input image pair.



(a) (b) (c)

Figure 2.7: The matched pairs (521 in number) obtained using CCD for an input image pair (a) and (b) having a large relative rotation of about 90°. For clarity, only a few of the matched pairs are labeled. The corresponding mosaiced output is shown in (c).

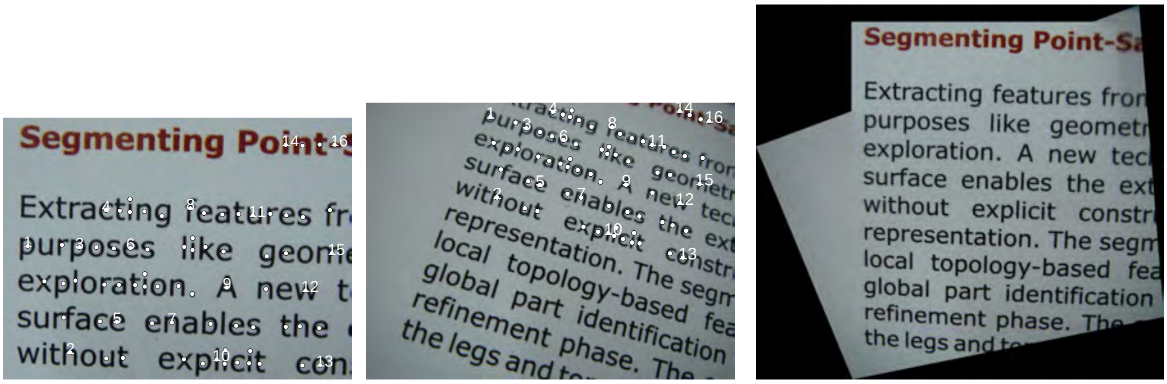


Figure 2.8: Successful feature matching across a pair of images with composite rotation, scale and perspective distortion. The matched pairs obtained using CCD are indicated by overlaying same numbers on the corresponding points on the input images. Only a few of them are labeled for better visibility. The corresponding mosaiced output is shown on the right.

In addition to being robust to photometric and geometric distortions, the method can also be applied to other scripts since CCD does not assume any characteristic of the script. Figure 2.9 shows the result of mosaicing for a pair of images containing English and Tamil text.

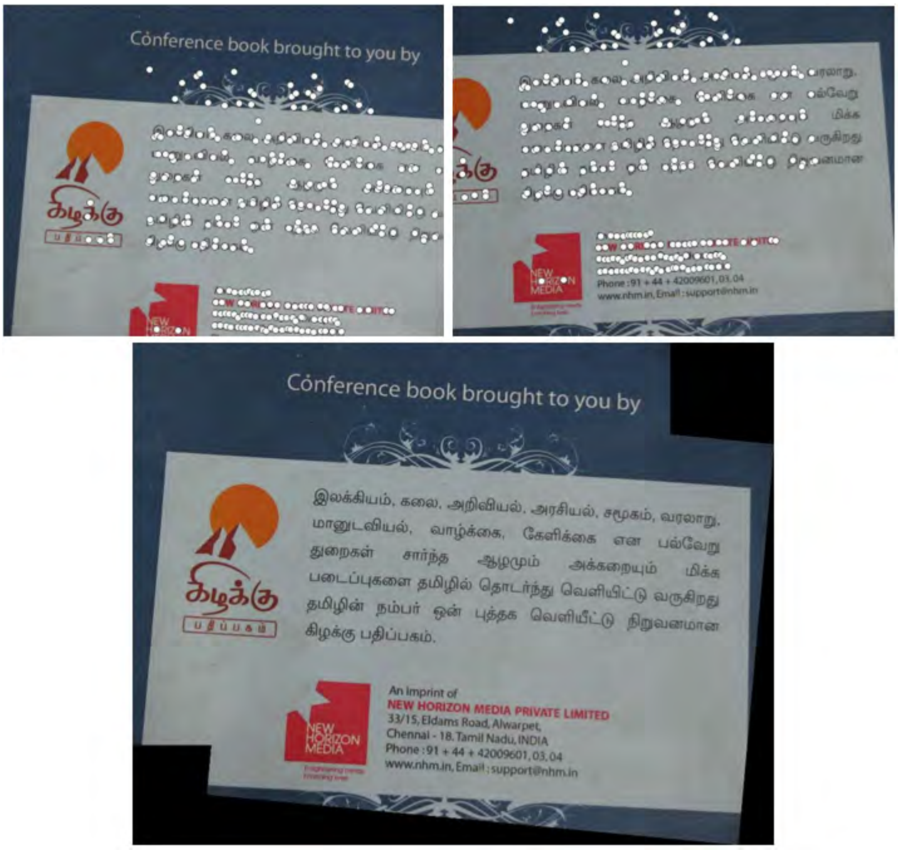

Cônference book brought to you by
An imprint of
NEW HORIZON MEDIA PRIVATE LIMITED
33/15, Eldams Road, Alwarpet,
Chennai - 18. Tamil Nadu, INDIA
Phone : 91 + 44 + 42009601, 03, 04
www.nhm.in, Email : support@nhm.in
NEW HORIZON MEDIA

Figure 2.9: Result of mosaicing a pair of images containing multiple scripts.

2.8 Conclusions

We have presented a new region-based descriptor, well suited for document image mosaicing. Connected components are natural candidates for localization of features in document images since they are highly stable and largely invariant to geometric and photometric distortions. The proposed method is thus guaranteed to have stable feature localization, which is a critical requirement in all feature-based approaches. The robustness of CCD is amply illustrated by our experiments. The discriminative power of CCD is further enhanced by augmenting it with those of its geometric neighbors. This ensures successful feature correspondence even in the case of occurrence of the same character in multiple locations in the document. However, the method may fail for images of low resolution and natural images, where connected components cannot be accurately identified.

Chapter 3

Hierarchical CCA and contour-based color clustering for color text binarization

"It doesn't matter if a cat is black or white, as long as it catches mice." - *Deng Xiaoping*

Abstract:

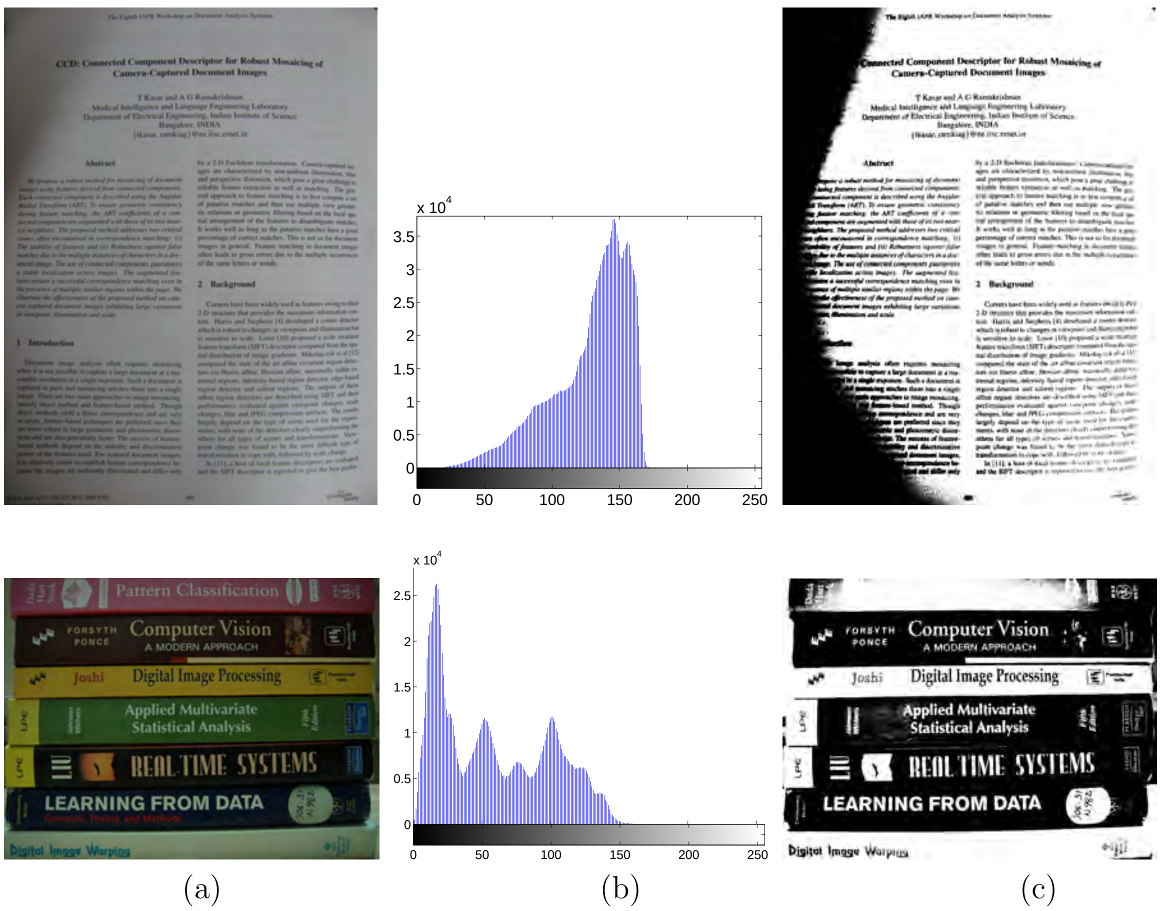
Complex color documents with both graphics and text, where the text varies in color and size call for specialized binarization techniques. This chapter presents edge-based connected component approach for binarizing such documents, where an adaptive threshold is used for each connected component. The threshold is automatically estimated from the image data without the need for any manual input parameter. The method is applicable to documents having text of widely varying sizes, usually not handled by local binarization methods. Unlike other conventional binarization techniques, it does not require a priori knowledge of the polarity of foreground-background intensities. However, it is sensitive to complex backgrounds since it relies on edge information to identify CCs. We extend the method to handle these issues by introducing a novel color clustering method for accurate CC extraction. The proposed methods have been applied to a broad domain of target document

types and are found to have a good adaptability.

3.1 Introduction

In most document processing systems, a binarization process precedes the analysis and recognition procedures. The use of two-level information greatly reduces the computational load and the complexity of the analysis algorithms. It is critical to achieve robust binarization since any error introduced in this stage will affect the subsequent processing steps. While it is relatively easy to segment characters from clean scanned documents, camera-based images are difficult to process due to the less-constrained mode of image acquisition. They inherently suffer from various geometric and photometric distortions that are absent in scanned documents. Uneven illumination is common, due to the physical environment such as shadows or reflective surfaces and artificial lighting. Camera-captured document images seldom have a bimodal distribution and thresholding such images using a single global threshold can never yield accurate binary images.

Another issue is the presence of inverse text that requires prior knowledge of the polarity of foreground-background intensities. Global thresholding with a fixed foreground-background polarity will lead to loss of useful information as some of the text may be assigned as background. The characters once lost cannot be retrieved back and are not available for further processing. Solutions need to be sought to handle the presence of inverse text so that any type of document may be properly binarized without the loss of textual information. In addition, variations in font, size, color, script, layouts and complex backgrounds pose challenges to document analysis and recognition. Fig.3.1 shows two sample camera-captured document images, the corresponding gray scale histograms and the results of global thresholding to illustrate the effects of non-uniform illumination and the presence of inverse text.



(a) (b) (c)

Figure 3.1: Effects of non-uniform illumination and presence of inverse text on global thresholding. (a) Sample camera-captured images with non-uniform illumination and multi-colored textual content (b) The corresponding gray level histograms (c) Outputs of Otsu thresholding. Clearly, a single fixed threshold does not work well for the entire image. Moreover, the assumption of a fixed polarity for the foreground-background intensities results in loss of textual information where some of the characters are assigned as background pixels.

3.2 Review of document binarization techniques

Document image binarization has been a vastly researched area. The simplest and earliest method is the global thresholding technique that is generally based on histogram analysis and statistical methods [33, 34, 35, 36, 37]. A single threshold classifies the image pixels into foreground or background classes. It is simple, fast and works well for scanned images that have well-separated foreground and background intensities. However, a global method has the disadvantage that it does not have any spatial discrimination and cannot handle non-uniform illumination. As such, it is not suitable for camera-captured images.

Local methods that use a dynamic threshold according to the local image statistics offer more robustness to non-uniform illumination and the complexity of the background. These approaches are generally based on a window and the threshold for a pixel is computed from the statistics of the gray values within the window centered at that pixel. One of the most widely-used local binarization method is the one proposed by Niblack [38] where the sample mean $\mu(x, y)$ and the standard deviation $\sigma(x, y)$ within a window W centered at the pixel location (x,y) are used to compute the threshold T(x, y) as follows:

$$T(x, y) = \mu(x, y) - k\,\sigma(x, y), \quad k = 0.2 \tag{3.1}$$

Yanowitz and Bruckstein [39] introduced a threshold that varies over different image regions so as to fit the spatially changing background and lighting conditions. Based on the observation that the gray level values at the edge points of the image are good choices for local thresholds, a threshold surface is created by relaxation initialized on the edge points. In [40], Trier and Jain evaluated 11 popular locally adaptive thresholding methods on scanned documents. In their evaluation, Niblack's method performed the best for OCR. However, the method produces a noisy output in homogeneous regions since the expected sample variance tends towards the background noise variance. To resolve this drawback, Sauvola and Pietikäinen [41] introduced a hypothesis that the gray values of the text are close to 0 (black) while the background pixels are close to 255 (white). The threshold is computed using an assumed dynamic range of standard deviation (R) which has the effect of modifying the contribution of standard deviation in an adaptive manner.

$$T(x, y) = \mu_{(x,y)}\,[1 + k\,(\frac{\sigma(x, y)}{R} - 1)] \tag{3.2}$$

where the parameters R and k are set to 128 and 0.5 respectively. This method minimizes the effect of background noise and is more suitable for document images.

Recently, Sezgin and Sankur [42] gave a comprehensive review of 40 binarization methods on document images degraded with noise and blur where Savoula's method was reported to be amongst the best performing local binarization algorithms. However, the method fails for images where the assumed hypothesis is not met and accordingly, Wolf and Jolion [43] proposed an improved threshold estimate by taking the local contrast measure into account:

$$T(x, y) = (1 - a)\mu(x, y) \;+\; aI_{min} + a\frac{\sigma(x, y)}{\sigma_{max}}\,[\mu(x, y) - I_{min}] \tag{3.3}$$

where I_{min} is the minimum value of the grey levels of the whole image, σ_{max} is the maximum value of the standard deviations of all windows of the image and a is a parameter fixed at 0.5. The Wolf's method requires two passes since one of the threshold decision parameter σ_{max} is the maximum of the standard deviations of all windows of the image. The computational complexity is therefore slightly higher in this case. This method combines Savoula's robustness with respect to background textures and the segmentation quality of Niblack's method.

Local methods offer more robustness to the background complexity, though at a cost of higher computational complexity. The performance of these methods depend on the size of the window used to compute the image statistics. They work well if the window encloses at least 1 character. For large fonts, where the text stroke is wider than the window, undesirable voids appear within the text stroke. This puts a constraint on the maximum font size and limits their application only to known document types. In addition, all these methods require a priori knowledge of the polarity of the foreground-background intensities and hence cannot handle documents that have inverse text.

Besides black text on white background, documents increasingly contain different levels of intensity and color for highlighting. Text may be represented in different colors or may be located within a highlight box that is intermediate in level between that of the text and the background. Such documents, when scanned/transformed to gray-scale, will usually have separate gray levels corresponding to the different colors. In such cases, one may require multiple thresholds to segment the text. Most multi-thresholding methods [44, 45, 46, 47] must know the number of background colors in advance and then determine appropriate threshold values from the color histogram. Such methods are suitable only for applications where the number of information levels is typically two or three, much smaller than the gray-scale quantization.

Color information is being increasingly used for the analysis of color documents. Badekas *et al.* [48] estimate dominant colors in the image and CCs are identified in each color plane. Text blocks are identified by CC filtering and grouping based on a set of heuristics. Each text block is applied to a Kohonen SOM neural network to output only two dominant colors. Based on the run-length histograms, the foreground and the background are identified to yield a binary image having black text on white background. The performance of this method relies on the accuracy of color reduction and text grouping, which are not trivial tasks for a camera-captured complex document image. The method does not consider isolated characters for binarization.

Sobottka et al. [49] presented a method for localization of text in color images that employs two methods in tandem - a top-down analysis by successive splitting of image regions and a bottom-up region growing algorithm. It is assumed that the text is horizontal and each segmentation scheme tries to find subsets of regions which are aligned horizontally. The resulting hypothesis for text regions are verified by combining the results of both methods. Text regions are binarized using the extracted text color information. Accurate identification of CCs is the key to the success of the above algorithms. Their performance significantly degrades, like most CC-based methods do, in the presence of complex backgrounds that interfere with the accurate identification of CCs.

In this chapter, we present two methods that address the issues discussed above.

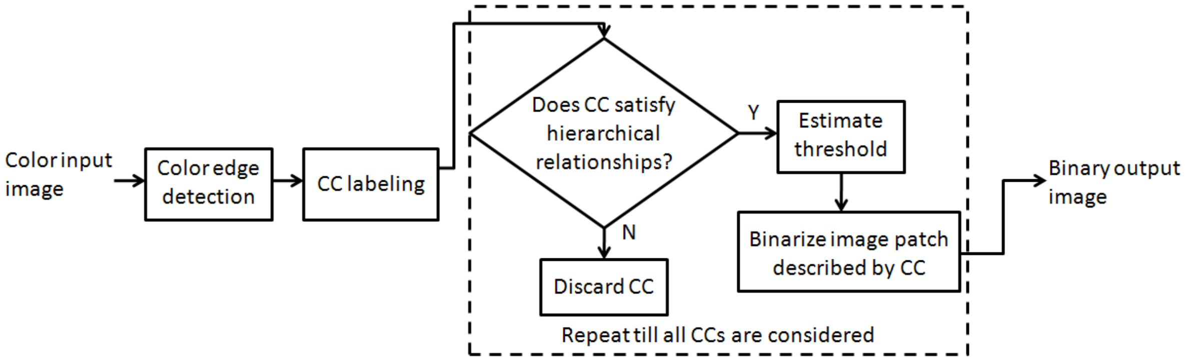

Color input image
Color edge detection
CC labeling
Does CC satisfy hierarchical relationships?
Y
Estimate threshold
Binarize image patch described by CC
N
Discard CC
Repeat till all CCs are considered
Binary output image

Figure 3.2: Schematic block diagram of HCCA for color text binarization.

Our first method [27] involves the use of the hierarchical relationships of text boundaries from the edge image. The method is discussed in detail in Section 3.3. The method can deal with arbitrary font size, color and effectively handles the presence of inverse text. However, the method is sensitive to complex backgrounds since it relies on the edge information to identify CCs. It also uses script-specific characteristics to filter out edge components before binarization and works well only for Roman script. In Section 3.4, we describe our second approach [50] that overcomes these limitations. We use a novel, unsupervised color clustering approach that operates on a 'small' representative set of color pixels identified using the contour information. Based on the assumption that every character is of uniform color, we analyze each color layer individually and identify potential text regions for binarization.

3.3 HCCA: Hierarchical connected component analysis for color text binarization

Text is the most important information in a document. We propose a novel method to binarize camera-captured color document images that uses an edge-based connected component approach to automatically obtain a threshold for each component. Figure 3.2 shows a schematic block diagram of the proposed binarization method.

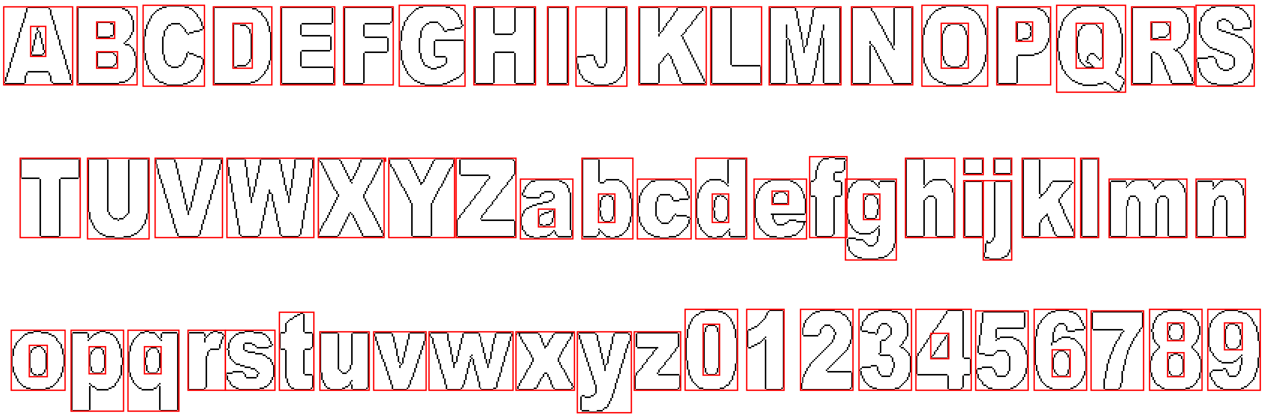


Figure 3.3: Hierarchical relationships of outer and inner text boundaries. Edge-boxes for the English alphabet and Indo-Arabic numerals are shown. The letter 'B' has one outer edge contour (parent component) and two other components (children) due to the inner edges. Note that there is no character that completely encloses more than two edge components.

3.3.1 Detection of color edges

We use the monochromatic approach to color edge detection owing to its simplicity. Canny edge detection is performed individually on each channel of the color image and the overall edge map **E** is obtained by combining the three edge images as described before in Section 2.4.1. This simple method yields the boundaries of all the characters present in the document image irrespective of its color, size or orientation. An 8-connected component labeling follows the edge detection step. We make some sensible assumptions about the document and use the area and the aspect ratios of the components to filter out the obvious non-text regions. The aspect ratio is constrained to lie between 0.1 and 10 to eliminate highly elongated regions. The area of the component should be greater than 15 pixels but smaller than 1/5th of the image dimension to be considered for further processing. These heuristic parameters significantly reduce the computational load by eliminating the obvious non-text elements.

3.3.2 Hierarchical CC analysis of edge-components

Since edge detection captures both the inner and outer boundaries of the characters, a particular connected component may completely enclose one or more other components. For example, the letter 'O' gives rise to two components; one due to the outer boundary (parent component) and the other due to the inner boundary (child component). A careful observation of the parent-child relationship of the edge components of the English alphabet and Indo-Arabic numerals (See Fig. 3.3) shows that no character has more than two children. This hierarchical relationship of the text boundaries leads us to a simple yet effective mechanism to filter out obvious non-text components while retaining all text-like components. If a particular component has exactly one or two children, the children can be conveniently ignored as they correspond to the inner boundaries of the character. On the other hand, if it has three or more children, only the children are retained while the parent is removed since such a component does not represent a character. However, this hierarchical relationship of the text boundaries holds only for isolated Latin characters and may not hold for other scripts. The pseudo code for the edge-component filtering is given below:

```
For $i = 1$ to $(\#CC)$
$N^i_{child}$ = Number of children of $CC^i$
      if ($N^i_{child}$ <3)
         {Reject $CC^i_{child}$, Accept $CC^i$}
     else
         {Reject $CC^i$, Accept $CC^i_{child}$}
     endif
endFor
```

where CC^i_{child} refers to the components that lie completely inside the i^{th} component under consideration CC^i and N^i_{child} is the number of children of the i^{th} component. These filtered set of components are then processed individually for binarization.

3.3.3 Estimation of binarization threshold

The edge-based CC approach captures all the characters, irrespective of their sizes thereby enabling us to perform local binarization without the need to specify any window. We estimate the foreground and background (FB) intensities for each of the filtered components and accordingly the threshold too. Fig. 3.4 shows the foreground and the background pixels, which are used for obtaining the threshold and decision for the inversion of the binarized component. The foreground intensity of a particular component CC^i is computed as the mean gray-level intensity of the pixels that correspond to the edge pixels.

$$I_{FG}^i = \frac{1}{N_E^i} \sum_{(x,y)\in\mathbf{E}^i} I(x,y) \tag{3.4}$$

where $\mathbf{E}^i$ represents the set of edge pixels, $I(x,y)$ represents the intensity value at the pixel (x,y) and N_E^i is the number of edge pixels of the component. For obtaining the background intensity, we consider a set of twelve pixels, three pixels each at the periphery of each corner of the bounding box as follows:

$$\begin{aligned}\mathbf{BG}^i \;=\; & \{I(x-1,y-1), I(x-1,y), I(x,y-1), I(x+w+1,y-1), I(x+w,y-1),\\ & I(x+w+1,y), I(x-1,y+h+1), I(x-1,y+h), I(x,y+h+1),\\ & I(x+w+1,y+h+1), I(x+w,y+h+1), I(x+w+1,y+h)\} \qquad (3.5)\end{aligned}$$

where (x,y) represents the coordinates of the top-left corner of the bounding-box of each edge component and w and h are its width and height, respectively.

Fig. 3.5 shows the output of the hierarchical CC filtering and the threshold parameters for each of the valid edge components. As observed in Fig. 3.5(c), the mean and median intensities are almost the same for the foreground pixels irrespective of the text orientation. However, for diagonally aligned text, the adjacent bounding boxes may overlap and interfere in the estimation of the background intensity. This is the case for the text 'FLEXIBLE 6 CHANNEL' printed in black in a semi-circular manner. Fig. 3.5(c) shows that their edge components have estimated foreground intensity lower than that of the

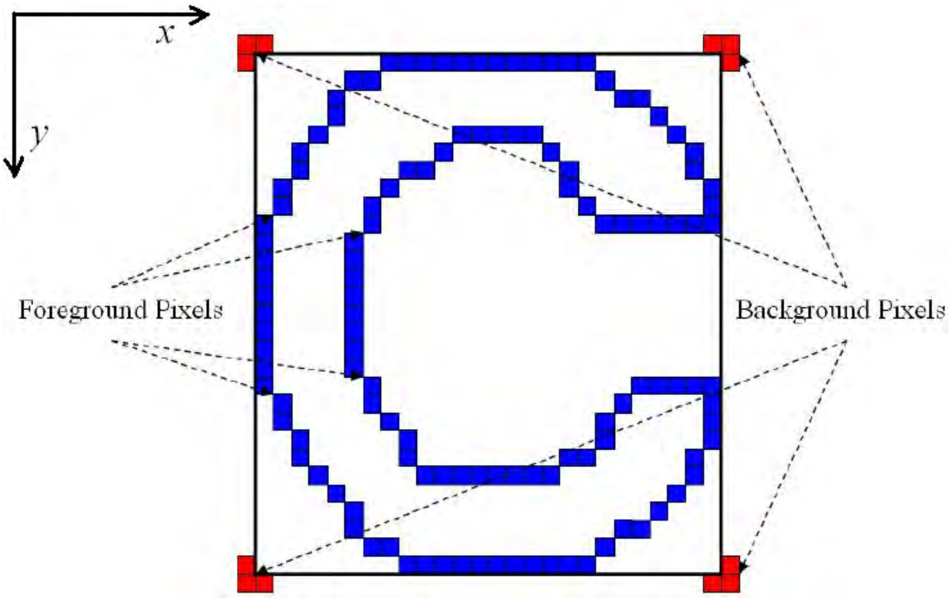


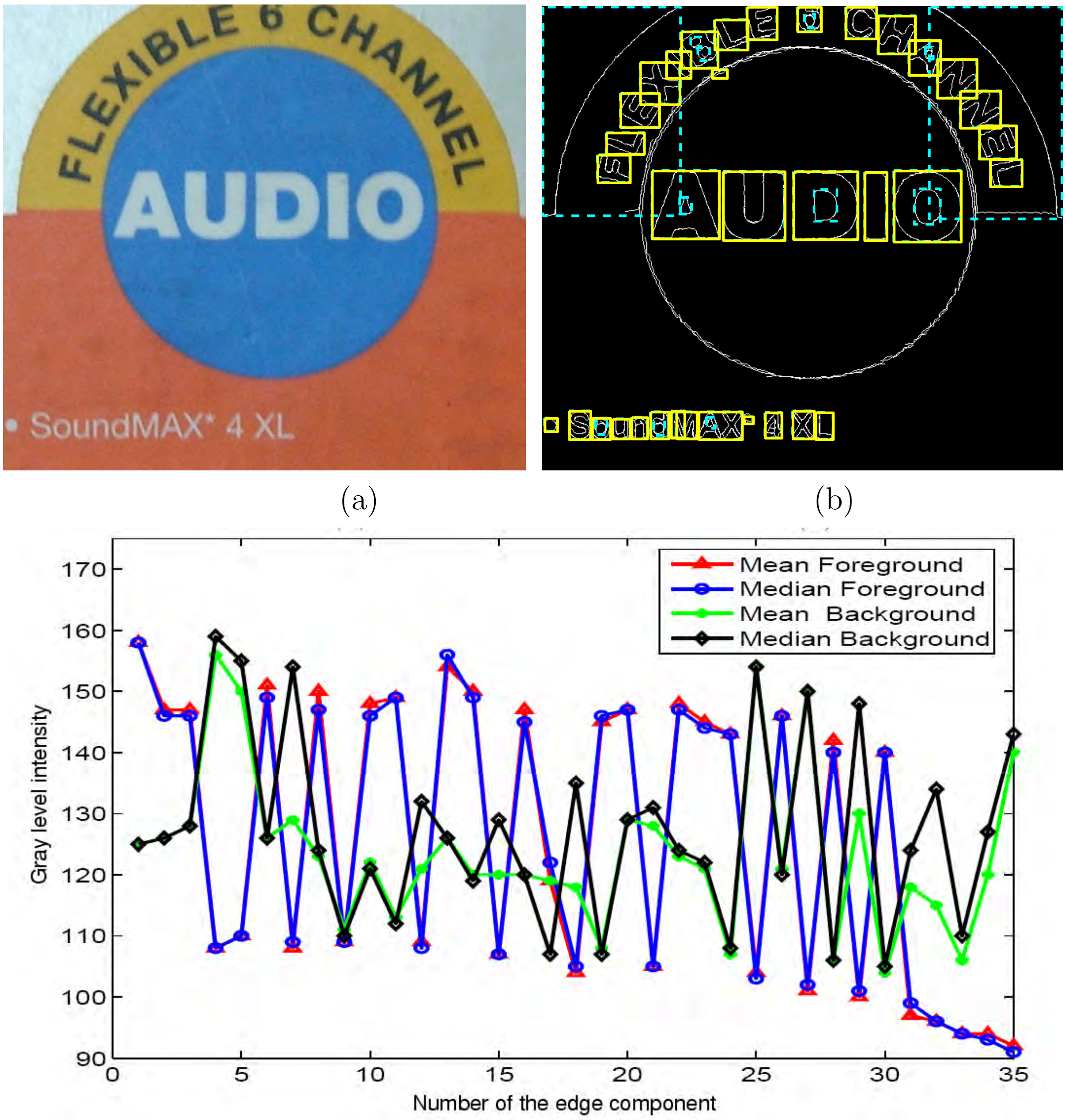
Figure 3.4: The foreground and the background pixels of each component used for deriving the value of the binarization threshold and to decide on the need to invert the binary output.

background. The mean background intensities of these components are affected by the adjacent components while the medians are not. Thus, the local background intensity is computed as the median intensity of the 12 background pixels.

$$I^i_{BG} = \text{median}(\mathbf{BG}^i) \tag{3.6}$$

Assuming that each character is of uniform color, we binarize the gray-scale image patch $R_{CC^i}(x, y)$ described by the component using the estimated foreground intensity as the threshold. For light text on a dark background, the estimated foreground intensity is greater than that of the background. By comparing these two estimates, the polarity of FB intensities can thus be determined. The binarized output B_{CC^i} is inverted, if required, so that the foreground text is always black and the background, always white irrespective of their original colors.

$$\text{If } I^i_{FG} < I^i_{BG},\ B_{CC^i}(x, y) = \begin{cases} 1,\ R_{CC^i}(x, y) > I^i_{FG} \\ 0,\ R_{CC^i}(x, y) \le I^i_{FG} \end{cases} \tag{3.7}$$



(a) (b)

(c)

Figure 3.5: Choice of binarization threshold and deciding on inversion. (a) An input Image (b) Output of hierarchical CC filtering. The dotted boxes in cyan are filtered out and only the yellow solid boxes (35 in number) are considered for binarization (c) The threshold parameters for the valid edge components. Observe that the mean and median intensities of the foreground pixels are almost the same for all characters. The same holds true for the background estimate for horizontally (or vertically) aligned text. However, when the text is aligned diagonally, the mean intensity of the background pixels is affected due to overlapping of the adjacent bounding boxes. The median intensity gives a more reliable logic for inverting the binary output.

$$\text{If}\, I^i_{FG} \geq I^i_{BG},\, B_{CC^i}(x,y) = \begin{cases} 0,\, R_{CC^i}(x,y) > I^i_{FG} \\ 1,\, R_{CC^i}(x,y) \leq I^i_{FG} \end{cases} \tag{3.8}$$

The threshold value is automatically derived from the image data and the method is thus completely free from user-defined parameters. It simultaneously handles the ambiguity in the relative polarity of the FB intensities. However, the method is sensitive to complex backgrounds since it relies on the edge information to identify CCs. It also uses script-specific characteristics to filter out non-text components before binarization and is best suited for isolated Roman letters.

3.4 COCOCLUST: Contour-based color clustering for color text binarization

This section describes a novel contour-based color clustering technique that overcomes the sensitivity of the edge-based CC method to complex backgrounds. Rather than operating on the entire image, a 'small' representative set of color pixels is first identified using the contour information. This significantly reduces the computational load of the algorithm since their number is orders of magnitude smaller than the total number of pixels in the image. A single-pass clustering is then performed on the 'reduced' color values to identify color clusters that serve as seeds for a subsequent clustering step using the k-means algorithm. Based on the assumption that every character is of a uniform color, we analyze each color layer individually and identify potential text regions for binarization. Figure 3.6 shows the schematic block diagram of the proposed binarization method.

3.4.1 Determination of representative colors

The segmentation process starts with the color edge detection obtained using Eqn. 2.8. The resulting edge image **E** gives the boundaries of all the homogeneous color regions present in the image. An 8-connected component labeling is performed on the edge image

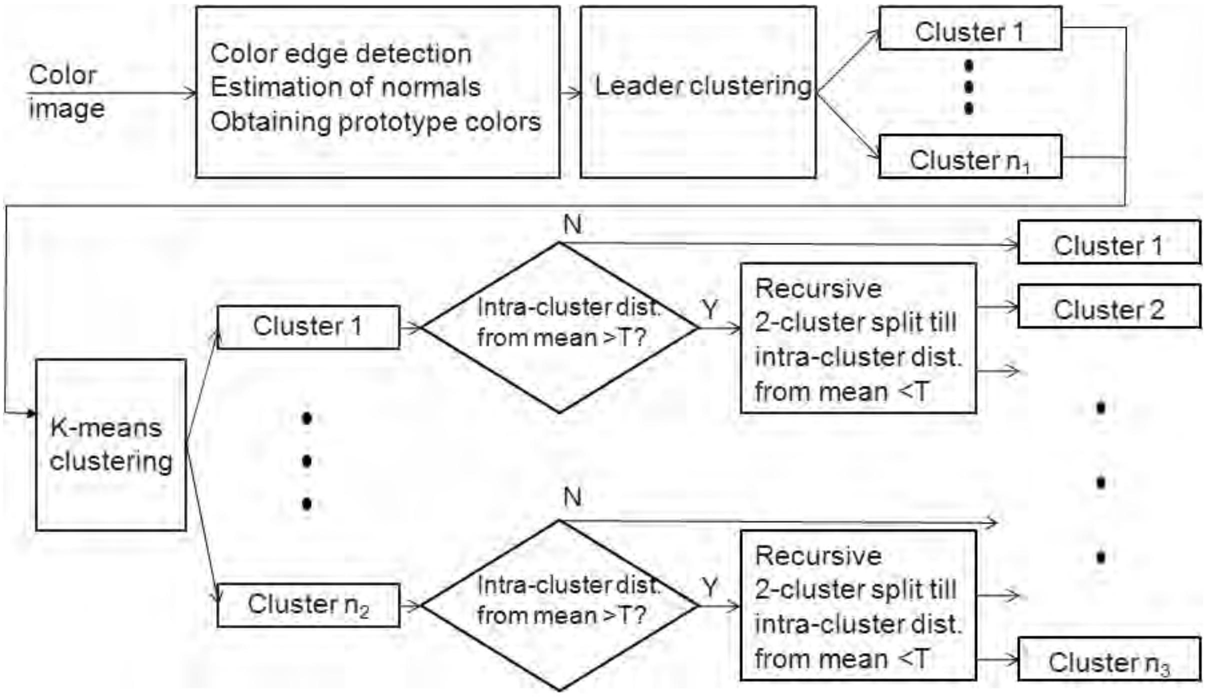


Figure 3.6: Block schematic of COCOCLUST for color text binarization.

to obtain a set of m disjoint components as follows:

$$\{CC^i\} \quad i = 1,\, 2\,, \cdots\,, m \text{ such that} \bigcup_{i=1}^{m} CC^i = \mathbf{E} \tag{3.9}$$

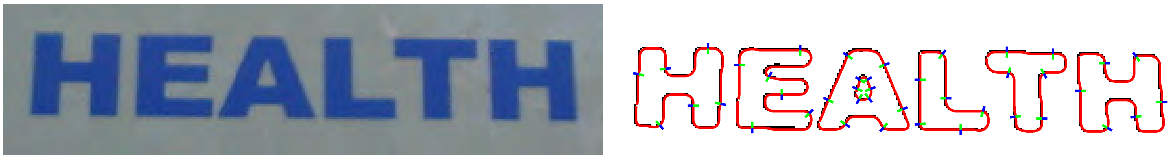
The boundary pixels are identified and represented as follows:

$$\mathbf{X}_j^i = \{x_j, y_j\} \quad j = 1,\, 2,\, \cdots\,, n_i \tag{3.10}$$

where n_i is the number of pixels that constitutes the boundary of the i^th connected component CC^i. Our method employs a few vectors normal to the edge contour for every CC. To estimate the normal vector, the edge contour is smoothed locally as follows:

$$\bar{\mathbf{X}}_j^i = \left\{ \frac{1}{s} \sum_{j-\frac{s-1}{2}}^{j+\frac{s-1}{2}} x_j,\, \frac{1}{s} \sum_{j-\frac{s-1}{2}}^{j+\frac{s-1}{2}} y_j \right\} \tag{3.11}$$

where s defines the span of pixels over which smoothing is performed and is set to 5 in



(a) (b)

Figure 3.7: Minimizing data for color clustering by choosing color prototypes. (a) A sample color image (b) Its edge contours and the computed normals that guide the selection of color prototypes. From each normal, one color value each is obtained from the pixels that lie 'inside' (green segment) and 'outside' (blue segment) the contour.

this work. Here, the index j takes a circular convention to maintain continuity of the contour. The normal vectors are then computed from the smoothed contour using the following relation.

$$\mathbf{n}_j^i \quad = \quad \begin{pmatrix} cos(\frac{\pi}{2}) & -sin(\frac{\pi}{2}) \\ sin(\frac{\pi}{2}) & cos(\frac{\pi}{2}) \end{pmatrix} \times \frac{1}{2} \left(\frac{\bar{\mathbf{X}}_j^i - \bar{\mathbf{X}}_{j-1}^i}{\parallel \bar{\mathbf{X}}_j^i - \bar{\mathbf{X}}_{j-1}^i \parallel} + \frac{\bar{\mathbf{X}}_{j+1}^i - \bar{\mathbf{X}}_j^i}{\parallel \bar{\mathbf{X}}_{j+1}^i - \bar{\mathbf{X}}_j^i \parallel} \right) \tag{3.12}$$

Here, the subscript j denotes the position of the boundary pixel at which the normal vector is computed and $\||\cdot\||$ denotes L_2 norm.

Since the edge image gives the boundaries of homogeneous color regions, the color values of a few pixels that lie normal to the contour are 'good' representatives of the colors present in the image. Figure 3.7(a) shows a sample color image and Figure 3.7(b) illustrates the selection of color values from a few pixels that lie normal to the edge contour. From each normal, we compute the median color of 3 pixels each that lie 'inside' $(\mathbf{n}_-^i)$ and 'outside'$(\mathbf{n}_+^i)$ the contour to obtain two color values. The common RGB color format is not suitable for color grouping tasks because it is not expressed in the way perceived by humans. Two different pairs of points with the same Euclidean distance in the RGB color space do not result in the same change in the perceived color. So, we use a uniform color space, namely CIE $L^*a^*b^*$, in which similar changes in color distance correspond to similar recognizable changes in the perceived color. The conversion from

RGB color space to that of $L^*a^*b^*$ is accomplished by the following relations [51]:

$$\begin{aligned} L^* &= 116Y^* - 16 \\ a^* &= 500(X^* - Y^*) \\ b^* &= 200(Y^* - Z^*) \end{aligned} \tag{3.13}$$

where

$$X^* = \begin{cases} \sqrt[3]{\frac{X}{X_n}} & \text{for } \frac{X}{X_n} > 0.008856 \\ 7.787(\frac{X}{X_n}) + 0.138 & \text{for } \frac{X}{X_n} \leq 0.008856 \end{cases}$$

Y^* and Z^* are similarly obtained by replacing X, X_n by Y, Y_n and Z, Z_n, respectively. (X_n, Y_n, Z_n) describes the white reference point and X, Y and Z are defined in terms of RGB values as follows:

$$\begin{aligned} X &= 0.490R + 0.310G + 0.200B \\ Y &= 0.177R + 0.812G + 0.011B \\ Z &= 0.000R + 0.010G + 0.990B \end{aligned}$$

The conversion between device-dependent color spaces such as RGB into device-independent ones such as XYZ is performed using an illuminant as the white point, which is dependent on the lighting condition. CIE Illuminant D50 is considered as the default illuminant. The color samples (CS), thus obtained, are stacked column-wise as follows.

$$CS = \left\{ \begin{array}{ccccc} L^*_{1-} & L^*_{1+} & \cdots & L^*_{N-} & L^*_{N+} \\ a^*_{1-} & a^*_{1+} & \cdots & a^*_{N-} & a^*_{N+} \\ b^*_{1-} & b^*_{1+} & \cdots & b^*_{N-} & b^*_{N+} \end{array} \right\}$$

where N is the number of normals along which the color values are sampled. In this work, we sample the color values from 6 regularly spaced points along the boundary of each CC yielding a total of 12 m colors for the entire image. This set of color values, though much smaller in number than the total number of pixels, captures all the colors present in the image. This offers a significant advantage in terms of less computation and provides an effective initialization of k-means algorithm regardless of the complexity of image content.

3.4.2 Unsupervised color clustering

A single-pass clustering is performed on color prototypes, as obtained above, to group them into clusters. The pseudo-code for the clustering algorithm is given below.

```
Input:Color Samples,CS = $\{C_1, C_2, \ldots, C_{2N}\}$
      Color similarity threshold, $T_s$
Output:Color clusters, CL
Assign CL[1] = $C_1$ and Count = 1
For i = 2 to 2N, do
   For j = 1 to Count, do
      If Dist(CL[j],$C_i$)$\leq$ $T_s$
         CL[j] = Mean($\{$CL[j],$C_i\}$))
         Break
      Else
         Count = Count + 1
         CL[Count] = $C_i$
      EndIf
   EndFor
EndFor
```

where $\mathrm{Dist}(\mathrm{C}_1, \mathrm{C}_2)$ denotes the distance between the colors $\mathrm{C}_1 = (L_1^*,\, a_1^*,\, b_1^*)^{\mathrm{T}}$ and $\mathrm{C}_2 = (L_2^*,\, a_2^*,\, b_2^*)^{\mathrm{T}}$ given by:

$$\mathrm{Dist}(\mathrm{C}_1,\, \mathrm{C}_2) = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2} \tag{3.14}$$

The threshold parameter T_s decides the similarity between the two colors and hence the number of clusters. Antonacopoulos and Karatzas [52] perform grouping of color pixels based on the criterion that only colors that cannot be differentiated by humans should be grouped together. The threshold below which two colors are considered similar was experimentally determined and set to 20. We use a slightly higher threshold to account for the small color variations that may appear within the text strokes. In our implementation, the threshold parameter T_s is empirically fixed at 45, after experimentation.

The color clusters, thus obtained, are used as seeds for a subsequent clustering step using the k-means algorithm. Each resulting cluster is then examined for 'compactness' by computing distances of all the pixels in that cluster from its mean color. The maximum intra-cluster distance from its mean color is ensured to be less than 75 % of T_s, by recursively splitting non-compact clusters, if any, into two using k-means algorithm initialized with the mean color and the one that is furthest from it. The color clusters obtained at the end of this splitting process are then used as the seed colors for a final pass of k-means clustering. Note that the whole clustering process is performed only on the selected prototypes. Finally, each pixel in the original image is assigned to the closest color cluster.

3.4.3 Adaptive binarization

Each color layer is separately analyzed and text-like components are identified and individually binarized. We first perform heuristic filtering of CCs based on size and aspect ratios as described before in Section 3.3.1. Overlapping CCs are resolved by retaining only the dominant component.

Estimation of binarization threshold

We employ an approach similar to HCCA to binarize each CC by estimating its foreground and background intensities. The foreground intensity I^i_{FG} of a component CC^i is computed from the edge pixels as before using Equation 3.4. Rather than using the bounding box to obtain an estimate of the background intensity, we use the available contour information that yields a more reliable decision for inversion in the presence of inverse text. Bounding boxes can have a significant overlap for inclined text and touching text lines that can result in incorrect inversion of the binary output. The contour is traced in a clock-wise direction and the normals are estimated. The background intensity is then computed as the median intensity value of the pixels along the normal direction 'outside' the boundary of the CC.

$$I^i_{BG} = \text{Median}(I(x,y)) \quad \text{where}\,(x,y) \in \mathbf{n}^i_+ \tag{3.15}$$

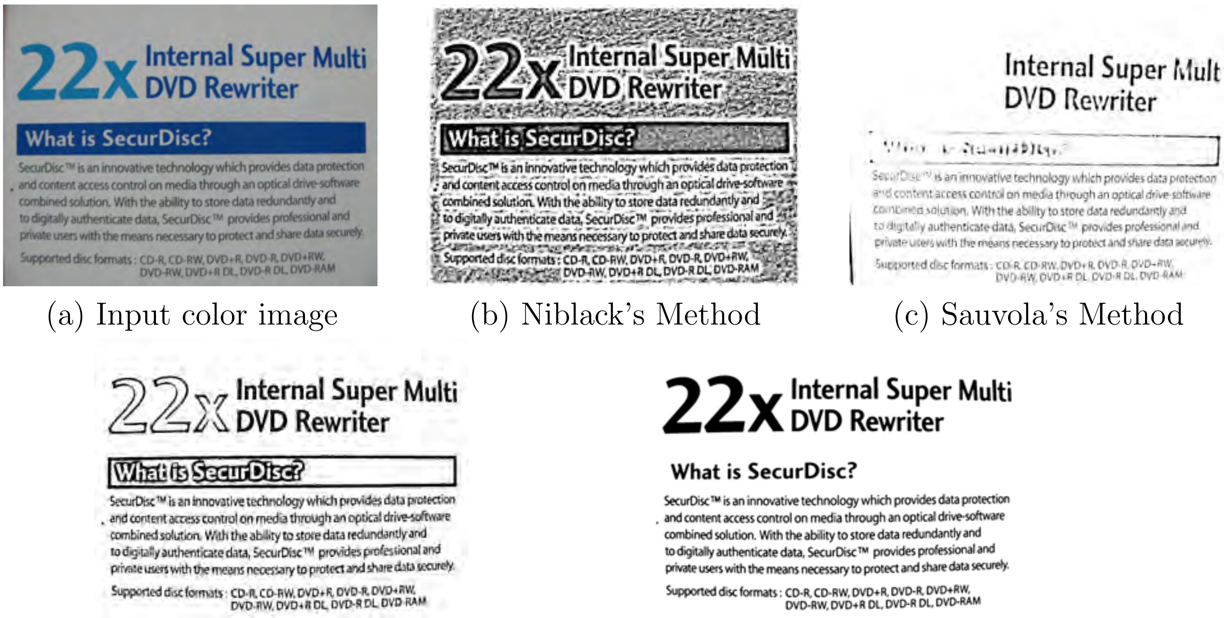
Note that the boundary of the CC is always 'closed' unlike during the prototype color identification stage, where we may have broken as well as bifurcating edge contours. The CC is binarized using the estimated foreground intensity as the threshold value.

$$B_i(x,y) = \left\{ \begin{array}{ll} 1 & \text{if } R_{CC^i}(x,y) > I^i_{FG} \\ 0 & \text{if } R_{CC^i}(x,y) \leq I^i_{FG} \end{array} \right\} \tag{3.16}$$

Whenever the estimated foreground intensity is higher that that of the background, the binary output is inverted as before to ensure that text is always represented by black pixels.

3.5 Experiments and results

The test images used in this work are acquired from a Sony digital still camera at a resolution of 1280 × 960. The images are taken from both physical documents such as book covers and newspapers and non-paper sources like text on 3-D real world objects. The connected component analysis performed on the edge map captures all the text characters irrespective of the polarity of their foreground and background intensities. The constraints on the edge components effectively remove the obvious non-text elements while retaining all the text-like components. From each valid edge component, the foreground and background intensities are automatically computed and each of them is binarized individually.


22x Internal Super Multi
DVD Rewriter
What is SecurDisc?
SecurDisc™ is an innovative technology which provides data protection and content access control on media through an optical drive-software combined solution. With the ability to store data redundantly and to digitally authenticate data, SecurDisc™ provides professional and private users with the means necessary to protect and share data securely.
Supported disc formats : CD-R, CD-RW, DVD+R, DVD-R, DVD+RW, DVD-RW, DVD+R DL, DVD-R DL, DVD-RAM

(a) Input color image (b) Niblack's Method (c) Sauvola's Method

(d) Wolf's Method (e) HCCA method

Figure 3.8: Comparison of some popular local binarization methods for a document image having multiple text colors and sizes. The proposed HCCA method is able to handle characters of any size. All other methods fail to properly binarize the components larger than the size of the window and require a priori knowledge of the polarity of foreground-background intensities as well.

Figure 3.8 compares the results of HCCA method with some popular local binarization techniques, namely, Niblack's method, Sauvola's method and Wolf's method on a document image having multi-colored text and large variations in sizes with the smallest and the largest components being 4×3 and 174×245 respectively. Clearly, these local binarization methods fail when the size of the window is smaller than the stroke width. The size of the window used here is 33 × 33. While small text regions are properly binarized, large characters are broken up into several components and undesirable voids occur within the character strokes. They require a priori knowledge of the polarity of foreground-background intensities as well. On the other hand, our method automatically derives the threshold from the image without any user-defined parameter. It can deal with characters of any font size and color since it only uses edge connectedness.

The HCCA method of binarization developed here is tested on documents having foreground text with different background shades. Though these kinds of images are

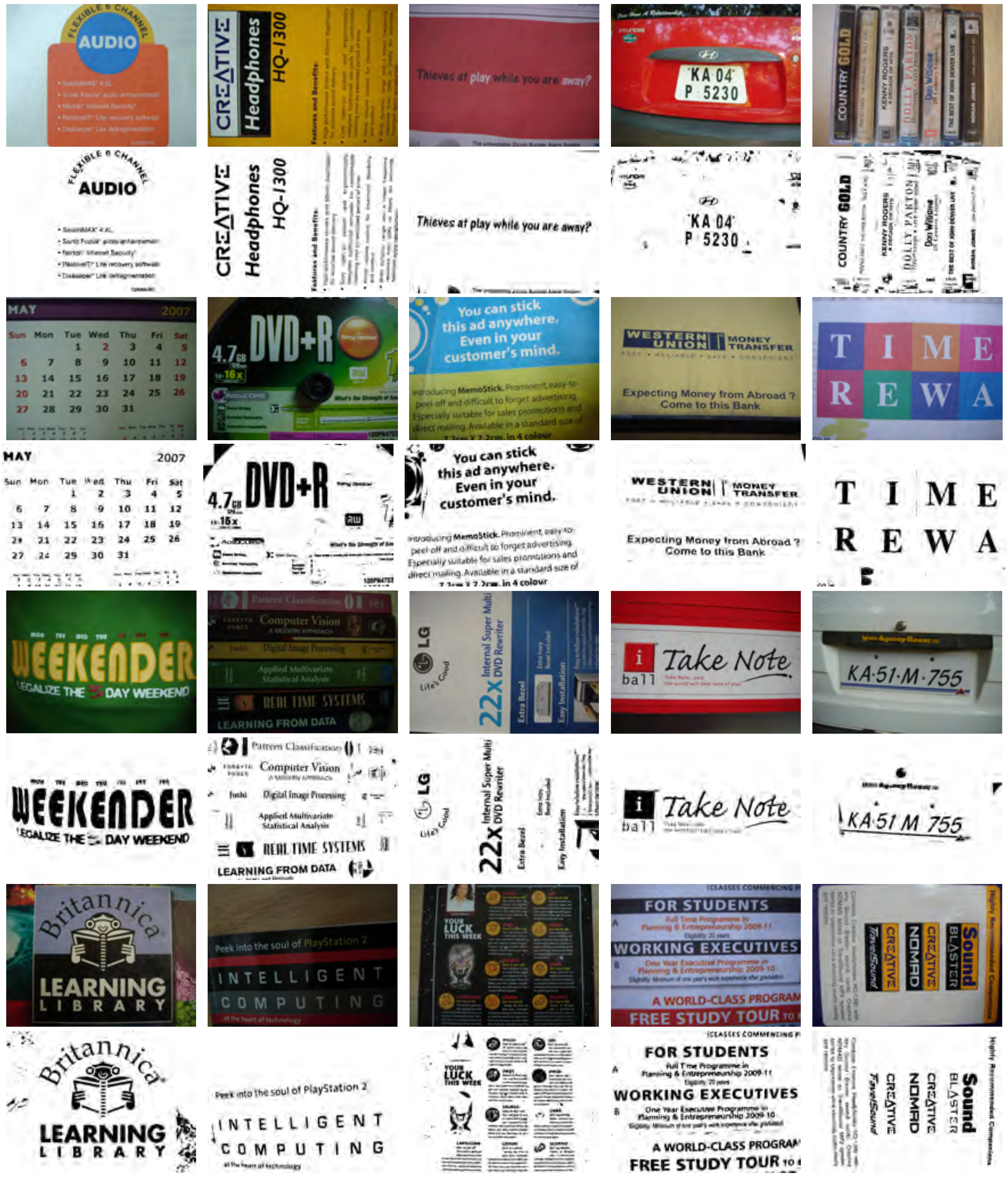
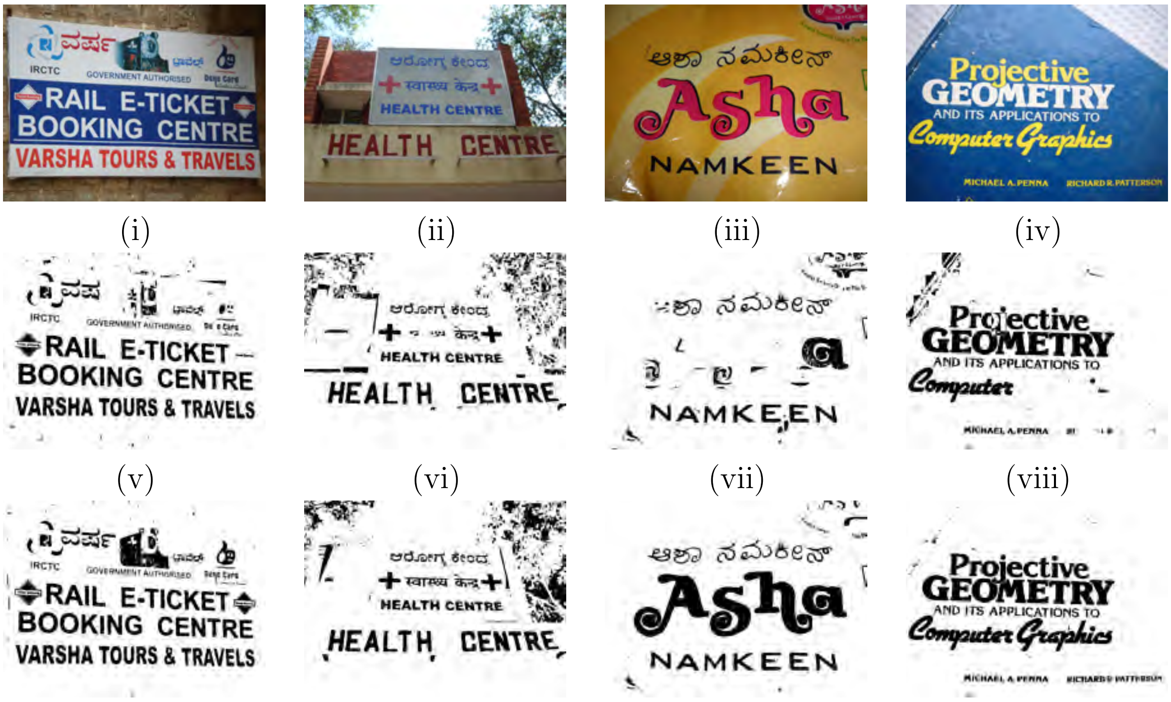


Figure 3.9: Representative results of binarization obtained using the HCCA method. Based on the estimated intensities of foreground and background, the appropriate binarized components are inverted so that all the text are represented in black and the background in white.



(i) (ii) (iii) (iv)

(v) (vi) (vii) (viii)

(ix) (x) (xi) (xii)

Figure 3.10: Improved results obtained by COCOCLUST over HCCA. (i - iv): Input images having graphic objects, inverse text, multiple scripts, complex backgrounds and cursive letters. (v - viii): The corresponding binary outputs obtained using the HCCA method. Clearly, the method is sensitive to background objects since it relies on the edge information to locate CCs. It also invokes script-dependent characteristics and works well only for isolated Roman letters. (ix - xii): Output images binarized using COCOCLUST. Color segmentation provides robustness to background complexity as well as independence to script.

quite commonly encountered, none of the existing binarization techniques can directly deal with such images. The generality of the algorithm is tested on more than 50 complex color document images and is found to have a high adaptivity and performance. Some results of binarization using HCCA are shown in Fig. 3.9, along with the respective input images. The algorithm deals only with the textual information and it does not threshold the edge components that were already filtered out. In the resulting binary images, as desired, all the text regions are output as black and the background as white, irrespective of their colors in the input images.

The HCCA method is sensitive to the background and occasional instances of text get filtered out since it uses an edge-based segmentation. Moreover, it uses script-dependent

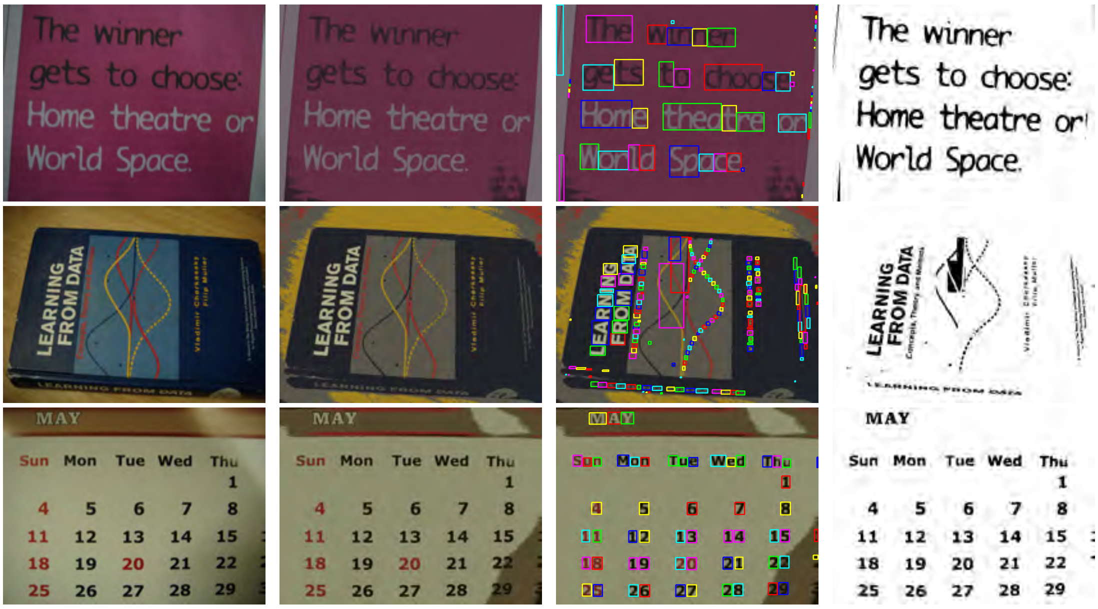


Figure 3.11: Additional results of COCOCLUST binarization. Input images, the corresponding color clusters, identified text regions and binarized outputs shown column-wise.

characteristics and works well only for isolated Roman letters. In Figure 3.10(i) and (iii), the background and graphic objects touch some text regions and they get filtered out in the corresponding output images (See 3.10(v) and (vii)). In Figure 3.10(vi-viii), the script-dependency of the method is clearly observed. In addition to the merged characters and cursive text being eliminated, Figure 3.10(viii) also shows an instance of incorrect inversion of the binary output due to overlapping text lines. In contrast, the contour-based color clustering approach shows a marked improvement thanks to the color clustering algorithm that enables an accurate identification of CCs (See Figure 3.10(ix-xii)). The color decomposition effectively disambiguates background objects interfering with text as they are separated into different color layers. Each CC is individually binarized based on a threshold derived from the estimates of its foreground and background intensities. The background intensity estimated using the contour normals provides a reliable decision to invert the binary output in the presence of inverse text. More results of the COCOCLUST on images that have inverse text, arbitrary text orientation and layout are shown in Figure 3.11.

However, the performance of the COCOCLUST degrades significantly in the presence

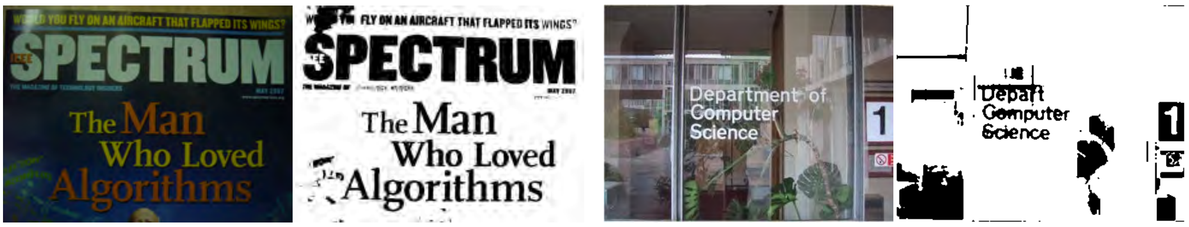


Figure 3.12: Failure cases of COCOCLUST: Some sample input images containing low resolution text, specular reflection and textured backgrounds and the corresponding outputs.

of low resolution text, blur and textured backgrounds since the edge information may not be reliable for such images. Some example failure cases are shown in Fig. 3.12.

3.6 Conclusions

This chapter describes an important preprocessing step for the analysis of color document images. We have developed two novel techniques for binarization of colored text. The first method employs an edge-based hierarchical CC approach and automatically determines a threshold for each component. It has a good adaptability without the need for manual tuning and can be applied to a broad domain of target document types and environment. It is free from any user-input parameter and can handle the presence of text of reverse polarity. The edge-based CC analysis captures all the characters, irrespective of their sizes thereby enabling us to perform local binarization without the need to specify any window. The use of CCs facilitates reliable, automated computation of the foreground and the background intensities and hence the required threshold for binarization. The proposed method retains the useful textual information more accurately and thus, has a wider range of applications compared to other existing binarization methods.

The edge detection method is good in finding the character boundaries irrespective of the foreground-background polarity. However, if the background is textured, the edge components may not be detected correctly due to edges from the background and CC filtering strategy fails. These limitations are overcome using a contour-based color segmentation that yields more accurate CC extraction in the presence of complex backgrounds.

It does not require a priori knowledge of the number of colors present or their initialization. The contour information is successfully exploited both in color segmentation that enables accurate identification of CCs and in the inversion of the binary output to deal with inverse text. The method is able to capture all the text while at the same time, eliminate most of the components due to the background. This is a desirable feature for processing camera-based images that are generally characterized by arbitrary content and layout. The results of our experiments on camera-captured images with variable fonts, size, color, orientation, script and the presence of inverse text are encouraging.

Chapter 4

Multi-script and multi-oriented text localization from scene images

"Seek and you will find, knock and the door shall be opened unto you." - The Bible

Abstract:*This chapter describes a maiden attempt at localizing multi-script and multi-oriented text from natural scene images. A representative set of colors is first identified using the edge information to initiate an unsupervised clustering algorithm. Connected components identified from each color layer are then verified using a combination of a support vector machine and a neural network classifier trained on a set of low-level features derived from the boundary, stroke and gradient information. By invoking the spatial regularity of text, adjacent text regions are further grouped to form words and text lines. Experiments on a created database of 100 camera-captured images that contain arbitrary orientation of text, variable font, size, color, irregular layout, non-uniform illumination and multiple scripts illustrate the robustness of the method. The proposed method yields precision and recall of 0.8 and 0.86 respectively on our database. The method is also compared with others in the literature using the ICDAR 2003 robust reading competition dataset.*

4.1 Introduction

Text represents the most important information in a document image. It provides useful semantic information that may be used to describe the image content. While it is relatively easy to segment characters from clean scanned documents, text extraction from natural scenes is difficult since scene text can appear on any surface, not necessarily on a plane. They are often characterized by arbitrary text layouts, multiple scripts, artistic fonts, colors and complex backgrounds.

The robust reading competition was held at the 7^{th} International Conference on Document Analysis and Recognition 2003 to find the system that best reads complete words in camera-captured images. The dataset contains various kinds of degradations such as uneven lighting conditions, complex backgrounds, variable font styles, low resolution and text appearing on shiny surfaces. However, there are no samples of inclined or multi-oriented text in the dataset. It is also limited to English. Recently, a separate competition has been organized for born-digital (web and e-mail) images as a part of the ICDAR 2011 robust reading competition but the dataset contains horizontal English text. In a multi-lingual country like India, it is common to find English words interspersed within sentences in Indic-script documents. Many documents, forms and signboards are generally bi-lingual or multi-lingual in nature. Every script has certain distinctive characteristics and may require script-specific processing methods. Therefore, the presence of multiple scripts require a special treatment.

There is a significant need for methods to extract and recognize text in scenes since such text is the primal target of camera-based document analysis systems. Scene text understanding normally involves a pre-processing step of text detection and extraction. The subsequent recognition task is performed only on the detected text regions so as to mitigate the effect of background complexity. Unlike the problem of classical object-driven image segmentation, such as separating sky from mountains, pixel-accurate segmentation is required for character recognition. Robust extraction of text is a critical requirement since it affects the whole recognition process that follows.

4.2 Review of text detection

The existing methods for text detection fall under the following two broad categories: Texture based methods and connected component based methods. Texture based methods exploit the fact that text has a distinctive texture. They use methods of texture analysis such as Gabor filtering, wavelets and spatial variance. Zhong et al. [53] compute local spatial variance in gray-scale images and locate text at high variance regions. Li and Doermann [54] use a sliding window to scan the image and classify each window as text or non-text using a neural network. Liu et al. [55] propose an edge-based approach for text localization. Sabari et al. [56] use multi-channel filtering with a Gabor filter bank on the gray-scale image. A graph-theoretical clustering is applied on the color space of the same image to identify iso-color components. The responses of the Gabor filters and color CC analysis are merged and text regions are obtained using geometrical and statistical information of the individual components.

Wu et al. [57] employ 9 derivative of Gaussian filters to extract local texture features and apply k-means clustering to group pixels that have similar filter outputs. Assuming that text is horizontally aligned, text strings are obtained by aggregating the filtered outputs using spatial cohesion constraints. Clark and Mirmehdi [58] apply a set of five texture features to a neural network classifier to label image regions as text and non-text. Chen and Yuille [59] extract features based on mean intensity, standard deviation of intensity, histogram and edge-linking and classify using an AdaBoost trained classifier. Shivakumara et al. [60] compute median moments of the average sub-bands of wavelet decomposition and use k-means clustering (k=2) to classify text pixels. The cluster with the higher mean is chosen as the text cluster. Boundary growing and nearest neighbor concepts are used to handle skewed text.

In CC based methods, the image is segmented into regions of contiguous pixels having similar characteristics like color, intensity or edges. These CCs are then classified using a set of features distinguishing textual and non-textual characteristics followed by grouping of the textual CCs. Robust segmentation of text CCs from the image is the most critical part in the process. Gatos et al. [61] segment both the original and the inverted gray-scale images by rough estimation of foreground and background pixels to form CCs which are further filtered using height, width and other properties of the CCs. Zhu et al. [62] use Niblack's method to segment the grayscale image into three layers, where the foreground and background layers are most likely considered to consist of text CCs. Each CC is described by a set of low level features and text components are classified using a cascade of classifiers trained with Adaboost algorithm. Pan et al. [63] have proposed a method to detect text using a sparse representation. Canny edges are extracted and grouped into CCs. Each CC is labeled as text or non-text by a two-level labeling process using an over-complete dictionary, which is learned from edge segments of isolated character images. Layout analysis is further applied to verify these text candidates.

Color information is being increasingly used for the analysis of newer document types, scene text in particular. Most approaches [64, 65, 66] involve clustering on the 3D color histogram followed by identification of text regions in each color layer using some properties of text. Song et al. [67] use two low level features namely intensity and color variances and text stroke features to locate text regions. An adaptive k-means color clustering is applied to the text blocks and the vertical projection profile is used to merge multiple color planes to obtain the desired binary image. Text blocks are grouped if they have more than 80% overlap; otherwise they are verified using an SVM classifier trained on wavelet features derived from 16 × 16 image blocks.

Garcia and Apostolidis [68] employ a recursive Deriche edge detector on each color channel and identify potential text areas by grouping CC based on an adaptive row and column distance tolerances. Color on each text block is quantized to yield 4 dominant colors and different binary images are obtained from each color component. Based on the vertical projection profiles (VPP), one of the binary images is classified as text. The method is tuned for horizontal text and the user-defined parameters cannot handle large variations in font size. Karatzas and Antonacopoulos [69] split the image into chromatic and achromatic regions based on histogram analysis and a tree structure of layers is created. CCs are then identified in the leaf layers of the tree structure. Subsequently, a merging process combines CCs of similar color and satisfactory interrelationship to progressively assemble characters in a bottom up fashion. Thillou and Gosselin [70] propose a selective metric based clustering (Euclidean and cosine based similarity) in RGB color space based on intensity and spatial information obtained using Log-Gabor filters.

CC-based approaches are suitable for camera-based images since they can deal with arbitrary font styles, sizes, color and complex layouts. However, their performance significantly degrades in the presence of complex backgrounds which interfere in the accurate identification of CCs. In this paper, we introduce a novel color clustering technique for robust extraction of CCs from complex images. A set of 'intrinsic' text features are designed from the geometric, boundary, stroke and gradient properties for classifying text CCs. Finally, adjacent text CCs are grouped together to form words, making use of the spatial coherence property of text.

4.3 Color text segmentation

Since the boundaries of characters are always closed, potential text candidates are identified from the regions bounded by closed edges. The segmentation process starts with color edge detection. The Canny edge detector normally yields a relatively clean edge image as compared to other edge detectors. However, since the gradient magnitude response is non-isotropic, the resulting edges are often broken during the hysteresis thresholding step. Therefore, we use a compass operator to compute the gradient magnitude in four directions and select the one that gives the maximum response. Each channel of the RGB color image is processed individually and the overall edge image is obtained by taking the union of the 3 edge images.

We perform an edge-linking procedure to bridge narrow gaps in the resulting edge image. The co-ordinates of the end points of all the open edges are then determined by counting the number of edge pixels in every 3×3 neighborhood and marking the center pixels when the count is 2. These pairs of edge pixels indicate the direction of the edge and are used to determine the direction for a subsequent edge-following and linking process. Depending on the arrangement of the two edge pixels, there can be 8 possible search directions namely North, North-East, East, South-East, South, South-West, West and North-West as shown below:

0	0	0
0	1	0
0	1	0

North

0	0	0
0	1	0
1	0	0

North-East

0	0	0
1	1	0
0	0	0

East

1	0	0
0	1	0
0	0	0

South-East

0	1	0
0	1	0
0	0	0

South

0	0	1
0	1	0
0	0	0

South-West

0	0	0
0	1	1
0	0	0

West

0	0	0
0	1	0
0	0	1

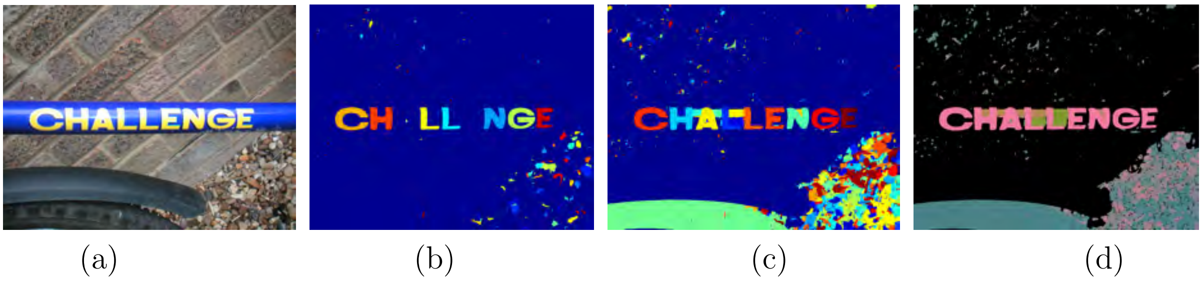
North-West

The edge-following process is attempted up to a distance λ (set to 5 pixels in the current implementation) in the estimated search direction. If an edge pixel is encountered within this distance, the traversed path is made 'ON' and is included in the edge map. On the other hand, if no edge pixel is found at the end of λ pixel traversal, the path is ignored. To close gaps between parallel edge segments, a simple bridging operation is also performed by dilating the pixels at the end points of all the open edges with a structuring element given below:

0	1	0
1	1	1
0	1	0

This post-processing results in closing all the small gaps that may have arisen during the thresholding step of Canny edge detection.

The color values of the region bounded by each of the closed edges are then considered for the subsequent color clustering step using COCOCLUST algorithm described in Section 3.4 of Chapter 3. Each of these blobs is described by its mean $L^*a^*b^*$ color. A one-pass clustering process (See Section 3.4.2) is applied on the mean colors of all the



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

Figure 4.1: Robustness of the proposed edge-guided color segmentation method. (a) A sample input image. (b) Regions segmented by Canny edge detection (c) Segmented regions obtained after edge-linking. (d) Segmented image. Note that the outputs shown in (b)-(d) are obtained after the area-based heuristic filtering.

segmented components. The obtained color clusters initialize a k-means clustering process. The clusters obtained at the end of convergence of the k-means algorithm represent the final segmented output. It may be observed that unlike that of COCOCLUST, where clustering is performed on the color of each pixel, the clustering is carried out at the component-level in this case. Each component, described by its mean color, is checked for similarity with that of other components. The CCs are grouped if the color distance is within a threshold value. Since each component is represented by a single color (mean color), it cannot be further split and the above clustering process only allows merging of 'similar' components that were pre-segmented by the edge detection process. The result of color segmentation is reasonably robust to non-uniform illumination and shadows since edge detection is largely unaffected by these photometric distortions.

By making some sensible assumptions about the document image, obvious non-text components are removed by an area-based filtering. Components whose area is less than 15 pixels or greater than one-tenth of the entire image area are not considered for further processing. Large components whose heights or widths are greater than one-third of the image dimension are also filtered out. This heuristic filtering removes a large number of non-text components thereby keeping the subsequent computations low. Figure 4.1 illustrates the robustness of the proposed method on an example color image where there is a gradual variation in the text color. The retained set of CCs are then passed on to a trained classifier for identifying text components in the image as explained below.

4.4 Feature extraction for text classification

Each individual color layer is analyzed and text-like components are identified employing a set of 12 low-level features that quantify the geometry, boundary, stroke and gradient characteristics of the CC.

Geometric features

Text CCs have geometric properties that can be used to distinguish them from non-text CCs. Their aspect ratios and occupancy ratios tend to cluster around a small range of values. Text CCs are normally much smaller than most of the background clutter. They are also characterized by a small number of convex deficiency regions since text CCs are composed of a few number of strokes. For every component i, we compute the following features:

$$Aspect\,Ratio, R_i = \min\left(\frac{W_i}{H_i}, \frac{H_i}{W_i}\right) \tag{4.1}$$

$$Area\,Ratio, AR_i = \frac{Area(CC_i)}{Area(Image)} \tag{4.2}$$

$$Occupancy, O_i = \frac{|CC_i|}{Area(CC_i)} \tag{4.3}$$

$$Convex\,Deficiency, D_i = \min\left(1, \frac{\#\,CDR_i}{\alpha}\right) \tag{4.4}$$

where W_i and H_i denote the width and height of the i^{th} CC, $|CC_i|$ denotes the number of 'ON' pixels, $Area(Image)$ is the entire area of the scene image being processed and $\#CDR_i$ represents the number of convex deficiency regions of CC_i. The parameter α is used to normalize the feature value so that it lies in the range [0,1].

Boundary features

Text CCs generally have smooth boundaries while non-text components tend to have jagged boundaries. Text components also have well-defined boundaries and hence have a higher degree of overlap with the edge image than the non-text components [62]. These characteristics are captured by the two features, namely boundary smoothness (BSM) and stability (BST).

$$BSM_i = \frac{|CC_i - (CC_i \circ S_2)|}{|CC_i|} \tag{4.5}$$

$$BST_i = \frac{|E_{CC_i} \bigcap \text{Boundary}(CC_i)|}{|\text{Boundary}(CC_i)|} \tag{4.6}$$

Here, E_{CC_i} denotes the set of edge pixels in the image region described by the bounding box of the component CC_i. $\circ$ refers to the morphological opening operation and S_2 a square structuring element of size 2×2.

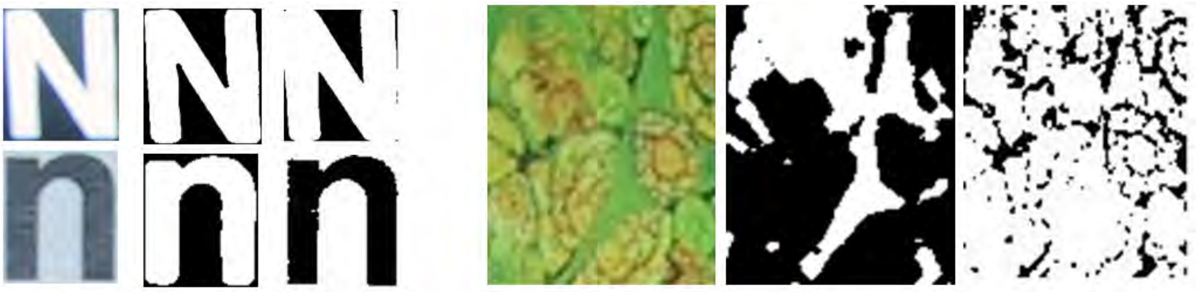
Stroke features

Text components are also characterized by an almost uniform stroke width all along the stroke. The stroke width of a text CC is also normally much smaller than its height. These properties are quantified by the features of stroke width deviation (SWD) and ratio of stroke width of CC to height (RSWH).

$$SWD_i = \frac{StdDev[SW(CC_i)]}{Mean[SW(CC_i)]} \tag{4.7}$$

$$RSWH_i = \frac{SW(CC_i)}{H_i} \tag{4.8}$$

where $SW(CC_i)$ is the estimated stroke width of CC_i. A fast, parallel thinning algorithm proposed by Zhang and Suen [71] is used to compute the stroke width. Assuming that each character is of a uniform color, the original connected component CC and its corresponding binarized version B should have a high degree of 'similarity' and a small degree of 'dissimilarity'. Here B is obtained by binarizing the region corresponding to the CC from the gray counterpart of the original color image. The converse holds for non-text regions since they are generally non-homogeneous in nature. This characteristic of text is named stroke homogeneity (SH) and is computed as follows.



Text patch CC Binary Non-text patch Color CC Binarized image

Figure 4.2: Illustration of the stroke homogeneity feature. It may be observed that for any text CC, the corresponding binary output is 'similar' to the CC obtained from color segmentation. But, non-text CCs do not exhibit such a property owing to the inhomogeneity.

$$B_i = Binarize(ImagePatch)$$

If $|CC_i \cap B_i| \geq |CC_i \cap \text{NOT}(B_i)|$

$$\begin{aligned} Sim_i &= |CC_i \cap B_i| \\ Dissim_i &= |\text{XOR}(CC_i, B_i)| \end{aligned}$$

Else

$$\begin{aligned} Sim_i &= |CC_i \cap \text{NOT}(B_i)| \\ Dissim_i &= |\text{XOR}(CC_i, \text{NOT}(B_i))| \end{aligned}$$

$$SH_i = min\left(1, \frac{Dissim_i}{Sim_i}\right) \tag{4.9}$$

Owing to its simplicity, we choose the block-based Otsu thresholding technique to binarize the gray-scale image patch described by each CC where the image patch is subdivided into 3×3 blocks. Due to the presence of inverse text, the binarized outputs may be inverted in some cases. This is illustrated in figure 4.2. The relation $|CC_i \cap B_i| \geq |CC_i \cap \text{NOT}(B_i)|$ holds for text brighter than the background. Thus, this test is used in obtaining the 'similarity' and 'dissimilarity' measures appropriately.

Gradient features

Text regions are characterized by a high density of edges. As pointed out by Clark and Mirmehdi [58], the gradient exhibits an anti-parallel property in text regions. The magnitude of edges in one direction tends to be matched by edges in the opposite direction of equal magnitude. Based on these observations, three features namely gradient density (GD), gradient symmetry (GS) and angle distribution (AD) are defined.

$$GD_i = \frac{\sum_{(x,y)\in E_{CC_i}} G(x,y)}{\mid CC_i \mid} \tag{4.10}$$

where $G(x,y)$ denotes the gradient magnitude obtained from the Gaussian derivative of the gray scale image.

$$\begin{aligned} GS_i &= \frac{\sum_{i=1}^{8}[A(\theta_i) - A(\theta_{i+8})]^2}{\sum_{i=1}^{8} A(\theta_i)^2} && (4.11) \\ AD_i &= \frac{\sum_{i=1}^{8}[A(\theta_i) - \bar{A}]^2}{\sum_{i=1}^{8} A(\theta_i)^2} && (4.12) \\ & && (4.13) \end{aligned}$$

where $\theta(x,y)$ is the gradient orientation quantized into 16 bins i.e. $\theta_i \in \left[(i-1)\frac{\pi}{8}, \frac{i\pi}{8}\right)$, where $i = 1, 2, \cdots, 16$; $A(\theta_i)$ is the magnitude of edges in direction θ_i and $\bar{A}$ is the mean gradient magnitude over all θ_i.

4.5 Classification of text/nontext CCs

In order to classify the segmented CCs into text and non-text classes, we employ two classifiers namely an SVM and a neural network (NN) classifier trained on the above set of features. The training images from the ICDAR 2003 robust reading competition dataset are segmented using the edge-guided color segmentation approach as described earlier in Section 4.3. We obtain a total of 4551 character CCs and 39599 non-character CCs for feature extraction and subsequent training of the classifiers. Some samples from the training set of text and non-text examples are shown in Figure 4.3.

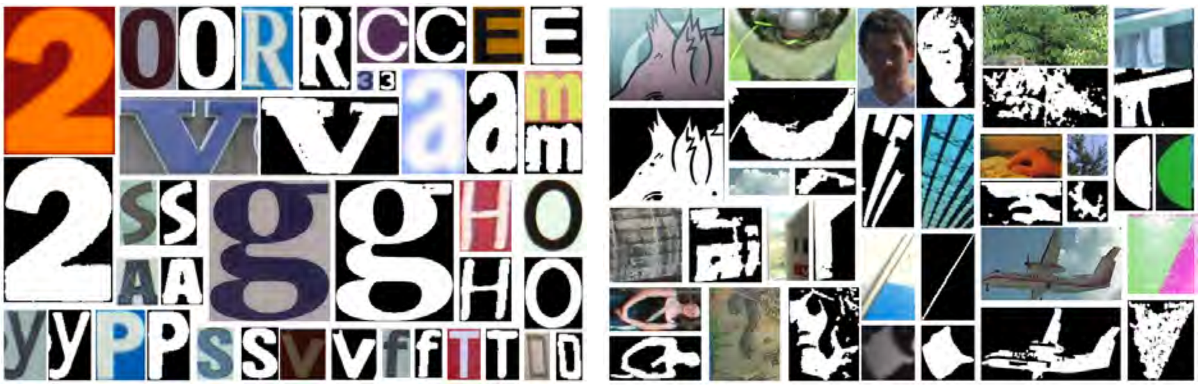


Figure 4.3: Sample text and non-text examples used for feature extraction and training the classifiers.

For implementing the SVM classifier, we use LIBSVM [72] toolkit. Radial basis function is used as the kernel for the SVM and the optimum parameters for the SVM are determined through a 5-fold validation process. The penalty parameters for the SVM were set according to the ratio of the text and non-text feature vectors to prevent biasing of the SVM towards the larger dataset.

In addition, we use the MATLAB Neural Network Toolbox to implement a two-layer feed-forward neural network with one sigmoidal hidden layer and output neurons. The NN is trained using conjugate-gradient back-propagation method and the optimal number of hidden nodes is obtained using the minimum mean-squared error criterion.

During the testing stage, the input image is first segmented into its constituent CCs which are classified as text or non-text using the two trained classifiers. We declare a test CC as text, if it is classified as text by either of the two classifiers.

4.6 Experiments and results

The proposed method is tested on our own database of camera-captured images with complex text layouts and multi-script content. Obviously, it is not appropriate to use rectangular word bounding boxes for quantifying the performance of the method as in the ICDAR 2003 text locating competitions. The ICDAR metric is strict and heavily penalizes errors in estimation of word boundaries that drastically affect the detection

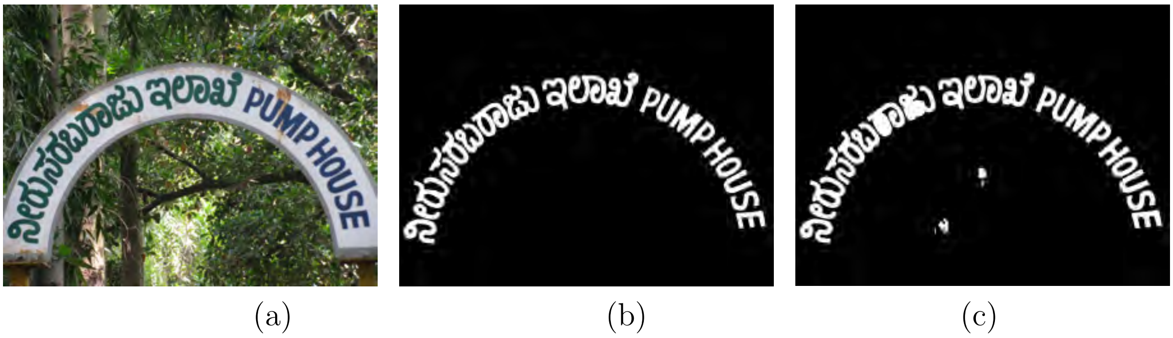


Figure 4.4: (a) A sample multi-script image from our database that contains curved text. Rectangular bounding boxes are inappropriate for quantifying text detection results on such images. (b) the corresponding pixel-accurate ground truth. (c) Text CCs identified by our method. The precision and recall measures are 0.88 and 0.99, respectively.

rate. Pixel-based evaluation of the performance of text detection is preferred since there is no need to consider the number of detected words. Hence, we develop a semi-automatic toolkit [73] for annotating generic scene images and the tool is available for free download. Using the toolkit, we obtain pixel-accurate ground truth of 100 scenic images containing text in various layout styles and multiple scripts. Figure 4.4 shows the ground truth created by our tool for a multi-script image where the text is laid out in an arc form. The availability of such a ground truthed data enables us to evaluate the performance using a simple technique that counts True Positive (TP), False Positive (FP) and False Negative (FN) pixels in order to compute the precision and recall measures. This pixel-based evaluation metric is also used in the Document Image Binarization Contest 2009 held in conjunction with ICDAR 2009.

1. A pixel is classified as True Positive (TP) if it is ON in both the ground truth image and the output of text detection.

2. A pixel is classified as False Positive (FP) if it is ON only in the output of text detection.

3. A pixel is classified as False Negative (FN) if it is ON only in the ground truth image.

The precision (p) and recall (r) measures are then computed as follows:

$$p = \frac{Number\,of\,TP}{(Number\,of\,FP + Number\,of\,TP)} \tag{4.14}$$

$$r = \frac{Number\,of\,TP}{(Number\,of\,FN + Number\,of\,TP)} \tag{4.15}$$

Since the features used for identifying text CCs do not assume any characteristic of the script, the method can detect text irrespective of the script and text orientation. Figure 4.5 shows some sample outputs of text detection on our database. Table 4.1 gives the overall performance of the proposed method, which yields $p = 0.80, r = 0.86$ and $f = 0.83$, respectively. The availability of pixel-accurate ground truth enables us to evaluate the performance of text detection directly without the need to group the text CCs into words.

Table 4.1: Overall result of text localization on our dataset containing multiple scripts and arbitrary text layouts indicating pixel-based evaluation metrics.

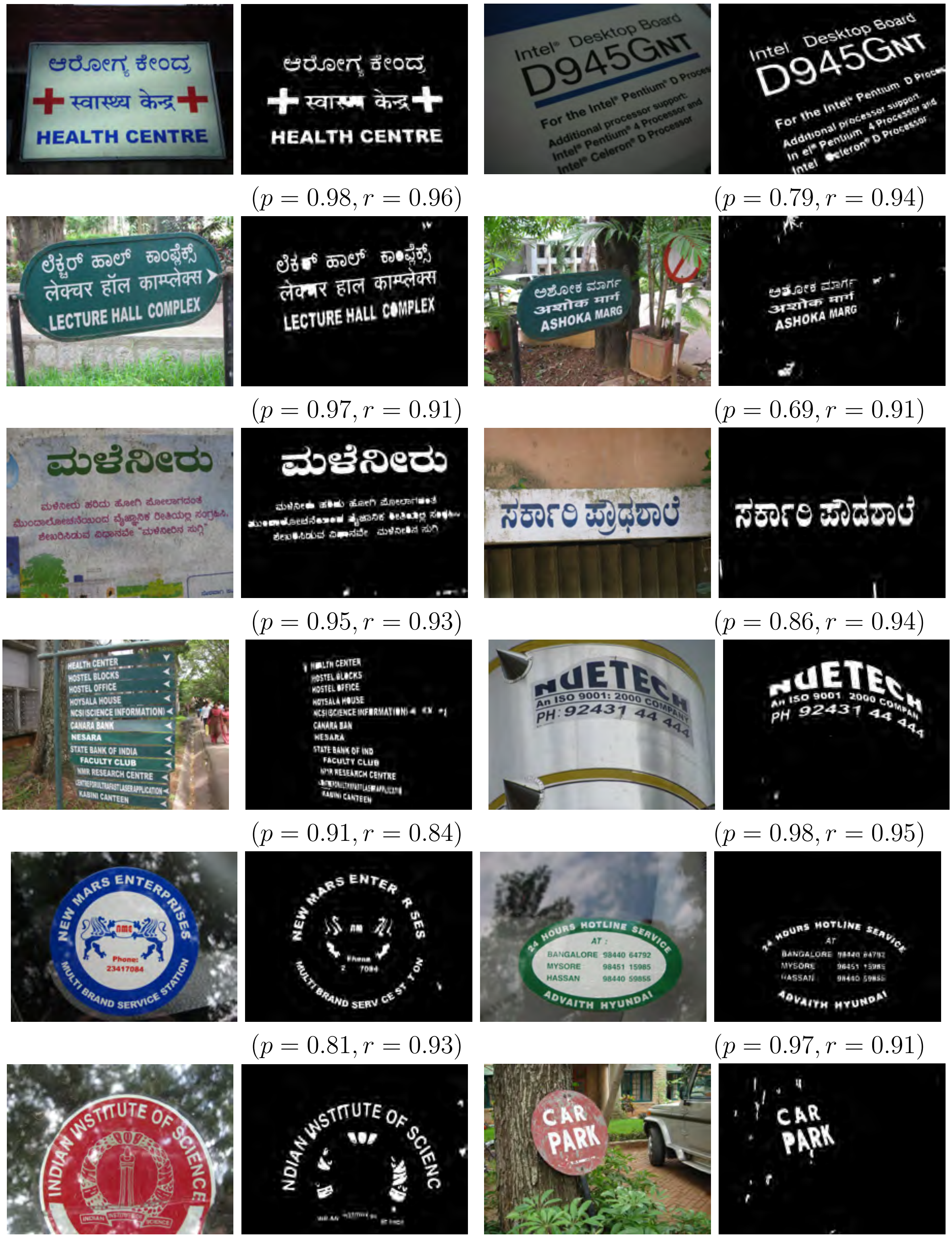
Precision	**Recall**	**f**
0.8	0.86	0.83

4.6.1 Comparison with other methods using the ICDAR dataset

The above results obtained on our database cannot be directly compared with the results of any technique in the literature, since none of them deal with text of multiple, arbitrary orientations in the same image. Thus, in order to make a comparison with the results of other methods in the literature, we have also obtained results on the ICDAR 2003 robust reading competition dataset. We also use the ICDAR 2003 evaluation metrics computed from rectangle-based area matching score to compute the precision and recall. This requires grouping of the detected text CCs into words and obtaining its bounding rectangles for quantifying the result of text detection.

Estimation of word bounding boxes

Since the ICDAR dataset contains only horizontally aligned text, we employ Delaunay triangulation to link adjacent CCs together and obtain those CCs that lie in a straight



$(p = 0.98, r = 0.96)$ $(p = 0.79, r = 0.94)$

$(p = 0.97, r = 0.91)$ $(p = 0.69, r = 0.91)$

$(p = 0.95, r = 0.93)$ $(p = 0.86, r = 0.94)$

$(p = 0.91, r = 0.84)$ $(p = 0.98, r = 0.95)$

$(p = 0.81, r = 0.93)$ $(p = 0.97, r = 0.91)$

$(p = 0.53, r = 0.9)$ $(p = 0.91, r = 0.95)$

Figure 4.5: Representative results of text localization on images with multi-script content and arbitrary orientations. The pixel-based precision $(p)$and recall (r) measures are indicated below each image.

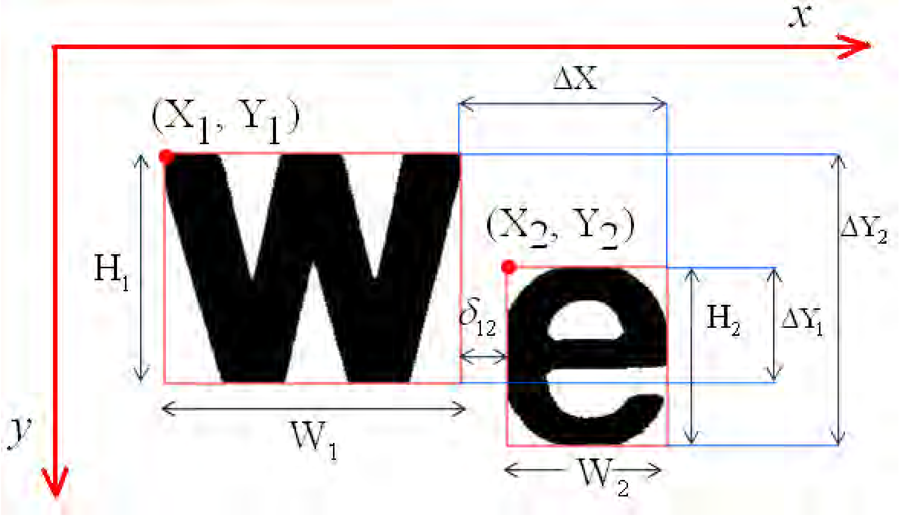


Figure 4.6: The parameters that guide the process of grouping of adjacent CCs into text lines.

line using simple heuristics such as position, height and area. A link map, $V = \bigcup v_{ij}$ is created, where v_{ij} is an edge of a triangle linking CC_i to CC_j obtained by applying Delaunay triangulation to the classified CCs. In the first step, the links are filtered by assigning labels based on height ratio and vertical overlap ratio:

$$\Phi(v_{ij}) = \begin{cases} +1, & [0.5 < \frac{H_i}{H_j} < 2] \wedge [\Delta x > 0] \wedge [(\Delta y_1, \Delta y_2) > 0] \\ -1, & \text{otherwise} \end{cases} \tag{4.16}$$

where (X_i, Y_i, W_i, H_i) are the attributes of the bounding box (See Figure. 4.6). The links with label -1 are filtered out and the remaining links between the characters are used to generate text lines. A text line is assumed to contain a minimum of 3 CCs and hence, text lines containing less than 3 characters are eliminated. This grouping procedure also acts as a verification technique wherein false positives are eliminated using the spatial regularity in appearance of text CCs. To handle isolated characters, we mark all components with high posteriors for text during the classification step and accept only those CCs whose likelihood exceeds 0.9.

Since the database contains only horizontal text, we make use of the spatial regularity in the occurrence of characters in a text line to recover false negatives. A straight line $F(x) = ax+b$ is fitted to the coordinates of the centroids $\{C_{xi}, C_{yi}\}$ of the bounding boxes

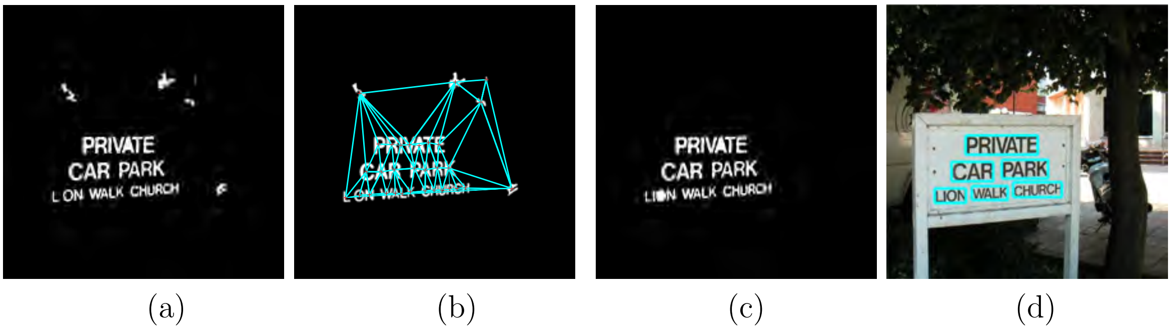


Figure 4.7: Grouping of verified CCs into words and recovery of false negatives using the spatial coherence property of text. Notice that the character 'I' in the last line, which was initially misclassified as non-text, is re-classified as text during the grouping stage since its size is comparable to that of the adjacent text CCs.

of all CCs in a text group. A component CC'_k, whose bounding box center (C'_{xk}, C'_{yk}) lies in the area covered by the text string, is re-classified as text component if the following criteria are satisfied:

$$\Lambda' < mean(\Lambda) + \beta H_{line} \tag{4.17}$$

$$\frac{\bar{A}_{CC}}{4} < Area(CC_k) < 2 \times \bar{A}_{CC} \tag{4.18}$$

where $\Lambda = abs(F(X_i) - Y_i)$, $\Lambda' = abs(F(X'_k) - Y'_k)$, β is a control parameter empirically set to 0.2, H_{line} is the height of bounding rectangle of text line and $\bar{A}_{CC}$ is the mean area of all CCs in the string. As illustrated in Fig. 4.7, it also enables the recovery of CCs that were misclassified by the classifiers, thereby increasing the performance of the method. The inter bounding box distance is used to cut each text line into words. The distance of every component CC_j to its adjacent component CC_i is calculated as follows:

$$\delta_{ji} = abs(X^i - (X^j + W^j)) \tag{4.19}$$

The text line is cut wherever the following condition occurs:

$$\delta_{ji} > \gamma \times mean(\{\delta_{ji}\}) \tag{4.20}$$

where γ is a control parameter empirically set to 2.65. Each segment of the text line is

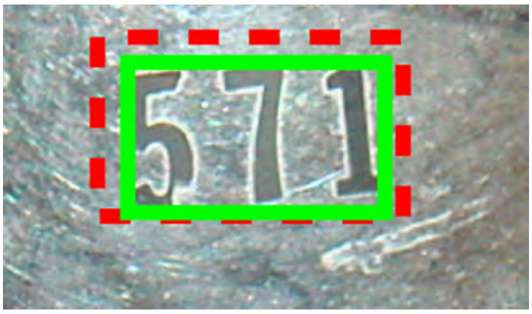


Figure 4.8: Ground truth (red dotted) rectangle for a sample image and the output (green solid rectangle) of our text detection method. Notice that the detected rectangle does not agree exactly with the ground truth rectangle.

considered as a word and its bounding rectangle is computed.

Performance evaluation

Figure 4.8 shows the ground truth data a text string provided in terms of the coordinates of the bounding rectangle and the text region detected by our method. For text localization, it is unrealistic to expect a system to agree exactly with the ground truth identified by a human tagger. So, the ICDAR 2003 robust reading competition organizers propose the following area-based matching to compute the precision and recall measures. We adopt the same performance evaluation system to compare our results on ICDAR dataset with other methods.

The best match $m(R, S)$ for a rectangle R in a set of rectangles S is defined as:

$$m(R, S) = max(m_p(R, R') \mid R' \in S) \tag{4.21}$$

$$\text{where}\, m_p(R, R') = \frac{2\, Area(R \bigcap R')}{Area(R) + Area(R')}$$

and $Area(.)$ is the area of the bounding rectangle. This measure of rectangle match is computed for all the rectangles in the set S and the one that yields the highest overlap measure is selected. For each rectangle in the set of estimates, we find the closest match in the set of targets and vice versa. Depending on whether the precision or recall is being computed, S represents the set of all the rectangles present in either the ground truth or the detected rectangles.

Precision (p) is then defined as the ratio of correct estimates to the total number of estimates.

$$p = \frac{\sum_{r_e \in D} m(r_e, T)}{|D|} \tag{4.22}$$

Recall (r) is defined as ratio of correct estimates to the total number of targets.

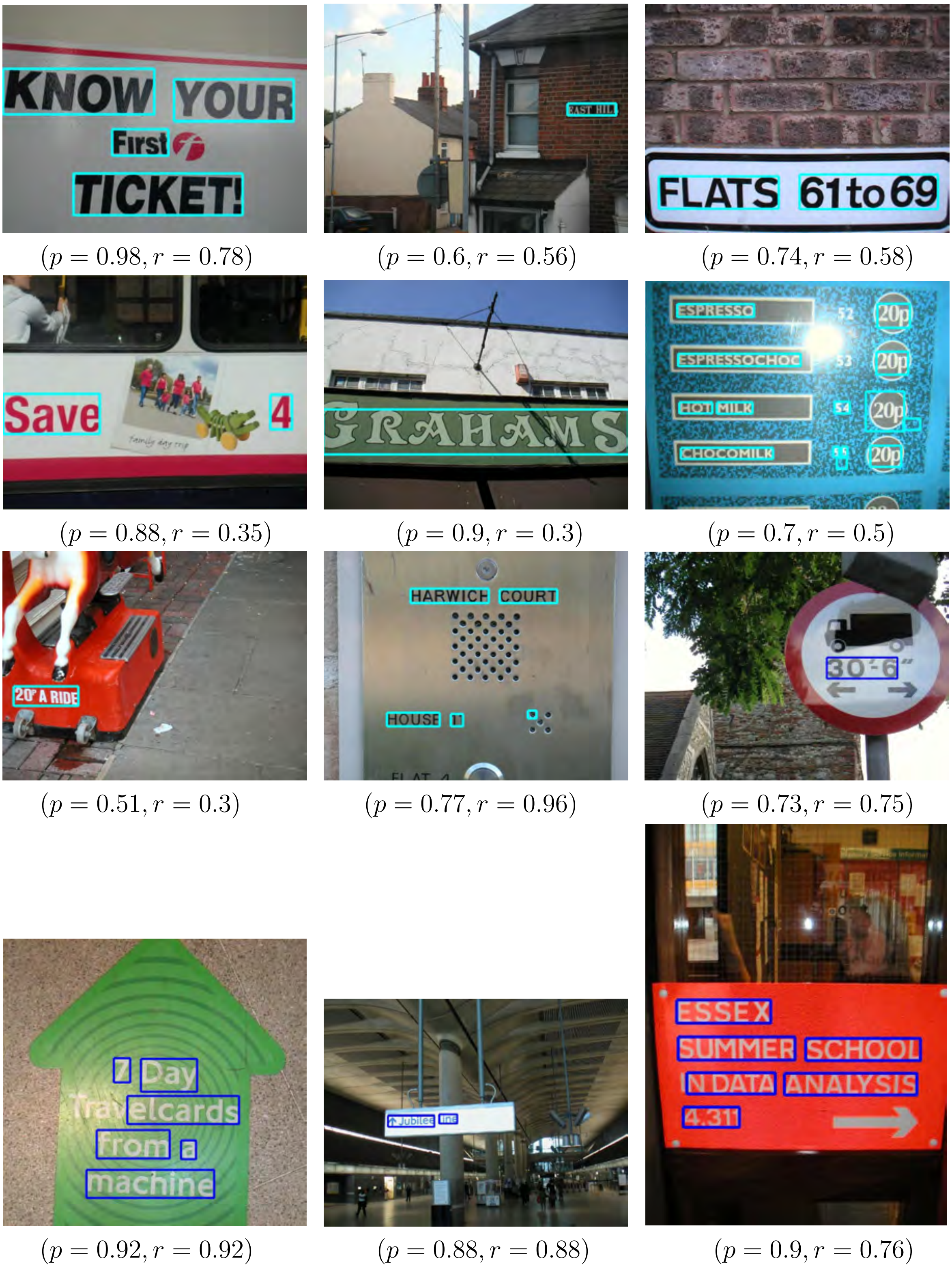
$$r = \frac{\sum_{r_t \in T} m(r_t, D)}{|T|} \tag{4.23}$$

where $|T|$ and $|D|$ are the total number of ground truth and detected rectangles. The standard f measure is adopted to combine the precision and recall values into a single measure of quality. The relative weights of these are controlled by η, which is set to 0.5 to give equal weight to precision and recall

$$f = \frac{1}{\eta/p + (1-\eta)/r} \tag{4.24}$$

Using the above rectangle-based evaluation metrics, we test the performance of our method on the ICDAR 2003 Robust Reading Competition test set containing 251 images. A few sample outputs of our text detection method are shown in Figures 4.9 and 4.10. The corresponding values of the precision and recall are also given below each image. It may be noted that all the text regions in Fig. 4.10 are correctly identified and should ideally yield 100% precision and recall measures. However, they range from 0.88 to 0.98 due to the differences between the detected and the ground truth rectangles.

Table 4.2 lists the performance of our method and compares it with other methods. The overall precision, recall and f-measures obtained on the whole test set are 0.63, 0.59 and 0.61 respectively. The dataset contains several images that contain a single character which gets filtered out during the grouping of CCs to estimate the bounding rectangles of words. There are also some images where the interpretation as text is controversial. Our method fails to detect any text in such images as shown in Figure 4.11.



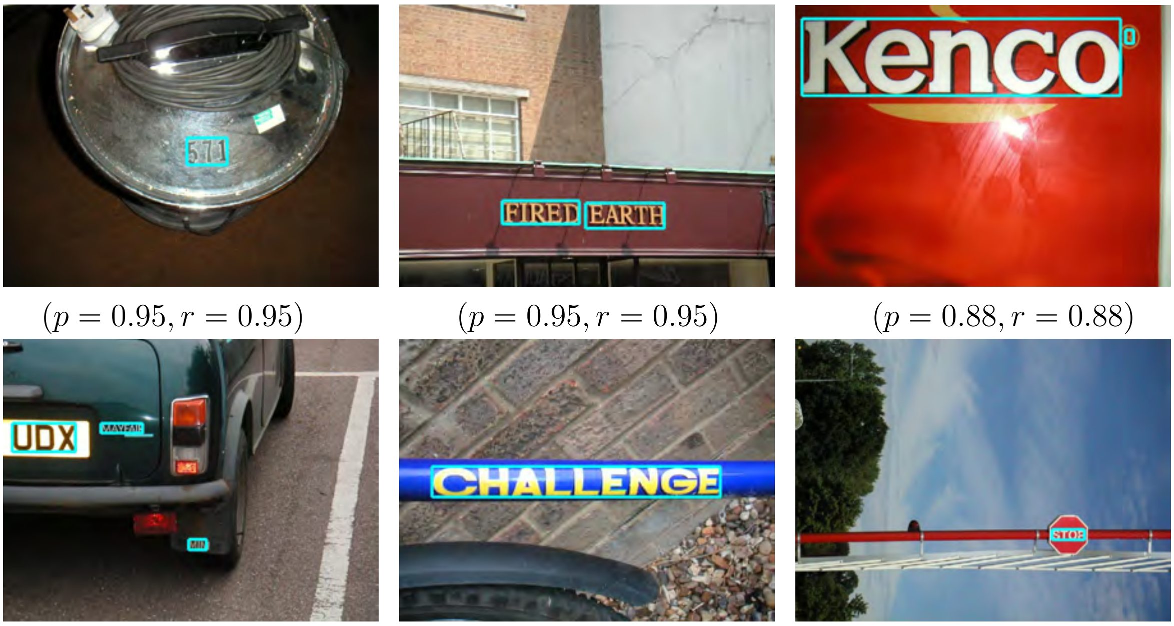
($p = 0.98, r = 0.78$) ($p = 0.6, r = 0.56$) ($p = 0.74, r = 0.58$)

($p = 0.88, r = 0.35$) ($p = 0.9, r = 0.3$) ($p = 0.7, r = 0.5$)

($p = 0.51, r = 0.3$) ($p = 0.77, r = 0.96$) ($p = 0.73, r = 0.75$)

($p = 0.92, r = 0.92$) ($p = 0.88, r = 0.88$) ($p = 0.9, r = 0.76$)

Figure 4.9: Representative results of text localization obtained using our method on the ICDAR 2003 dataset.



$(p = 0.95, r = 0.95)$ $(p = 0.95, r = 0.95)$ $(p = 0.88, r = 0.88)$

$(p = 0.89, r = 0.89)$ $(p = 0.98, r = 0.98)$ $(p = 0.94, r = 0.94)$

Figure 4.10: Example results where the proposed method yields perfect text localization. Notice that the precision and recall measures lie between 0.88 to 0.98.

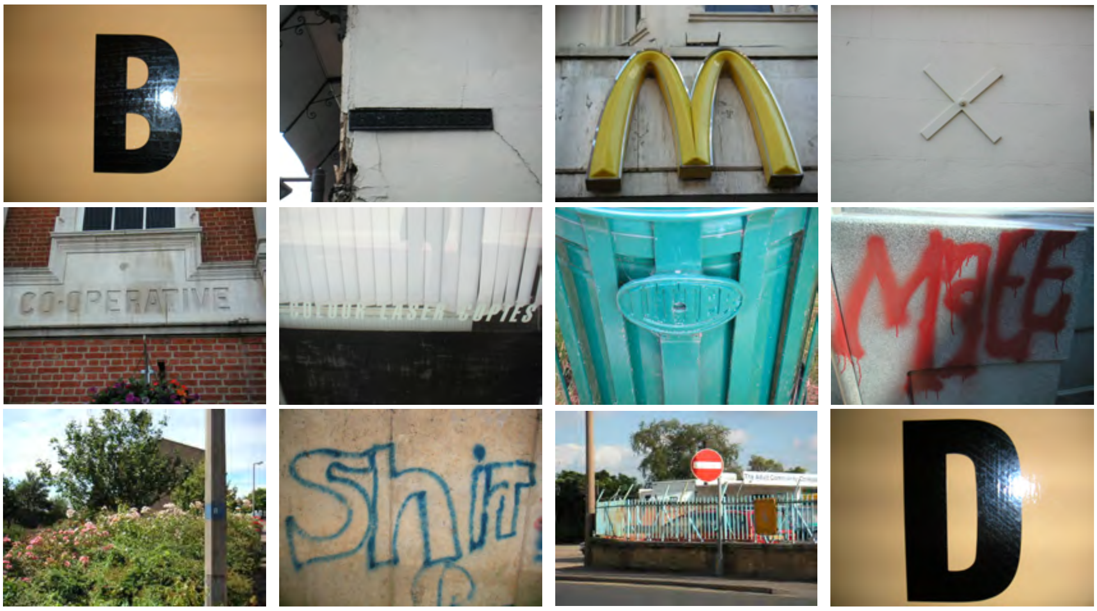


Figure 4.11: Images from the ICDAR 2003 dataset for which the proposed method fails to localize any text.

Table 4.2: Comparison of the proposed method with the performance of other techniques on the ICDAR2003 robust reading competition dataset.

	Method	**Precision**	**Recall**	**f**
ICDAR 2003	Ashida	0.55	0.46	0.5
	HW David	0.44	0.46	0.45
	Wolf	0.3	0.44	0.35
	Todoran	0.19	0.18	0.18
ICDAR 2005	Hinnerk Becker	0.62	0.67	0.62
	Chen and Yuille	0.6	0.6	0.58
	Qiang Zhu	0.33	0.4	0.33
	Jisoo Kim	0.22	0.28	0.22
	Nobou Ezaki	0.16	0.36	0.22
Recent methods	Hanif and Prevost [74]	0.25	0.35	0.29
	Minetto et al. [75]	0.63	0.61	0.61
	Nuemann and Matas [76]	0.59	0.55	0.57
	Epshtein et al. [1]	0.73	0.6	0.66
	Proposed method	**0.63**	**0.59**	**0.61**

The performance of our method is comparable to that of the best-performing methods in the ICDAR 2005 text locating competition. It performs worse than a recent method proposed by Epshtein et al. [1] which is designed for locating only horizontal text. Our method, however, works well for generic scene images having arbitrary text orientations.

4.7 Conclusions

This chapter describes an important preprocessing for scene text analysis and recognition. CC-based approaches for text detection are known for their robustness to large variations in font styles, sizes, color, layout and multi-script content. Such approaches are therefore more suitable for processing camera-based images. However, extraction of CCs from complex background is not a trivial task. To this end, we have introduced a robust color segmentation technique suitable for extracting text CCs from complex backgrounds. The edge information is used to obtain a set of colors which in turn is used to initiate an unsupervised clustering algorithm that yields an accurate and robust identification of CCs. We also design a set of 'intrinsic' features of text for classifying each of the identified CC as text or non-text. These features enable robust text detection independent of the text layout and the script. The system's performance is enhanced by invoking the spatial regularity property of text that effectively filters out inconsistent CCs while at the same time aids in recovering some of the missed text CCs. Experiments on camera-captured images that contain variable font, size, color, irregular layout, non-uniform illumination and multiple scripts illustrate the robustness of the method.

Chapter 5

Alignment of curved character strings

"A smile is a curve that sets everything straight." - Phyllis Diller

Abstract:*Conventional optical character recognition systems, designed to recognize linearly aligned text, perform poorly on document images that contain multi-oriented text lines. This chapter describes a novel technique that can detect text lines of arbitrary curvature and align them horizontally. By invoking the spatial regularity properties of text, adjacent components are grouped together to obtain the text lines present in the image. To align each identified text line, we fit a B-spline curve to the centroids of the constituent characters and normal vectors are computed all along the resulting curve. Each character is then individually rotated such that the corresponding normal vector is aligned with the vertical axis. The method has been tested on a data set consisting of 50 images with text laid out in various forms namely arc, wave, triangular and a combination of these with linearly skewed text lines. It yields 95.9% recognition accuracy on text strings where state-of-the-art OCRs fail, before alignment.*

5.1 Introduction

Digital cameras are increasingly used for acquiring document images. In addition to imaging hard copy documents, they are now used for acquiring text present in 3D real world objects such as signboards, buildings, vehicles and T-shirts. Applications are being developed, where hand-held imaging devices can be used to read text from such sources. Texts from such sources are often oriented arbitrarily for artistic and other purposes. Figure 5.1 shows a few examples of such images. State-of-the-art OCR packages are not designed to handle text laid out in a curvilinear fashion where the skew of the characters in the text string changes gradually and no two characters may have the same skew. Though there has been a lot of research on document image analysis, there is very little work on retrieving text information from new document types that contain text lines in arbitrary orientations.

Conventional OCR systems perform well only in recognizing texts that are linearly aligned as is evident from Figure 5.2. While the horizontally aligned text is easily detected and recognized, curved text not only poses a challenge to recognition but also makes the text segmentation process difficult. In most existing OCR systems, a skew correction process is often performed prior to recognition, should a need arise. Most skew estimation techniques assume the presence of long and straight text lines. This assumption may not always be true for scene images, which often contain short text lines that could be oriented in an arbitrary direction. It may also contain text laid out in a curvilinear fashion where every character in the text string is skewed differently. These issues call for specialized pre-processing techniques that estimate and correct the skew of individual characters before feeding such a document to an OCR system.

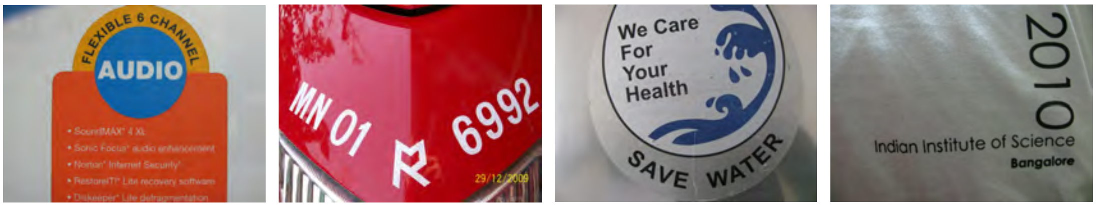
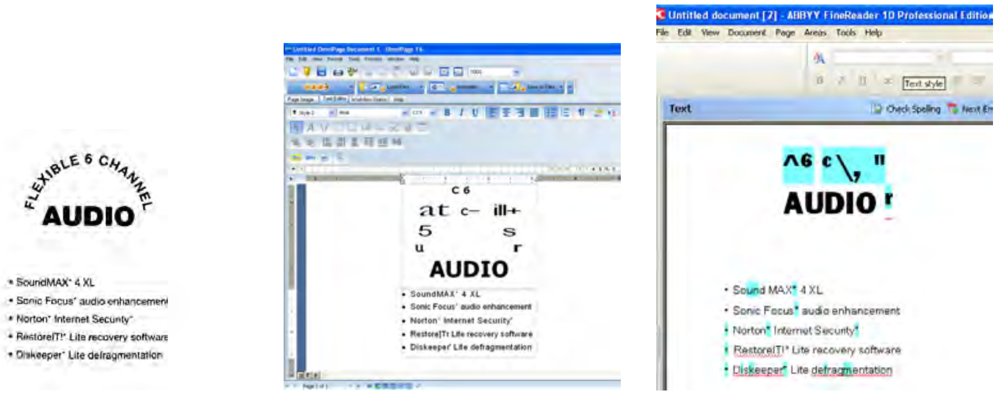


Figure 5.1: Sample camera-captured images that contain curved text strings.


FLEXIBLE 6 CHANNEL
AUDIO
• SoundMAX® 4 XL
• Sonic Focus® audio enhancement
• Norton® Internet Security®
• RestoreIT!® Lite recovery software
• Diskeeper® Lite defragmentation
c 6
at c– ill+
5 s
u r
AUDIO
Untitled document [2] - ABBYY FineReader 10 Professional Edition
File Edit View Document Page Areas Tools Help
Text style
Text
Check Spelling
^6 c\, "
AUDIO r

(a) (b) (c)

Figure 5.2: Performance of state-of-the-art OCR systems on curved text strings. (a) Binary input image containing a curved text string (b) Output of Nuance Omnipage Professional 16 OCR (Trial version) (c) Output of ABBY FineReader 10 Professional OCR (Trial version). While the horizontally aligned text is easily detected and recognized, curved text line poses difficulty in segmentation as well as recognition.

5.2 Review of related work

Documents that have long parallel text lines require a simple global skew detection and correction. There are a host of well-performing techniques that use projection profile [77, 78], Hough transform [79, 80, 81], principal component analysis [82, 83], nearest neighbor clustering [84, 85, 44] or a combination of them [86] to de-skew such document images.

Pal *et al.* [87] reported a maiden attempt to handle multiple skew in Bangla and Devanagari script documents using inherent characteristics of the script. Both the scripts have a headline that connects characters into a word. The method searches for the upper envelopes of each CC to obtain the headlines. A clustering procedure is performed on the detected headlines and each of the resulting clusters gives an estimate of the skew of an individual text line. Though the method can handle multiple skew, it requires that the document contain long straight text lines and is heavily dependent on the characteristics of the script. It is not applicable to the majority of scripts since they do not have a headline. The method is also restricted to documents with skew angle less than 45°. These assumptions may not always be met for camera-based images that are characterized by acquisition with fewer constraints.

In [88], the authors proposed a new technique for skew correction in documents that may contain several areas of text with different skew angles. Adjacent connected components are grouped using a nearest neighbor approach to form words which are further grouped into text lines. Then, the top-line and baseline for each text line are estimated using linear regression. Finally, the local skew angle of each text area is estimated and corrected independently to horizontal or vertical orientation. However, the method is limited to de-skewing straight text lines only.

Zhang and Tan [89] use connected component analysis and regression technique to restore curved text lines that arise around the spine of scanned pages of a book. The method first identifies the shaded region and binarizes the image using a modified Niblack's method to remove the shade. Then, clustering is performed on the connected components in the clean area to obtain text lines, which are modeled by straight reference lines using linear regression. A bottom-up approach is applied to cluster the connected components in the shaded area into warped text lines, and polynomial regression is used to model the warped text lines with quadratic reference curves. The method assumes that the document with moderate skew is scanned in such a way that the text lines are horizontal. It also requires that there are partially straight text lines in the image.

Uchida *et al.* [90] proposed an interesting approach that could identify the skew angles of individual characters in a document image using an instance-based matching method. Unlike other skew correction techniques, the method does not rely on the local straightness of the text lines and/or character strokes. It can handle documents where the characters do not form long straight lines. The method computes rotation variants and invariants for each character and stores them as instances. Skew is estimated at the connected component level. It identifies the character category, estimates local skew from the stored instances of the identified category and finally determines global skew employing a voting strategy. The font image and the rotation angle for which the match score is maximum yield the character category and the local skew angle. However, the method still assumes a single skew for the whole page and also requires that the documents have fonts similar to the ones that are stored as instances.

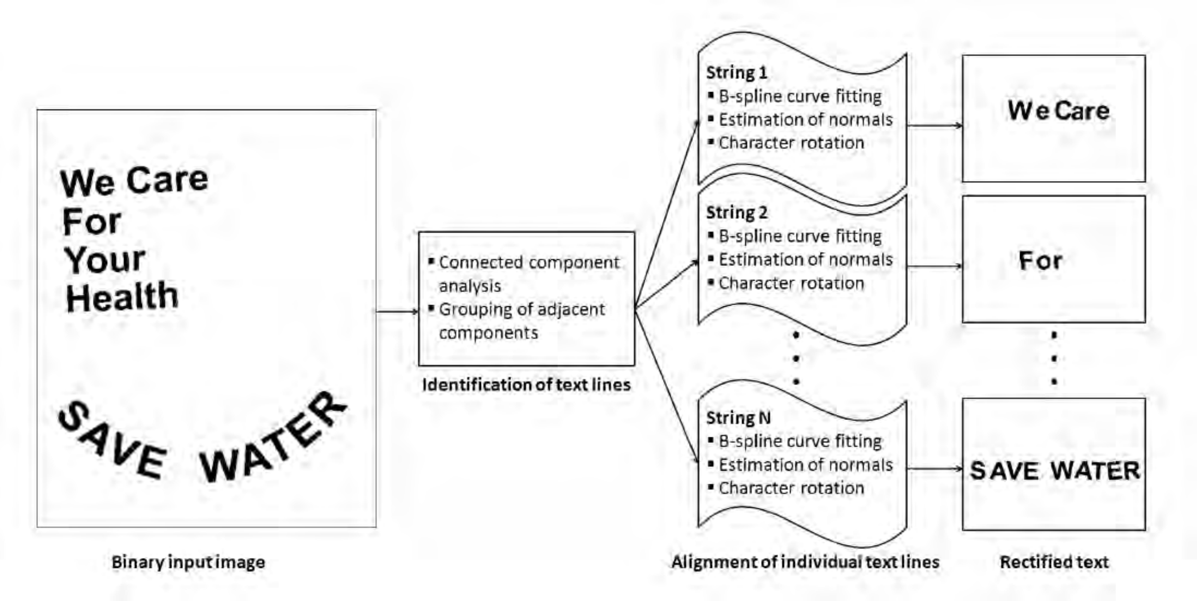


Figure 5.3: Block diagram of the proposed text alignment method where text strings are first identified and then individually processed to align them horizontally.

Two methods [91, 92] address curved text but yield only partial success. In [91], partial images of character strings are cut out from a color document and assuming that the sequence of the characters constituting the text line is known, quadratic functions are used to approximate the curvature of the string. The type of curvature of the string is determined using heuristics and a suitable correction scheme is applied. The method assumes only one text string per image and cannot handle documents containing multiple text lines. Only linearly skewed and arc-form text are considered and it cannot handle other text layouts such as ‘triangular’ or ‘wave’ or a combination of different forms.

In [92], a method to transform arc-form text to linear-form-text using concentric ellipsoidal contours is proposed. The method assumes that the text in the document is either circular or elliptical in shape and is limited to only the upper half circle or ellipse. The use of ellipsoidal contours also leads to severe distortion of the characters during the arc-to-linear transformation. This calls for a further de-tilting step. This post-processing step works only for upper case Latin script. Documents with multiple script, multiple text lines and multiple text orientations cannot be handled. It also focusses only on the alignment of individual characters and does not account for the inter-word gap that is essential to obtain meaningful text.

We propose a novel method to extract and align multiple text strings in documents each having arbitrary curvature. The text line extraction module relies on the spatial regularity properties of text. A B-spline curve is fitted to each identified text string and its constituent characters are then individually aligned by estimating vectors normal to the fitted curve. Figure 5.3 shows the block diagram of the proposed method. This approach also addresses the inter-word gaps in curved text strings.

5.3 Proposed methodology

5.3.1 Connected component labeling

We assume that the input image contains text extracted from camera-captured images. We use an adaptive binarization technique proposed in [27] which performs well for a wide range of documents with non-textured color backgrounds. The input to the text alignment block is the resulting clean binary output where all the text components are represented in black irrespective of their original colors in the input image. An 8-connected component labeling is performed on this binary image to obtain a set of 'M' disjoint components $\{CC^j\}, j = 1, 2, \cdots, M$. Small and spurious components from this binary image are removed by an area-based heuristic filtering.

5.3.2 Identification of text strings

Since text normally exhibits spatial regularity, we seek to obtain text strings by grouping adjacent CCs based on their spatial proximity and regularity of size and orientation. The heights of characters in a text line are generally more 'stable' than their widths and do not vary even if they are represented in italics or bold. To make use of this observation during the search for and grouping of similar adjacent CCs, the local orientation of the text is first determined from the angle between the centroids of a CC and its nearest neighbor (NN). If the angle lies in the intervals $[45°, 135°]$ or $[225°, 315°]$, the orientation of the text line is assumed to be vertical; otherwise, it is horizontal. The estimated angle

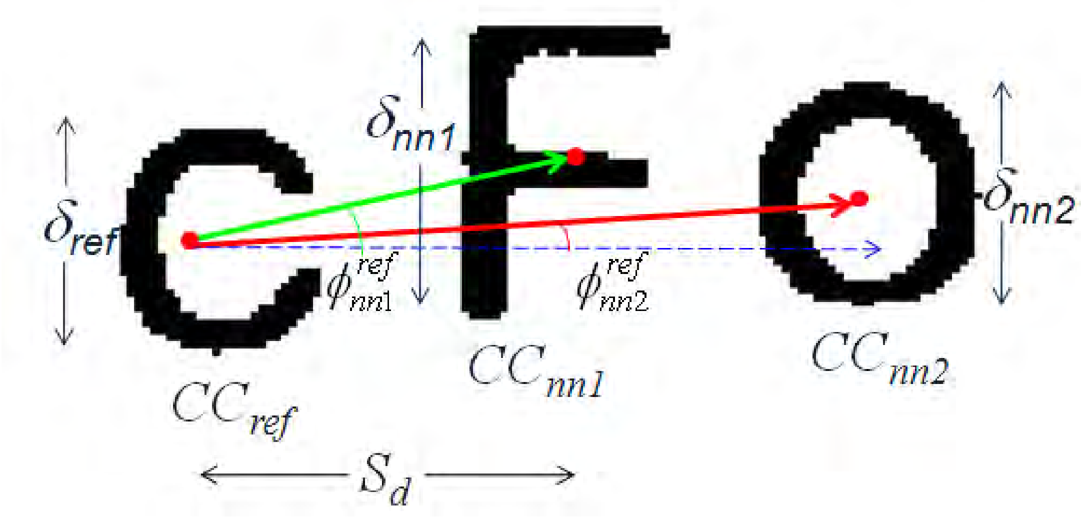


Figure 5.4: The parameters that guide the successive nearest neighbor search for the identification of text lines.

and orientation guide the subsequent NN search and grouping. Depending on whether the estimated text orientation is horizontal or vertical, the NN-search distance is made proportional to the height or width, respectively of the reference CC denoted by CC_{ref}.

For a given CC_{ref} and its nearest neighbor CC_{nn1}, we find CC_{nn2}, which is the subsequent NN of CC_{nn1}. The resulting triplet is analyzed for similarity of sizes and angles and is grouped together whenever the following conditions are satisfied:

$$|\phi^{ref}_{nn1} - \phi^{ref}_{nn2}| < T_1 \tag{5.1}$$

$$T_2 \leq \frac{\delta_{ref}}{\delta_{nn1}}, \frac{\delta_{ref}}{\delta_{nn2}} \leq T_3 \tag{5.2}$$

$$S_d < T_4\, \delta_{ref} \tag{5.3}$$

where $S_d = \text{Dist}(CC_{ref}, CC_{nn1})$ refers to Euclidean distance between the centroids of CC_{ref} and CC_{nn1}, ϕ^{ref}_{nni} denotes the anti-clockwise angle of the line joining the centroids of the reference component and its i^{th} neighbor with respect to the horizontal. The parameters δ_{ref}, δ_{nn1} and δ_{nn2} represent the heights or widths of CC_{ref}, CC_{nn1} and CC_{nn2}, respectively depending on the search direction. These parameters are illustrated in Figure 5.4. Equation 5.1 checks that the difference between the angles of the two neighbors with respect to the reference component lies within a threshold value. Equation 5.2 tests the similarity in their sizes while Equation 5.3 restricts the search distance for locating the next neighbor. Since components of a text string exhibits some degree of spatial, size and orientation regularity in general, the parameters T_1, T_2, T_3 and T_4 work well for values in the range [20 30], [0.25 0.5], [1.5 2.0] and [2 3] respectively. In this work, these heuristic parameters are set to 25, 0.35, 1.75 and 2.5, respectively. The search process is repeated again by making CC_{nn1} as the reference component CC_{ref}. This process is continued till all the CCs are considered and we obtain the individual text lines $\{T^k\}, k = 1, 2, \cdots, K$ where K is the total number of detected text lines. Figure 5.5 shows the identified text strings for images that have multiple text lines, oriented arbitrarily.

5.3.3 B-spline curve fitting

B-splines are piecewise polynomial functions that can provide local approximations of contours of arbitrary shapes using a small number of parameters [93]. We employ B-splines in our work because they exploit the smoothness inherently present in the text layout. Since polynomial fitting results in numerical problems for vertically aligned text, the usual practice is to swap the x and y coordinates before curve fitting [91]. The issue still remains for an image that contains horizontal as well as vertical text lines. B-splines can be used to represent curves of any shape thereby making it an ideal choice for handling images that contain multiple text lines with different curvatures.

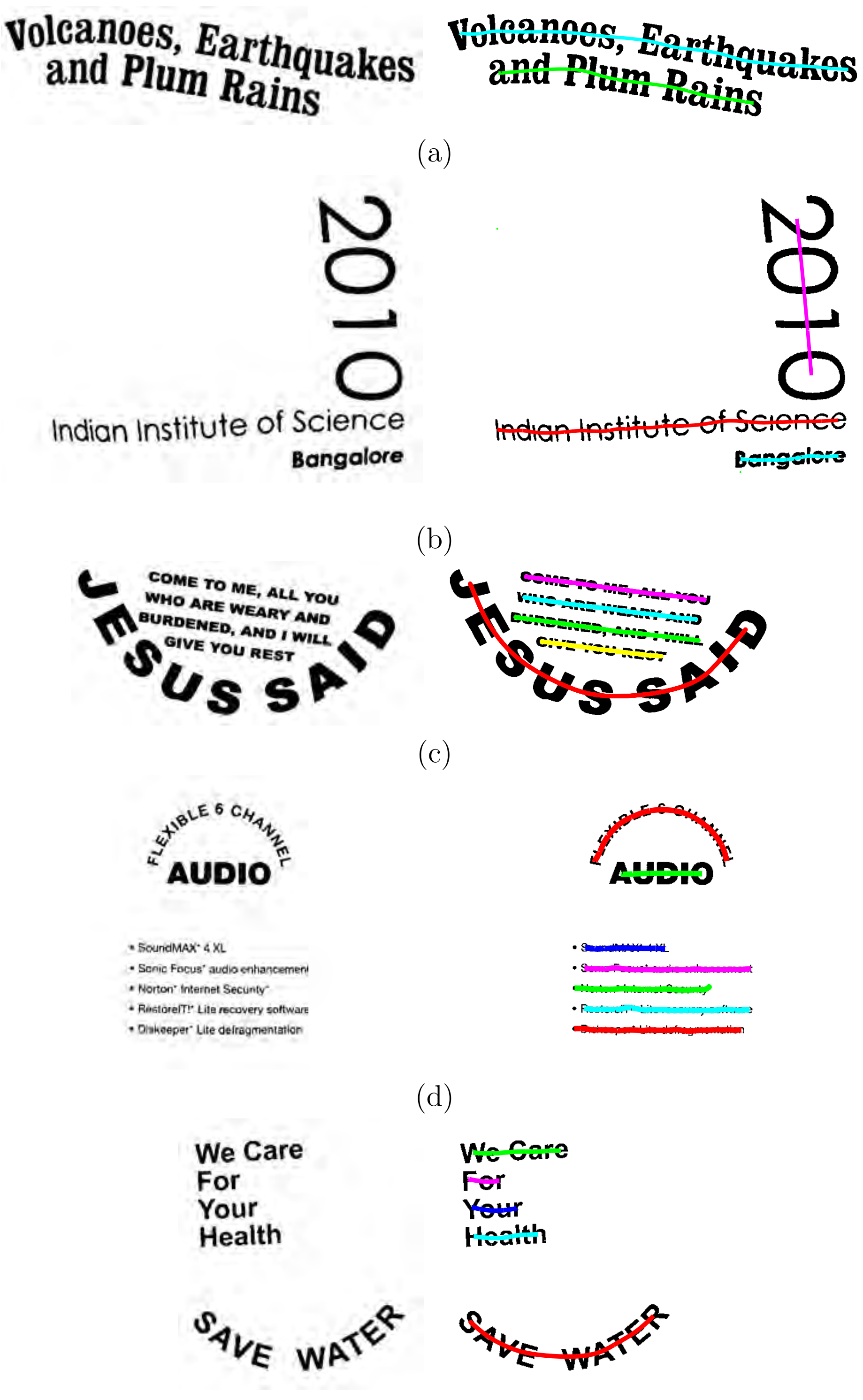
For each identified text string T^k, the centroids of the constituent characters are identified and represented as follows:

$$\mathbf{C}_i^k = \{(C_{x,i}^k, C_{y,i}^k)\}, \quad i = 1,\, 2,\, \cdots,\, N^k \tag{5.4}$$

where 'N^k' is the number of components in the k^{th} text string. These points serve as the control points for fitting B-spline curves. The i^{th} point in the resulting curve is represented as follows:

$$\mathbf{X}_i^k = \{x_i^k, y_i^k\} \tag{5.5}$$

The order of the spline curve is chosen to be 5 so that the resulting curve is smooth



(a)

(b)

(c)

(d)

(e)

Figure 5.5: Results of text line extraction on binary images containing multi-oriented characters and multiple text lines.

and offers robustness to the zig-zag pattern of the control points. The zig-zag pattern arises due to the presence of ascenders and descenders. The normal vector at the i^{th} point of the curve is then computed using Equation 3.12 of chapter 3.

$$\mathbf{n}_i^k = \begin{bmatrix} cos(\frac{\pi}{2}) & -sin(\frac{\pi}{2}) \\ sin(\frac{\pi}{2}) & cos(\frac{\pi}{2}) \end{bmatrix} \times \frac{1}{2}\left(\frac{\mathbf{X}_i^k - \mathbf{X}_{i-1}^k}{\|\mathbf{X}_i^k - \mathbf{X}_{i-1}^k\|_2} + \frac{(\mathbf{X}_{i+1}^k - \mathbf{X}_i^k)}{\|\mathbf{X}_{i+1}^k - \mathbf{X}_i^k\|_2}\right) \tag{5.6}$$

Here, $\mathbf{X}_i^k$ denotes the position on the spline curve at which the normal vector is computed.

5.3.4 Horizontal alignment of text

The normal vectors, thus locally determined, give a fairly good estimate of the orientation of the individual characters. Thus, we can obtain the rectified text string by simply rotating each character such that the normal vector is aligned vertically. The required angle of rotation for any character is computed as:

$$\theta_i^k = (90^\circ - \angle \mathbf{n}_i^k) \quad i = 1, 2, \cdots, N^k \tag{5.7}$$

The values of θ^k are indicative of the type of orientation of the text string T^k. Here, a positive/negative value of θ implies rotation in an anti-clockwise/clockwise direction. For a linearly skewed text string, the standard deviation of the θ^k is 'small'. In such a case, the whole text string is rotated by a global angle which is computed as the median value of θ^k. On the other hand, if θ^k varies progressively from positive to negative values or vice versa once or multiple times, the text string is curved. In this case, each character is individually rotated.

Though the method can deal with arbitrary text line orientations by estimating and correcting the skew of each character individually, the way the CCs are grouped into text lines ignores small components such as punctuation marks and the '·' associated with the letters 'i' and 'j'. In order to group punctuation marks and compound characters that comprise multiple CCs, we design a structuring element as shown in Figure 5.6 whose size is proportional to the height H of the character under consideration and the black

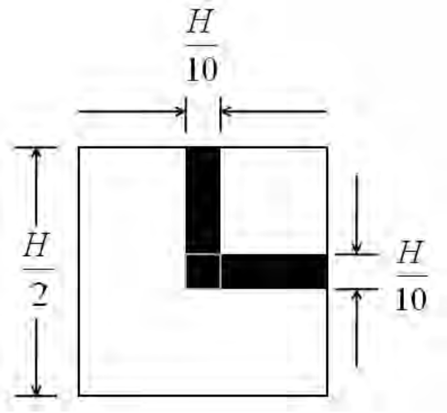


Figure 5.6: Structuring element used to group punctuation marks and compound characters. The size of the structuring element is proportional to the height of the character being processed.

regions indicate 'ON' pixels.

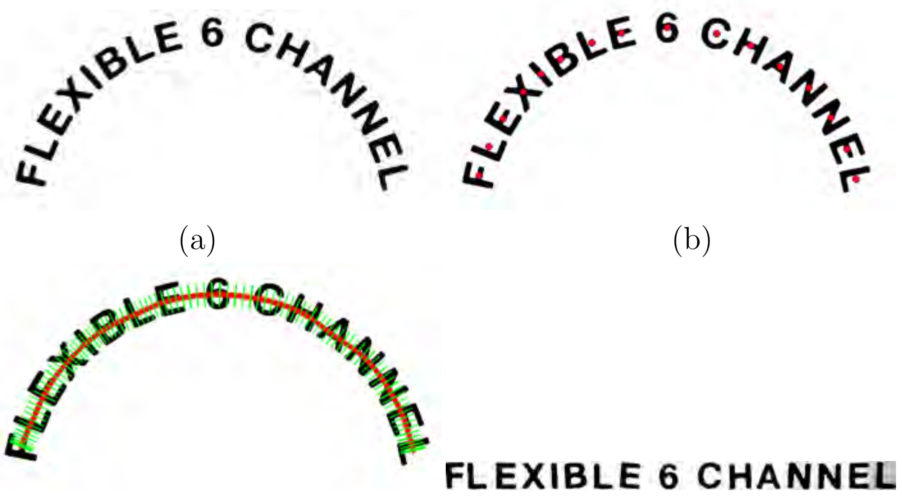
Each character in a text string is first dilated using the structuring element oriented along the estimated normal. Using the dilated region as a region of interest (ROI) mask, we identify the labels of all CCs within the mask and check for the existence of new CCs that do not belong to any of the identified text strings. If any such CCs are encountered, they are combined with the character under consideration. The size of the structuring element is adaptively set to half the height of the character being processed.

The output image is generated by stacking the rotated characters in a left to right direction such that the spacing between the characters is proportional to the corresponding inter-character centroid distances. The inter-character spacing ($S^k_{i,i+1}$) between the component CC_i and CC_{i+1} is computed as follows:

$$S^k_{i,i+1} = D^k_{i,i+1} - \left(\frac{W^k_i + W^k_{i+1}}{2}\right) \tag{5.8}$$

where $D^k_{i,i+1} = \sqrt{(C^k_{x,i} - C^k_{x,i+1})^2 + (C^k_{y,i} - C^k_{y,i+1})^2}$ is the Euclidean distance between the centroids of the i^{th} and $(i+1)^{th}$ components and W^k_i represents the width of the component CC_i after rotation.

Due to the presence of ascenders and descenders, the rotated characters need to be positioned with proper offset values in the vertical axis. The spline curve represents



(a) (b)

(c) (d)

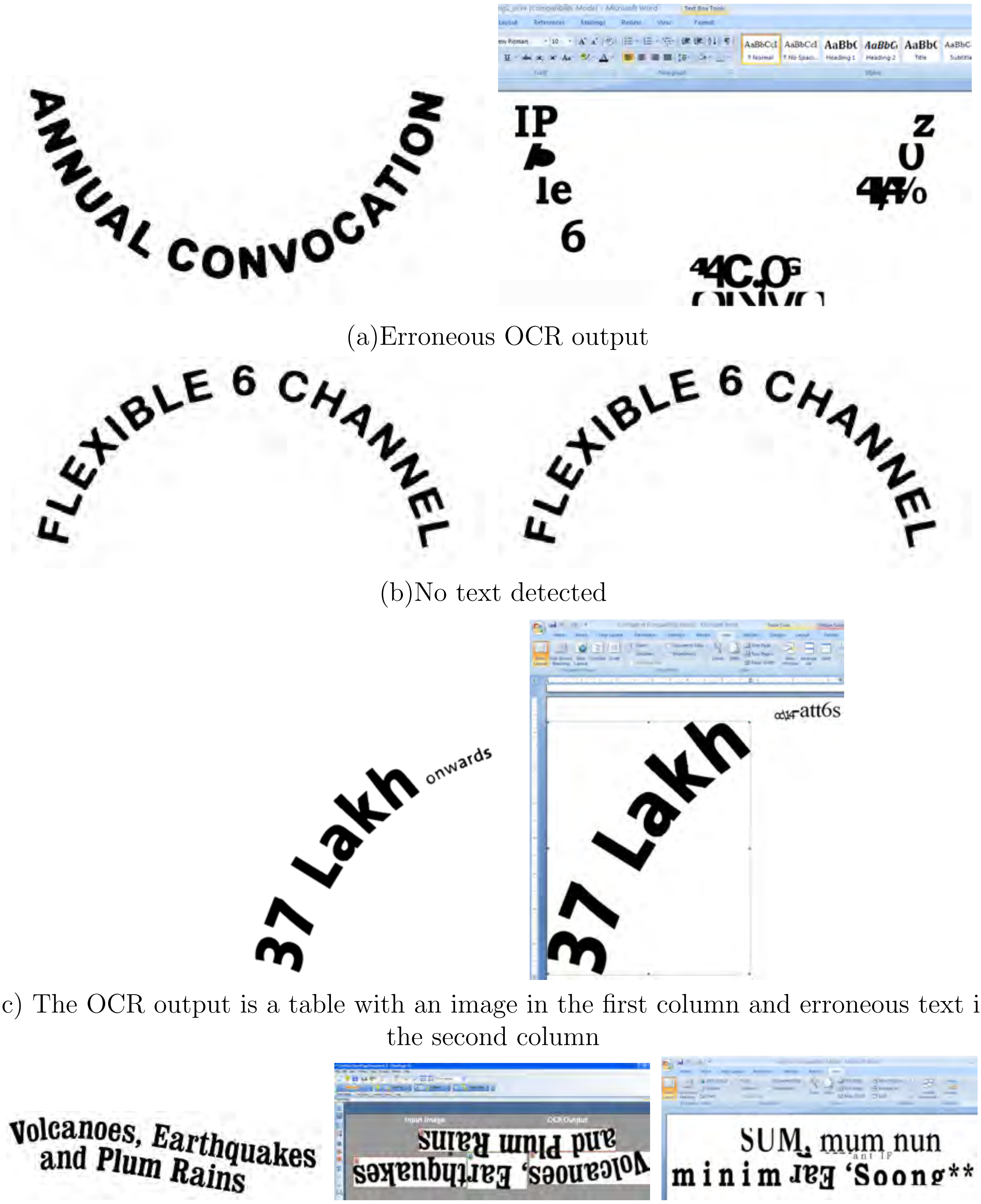
Figure 5.7: (a) Input text string (b) Control points derived from the centroids of the characters (c) B-spline curve fitting and the estimated normal vectors (d) Horizontally aligned image.

the curvature of the text string and is used for aligning the rotated characters. The points where the spline curve intersects each character of a text string are determined by computing the mid-point of all the pixels of each character intersected by the fitted curve. The rotated characters are then arranged such that these computed mid-points are aligned with the horizontal axis. However, it may be observed that ‘small’ vertical misalignments still remain due to the zig-zag pattern of the centroids of the characters in a text string. Fig 5.7 shows the results of all the intermediate steps involved in the alignment process of a sample text string.

5.4 Experiments and results

There are no standard databases available for alignment of curved text. The ICDAR 2003 robust reading competitions dataset contains only horizontal text. Hence, in order to test the performance of the method, we have collected a set of 50 images captured using a hand-held camera. Even though the database is small, it contains all the possibilities of text orientations and layout styles such as arc, wave, triangular and a combination of these with linearly skewed text lines. We use Nuance Omnipage professional 16 (trial version) OCR software to evaluate the performance of our method. Some example outputs of applying OCR directly on the input images are shown in Figure 5.8. Without the text alignment preprocessing step, the OCR software yields only erroneously recognized characters in most cases (see 5.8(a)) or even fails to detect any text at all (see 5.8(b)) in some images. Figure 5.8(c) shows a case where a curved text string is detected as a 2-column table with a partial image in one column and text in the other. Figure 5.8(d) shows a particular case where the input image is automatically inverted by the software before recognition and subsequently identifies 3 text blocks and 1 image block. The resulting OCRed output is therefore fully erroneous. Clearly, a document input with straight text lines seems to be necessary for current OCR systems to work reliably. After applying the proposed text alignment technique, the recognition accuracy improves significantly. Some example results of text alignment and the corresponding OCR outputs are shown in Table 5.1. The data set contains 604 characters in all, out of which 25 characters are erroneously recognized yielding an overall recognition accuracy of 95.9%. In the last row of Table 5.1, a character ‘R’ of the rectified word image is wrongly recognized. This may be attributed to its unconventional font style. Since the OCR software yields only erroneous results on the raw input images, they are not included in the table.

Performance comparison with other methods: The method proposed in [92] can only deal with uppercase English letters laid out in the form of an arc; it will fail in all other cases. The methods proposed by Hase et al.[91] and Vasudev et al.[92] can only process a single text string at a time and cannot handle the presence of multiple text lines such as the input image shown in Figure 5.5(a), where text is laid out in a wavy fashion. Our method successfully handles any type of text orientation without any prior knowledge of the orientation of the text string. Figure 5.5(b) shows an example image which contains text oriented in both vertical as well as horizontal directions for which methods such as [91] that use polynomial fitting will have numerical problems. The usual practice of swapping the x and y coordinates when dealing with vertically aligned text will be ineffective in this case. Since we use B-splines, our method can deal with any type of text orientation.



(a)Erroneous OCR output

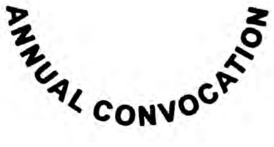
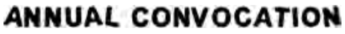
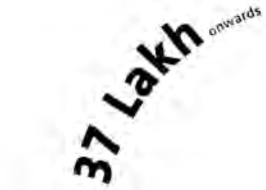
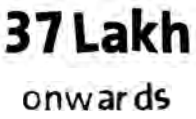
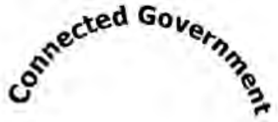
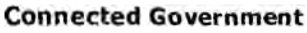
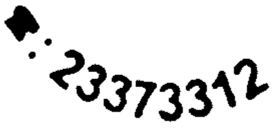
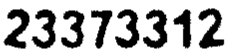
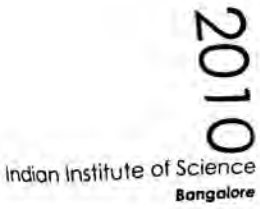
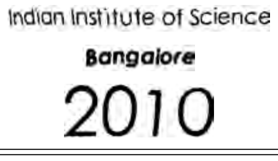
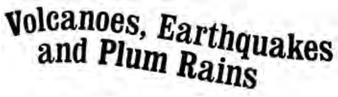
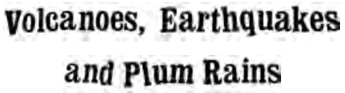
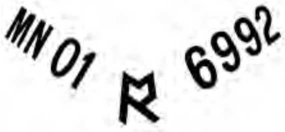
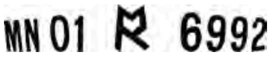
(b)No text detected

(c) The OCR output is a table with an image in the first column and erroneous text in the second column

(d)Image inverted by software and the corresponding OCR output

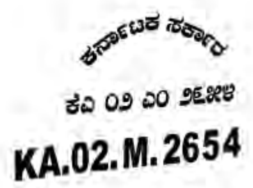
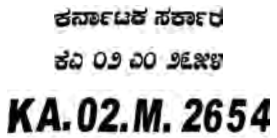
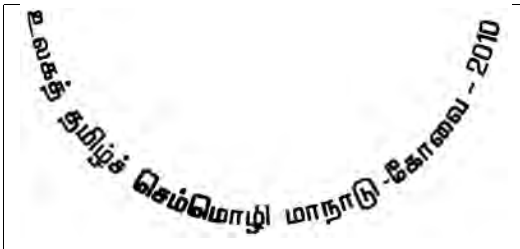
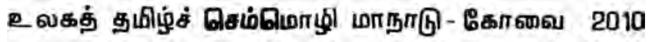
Figure 5.8: OCR readability on images with curved text lines.

Table 5.1: Results of the proposed method on images with multi-oriented text. The corresponding horizontally aligned images and the OCR outputs obtained on them using Omnipage Professional 16 (trial version) are shown.

Input images	Rectified character strings	Corresponding OCR outputs
		ANNUAL CONVOCATION
		37 Lakh onwards
		Connected Government
		23373312
		Indian Institute of Science Bangalore 2010
		Volcanoes, Earthquakes and Plum Rains
		MN 01 i.I 6992

Once the individual text strings are identified and the corresponding B-spline fit is obtained, the estimation of normals does not depend on the individual characters. Therefore, the method can handle a variety of text layouts. The method relies only on the overall curvature of the text string and does not assume any characteristics of the script. Table 5.2 shows some example outputs of the method on multi-script documents containing containing English, Kannada and Tamil texts.

Table 5.2: Example outputs of the proposed method on images with multi-script content.

Input images	Rectified character string images
	
	

5.5 Conclusions

We have proposed a novel method for the alignment of character strings laid out in arbitrary orientations. Just like the skew detection and correction steps in conventional OCRs, alignment of curvilinear to rectilinear text is an indispensable preprocessing step in the analysis of newer document types that contain multi-oriented text strings. The effectiveness of the method is amply illustrated by our experiments with various orientations of text strings and multi-script document images.

The text grouping step successfully identifies all the individual text lines present in the images regardless of their orientations. However, due to the presence of acsenders and descenders in lower case letters, the positions of the centroid exhibit a zig-zag pattern resulting in small errors in the local skew estimate of some characters. This is observed to be within the skew tolerance of the OCR system. The recognition accuracy after text alignment is comparable to that of unskewed text.

Chapter 6

Conclusion and future work

"Every end is a new beginning." - *Anonymous*

6.1 Conclusion

Traditional scanner-based OCR systems cannot be directly applied on camera-captured images due to the variations in the imaging condition as well as the target document type. In this thesis, we propose novel pre-processing techniques that can enable the use of off-the-shelf commercial OCR packages on camera-captured images without any modification of the OCR. All the methods make use of CCs since they are known for their robustness to complex backgrounds, variations in font styles, size and color. Our focus has been on accurately extracting CCs even in the presence of complex backgrounds so as to harness the potential advantages of CC-based methods. These CC-based methods are applicable to documents with multi-script content and arbitrary text layouts. Since CC labeling is a fundamental processing step in many OCR systems, the proposed methods can easily be integrated with other processing modules required for character recognition.

6.2 Main contributions of the thesis

The main contributions of the thesis are summarized below.

- proposing connected component descriptor as a robust feature that increases the stability of feature localization and matching in spite of the multiple instances of same characters in the images. The proposed new features are largely invariant to scaling, rotation, translation and even perspective distortion.
- proposing HCCA and COCOCLUST, parameter-free approaches for robust binarization of camera captured images, that can handle text of any font, size, color, orientation, script and polarity.
- proposing a new scene text extraction method that works on multiscript, multioriented color text strings. This uses a combination classifier to verify text CCs, in addition to regularity constraints.
- proposing a novel method to horizontally align multiple arbitrarily curved, multiscript text strings from binarized images. It renders layouts such as arc, wave, triangular or a combination of them with linearly skewed text to a form suitable to be directly processed by current OCRs.

6.3 Scope for future work

The major challenge in camera-based document image analysis is to render any camera-captured image into an image that is as close to that of scanned document as possible. Some of the recent commercial OCR packages, such as Omnipage Professional 16, have 3D technology to deal with perspective distortion and warping of curled document pages captured from a camera. However, they cannot deal with scene images. Camera-based OCR technology is still in its infancy and there is a lot of scope for advancing the available technology in this new field.

The capability of extracting and recognizing characters present in scene images will allow users to ‘scan’ the world providing access to an endless range of information. This will help bridge the gap between physical and digital worlds with immense scope for newer applications. Pervasive use of camera phones and hand-held digital cameras has a lot of promising applications. Prototype systems are being developed for recognizing faces, road signs, text, translating documents and performing image-based search. Integrating image capture, processing, and network communication on a portable hand-held device can offer seamless interaction between electronic media and the physical world around us.

Publications out of this thesis

Patent

1. T. Kasar and A. G. Ramakrishnan, "*A method to extract and align text of an image captured and a system thereof*", Indian Patent Office Reference. No: 109/CHE/2011, 2011.

Journal Papers

1. T. Kasar and A. G. Ramakrishnan, "*Alignment of Curved Character Strings for Enhanced OCR Readability*", Submitted to Intl Jl. Image and vision Computing, Elsevier Science, 2011.

2. T. Kasar and A.G Ramakrishnan, "*DeteXt: Text Detection in Natural Scene Images*", manuscript under preparation.

Conference Papers

1. T. Kasar and A.G. Ramakrishnan, *Multiscript and multioriented text localization from scene images*", Proc. 4th International Workshop on Camera-based Document Analysis and Recognition, pp. 15-20, 2011, Beijing, China.

2. T. Kasar, D. Kumar, A. M. Prasad, D. Girish and A.G. Ramakrishnan, MAST: *Multi-script annotation toolkit for scenic text*", Proc. 2nd International Workshop on Multilingual OCR, 2011, Beijing, China.

3. T. Kasar, A.G. Ramakrishnan, Amey A. P. Dharwadker and Abhishek Sharma, *TexTraCC: Text Extraction Using Color Connected Components*", Submitted to EE centenary conference, 2011, IISc Bangalore, India.

4. T. Kasar and A. G. Ramakrishnan, "*COCOCLUST: Contour-based Color Clustering for Robust Binarization of Colored Text*", Proc. 3rd Intl. Workshop on Camera Based Document Analysis and Recognition, pp. 11-17, 2009, Barcelona, Spain.

5. T. Kasar and A. G. Ramakrishnan, "*CCD: Connected Component Descriptor for Robust Mosaicing of Camera-Captured Document Images*", Proc. 8th IAPR Intl. Workshop Document Analysis Systems, pp. 480-487, 2008, Nara, Japan.

6. T. Kasar, J. Kumar and A.G Ramakrishnan, "*Specialized Text Binarization Technique for Camera-based Images*", Proc. Workshop on Image and Signal Processing, pp.28-31, 2007, Guwahati, India.

7. T. Kasar, J. Kumar and A. G. Ramakrishnan, "*Font and Background Color Independent Text Binarization*", Proc. 2nd Intl. Workshop on Camera Based Document Analysis and Recognition (a satellite workshop of ICDAR 2007), pp. 3-9, 2007, Curitiba, Brazil.

8. J. Kumar, T. Kasar and A. G. Ramakrishnan, "*Edge-Based Connected Component Analysis for Skew Correction of Complex Color Document Images*", Proc. IEEE TENCON DOI: 10.1109/TENCON.2007.4429083, Taipei, Taiwan, 2007.

9. T. Kasar and A. G. Ramakrishnan, "*Block-Based Feature Detection and Matching for Mosaicing of Camera-Captured Document Images*" Proc. IEEE TENCON, DOI: 10.1109/TENCON.2007.4429095, Taipei, Taiwan, 2007.

Bibliography

[1] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2963 –2970, 2010.

[2] D. Doermann, J. Liang, and H. Li, "Progress in camera-based document image analysis," *Proc. Intl. Conf. Document Analysis and Recognition*, vol. 1, pp. 606–615, 2003.

[3] Gartners Report, *http://www.gartner.com/it/page.jsp?id=1278413.*

[4] X. Liu and D. Doermann, "Computer vision and image processing large techniques for mobile applications," *LAMP-TR-151, November*, 2008.

[5] M. J. Black, F. Berard, A. Jepson, W. Newman, E. Saund, G. Socher, and M. J. Taylor, "The digital office: Overview," *AAAI Spring Symposium on Intelligent Environments*, 1998.

[6] J. Coughlan, R. Manduchi, and H. Shen, "Cell phone-based wayfinding for the visually impaired," *Proc. International Workshop on Mobile Vision*, 2006.

[7] X. Liu and J. Samarabandu, "An edge-based text region extraction algorithm for indoor mobile robot navigation," *Proc. IEEE International Conference on Mechatronics and Automation*, pp. 701 – 706, 2005.

[8] Y. Watanabe, K. Sono, K. Yokomizo, and Y. Okada, "Translation camera on mobile phone," *Proc. Intl. conf. Mutlimedia and Expo*, vol. 2, pp. 177 – 180, 2003.

[9] I. Haritaoglu, "Scene text extraction and translation for handheld devices," *Proc. IEEE Conf. Computer vision and Pattern Recognition*, vol. 2, pp. 408 – 413, 2001.

[10] L. Jagannathan and C. V. Jawahar, "Crosslingual access of textual information using camera phones," *Proc. Intl. Conf. Cognition and Recognition*, pp. 655 – 660, 2005.

[11] KABIS Automatic Book scanners, *http://www.kirtas.com/kabisIII.php.*

[12] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977 – 997, 2004.

[13] R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics and Applications*, vol. 16, p. 22 30, 1996.

[14] S. Peleg and J. Herman, "Panoramic mosaicing with videobrush," *DARPA Image Understanding Workshop*, pp. 261–264, 1997.

[15] M. Irani and P. Anandan, "Video indexing based on mosaic representation," *Proc. IEEE*, vol. 86, no. 5, pp. 905–921, 1998.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2002.

[17] C. Harris and M. Stephens, "A combined corner and edge detector," *In Proc. 4th Alvey Vision Conf.*, pp. 147–151, 1988.

[18] D. Lowe, "Object recognition from local scale-invariant features," *Proc. Intl. Conf. Computer Vision*, pp. 1150–1157, 1999.

[19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Proc. IEEE Conf. Computer Vision Patt. Recog.*, vol. 2, pp. 257–263, 2003.

[20] J. Lian, D. DeMenthon, and D. Doermann, "Camera-based document image mosaicing," *Proc. Intl. Conf. Pattern Recognition*, 2006.

[21] E. N. Mortensen, H. Deng, and L. Shapiro, "A SIFT descriptor with global context," *Proc. IEEE Conf. Computer Vision Patt. Recog.*, vol. 1, pp. 184–190, 2005.

[22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Intl. Jl. Computer Vision*, vol. 65, pp. 43–72, 2005.

[23] A. P. Wichello and H. Yan, "Document image mosaicing," *Proc. Intl. Conf. Pattern Recognition*, vol. 2, pp. 1081–1083, 1998.

[24] M. Pilu and F. Isgro, "A fast and reliable planar registration method with applications to document stitching," *Proc. IEEE Workshop on Application of Computer Vision*, pp. 245–250, 2002.

[25] S. M. Smith and J. M. Brady, "Susan: A new approach to low-level image processing," *Intl. Jl. Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[26] A. Zappala, A. Gee, and M. Taylor, "Document mosaicing," *Image and Vision Understanding*, vol. 17, pp. 589–595, 1999.

[27] T. Kasar, J. Kumar, and A. G. Ramakrishnan, "Font and background color independent text binarization," *Proc. Workshop on Camera Based Document Analysis and Recognition*, pp. 3–9, 2007.

[28] J. Canny, "A computational approach to edge detection," *IEEE Trans. PAMI*, vol. 8, no. 6, p. 679 698, 1986.

[29] Y. S. Kim and W. Y. Kim, "A new region-based descriptor," *ISO/IEC MPEG99/M5472, Maui, Hawaii*, Dec 1999.

[30] M. Bober, "MPEG-7 visual shape descriptors," *IEEE Trans. Circuits Systems and Video Technology*, vol. 11, no. 6, pp. 716–719, 2001.

[31] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commmunications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[32] G. Wolberg, *Digital Image Warping.* IEEE computer Society Press, 1990.

[33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems Man Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[34] J. N. Kapur, P. K. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision Graphics Image Processing*, vol. 29, pp. 273–285, 1985.

[35] W. Tsai, "Moment preserving thresholding: A new approach," *Computer Vision Graphics Image Processing*, vol. 29, pp. 377–393, 1987.

[36] T. Pun, "Entropic thresholding: A new approach," *Computer Graphics and Image Processing*, vol. 16, pp. 210–239, 1981.

[37] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[38] W. Niblack, *An Introduction to Digital image processing.* Englewood Cliffs, N.J., Prentice Hall, 1986.

[39] S. Yanowitz and A. Bruckstein, "A new method for image segmentation," *Computer Vision, Graphics and Image Processing*, vol. 46, no. 1, pp. 82–95, 1989.

[40] O. D. Trier and A. Jain, "Goal-directed evaluation of binarization methods," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, pp. 1191–1201, 1995.

[41] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, pp. 225–236, 2000.

[42] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, 2004.

[43] C. Wolf, J. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," *Proc. Intl. Conf. Pattern Recognition*, vol. 4, pp. 1037–1040, 2002.

[44] L. O'Gorman, "Binarization and multithresholding of document images using connectivity," *Graph Model Image Processing*, vol. 56, no. 6, pp. 494–506, 1994.

[45] F. Murtagh and J. Starck, "Quantization from Bayes factors with application to multilevel thresholding," *Patt. Recog. Letters*, vol. 24, no. 12, pp. 2001–2007, 2003.

[46] C. Chang and L. Wang, "A fast multilevel thresholding method based on lowpass and highpass filtering," *Pattern Recognition Letters*, vol. 18, no. 14, pp. 1469– 1478, 1997.

[47] M. Sezgin and R. Tasaltin, "A new dichotomization technique to multilevel thresholding devoted to inspection applications," *Pattern Recognition Letters*, vol. 21, no. 2, pp. 151–161, 2000.

[48] E. Badekas, N. Nikolaou, and N. Papamarkos, "Text binarization in color documents," *Intl. Jl. Imaging, Systems and Technolology*, vol. 16, pp. 262–274, 2007.

[49] K. Sobottka, H. Bunke, and H. Kronenberg, "Identification of text on colored book and journal covers," *Proc. Intl. conf. Document Analysis and Recognition*, pp. 57–62, 1999.

[50] T. Kasar and A. G. Ramakrishnan, "COCOCLUST: Contour-based color clustering for robust binarization of colored text," *Proc. Intl. Workshop Camera Based Document Analysis and Recognition*, pp. 11–17, 2009.

[51] A. Koschan and M. Abidi, *Digital Color Image Processing, John Wiley & Sons, Inc.* 2008.

[52] A. Antonacopoulos and D. Karatzas, "Fuzzy segmentation of characters in web images based on human colour perception," *Proc. Workshop Document Analysis Systems*, pp. 295–306, 2002.

[53] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," *Pattern recognition*, vol. 28, no. 10, pp. 1523 – 1535, 1995.

[54] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Trans. on Image Processing*, vol. 9, no. 1, pp. 147 – 156, 2000.

[55] Y. Liu, S. Goto, and T. Ikenaga, "A robust algorithm for text detection in color images," *Proc. Intl. Conf. Doc. Anal. Recog.*, vol. 1, pp. 399–403, 2005.

[56] S. S. Raju, P. B. Pati, and A. G. Ramakrishnan, "Gabor filter based block energy analysis for text extraction from digital document images," *Proc. Intl. Workshop on Document Image Analysis for Libraries*, pp. 233 – 243, 2004.

[57] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: an automatic system to detect and recognize text in images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1124 – 1129, 1999.

[58] P. Clark and M. Mirmehdi, "Finding text using localised measures," *Proc. British machine vision conference*, pp. 675 – 684, 2000.

[59] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," *Proc. IEEE Intl. Conf. Computer Vision Pattern Recognition*, vol. 2, pp. 366 – 373, 2004.

[60] P. Shivakumara, A. Dutta, C. L. Tan, and U. Pal, "A new wavelet-median-moment based method for multi-oriented video text detection," *Proc. Intl. Workshop on Document analysis and systems*, pp. 279 – 286, 2010.

[61] B. Gatos, I. Pratikakis, K. Kepene, and S. J. Perantonis, "Text detection in indoor/outdoor scene images," *Proc. Intl. Workshop Camera Based Document Analysis and Recognition*, pp. 127 – 132, 2005.

[62] K. Zhu, F. Qi, R.Jiang, L. Xu, M. Kimachi, Y. Wu, and T. Aizawa, "Using adaboost to detect and segment characters from natural scenes," *Proc. Intl. Workshop Camera Based Document Analysis and Recognition*, pp. 52–59, 2005.

[63] W. Pan, T. Brui, and C. Suen, "Text detection from scene images using sparse representation," *Proc. Intl. Conference on Pattern Recognition*, pp. 1 – 5, 2008.

[64] B. Wang, X. F. Li, F. Liu, and F. Q. Hu, "Color text image binarization based on binary texture analysis," *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, vol. 3, pp. 585 – 588, 2004.

[65] A. Jain and B. Yu., "Automatic text location in images and video frames," *Pattern Recognition*, vol. 3, no. 12, pp. 2055–2076, 1998.

[66] Y. Zhong, K. Karu, and A. Jain, "Locating text in complex color images," *Pattern Recognition*, vol. 28, no. 10, pp. 1523 – 1535, 1995.

[67] Y. J. Song, K. C. Kim, Y. W. Choi, H. R. Byun, S. H. Kim, S. Y. Chi, D. K. Jang, and Y. K. Chung, "Text region extraction and text segmentation on camera-captured style images," *Proc. Intl. Conf. Document Analysis and Recognition*, vol. 1, pp. 172 – 176, 2005.

[68] C. Garcia and X. Apostolidis, "Text detection and segmentation in complex color images," *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2326 – 2329, 2000.

[69] D. Karatzas and A. Antonacopoulos, "Colour text segmentation in web images based on human perception," *Jl. Image Vison Comp.*, vol. 25, no. 5, pp. 564 – 577, 2007.

[70] C. M. Thillou and B. Gosselin, "Color text extraction with selective metric-based clustering," *Computer Vision and Image Understanding*, vol. 107, pp. 97 – 107, 2007.

[71] T. Y. Zhang and C. Suen, "A fast parallel algorithm for thinning digital patterns," *Proc. IEEE Intl. Conf. Comp. Vision Patt. Recog.*, vol. 27, no. 3, pp. 236 – 239, 1984.

[72] C. C. Chang and C. Lin, *LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/∼cjlin/libsvm.*

[73] T. Kasar, D. Kumar, M. N. A. Prasad, D. Girish, and A. G. Ramakrishnan, *MAST: Multi-scipt Annotation Toolkit for Scenic Text, http://mile.ee.iisc.ernet.in/mast.*

[74] S. M. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained adaboost algorithm," *Proc. Intl. Conf. Document Analysis and Recognition*, pp. 1 – 5, 2009.

[75] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui, "Snoopertext: A multiresolution system for text detection in complex visual scenes," *Proc. IEEE Intl. Conf. on Image Processing*, pp. 3861 – 3864, 2010.

[76] L. Nuemann and J. Matas, "A method for text localization and recognition in real-world images," *Proc. Asian Conf. Computer Vision*, vol. 3, pp. 770 –783, 2010.

[77] A. Bagdanov and J. Kanai, "Projection profile based skew estimation algorithm for JBIG compressed images," *Proc. Intl. Conf. Document Analysis and Recognition*, pp. 401 – 405, 1997.

[78] H. Baird, "The skew angle of printed documents," *Proc. Conf. Society of Photographic Scientists and Engineers*, vol. 40, pp. 21 – 24, 1987.

[79] U. Pal and B. Chaudhuri, "An improved document skew angle estimation technique," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 899 – 904, 1996.

[80] S. N. Srihari and V. Govindaraju, "Analysis of textual images using the hough transform," *Machine Vision and Applications*, vol. 2, no. 3, pp. 141 – 153, 1989.

[81] B. Yu and A. Jain, "A fast and robust skew detection algorithm for generic documents," *Pattern Recognition*, vol. 29, no. 10, pp. 1599 – 1629, 1996.

[82] O. Okun, M. Pietikainen, and J. Sauvola, "Document skew estimation without angle range restriction," *Intl. Jl. Doc. Anal.s and Recog.*, vol. 2, pp. 132 – 144, 1999.

[83] T. Steinherz, N. Intrator, and E. Rivlin, "Skew detection via principal components analysis," *Proc. Intl. Conf. Doc. Anal. and Recog.*, pp. 153 – 156, 1999.

[84] A. Hashizume, P. Peh, and A. Rosenfeld, "A method for detecting the orientation of aligned component," *Pattern Recognition Letters*, vol. 4, pp. 125 – 132, 1986.

[85] Y. Lu and C. L. Tan, “A nearest-neighbor chain based approach to skew estimation in document images,” *Jl. Patt. Recog. Letters*, vol. 24, no. 14, pp. 2315 – 2323, 2003.

[86] K. Mahata and A. Ramakrishnan, “Precision skew detection through principal axis,” *Proc. Intl. Conf. Multimedia Processing and Systems*, pp. 186 – 188, 2000.

[87] U. Pal, M. Mitra, and B. Chaudhuri, “Multi-skew detection of Indian script documents,” *Proc. Intl. Conf. Document analysis and Recognition*, pp. 292 – 296, 2001.

[88] P. Saragiotis and N. Papamarkos, “Local skew correction in documents,” *Intl. Jl. Pattern Recognition and Artificial Intelligence*, vol. 22, no. 4, pp. 691 – 710, 2008.

[89] Z. Zhang and C. Tan, “Correcting document image warping based on regression of curved text lines,” *Proc. Int. Conf. Document Analysis and Recognition*, vol. 1, pp. 589–593, 2003.

[90] S. Uchida, M. Sakai, M. Iwamura, S. Omachi, and K. Kise, “Skew estimation by instances,” *Proc. Intl. Workshop Document Analysis Systems*, pp. 201 – 208, 2008.

[91] H. Hase, M.Yoneda, T. Shinokawa, and C. Y. Suen, “Alignment of free layout color texts for character recognition,” *Proc. Int. Conf. Document Analysis and Recognition*, pp. 932 – 936, 2001.

[92] T. Vasudev, G. Hemanthkumar, and P. Nagabhushan, “Transformation of arc-form-text to linear-form-text suitable for OCR,” *Pattern Recognition Letters*, vol. 28, pp. 2343–2351, 2007.

[93] D. Rogers and J. Adams, *Mathematical Elements for Computer Graphics.* McGraw-Hill, 1990.