

1-1-2008

Techniques for document image processing in compressed domain

Shulan Deng
University of Nevada, Las Vegas

Follow this and additional works at: <https://digitalscholarship.unlv.edu/rtds>

Repository Citation

Deng, Shulan, "Techniques for document image processing in compressed domain" (2008). *UNLV Retrospective Theses & Dissertations*. 2442.
<http://dx.doi.org/10.25669/03y8-qzmr>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Dissertation has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

TECHNIQUES FOR DOCUMENT IMAGE PROCESSING
IN COMPRESSED DOMAIN

by

Shulan Deng

Bachelor of Engineering
Beijing University of Aeronautics and Astronautics
Beijing, China
1991

Master of Science
Beijing University of Aeronautics and Astronautics
Beijing, China
1994

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy Degree
Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2000

UMI Number: 3001913



UMI Microform 3001913

Copyright 2001 by Bell & Howell Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346



Dissertation Approval
The Graduate College
University of Nevada, Las Vegas

November 15, 2000

The Dissertation prepared by

Shulan Deng

Entitled

Document Image Processing in Compressed Domain

is approved in partial fulfillment of the requirements for the degree of

Doctor of Philosophy


Examination Committee Chair


Dean of the Graduate College


Examination Committee Member


Examination Committee Member


Examination Committee Member


Graduate College Faculty Representative

ABSTRACT

Techniques for Document Image Processing in Compressed Domain

by

Shulan Deng

Dr. Shahram Latifi, Examination Committee Chair
Professor of Electrical and Computer Engineering
University of Nevada, Las Vegas

The main objective for image compression is usually considered the minimization of storage space. However, as the need to frequently access images increases, it is becoming more important for people to process the compressed representation directly. In this work, the techniques that can be applied directly and efficiently to digital information encoded by a given compression algorithm are investigated. Lossless compression schemes and information processing algorithms for binary document images and text data are two closely related areas bridged together by the fast processing of coded data. The compressed domains, which have been addressed in this work, i.e., the ITU fax standards and JBIG standard, are two major schemes used for document compression. Based on ITU Group IV, a modified coding scheme, MG4, which explores the 2-dimensional correlation between scan lines, is developed. From the viewpoints of compression efficiency and processing flexibility of image operations, the MG4 coding principle and its feature-preserving behavior in the compressed domain are investigated and examined. Two popular coding schemes in the area of

bi-level image compression, run-length and Group IV, are studied and compared with MG4 in the three aspects of compression complexity, compression ratio, and feasibility of compressed-domain algorithms. In particular, for the operations of connected component extraction, skew detection, and rotation, MG4 shows a significant speed advantage over conventional algorithms. Some useful techniques for processing the JBIG encoded images directly in the compressed domain, or concurrently while they are being decoded, are proposed and generalized.

In the second part of this work, the possibility of facilitating image processing in the wavelet transform domain is investigated. The textured images can be distinguished from each other by examining their wavelet transforms. The basic idea is that highly textured regions can be segmented using feature vectors extracted from high frequency bands based on the observation that textured images have large energies in both high and middle frequencies while images in which the grey level varies smoothly are heavily dominated by the low-frequency channels in the wavelet transform domain. As a result, a new method is developed and implemented to detect textures and abnormalities existing in document images by using polynomial wavelets. Segmentation experiments indicate that this approach is superior to other traditional methods in terms of memory space and processing time.

TABLE OF CONTENTS

ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
ACKNOWLEDGMENTS	x
CHAPTER 1 INTRODUCTION	1
Document Images and Document Analysis	1
Document Analysis in Compressed Domain	4
Research Problems	7
Dissertation Overview	8
CHAPTER 2 DOCUMENT IMAGE COMPRESSION	9
Introduction	9
Overview of Related Work	11
CHAPTER 3 MODIFIED GROUP IV	15
Group IV Coding Scheme	15
Modified Group IV (MG4)	17
New Vertical Mode	19
New Pass mode	19
New Horizontal Modes	21
Coding Procedure	21
Experimental Results for Compression	24
Mode Analysis	26
A General Procedure for MG4 Compressed-domain Processing	29
Applications	32
Connected Component Extraction Analysis	33
Skew Detection	36
Skew Correction	42
Evaluation of MG4	47
Conclusion	49
CHAPTER 4 COMPRESSED-DOMAIN TECHNIQUES FOR JBIG-ENCODED IMAGES	53

Introduction	53
JBIG Compression Scheme.....	54
Compressed-domain Processing	56
Connected Component Detection	58
Page Segmentation	62
Data Complexity Analysis	65
Experimental Results	65
Conclusion	68
 CHAPTER 5 WAVELET TRANSFORMS	75
Introduction	75
Wavelets and their Characteristics	76
Discrete Wavelet Transform	77
 CHAPTER 6 DOCUMENT SEGMENTATION USING WAVELET METHOD	80
Introduction	90
Document Segmentation	83
B-spline Wavelets	85
Proposed Segmentation Algorithm	86
Experimental Results	90
Effects of System Parameters and Time Analysis	97
Decomposition Levels	100
Frequency Bands	101
Feature Selection	102
Wavelet Functions	102
Time Analysis	103
Summary and Conclusion	107
 CHAPTER 7 CONCLUSIONS AND FUTURE WORK	109
Summary of the Contributions	111
Future Work	112
 BIBLIOGRAPHY	114
 VITA	123

LIST OF TABLES

Table 3.1	Compression Performance of MG4	25
Table 3.2	Experimental Results Using the MG4 Coding Scheme	25
Table 3.3	Frequency of Vertical Modes $V(a, b)$ When $ a \leq 2$ and $ b \leq 2$	28
Table 3.4	Skew Detection Performance of MG4	42
Table 3.5	Measured Time Required for Rotation Based on MG4 (CPU seconds)	47
Table 3.6	Summary of Evaluation of the Three Algorithms	49
Table 4.1	Time Comparison for High and Low Resolution Processing	67
Table 6.1	The Percentage of Segmentation Error Using Different Levels of Cubic B-spline Wavelets	101
Table 6.2	The Percentage of Segmentation Error (P_e) Using Different Combination of Frequency Bands	102
Table 6.3	Segmentation Results Comparing the Performance of Different Wavelet Functions by the Number of Images with a Pe in the range of (a% - b%)	103

LIST OF FIGURES

Figure 1.1	Document Image: ITU1	3
Figure 1.2	A Hierarchy of Document-Processing Subcategories	4
Figure 1.3	A Lossless Compression System	5
Figure 1.4	Application: (a) Conventional; (b) Proposed	6
Figure 3.1	The Changing Picture Elements	16
Figure 3.2	G4. Pass Mode	17
Figure 3.3	G4. Vertical Mode and Horizontal Mode	17
Figure 3.4	MG4. Reference Run and Coding Run	18
Figure 3.5	MG4. Vertical Mode	20
Figure 3.6	MG4. Pass Mode P(0)	20
Figure 3.7	MG4. Pass Mode P(2)	20
Figure 3.8	MG4. Horizontal Mode.....	21
Figure 3.9	MG4 Coding Scheme Flow Diagram	23
Figure 3.10	An Example of Characters "A B " and Their MG4 Encoding	24
Figure 3.11	Mode Analysis for Characters	27
Figure 3.12	Frequency Analysis of the Three Modes of MG4	28
Figure 3.13	Frequency Analysis for Pass Mode P(n)	29
Figure 3.14	MG4. Building Database for Connected Component Extraction	35
Figure 3.15	MG4. Flow Chart of Connected Component Extraction.....	36
Figure 3.16	Time Comparison for Connected Component Extraction Between MG4 Algorithm and Bartneck Algorithm	37
Figure 3.17	Detecting the Key Points for the Skew Angle Represented by the Horizontal Mode	39
Figure 3.18	Distribution of Horizontal Modes in ITU 1	40
Figure 3.19	Energy Projection for ITU 5 with a Skew Angle of 5°	41
Figure 3.20	Rotating an Image in the MG4 Domain.....	44
Figure 3.21	Original Image: ITU 7	51
Figure 3.22	Finding the Beginning Points and Ending Points of Vertical Runs...	52
Figure 3.23	An Example of Rotation	52
Figure 4.1	Two-line Template for the JBIG Context	55
Figure 4.2	Pixels Used to Determine the Value of a Lower-level Pixel	56
Figure 4.3	Concurrent Processing of JBIG Decompression and Feature Extraction	57
Figure 4.4	An Example of a Simple Binary Image and its Corner Image.....	59
Figure 4.5	16 Possible Patterns of the 2 × 2 Masks.....	60
Figure 4.6	Setting Corner Patterns into the JBIG Context.....	60
Figure 4.7	Connected Component Detection	62
Figure 4.8	Automaton to Detect White Runs	64

Figure 4.9	The Original Document Image	69
Figure 4.10	(a) The Lowest Resolution Layer of JBIG Compressed Document Images; (b) Horizontal Smearing of (a); (c) Vertical Smearing of (a); (d) Merging (b) and (c) by Using OR Operation; (e) the Bottom up Result from (f); (f) The Image Obtained from Simple Resolution Reduction	70
Figure 4.11	ITU 4 Document Image	71
Figure 4.12	(a) Horizontal Smearing of the Lowest Layer Image; (b) Vertical Smearing of the Lowest Layer Image; (c) Merging (b) and (c) by Using OR Operation; (d) Segmentation Result	72
Figure 4.13	Segmentation Results Based on (a) LRI; (b) FUI	73
Figure 4.14	Corner Detection and Connected Component Detection for (a) LRI (b) FUI	74
Figure 5.1	Perfect Reconstruction Filter Bank for Fast Wavelet Transform Algorithm	79
Figure 6.1	Cubic B-spline. (a) Scaling Function; (b) Wavelet Function	86
Figure 6.2	Performance of the Proposed Segmentation Method on a Document Image. In this image, Black, White and Gray colors Label the Picture, Text and Background, Respectively. (a) Original Image; (b) Output Image of Three-means Segmentation; (c) Horizontal Merging; (d) Vertical Merging; (e) OR Operation on (c) and (d); (f) Median Filtering Result; (g) Dilation Processing; (h) Final Segmentation Result	91
Figure 6.3	An Example of Cubic B-spline Wavelet Transform. (a) Original Image Scanned at 150dpi (829 × 687). (b) One-level Decomposed Image Using a Cubic Spline Wavelet	92
Figure 6.4	Another Example (756 × 595). (a) Original Image Scanned at 200 dpi; (b) Output Image of Three-means Classification	93
Figure 6.5	Performance of the Proposed Method on a Skewed Document Image. (a) Input Document Image Rotated by 10°, Scanned at 300 dpi (872 × 580); (b) Output Image of Three-means Classification	94
Figure 6.6	Performance of the Proposed Segmentation Method on a Handwritten Document Image (775 × 411). (a) Input Handwritten Document Image Scanned at 200 dpi; (b) Output Image of Two-means Classification .	94
Figure 6.7	Cluster Centers of the Nine Features from Three Bands Corresponding to the Image in (a) Figure 6.3(a); (b) Figure 6.5(a) (Unrotated Version); (c) Figure 6.5(a). In these Plots, Text Class is Marked by Solid Line, Picture Class by Dash-dot Line, and Background Class by Doted Line.	95
Figure 6.8	Segmentation Failure Due to Insufficient Distinguishing Power	96
Figure 6.9	Segmentation of a Document Image with the Size of 791 × 1024. (a) Original Image; (b) Output Image of Two-means Classification; (c) Segmentation Results; (d) Extracted Text Image	97
Figure 6.10	Processing of Some Degraded Images. (a) A Degraded Version of Figure 6.9(a); (b) Segmentation Result	99

Figure 6.11 Output Images of 3-means Classification by Applying a Window
Size of (a) 2×2 ; (b) 3×3 ; (c) 4×4 ; (d) 6×6 ; (e) 8×8 ; (f) 16×16
..... 105

Figure 6.12 Extracted Text by Applying a Window Size of (a) 2×2 ; (b) 3×3 ;
(c) 4×4 ; (d) 6×6 ; (e) 8×8 ; (e) 16×16 106

Figure 6.13 Time Comparison by Using Different Block Sizes 107

ACKNOWLEDGMENTS

I would like to take this opportunity to express my sincere gratitude to my advisor, Professor Shahram Latifi, for his financial support, and for his setting a high standard for me, which helped me learn and grow, and for his meticulously correcting my writing and, more importantly, for his continuous encouragement and believing in me.

I wish to thank Dr. Emma Regentova and the members of my dissertation committee: Eugene McGaugh, Henry Selvaraj, Kazem Taghva and Zhiyong Wang. Dr. Regentova was involved in this work and actually contributed to this dissertation. Dr. Wang took time to share his research experiences with me and I enjoyed our discussions. Dr. Taghva gave concise, yet deep comments on my research and I also appreciate his help on accessing ISRI image database. Dr. McGaugh and Dr. Selvaraj have always expressed their concern for this work each time we met.

My sincere thanks go to Edna Zhuo and her husband Sammy Dong for their incessant caring for me. My gratitude also goes to Joan and John McS, whose friendship and love made my school years here easier and more pleasant. I am especially thankful for the constant and unconditional support from my mother, my sisters and my brother throughout the course of this study.

I greatly acknowledge the financial support of NSF Research Grant IRI-9616206, NASA Research Grant NAG5-3994 and the UNLV Graduate College.

I would like to extend my thanks and gratitude to all the professors and graduates in the Electrical and Computer Engineering Department, and the Computer Science Department, my friends and all those who gave me their valuable assistance in the past three and half years.

CHAPTER 1

INTRODUCTION

1.1 Document Images and Document Analysis

Due to the progress in computer and communication technology, documents are increasingly scanned and stored in digital form. The explosion of digital data on the nation's information superhighways brings out a series of problems such as how to collect, transmit, store, and organize information so that necessary access to these data is possible. The workload will be reduced if the data can be compressed beforehand and then be processed in such way that useful characteristics of the compression techniques can be taken advantage of.

Textual objects and images constitute a significant portion of digital information. This dissertation is concerned with the advanced techniques of compressed-domain processing for document images. As we know, documents are part of our daily life and include newspapers, books, memos, manuals, business letters, reports, faxes, etc. They may be captured and digitized from a video source using an Analog to Digital Converter (ADC) device. Obtained digital images consist of an array of picture elements (pixels) resulting from the regular sampling of an analog image. Figure 1.1 presents a typical bilevel document image, which shows that document images contain mostly printed and aligned text.

As many paper documents have been transmitted, received via fax machine, and stored digitally in document databases, people need to view and print them, and more importantly, analyze them. Much research was performed to read text, find

fields, or locate lines or some particular symbols in a given document image or a given document database. Commercial systems are built to conduct these tasks.

The field of document image analysis has grown rapidly in recent years. The first reason for this growth is due to the increasing accuracy rates of omnifont and hand-print optical character recognition (OCR), which results directly from the progress in document image analysis. Secondly, decreasing costs for the computational power such as fast computers, larger computer memory and inexpensive scanners, make it possible to run some sophisticated algorithms. Lastly, with the emergence of new application areas such as the World Wide Web (WWW) and digital libraries, more and more information is stored, propagated, and processed in digital form.

Document images are usually analyzed in the following steps. First, pixel-level processing is performed after a document image is obtained by digital scanning. At this level, the processing includes operations that are applied to all image pixels; for instance, noise removal, image enhancement, and segmentation of image components into text and graphics. Then we proceed to feature-level processing, which treats groups of pixels as entities such as line and curve detection and shape description. The objective of document image analysis is to recognize the text and graphics components in images and extract the intended information as a human would. Therefore, document processing can be broken down into two categories, i.e., textual processing and graphics processing. The former deals with the text part of images. The concerned problems can be recognizing the text by OCR technique, determining the skew, or finding columns, paragraphs, text lines and words. The latter involves the nontextual lines and symbol components that make up line diagrams, logos and so on. Figure 1.2 illustrates such a classification.

THE SLEREXE COMPANY LIMITED

SAPONS LANE - BODLE - DORSET - BH25 8ER

TELEPHONE BODLE (54513) 51617 - TELEX 123654

Our Ref. 150/PJC/EAC

18th January, 1972.

Dr. P.M. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Registered in England. No. 2039
Registered Office: 60 Victoria Lane, Hove, Sussex.

Figure 1.1: Document Image: ITU1

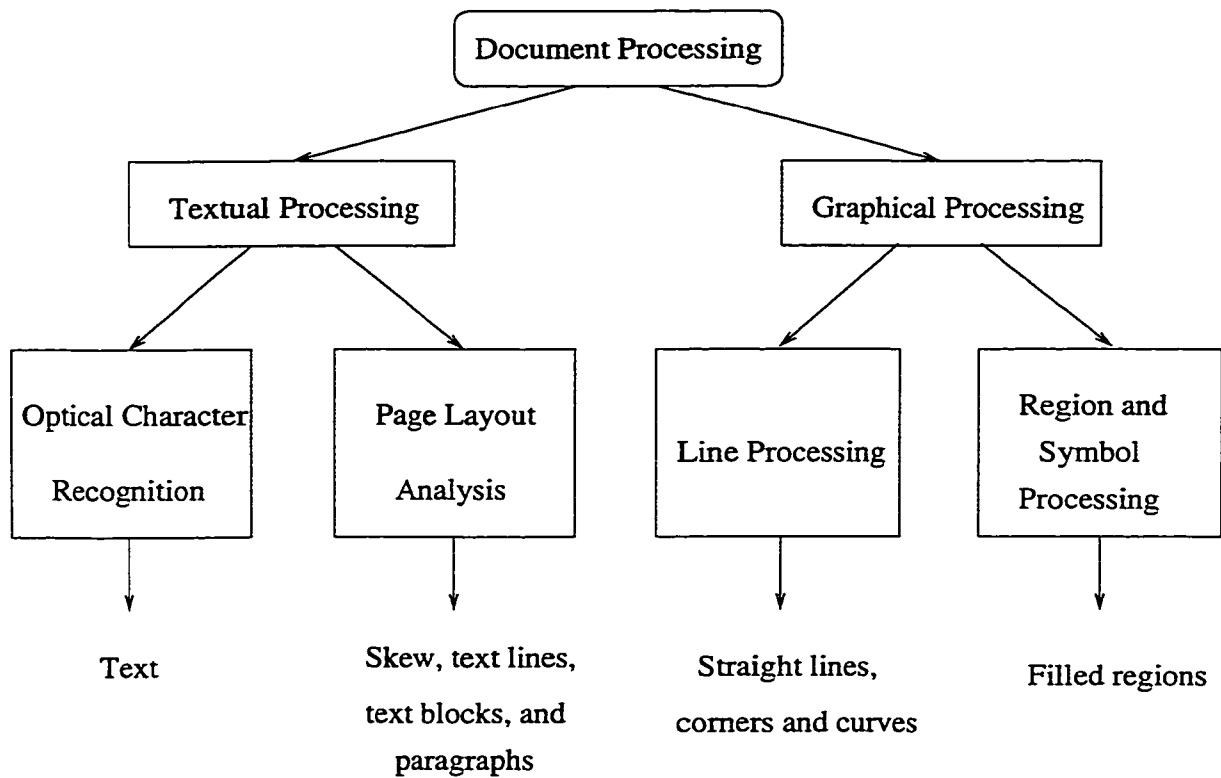


Figure 1.2: A Hierarchy of Document-Processing Subcategories

1.2 Document Analysis in Compressed Domain

Data compression techniques have been widely used to store and exchange digital information [57, 76]. The basic goal of image compression is to convert an original sampled high bit image into a compressed form with a low bit rate. Under the available storage or communication capacity, image compression achieves the best possible quality. The need for data compression is evident. Without compression, a letter sized document image sampled at 300dpi would need about 2 Megabytes of space to store. This is true whether or not the image contains a single line or 20 lines. It would take 2.5 minutes to transmit it over a 56K bits per second telephone line. However, only several kbytes for storage and several seconds for transmission via the same line are needed using an efficient compression technique on the image. Moreover, compressed images have their own advantages over original images. The

resulting benefits are shown in a number of ways:

- Compression conserves storage space so that larger inventories are allowed. Furthermore, lower data rate means reducing the transmission bandwidth.
- Good compression techniques make progressive transmission possible and produce an increasingly good reconstructed image.
- Compression system can mean reduced signal processing.
- Compression system can incorporate encryption for data security.

Figure 1.3 presents a lossless compression system for image data. The compression system consists of two components:

1. Transformation or decomposition: This process takes in the original data as input and transforms or decomposes it into symbols which can be encoded efficiently. The basic function of this step is to reduce data dependency (decorrelation).
2. Symbol encoding: Symbols are encoded as a variable- or fixed-length code.



Figure 1.3: A Lossless Compression System

The data compression research is mainly focused on improving compression ratio under a fixed quality of images recovered from lossy or lossless compression methods. Traditionally, image processing research deals with raw (uncompressed) data. This work attempts to bridge data compression and information processing together,

propose algorithms for the direct processing of images compressed by standard techniques, and more importantly, promote a new way of thinking.

Usually, when people want to browse or process the information stored in a compressed form on a disk, it needs to be loaded into main memory, get decompressed, and then be processed according to the type of operation needed. The processed information can then be compressed and stored back on the disk.

Suppose the data could be processed in its compressed form; two steps including decompression before processing, and compression after processing the data, can essentially be phased out (see Figure 1.4), and some processing time would be saved. In addition, the I/O bandwidth can be utilized more effectively since a smaller size of data needs to be accessed. Furthermore, less memory is required as opposed to the decompressed data, which may have to be broken up and brought into memory in pieces due to lack of sufficient memory space. Direct processing of images has

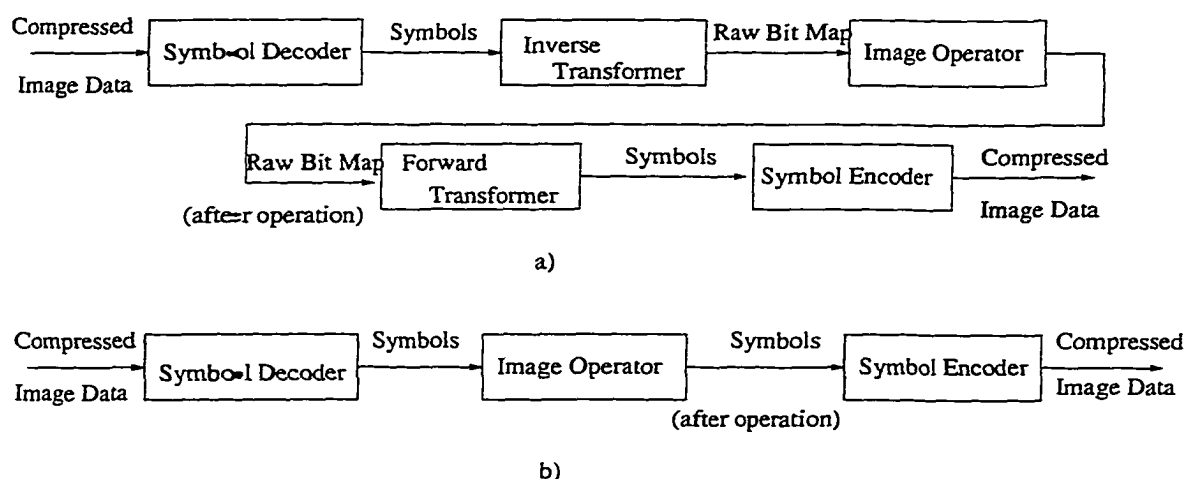


Figure 1.4: Application: (a) Conventional; (b) Proposed

attracted increased attention recently. An extensive literature review in this area is given in Chapter 2.

1.3 Research Problems

Generally speaking, this research investigates the following two problems:

1. develop new compression algorithms that allow given operations to be rapidly performed on compressed data.
2. develop new algorithms for processing compressed data without fully decompressing them.

The first part of this work will focus on lossless compression techniques and information processing algorithms for binary document images and text data. Information in compressed data cannot be easily retrieved or modified without decompression for some coding schemes. It is of interest to analyze current compression technologies and explore a new approach to document image compression that is efficient in both space and time. To transmit and store digital data efficiently, a variety of data compression algorithms have been developed. The ITU fax standards and JBIG standard are widely used for bilevel document image compression. While the research body on compression schemes is quite extensive, algorithms for retrieving and modifying information in compressed data have not been widely investigated. It is highly desirable to develop a new compression scheme whose compression efficiency will be comparable to the traditional compression schemes. Furthermore, under this coding scheme, the rapid execution of a given set of operations should be possible.

To further the effort of compressed-domain processing, the research will concentrate on document images decomposed by wavelet transforms in the latter part of this work. One operation discussed is document segmentation. Most documents contain graphics and images in addition to text. Therefore, we need to segment the document images to identify text regions. Then OCR techniques can be applied to these isolated regions. Wavelet transform gives a time-frequency localization of the signal and

the calculation of the coefficients from the signal can be carried out efficiently. Some work on segmentation using wavelet methods has been reported in [36, 40, 51, 71]. Textured images can be distinguished from each other by examining their wavelet transforms. Obviously, it can be used for document image segmentation which is an essential part of early document analysis. This work will investigate a new method to detect texture and abnormalities existing in document image by means of wavelet analysis.

1.4 Dissertation Overview

Chapter 2 introduces document compression schemes and surveys related work in the area of compressed-domain image processing. Chapter 3 first states the ITU Group IV coding scheme and then presents the modified Group IV coding scheme. In Chapter 4, some useful techniques are presented for processing the JBIG encoded images directly in the compressed domain or concurrently while they are being decoded. Then basic concepts about wavelet transform are introduced in Chapter 5. Our research work on segmenting document images using wavelet method and its performance are given in Chapter 6. Chapter 7 summarizes the results and contributions and discusses areas worthy of future investigation.

CHAPTER 2

DOCUMENT IMAGE COMPRESSION

This chapter briefly introduces the general field of lossless compression algorithms for document images including facsimile compression and JBIG algorithm. In addition, a brief literature survey on existing compressed-domain algorithms for the above mentioned compression schemes is presented below.

2.1 Introduction

Due to the need of reducing data transmission time and storage space, many efficient coding schemes for bilevel images have been proposed and studied. Among such schemes, the International Telecommunication Union (ITU)¹ fax standard governs how present-day facsimile machines encode and compress images for transmission. In addition, the more recent Joint Bilevel Image Experts Group (JBIG) standard is also designed for bilevel images although it can be used for gray scale images.

It was in the late 1970s that document image compression was first regarded as a special case of compression by ITU. Because of the emerging technologies for document scanning and facsimile transmission, a standard for fax machines was in demand. As a result, the ITU Group III, the first comprehensive standard, was released in 1980 and amended in 1984 and 1988 [12], respectively. Group III machines usually spend one minute to transmit a page of fax over the standard telephone line. Thus this scheme worked better than the earlier Group I and Group II fax

¹ITU are formerly CCITT (the International Telegraph and Telephone Consultative Committee) and ITU-T recommendations are formerly CCITT T Recommendations.

standards, which took about 6 and 3 minutes, respectively. In 1984 the ITU Group IV (G4) standard [13] was published, which is essentially the same as Group III but is designed for error-free digital facsimile transmission mainly on public data networks rather than conventional analog telephone circuits. Group III standard includes two coding schemes: a one-dimensional coding scheme that treats each scan line independently, and a two-dimensional coding scheme that explores correlation of pixels in two successive lines. Group IV standard is a two-dimensional coding scheme, which is similar to that of Group III. In Group IV high compression is achieved largely due to the reduction of two-dimensional redundancy. Both schemes use a run-length-based approach. The simplicity of run-length codes is particularly attractive from the implementation viewpoint.

Although the ITU fax standards provide considerable compression for most bilevel document images, it was found that one can do better by using template models and adaptive arithmetic coding to encode the predictions. This observation led to the design of the JBIG method, which is a standard for progressive encoding of bilevel images. The major benefit of the JBIG [32] is that it achieves higher compression efficiency of around 1.1 to 1.5 times better compression ratio on typical scanned documents compared to G4 fax compression. It also works well on halftone or dithered images which usually frustrate the ITU fax standards. Up to 30 times better compression of scanned images with dithered images compared to G4 fax compression can be obtained.

The JBIG compression algorithm also offers other features as follows:

- Around 2 times better compression on typical 300 dpi documents compared to 'gzip -9' on raw bitmaps.
- Around 3-4 times better compression than GIF on typical 300 dpi documents.
- Much better competitive compression results on computer generated images

which are free of scanning distortions.

The above coding schemes offer reasonable compression. However, due to the development of image processing techniques and increased need for accessing compressed representations, it is most desirable to be able to acquire information directly in the compressed domain. Traditional algorithms for image processing based on pixel values are well established and offer the flexibility to handle different documents. However, these algorithms are computationally expensive since they are performed in the spatial domain. It is reasonable to consider processing image data in the compressed domain. For instance, Mclean [43] extracts edge structure from compressed image representations by evaluating the primary edge and the secondary edge respectively and the techniques provide better performance in comparison to other algorithms in spatial domain.

2.2 Overview of Related Work

Group III is a run-length-based algorithm. The algorithms for run-length-based coding schemes are rich, well-developed, and efficient. Pavlidis proposed [48] a hybrid algorithm for vectorization of documents. By analyzing the Line Adjacency Graph (LAG) and performing polygonal approximation, characters with well-defined strokes are detected. Complex areas are vectorized by sequential thinning on pixels. Extending the result of the hybrid vectorization algorithm, Pavlidis [47, 49] describes a thinning method to operate directly on the run length encoding of the bilevel image, which is used in the feature extractor and the document recognition system. Compress-LAG, a transformation of the LAG is used in the analysis. After the algorithm is applied on text images, results show it is faster by a factor of about three than the hybrid method and faster by at least an order of magnitude than conventional pixel-based thinning algorithms.

In another work, Ronse *et al.* [55] manipulates rundata efficiently by a two-path implementation of a connected component algorithm. The first top-down path identifies equivalent classes and the second bottom-up path merges equivalent classes and assigns a label to each class. Before document analysis is performed, skewed images must be rotated to be deskewed. Shima *et al.* [60] execute rotation of the binary image based on coordinates of data for the start and the end of the run. The rotation method is used to successively perform the skew coordinate transformations in the vertical and horizontal directions. The resulting speed is improved by one order of magnitude compared with pixel-based operations. Also, the author improves propagation-type component labeling through modifying label propagation for block-sorted runs, and experiments ascertain that the processing time is 16.7 times faster than the conventional pixel-based labeling method in [61].

Based on the Hough transform for the detection of document skew and interline spacing, Hinds *et al.* [24] create a gray scale “burst image” from the black run lengths that are perpendicular to the text lines by placing the length of the run in the run’s bottom-most pixel. This data reduction procedure reduces the processing time of the Hough transform by a factor of as much as 7.4 for documents with gray scale images. Wahl *et al.* have developed a newspaper understanding system for run-length coded documents which uses the constrained run length algorithm to partition and label interested areas [73]. The homogeneous rectangles are extracted by using algorithms like Run-Length Smoothing Algorithm (RLSA) or Recursive X-Y Cuts (RXYC). These rectangles are further classified using statistical textual features and feature space decision techniques. To process an optically scanned technical document compressed with techniques based on run-length coding, Nagy proposes a run-length encoded data structure for the hierarchical representation of document (X-Y tree) in [45]. The document is classified into nested rectangles corresponding to meaningful

blocks. Then labels are assigned to the blocks. Aimed at developing a fast pattern matching algorithm directly based on 1-D run data, Amir *et al.* propose an optimal compressed matching method in [2]. Almost optimal compressed matching algorithm for 2-D run-length data is presented in [3].

In the ITU Group IV domain, some pioneering work has been done. Spitz [63] adapts the Baird's alignment technique to look for the pass modes which are heavily populated at the fiducial point positions of data for detecting skew. The method introduced there relies on iteratively testing the suitability of test alignments of reference points whose position is detectable in the compressed domain. He obtains quite accurate results. Later, Spitz [64] defines logos as a spatial arrangement of structures and applies this to index distinctive image characteristics through fiducial points in the compressed domain. Maa [38] observes that bars and space of barcodes are upright and should be perfectly aligned and their Group IV representation is vertical coding. Some work on recognizing barcodes has been presented by finding the vertical mode codeword in the compressed domain. Recently, Hull [26] compares the locations of pass modes contained in two bounding boxes for possible matching while addressing the problem of document image matching.

Generally, image processing in compressed domain is fast. The complexity of algorithms is approximately the order of the number of run length code segments. This speed-up is not surprising because the algorithm operates on large groups of pixels rather than the individual pixels. So far, there have been many new discoveries in the processing of compressed data. On the other hand, developing a new compression scheme to address a variety of image operations in a compressed domain has not been extensively studied. To do this, the features and/or structures useful for performing image operations should be well preserved while designing a new compression schemes. In the next chapter, the coding rules for the ITU Group IV will be modified

to derive a new coding scheme call Modified G4 or MG4, whose flexibility for storage and operations will be evaluated and demonstrated as well.

CHAPTER 3

MODIFIED GROUP IV

To speed up the processing of binary document images, a novel approach to image compression has been developed in this chapter. Similar to the ITU Group IV compression scheme, this new coding scheme exploits the two-dimensional correlation between scan lines and achieves high performance for image operations with low computational complexity. Specifically, based on the coded data, operations such as connected component extraction, skew detection and image rotation are investigated. Feasibilities of performing these operations in the compressed domain are analyzed and demonstrated. Furthermore, the new algorithms are proposed to process binary images and are also tested using a set of test images. Experimental results show these operations run faster in the compressed domain than in the spatial domain.

3.1 Group IV Coding Scheme

Group IV encoding is a run-length-based two-dimensional coding [13]. Each coding line is coded with respect to the previous line, i.e., “reference line”. Since scan lines in a document file are highly correlated, we can take advantage of this characteristic to explore pixel-level structures within several adjacent lines so that the two-dimensional redundancy is reduced.

The coding scheme for Group IV includes three coding modes: pass mode, vertical mode, and horizontal mode. These modes are defined by relationship among the following five changing picture elements as shown in Figure 3.1.

a_0 : transition position on the coding line

a_1 : the next transition position to the right of a_0

a_2 : the next transition position to the right of a_1

b_1 : the first transition position to the right of a_0 and opposite color to a_0

b_2 : the next transition position to the right of b_1

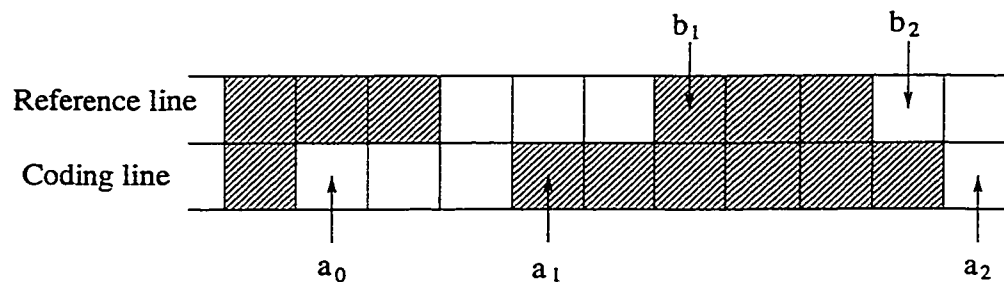


Figure 3.1: The Changing Picture Elements

It easily follows that these points are defined based on transition position and their color can be black or white in a bilevel image depending on the specific environment.

In Figure 3.2, the pass mode is defined when the position of b_2 lies to the left of a_1 . For the next mode to encode, a new a_0 , a'_0 , is set on the element of the coding line just below b_2 . In this mode, a'_0 will keep the same color as the previous a_0 . Although the white pass can be used to determine the fiducial point of a character or a component, it is also true that the white run is decomposed into two parts and many run-length based algorithms could not be directly applied to the Group IV compressed domain.

Figure 3.3 illustrates an instance of the vertical mode, in which the distance $|a_1 b_1|$ is less than or equal to 3. The next position of a_0 is set on a_1 after the vertical mode has occurred. Notice that the color of a_0 has changed after this mode.

If a situation could be coded by using neither pass mode nor vertical mode, then it would be represented by a mode called horizontal mode (see Figure 3.3). When

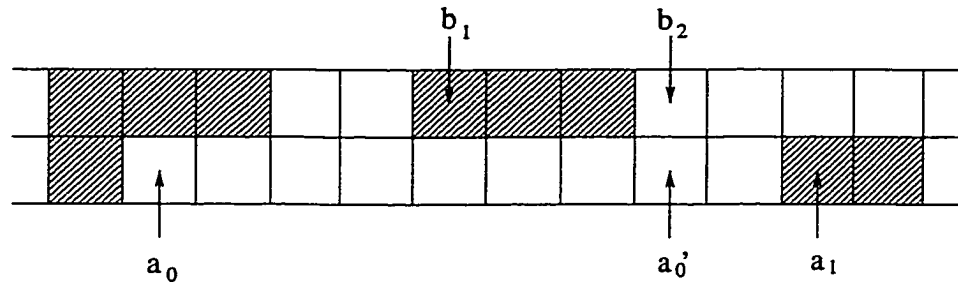


Figure 3.2: G4. Pass Mode

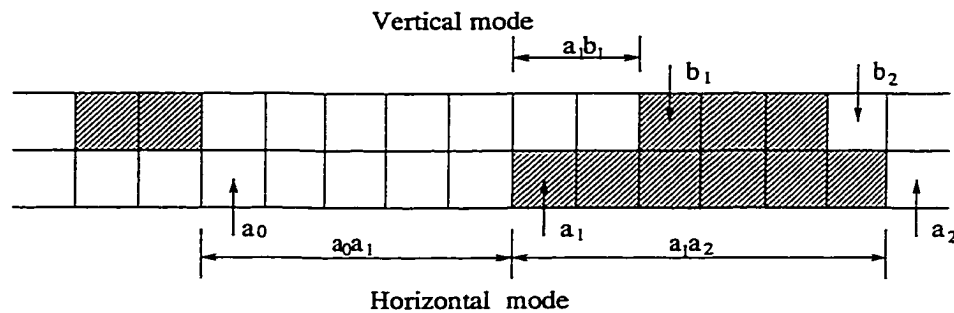


Figure 3.3: G4. Vertical Mode and Horizontal Mode

this mode is identified, both run-lengths a_0a_1 and a_1a_2 are coded using the run-length codewords. After a horizontal mode coding, the next position of a_0 , which has the same color as before, is set on a_2 . This mode is actually a run-length coding.

In summary, the main concern of Group IV is compression, a goal which is achieved rather well. However, some useful features of run length coding become obscured in order to achieve higher compression.

3.2 Modified Group IV (MG4)

Last section shows that an object of interest cannot be easily discriminated from the background in the Group IV domain. It is also difficult to perform other image processing in this domain. To overcome this deficiency, a modified Group IV coding

scheme (MG4) is proposed.

We regard the image as black objects on a white background, as is the case for most applications. This new coding scheme is designed based on the following three considerations:

- To encode the data based on runs for efficient processing
- To keep the original structure in the compressed domain as much as possible
- To reduce two-dimensional redundancy to improve compression

Considering a bilevel document image, three modes, new vertical mode, new horizontal mode, and new pass mode are defined. Compared with Group IV, each mode has either an extended or a different meaning. The scheme is based on the notions of reference run and coding run. Reference run is on the previous scan line, i.e., reference line, while a coding run is the run to be currently encoded, which is on the coding line (Figure 3.4). Let b_1, b_2 denote the coordinates of the endpoints of the reference run and a_1, a_2 denote the coordinates of the end points of the coding run. If the intervals $[b_1, b_2]$ and $[a_1, a_2]$ have overlapping projections, i.e., the following inequalities will hold:

$$a_1 \leq b_2 \quad \text{and} \quad b_1 \leq a_2 \quad (3.1)$$

then, $b_1 b_2$ and $a_1 a_2$ are called the reference run and the coding run respectively. If a coding run has more than one reference run, then for simplicity, the leftmost reference run is chosen to encode the coding run.

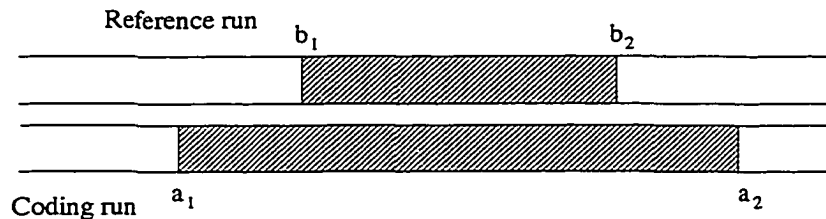


Figure 3.4: MG4. Reference Run and Coding Run

The basic idea is to code a run relative to a unique reference run which has been processed before. Sometimes, several runs on the reference line are skipped in order to locate an appropriate reference run for the coding run. A parameter is associated with the pass mode to record the number of skipped runs on the reference line. It is also possible that a coding run has the same reference run as that of the last coding run. If a reference run is not available, this indicates that two-dimensional correlation could not be utilized and the horizontal mode is employed to perform one-dimensional compression.

3.2.1 New Vertical Mode

The definition of the vertical mode in Group IV is extended here. In Group IV, this mode is defined when the distance of a_1b_1 is not greater than 3 (see Figure 3.3). Hence, the vertical mode include seven cases: $L(1), L(2), L(3), L(0), R(0), R(1), R(2), R(3)$, where, “L” denotes a_1 is to the left of b_1 , “R” denotes a_1 is to the right of b_1 and the number in parenthesis gives the absolute distance between a_1 and b_1 . However, in MG4, if a coding run has a reference run, then the coding run can be represented by a vertical mode. In other words, this mode occurs when the run on the reference line has overlapping pixels with the coding run (satisfy Inequality 1). In this mode, there are two parameters with sign to encode. It can be represented as $V(a_1b_1, b_2a_2)$ (see Figure 3.5).

3.2.2 New Pass mode

The new pass mode can be denoted as $p(n)$, where n is a non-negative integer. When n is 0, the coding run uses the same reference run as the last coding run (see Figure 3.6). When n is nonzero, the case in Figure 3.7 occurs. In this case, n records the number of runs skipped on the reference line, then a vertical mode can

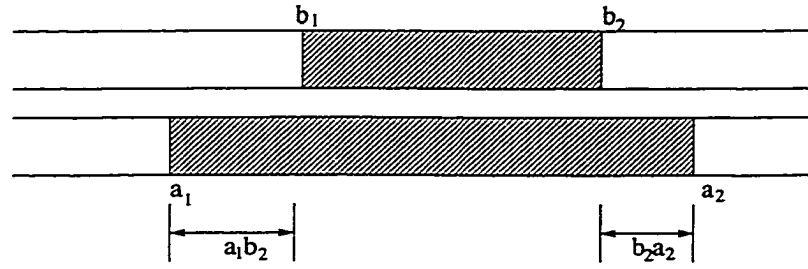


Figure 3.5: MG4. Vertical Mode

be employed to encode the coding run.

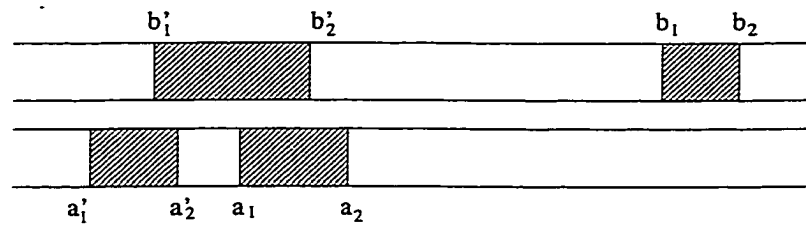


Figure 3.6: MG4. Pass Mode P(0)

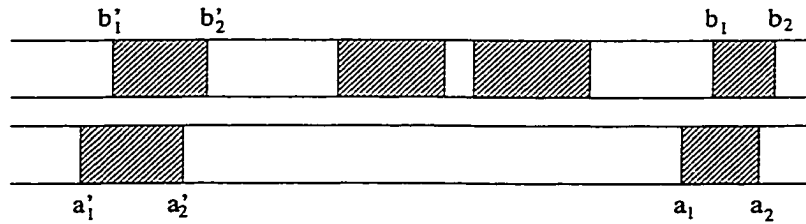


Figure 3.7: MG4. Pass Mode P(2)

By default, image data is compressed and stored in scan order. Suppose the two runs $a'_1a'_2$, a_1a_2 in Figure 3.6 are to be encoded. Since the run $b'_1b'_2$ satisfies Inequality 1, it is the reference run of the coding run $a'_1a'_2$. So the $V(a'_1b'_1, -a'_2b'_2)$ is used to encode $a'_1a'_2$. Then, the next coding run is a_1a_2 . It is seen that $b'_1b'_2$ still satisfies Inequality 1 and is the reference run of a_1a_2 . The modes $P(0)$, $V(-b'_1a_1, b'_2a_2)$ are the

representation of a_1a_2 in the compressed domain. For the same reason, we encode the case of Figure 3.7 as $V(a'_1b'_1, -a'_2b'_2), P(2), V(a_1b_1, -a_2b_2)$. Because of the pass mode, two dimensional correlation can be utilized to maximum extent while compressing a image. So the pass mode implies compression efficiency and structure information. The reason for designing this mode is to keep track of the position of the reference run during image decoding.

3.2.3 New Horizontal Modes

This mode occurs when a reference run cannot be found for the coding run. It represents a situation that the run just above the coding run on the reference line is a white run. Obviously, it implies the beginning of a local characteristic region or a connected component. The white run and black run are coded as $H(a'_2a_1, a_1a_2)$ (see Figure 3.8).

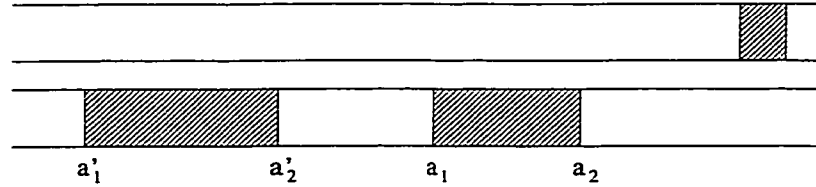


Figure 3.8: MG4. Horizontal Mode

3.2.4 Coding Procedure

We define four positions, b_1, b_2, a_1, a_2 , which represent the absolute coordinates of start and end positions of black runs (probably it is a reference run) on the reference line and coding line, respectively. The flow digram of the coding scheme is shown in Figure 3.9 and the processing steps are described below.

First, in order to compress an image, a reference line and a coding line are defined.

To encode the first line, an imaginary white line is added. Then the coding run a_1b_1 and the reference run b_1b_2 are detected. They are all black runs. If the reference run exists, vertical mode occurs and we continue to find out whether $P(0)$ occurs. This is realized by a loop in the diagram to check whether the next coding run shares the reference run with the previous coding run. If it does, a pass mode with parameter zero (0) is employed to encode this case. This characteristic is useful since it can be concluded that the coding run is connected with the immediately previous run when the pass modes with zero (0) parameter is detected in the compressed domain.

If b_1b_2 does not overlap with a_1a_2 , then the next black runs on the reference line is searched to see if there is any run to be the reference run of a_1a_2 . During the search, n , the number of runs on the reference line passed, is recorded. If there is such a run, a pass mode with nonzero parameter n and vertical mode is added. Otherwise only a horizontal mode is added.

To clarify the procedure, a typical example for text document is shown in Figure 3.10 and encoded according to MG4. We assume there is an imaginary white line before the first scan line and the symbol *EOL* represents the end of the line. Then the MG4 codes for this example are shown in Figure 3.10. During encoding, highly probable modes will get shorter codes while less probable modes will get longer codes. Notice that the vertical mode occurs frequently. So, the shortest code will be assigned to it and MG4 thus gains compression efficiency.

The MG4 code is based on the modified Huffman scheme. The code for each mode has two parts: a flag indicating the mode, and a value or two values representing the lengths of the black runs or the white runs. These values are taken from the white-run and black-run code tables of one dimensional coding.

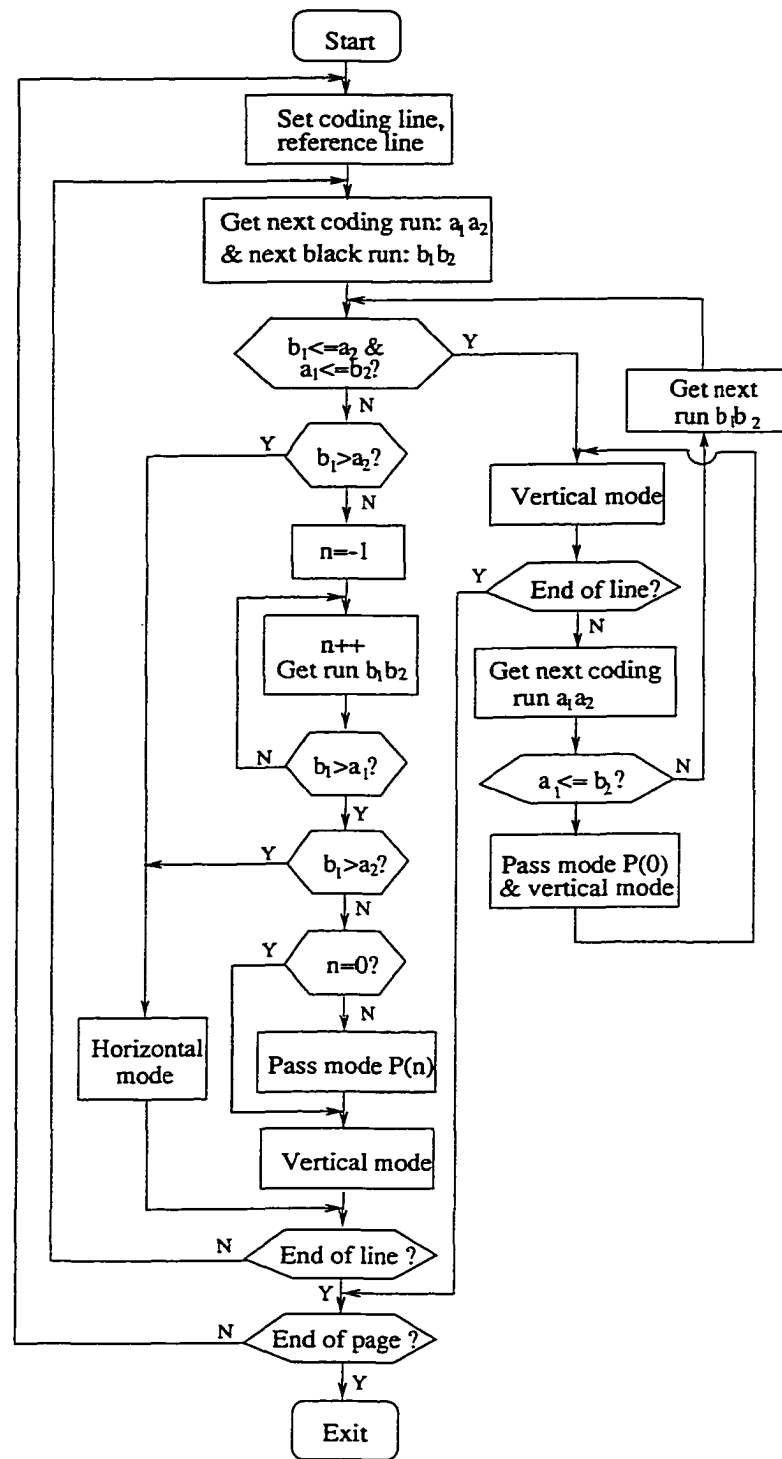
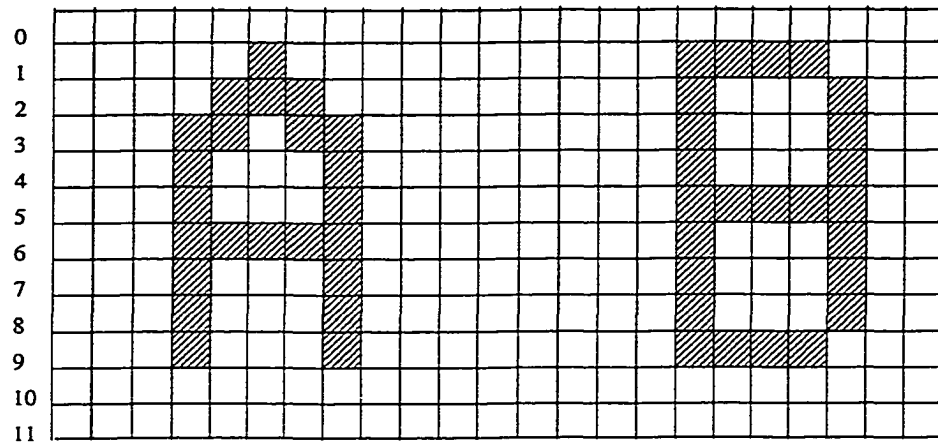


Figure 3.9: MG4 Coding Scheme Flow Diagram



Line:

```

0  EOL
1  H(5,1), H(10, 4), EOL
2  V(1,1), V(0, -3), P(0), V(-4,1), EOL
3  V(1, -2), P(0), V(-2,1), V(0,0), V(0,0), EOL
4  V(0,-1), V(-1, 0), V(0,0), V(0,0), EOL
5  V(0, 0), V(0,0), V(0, 4), EOL
6  V(0,4), P(1), V(0, -4), P(0), V(-4, 0), EOL
7  V(0, -4), P(0), V(-4, 0), V(0,0), V(0,0), EOL
8  V(0,0), V(0,0), V(0,0), V(0,0), EOL
9  V(0,0), V(0,0), V(0,3), EOL
10 EOL
11 EOL

```

Figure 3.10: An Example of Characters “A B ” and their MG4 Encoding

3.3 Experimental Results for MG4 Compression

Since one of the concerns about MG4 is its compression performance, the experiment was conducted to a set of 100 fax document images using this compression scheme. This set of images was divided equally into two categories: text (txt) only and mixture of text, graphics, table, etc (mix). The new coding scheme was implemented in a UNIX environment on an O2 workstation (all the following experiments are conducted in the same environment unless specified). Based on this classification, some randomly picked experimental results are presented in Table 3.6, which suggest

that the category of mixture generally gains higher compression ratio than the text only, mainly due to the large white area contained in images of graphics, table, etc. The mean compression ratio of MG4 for our test data set is around 15, which is about 12% lower than that of Group IV, i.e., 17.

In another experiment, we chose ITU1-ITU8, the *de facto* standard test images of size 513229 bytes. Using MG4, the eight images were compressed. Then they were decompressed and the original images were restored losslessly. It is found that MG4 achieves high compression which is close to Group IV.

Table 3.1: Compression Performance of MG4

Item	Txt 1	Txt 2	Txt 3	Txt 4	Mix 1	Mix 2	Mix 3	Mix 4	Ave.
MG4	16.2	10.4	9.4	13.7	20.7	16.8	13.7	18.1	15

Table 3.2 shows the experimental results. The unit of the file size is a byte. On

Table 3.2: Experimental Results Using the MG4 Coding Scheme

Test Image	Size after MG4 comp.	CF (MG4)	CF (G4)	% Increase in storage
ITU 1	25481	20.1	24.7	22.9
ITU 2	15891	32.3	38.6	19.5
ITU 3	37709	13.6	16.4	20.6
ITU 4	82849	6.2	6.8	9.7
ITU 5	40391	12.7	14.4	13.4
ITU 6	24107	21.3	26.0	22.1
ITU 7	80679	6.4	6.6	3.1
ITU 8	23879	21.5	23.3	8.4

average, the compression ratio is about 17:1. Compared to Group IV, MG4 has an increase in storage of around 10-20%. The comparison of storage requirement is in favor of Group IV; nevertheless, this comparison is not fair as the MG4 code has not been optimized for the standard eight test images used to evaluate the Group IV

code. We believe that this optimization will further improve the compression ratio rendered by MG4. Meanwhile, the reason for this increase is that MG4 keeps more structural information, which is very useful for image manipulation in the compressed domain, than Group IV does.

3.4 Mode Analysis

The three modes of MG4 are designed based on Group IV. Each mode has its own feature preserving ability. Figure 3.11 presents some general encoding situations for 26 English characters, where \downarrow and \rightarrow denote the locations of the horizontal mode and the vertical mode, respectively. Notice that the numbers of horizontal and pass modes listed in the figure are based on single characters. Therefore, some extra pass modes may be generated between characters and are skipped for only demonstration purpose. In Group IV, there is no distinction between objects (black runs) and background (white runs) during encoding, i.e., they are treated equally. However, MG4 focuses on objects only and encodes them in a way different from background so that the manipulation of these objects would be convenient and efficient when needed. On average, about two horizontal modes per character will be generated in the coded image. Vertical modes, which are not listed in the figure, are actually default modes because of their high frequency of occurrence over the other two modes. By inspection, for each connected component, at least one horizontal mode that will occur represents the critical point for text component. This critical point is useful for document analysis because the placements of horizontal modes form a base line corresponding to the text lines of document images.

In image compression, compression ratio is highly data dependent. Therefore, in the following discussion some statistics gained from ITU1-ITU8, the standard test images, are presented for the ease of comparison. The frequencies of vertical modes,

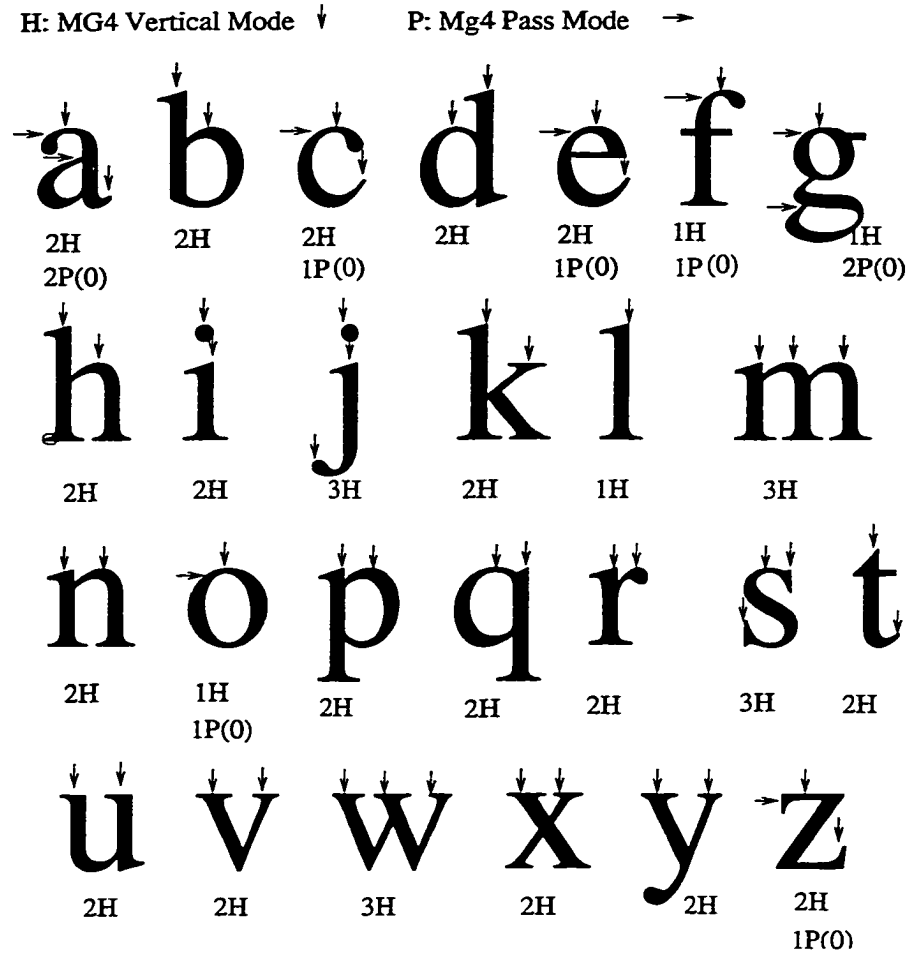


Figure 3.11: Mode Analysis for Characters

horizontal modes and pass modes in the images are schematically shown in Figure 3.12, where the highest frequency (83081) is the vertical mode in ITU 4 and the lowest (453) is the horizontal mode in ITU 2. On the average, for the eight test images, the occurrence probabilities of the three modes of MG4 are 83.7% (vertical), 5.6% (horizontal) and 10.7% (pass), respectively. As described before, the vertical mode ($V(a, b)$) is constructed based on the special characteristics of texts, i.e., information is highly correlated within adjacent scan lines. The two values a and b , are actually the differences of the coding run and the reference run; in other words, the edge information is kept in a coded form. This is another property of MG4 that will

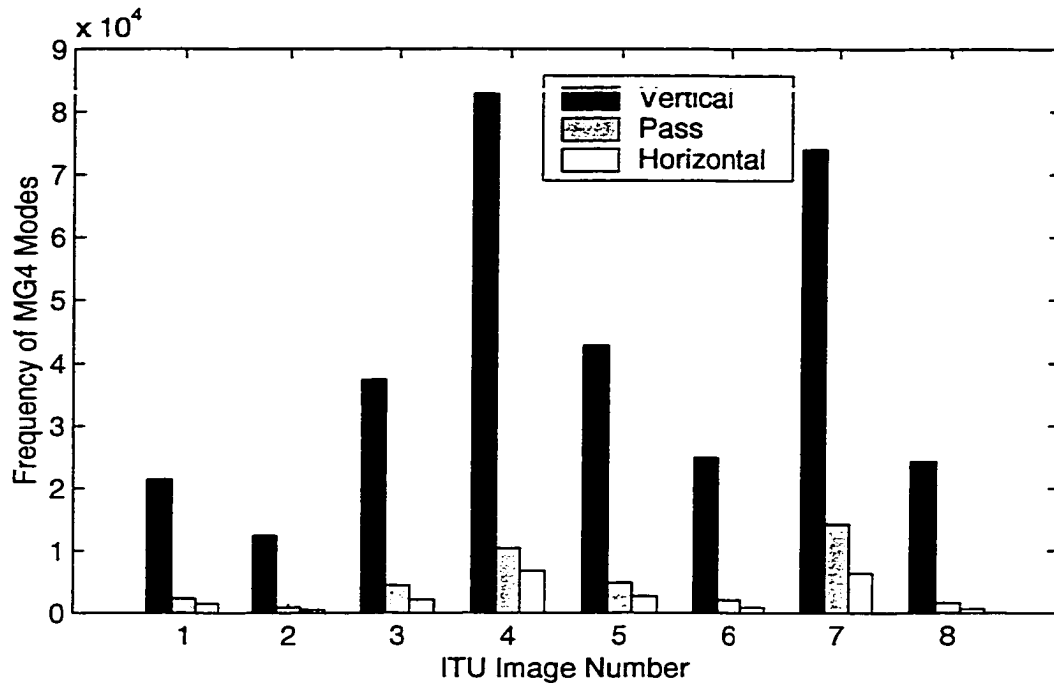


Figure 3.12: Frequency Analysis of the three Modes of MG4

be useful for document analysis. Table 3.3 shows that for the vertical modes $V(a, b)$, the cases of $a \leq 2$ and $b \leq 2$ account for about 90% of the total occurrences of vertical modes in the image (N). Thus, short codes can be assigned to these modes with high probability and compression will be achieved in general. Notice that there is a

Table 3.3: Frequency of Vertical Modes $V(a, b)$ When $|a| \leq 2$ and $|b| \leq 2$

Image	ITU1	ITU2	ITU3	ITU4	ITU5	ITU6	ITU7	ITU8
$V(a,b)/N$	93.6%	91.6%	91.5%	90.1%	92.5%	92.1%	85.1%	91.3%

parameter n associated with pass modes, which records the number of runs skipped on the reference line. Figure 3.13 shows that the frequency of $P(n)$ is dramatically decreasing as n increases. This implies some physical structures existing in most document images, (i.e., one or two runs), will often be skipped and other situations

($n > 2$) actually rarely happen.

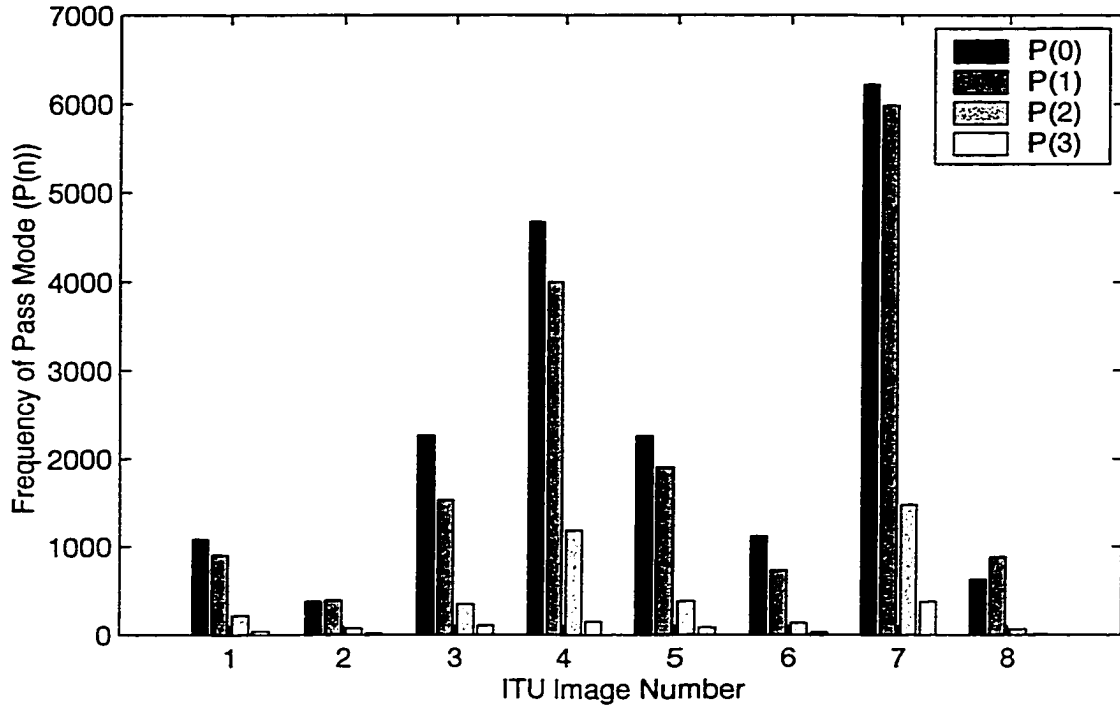


Figure 3.13: Frequency Analysis for Pass Mode P(n)

In summary, the above discussion suggests that MG4 is a structured coding scheme for binary document images, which can be characterized as being object-oriented, differential and run-length-based.

3.5 A General Procedure for MG4 Compressed-Domain Processing

A general procedure for MG4 compressed-domain processing is developed in this section. For any coding scheme, there are two stages for compressing an input data I : transformation (T) and encoding (E). It can be denoted as

$$T(I) = O'$$

$$E(O') = O$$

where, I is the input data and O is the final output, namely, compressed data. In general, direct processing of an encoded data O is not possible. For example, for image processing based on run data, the actual runs are available for processing only after symbol decoding. In practice, the transformed data O' retains most characteristics of the input data. Therefore, the operation can be applied to the transformed data O' , with the time complexity of $f1(m)$, where m is the number of data to be processed, i.e.,

$$P(O') = Y'$$

In contrast, suppose the operation applied to decompressed data possesses a time complexity of $f2(l)$, where l is the number of data to be processed in the compressed domain. The operation can be expressed as:

$$P(T^{-1}(O')) = M$$

$$T(M) = Y'$$

In general, if the relationship $f1(m) < f2(l)$ is always true, then it means some time has been saved because of the direct processing of compressed data. Even though the algorithms for direct processing are highly data dependent, in most cases the data complexity of m is much less than the data complexity of l , which consequently makes the algorithm more efficient.

By exploiting the mode features in MG4, some techniques are developed below to address the problems of image processing in general. The basic procedure for a certain MG4-code-based operation consists of three stages:

1. Symbol Decoding

In this step, the compressed image is loaded into memory and symbol decoding is implemented in order to get sensible information stored in the compressed data. At this stage, the image is represented by a set of MG4 modes which correspond to the critical points of the original images.

2. Feature Extraction

Basically, this step is based on mode analysis. Since the object components in an image are usually the focus of image processing, the horizontal mode, which denotes the initialization of a new component, will be detected and referred to as Information 1 ($I1$). The vertical mode is useful for both edge processing and run-length based operations, and is named Information 2 ($I2$). The pass mode, preserving the information of physical structure of components and inter-connections between components, is extracted and referred to as Information 3 ($I3$). For some operations like skew detection, this step is conducted to isolate points used in determining the base lines of text in an image. This procedure will reduce the amount of data that must be processed in the following stages.

3. Feature Processing and Optimization

Once the features $I1$, $I2$ and $I3$ have been extracted, they can be manipulated or processed in a fashion determined by the specific application. For skew detection, $I1$ has been utilized and projection is performed to search for the skew estimate by finding a maximum accumulator. In rotation, $I2$ is the major input information for manipulating run length data in horizontal direction and vertical direction, respectively. The optimization is necessary for some operations such as connected component extraction, where the components found to be connected will be merged into a single connected component at a later time.

Assuming an input of MG4 image I is given and the operation applied to I is denoted as P , then algorithm for processing an image I in the MG4 domain can be described as follows:

MG4 – Operation(I, P, Y)

I = MG4 source image, Y = processed image

for each Mode(R_{cod}, R_{ref}) $\in I$

```

if Mode is a horizontal mode then
    Obtain  $I1(R_{cod}, R_{ref})$ 
elseif Mode is a vertical mode then
    Obtain  $I2(R_{cod}, R_{ref})$ 
else
    Obtain  $I3(R_{cod}, R_{ref})$ 
endif
 $Y = P(I1, I2, I3)$ 
endfor

```

The time complexity of this algorithm is $O(N)$, where N is the number of modes for the MG4 source image. Since the pass mode contains physical structure information, which will be merged into the other two modes, the time complexity can be estimated as $O(N_v + N_h)$, where N_v, N_h are the frequencies of vertical modes and horizontal modes, respectively. By observation, in vertical modes, the coding run is coded with respect to the reference run for text images; N_v, N_h , and the number of black runs in the source image, N_{brun} , would satisfy the equation: $N_{brun} = N_v + N_h$. Therefore, $O(N_{brun})$ could be a good estimation of the time complexity of MG4 operations. Since the standard document has a size of 1628×2376 pixels, the total number of pixels is 4,105,728. Experimental results show that the number of black runs is 42,639.6 in the mean for the ITU test images and thus the ratio of the number of black run to the number of pixel is 96.29, showing that the number of runs is less than the number of pixels by about two order of magnitude.

3.6 Applications

To perform data compression, the structural information existing in the data needs to be observed, identified and utilized. If, in the compressed domain, the

characteristics reflecting the structural information of the original images could be detected, extracted and further used, then the processing algorithms may be directly organized base on coded data and thus image operations would be facilitated. In the following subsections, we will discuss several major image operations in the MG4 domain: connected component extraction, skew detection and deskew.

3.6.1 Connected Component Extraction Analysis

An important part of image analysis is to decompose the image into entities (objects, regions, and blobs). The general aims for an appropriate description of connected components are to obtain an exact description making further processing possible, extracting features like size, location, and shape of the region so it can be compared with the corresponding part of the model world. The other objectives are to improve processing speed and reduce memory usage. Specifically, features effective for a subsequent analysis are:

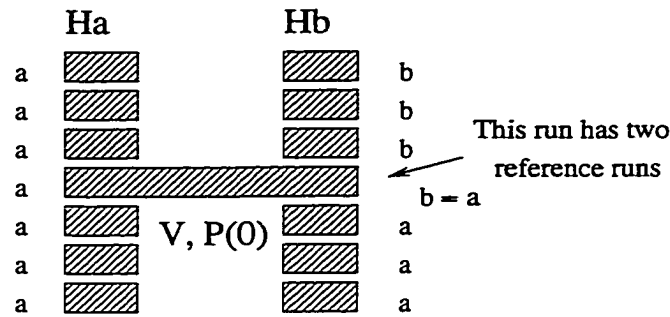
- area
- perimeter
- coordinates of the surrounding rectangle
- coordinates of the center of gravity
- measure of roundness
- minimal and maximal distance between the center of gravity and the contour
- different kind of ratios (e.g. area/area of surrounding rectangle)

Connected components are extracted and used in a number of ways, for example, calculating the number of objects, object matching and analysis of components. In traditional methods, the extraction of components is often done by using a 2×2 or 3×3 pixel window to scan the image as a means of contour detecting [7], making

the procedure time-consuming. MG4 is actually a differential coding scheme which keeps the information about object contour. Because of this characteristic, it is possible to organize useful information in the compression domain to extract connected components.

In MG4, the horizontal mode often indicates the start point of a component. So for every horizontal mode that is encountered, a database is built. The vertical mode is the most frequently occurred mode. By definition, the coding run and its reference run belong to the same connected component and thus they should belong to the same database. Sometimes there are several horizontal modes for a single connected component. Hence, databases representing the same connected component need to be merged through detection of a run having two or more reference runs. If a pass mode with parameter zero (0) occurs, the next run coded by the vertical mode is also connected with the component. Making use of an example, Figure 3.14 illustrates how to utilize characteristics existing in the compressed domain to isolate information about connected components. In Figure 3.15, a flow chart for connected component extraction is presented. The procedure to extract connected components can be generalized in the following steps:

1. The information about two adjacent scan lines is always kept and renewed.
2. The horizontal code is searched in the compressed domain. If a horizontal code, whose black run represents the beginning of a component, is found, then a database is generated for the new component.
3. According to the particular property of each mode, parts of the component on subsequent lines are extracted and saved into corresponding databases.
4. Databases, which belong to the same component, are merged to form a single database. This database will contain complete information about a connected



H: Horizontal mode V: Vertical mode P: Pass mode

A component database is built for every horizontal mode encountered.

Ha -> database "a" Hb -> database "b"

If a vertical mode is encountered, the current coding run and its reference run belong to the same database.

If a run has two or more reference runs, then the databases including these reference runs are merged into one database (i.e., $b = a$).

If P(0) is encountered, a subsequent run represented by a vertical mode is connected with this component.

Figure 3.14: MG4. Building Database for Connected Component Extraction

component.

Applying the proposed algorithm to this operation, a representation for the exact description of connected components can be obtained. This representation may be used for a subsequent analysis such as determining area, perimeter, measures of roundness, etc. In the experiments, the complexity of algorithms is approximately in the order of the number of run length code segments. As a result, the processing is much faster than that of spatial domain. To give a more concrete speed estimation, the measured time for the MG4 algorithm and the Bartneck algorithm were recorded and plotted in Figure 3.16. By processing the MG4 compressed data, the algorithm is advantageous in saving processing time, i.e., on average, 1/6 of the original time for those eight test images is needed.

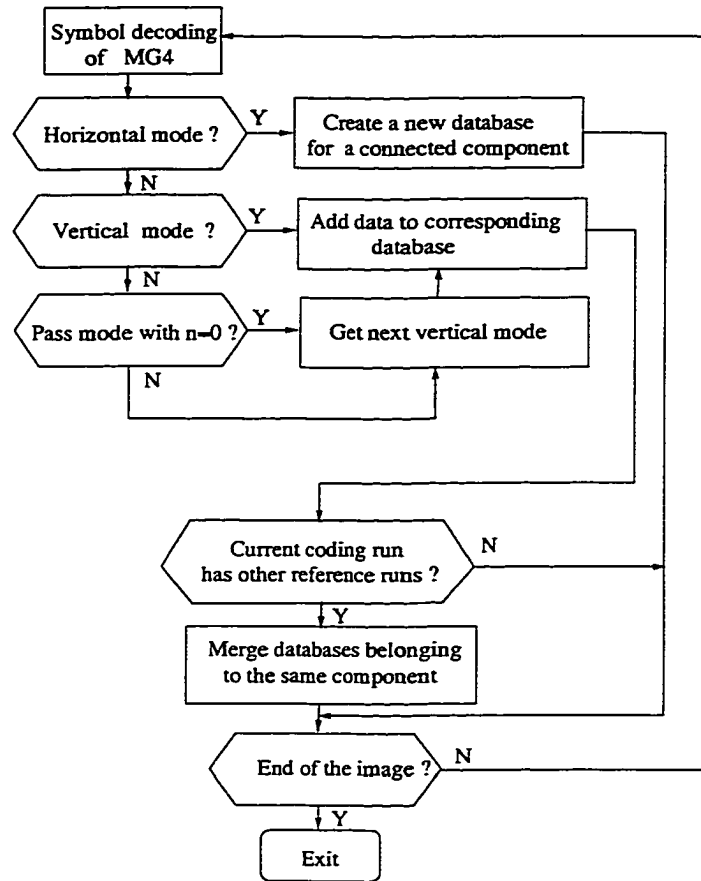


Figure 3.15: MG4. Flow Chart of Connected Component Extraction

3.6.2 Skew Detection

Obtained images are subject to be tilted when the user scans some documents. However, for aesthetic reasons, usually people expect the image to view on the screen with no skew. Skewed images are not easy to compress and result in increased image size after compression. Moreover, recognition accuracy can be improved, layout analysis can be simplified, and baseline determination at OCR stage can be improved by just correcting skewed images in the preprocessing phase [8].

Skew detection is not a new topic. The techniques appearing in literature include text row accumulation [62], recursive morphological transformation [16], local region complexity [77], Fourier transformation [23], Hough transformation [24], the most

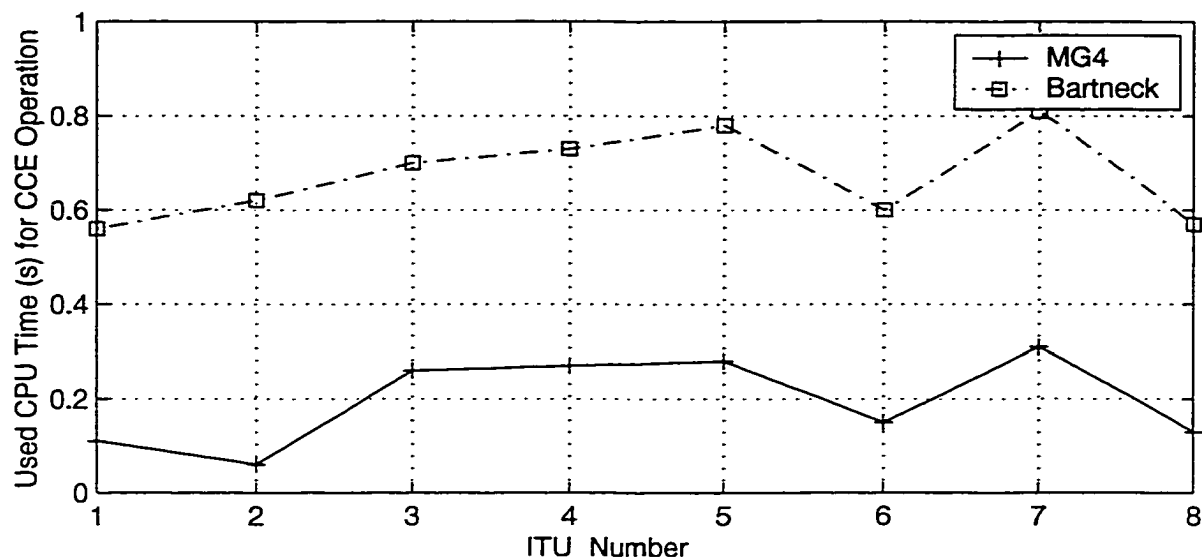


Figure 3.16: Time Comparison for Connected Component Extraction Between MG4 Algorithm and Bartneck Algorithm

common approach to skew detection, and the projection profile technique [4].

Baird's technique uses the basis of aligning fiducial points associated with selected connected components in an image. He chose the bottom center of the connected component bounding box for the fiducial point. Then fiducial points are assigned to each connected component. The alignment becomes efficient by processing the relatively small number of points as representative of the entire image and calculating the alignment on the basis of the sum of square of the counts of the fiducial points that appear in a series of rotationally aligned bins. However, the disadvantage is that the work of locating connected components must be done first.

Different from Hough transformation approach, the projection profile uses a linear projection into a partitioned accumulator array instead of a sinusoidal projection. The basic idea for this algorithm is to project the set of representative points along parallel lines oriented at some search angle. So an important step for this algorithm is how to determine these representative points, which leads several projection profile based

algorithms appearing in the literature.

Framework for Projection Profile Based Algorithms

Usually, there are three stages for this algorithm. First, a set of representative points is extracted from the source image. The purpose of this step is to isolate the points that correspond to the baseline of the text of the image. In addition, this also reduces the amount of data that must be projected in the next two steps. The significant difference between the compressed domain algorithms and the spatial domain algorithm is the complexity of finding those representative points. Most compression algorithms act like some feature extractor while compressing the images. As mentioned before, Spitz finds the skew angle by detecting the occurrence of pass modes in the Group IV domain. He locates the base point of every character. It turns out that in the MG4 domain, there is the same convenience to detect the skew angle with high accuracy as in the Group IV domain.

Then, the representative points from the above step are projected along parallel lines into an accumulator array, which is partitioned into fixed height bins. The selection of bin size is determined so as to maximize the chance of representative points that belong to the same text line being projected into the same bin.

A *projection profile estimator* (PPE) is defined as a pair, (F, ϕ) , where F is representative reduction of a source image to some subset of points, and ϕ is an alignment premium whose function is to take an accumulator array and return a measure of alignment for the projection at that angle. Then, ϕ is optimized over the range of search angles specified by the interval $[\theta_{min}, \theta_{max}]$. Therefore, F will reduce the original image into a set of triples, (x, y, w) , which represents the position and the weight of a representative point. A skew estimation can be conducted by the following algorithm if a PPE, (F, ϕ) , is given.

$$PPE - ESTIMATE(I, w, h, (F, \phi))$$

$I = \text{source image}, w = \text{image width}, h = \text{image height}, (F, \phi) = a \text{ PPE}$
 for $(x, y, w) \in F(I)$
 for $\theta \in [\theta_{min}, \theta_{max}]$
 $\rho = y + x * \tan \theta$
 $A[\theta, \rho / BINSIZE] += W$
 endfor
 endfor
 $PPE - Est = \theta \text{ such that } \phi(\theta) = \max_{\theta} \phi(A[\theta])$

Horizontal Mode Code Detection

The basic idea is that the horizontal mode is often distributed at the top point of every character. Figure 3.17 illustrates the distribution of the horizontal mode in a document image. Thus, many of these points are on or near the x-height line of a text-line. The skew angle is determined according to the distribution of horizontal runs. Combining the alignment-measure approach with locating the maximum techniques [4], the skew angle is detected in the MG4 domain. Experiments show that the accuracy and the speed for estimating skew angle is comparable with that in the Group IV domain.



Figure 3.17: Detecting the Key Points for the Skew Angle Represented by the Horizontal Mode

ITU 1 was used as a test image here and only horizontal modes are detected. The

distribution map of horizontal modes is presented in Figure 3.19.

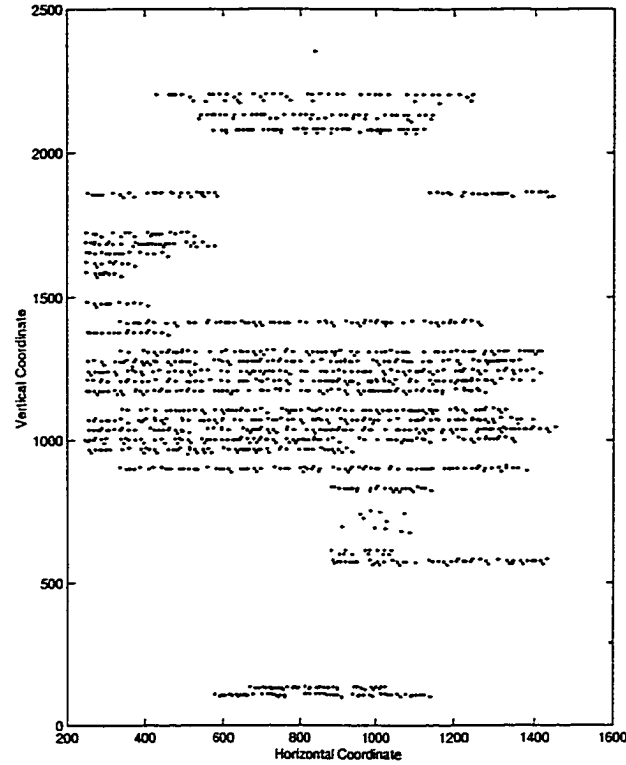


Figure 3.18: Distribution of Horizontal Modes in ITU 1

The PPE, (F_h, ϕ_h) , is described as:

$$F_h(I) = \{(x, y, 1) | (x, y) \text{ is a horizontal mode code in the MG4 source image} \}$$

and

$$\phi_h(A[\theta]) = \sum_{\rho=0}^{\text{height}} A[\theta, \rho]^2$$

A test set consisting of 100 pages is randomly selected from document image databases. Using the test set, a number of experiments were conducted to evaluate the performance of our algorithm. Table 3.4 compares the results of applying the Baird techniques [4], the white pass alignment technique [64] and the MG4 algorithm to the eight images from the ITU test set. The bitmaps were digitally rotated by -3.0° and

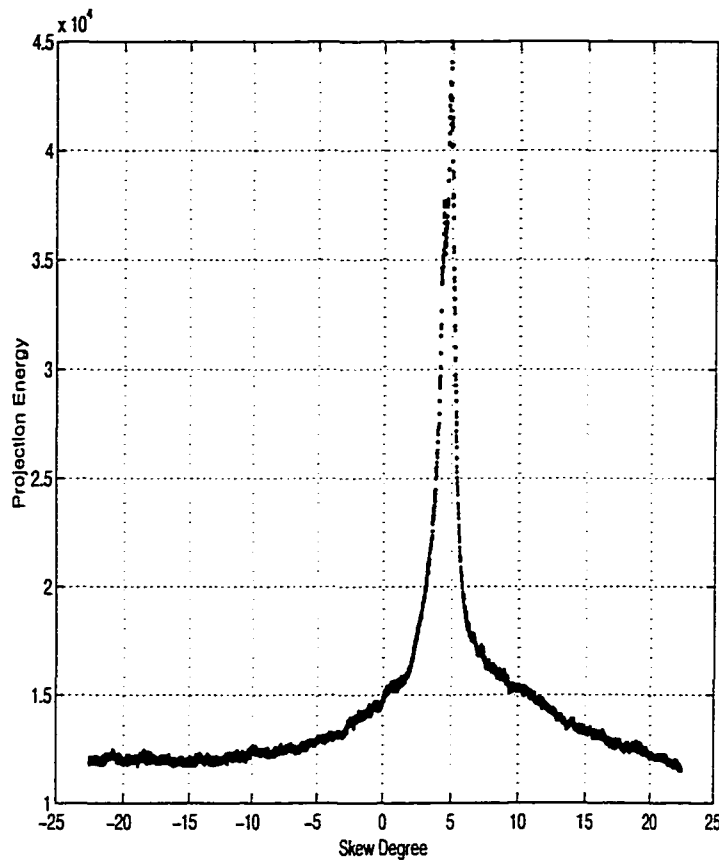


Figure 3.19: Energy Projection for ITU 5 with a Skew Angle of 5°

5.0° . In each instance three skew angle detection algorithms were applied to these images to determine the amount of skew in the supposedly unskewed image. The implementation of the MG4 algorithm is fast because only horizontal modes (about 10% of the total number of the MG4 modes) are extracted for processing. Clearly, the content of the document has a profound effect on the detection results. The table shows that the skew estimate, using horizontal modes of MG4 as representative points of text lines, is comparable to that of Group IV, and for some cases, even lower estimation errors are obtained. For every algorithm, Root Mean Square (RMS), a measure of estimation accuracy, is calculated over skew estimates of rotated ITU-ITU8 for the cases of 0° , -3° and 5° skew, respectively. The table shows that the lowest

Table 3.4: Skew Detection Performance of MG4

Algorithm	ITU1	ITU2	ITU3	ITU4	ITU5	ITU6	ITU7	ITU8	RMS
Baird (0°)	0.20	-1.00	-0.32	-0.04	0.52	0.08	0.16	0.08	1.21
Pass (0°)	0.24	0.10	-0.32	-0.08	0.48	0.24	0.16	-0.20	0.73
MG4 (0°)	-0.12	-0.02	-0.34	-0.13	0.15	0.03	0.02	0.31	0.52
Baird (-3°)	-2.76	-4.00	-3.16	-3.00	-2.48	-2.88	-2.84	-3.20	1.20
Pass (-3°)	-2.80	-2.12	-3.24	-3.00	-2.52	-2.88	-2.88	-2.96	1.06
MG4 (-3°)	-2.90	-2.98	-3.25	-3.08	-2.95	-3.00	-3.01	-2.89	0.31
Baird (+5°)	5.20	6.00	4.88	4.92	5.52	5.08	5.08	4.92	1.16
Pass (+5°)	5.24	5.02	4.76	5.00	5.48	5.12	5.12	5.12	0.62
MG4 (+5°)	5.13	5.09	5.24	5.14	4.87	4.96	5.09	5.21	0.42

RMS is achieved by the MG4 algorithm based on the listed experimental results.

3.6.3 Skew Correction

Skew correction, using individual pixels, requires considerable computational effort. Shima *et al.* [60] rotate the binary image based on run data. Here the image rotation was implemented in the MG4 domain directly. The algorithm to correct skew is by 1-dimensional shearing and expansion or contraction along the horizontal direction and vertical direction, respectively. The two-pass algorithm is based on the following decomposition of rotation matrix:

$$\begin{aligned}
 R(\theta) &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 \\ -\tan\theta & \frac{1}{\cos\theta} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ 0 & 1 \end{bmatrix}
 \end{aligned}$$

The Proposed Algorithm

Figure 3.20 shows the major steps for rotating bilevel images compressed by MG4. Suppose (x, y) is the start point or the end point of run data in a skewed image, and (x', y') is the corresponding point after rotation of the image. First, it is easy to get the

run data after the symbol decoding of MG4. Transformation in horizontal direction is performed according to the formula: $x' = x * \cos\theta + y * \sin\theta$, where θ is the skew angle. Then, the horizontal run is converted into vertical run. The start points and end points of the vertical runs can be detected by examining the occurred modes, for instance, the horizontal mode indicates the beginning of the vertical run whereas pass mode indicates the end of the vertical run. In this algorithm, the transformation from the horizontal run into the vertical run is efficient because its computation complexity is based on the number of horizontal runs. Next, using $y' = y * \sec\theta - x * \tan\theta$, transformation in vertical direction is carried out. The time cost of this step is also proportional to the number of vertical runs. The above steps are repeated until all the symbols in the image are processed. At this time, the information of the rotated image is stored in the form of vertical runs. Finally, vertical runs are converted into horizontal runs in the same way as converting horizontal runs into vertical runs and then the image is saved in the MG4 form.

To illustrate the above steps, the original image, i.e., ITU 7, shown in Figure 3.21(a) is employed to present intermediate results. Suppose (x, y) is the beginning point or ending point of run data in a skewed image, and (x', y') is the corresponding point after rotation of the image. Then, the steps for rotation can be described as follows:

1. Transformation in horizontal direction by:

$$x' = x * \cos\theta + x_0 \quad \text{where} \quad x_0 = y * \sin\theta$$

The transformation result of this step is shown in Figure 3.21(b), where the x - coordinates of the black pixels are adjusted depending on the rotated angle.

2. Conversion of horizontal run into vertical run

This step makes the vertical transformation possible and Figure 3.21(c) gives the output at the current stage.

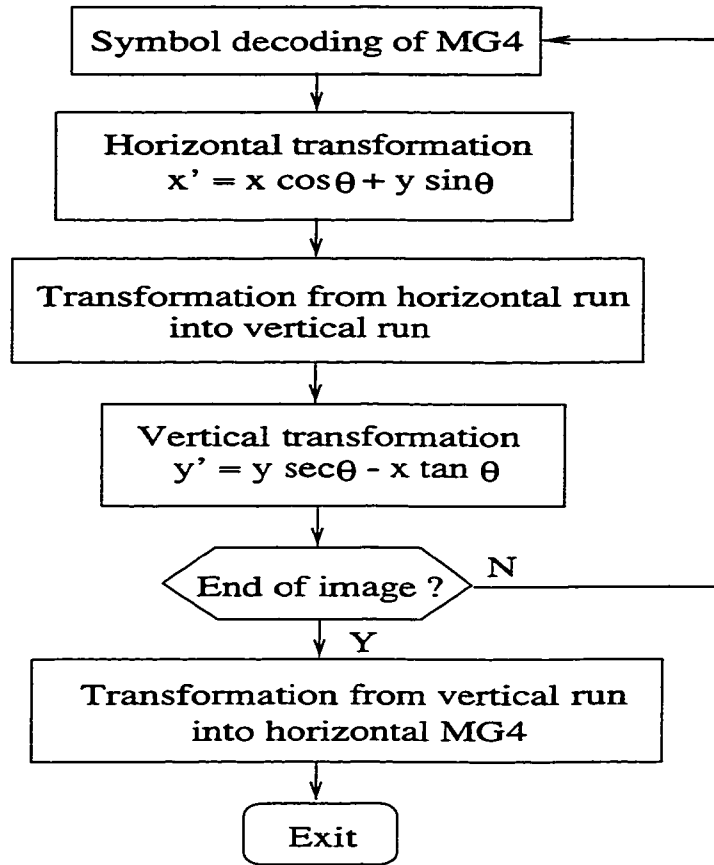


Figure 3.20: Rotating an Image in the MG4 Domain

3. Transformation in vertical direction using:

$$y' = y * \sec\theta + y_0 \quad \text{where} \quad y_0 = -x * \tan\theta$$

Figure 3.21(d) presents the output images after transformation in vertical direction.

4. Transformation of the vertical run into the horizontal MG4

Converting horizontal runs into vertical runs

As discussed in Chapter 2, Shima *et al.* rotate binary images by manipulating their run data. However, in the paper, the conversion of horizontal runs into vertical runs, an important step of rotation, is implemented by examining every black pixel in the images [60]. The operation of converting horizontal runs into vertical runs on compressed data is an open problem worthy of further research. After studying the procedure for the conversion in the existing algorithms, in this work a new method is proposed to facilitate the step of obtaining vertical runs for rotation without inspecting each black pixels in the images.

By observation, the coding situations, i.e., how the coding run and the reference run are possibly positioned with respect to each other, could be generalized into six cases, as shown in Figure 3.22. In part A of Figure 3.22, the three cases are used for detecting the beginning of the vertical runs. This happens where the start position of the coding run is located to the left of the reference run. The black thick line stands for the start points of the vertical runs. In part B of Figure 3.22, the start position of the coding run is located to the right of the reference run and the end points of the vertical run are also highlighted by the black thick lines.

In step 2, while converting horizontal runs into vertical runs, the start and the end points of the vertical runs can be determined by examining the MG4 modes, i.e., the horizontal mode indicates the beginning of a vertical run whereas the pass mode indicates the ending of a vertical run.

X' in Figure 3.22 is the reseted start point for the next comparison. Obviously, for the start point of each vertical run, there always exists a corresponding end point, which can be extracted on the basis of a bunch of pixels (the number of pixels is defined by the length of the run). Without examining each black pixel in an image

during the processing, the proposed algorithm can efficiently conduct the operation of converting horizontal runs into vertical runs and gain speed advantage. Similarly, this method can also be used to convert vertical runs into horizontal runs.

Speed Analysis

In practice, the first three steps could be combined into one step: First the horizontal coordinate transform is implemented to calculate new coordinates; then judgment is made to see if the horizontal run or part of it is the start point or end point of a vertical run. If it is, the coordinate transformation along the vertical direction is performed and obtained result about the vertical run is saved. Then we can proceed to the next run and process it in the same way. Figure 3.23 shows the performance of our rotation algorithm. It is hard to distinguish the image rotated by the algorithm with the image rotated pixel-by-pixel. Since this algorithm is based on run and in ordinary documents the number of runs is less by about two orders of magnitude than the number of pixels, the computation complexity is significantly reduced.

Table 3.5 shows the measured time required for rotation. The steps for rotation operation can be generalized as horizontal transforming, converting horizontal runs into vertical runs, vertical transforming and converting vertical runs into horizontal MG4. The rotation angle is set as 4.5° in the experiment. For ITU 1, the time for symbol decoding and the first three steps of rotation listed above is 0.12 s (CPU seconds) while the time for the step 4 is 0.11 s. Consequently, 0.23 s are needed to rotate the MG4 encoded ITU 1. In another experiment, the rotation is performed based on the pixel data according to rotation formula and then the used time is recorded. The coordinate calculation is required for each pixel for this pixel-based method. Altogether 3.63 seconds are needed for an image of 1727×2287 to be rotated. In contrast, in the rotation based on MG4 code, the mean time for the eight ITU test images is 0.46 seconds. Consequently, the processing speed is improved by a factor

of 7.89 compared with the rotation, pixel by pixel, for the test images.

Table 3.5: Measured Time Required for Rotation Based on MG4 (CPU seconds)

Item	ITU1	ITU2	ITU3	ITU4	ITU5	ITU6	ITU7	ITU8	Ave.
Syb. Decod.+ First 3 Steps	0.12	0.07	0.18	0.47	0.64	0.12	0.44	0.12	0.22
Step 4	0.11	0.06	0.16	0.43	0.43	0.12	0.54	0.10	0.24
Total	0.23	0.13	0.34	0.90	0.22	0.24	0.98	0.22	0.46

In the above, three important document operations, CCE, skew detection and rotation, have been selected and implemented by applying the proposed general algorithm to the test images. Experimental statistics show that the processing speed in the MG4 domain is improved significantly over that of spatial domain in our implementation. It should be clear that objectively performing an accurate quantitative comparison of the speed efficiency on the proposed algorithms and the existing techniques is not an easy task due to many factors such as no detail provided regarding the sample document images and parameter settings used for experimental trials for many of the results reported in literature. In addition, comparing the speed of different algorithms is difficult because it depends on how well the algorithms have been implemented and how complex the architecture of the machine it runs on is, and how good the compiler is. All the algorithms evaluated here are the current authors' implementations, and the emphasis of this evaluation is placed on both correctness and efficiency.

3.7 Evaluation of MG4

Usually, two important figures of merit for a coding scheme are coding complexity and compression efficiency. However, due to the nature of this research, the third significant measurement, the compressed-domain processing is included to evaluate

the merit of a compression algorithm.

The implementation of MG4 has been demonstrated in Section 3.2 and its detailed flow chart is shown in Figure 3.9. The MG4 scheme is easy to implement and its memory requirement is low. Generally, the two scan lines of the image need to be stored in the memory. For fax images, the width of a scan line is 1728 pixels, therefore 4k memory is enough for this purpose. After the coding line has been coded, it becomes the reference line for next compression iteration. All the black runs on the coding lines have been experienced to detect their mode attributes. Then these modes are used as indexes of look-up table to get their code words, which are the final form of the compressed image. MG4 and Group IV have identical coding principal, i.e., removing two-dimensional redundancy as well as one-dimensional redundancy, making them possess similar coding procedures and thus comparable time complexity. In our implementation, an average of 0.43 and 0.44 CPU seconds are needed to compress ITU standard images for MG4 and Group IV encoding respectively.

Run length encoding is included for the evaluation of MG4 because many algorithms have been developed to process the run data [24, 49, 60] and hence in this sense it can be regarded as a relatively flexible coding scheme compared with existing compression algorithms. In addition, its implementation is simple and easy to realize. In this implementation, 0.39 CPU seconds are needed on average to compress a standard ITU image. However, the compression ratio for this scheme is somewhat restricted by the unexploited two-dimensional correlation.

A comprehensive comparison was made between Group IV, run-length, and MG4 by addressing the major issues of a coding scheme, i.e., implementation complexity, compression ratio, and compressed-domain algorithm feasibility. Notice that, because of the nature of MG4, algorithms applied to run-length encoding may be utilized by MG4 as well, after a slight variation. The results listed in Table 3.6 suggest that

Group IV has the best compression but lacks processing flexibility and is not fully exploited in this aspect; that run-length encoding is flexible for many operations of document analysis, but compression is relatively low; and that MG4 combines the

Table 3.6: Summary of Evaluation of the three Algorithms

Item/Algorithm	Run-length	G4	MG4
Compression Complexity (CPU second)	0.39	0.43	0.44
Compression Ratio	7	17	15
Compressed-domain Algorithm Feasibility:			
Skew Detection	No	Yes	Yes
Rotation	Yes	No	Yes
Connected Component Detection	No	No	Yes
Similarity Estimation	No	Yes	Yes
Segmentation	Yes	No	Yes
Feature Extractor for Document Recognition	Yes	No	Yes
Layout Analysis	Yes	No	Yes
Others	Yes	No	Yes if yes for Run-length

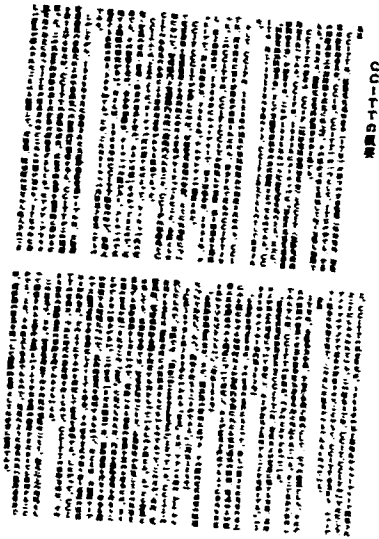
advantages of both Group IV and run-length encoding and achieves good compression while being flexible in image processing. Therefore, in terms of compression ratio, implementation complexity and compressed-domain processing feasibility, MG4 is a good choice between those algorithms efficient in compression and those flexible in image processing.

3.8 Conclusion

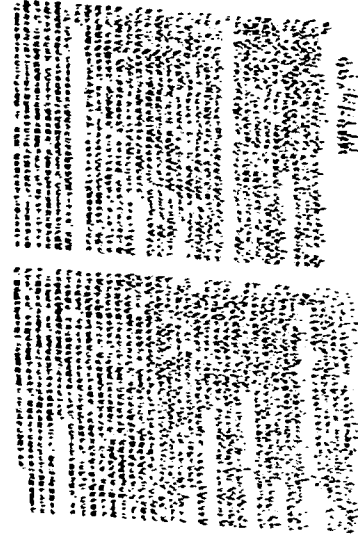
A comprehensive study on effectiveness of the Modified Group IV compression code has been performed in this chapter. Different from traditional lossless compression algorithms which are designed to yield merely high compression in most cases, MG4 is aimed at achieving both processing and compression efficiency. The Modified

Group IV preserves structural information of original image data and thus reasonably facilitates important steps of document analysis. The objective of this research is to pursue a good trade-off between image processing and efficient compression so that some major issues of compressed-domain processing will be addressed.

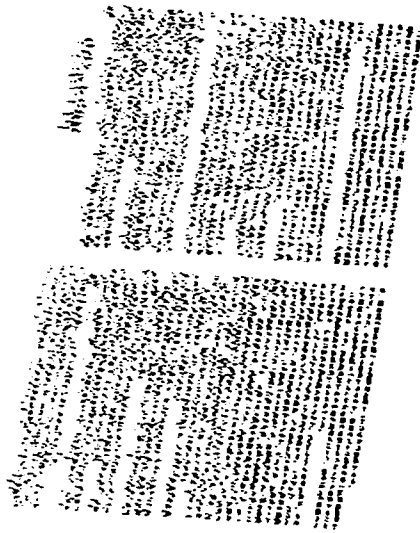
To illustrate this idea, in the above sections, from the viewpoints of compression efficiency and processing flexibility of image operations, the MG4 coding principle and its feature-preserving behavior in compressed domain have been investigated and examined. The two popular coding schemes in the area of bi-level image compression such as run-length and Group IV have been studied and compared with MG4 in the three aspects of compression complexity, compression ratio, and feasibility of compressed-domain algorithms. The features of the MG4 coding scheme can be generalized as object-oriented, differential, and run-length based and their usefulness is also ensured by the proposing of a general procedure targeted for document analysis. The future work will involve further exploration of the characteristics existing in the compressed domain after identifying the needs of an application, in order to allow fast processing while achieving efficient compression.



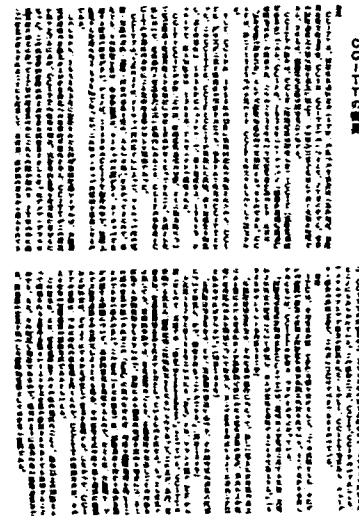
(a)



(b)



(c)



(d)

Figure 3.21: Original Image: ITU 7

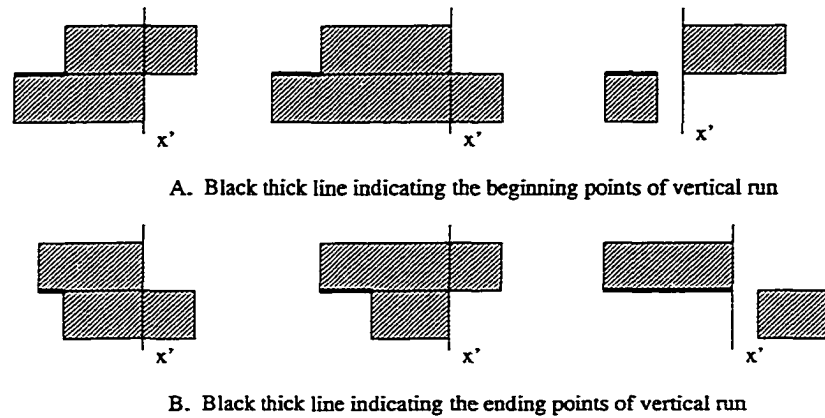


Figure 3.22: Finding the Beginning Points and Ending Points of Vertical Runs

In facsimile a photocell is
 subject copy. The variations
 the photocell to generate a
 signal is used to modulate a
 destination over a radio

In facsimile a photocell is
 subject copy. The variations
 the photocell to generate a
 signal is used to modulate a
 destination over a radio

Image rotated by MG4

Image rotated pixel-by-pixel

Figure 3.23: An Example of Rotation

CHAPTER 4

COMPRESSED-DOMAIN TECHNIQUES FOR JBIG-ENCODED IMAGES

In this chapter, some useful techniques are presented for processing the JBIG encoded images directly in the compressed domain or concurrently while they are being decoded. Features, extracted from partially decompressed JBIG representations of images, are employed in the algorithms for connected component extraction, and document segmentation to achieve considerable improvement in terms of computation and memory savings. Two schemes of JBIG compression coding, namely context modeling and progressive coding, have been utilized to obtain an effective method for feature extraction in the compressed domain.

4.1 Introduction

Most of today's image compression methods are concerned mainly with signal distortion, bit rate, and coding complexity. However, document analysis algorithms, commonly based on raster images, are usually independent from the design of image compression algorithms. Before feature extraction, traditional document analysis algorithms decompress the document image completely. The overall system performance can be improved by joint study of image compression and feature extraction since there should be considerable synergy between them.

Given the existing binary image compression techniques, JBIG [27], what can be possibly done in the compressed domain? There are several advantages to pursue as many functionalities as possible in the compressed domain. First, since there

is less data in the compressed domain than in the original uncompressed domain, it is possible to reduce overall computational cost. Second, most stored document materials are compressed. Analyzing images in the compressed domain can avoid the overhead of decoding and re-encoding of already compressed images. In fact, many compression algorithms act as some kind of information filter or content transformer, which can provide good pre-processing for subsequent image analysis. Algorithms that extract features during the compression or decompression process also result in significant computational savings.

After a thorough study of the JBIG standard, some useful techniques that permit processing and analysis while decompressing the JBIG compressed images are found. Section 4.2 provides an overview of the JBIG standard to the extent that is necessary for following the chapter. In Section 4.3, the techniques and their applications in connected component detection and document segmentation are explained. Data complexity analysis of JBIG-domain processing is implemented in Section 4.4. Experimental results are shown in Section 4.5. And the conclusion is drawn in Section 4.6.

4.2 JBIG Compression Scheme

The JBIG standard can be viewed as a combination of two algorithms, a progressive transmission scheme and a lossless compression algorithm. Essentially, JBIG is a context-based prediction method coupled with arithmetic coding. Coding is based on a template model prediction and adaptive arithmetic coding is used to encode the predictions. Figure 4.1 shows a two-line template to determine the context of the currently processed pixel. (The JBIG specification also supports a three-line template.) The pixel to be coded is marked X, while the pixels to be used for template are marked O or A. The O and A pixels are previously encoded pixels and are available

to both encoder and decoder. The A pixel can be thought of as a floating member of the neighborhood. Its placement is dependent on the input being encoded. As there are 10 bits in this template, there are 1024 different arithmetic coders. Each pixel is encoded using one of 1024 parallel arithmetic encoders/decoders according to the context (pattern) made by its neighbor pixels.

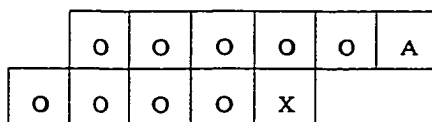


Figure 4.1: Two-line Template for the JBIG Context

The JBIG standard allows for the generation of progressively lower-resolution images. The JBIG specification recommends generating one lower-resolution pixel for each two-by-two block in the higher-resolution image. It provides a table-based method for resolution reduction. The table is indexed by the neighboring pixels shown in Figure 4.2, in which the circles represent the lower-resolution layer pixels and the squares represent the higher-resolution layer pixels. Each pixel contributes a bit to the index. The table is formed by computing the expression

$$4e + 2(b + d + f + h) + (a + c + g + i) - 3(B + C) - A$$

If the value of this expression is greater than 4.5, the pixel X is tentatively declared to be 1. The table has certain exceptions to this rule to reduce the amount of edge smearing, generally encountered in a filtering operation. The progressive encoding mode utilizes a very sophisticated resolution reduction algorithm which offers the highest quality for low resolution version that preserves the shape of characters as well as the integrity of thin lines and dithered images.

The JBIG standard defines two kinds of contexts: one for the base, or lowest, resolution layer (as shown in Figure 4.1), and the other for the differential, or higher, resolution layers. In this chapter, the former will be addressed.

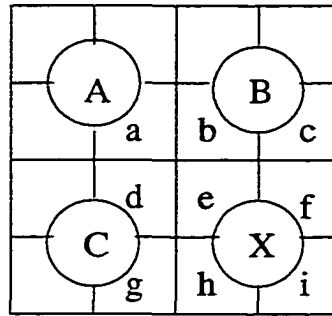


Figure 4.2: Pixels Used to Determine the Value of a Lower-level Pixel

4.3 Compressed-domain Processing

By compressed-domain processing, we mean performing feature extraction on compressed images without decoding, or with minimal decoding only, or performing feature extraction in the course of decoding the compressed image.

In image processing, there are algorithms that find patterns of pixels in adjacent rows of the image raster to get some useful information for feature extraction. If the patterns can match or can be matched by the JBIG coding template, there comes a synergy. The image can be decoded and the algorithm can be executed concurrently instead of running the algorithm after the image is decoded. This will mean sharing resources and avoiding repeated operations, hence saving storage space and computation time.

When the resolution of an image is reduced, some features are lost, yet some are preserved. Even some features are easier to be obtained through a low-resolution image. To save computation time and space, some algorithms first introduce resolution reduction as a type of pre-processing. Recall that JBIG compression algorithm supports a progressive scheme that offers free low-resolution images. To extract such features, fully decompressing that image is not necessary. The image is only decoded into the lowest-resolution. The strategy in capturing features from lowest layer images is illustrated in Figure 4.3.

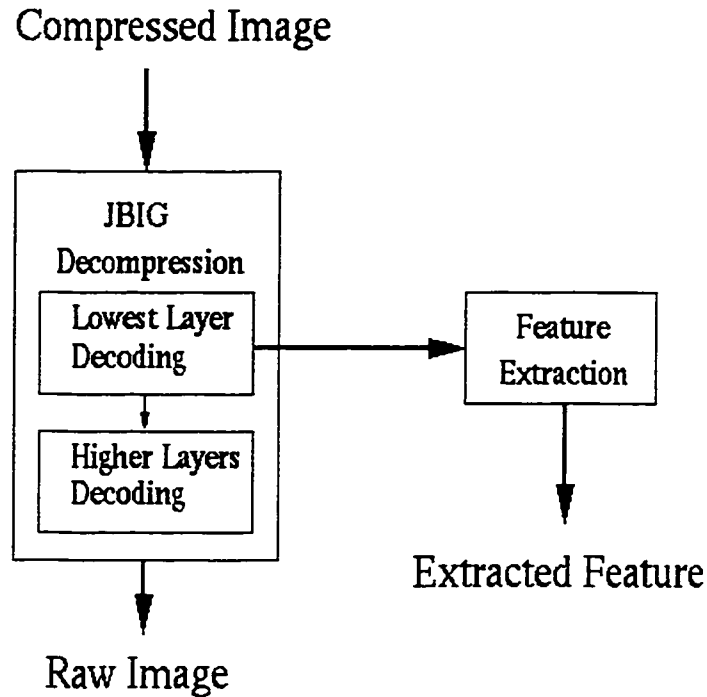


Figure 4.3: Concurrent Processing of JBIG Decompression and Feature Extraction

As an example, Kanai and Bagdanov's work [31] takes advantage of both the JBIG coding template and resolution reduction to develop a compressed-domain processing algorithm for skew estimation. The algorithm takes the fiducial reduction strategy used for processing ITU Group IV compressed images proposed by Spitz [11]. The ITU Group IV compression algorithm is a two-dimensional coding scheme that encodes the current coding line with respect to the previous (reference) line [13]. The interesting white pass-mode coding situation can be detected in the processes of encoding and decoding an image by analyzing the sequence of coding contexts. The results show that the resolution reduction scheme of JBIG minimizes the effects of noise and graphic objects in a page in the skew estimation process, and increases the computational speed. Some other applications of this approach are introduced below.

4.3.1 Connected Component Detection

In document image analysis, connected components (characters, image objects, etc.) are to be detected for the purpose of OCR or document segmentation. Images are scanned line by line and connected components are marked and labeled separately. Each label uniquely identifies a connected component. As an example, a description of connected components is presented by Bartneck [7]. Again, as mentioned in the preceding section, the patterns of pixels in two adjacent rows of the image raster are found to get some useful information for feature extraction. Then, the JBIG coding templates are exploited to match the patterns and borrowed to realize the algorithm while decoding the JBIG compressed document images simultaneously.

For the lowest layer of image obtained after resolution reduction, the number of connected components is generally reduced, compared to that of the original image. During resolution reduction, a single connected component may be split into two disconnected components, or vice versa. However, changes in component number or physical shape of specific connected components do not influence the global layout and texture characteristics of the image. In other words, features extracted from the lowest layer are still useful for image segmentation or other layout analysis.

In the following two efficient algorithms for connected component extraction are introduced, i.e., corner-detection-based and black-run-based methods.

Method 1:

Bartneck [6] described a method for binary image segmentation and generation of an image description graph, which is performed by an algorithm for contour coding, called region border line coding (RLC-algorithm). The central part of the algorithm is contour tracing method. The binary image (Figure 4.4a) is scanned line by line with a 2×2 window. The surrounding of the image is supposed to be white.

The contents of the window can assume any of 16 (2 to power of 4) different states,

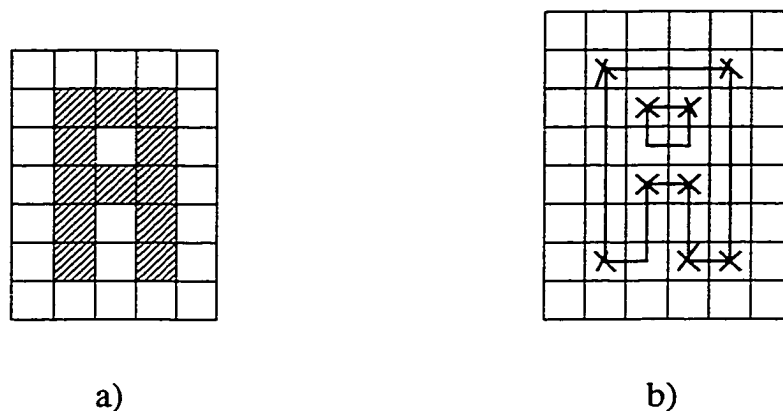


Figure 4.4: Example of a Simple Binary Image and Its Corner Image

as shown in Figure 4.5. Only those states representing corners are of interest. Concave and convex corners are distinguished. The coordinates and the type of discovered corners are stored in a corner list. The corner list is a complete representation of the binary image and is equivalent to an image, where all the discovered corners are marked (Figure 4.4(b)). The contour tracing algorithm operates on the corner list to segment the objects in the image. Further analysis can be performed to extract features of the segmented objects, such as the size, the shape, and the location of the objects.

If the image is JBIG compressed, contour tracing can be performed concurrently with the JBIG decompression process. Corner patterns can match the JBIG contexts. Figure 4.6 shows the correspondence of JBIG coding context templates to the corner patterns. A 1024 entry transition table is set up to store the corner information, each entry corresponding to one coding context represented by a 10-bit integer. Each entry in the tables consists of two bits: one to determine whether a corner is detected or not, and one for the corner type. When executing, the current coding context (a 10-bit integer) is used to index the appropriate transition table. Value "00" of the table means no corner is detected; value "10" means a concave corner is encountered, and value "11" implies a convex corner. The corner type is obtained by searching the

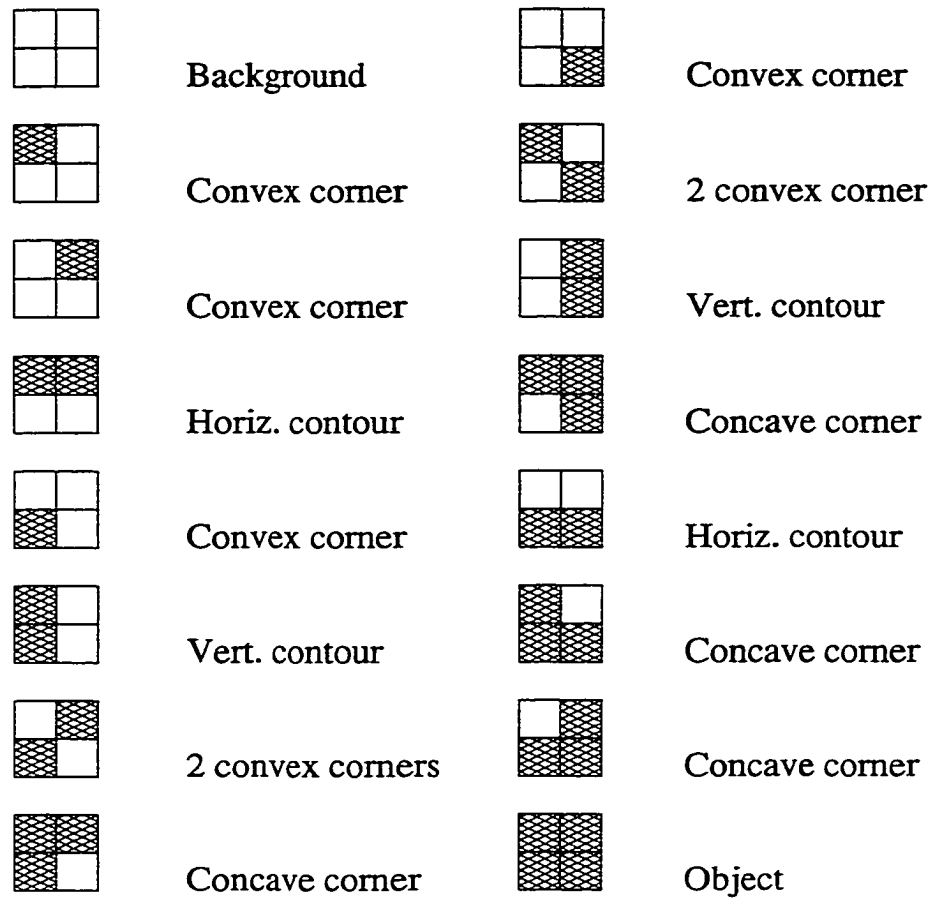
Figure 4.5: 16 Possible Patterns of the 2×2 Masks

table and is saved in the corner list with the current coordinates of the corners for further analysis.

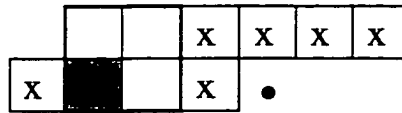


Figure 4.6: Setting Corner Patterns into the JBIG Context

Method 2: Instead of detecting the corners of connected components, the start of every single component is searched by looking for black runs which do not touch previous components. The beginning and ending of a black run could be obtained by utilizing JBIG context patterns.

The following is the main steps of the algorithm:

1. Detect the endpoints of the black runs in the decoded line and assign labels to the runs;
2. Detect the overlaps and the touches of the black runs in adjacent lines, which mean connection;
3. Assign the label of the black run on the previous line (say L) to its counterpart in the current line if connection is detected.

Identically, a table with 1024 entries that correspond to the 1024 JBIG coding contexts is constructed. Each entry is assigned four bits, i.e. S, F, R, and A. These bits control the execution of the following operations:

1. S to record the leftmost coordinate of the black run in the decoding line and assign a new label;
2. F to record the rightmost coordinate of the black run in the decoding line;
3. R to read out the label (L) of the black run on the previous line;
4. A to assign label (L) to the record list of the current line black run.

Figure 4.7 shows how bit values are assigned for different types of JBIG coding contexts, according to the patterns of four pixels in the context. Bit value of 1 will generate the corresponding operation. "x" in the contexts denotes the bit value can be zero or one, and "•" denotes position of the decoded pixel. The operations are implemented with respect to the coordinate Y of the currently decoded pixel, i.e. S and R operations get the coordinate (Y-2), F the coordinate (Y-3). Operations are implemented in the listed order, from left to right.

The coordinates and labels of connected components, which is a complete description of the raster image, are input data for further image analysis.

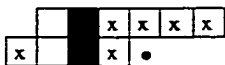
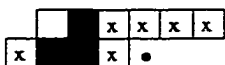
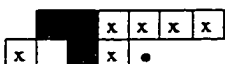
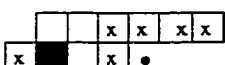
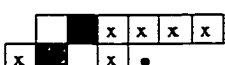
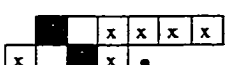
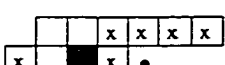
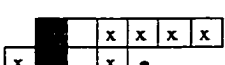
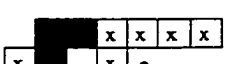
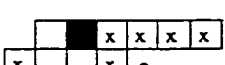
Bit-plane	Context value	S	R	F	A
	01XXXXX01X	1	1	0	1
	01XXXXX11X	0	1	0	1
	11XXXXX01X	1	0	0	1
	00XXXXX10X	0	0	1	0
	00XXXXX01X	0	1	1	1
	10XXXXX01X	1	0	0	1
	00XXXXX01X	1	0	0	0
	10XXXXX10X	0	0	1	0
	11XXXXX10X	0	0	1	0
	00XXXXX01X	0	1	0	0

Figure 4.7: Connected Component Detection

4.3.2 Page Segmentation

The aim of the segmentation process is to divide the document image into regions that contain either text, or graphics, or a halftone picture, and to find the logical components of the page. Thereafter, processing of text (usually by OCR) and figures becomes possible. Traditionally, there are two philosophies for segmentation: top-down and bottom-up. The former views the entire image globally, finds and cuts along the separators between components based on the horizontal and vertical projection profiles. In the latter, small foreground objects (i.e. marks, letters, lines, etc.) are merged together to form a complete component based on their physical proximity. Usually, top-down processing is performed on the result of bottom-up processing.

Several methods based on these two philosophies are reported. For example, Amamoto et al. [1] proposed the method to segment blocks and extract text area using the white spaces of the document image. Wieser et al. [75] presented an approach to enhance and combine the two commonly used methods to segment pages of a newspaper. Their approaches were based on uncompressed images.

Here, a method that directly performs bottom-up and top-down segmentation on the JBIG-compressed document images is presented. First, the JBIG file is decoded into the lowest-resolution layer. While decoding, the sequence of coding context that is used to decode incoming pixels is analyzed to detect a white run using the automaton shown in Figure 4.8. Each time when a white run that is longer than a pre-defined length t is detected, the corresponding portion of the black image is changed into white. This is the same as bottom-up processing, which smears the binary image by filling in the pixels between any two black pixels that are less than a certain threshold distance t apart. Assume $t > 3$ in order to match the JBIG context, which is reasonable and practical since two components that are 3 pixels apart or less should belong to the same region at lowest resolution. In Figure 4.8, a shaded box in a context displays a black pixel while an empty box indicates a white one. “ x ” means don’t-care. “ \bullet ” denotes position of the decoded pixel. The automaton works as below.

In State A, no white run that is longer than 3 has been encountered. Once a context with pattern like $c1$ appears, which indicates the beginning of a white run that is longer than 3, the machine moves to State B and output 1 to have the coordinate $x1$ of the current pixel recorded. Otherwise it remains in State A.

In State B, if a context with pattern like $c2$ appears, which indicates the end of a white run that is longer than 3, the machine moves to State A and output 1 to have the coordinate $x2$ of the current pixel recorded. Otherwise it remains in State B.

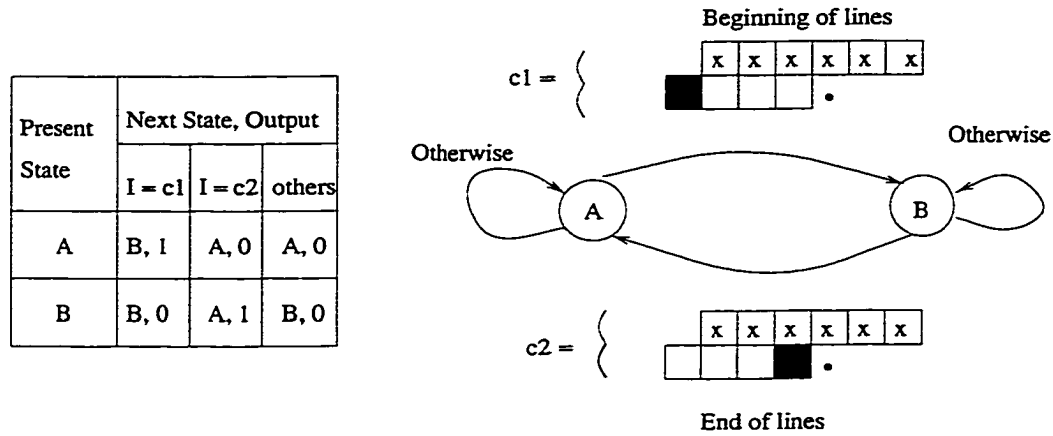


Figure 4.8: Automaton to Detect White Runs

The length L of the run and coordinates B , E of the beginning and end points of the run can be counted as below,

$$L = x2 - x1 + 2$$

$$B = x1 - 3$$

$$E = x2 - 2$$

Again two lookup tables of the coding contexts are used to encode the automaton. Every table contains 1024 entries, each of which consists of 2 bit: one is for state assignment, the other is for the output value. The implementation of the above machine is particularly easy. Only a small additional memory is needed. For example, bottom-up and top-down on the reduced resolution image of the original one (Figure 4.9) are processed and demonstrated. After finishing horizontal smearing of the image (Figure 4.10(b)), vertical bottom-up (Figure 4.10(c)) and top-down processing (Figure 4.10(d)) are performed on the lowest-resolution layer of the JBIG image. And further operation of segmentation can be done based on the result. The resulting bottom-up output is shown in Figure 4.10(e). Comparing (d) and (e) of Figure 4.10, the intermediate results based on the lowest resolution of JBIG have better potential

to achieve good segmentation. As mentioned before, some reported methods for page segmentation first take subsampling of the original image as the pre-process of block segmentation for the purpose of high speed processing. This further proves the usefulness of the investigation in the JBIG compressed domain.

4.4 Data Complexity Analysis

The data complexity of JBIG-domain processing is examined using a ITU standard document image, whose size before compression is 1728×2376 pixels. Representing the binary image by the bit map, the data complexity S is 4,105,728 bits. On the other hand, most compressed images have three or more levels of resolution reduction after JBIG encoding. Therefore, for a 200dpi source image, the lowest resolution layer will be 25dpi. Thus S , the data complexity of the image processing algorithm based on lowest resolution would be reduced by a factor of 64, i.e., 64,152. The manipulation of document images based on pixelwise data needs not only large memory space to store the decompressed images, but also long time to process these data. However, compressed-domain techniques for JBIG-encoded document images make use of only the lowest resolution image to obtain layout information of the original image and hence dramatically speed up some important steps of image processing.

4.5 Experimental Results

So far, connected component extraction and page segmentation on the lowest-resolution layer of the JBIG image are discussed. Note that bottom-up and top-down are processed on the reduced resolution image of the original one (Figure 4.9). This reduces the time complexity substantially since there is less data. The JBIG progressive encoding mode adopts a special resolution reduction algorithm that preserves the shape of characters as well as the integrity of thin lines and dithered images, thus

keeping enough information for page segmentation (see the difference between Figure 4.10(a) and Figure 4.10(f)). Hence, the result is the same as that performed on the original image, or even better since resolution reduction actually performs some kind of noise reduction function. Figure 4.10(f) presents an version of simple resolution reduction for the same original image, in which the critical edge information of graphics are destroyed severely. In our case, JBIG has already finished the preprocessing work for us and actually does a better work (see the difference between Figure 4.10(d) and Figure 4.10(e)).

A number of experiments were conducted to evaluate the performance of the algorithm, using a free implementation of the JBIG standard [34]. The experiments used the eight standard test images provided by ITU. The processing was performed for two different resolutions: Fully Uncompressed Image (FUI) and the Lowest Resolution Image (LRI). In our experiment, “hseg” and “hcc” are two algorithms used to process segmentation and connected component detection of the FUI respectively, “lseg” and “lcc” have the same functions but for the LRI. All the experiments were performed under the same computational setting, i.e. a Pentium II running at 200 MHz frequency was used, the operating system was BSD UNIX.

Table 4.1 summarizes the execution times of each algorithm for high and low resolution version of the eight ITU standard test images (in seconds).

Even though all eight ITU standard test images have been used for performance evaluation of our algorithm, the No.4 image was chosen as a representative to show visually the performance of segmentation algorithms from Figure 4.11 through Figure 4.13. In the experiments, it is found that the processing speed for the LRI is substantially faster than that for the FUI on two operations of connected component detection and segmentation by an average factor of 73 and 47 respectively. The data that needs to be processed for the LRI is only about 8K bytes while that for the FUI is

Table 4.1: Time Comparison for High and Low Resolution Processing

Image	hseg (s)	lseg (s)	hse:lseg	hcc (s)	lcc (s)	hcc:lcc
ITU1.JBG	53.20	0.68	78:1	20.95	0.44	48:1
ITU2.JBG	54.77	0.65	84:1	21.68	0.56	39:1
ITU3.JBG	56.04	1.00	56:1	21.55	0.43	50:1
ITU4.JBG	55.56	0.75	74:1	21.48	0.54	40:1
ITU5.JBG	53.11	0.67	79:1	22.40	0.45	50:1
ITU6.JBG	50.03	0.70	71:1	23.37	0.46	51:1
ITU7.JBG	50.42	0.69	73:1	21.56	0.43	50:1
ITU8.JBG	50.65	0.67	76:1	21.88	0.45	49:1
Average	52.97	0.73	73:1	21.86	0.47	47:1

about 512K bytes. Thus, the difference of the data size between the lowest resolution layer and the highest resolution layer of the image determines the performance of the algorithm.

Figure 4.13 presents segmentation results by applying the segmentation algorithm mentioned before to the lowest layer and the full resolution layer of ITU 4 image, respectively. The segmentation result based on the lowest layer is comparable with the one based on spatial domain and even shows better denoising ability generally due to the resolution reduction.

Techniques for corner detection and connected component detection were applied to LRI and FUI of ITU5, respectively and experimental results are demonstrated in (a) and (b) of Figure 4.14. The connected components obtained from LRI may not offer valuable information about single component such as the height, the width, or the area of the object, but, they do provide some important features for segmentation, or for further layout analysis.

4.6 Conclusion

Two features of JBIG compression coding are used to obtain an effective method for feature extraction in the compressed domain. One is the context modeling, the other is the progressive coding. It can be seen that these two features are independent from the JBIG compression method. Therefore, for any context model based compression method, concurrent extraction of certain features is possible if the features can be analyzed from some patterns of pixels that match the coding context templates. For any progressive coding method, some information is still preserved in the reduced resolution images such that, with minimal decoding only, feature extraction can be performed in the compressed domain. Reduction of time complexity and saving of memory space are obvious. Sometimes, an even more robust result can be achieved.

A general approach and the implementation of some compressed-domain image processing technologies for the JBIG compression algorithm have been presented. The extension of these techniques to other compression methods with certain identical features with the JBIG algorithm was discussed. It is believed that the direct information accessibility for the traditional compression algorithms can be achieved to some extent by taking advantage of some good properties that exist in both compression scheme and conventional image processing algorithms.

Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de f pour les valeurs numériques indiquées page précédente.

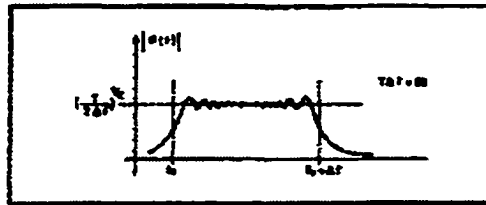


FIG. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 < f < f_0 + \Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0 + \Delta f$, filtre ne modifiant pas la phase des composants le traversant ;



FIG. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe T_R décroissant linéairement avec la fréquence f suivant l'expression :

$$T_R = T_0 + (f_0 - f) \frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).

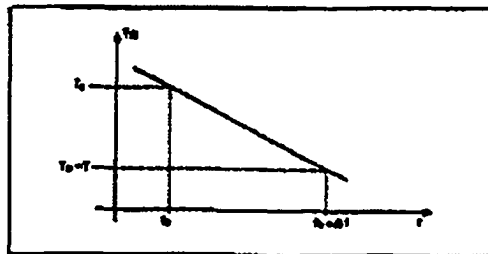


FIG. 4

telles ligne à retard est donnée par :

$$\varphi = -2\pi \int_0^f T_R df$$

$$\varphi = -2\pi \left[T_0 + \frac{f_0 T}{\Delta f} \right] f + \pi \frac{T}{\Delta f} f^2$$

Et cette phase est bien l'opposé de $\phi(f)$.

à un déphasage constant près (sans importance) et à un retard T_0 près (inévitables).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard T_0 près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre f_0 et $f_0 + \Delta f$, et nulle de part et d'autre de f_0 et de $f_0 + \Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0 + \Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression

$$\text{est de } \frac{T}{1/\Delta f} = T\Delta f$$

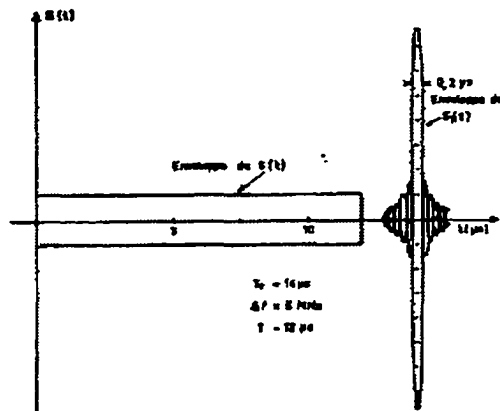


FIG. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse f_0 , qui met un temps T_0 pour traverser. La fréquence f entre à l'instant $t = (f - f_0) \frac{T}{\Delta f}$ et elle met un temps

$T_0 - (f - f_0) \frac{T}{\Delta f}$ pour traverser, ce qui la fait ressortir à l'instant T_0 également. Ainsi donc, le signal $S(t)$

Figure 4.9: The Original Document Image

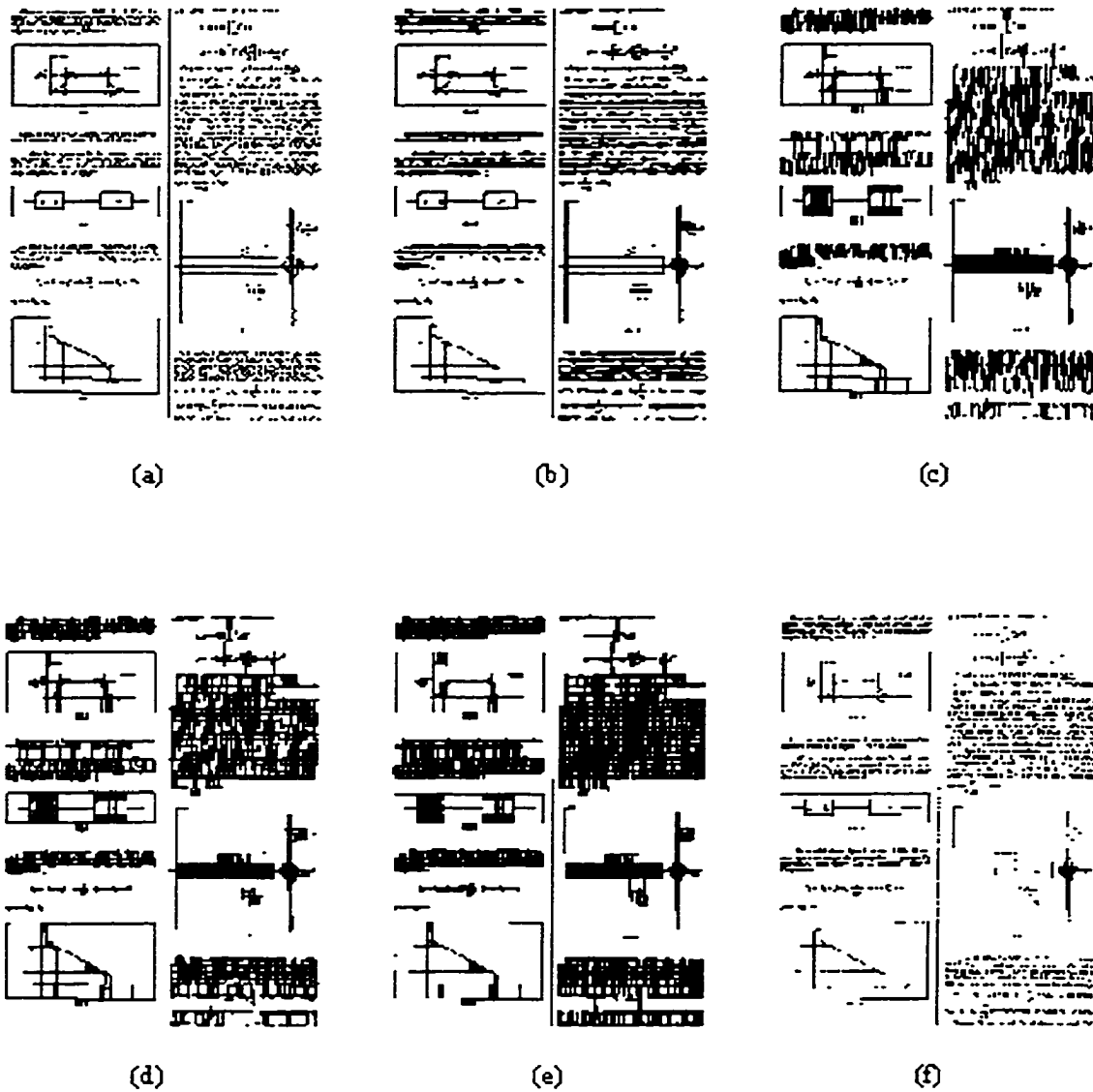


Figure 4.10: (a) The Lowest Resolution Layer of a JBIG Compressed Document Image; (b) Horizontal Smearing of (a); (c) Vertical Smearing of (a); (d) Merging (b) and (c) by Using OR Operation; (e) the Bottom up Result from (f); (f) The Image Obtained from Simple Resolution Reduction

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.

Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analyses-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 8 à 10 personnes représentant les services les plus divers concernés par le projet, et comporte obligatoirement un bon analyste attaché à l'application.

II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.

À l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.

Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins les tiers des données seront concernées par des traitements en temps réel.

Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.

L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.

L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.

L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens ; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger, et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gèrera" environ un million d'abonnés à la fin du VIème Plan.

La mise en place de ces centres a débuté au début de l'année 1971 : un ordinateur IHIS 50 de la Compagnie Internationale pour l'Informatique a été installé à Toulouse en février ; la même machine vient d'être mise en service au centre de calcul interrégional de Bordeaux.

Photo n° 1 - Document très dense lettre 1,5mm de haut -
Restitution photo n° 9

Figure 4.11: ITU 4 Document Image

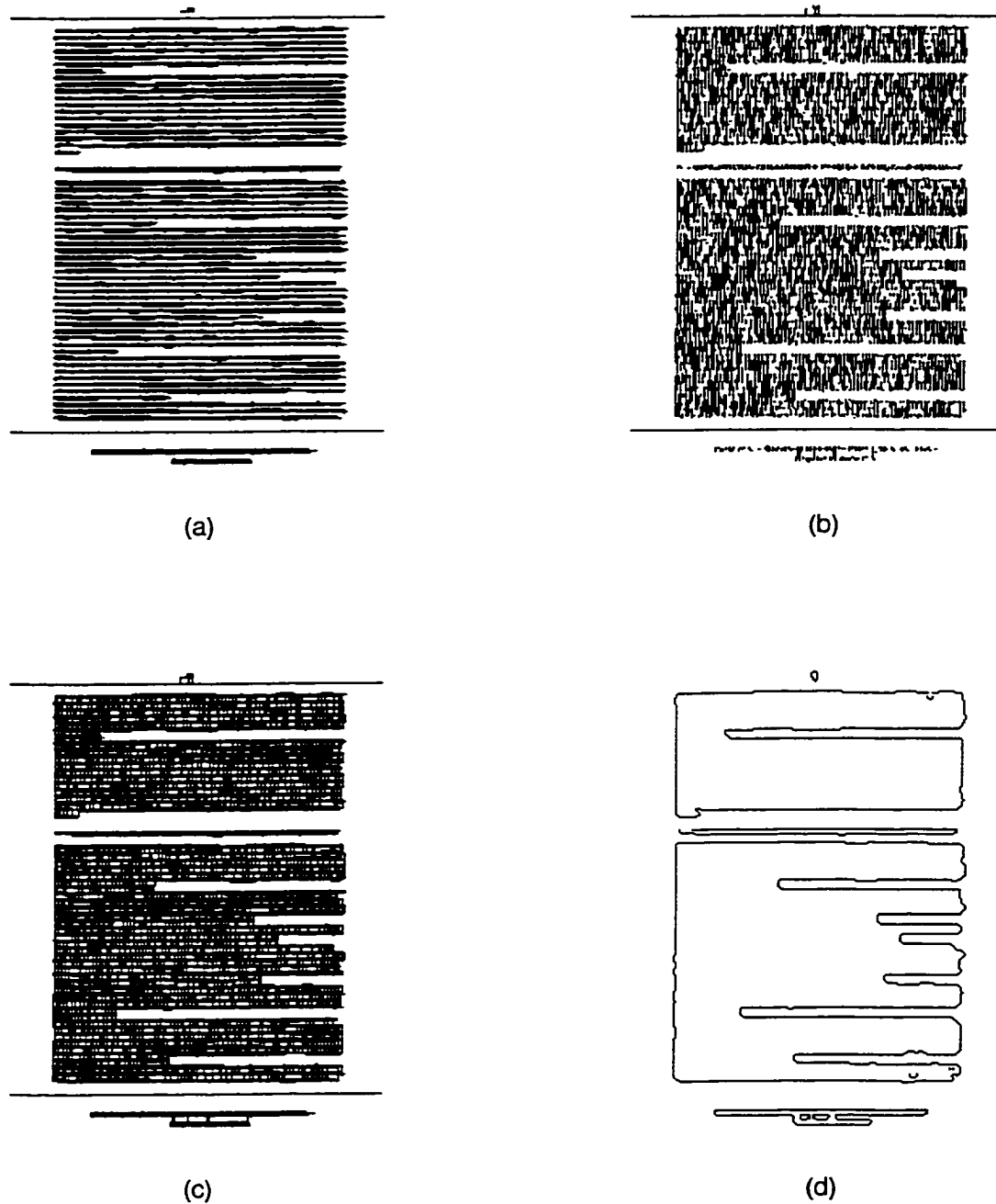


Figure 4.12: (a) Horizontal Smearing of the Lowest Layer Image; (b) Vertical Smearing of the Lowest Layer Image; (c) Merging (b) and (c) by Using OR Operation; (d) Segmentation Result

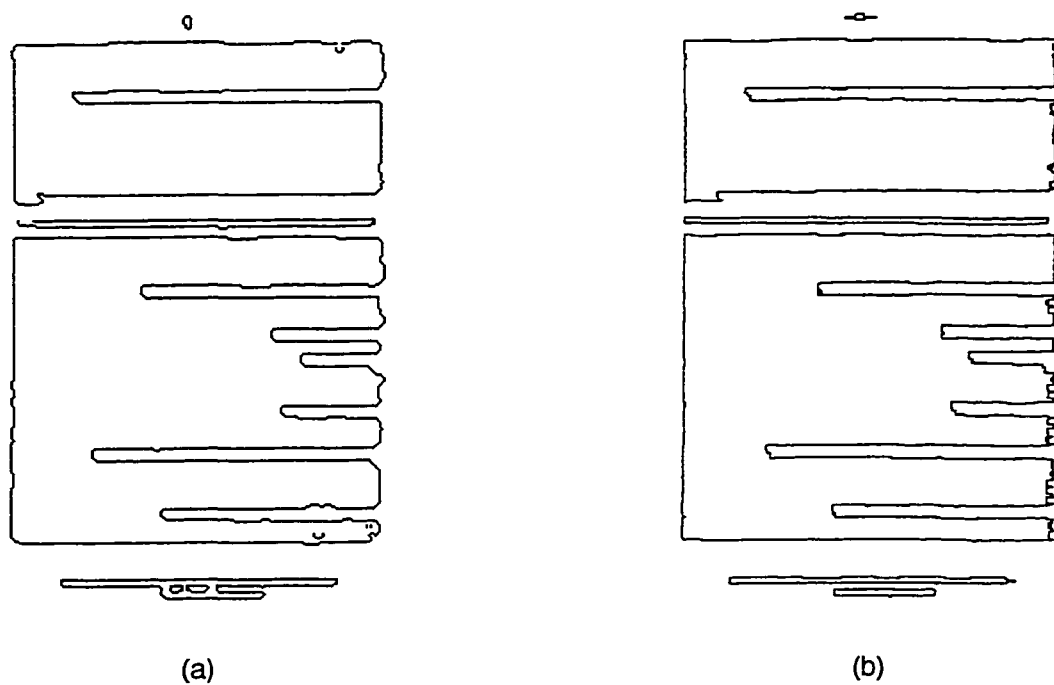


Figure 4.13: Segmentation Results Based on (a) LRI; (b) FUI

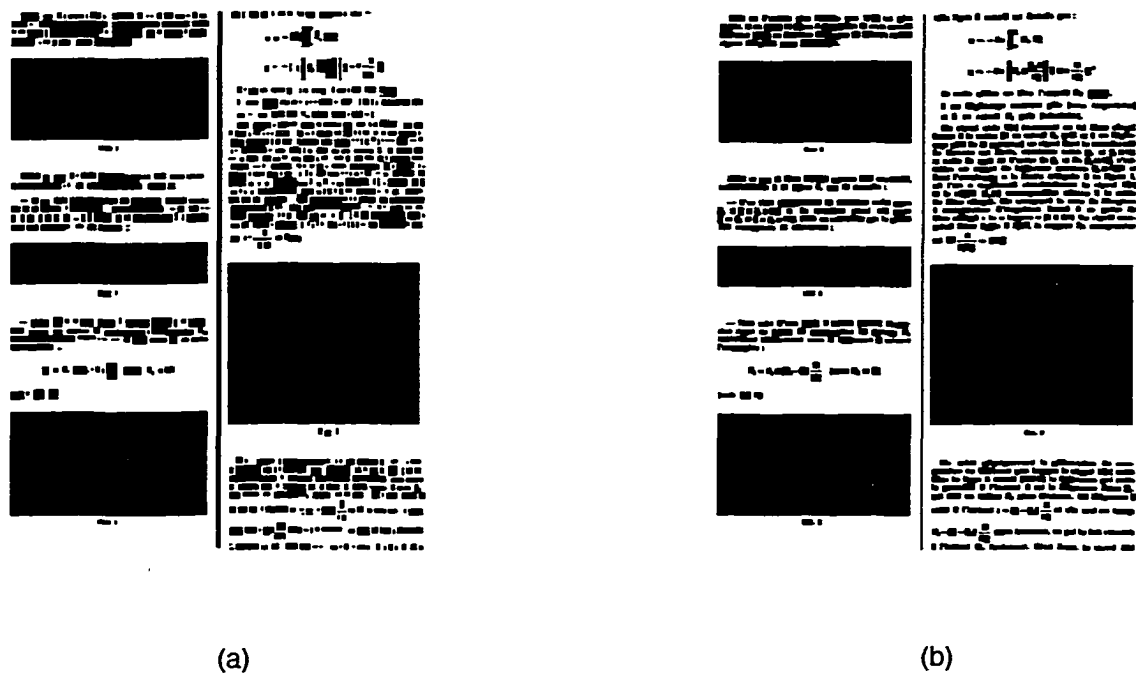


Figure 4.14: Corner Detection and Connected Component Detection for (a) LRI; (b) FUI

CHAPTER 5

WAVELET TRANSFORMS

This chapter will briefly introduce the concepts of wavelets and the multiresolution analysis. The general properties of wavelet systems are listed below. In particular, the discrete wavelet transform, a useful tool for image processing, is explained and the implementation of its fast algorithm is also illustrated.

5.1 Introduction

The early work of wavelets can be dated back to the 1980's by Morlet, Grossmann, Meyer and Mallat. Their studying of wavelet transforms was motivated by the fact that some seismic signal could be well presented by translations and dilations of a simple, oscillatory function called *wavelet*. Later, a wavelet paper written by Ingrid Daubechies [17] in 1988 caused the attention for its applications in many areas such as signal processing, statistics, and numerical analysis.

As we know, Fourier analysis expands signals or functions in terms of sinusoids, which proves extremely valuable for periodic, time-invariant, or stationary phenomena. However, the goal of wavelets is to create a set of basis functions so that the signal can be represented based on these functions. Wavelet can be defined as an oscillating function of time or space and allows simultaneous time and frequency analysis with a flexible mathematical foundation. In other words, it provides localization in both time and frequency and is therefore a tool for the analysis of transient, nonstationary, or time-varying signals.

5.2 Wavelets and their Characteristics

In fact, wavelets are based on the concepts that already exist in different disciplines. The “scale-space” for multiscale visual modeling has been proposed by computer scientists before the appearance of wavelets. However, wavelets provide a sound and rigorous formulation of these concepts. Perhaps, the most significant advancements made from the traditional scale-space representations are that multiscale representations are designed to be invertible, compact and computationally-efficient. B-spline approaches can exploit their intrinsic relations. Almost every wavelet (continuous, orthogonal, bi-orthogonal ...) can be factorized or derived from B-spline kernels. As a result, these wavelet transforms could be fast implemented using filter bank techniques. These observations show that wavelet is an evolution of scale-space, but with a solid mathematical foundation. Nevertheless, it should be noted that wavelet theory also has its own limitations, i.e., it is just one model for linear multiscale signal smoothing. There are many other nonlinear models for multiscale modeling in the scale-space literature.

In wavelet representation, a signal is represented in terms of basis functions that are localized in time and frequency. For example, Haar wavelet, the simplest possible orthogonal wavelet system, is shown as follows,

$$\psi_{0,0}(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \end{cases}$$

From the above mother function, the following set of functions can be obtained:

$$\psi_{j,k}(x) = \psi_{0,0}(2^j x - k)$$

Notice that the functions become more and more localized in time as j (defined as dilation) increased. This localization action allows us to represent local changes accurately using few coefficients. And the effect of k is to translate or shift the

wavelet. Therefore, from a mother wavelet the components of a wavelet expansion can be obtained through the actions of dilation and translation.

In brief, the three general characteristics of wavelet systems are:

- They are a two-dimensional expansion set (a basis) for some class of one or multi-dimensional signals.
- The wavelet expansion gives a time-frequency localization of the signal. It means most of the energy of the signal is well represented by a few expansion coefficients, $C_{j,k}$.
- The calculation of the coefficients from the signal can be implemented efficiently. Many wavelet transforms can be calculated with $O(N)$ operations. Generally wavelet transforms require $O(N\log(N))$, same as Fast Fourier Transform (FFT).

More specific characteristics of wavelet systems are :

- They are generated from a single scaling function or wavelet by simple scaling and translation.
- Almost all useful wavelet systems also satisfy the multiresolution conditions.
- The lower resolution coefficients can be calculated from the higher resolution coefficients by a tree-structured algorithm called a *filter bank*, i.e., efficient calculation.

5.3 Discrete Wavelet Transform

The discrete wavelet transform means the signal decomposition with a family of real orthonormal bases generated from the mother wavelet $\psi_{m,n}(x)$ by scaling and translation,

$$\psi_{m,n} = 2^{-m/2}\psi(2^{-m}x - n) \quad (5.1)$$

where m, n are integers.

Then the signal $f(x)$ can be expanded as,

$$f(x) = \sum_{m,n} c_{m,n} \psi_{m,n}(x) \quad (5.2)$$

where wavelet coefficients $c_{m,n}$ can be calculated by inner products, i.e.,

$$c_{m,n} = \int f(x) \psi_{m,n}(x) \quad (5.3)$$

To construct the wavelet function $\psi_{m,n}(x)$, a scale function $\phi(t)$ need be determined. The $\phi(t)$ can be expressed in terms of a weighted sum of shifted $\phi(2t)$ as

$$\phi(x) = \sum_k h(k) \sqrt{2} \phi(2x - k) \quad (5.4)$$

And the mother function $\psi(x)$ has a relation with $\phi(t)$ according to

$$\psi(x) = \sum_k g(k) \sqrt{2} \phi(2x - k) \quad (5.5)$$

where $g(k)$ should satisfy,

$$g(k) = (-1)^k h(1 - k) \quad (5.6)$$

The coefficients $h(k)$ and $g(k)$ are very important for calculating the wavelet coefficients $c(k)$ and detail coefficients $d(k)$ decomposition. As mentioned before, one of the advantages of wavelet analysis is that the lower resolution coefficients can be calculated from the higher resolution coefficients by the filter bank. Figure 5.1 schematically presents a perfect reconstruction filter bank (where $M = 2$), which provides a recursive algorithm for wavelet decomposition through $h(k)$ and $g(k)$ as well as reconstructing the original function from wavelet coefficients $c(k)$ and detail coefficients $d(k)$.

If the output of the synthesis filter bank is equal to the input of the analysis filter bank, then the filter bank has perfect reconstruction. The symbol $\downarrow M$ indicates the operation of down-sampling by M , i.e., the process of discarding all but every M 'th

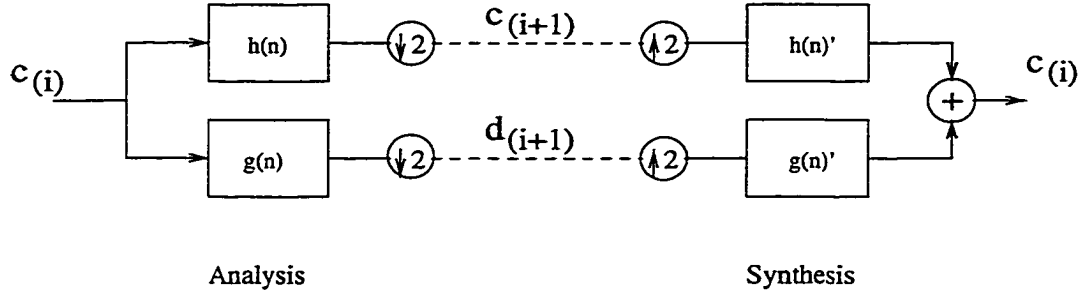


Figure 5.1: Perfect Reconstruction Filter Bank for Fast Wavelet Transform Algorithm.

sample, while the symbol $\uparrow L$ stands for the operation of up-sampling by L , i.e., the process of inserting all but every L 'th sample.

In this work, the 2-D discrete wavelet is of interest, whose formulation can be regarded as separable, i.e., the wavelets may be expressed as the products of two one-dimensional functions. The corresponding 2-D filter coefficients can be represented by the tensor product,

$$h_{LL}(k, l) = h(k)h(l), \quad h_{LH}(k, l) = h(k)g(l)$$

$$h_{HL}(k, l) = g(k)h(l), \quad h_{HH}(k, l) = g(k)g(l)$$

The filter outputs are called low-low (LL), low-high (LH), high low (HL), and high-high (HH) bands (detailed description can be found in the wavelet literature [9]).

Roughly, the magnitude of the wavelet coefficient indicates the presence or absence of a particular pattern while its sign indicates the phase of the pattern. In wavelet transform domain, the underlying texture is the same regardless of the small shifts and therefore wavelet analysis has been applied to texture detection in image processing for more than a decade. In the following chapter, the state of the art techniques will find their applications in document image processing.

CHAPTER 6

DOCUMENT SEGMENTATION USING WAVELET METHOD

Wavelet transforms have been widely used as effective tools in texture segmentation in the past decade. Segmentation of document images, which usually contain three types of texture information, namely, text, picture and background, can be regarded as a special case of texture segmentation. B-spline wavelets possess some desirable properties such as being well localized in time and frequency, and being compactly supported, which make them an effective tool for texture analysis. In this chapter, cubic B-spline wavelets are applied to document images; thereafter, a simple and efficient new algorithm for document segmentation is developed. Each texture is featured by several regional and statistical characteristics estimated at the outputs of high frequency bands of spline wavelet transforms. Then three-means classification is applied for classifying pixels with similar features. We also examine and evaluate the contributions of different factors to the segmentation results from the angles of decomposition level, frequency band, feature selection and wavelet function. Segmentation experiments indicate that this approach is superior to other traditional methods in terms of memory space and processing time.

6.1 Introduction

Wavelet transforms play an important role in texture classification and image segmentation. Providing an alternative to the classical Short-Time Fourier Transform (STFT), the state of the art wavelet transforms are most suited for the analysis of non-

stationary signals. Usually, for 1-D and 2-D signals, singularities and abrupt variation points carry more information than other slowly changed points. Given a document image, the locations of these singularities and abrupt variation points provide the image features, which is closely related to wavelet transform. Observing that an image can be best segmented using different features in different areas, Porter *et al.* [51] propose an automatic scheme which can select the optimal features using wavelet analysis. In [36], wavelet packet representations at multiple scales are used as the texture signature, and appropriate features extracted from wavelet decompositions are measured. Based on this “signature”, classification can then be performed.

Among wavelet families, orthogonal wavelet systems attract most attention in both research and applications due to their many practical features, such as yielding robust and symmetric formulations. However, resultant disadvantages are also obvious. Complicated design equations are needed and desired linear phase analysis is unavailable. It has also been shown that real-valued compactly supported orthonormal wavelets are not symmetric nor antisymmetric. Yet, since the wavelet is used as a bandpass filter in signal analysis, symmetry and antisymmetry are equivalent to linear phase and generalized linear phase, respectively. So to maintain the minimum support, one must give up a certain structure in wavelet function. Actually at the expense of energy partitioning efficiency, the biorthogonal wavelet system, which uses a nonorthogonal basis and dual basis, can be developed and then greater flexibility can be obtained. B-splines biorthogonal wavelets are compactly supported for decomposition and reconstruction. Other properties include symmetry and exact reconstruction which are possible with Finite Impulse Response (FIR) filters (but not in orthogonal case).

Unser *et al.* show the usefulness of the spline wavelet transform for texture analysis and segmentation in [70]. Different types of spline wavelet decompositions are

implemented, and their properties are compared and evaluated. Unser *et al.* [71] characterize different textures using a set of channel variances estimated at the outputs of the corresponding filter banks. In the same paper, comparison of the classification performance of various orthogonal and biorthogonal wavelets is made. Orthogonal representation is especially relevant in coding application. The cardinal representation is well suited for the conventional signal processing for it provides a precise rendition of the underlying continuous functions in terms of the sampled values. The B-spline wavelets can be closely approximated by Gabor functions, and are therefore very well localized in space and frequency. Another feature of the B-spline representation is the compact support of the basis functions, which could be very useful for performing numerical computation. The main advantage of the dual representation is the simplicity of the analysis filters (useful for some pattern recognition applications). It seems that B-spline and D-spline wavelets can be the candidate wavelets for document segmentation.

This work addresses an important step in document analysis, i.e., segmenting document images using wavelets. The significance of document image segmentation is evident and much research has been done in this area. However, major work focuses on finding discriminators for different textures based on feature extraction in spatial domain. Using wavelet-domain features only, without combining spatial-domain information to facilitate the operation of image segmentation, is an open research problem with great significance. Due to the reasons listed in the previous paragraphs, cubic B-spline wavelets will be used in the study of document image segmentation. In addition, another useful property of cubic B-spline wavelets is the availability of an efficient implementation, i.e., a fast algorithm based on this wavelet basis could be conducted by means of Quadrature Mirror Filter (QMF) [71].

The rest of the chapter is organized as follows: In Section 6.2, the previous work on

document segmentation will be reviewed and the design strategy of this work will also be stated. Then an introduction of B-spline wavelets used in this research is presented in Section 6.3. The new segmentation algorithm based on B-spline wavelet transform is proposed in Section 6.4. After applying the new segmentation algorithm to a series of test data, we present some experimental results and discuss the performance associated with several parameters considered in this investigation in Section 6.5 and 6.6, respectively. Finally, in Section 6.7 we summarize the results and conclude the chapter.

6.2 Document Segmentation

In the early stage of document analysis, structure layout analysis [19] is performed to obtain groups of document components by a physical segmentation. Usually, segmentation can be implemented to isolate words, text lines, and structural blocks, i.e., separated paragraphs or table-of-contents entries. Thus, functional layout analysis can be conducted by labeling the structural blocks and indicating functions of the blocks. Then, OCR techniques are applied to these isolated regions. In addition, digital computers and computer networks make it possible to access the electronically stored documents. We need to isolate text from pictures and graphics so that the text region can be stored and searched efficiently.

The established segmentation algorithms for document images can be generally grouped into top-down and bottom-up classes. Most top-down techniques are developed based on RLS algorithm and the projection profile cut. RLS algorithm is applied in both horizontal and vertical directions to binary images, and then the logic AND of these two output images would consist of blocks which represent different functional areas. Such blocks can be further classified and assigned a specific class. This algorithm first views the entire image globally and then divides the image into small

areas by $x - y$ cuts. The bottom-up algorithm is different from the top-down in the sense that it first groups pixels into connected components and then merges them into larger areas.

Since text regions in a document are very unique and are quite distinct from the texture of background or non-text regions, it is possible to separate text from other non-text regions, such as graphics and pictures. Document segmentation can be implemented in a number of ways. In [29], Jain *et al.* propose a method based on multichannel filtering techniques for document segmentation. Two-dimensional Gabor filters are used to extract texture features and this algorithm has been shown to perform very well for a variety of texture classification and segmentation tasks. In the same paper, the Gabor filters' special characteristics of being well localized in space and frequency are utilized. However, the disadvantage of the above method is also obvious. First, it has intensive computational complexity. Second, the outputs of Gabor filter banks are not mutually orthogonal and there is considerable correlation between texture features. These problems can be avoided if one uses a different wavelet transform which would give a time-frequency localization of the signal and at the same time, the calculation of the coefficients from the signal can be carried out efficiently. Polynomial splines basis functions (B-spline) have a relation to the multi-resolution analysis of L_2 . B-spline wavelet transforms possess a property which is very similar to that of Gabor functions. One of the attractive features of polynomial splines is that they can be generated by repeated convolutions of a rectangular pulse. Moreover, polynomial splines have good smoothness properties. It is possible to achieve any degree of regularity by simply increasing the order of the splines. Another advantage of splines is that they can be manipulated easily because of their analytic forms. Operations such as differentiation and integration can be performed in a straightforward manner. Above properties imply the possibility of simplifying the

development of numerical algorithms for image segmentation using a B-spline-based approach. Some related work on texture segmentation has been reported in [39, 71].

6.3 B-spline Wavelets

Polynomial splines are usually used in approximation theory and numerical analysis. They possess some attractive features [59] which can lead to a variety of applications. They have relatively smooth functions and also easy to store, manipulate, and evaluate on a digital computer. They have a simple and explicit analytical form. The basis functions of B-splines are compactly supported and can be generated by repeated convolution of a rectangular pulse. These features mean that fast algorithm can be obtained in the development of numerical algorithms for wavelet transforms in image processing.

The central B-spline with an order of n , denoted by β^n , can be generated by repeated convolution of a B-spline of order 0.

$$\beta^n(x) = \beta^0 * \beta^{n-1}(x) \quad (6.1)$$

In the above equation, $\beta^0(x)$ denotes the characteristic function in the interval $[-\frac{1}{2}, \frac{1}{2})$. Based on the B-spline polynomial, B-spline wavelets with underlying symmetric FIR filters can be constructed [68]. Figure 6.1 illustrates a representation of B-spline for $n = 3$ (cubic B-spline) scaling function and wavelets. Notice that the wavelets are symmetric. It is well-known that one of the advantages of wavelet analysis is that the lower resolution coefficients can be calculated from the higher resolution coefficients by a tree structure called filter bank. By means of QMF, a fast algorithm for computing orthogonal wavelet transforms was proposed by Mallat in [40]. A number of operations which are proportional to the size of initial discrete data is needed in a Fast Wavelet Transform algorithm (FWT). Unser *et al.* [68] show that this fast algorithm can also be extended for nonorthogonal basis functions. The

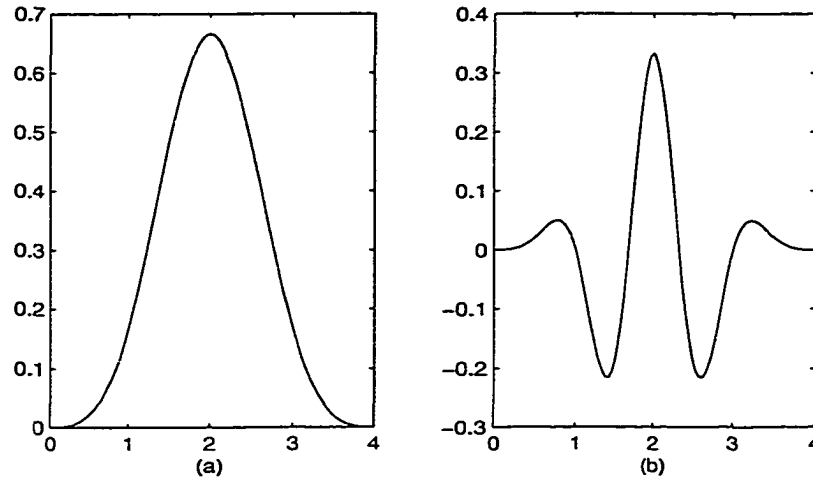


Figure 6.1: Cubic B-spline. (a) Scaling Function; (b) Wavelet Function

wavelet approximation coefficient $c(k)$ and detail coefficient $d(k)$ at level $(i + 1)$ can be obtained as follows:

$$c_{(i+1)}(k) = \langle \sum_{l=-\infty}^{+\infty} c_{(i)}(l) \varphi_{i,l}, \tilde{\varphi}_{i+1,k} \rangle = \frac{1}{\sqrt{2}} [h * c_{(i)}]_{\downarrow 2}(k) \quad (6.2)$$

$$d_{(i+1)}(k) = \langle \sum_{l=-\infty}^{+\infty} c_{(i)}(l) \varphi_{i,l}, \tilde{\Psi}_{i+1,k} \rangle = \frac{1}{\sqrt{2}} [g * c_{(i)}]_{\downarrow 2}(k) \quad (6.3)$$

where, φ , Ψ are scaling function and wavelet function, and $\tilde{\varphi}$, $\tilde{\Psi}$ are their duals, respectively.

6.4 Proposed Segmentation Algorithm

Several segmentation algorithms based on extracting features from wavelet transform domain have been proposed recently. In [51] a criterion which utilizes ratio of the mean energy in the four low-frequency channels to that in three middle-frequency channels is suggested to distinguish smooth images from textured images. Li *et al.* [37] observe that text images and picture images are different in the continuity of their distributions and the wavelet coefficients in the high frequency bands usually follow a Laplacian distribution. They thus define features depending on the shape of the

histogram of wavelet coefficients. In [36], a multiple scale approach is introduced and the performance of wavelet packet is investigated. The authors conclude that multiresolution properties of wavelet transforms are useful for applications such as segmentation, classification, and discrimination of textures.

Based on the observation that the energy of text texture, which is well preserved after wavelet transforms, could be used as a signature distinguishing itself from textures of background and smooth picture, a new algorithm is thus proposed for document image segmentation. The basic procedure for the proposed segmentation algorithm consists of four steps as follows:

1. Image Decomposition

As mentioned earlier, text texture features dense distribution of abrupt variation points, which can be detected by wavelet analysis. The step of image decomposition uses a biorthogonal wavelet transform, i.e., B-spline wavelets [71] to locate these singularities of original images.

2. Feature Extraction

Determining feature vectors is concerned with finding the best representation of original data so discrimination and classification of textures can be implemented based on the reduced data set. Obviously, the benefit of this step is the direct reduction of computational complexity for the remaining processing. In this algorithm, the two-dimensional discrete wavelet transform and cubic B-spline basis are used to decompose images into four frequency bands of LL, LH, HL, and HH [72]. Feature vectors are formed from HL, HH or LH, HH subbands because these high frequency subbands tend to amplify corresponding horizontal, vertical, and diagonal edges.

3. Classification

The K-means classification is widely used for grouping together those pixels in

the image which have similar features in the feature space. The major step of the K-means classification is to calculate the Euclidean distance function $d(C_k, X)$, where X represents the pixel's feature vector, and C_k is the cluster center, defined as the mean vector of all its members, $k \in \{1, 2, \dots, K\}$. Pixels are then relabeled according to their new nearest cluster centers. The process is repeated until a stable feature partition is obtained. In this work, the classification algorithm is either 2-means or 3-means classification depending on specific applications.

4. Post-Processing

After classification, mis-segmentation will usually occur due to the irregularity of input data or the poor quality of original images. Most of these errors can be eliminated by post-processing output of the previous step, which could be combining context information or checking pattern template to remove the errors.

The segmentation can be supervised or un-supervised. In the former, no *a priori* information about the image is required while in the latter, some assumptions about the texture are given. Most segmentation techniques rely on prior knowledge or assumptions about the generic document layout structure. In this work, we simply assume that the number of different textures in the processed image is given.

In practice, features are calculated from small regions which are obtained by dividing these subbands into blocks of size $M \times M$. Basically, the following three structural features are considered for the classification [71, 11]:

- Mean feature of the region (r, c) defined by,

$$m_{r,c} = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M c(i, j)$$

where, $c(i, j)$ is the wavelet coefficient at the location of (i, j) .

- Standard deviation of the region (r, c) computed as follows:

$$\sigma_{r,c} = \sqrt{\frac{1}{M^2 - 1} \sum_{i=1}^M \sum_{j=1}^M (c(i, j) - m)^2}$$

where, m is the mean feature of the region (r, c) and $c(i, j)$ is the wavelet coefficient at the location of (i, j) . Usually, for the background texture, the value of this feature tends to be zero while for the text texture, the value is much larger due to the fact that many abrupt changes exist. The gray-level values of smooth pictures change so slowly that their standard deviation are close to that of background.

- Energy in the (r, c) region defined by:

$$e_{r,c} = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M |c(i, j)|$$

The energy distribution in the four bands of wavelet transform domain is quite different depending on the type of the texture. It is observed that in the gray-level smoothly varied picture image most energies distribute at the low-frequency bands. However, in the gray-level sharply changed text image, large energies exist in both the low and high frequency bands [51].

Figure 6.2(b) shows an intermediate output image of k-means classification, in which some image parts are labeled erroneously as text or background regions in the output image. Therefore, the subsequent post-processing is necessary for good segmentation results. This step includes same texture merging and segmentation refining in two stages. First, for the output image of k-means classification, text blocks which equal or less than a threshold apart are merged into a continuous region of same texture, in either horizontal direction or vertical direction. The choice of thresholds are determined by document layout and how fine you want to segment the text blocks. The merging is implemented in horizontal and vertical directions,

respectively. The thresholds of 3 in horizontal and 2 in vertical directions are used in the following experiments. The technique we adopt is similar to RLS algorithm [74] except that the logical OR function on the two outputs is conducted to determine the functional regions. The incurred “noise”, for instance, some isolated blocks labeled as “text” existing in the picture (see Figure 6.2(b)), can be removed by incorporating more contextual evidence and merging surrounding blocks, which is conducted via a median filter. We have to further refine the result so that text information would not be lost around the segmentation edge areas. Dilation operation is introduced to add blocks to the boundaries of text regions.

6.5 Experimental Results

The set of test images consists of 100 typical document images scanned from newspapers, technical journals and books. Figure 6.3(a) gives a document image which is scanned at 150 dpi with a size of 829×687 . This test image includes three types of textures, namely, text, smooth picture and background. The decomposition results of a cubic B-spline wavelet are illustrated in Figure 6.3(b). We can see that edges of the original image are well preserved in wavelet transform domain, which reflects the good property of cubic B-spline wavelets. Figure 6.2 illustrates a segmentation example and shows some intermediate and final segmentation results obtained at different phases of the experiment. The output image from the previous stage automatically becomes the input data of next processing stage. The choosing of block size, a problem concerning segmentation accuracy and processing speed, will be discussed in detail in Section 5, while in this case, the block size of 3×3 has been selected. Original image in Figure 6.2(a) is the test image and inputted into our segmentation system. The output image of three-means classification is shown in Figure 6.2(b). To extract text regions from the image and for the convenience of processing, we only

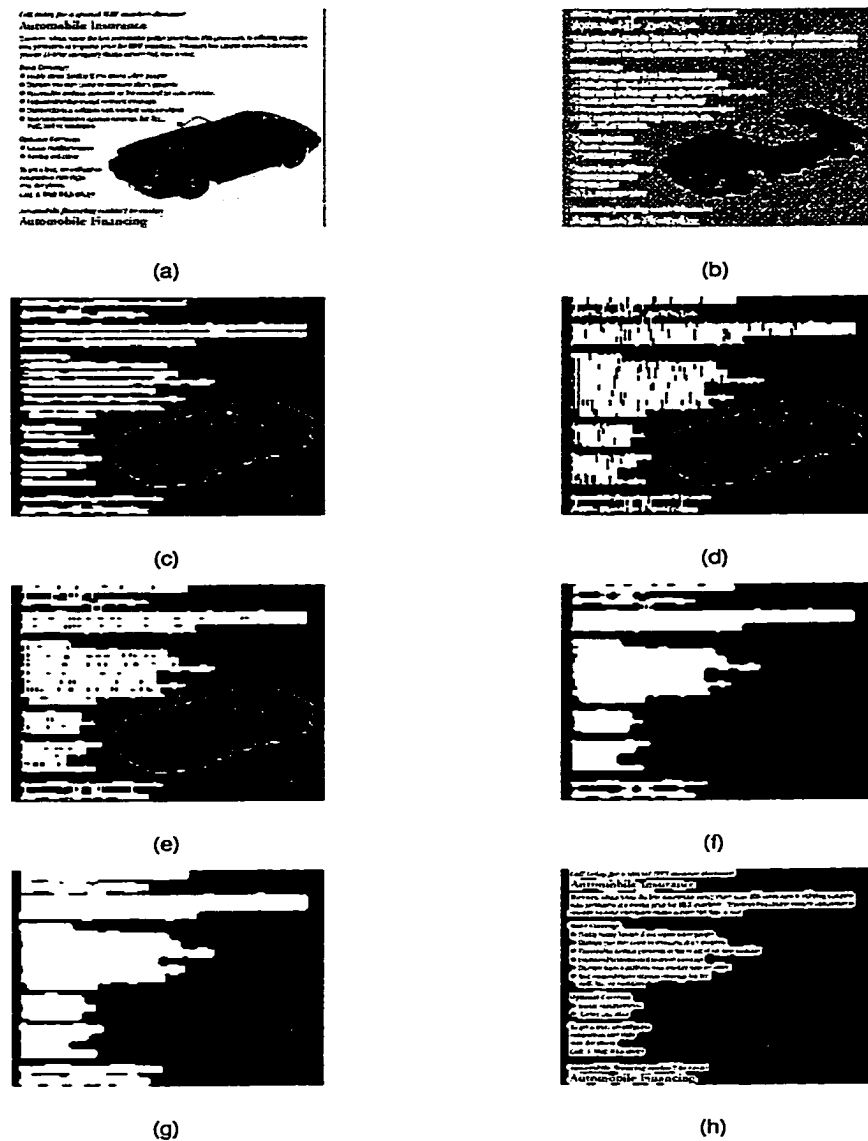
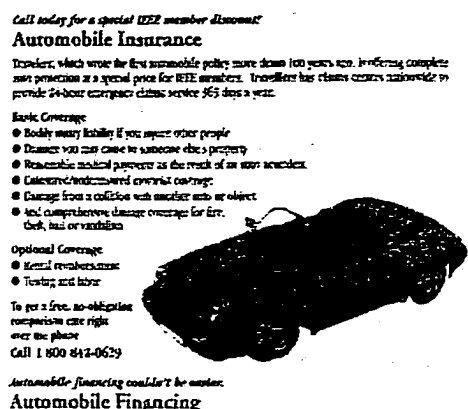


Figure 6.2: Performance of the Proposed Segmentation Method on a Document Image. (In this image, black, white and gray colors label the picture, text and background, respectively.) (a) Original Image; (b) Output Image of Three-means Segmentation; (c) Horizontal Merging; (d) Vertical Merging; (e) OR Operation on (c) and (d); (f) Median Filtering Result; (g) Dilation Processing; (h) Final Segmentation Result

distinguish text from nontext. The merging results in horizontal and vertical directions are presented in Figures 6.2(c) and (d) respectively. The logical OR operation



(a)



(b)

Figure 6.3: An Example of Cubic B-spline Wavelet Transform. (a) Original Image Scanned at 150dpi (829×687). (b) One-Level Decomposed Image Using a Cubic Spline Wavelet

(see Figure 6.2(e)) is applied on Figures 6.2(c) and 6.2(d). Then after median filtering (see Figure 6.2(f)) and dilation processing (see Figure 6.2(g)), the final segmentation result is presented in Figure 6.2(h). Another example is demonstrated in Figure 6.4. Many results demonstrate that text regions can be segmented correctly with high accuracy just after three-means classification and most picture regions also can be isolated out without error after post-processing. One of the frequently incurred errors in three-means classification is that some parts of pictures are misclassified as the background texture by the system.

The skew often occurs when an image is scanned. We know that many algorithms work poorly on skewed images (top-down based algorithms). To prove the robustness of our algorithm, we specially include an image with a skew of 10° (see Figure 6.5(a)). The algorithm is applied to this image and the segmentation result is shown in Figure 6.5(b), where text and picture regions can still be segmented. The example and other experiments performed later illustrate that this algorithm can

en inventors on the cutting ro

tion, but what about the "En-
gineers" at Moore School of
Electrical Engineering in
Philadelphia? Originally hired
to calculate shell and missile
trajectories for the U.S. War
Department in the 1940s, they
programmed the world's first
general purpose electronic com-
puter.

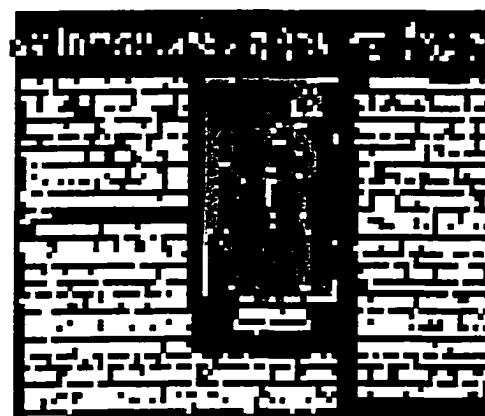
But decades before the Eni-
gma, women in the U.S. and
Europe were inventing calculat-
ing and adding machines. Stan-
ley cites, among many others,
Margaret K. Butler and Thelma
Estes as hardware pioneers and
Evelyn Berezin for her achieve-
ments in word processing. Ma-
chine Kumpf is noted for inventing the Katalavox, which
makes possible voice-controlled wheelchairs and micro-
scopes for delicate surgery, and Pat Werner is recognized for
her many inventions, including technology underlying the
Internet.



Augusta Ada
"Lady Lovelace"

RECOGNITION. A handful of
in the home, farms and in-
ventions. Some even gave
and other male faculty mem-
ber impossible to know how ex-
tension. "People think women
ley said, but in many ways
as with their husbands, ma-
ty William and Ann Maun-
lution than the electric mot-
port was developed in some
Disenport, according to Sta-
linked in some accounts.

Some inventions from the
domestic in nature or relatin-
unrealized. But if you really e-
inventions in a non-scienti-
ingenuous they are not only
standpoint, but from their s-
of life. When you get right
invented by Josephine G. C.



(a)

(b)

Figure 6.4: Another Example (756 × 595). (a) Original Image Scanned at 200 dpi; (b) Output Image of Three-means Classification

handle skewed images or slanted text strings. Experimental results also show that proposed algorithm can easily deal with handwritten text as well. This is illustrated in Figure 6.6, which consists of a handwritten text image with a resolution of 200 dpi and the segmentation result obtained after two-means classification.

Figure 6.7 plots the final cluster centers, i.e., the mean vectors for three images of Figure 6.2(a), Figure 6.5(a) (unrotated version) and its rotated version. The mean, standard deviation and energy features are obtained from first-level HL, LH and HH bands, respectively. The solid lines corresponding to the text texture have the highest mean values for almost nine features and thus can be used as a criterion to distinguish the text texture from other textures. Such high values also demonstrate high intensity variation. Similarly, lowest mean values corresponding to the background reveal low intensity variation in spatial domain. Just the very different mean centers for these three textures makes segmentation possible. Looking back at the original image of Figure 6.3(a), we see that the picture part displays significant high- and low-



(a)



(b)

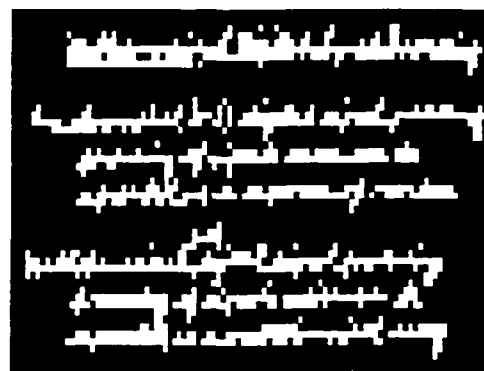
Figure 6.5: Performance of the Proposed Method on a Skewed Document Image. (a) Input Document Image Rotated by 10° , Scanned at 300 dpi (872×580); (b) Output Image of Three-means Classification

Multidimensional Digital signal Processing

One dimensional (1-D) digital signal processing
processing of signals which can be
represented as a sequence of numbers

Multidimensional ^(n-D) digital signal processing
processing of signals which can be
represented as a multidimensional array

(a)



(b)

Figure 6.6: Performance of the Proposed Segmentation Method on a Handwritten Document Image (775×411). (a) Input Handwritten Document Image Scanned at 200 dpi; (b) Output Image of Two-means Classification

intensity contrast due to the effects of newspaper texture in the course of digitization (i.e., degraded original image). Therefore many small edges of picture texture are

observed in the HH band which explains the apparent anomaly of having high standard deviation and energy values for this band in the plot of Figure 6.7(a). Despite the abnormal features of picture texture in the HH band, we still segment the image correctly. However, if insufficient distinguishing power is offered to the classifier, the

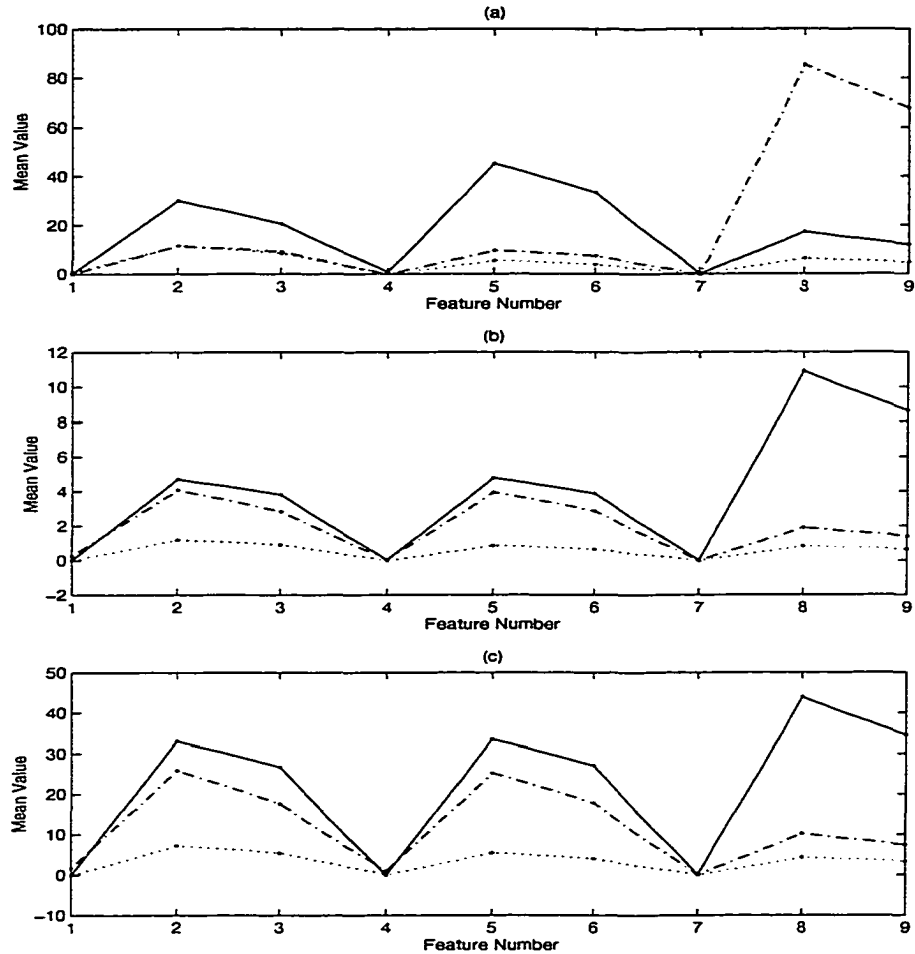


Figure 6.7: Cluster Centers of the Nine Features from Three Bands Corresponding to the Image in (a) Figure 6.3(a); (b) Figure 6.5(a) (Unrotated Version); (c) Figure 6.5(a). In These Plots, Text Class is Marked by Solid Line, Picture Class by Dash-dot Line, and Background Class by Dotted Line.

desired classification result may not be obtained. Figure 6.8 shows such an example where the subbands of HL and LH are used to classify three textures of text, smooth picture and background. Because the important features of high frequency band (HH)

are not included, the classifier could not distinguish text from smooth pictures and segmentation process thus fails at the stage of three-means classification.

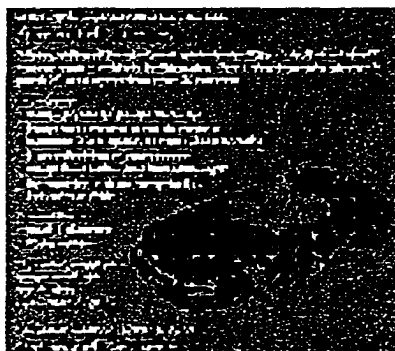


Figure 6.8: Segmentation Failure Due to Insufficient Distinguishing Power

Generally, this wavelet-based algorithm works on a variety of document images with different fonts, size and spacing as shown in the above examples. The edges in spatial domain can be found from the wavelet transform by identifying peaks at corresponding locations, i.e., finding large value features. The same segmentation result should be achieved if the different fonts provides roughly same amount of edges (this is usually true). For the different font size, edge density is still an important measurement for good segmentation results. If the font size is so large that distribution pattern of big wavelet coefficients for the text in the wavelet transform domain would be similar to that of smooth pictures, then only small portions of the text (edge area of the font) will be correctly classified. However, even this small portion could be automatically modified to the labeling of picture or background later in the post-processing since the remaining portion of the text (context) is classified into smooth pictures or background. For instance, very large titles of newspapers are hard to be distinguished from smooth picture images. The space between words or lines helps segmentation. Depending on applications, the importance for spacing varies.

If we want to detect how many lines an image consists of, the space between lines will make this work easy with no doubts. If we wish the text to appear in connected blocks, then the space plays little role because merging thresholds will be increased accordingly and cancel the effects of space except for margin areas.

Another advantage of the algorithms is that it can detect more than one picture image and handle some complex cases (see Figure 6.9). Figure 6.9(a) shows an original image. We would like to extract the text out from the image and 2-means classification is applied (see Figure 6.9(b)). The final segmentation result is shown in Figure 6.9(c).

There are two restrictions for this algorithm. For some degraded images, there will be more misclassification. For instance, if due to the bad quality of original document, text in a scanned digital image is blurred severely, then the segmentation system would have a tendency to recognize this part as image or background (see Figure 6.10). However, after some preprocessing such as contrast enhancement, most of misclassified areas can be removed as shown in Figure 6.9. Secondly, if the picture image contains unusually active information (i.e., edges is densely distributed), then corresponding part of picture image will be incorrectly segmented as texts. In addition, large bold characters are easy to be excluded from text regions by the systems as mentioned before.

6.6 Effects of System Parameters and Time Analysis

Notice that the proposed algorithm builds feature vectors based on first-level coefficient information. Will results be better if high-level band information is included? Also, if different wavelet functions are involved, it is desirable to know how the result changes. Therefore, in the following subsections, we will further compare and analyze the segmentation results by considering the following factors: decomposition

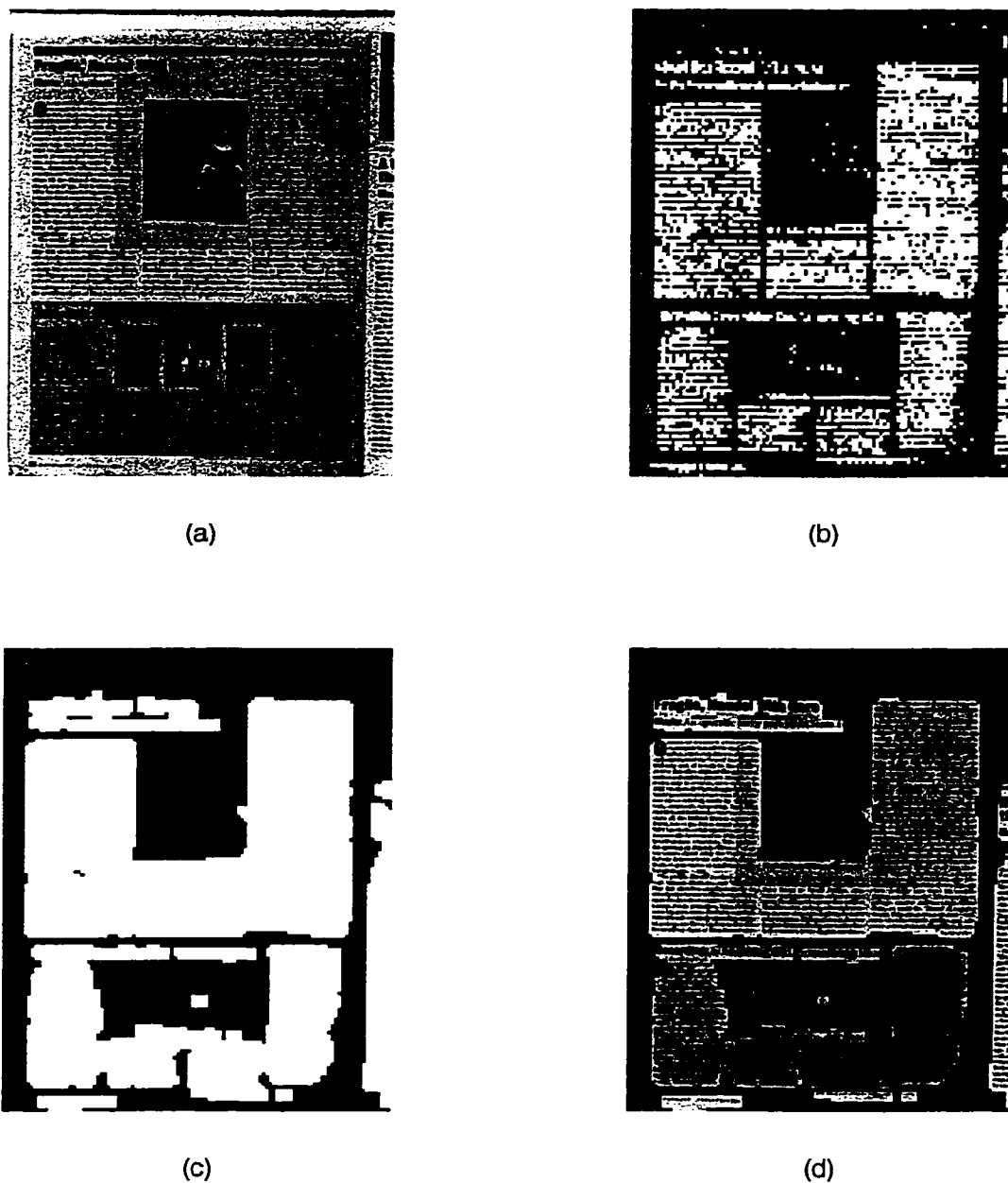
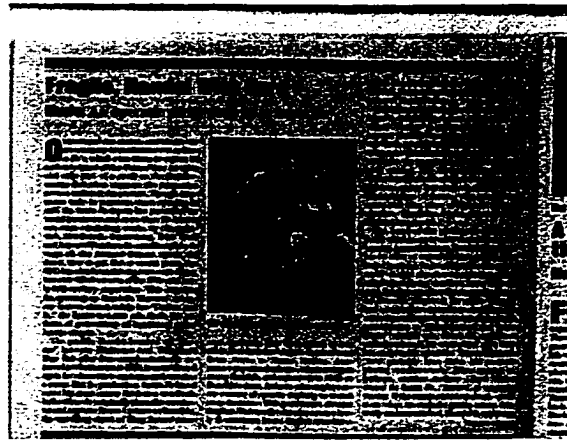
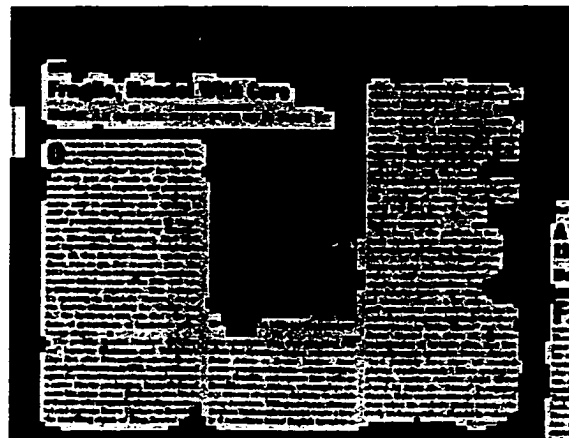


Figure 6.9: Segmentation of a Document Image with the Size of 791×1024 . (a) Original Image; (b) Output Image of Two-means Classification; (c) Segmentation Result; (d) Extracted Text Image

levels, frequency bands, feature selection, and wavelet functions. We measure the effectiveness of the proposed algorithm by introducing the concept of “Percent of



(a)



(b)

Figure 6.10: Processing of Some Degraded Images. (a) A Degraded Version of Figure 6.9(a); (b) Segmentation Result

segmentation error", denoted as P_e , and defined by,

$$P_e = \frac{\text{Number of misclassified pixels}}{\text{Total Number of pixels in the image}} \times 100\%$$

where, the number of misclassified pixels with respect to the human labeling for every output image of k-means classification is counted. Instead of analyzing the final segmentation results, the segmentation results at the stage of k-means classification

have been evaluated because the output of k-means classification directly reflects the effects and capabilities of the above four factors on image segmentation. Due to the fact that the background is easy to classify and it is rarely labeled as other textures by mistake, the misclassified pixels have been grouped into three cases: text region misclassified as picture, picture region misclassified as text and picture region misclassified as background. Without considerably influencing the P_e results, the situation that the text is incorrectly segmented as background has been ignored for the simplicity. In the following, we experiment with the 100 test images mentioned in Section 6.5 and the results listed in the tables are randomly picked from the outputs of the set of test images.

6.6.1 Decomposition Levels

In the previous sections, only the first-level wavelet coefficients have been analyzed to segment document images for the sake of simplicity. It has already been shown that segmentation could be achieved by taking advantage of the first-level coefficients. As mentioned earlier, an important property of wavelet transforms is providing a multiresolution analysis. Therefore, to investigate the significance of high level wavelet energy, we make 2-level decompositions of cubic B-spline wavelets on test images and then extract features (energy and standard deviation features) from HL, LH, and HH bands. It is found that the image could not be correctly segmented by processing second-level wavelet coefficients only. Table 6.1 presents the results by distinguishing three conditions, i.e., using first level only, second level only, and both first and second levels, where “F” denotes the failure of segmentation (see Figure 6.8 for such an example). By combining features from first- and second-level together and maintaining other conditions unchanged, a total of 12 features have been obtained. It is shown that segmentation results could not be considerably improved just by com-

binning more information from higher level bands. We notice that the percentage of segmentation error even increases for some images, which suggests that redundancy in feature vectors may increase uncertainty of classification process.

Table 6.1: The Percentage of Segmentation Error (P_e) Using different Level of Cubic B-spline Wavelets (%)

Level	No. of Feat.	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
1	6	0.92	0.57	1.69	1.78	1.20	4.09
2	6	F	F	F	F	F	F
1, 2	12	0.71	0.80	1.50	1.30	1.29	4.90

6.6.2 Frequency Bands

It is known that LH, HL and HH bands of wavelet transforms preserve horizontal-, vertical- and diagonal-direction edge information, respectively. We treat coefficients of these three bands equally before, but actually they play different roles in solving the segmentation task as shown in Table 6.2. In the experiments, we calculate three features of mean, standard deviation and energy for the frequency bands as specified in Table 6.2 after applying the cubic B-spline wavelet decomposition to the test image. The results show that the high frequency band band, HH, obviously has more discriminating power compared to other bands in the algorithm and the segmentation goal could not be achieved without information of this band. The segmentation result obtained using (LH, HH) or (HL, HH) is comparable to that of using three of them (HH, LH, HL) although for some images the segmentation error may slightly increase.

Table 6.2: The Percentage of Segmentation Error (P_e) Using Different Combination of Frequency Bands (%)

Band	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
LH, HL, HH	0.90	0.57	1.60	1.88	1.20	4.06
LH, HL	F	F	F	F	F	F
LH, HH	1.03	0.68	1.98	2.69	1.89	5.32
HL, HH	1.10	0.60	2.01	1.79	1.65	4.89

6.6.3 Feature Selections

Selecting a good set of features can not only reduce the dimension of feature vectors, but also speed up the classification and improve the segmentation accuracy. In the above experiments, mean, standard deviation and energy are used to form feature vectors for K-means classification. To investigate the contribution of every feature to segmentation, three experiments have been performed and in each of them, mean, standard deviation and energy features have been used respectively while other conditions have been kept the same. It is found that using standard deviation, or energy, or both together can successfully segment the test images without noticeably degrading the quality of segmentation results. But if only using mean feature, the segmentation result would be incorrect, which means that it is impossible to segment an image by adopting mean feature alone. This conclusion could also be drawn from Figure 6.7, which shows that mean features of each subband of images are about zero and thus contribute little or nothing to segmentation process.

6.6.4 Wavelet Functions

Haar wavelet, Daubechies wavelets D_2 , D_8 and D_{20} are used along with Cubic B-Spline wavelets in this experiment. Haar wavelet is the simplest possible orthogonal wavelet system and finds a lot of applications in image processing. Orthonormal

wavelets D_2 , D_8 and D_{20} are similar wavelets, but the first had fewest vanishing moments and the last has largest regularity. In this experiment, energy and standard deviation features are extracted from high frequency bands, i.e., HH, LH, and text and nontext, two types of textures, are distinguished using 2-means classification. To statistically compare the distinguishing power of different wavelets, the number of images, whose P_e by using one of the wavelets provided above falls in a certain given range, is recorded. Table 6.3 suggests that the number of images with low percentages of segmentation error using Cubic B-Spline is generally less than that of other wavelet analysis functions based on the set of 100 document images. Experimental results also show that cubic B-Spline has reduced percentage of segmentation error by around 0.1-0.5% for the experimental set of images. The other conclusion is that the longer analyzing functions (D_{20}) does not necessarily provide a more efficient representation for texture discrimination than the shorter one. On the contrary, the D_8 wavelets have shown better performance in terms of percentage of segmentation error than D_2 , and D_{20} .

Table 6.3: Segmentation Results Comparing the Performance of Different Wavelet Functions by the Number of Images with a P_e in the range of (a% - b%)

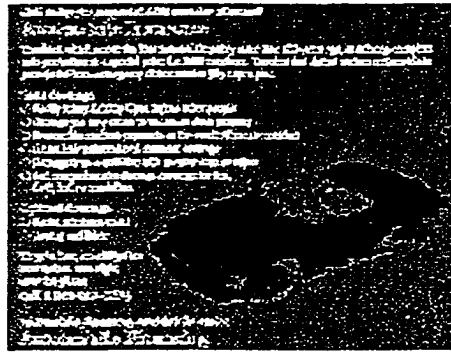
Wavelet Function	0-0.5	0.5-1	1-1.5	1.5-2.5	2.5-4	4-5
<i>Haar</i>	17	19	25	24	11	4
D_2	20	18	19	13	16	14
D_8	27	20	23	12	8	10
D_{20}	19	14	17	28	10	12
Cubic B-Spline	30	25	21	9	8	7

6.6.5 Time Analysis

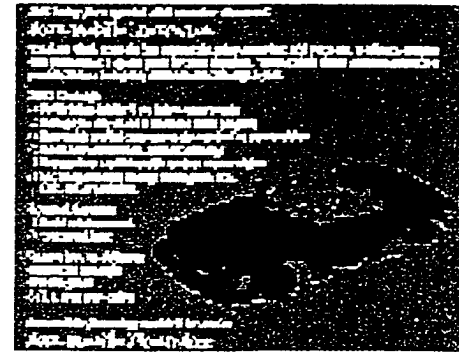
The current implementation of the proposed algorithm requires about 12 s (We mean CPU seconds in this context) to process a 512×512 image using MATLAB

software on a Pentium II processor (400Mhz). In a particular experiment, energy and standard deviation features are produced from HH and LH subbands using 3×3 block. Then 0.52 s and 1.70 s are used for wavelet transforms and feature extraction. The two-means clustering operation needs 9.83 s. Post-processing consumes 0.61 s. In the same software and hardware environments, segmentation algorithm based on Gabor filter [29] requires about 37 seconds of CPU time to process a 512×512 image. Our algorithm gains speed advantage while achieving comparable segmentation results. The speed-up of our algorithm is largely due to the reduction of volume of data set involved in computation. Before the feature extraction, the data is reduced by approximately a factor of 2 since two wavelet subbands are used. The data is further reduced significantly after feature extraction. The other reason for the speed-up is simplicity of the algorithm. Directly taking advantage of edge detection ability of wavelet transforms, the algorithm characterizes text regions by their regular edge distribution versus background or smooth picture images.

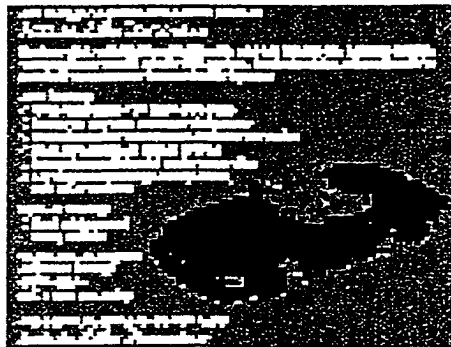
Because the algorithm is block-based, segmentation accuracy is largely determined by the size of blocks. The block size is closely connected with the amount of data to be processed and further affects the segmentation speed. Figures 6.11 and 6.12 show 3-means classification results and extracted text by applying different block sizes ranging from 2×2 to 16×16 . The criteria for choosing the window size is to find possible large size which helps to achieve fast processing while maintaining acceptable segmentation results. While the smaller block size means more accurate segmentation as shown in Figure 6.12, the disadvantage is the slow-down of processing. Figure 6.13 shows CPU time comparison of segmentation based on the same image and under same conditions by varying block size. The processing time used in the above experiment is decreasing dramatically as the block increases. By considering the factors of segmentation accuracy and time complexity together, the sizes of 3×3 and 4×4 are



(a)



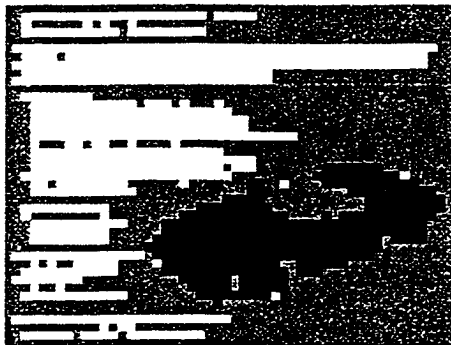
(b)



(c)



(d)



(e)



(f)

Figure 6.11: Output Images of 3-means Classification by Applying a Window Size of (a) 2×2 ; (b) 3×3 ; (c) 4×4 ; (d) 6×6 ; (e) 8×8 ; (f) 16×16 .

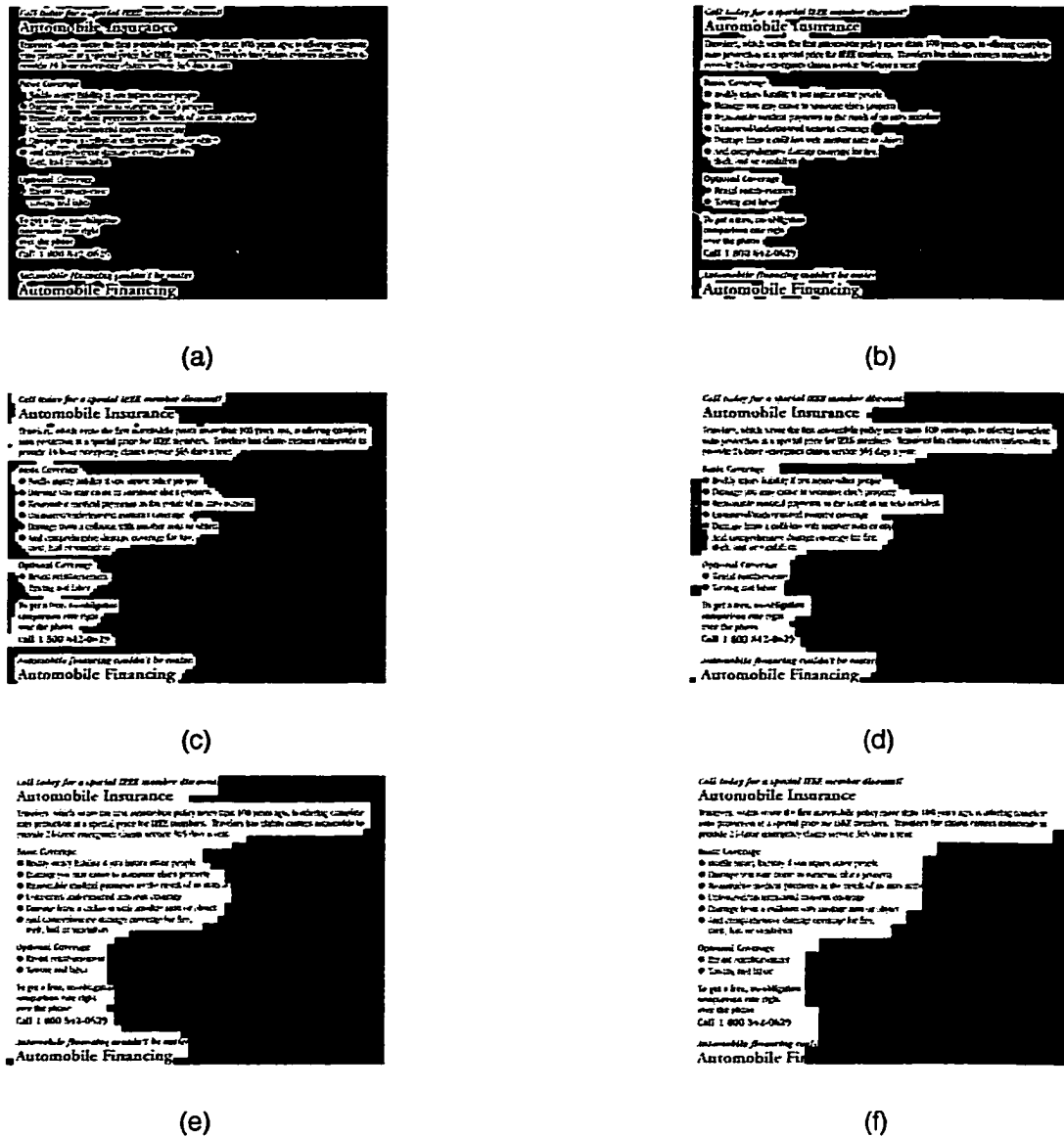


Figure 6.12: Extracted Text by Applying a Window Size of (a) 2×2 ; (b) 3×3 ; (c) 4×4 ; (d) 6×6 ; (e) 8×8 ; (e) 16×16 .

usually chosen and used in this research. Practically, if the objective of image processing is just to gain a coarse segmentation, then larger block size would be considered. Hence, depending on the application and the desired computational performance, the block size could be determined. Notice that as the block size increases, the time for

three-means classification is considerably reduced. Especially, about 1.5 seconds are needed for segmentation using larger block size 16×16 .

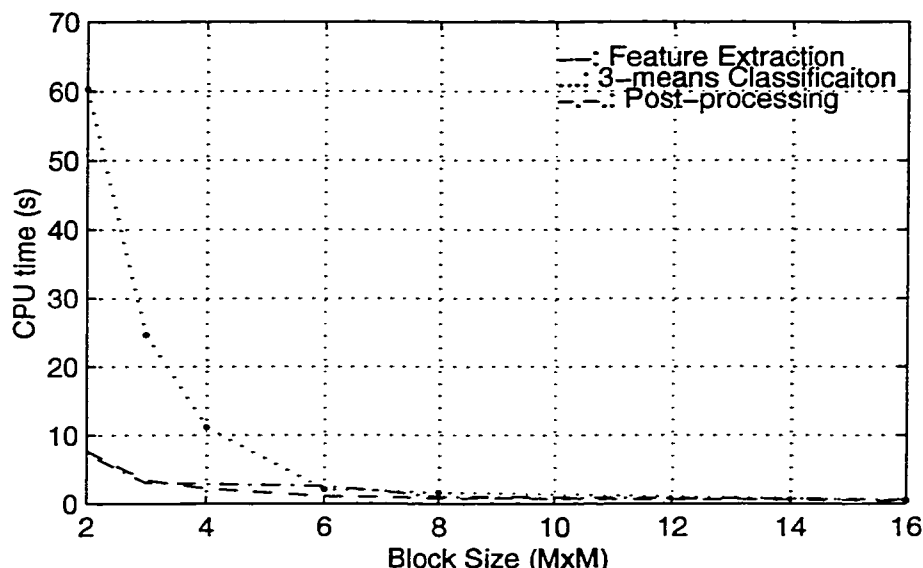


Figure 6.13: Time Comparison by Using Different Block Sizes

6.7 Summary and Conclusion

This work is related to the spline wavelet application on document segmentation. The possibilities and efficiency of such wavelet application were investigated. The capability of the B-spline transform to distinguish text and picture textures was analyzed to segment gazetteer images acquired from scanners.

The standard deviation and energy of the transformed coefficients are used to build feature vectors. Three-means classifier is employed to group pixels according to the homogeneity of the features. Post-processing of the classification refines this grouping and allows to extract text blocks from the images, which are of interests of OCR systems. The foregoing analysis suggests that the HH band of wavelet transform is very important for segmentation and the standard deviation and energy features

contains enough texture information which helps to classify the image. Such conclusions can help reduce the time complexity of the algorithm and improve the efficiency of segmentation as well. Different wavelets are analyzed in order to compare their contribution power to the feature vectors. The results have proved that B-spline is superior by its properties for the above mentioned application.

Various experimental results have revealed the abilities of this technique to cope with the different problems arising in the course of document segmentation. Skewed and handwritten documents are segmented correctly. In the case of low quality image, preprocessing is implemented first, i.e., contrast enhancement is applied, and then segmentation is performed based on the sharpened version.

The set of documents which is limited to the newspaper or magazine articles has been considered in the experiments. In practice, a variety of other classes such as graphics (chart, diagrams), tables, and formulas, etc, can be encountered in document images. The segmentation of such compound documents might involve more features for classification by exploiting information from both spectral and spatial domains. Future work will be concerned with extracting more efficient feature vectors from multi-level decompositions and investigating the pyramidal approach for achieving accurate segmentation.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In the previous chapters, document compression schemes have been reviewed and the existing techniques for image processing in compressed domain have been examined. The goal of this research is to analyze current compression schemes, in conjunction with image processing techniques, from a higher level and try to identify a synergy between them prior to the design of an efficient compression scheme. A new algorithm which facilitates image processing while maintaining reasonable memory space is thus developed for binary document images. Under the same principle, extending this work to other compressed domains such as the JBIG standard and wavelet-based compressions, some new algorithms have been proposed and investigated to address document image processing or some operations of it, in detail.

One of the basic problems that has been tackled is how to take advantage of structural information in compressed domain. This means that features should be characterized and organized based on coded data. The corresponding analysis for ITU fax standards and JBIG standard for document images has been provided in Chapter 2, 3, and 4. Generally, this research presents a clear picture for deciding how to handle compressed images by analyzing and capturing the existing characteristics. Other research on speeding image analysis, by processing compressed images directly or concurrently while decompressing, has also inspired this work.

The second basic problem considered was to investigate the possibility of modifying existing schemes so that images can be processed efficiently. Some research published in literature provided fast and useful algorithms for a specific compression

scheme, for instance, skew detection in G4 domain and rotation for run data. Obviously, there are restrictions for this type of research since the compression schemes are fixed and are impossible to adjust for fast image processing. As mentioned earlier, for most existing compression algorithms, signal distortion, bit rate, and coding complexity are major issues to be considered. This kind of compressed-domain processing is limited by unorganized coding structures. Since processing speed is an important factor for a competitive document processing system, it may be beneficial to design some new document compression algorithms or modify current coding schemes, so that a balance between compression rate and compressed-domain accessibility is achieved. The novelty of this work lies in implementing extensive research to investigate how a coding scheme could be constructed to benefit accessibility of image data.

The final basic problem addressed was how to conduct image analysis using the emerging technology of wavelet analysis. Document images are distinguished from other types of images by offering densely distributed singularities and abrupt variation points in the text part of images. Wavelet transforms have shown the abilities to detect these fast changing points. To conduct this analysis effectively, the feature vectors have been constructed based on wavelet coefficients, which mean reduced data sets, compared to the original images. An efficient classifier, K-means classifier, has been applied to wavelet high frequency bands of document images so that different textures of document images can be recognized in an efficient way.

This dissertation has made a number of contributions to the area of compressed-domain processing, which are summarized in the following section. Further, the dissertation will be concluded by suggesting several avenues for future work based on this research.

7.1 Summary of the Contributions

- A new compression scheme, MG4, for binary document images has been designed, tested, and verified. In addition to being competitive with existing compression standards, the new scheme allows for rapid manipulation of compressed binary data. The algorithms developed in this research can be utilized by a new generation of document image analysis systems.
- For bilevel images, the rotation algorithm requires the conversion of horizontal run-length data into vertical run-length data. The operation of conversion on compressed data is still an open problem, to which a novel solution has been provided in this work. The new algorithm is based on the relative locations of the coding run and the reference run. Since the process of conversion is operated on run data instead of on single pixel, the processing speed for image rotation would be considerably improved.
- A general approach and the implementation of some compressed-domain image processing techniques for JBIG compression algorithm has been provided. In particular, some operations such as corner detection, connected component extraction, and page segmentation are conducted by applying suggested algorithms, which can be further extended to other compression methods with certain identical features with the JBIG algorithm.
- Applying wavelet analysis to the field of document segmentation, three types of texture information, i.e., text, picture and background, can be precisely recognized. Cubic B-spline wavelets are chosen as a transformation tool because of their useful properties for texture recognition. The research further suggests that the HH band of wavelet transform has a special contribution to segmentation compared to other frequency bands. Experimentally, the importance of

wavelet selection, feature vector formation, wavelet band selection and decomposition level selection on segmentation have been examined and discussed. Segmentation experiments indicate that this approach is advantageous over other traditional methods in terms of memory space and processing speed.

This research is significant because it provides a comprehensive study of image processing techniques in compressed domain, especially for bilevel document images. In addition to describing a framework for current research in this area, this work provides an understanding of factors that affect the desired performance. An understanding of these issues will be useful to build more efficient and flexible compression and processing systems, which eventually make data transmission, storage and access more rapid, convenient and reliable.

7.2 Future Work

Possible future directions of this research are as follows:

- A. The new coding scheme proposed in this dissertation is directly applicable to document images. It is believed that the algorithm can be greatly enhanced if further investigated. Due to resource and time constraints, experiments were carried out using limited images. Improving the algorithm to increase the compression ratio and implementing it on a more extensive image data set would be beneficial.
- B. The efficiency of the proposed wavelet-based algorithm on images containing text and pictures was investigated. In practice, a variety of other features, such as graphics, and tables, can be encountered in document images. Therefore, a more robust and accurate segmentation algorithm is yet to be developed. The future work will be concerned with extracting more efficient feature vectors

from multi-level decompositions for fast segmentation, and the resulting effect on document segmentation should be investigated.

- C. Wavelet analysis found a lot of applications in images processing. But applying this technology to document images is a relatively new topic. This dissertation discussed document segmentation and confined the current investigation mainly to text extraction. Applying the concepts presented to other document image operations such as image similarity estimation and difficult text segmentation, requires additional research.
- D. Most of the discussions and evaluations assumed normal images with no noise or degradation. However, the real image can be skewed or smeared and characters can be broken. How to adapt current proposed algorithms to fit in this category of images would be interesting future work.

This research has addressed the problem of providing techniques for compressed document images. While this dissertation outlined and evaluated a solution to the problem of facilitating image transmission and processing, a number of issues clearly remain unanswered. It is hoped that this research will stimulate additional interest in the research area of compressed-domain processing.

BIBLIOGRAPHY

- [1] N Amamota, S Torigoe, Block Segmentation and Text Area Extraction of Vertically/Horizontally Written Document, IEEE, pp. 739-742, 1993.
- [2] A. Amir and G. M. Landau and U. Vishkin, Efficient Matching with Scale, *Proceedings of 1st ACM-SIAM Symposium on Discrete Algorithm*, pp. 344-357, 1990.
- [3] A. Amir and G. Benson, Efficient Two-Dimensional Compressed Matching, *Proceedings of Data Compression Conference 1992*, pp. 279-288, 1992.
- [4] H. S. Baird, The Skew Angle of Printed Documents, in *Proceedings of SPSE Symposium on Hybrid Imaging Systems*, Rochester, N.Y., pp. 21-24, 1987.
- [5] H.S.Baird, H.Bunke, K.Yamamoto, Structured Document Image Analysis, Springer-Verlag, Berlin Heidelberg, New York, 1992.
- [6] N. Bartneck, Image Analysis Based on Image Description Graphs with Contour Coded Objects, IEEE, pp. 1108-1110, 1984.
- [7] N. Bartneck, A General Data Structure for Image Analysis Based on a Description of Connected Components, *Computing*, 42/1, pp. 17-34, 1989.
- [8] D.S.Bloomberg, G.E.Kopec, and L. Dasari, Measuring document image skew and orientation, *SPIE Proceedings Series: Document Recognition II*, pp. 302-316, San Jose, CA, 1995.

- [9] C. Sidney Burrus, Ranesh A. Gopinath, and Haitao Guo, Introduction to wavelets and wavelet transforms, *Prentice Hall*, Upper Saddle River, New Jersey, 1998.
- [10] K. R. Caastleman, Digital Image Processing, Prentice Hall, 1996.
- [11] J.L.Chen and A.Kundu, Rotation and Gray Scale Transform Invariant Texture Identification Using Wavelet Decomposition and Hidden Markov Model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, February, 1994.
- [12] CCITT Recommendation T.4, Standardization of Group 3 Facsimile Apparatus for Document Transmission, *Terminal Equipment and Protocols for the Telematic Services*, Vol. VII, Fascicle VII.3, Geneva 1989.
- [13] CCITT Recommendation T.6, Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, *Terminal Equipment and Protocols for the Telematic Services*, Vol. VII, Fascicle VII.3, Geneva, 1989.
- [14] T. Chang, Kuo, CC-J, Texture analysis and classification with tree-structured wavelet transform, *IEEE Transactions on Image Processing*, 2(4), pp.429-441, 1993.
- [15] S-F. Chang, Exploring functionalities in the compressed image/video domain, *Computing Surveys*, pp. 573-575, 24(4), 1995.
- [16] S. Chen, R. Haralick, and I. Philillips, Automatic text skew estimation in document images, in *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 1153-1156, Montreal, Canada, 1995.
- [17] Ingrid Daubechies, Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, 41, pp. 909-996, November, 1998.

- [18] RA. DeVore, B. Lucier, and Z. Yang, Feature extraction in digital mammography, *Wavelets in Medicine and Biology*, CRC Press, Ed. Aldroubi A, Unser M, pp. 145-161, 1996.
- [19] L. O’Gorman, R. Kasturi, Document Image Analysis, *IEEE Computer Society Press*, Las Almitos, California, 1994.
- [20] OD. Faugeras, and WK. Pratt, Decorrelation methods of texture feature extraction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(7), pp. 323-332, 1980.
- [21] J. Froment, S. Mallat, Second generation compact image coding with wavelets, *Wavelet Analysis and its Applications*, 2, Academic Press, 1992.
- [22] R.M. Haralick, Image texture analysis, New-York, Plenum, 1981.
- [23] M. Hase and Y. Hoshino, Segmentation method of document images by two-dimensional fourier transformation, *Systems and Computers in Japan*, 16(3), 1985.
- [24] S. C. Hinds, J. L. Fisher, and D. P. D’Amato, A Document Skew Detection Method Using Run-length Encoding And The Hough Transform, in *Proceedings of 10th Int’l Conf. Pattern Recognition*, pp. 464-468, 1990.
- [25] T. Huang, Coding of Two-Tone Images , *IEEE Transactions on Communication*, Vol. COM-25(11), pp. 1406-1424, November, 1977.
- [26] J. J. Hull and J. F. Cullen, Document Image Similarity and Equivalence Detection, in *Proceedings Of 4th International conference on Document Analysis and Recognition*, Ulm, Germany, pp. 308-312, August 18-20, 1997.

- [27] International Standard ISO/IEC 11544:1993 and ITU-T Recommendation T.82 (1993) Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression.
- [28] AK. Jain, Fundamentals of digital image processing, Prentice-Hall, 1989.
- [29] A. K. Jain, S. Bhattacharjee, Text Segmentation Using Gabor Filters for Automatic Document Processing, *Machine Vision and Applications*, 5, pp. 169-184, 1992.
- [30] CE. Jacobs, A. Finkelstein, D. Salesin, Fast multiresolution image querying, in *Proceedings of SIGGRAPH 95*, LA, pp.277-286, 1995.
- [31] J. Kanai, AD. Bagdanov, Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images, *International Journal of Document Analysis and Recognition*, Springer-Verlag, 1:1:pp. 43-51, 1997.
- [32] W. Kou, Digital Image Compression Algorithms and Standards, *Boston: Kluwer Academic Publishers*, 1995.
- [33] RP Kruger, WB Thompson, and AF Turner, Computer diagnosis of pneumococcosis, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-4, 1(1) , pp. 40-49, 1974.
- [34] M. Kuhn, JBIG-KIT V1.0 Available from *ftp : //ftp.informatik.uni - erlangen.de/pub/doc/ISO/JBIG/*
- [35] O-J. Kwon, and T. Chellapa, Region adaptive subband image coding, *IEEE Transactions On Image Processing*, 7(5), pp.632-645, 1998.
- [36] A. Laine, F. Jian, Texture Classification by Wavelet Packet Signatures, *IEEE Transactions on Image Processing*, 15(11), 1186-1191, 1993.

- [37] Jia Li, Robert M. Gray, Text and Picture Segmentation by the Distribution Analysis of Wavelet Coefficients, *International Conference on Image Processing*, Chicago, USA, 1998.
- [38] C. Maa., Identifying the existence of bar codes in compressed images. *CVGIP: Graphical Models and Image Processing*, Vol. 56, No. 4, pp. 352-356, July, 1994.
- [39] S. Mallat, A theory of multiresolution signal decomposition: the wavelet representation, *IEEE Pattern Anal. Machine Intell.*, 11(7), pp. 674-693, 1989.
- [40] S. G. Mallat, Multifrequency Channel Decompositions of Images and Wavelet Models, *IEEE Transaction on Image Processing*, 37(12), pp. 2091-2110, 1989.
- [41] S. Mallat, and S. Zhong, Characterization of signals from multiscale edges, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 14(7), pp. 710-732, 1992.
- [42] B.S. Manjunath, and W.Y. Ma, Texture Features for Browsing and Retrieval of Image Data, *IEEE Trans. On PAMI*, vol. 18, no. 8, pp. 837-841, August 1996.
- [43] G. F. Mclean, Codebook Edge Detection, *Graphical Models and Image Processing*, Vol. 55, No. 1, pp. 48-57, Jan 1993.
- [44] R. Mehrotra, JE. Gary, Similar shape retrieval in shape data management, *IEEE Computer Magazine*, 28(9), pp. 57-62, 1995.
- [45] G. Nagy, S. C. Seth, and S. D. Stoddard, Document Analysis With an Expert System, in *Proceedings of ACM Conference on Document Processing Systems*, pp. 169-176, 1988.

- [46] Y. Nakamura, T. Yoshida, Learning two-dimensional shapes using wavelets local extrema, in *Proceedings of 12-th IARP International Conference on Pattern Recognition*, 11, pp. 48-52, 1994.
- [47] T. Pavlidis, Algorithms for Graphics and Image Processing, *Computer Science Press, Rockville, MD*, 1982.
- [48] T. Pavlidis, A Hybrid Vectorization Algorithm, *Proceeding of the Seventh International Conference on Pattern Recognition*, pp. 490-492, Montreal, 1984.
- [49] T. Pavlidis, A Vectorizer and Feature Extractor for Document Recognition, *Computer Vision, Graphics, and Image Processing*, vol. 35, No. 1, pp. 111-127, July, 1986.
- [50] PW. Picard, and TP. Minka, Vision texture for annotation, *Multimedia Systems*, Spriner-Verlag, 3, 1995.
- [51] R. Porter, N. Canagarajah, A robust automatic clustering scheme for image segmentation using wavelets, *IEEE Transactions on Image Processing*, 5(4), pp. 662-665, 1996.
- [52] WK. Pratt, Digital Image Processing, Second Edition, John Wiley & Sons, 1991
- [53] RM. Rao, and AS. Bopardikar, Wavelet transforms: introduction to theory and applications, Addison-Wesley, 1998.
- [54] T. Renden, J. Husoy, Image content search by color and texture properties, *International Conference. On Image Processing I*, pp. 580-586, 1997.
- [55] C. Ronse and P.A. Devijver, Connected Components in Binary Images: The Detection Problem, *Research Studies*, Letchworth, Herts, England, 1984.

- [56] A. Rosenfeld, and E.B. Troy E.B, Visual texture analysis, *Proc. UMR-Mervin J. Kelley Communications Conference, University of Missouri-Rolla, MO, Sect.10-1*, Oct 1970.
- [57] K. Sayood, Introduction to Data Compression, Morgan Kaufmann Publishers, Inc., 1996.
- [58] I. J. Schoenberg, Contribution to the problem of approximations of equidistant data by analytic function, *Quart. Appl. Math*, 4, pp. 45-99, pp. 112-141, 1946.
- [59] L. L. Schumakekr, Spline Functions: Basic Theory”, Wiley, New York, 1981.
- [60] Y. Shima, S. Kashioka, and J. Higashino, A High-Speed Rotation Method for Binary Images Based on CoordinateOperation of Run Data, *Systems and Computers in Japan*, Vol.20, No.6, pp. 91-102, 1989.
- [61] Y. Shima, T. Murakami, M. Koga, H. Yashiro., and H. Fujisawa, A High Speed Algorithm for Propagation-type Labeling based on locking Sorting of Runs in Binary Images, in *Proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, June 16-21, pp. 655-658, 1990.
- [62] R. Smith, A simple and efficient skew detection algorithm via text row accumulation, in *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 1145-1148, Montreal, Canada, 1995.
- [63] A. L. Spitz, Skew Determination in CCITT Group 4 Compressed Document Images, *Proceedings of the First Symposium on Document Analysis and Information Retrieval*, pp. 11-25, 1992.
- [64] A. L. Spitz, Logotype Detection in Compressed Images Using Alignment Signatures, in *Proceedings of the Fifth Symposium on Document Analysis and Information Retrieve-Al*, pp. 303-310, 1996.

- [65] HS. Stone, and C-S. Li, Image matching by means of intensity and texture matching in *the Fourier domain*, in Proceedings of SPIE on Image and Video Databases, San Jose, CA, 1996.
- [66] RN. Sutton, and EL. Hall, Texture measures for automatic classification of pulmonary diseases, *IEEE Transactions on Computers*, C-21(7), pp. 667-676, 1972.
- [67] T. Huang, Coding of Two-Tone Images ,*IEEE Transactions on Communication*, Vol. COM-25(11), pp. 1406-1424, November 1977.
- [68] M. Unser, A. Aldroubi, Polynomial Splines and Wavelets - A Signal Processing Perspective, *Wavelets-A Tutorial in Theory and Applications*, pp. 91-122, Academic Press, Boston, 1992.
- [69] M. Unser, A. Aldroubi, M. Eden, On the Asympotitic Convergence of B-Spline Wavelets to Gabor Functions, *IEEE Transactions on Information Theory*, 38(2), pp. 864-872, 1992.
- [70] M. Unser, A. Aldroubi, M. Eden, family of polynomial spline wavelet transform, *Signal Processing* 30, pp. 141-162, 1993.
- [71] M. Unser, Texture Classification and Segmentation Using Wavelet Frames, *IEEE Transactions on Image Processing*, 4(11), pp. 1549-1560, 1995.
- [72] Martin Vetterli, *Wavelets and Subbabd Coding*, Prentice Hall, N.J., 1995.
- [73] F. M. Wahl, K. Y. Wong, and R.G.Casey, Block Segmentation and Text Extraction in Mixed Text/Image Documents, *Computer Graphics and Image Processing*, Vol. 20, No. 4, pp. 375-390, December, 1982.

- [74] D. Wang and SN Srihari Classification of Newspaper Image Blocks Using Texture Analysis, *Computer Vision, Graphics, and Image Processing*, 47, pp. 327-352, 1989.
- [75] J. Weiser, A. Pinz, Layout and Analysis: Finding Text, Titles, and Photos in *Digital Images of New*, 1993.
- [76] I. H. Witten, A. Moffat, and T. C. Bell, Managing Gigabytes, Van Nostrand Reinhold, 1994.
- [77] Y. Ishitani, Document skew detection based on local region complexity, in *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 49-52, Los Alamitos, CA, 1993.

VITA

Graduate College
University of Nevada, Las Vegas

Shulan Deng

Local Address:

1600 E. Rochelle Avenue, #30
Las Vegas, NV 89119

Home Address:

Shanghai Road, 2 – 2 – 12
Nanchang, Jiangxi, China, 330029

Degrees:

Bachelor of Engineering, Electrical Engineering, 1991
Beijing University of Aeronautics and Astronautics

Master of Science, Electrical Engineering, 1994
Beijing University of Aeronautics and Astronautics

Dissertation Title: Techniques for Document Image Processing in Compressed Domain

Dissertation Examination Committee:

Chairperson, Dr. Shahram Latifi, Ph.D.
Committee Member, Dr. Eugene McGaugh, Ph.D.
Committee Member, Dr. Henry Selvaraj, Ph.D.
Committee Member, Dr. Kazem Taghva, Ph.D.
Graduate Faculty Representative, Dr. Zhiyong Wang, Ph.D.