1996

Restoration and segmentation of machine printed documents.

Su. Liang
University of Windsor

Restoration and Segmentation of Machine Printed Documents

by

Su Liang

A Dissertation
Submitted to the Faculty of Graduate Studies and Research
Through the Department of Electrical Engineering
in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada
1996

0-612-30285-7

ABSTRACT

The Restoration and Segmentation of Machine Printed Documents

by

Su Liang

OCR (Optical Character Recognition) has been confronted with the problems of recognizing degraded document images such as text overlapping with non-text symbols, touching characters, etc. The recognition rate for those degraded document images will become unacceptable or completely fail if pre-processing algorithms are not performed before segmentation recognition algorithms are applied. Therefore, the principle objective of this thesis is to develop effective algorithms for tackling those problems in the field of document analysis. We focus our efforts only on the following aspects:

1. A morphological approach has been developed to extract text strings from regular periodic overlapping text/background images, since most OCR systems can only read traditional characters: black characters on a uniform white background, or vice versa. The proposed algorithms that perform text character extraction accommodate document images that contain various kinds of periodically distributed background symbols. The underlying strategy of the algorithms is to maximize background component removal while minimizing the shape distortion of text characters by using appropriate morphological operations.

2. Real-world images, which are frequently degraded due to human induced interference strokes, are inadequate for processing by document analysis systems. In order to process those document images, containing handwritten interference marks which do not possess the periodical property, a new algorithm combining a thinning technique and orientation attributes of connected components has been developed to effectively segment handwritten interference strokes. Morphological operations based on orientation map and skeleton images are used to successfully prevent the "flooding water" effect of conventional morphological operations for removing interference strokes.

3. Segmenting a word into its character components is one of the most critical steps in document recognition systems. Any failures and errors in this segmentation step can lead to a critical loss of information from documents. In this thesis, we propose new algorithms for resolving the ambiguities in segmenting touching characters. A modified segmentation discrimination function is presented for segmenting touching characters based on the pixel projection and profile projection. A dynamic recursive segmentation algorithm has been developed to effectively search for correct cutting points in touching character components. Based on 12 pages of "NEWSLINE", the University of Windsor's publication, a 99.6% character recognition accuracy has been achieved.

To my wife Qiu-Ping LI, without whom I would have quit my Ph.D program.

To my parents: Ying-Hua LIANG and Ying JIN

and

To my parents-in-law: Zhong-Zhen LI and Mie-Mie WANG

ACKNOWLEDGMENTS

I am sincerely grateful to my advisors Dr. M. Ahmadi and Dr. M. Shridhar for valuable guidance, and encouragement throughout the progress of the dissertation, and for financial support through the Natural Science and Engineering Research Council (NSERC) of Canada, and MicroNet. Thanks are also due to the department readers, Dr. G.A. Jullien and Dr. J. Soltis, and to the outside department reader, Dr. W. North, of my doctoral committee, for constructive comments. I also thank the external examiner, Dr. A. K. Jain of the Michigan State University for useful comments, Dr. G.R.G. Raju, the head of the department for his encouragement during my Ph.D program. I would like to acknowledge the OGS committee for awarding me the OGS graduate scholarship for one year, and the U. of Windsor for awarding me the U. of Windsor Tuition Scholarship for one year.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF SYMBOLS

$\oplus$	Dilation
$\ominus$	Erosion
$\circ$	Opening
$\bullet$	Closing
$\supseteq$	Contains
$\subseteq$	Is contained by
$\cap$	Intersection
$\cup$	Union
$\in$	Belongs to
$\forall$	For all
[•••]:	At least one of the points inside the square bracket must be 1
↓ :	The origin location of the structuring element
* :	Don't care

Chapter I

Introduction

Today the number of publications such as newspapers, journals, magazines, and various sorts of desktop publishing materials is dramatically increasing. As a result, it has become too costly to handle massive amounts of documented information by traditional manual keying without highly accurate document recognition systems. Recent advanced technologies have brought significant improvement to OCR (Optical Character Recognition) in terms of recognition accuracy and noise tolerance, and made it a viable alternative to manual data entry for a wide range of documents [1]-[12], although it is a difficult task to make document recognition systems capable of identifying thousands of typefaces and recognizing degraded character images in a practical use level. We predict that, in the coming years, research on document analysis and recognition will play an important role in the area of data automation and future generations of OCR systems that require neither user training nor manual intervention will be developed eventually.

Document analysis systems aim at the transformation of any printed information presented on paper into an equivalent symbolic representation that is accessible and recognizable to computer-based information processing systems. In this introduction, the state of the art in document analysis systems and the different approaches used by researchers in tackling challenging problems is addressed in the following sections.

1.1 State of the Art -- Document Analysis

Document analysis is used to extract a geometric structure of documents, and then generate editable ASCII text, encoded graphics and half-tone images in documents such as newspapers, technical reports, etc. Therefore, a document image is classified into several blocks, which represent coherent components of a document (such as text lines, headlines, graphics, etc.), by using the information regarding specific formats and approaches. The major components of a document-analysis system are illustrated in Figure 1.1. The utilities and functionality of these components will now be described :

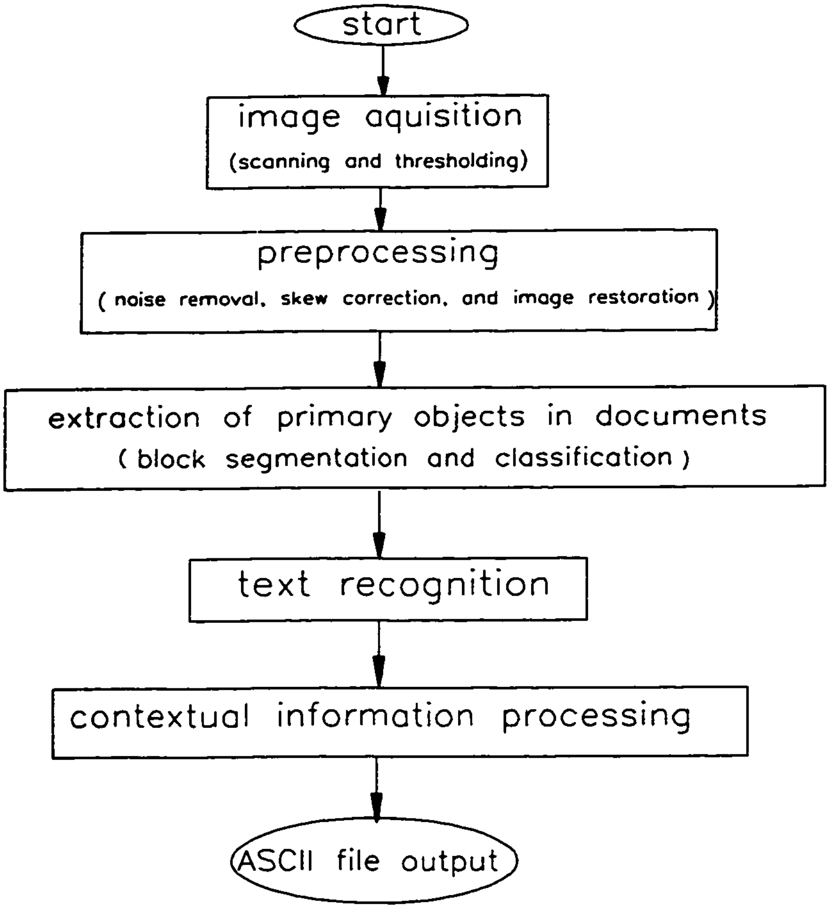

start
image aquisition
(scanning and thresholding)
preprocessing
(noise removal, skew correction, and image restoration)
extraction of primary objects in documents
(block segmentation and classification)
text recognition
contextual information processing
ASCII file output

Figure 1.1 The major components of document analysis systems

(i) Image Acquisition

A paper document can be converted into a bit-map format by an optical scanner, yielding what is referred to as a document image. The first step in document analysis is to acquire a digitized raster image of the document using a suitable scanning system. The resolution for scanned images must be carefully chosen since this parameter can strongly affect the analysis results [13]. The obtained grey level image is then thresholded to a bi-level format. Thresholding techniques have been widely investigated by many researchers in the areas of image processing and document image processing [13]-[18].

(ii) Preprocessing : Noise Removal, Skew Correction and Document Image Restoration

Preprocessing can be defined as any procedures which transform the presented documents into formats suitable to the document analysis system for further segmentation and recognition. For general document images composed of thousands cf characters in high or medium quality printing, the preprocessing mainly refers to the process for removing the noise caused by the digitizing and binarizing process. A typical operation applied to binary raster images in an efficient way is the morphological opening operation, by which all connected components below certain size are removed and larger objects remain substantially unchanged [19][20]. For degradation that occurs in documents while photocopying or scanning thick or bound documents, the preprocessing will provide an inverse perspective process, in which the distortion mechanism is modelled by carefully studying the underlying perspective geometry of the optical system of photocopiers and scanners [21][22]. For a document image scanned at a slant, the system should be equipped with a function for skew detection and correction since the skew effect may break down the performance of any document analysis systems and it could make any segmentation and recognition algorithms ineffective [23][24][25]. The preprocessing is also required to extract text parts overlapping with non-text parts in order to handle complex documents containing tables, forms, advertisement articles, engineering drawings, etc. Characters touching lines, background symbols, or hand-written markings frequently appear in applications such as address reading or form reading, where, for example, the address or the check amount is written on a given baseline. The algorithms to remove those non-text components are necessary for high performance in current document analysis systems for these applications [26]-[33].

(iii) Extracting Primary Objects of Documents : Block Segmentation and Classification

Page segmentation involves determining the structure of the document images, separating pictures from text, and partitioning the text into columns, lines, words, and characters. Algorithms for page segmentation include projection profile cuts [23], run-length smearing [34], connected component analysis [30], and segmentation by white streams [35].

There are two approaches for extracting a geometry structure [9]: (1) the top-down knowledge based approach divides the document into major regions which are further divided into sub-regions, and so on [23][34][36]-[40]; (2) the bottom-up data driven approach progressively refines document image regions by grouping operations [11][24][30]. The top-down approach is fast and very effective for processing documents that have a regular format, e.g. one column or two column documents. Based on the observation that printed pages are primarily made up of rectangular blocks, RLSA (Run-Length Smearing Algorithm)[34] and

Projection Profile Cut [23] are applicable to those documents. The Projection Profile Cut method is useful especially for processing oriental language newspaper with vertical text lines [23]. The top-down approach may fail if documents are skewed or the spacing is non-uniform within a page. Moreover, it is incapable of extracting text embedded in graphics, engineering drawings, maps, and tables with different orientations [30][32][41]-[43]. The bottom-up method, therefore, is more suitable for processing those types of documents. The bottom-up approach first extracts the individual connected components, and then determines whether a connected component is a part of a text, a line drawing, or a half-tone image. In the process of grouping text components, connected components are first merged into words, subsequent words are merged into lines, lines into paragraphs, and so on. Whether a connected component is classified as a part of a text, a line drawing, or a half-tone image depends on the specific features for performing these classifications, such as size, geometrical complexity, topological structure, and shape measures [44][45].

(iv) Text Recognition

Character segmentation and recognition is the most critical stage in the document analysis system. Any failures in this stage will affect the overall performance of the system. There are two major classes of methods for text recognition. They are generally described as character-based and word-shape-based.

The word-shape-based recognition algorithms [46]-[49] determine features from a whole word shape and use these descriptions to calculate a group of words in the dictionary or lexicon that match the inputs. Such methods are especially suitable for images that are difficult to be segmented into characters. Currently, the word-shape-based approaches are being explored in the research stages, and restricted to very specific applications e.g. ZIP code recognition. These techniques need lots of computer memory to store word images and word lexicons.

Most recognition algorithms available in the current literature are character-based. In character-based recognition algorithms, technical challenges in printed character recognition arise mainly from three types of sources [2][3][13]:

1 Omnifonts and deformation: omnifonts represent possible symbol typefaces currently used in word processors. Deformation means shape variations in the same font.

2 Degraded text image: imperfections in images due to printing, scanning, binarization, etc.

3 Touching and broken characters: touching characters are represented by a single connected component while containing more than one character. Broken characters are observed as a character that may be split into multiple pieces.

The mandate for text recognition is to recognize the pixel image in each segment. If a segment contains a properly segmented character image, ideally the output is a specific character with a relative high confidence value. If the input segment contains touching character components, then this segment must be cleaved into distinct character components and recognized correctly by a classifier [2][50]-[54].

There are two distinct problems raised by document image degradation. Firstly, individual characters do not always have the same shape with respect to the prototypes of the character sets. For example, an 'e' may have its hole filled in, or look like 'c' with a broken crossbar. Some badly blurred samples even confuse a human. Some of the most promising approaches to omnifont problems fail on degraded images[2][3][55]. They reject scanned document images because of degradation caused by the paper type (e.g. grain, glossiness), printing technology, ink quality, scanner resolution, etc. Therefore, it becomes necessary to model image defects in order to build accurate classifiers with limited character samples. Pseudo-random defect generators [55][56] are used to read one or more sample images and produce an arbitrarily large number of distorted versions. Alternative classifiers such as neural networks [57][58], which are trainable and model free classifiers, become more significant since they are highly resistant to character image defects.

Secondly, individual character images are frequently difficult to isolate in scanned images. Adjacent characters may be joined to each other or may overlap each other; or a character may be broken into several pieces. Touching characters are particularly common in fonts using proportional spacing or serif. Even on a clean magazine page scanned at optimal threshold with good-quality scanner, some characters are still touching. If the page is a dark photocopy or scanned at a low threshold, many characters could be joined; if the page is a light photocopy or scanned at a high threshold, many characters could be split [2].

Many solutions have been proposed for character segmentation/recognition, especially for segmenting touching characters. Most approaches to solving touching characters are accomplished in four steps: segmentation, character component extraction, classification, and resegmentation. Algorithms for segmenting touching characters include cutting discrimination functions with contour analysis [54], break point searching with tree structures [10], aspect-ratio estimation with character models [59], and recursive processing with matching approaches [60]. Any uncertainties of recognition results will be resolved by using language-specific knowledge or contextual information in post-processing stages.

(v) Contextual Information Processing

Contextual information processing is often effective in resolving residual ambiguities of shape caused by incorrectly segmented symbols, typeface variations, and distortions due to imaging defects. There are two types of context information: layout context information and linguistic context information. The former covers baseline information and the location of characters with respect to their neighbours. The latter includes spelling, grammar, and punctuation rules [54]. Linguistic contextual processing can be classified as either statistical or dictionary based [61]-[65]. Integration of the lower level character recognition process with the higher level contextual abstraction is necessary since efforts at merely the lower level cannot achieve performance comparable with that of humans.

1.2 Goals of This Thesis

A survey of the relevant literatures in the area of document image processing revealed that many effective algorithms and techniques have been developed for current document analysis systems. These algorithms are capable of segmenting and recognizing a variety of documents. However, few reliable algorithms have been derived for text extraction from overlapped non-text components, such as background symbols and handwritten interference markings, which are essential to enhance the performance of document analysis systems.

With current technology, it is not difficult to design an optical character recognition system that can recognize well-formed and well-spaced printed characters at a very high recognition rate [4] . However, touching characters commonly occurring in certain fonts may deteriorate the performance of recognition systems. As a result, many researchers have focused their efforts on developing efficient algorithms for document segmentation and restoration systems. This thesis, therefore, will address the issues regarding document segmentation and restoration. It is desired especially to develop:

a) an effective approach to text string extraction from regular periodic overlapping text/background images.

b) an appropriate algorithm for implementing removal of handwritten interference marks from document images.

c) a new scheme for segmenting and recognizing machine printed touching characters.

1.3 Outline of the Thesis

Chapter II will describe the development and implementation of a morphological approach to character string extraction from overlapping text/background images that minimize the shape distortion of characters.

A new algorithm to effectively segment not only straight lines but also arbitrarily oriented curves will be discussed in Chapter III. The morphological operations based on orientation map and skeleton images are adopted to successfully prevent the "flooding water" effect of the conventional morphological operations.

In Chapter IV, we will present a new discrimination function for segmenting touching characters based on pixel and profile projections. A dynamic recursive segmentation algorithm has been developed for effectively segmenting touching characters. Contextual information and spell checking are used to correct errors caused by incorrect segmentation and recognition.

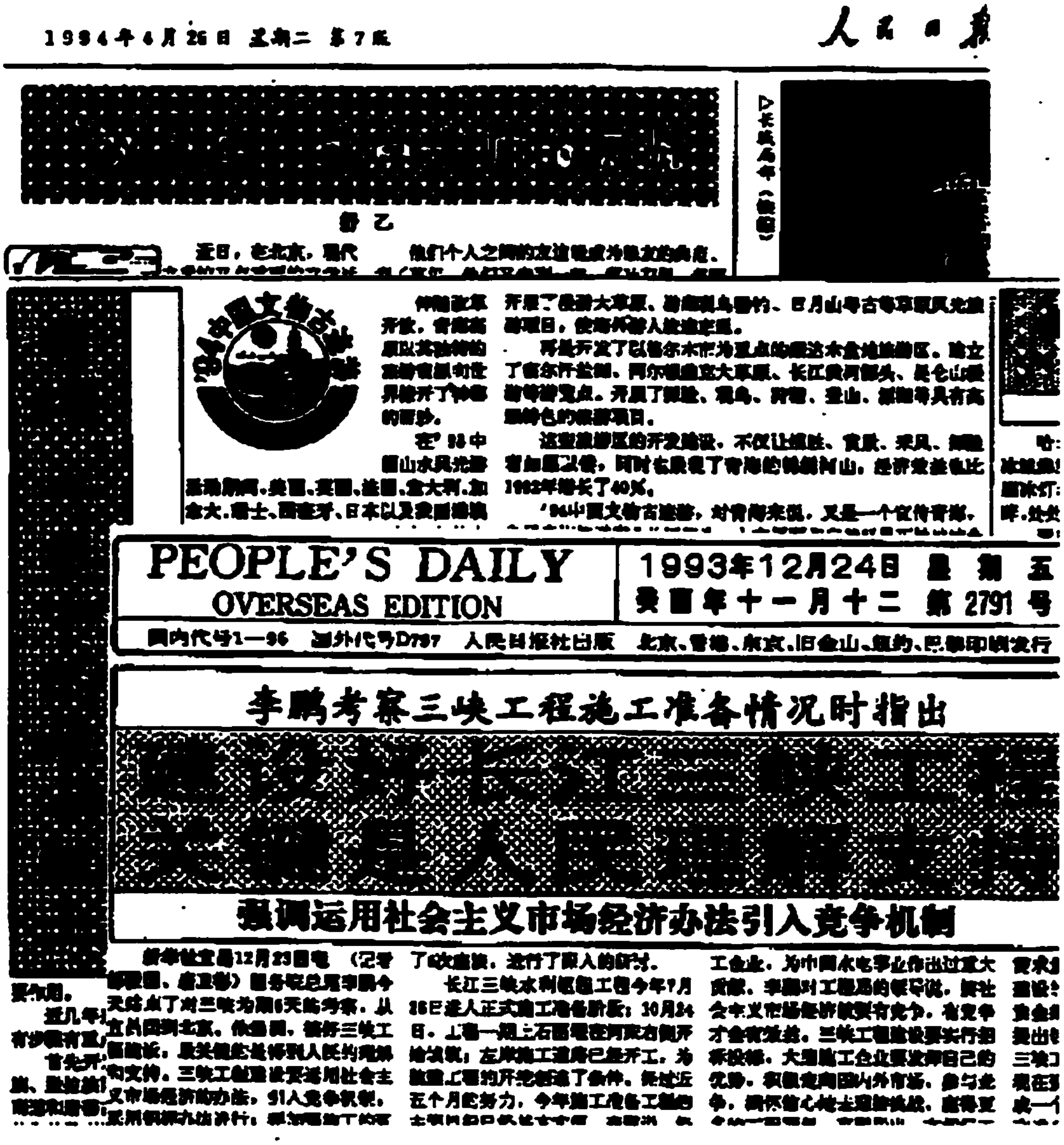
The conclusions and contributions of this thesis are summarized in Chapter V.

Chapter II

A Morphological Approach to Text String Extraction from Regular Periodic Overlapping Text/Background Images

2.1 Introduction

The Optical Character Reader (OCR) is a common tool used to recognize characters. However, most OCR systems can only read traditional characters: black characters on a uniform white background, or vice versa. However, characters are often printed over some complex backgrounds. Examples can be found in mail pieces where the addresses are written on pattern papers [13], the headlines of Japanese newspapers[26], and the headlines of Chinese newspapers (see Figure 2.1). Some text blocks that are decorated with uniformly distributed graphical symbols in their backgrounds are designed to help readers to distinguish various parts of a text and to make some articles in newspaper more attractive. Figure 2.2 shows an article from NewsLine, a publication of the University of Windsor, which contains text printed on a regular periodic background of dots. People can differentiate graphical symbols from text, and instinctively remove the background symbols without difficulty. However, it is not simple for a computer to distinguish text strings from background symbols. The whole text blocks usually become unreadable and are incorrectly recognized as graphics and removed from document images by the text recognition systems. Therefore, it is necessary to perform a pre-processing procedure to extract text images before a recognition algorithm is applied.



1994年4月26日 星期二 第7版

人民日報

[illegible]

舒乙

近日，在北京，现代[illegible]

他们个人之间的友谊也成为[illegible]

▷长城局部（[illegible]）

伴随改革开放，青海高原以其独特的高原风貌向世界掀开了神秘的面纱。

在’93中国山水风光游[illegible]活动期间，美国、英国、法国、意大利、加拿大、瑞士、西班牙、日本以及我国港澳[illegible]

开展了[illegible]大草原、[illegible]鸟岛[illegible]、日月山等古专草原风光旅游项目，使海内外游人流连忘返。

再是开发了以格尔木市为重点的柴达木盆地旅游区。建立了察尔汗盐湖、[illegible]大草原、长江黄河源头、昆仑山[illegible]等游览点。开展了探险、观鸟、狩猎、登山、[illegible]等具有高原特色的旅游项目。

这些旅游区的开发建设，不仅让[illegible]、[illegible]、[illegible]、[illegible]有[illegible]，同时也展现了青海的锦绣河山。经济效益也比1992年增长了40%。

’94中国文物古迹游，对青海来说，又是一个宣传青海、[illegible]

PEOPLE’S DAILY
OVERSEAS EDITION

1993年12月24日 星期五
癸酉年十一月十二 第2791号

国内代号1—96 国外代号D797 人民日报社出版 北京、香港、东京、旧金山、纽约、巴黎印刷发行

李鹏考察三峡工程施工准备情况时指出

[illegible]

强调运用社会主义市场经济办法引入竞争机制

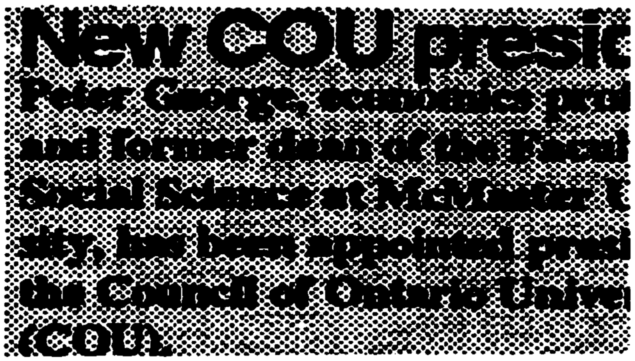
新华社宜昌12月23日电 （记者[illegible]、[illegible]）国务院总理李鹏今天结束了对三峡为期6天的考察，从宜昌回到北京。他指出，搞好三峡工程建设，最关键的是得到人民的理解和支持。三峡工程建设要运用社会主义市场经济的办法，引入竞争机制，采用招标办法进行，[illegible]

了[illegible]，进行了深入的研讨。

长江三峡水利枢纽工程今年7月26日进入正式施工准备阶段；10月24日，上游一期土石围堰在河床右侧开始填筑；左岸施工道路已经开工，为[illegible]工程的开挖创造了条件。经过近五个月的努力，今年施工准备工程的[illegible]

工企业，为中国水电事业作出过重大贡献。李鹏对工程局的领导说，[illegible]社会主义市场经济就要有竞争，有竞争才会有发展。三峡工程建设要实行招标投标，大型施工企业要发挥自己的优势，积极走向国内外市场，参与竞争，满怀信心地去迎接挑战，[illegible]

Figure 2.1 Examples of Chinese newspaper headlines



New COU presi
Peter George, economics pro
and former dean of the Facul
Social Science at McMaster U
sity, has been appointed presi
the Council of Ontario Unive
(COU).

Figure 2.2 An image sample with overlapped text/background
(from the NewsLine of the University of Windsor)

The extraction of text from various kinds of images in which text strings touch and intersect lineworks [29], or scratches [66], or noise backgrounds [67], or geometric background patterns [15][17][26] has become the subject of extensive research. Thresholding [14][15][18] is a popular tool for segmenting grey level images. The approach is based on the assumption that objects and background pixels in an image can be distinguished by their grey level values. By appropriately choosing a grey level threshold between the dominant values of the object and background intensities, the original grey level image can be transferred into a binary form so that the image pixels associated with the objects and the background obtain values 1 and 0, respectively. White and Rohrer [15] described an image thresholding technique based on boundary characteristics to suppress unwanted background patterns so that only printed or hand-written characters may be captured as electronic images. Liu et al [68] proposed a new scheme by using the underlying properties, such as the run-length histogram and the texture attributes, to correctly binarize document images. Yamada et al [17] developed a recognition method for characters stamped on metal. They attempted a local binarization after smoothing the uneven background to generate features for distinguishing figures or characters from backgrounds. Billawala et al [66] described a technique referred to as the image continuation algorithm to remove scratches and blemishes in binarized images. H. Ozawa et al [26] proposed a method to remove geometric pattern backgrounds in Japanese newspaper headlines.

In this Chapter, we will present the algorithms to deal with binarized documents containing text with regular periodic overlapping backgrounds. The proposed algorithms that perform text character extraction accommodate document images that contain various kinds of periodically distributed background symbols. Mathematical morphology, because of its ability to grasp the geometry and structure of images, is adopted to realize this new scheme. The effectiveness of the proposed algorithms is demonstrated by several experiments that we have conducted.

2.2 Basic Morphological Tools

Many theoretical results relating to the mathematical morphological operations can be found in [19][20][69]-[73]. These operations have been applied successfully to a large variety of image processing and analysis applications, such as biomedical image processing, cellular automata, automated industrial visual inspection, etc [69]. For instance, basic morphological operations can be used for noise suppression [73], texture analysis and image enhancement [14], and shape analysis [72].

Mathematical morphology is a set-theoretic approach to image processing and analysis that considers images to be sets in underlying space and manipulates them using set-based operations such as union and intersection. The fundamental operations, erosion and dilation, are implemented by "AND"ing or "OR"ing the images that have been translated by structuring elements to generate eroded or dilated images.

Definitions:

Let X be a set representing a binary image. Let B denote a structuring element that describes a simple shape (e.g. square, circle). $(X)_y$ is defined as the translation of X by vector y, i.e., $(X)_y = \{ x + y \mid x \in X \}$. Two fundamental morphological operations on X are defined as follows:

$$\text{erosion:} \quad X \ominus B = \cap_{b \in B} (X)_{-b} \tag{2.1}$$

$$\text{dilation:} \quad X \oplus B = \cup_{b \in B} (X)_b \tag{2.2}$$

Erosion is a shrinking of the original image and dilation is an expansion of the original image. In practice, dilations and erosions are usually employed consecutively: either an image is dilated then eroded or an image is eroded then dilated. In either case, iteratively applying dilations and erosions eliminates specific components smaller than the structuring element without any global geometric distortion.

The opening $\circ$ and closing $\bullet$ operations are defined as

$$X \circ B = (X \ominus B) \oplus B \tag{2.3}$$

$$X \bullet B = (X \oplus B) \ominus B \tag{2.4}$$

Closing is extensive, it always gives us a closed image object which is equal to or larger than the original image object. On the other hand, the opening is anti-extensive, it always gives us an opened image object which is equal to or smaller than the original image object. Therefore, for any given structuring element B,

$$X \circ B \subseteq X \tag{2.5}$$

$$X \bullet B \supseteq X \tag{2.6}$$

Figure 2.3 shows illustrations of the basic morphological operations (erosion, dilation, opening, and closing).

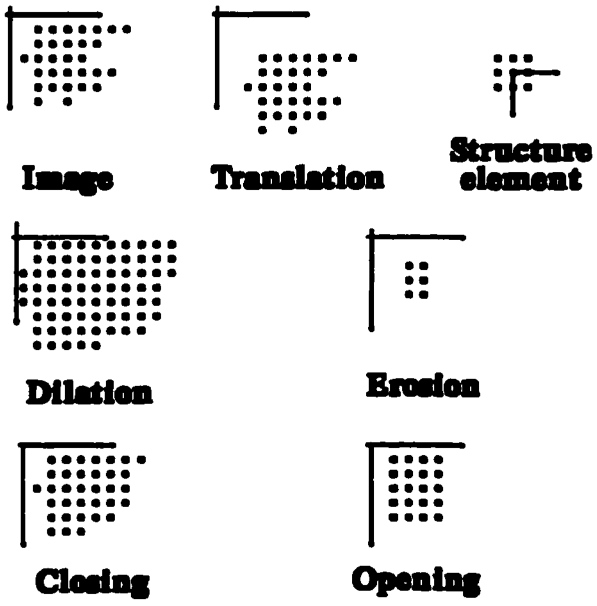


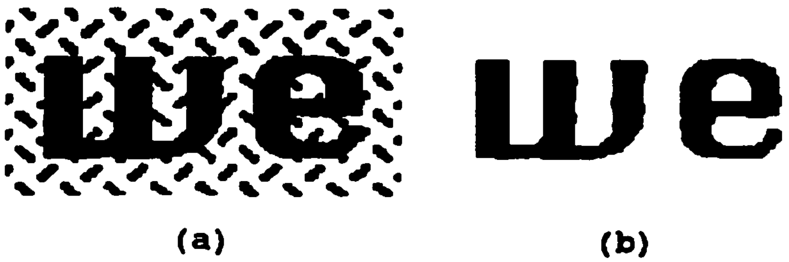
Figure 2.3   Examples of basic morphological operations

2.3 Character String Extraction from Overlapping Text/Background Images

Overlapping text/background images can be directly opened with an appropriate structuring element to remove the background components that touch character strings [73]. All connected components below the size of a given structure element are removed and the lager objects remain substantially unchanged. This approach to background removal is only suitable when the ratio of the width of text characters to the width of the background symbols is appropriately large. In Fig. 2.4, the word 'we' was almost perfectly extracted from the overlapped background by simply opening the image with the following equation:

$$X \circ (B \oplus B) = (((X \ominus B) \ominus B) \oplus B) \oplus B \qquad (2.7)$$

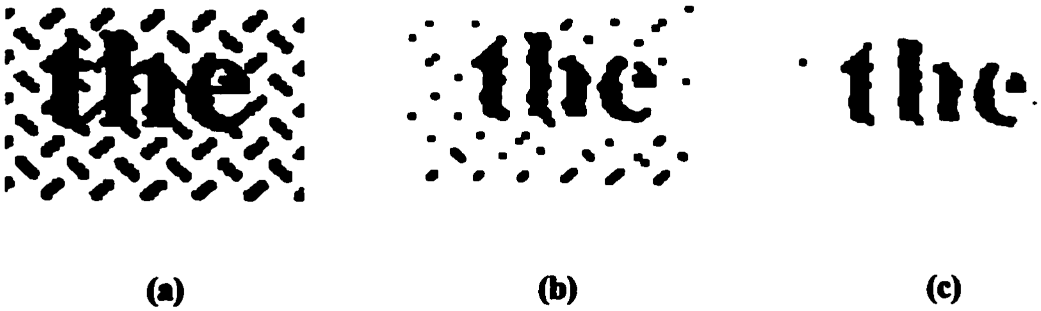
where X is an image set, and B is a 3 X 3 square structuring element.



(a) (b)

Figure 2.4 (a) The word 'we' with an uniform background
(b) Background removal after opening the image with a 9 X 9 square structuring element

However, a simple opening operation is not an effective approach to obtaining high quality text strings from text/background images which have a small ratio of the width of text characters to the width of background symbols. The results of opening an image set (Figure 2.5(a)) with unsuitable structuring elements are shown in Figure 2.5(b) and 2.5(c).



(a) (b) (c)

Figure 2.5 (a) The original image
(b) The result opened with a too small structuring element
(c) The result opened with a too large structuring element

The algorithm that we are presenting can extract character strings from regular periodic backgrounds regardless of the orientation and style of background symbols. The underlying strategy of our algorithm is to maximize background component removal while minimizing the shape distortion of text characters by using appropriate morphological operations. For proper segmentation, the overlapping text/background images should meet the following requirements:

(1) The graphic symbols in the background are periodically distributed.

(2) The width ratio of the minimal stroke of the character strings to the background symbols is approximately equal to 1.

(3) The resolution of the digitizer should be high enough such that the topological property of each character would not be eliminated because of low resolution.

However, acceptable segmentation results are still obtained even if some of these constraints are not strictly satisfied. Figure 2.6 shows a flow chart of our text/background segmentation system. The algorithm will be described in detail in the following sections.

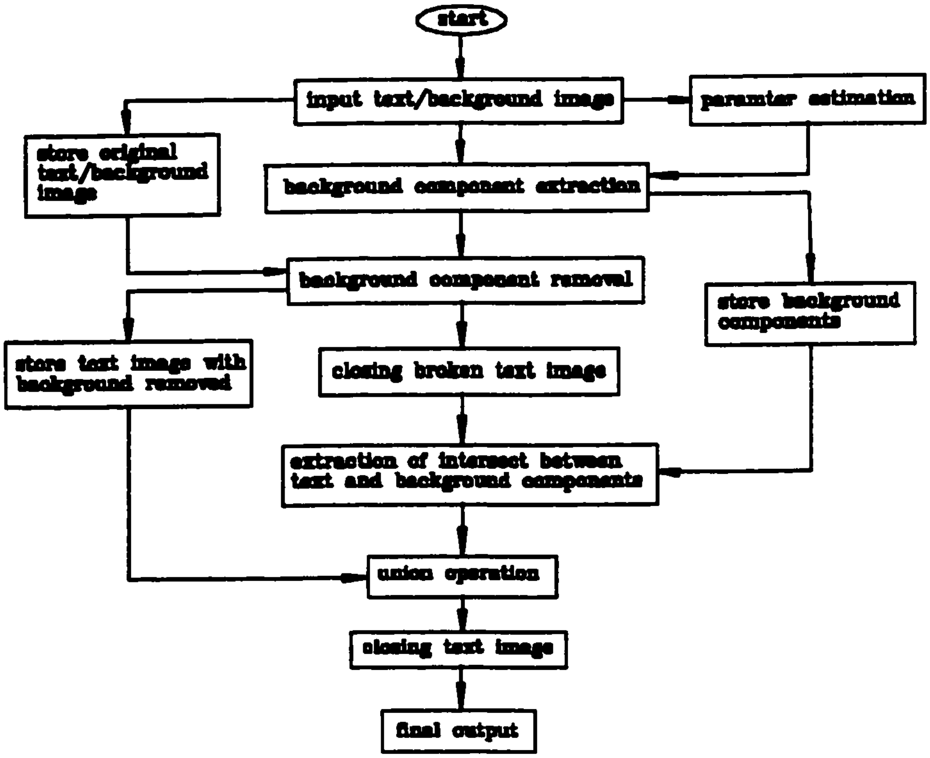

start
input text/background image
paramter estimation
store original text/background image
background component extraction
background component removal
store background components
store text image with background removed
closing broken text image
extraction of intersect between text and background components
union operation
closing text image
final output

Figure 2. 6 A flow chart of the text extraction system

2.3.1 Parameter Estimation of Uniform Backgrounds

By observing the structure of uniform backgrounds, we can easily see that the background symbols are periodically distributed in both the horizontal and vertical directions. Hence, it is possible to extract the background symbols from the text/background image when their 'frequency' of repetition is known and proper structuring elements are chosen. For expression simplicity, we use Periodic Distance (in pixel units) to describe the periodicity of the background symbols. The PDH and PDV represent the Periodic Distance in the horizontal and vertical directions respectively. They satisfy the following inequalities:

$$CL((X \ominus T1) \ominus B_{PDH}) > CL((X \ominus T1) \ominus B_i) \qquad (2.8)$$

$$i = 1, 2, \ldots, M, \; i \neq PDH$$

$$CL((X \ominus T2) \ \ominus B_{PDV}) > CL((X \ominus T2) \ominus B_j) \qquad (2.9)$$

$$j = 1, 2, \ldots, M, \; j \neq PDV$$

X is the text/background image set, CL(E) is a pixel counting function that calculates the total number of image '1' pixels for a given binary image set E, M is a constant, and B_i and B_j are two *point pair* structuring elements with i (≤ M) and j (≤ M) apart respectively as defined in Figure 2.7(a) and 2.7(b).

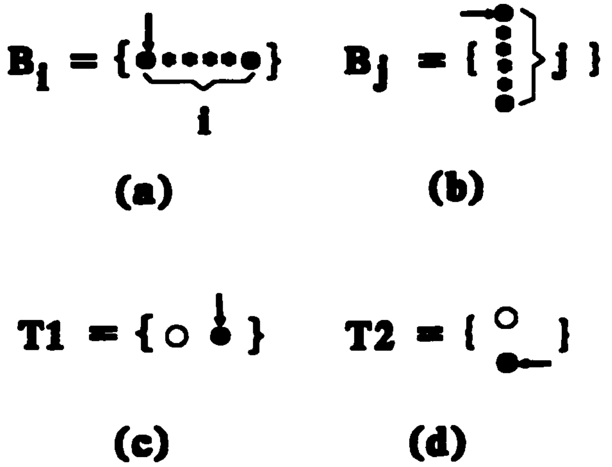


Figure 2.7 (a)(b) Point pair structuring elements B_i and B_j
(c)(d) The transit structuring elements T1 and T2
•: points of the structuring element belonging to binary image set X
*: don't care points
↓, ←, →: location of the origin associated with the structuring elements

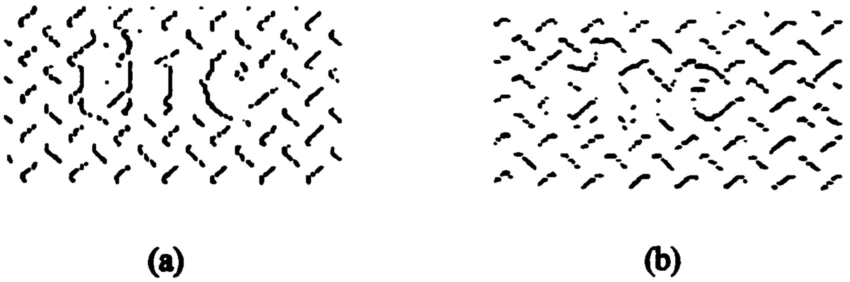


Figure 2.8 (a) The left edges of the text/background components of Figure 2.5(a)
(b) The top edges of the text/background components of Figure 2.5(a)

Two transit structuring elements, T1 and T2, defined in Figure 2.7(c) and 2.7(d), erode the image set X to extract the left and top edges; that is, only image pixels '1' which have blank pixels '0' in either their left or top sides are retained. The resultant left and top edges of Figure 2.5(a) are shown in Figs. 2.8(a) and 2.8(b). Because most redundant character and background pixels that may cause ambiguities and estimation errors have been removed, using the edge information of character and background components instead of all of the image pixels to estimate the PDH and PDV parameters produces accurate results. The procedure for calculating PDH and PDV is described in Appendix 1.

Figure 2.9 is a simple example of how the erosion operations are performed by the structuring element B_i (i = 1, 2, 3, 4) on a given edge image E. B_i (i = 1, 2, 3, 4) is applied respectively to erode the image E. The resulting images (Ei (i =1, 2, 3, 4)) are shown in Figs. 2.9(b)-2.9(e). o points and • points in these figures denote the removed pixels and retained pixels which belong to the original image E. After erosion process is completed, the black pixel counting function is applied to the eroded images Ei (i =1, 2, 3, 4) to calculate the total number of image pixels remaining in the eroded images. The black pixel counting procedures are simply expressed by CL(Ei) (i = 1, 2, 3, 4). For example, after the structure element B_1 erodes the image E, there remain 4 black image pixels (Fig. 2.9(b)). As seen in Figs 2.9(b)-2.9(e), CL(Ei) is maximized with i = 3 after application of B_3, CL(E3) = 32 (see Fig. 2.9(d)). According to Eq. (2.8), the subscript index 3 of the structuring element B_3 represents the PDH value of the edge image E. Similarly, the PDH and the PDV are obtained (Fig. 2.10) based on the edge images Fig. 2.8(a) and 2.8(b), and are used in the next step to extract the background symbols from the text/background images.

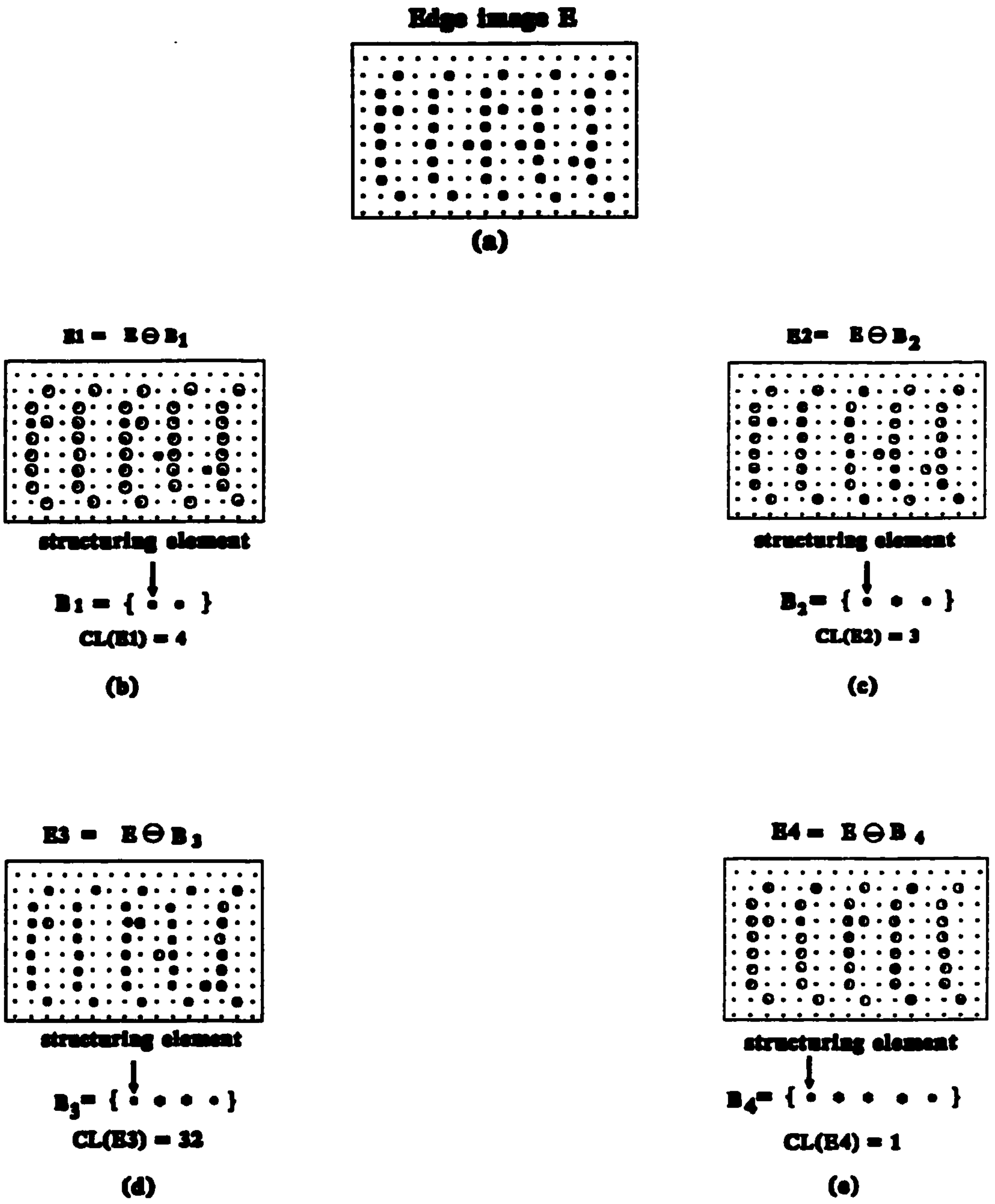


Figure 2. 9 (a) An initial edge image E generated for illustration
(b) - (e) the symbols concerning the structuring elements B_i, i = 1, 2, 3, 4
•: points of the image which belong to Ei.
o: points of the image which belong to Ei^c.
.: points of the image which indicate pixel positions.
↓:location of the origin of the structuring elements.
*: don't care and CL(X) is the pixel counting function on a given image set X

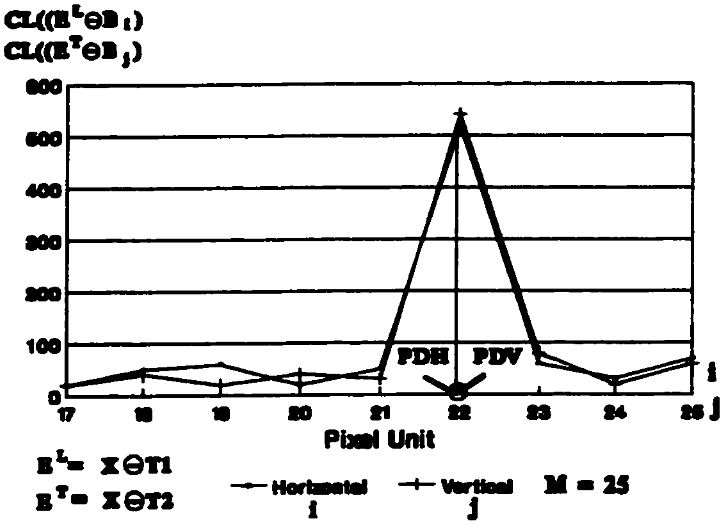


Figure 2.10 Histograms of the CL($E^L \ominus B_i$) and CL($E^T \ominus B_j$)
E^L and E^T are the left edges and top edges.
$E^L = X \ominus T1$, $E^T = X \ominus T2$, X is a text/background image set.
PDH and PDV indicate values corresponding to the maximums of the histograms in the horizontal and vertical directions.

2.3.2 Morphological Operations for Extracting Background Components

A morphological process is designed to extract the background symbols or remove the character strings from a specific text/background image. Based on the parameters PDH and PDV obtained from Eq. (2.8) and Eq. (2.9), four structuring elements, S1, S2, S3, and S4, are designed to realize the appropriate morphological erosion operations (Fig. 2.11). These erosion operations are used to remove the text strings from the text/background images. For example, an erosion operation with the structuring element S1 allows a scanned image pixel '1' to be removed only if all three pixels PDH-1, PDH, and PDH+1 pixel units to the right of the current pixels are '0' pixels. Otherwise, the scanned image pixel '1' remains unchanged. Using three reference points in the structuring elements to determine the binary value of image pixels helps to prevent the erosion operations from excessively eliminating background pixels that have been distorted by optical devices such as digitizers, laser printers, and etc.

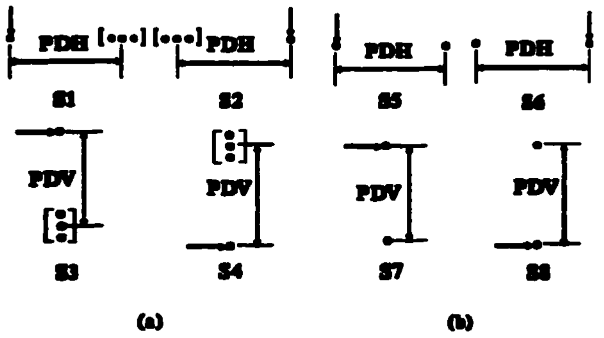


Figure 2.11 (a) Structuring elements (S1, S2, S3, S4)
(b) Structuring elements (S5, S6, S7, S8)
•: points of the structuring element which must belong to image set X
[•••]: At least one of the points inside the square bracket must belong to image set X
↓ : locations of the origin of the structuring elements

The recursive erosion $RE_i(X)_s$ of an object image X with the structuring element S is defined by the following relation:

$$RE_i\,(X\,)_S \;=\; (X\,\Theta\,S\,)_i \;=\; ((...((X\,\Theta\,S)\,\Theta\,S)...\Theta\,S))\ i\ time \qquad (2.10)$$

therefore, the whole procedure can be described by the recursive relation

$$RE_i(X\,)_{(S1|S2|S3|S4)} \;=\; (((((X \underset{(LR)}{\Theta} S1) \underset{(TB)}{\Theta} S3) \underset{(RL)}{\Theta} S2) \underset{(BT)}{\Theta} S4))\,)_i \qquad (2.11)$$

S1 and S3 erode the image X from left to right (LR) and from top to bottom (TB) respectively. Similarly, S2 and S4 erode the image X from right to left (RL) and bottom to top (BT) respectively. This alternative multidirectional scan increases the convergence rate and ensures that all possible image pixels that do not belong to the background are replaced with blank pixels.

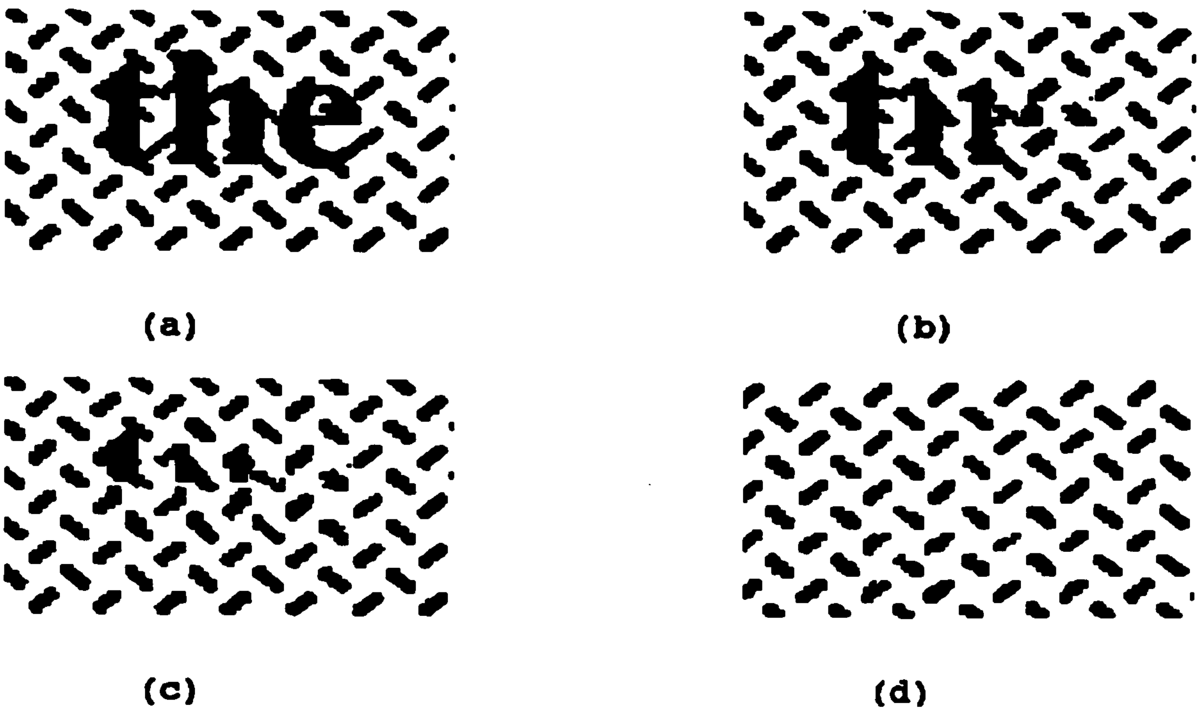
The erosion procedure is an iterative erosion of the image X until the termination condition

$$X_i = X_{(i-1)}\,\Theta\,Sj, \quad j = 1, 2, 3 \text{ and } 4. \qquad (2.12)$$

is satisfied. X_i and X_{i-1} denote the current eroded image and the previous eroded image respectively. When the termination condition (2.12) is satisfied, it also means that $CL(X_i)-CL(X_{i-1}) < \alpha$. α is a small constant and CL(X) is the pixel counting function applied to the image set X. Fig. 2.12 shows the result of the background extraction process after 9 iterations of the erosion operation with the structuring elements S1, S2, S3, and S4. Fig. 2.13 shows the result of the background extraction process after 4 iterations of the erosion operation with the ***point pair*** structure elements S5, S6, S7, and S8. Comparing Fig. 2.12 and Fig. 2.13, although the convergence rate of the background extraction process is faster with structuring elements

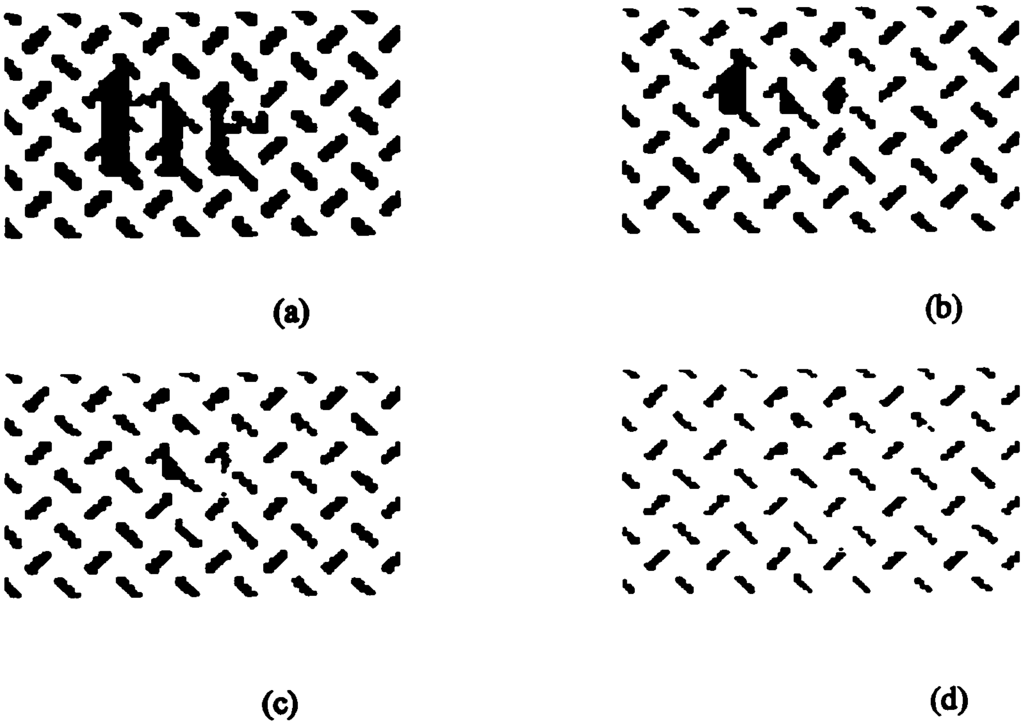
Si (i = 5, 6, 7, 8) than that with structuring elements Si (i = 1, 2, 3, 4), the significant loss of the background pixels can be obviously observed in Figure 2.13(d). The loss of background pixels will produce noise on text images after background removal. The extra noise filtering will degrade the quality of restoration of text parts.

The procedure for extracting background pixels is described in Appendix 2.



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

Figure 2.12 (a) The original text/background image
(b) The image eroded by S1 after one iteration.
(c) The image eroded by S2 after two iterations
(d) The background extracted after nine iterations



(a)

(b)

(c)

(d)

Figure. 2.13 (a) The image eroded by S5 after one iteration.
(b) The image eroded by S6 after two iterations.
(c) The image eroded by S7 after three iterations
(d) The image eroded by S8 after four iterations

2.3.3 Extraction of Character Strings from the Background

After the background extraction procedure, the background removal operation (exclusive-OR (XOR) operation)

$$Z = XOR(X, Y) \tag{2.13}$$

is performed on the original text/background image set X (Fig. 2.14(a)) and the background image set Y (Fig. 2.14(b)). The result Z denotes the character string image with removed background (Fig. 2.14(c)). Directly applying the closing operation on the image Z (i. e., Z • B, where B is an appropriate Structuring element) fills the internal gaps of the character strings. However, this produces serious shape distortion. To improve the results of the morphological operation, a conditional dilation is performed before the closing operation:

$$W = (((Z \oplus B1) \cap Y) \cup Z) \bullet B2 \tag{2.14}$$

B1 and B2 are two suitable structuring elements. The size of these structuring elements depends largely on the resolution of the image and the size of the background symbols.

Dilation along the background pixels, i.e. $D = ((Z \oplus B1) \cap Y) \cup Z$ (Fig. 2.14(d)), compensates for the pixels lost during background removal. Fig. 2.14(e) shows the image after application of Eq. (2.14). However, Eq. (2.14) still causes a loss of image detail even though it preserves most topological properties of the character strings.

The final operation for character string extraction is

$$R = ((W \cap Y) \cup Z) \bullet B3 \tag{2.15}$$

Here B3 is a proper structuring element that is used to fill the narrow crack between the intersect components ($U = W \cap Y$, in Fig. 2.14(f)) and the characters with internal gaps (Fig. 14(c)). R represents the final image (Fig. 2.14(h)).

Those text character pixels that do not overlap with background components should remain unchanged if the character shape distortion is to be minimized. The operation described by Eq. (2.15) can be effectively applied to improve the performance of the character extraction (Fig. 2.14(h), compare Fig. 2.14(e)).

The procedure for text restoration is described in Appendix 3.

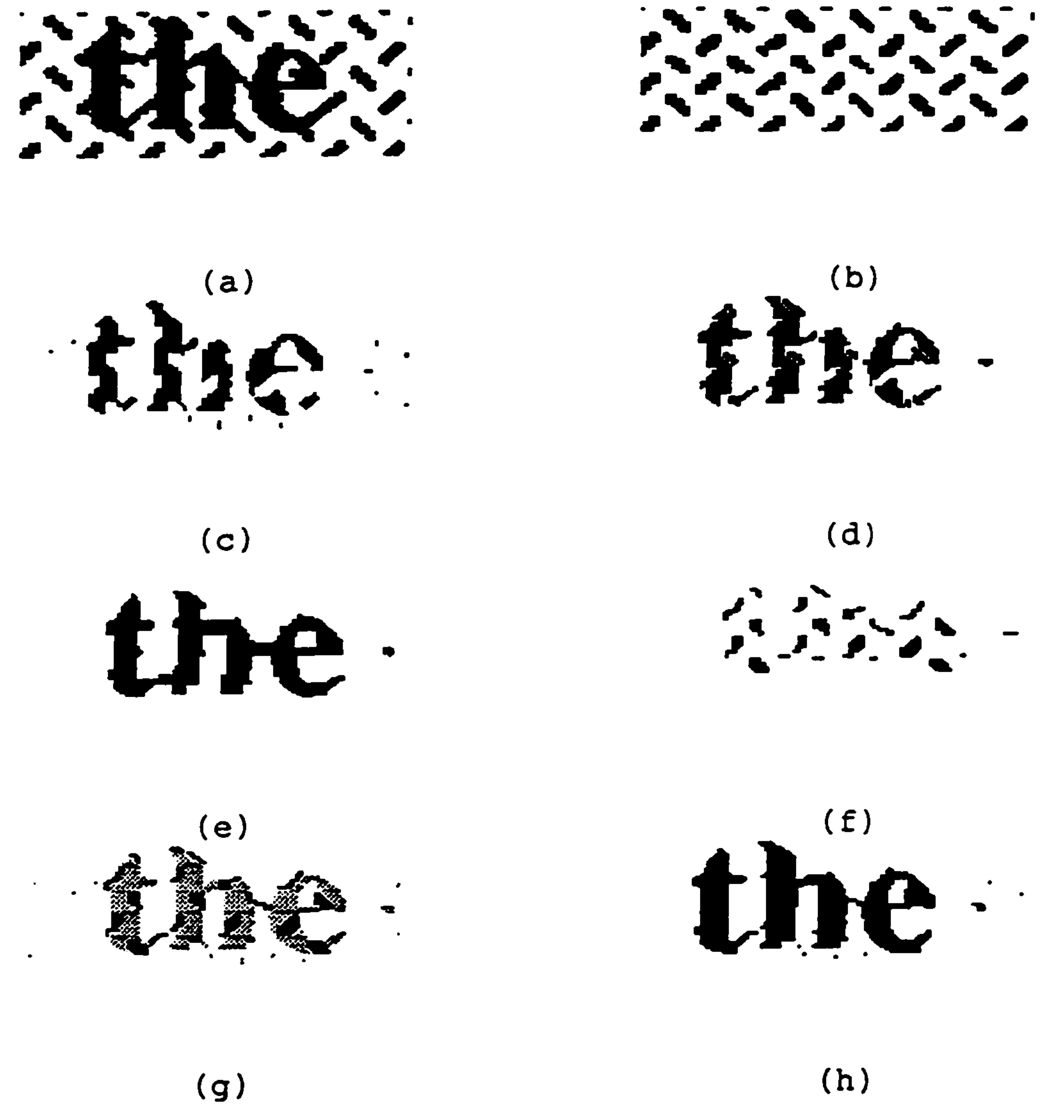


Figure 2.14 (a) The original overlapped text/background image (X)
(b) The extracted background components (Y)
(c) The image after background removal (Z)
(d) The characters obtained by a conditional dilation operation $D= ((Z \oplus B1) \cap Y) \cup Z$
(e) The characters by closing operation ($W = D \bullet B2$)
(f) The intersect parts of (b) and (e) (($U = W \cap Y$)
(g) The characters after ORing (f) and (c)($U \cup Z$)
(h) The final result ($R = ((W \cap Y) \cup Z) \bullet B3$)

2.4 Experimental Results

Seven test images, including six English texts and one Chinese newspaper headline with different patterns of periodically distributed background symbols, were obtained by a scanner (HP DeskScan II) with 150 dpi resolution. Programs were written in C language. Fig. 2.15 and Fig. 2.16 show the experimental results after extracting character strings from those overlapping backgrounds. The test results demonstrate the algorithm's ability to extract text strings from overlapping text/background images with acceptable shape distortion. The commercial OCR (HP DeskScan II WordScan 3.0) was applied to those English text images with and without periodically distributed background symbols to evaluate performance of the proposed algorithm. As shown in Fig. 2.15, the OCR completely fails to recognize the text with overlapping backgrounds. After background symbol removal, significant improvement (85% recognition rate) has been achieved based on those test images.

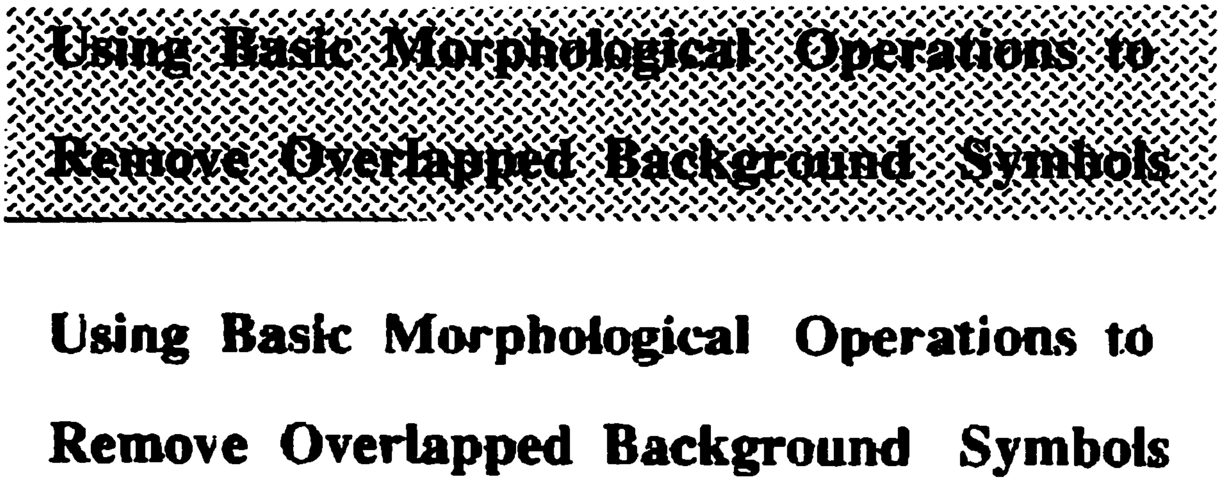
By using multiple reference points in structuring elements to determine binary value of image pixels can prevent erosion operations from excessively eliminating background pixels that have been distorted by optical devices such as digitizer, laser printer, etc. This new scheme can maximize background component removal while minimizing shape distortion of characters. If an image conforms to the constraints mentioned in section 2.3, the performance of the algorithm is effective regardless of the orientation and style of background symbols. However, the proposed algorithm may fail to remove those background symbols which do not have the properties of periodicity, and also satisfactory results may not be obtained by applying the proposed algorithm to the overlapping text/background images with small width ratio of the characters to the background symbols.

The proposed algorithm is composed of a number of erosion and dilation operations. All those operations are performed by mapping the current image to a two-dimensional image buffer with union and intersection operations. The amount of computation of the algorithm is measured by the number of union and intersection operations determined by the total number of image pixels in current image array and the size of the structuring element.

For a digital image, the number of image pixels is determined by image resolution. The more pixels an image has, the larger amount of computation would be taken to process the image. For example, if an image array contains 100 character image pixels, the computation cost will be 100 times that of an image array which contains only 1 character image pixel. The same relationship also exists among image resolution, the size of structuring elements and the amount of computation needed. The size of a structuring element is proportional to image resolution. Given a certain image, the size of a structuring element should be chosen according to the image resolution. The higher the resolution value is, the bigger the structuring element should be applied, or vice versa. For example, if the resolution is set to 150 dpi, a 5x5 structuring element is chosen to implement the operations. When resolution is increased to 300 dpi, the size of the structuring element should be 10x10 accordingly.

The memory requirement of this algorithm includes three two-dimensional arrays (3 x image_row x image_column bytes) to store the original image, two image buffers for morphological operations and the program itself.. In the experiment conducted, the test image size is 100 pixels by 450 pixels obtained with 150 dpi and 5 x 5 structuring elements were used in the algorithm. It took 20 seconds to finish the text extraction algorithm on SUN SPARC 10 work station.

As observed in our experimental results, morphological operations are quite useful for image processing and analysis. The algorithms can be simply designed by iteratively applying dilation and erosion operations. However, the simplicity of algorithms implies a great amount of repetitive computations. Thanks to today's advanced VLSI techniques, many special-purpose parallel image processors have been developed and are commercially available in recent years [67][69][74][75]. it is even possible to implement these operations on large images in real time [75].



Using Basic Morphological Operations to
Remove Overlapped Background Symbols

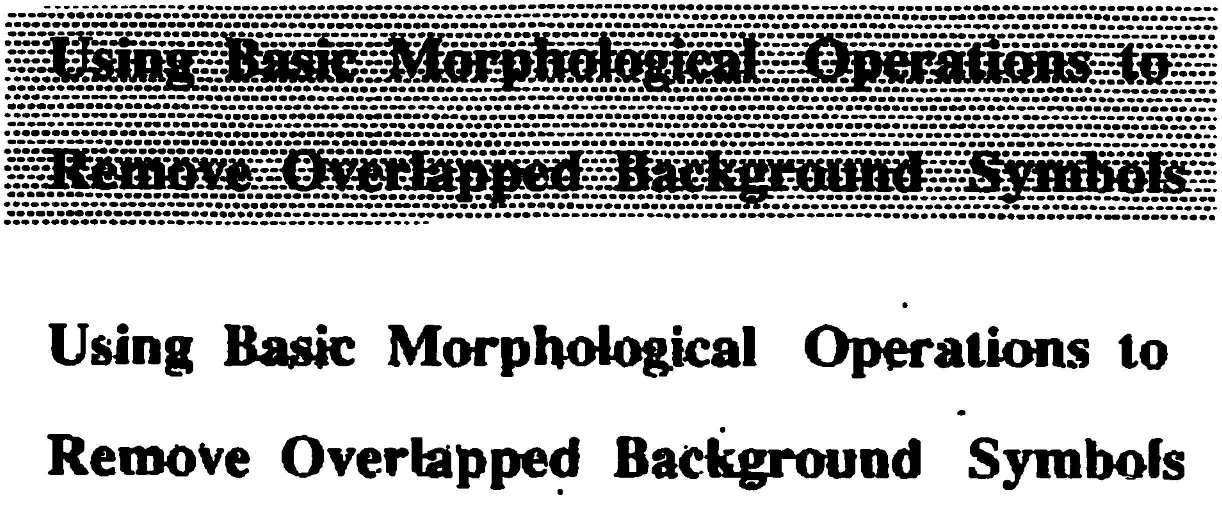
Using Basic Morphological Operations to
Remove Overlapped Background Symbols

(a)

```
ed.1 CR.1      //drare-s-dr.
%          %% %
       o°.@
                    %am
                  .....
                        %%
       %%% %z%z%@A4@
             %
wd%%( % % % %
```

```
Using Basi-c N4orphologic&al Operations to
Remove Overlapped Background Symbols
```

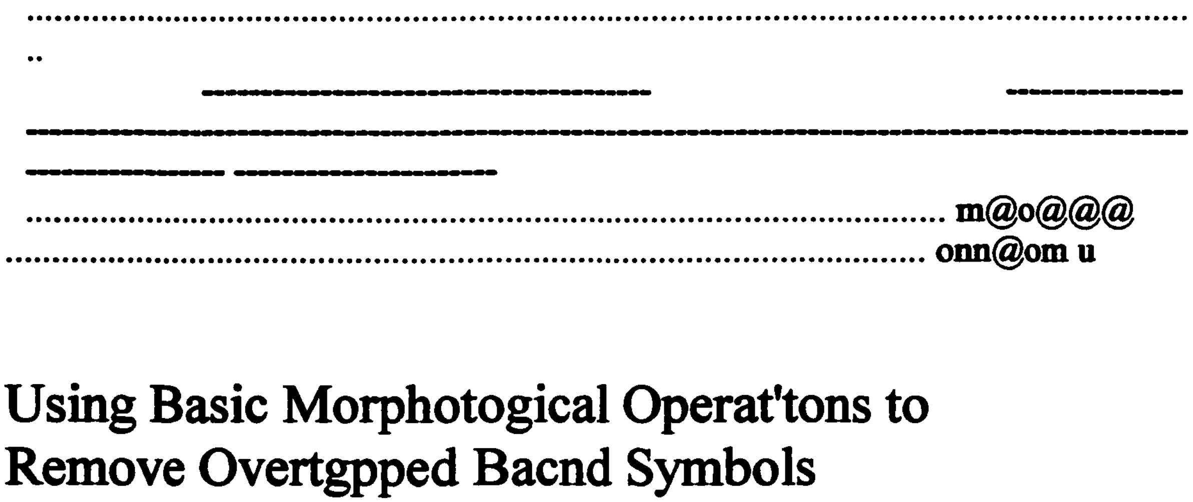
(b)



Using Basic Morphological Operations to
Remove Overlapped Background Symbols

Using Basic Morphological Operations to
Remove Overlapped Background Symbols

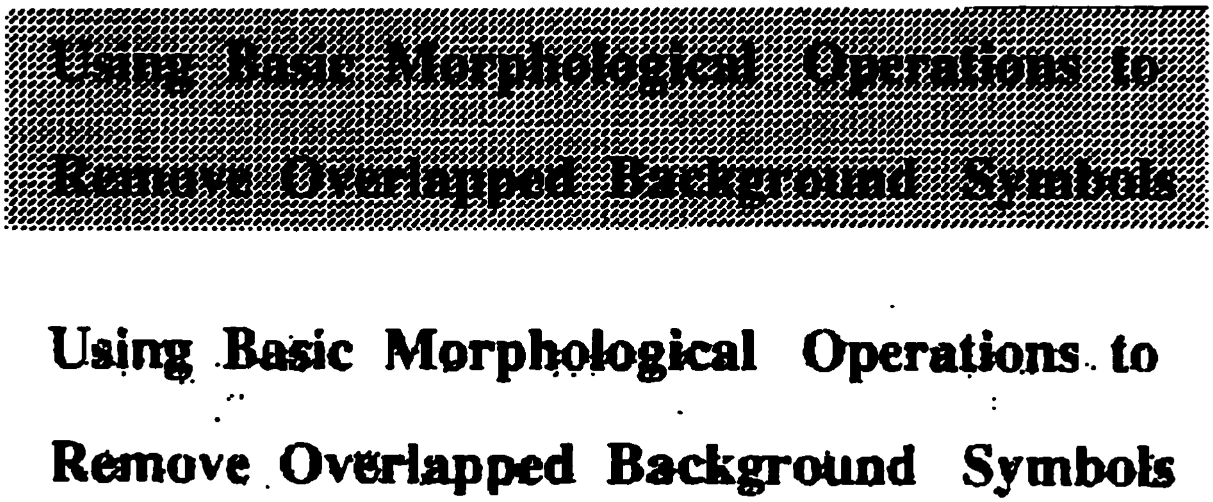
(c)



m@o@@@
onn@om u

Using Basic Morphotogical Operat'tons to
Remove Overtgpped Bacnd Symbols

(d)



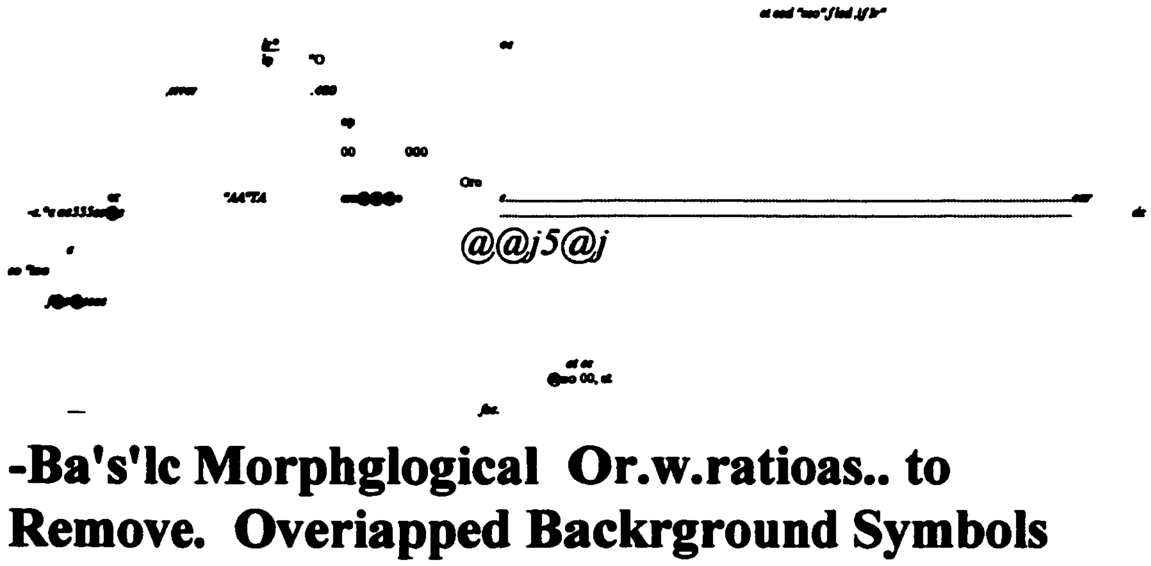
Using Basic Morphological Operations to

Remove Overlapped Background Symbols

Using Basic Morphological Operations to

Remove Overlapped Background Symbols

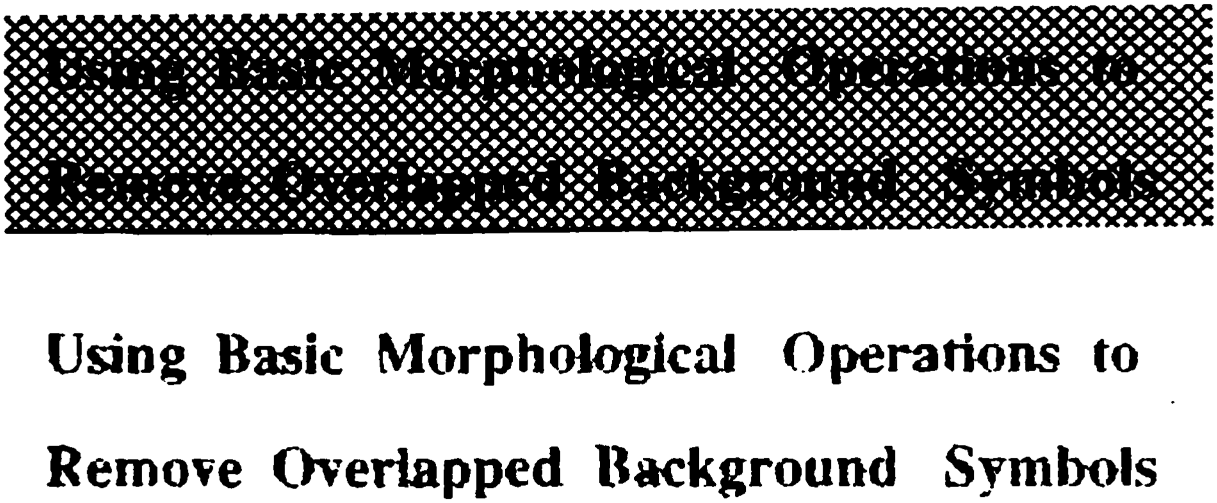
(e)



-Ba's'lc Morphglogical Or.w.ratioas.. to

Remove. Overiapped Backrground Symbols

(f)

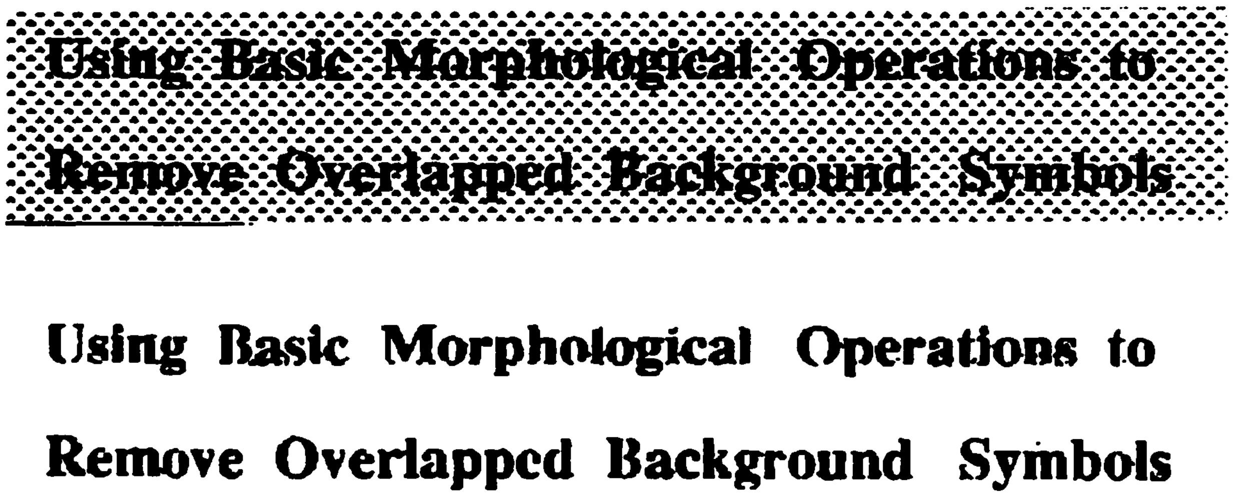

Using Basic Morphological Operations to
Remove Overlapped Background Symbols
Using Basic Morphological Operations to
Remove Overlapped Background Symbols

(g)

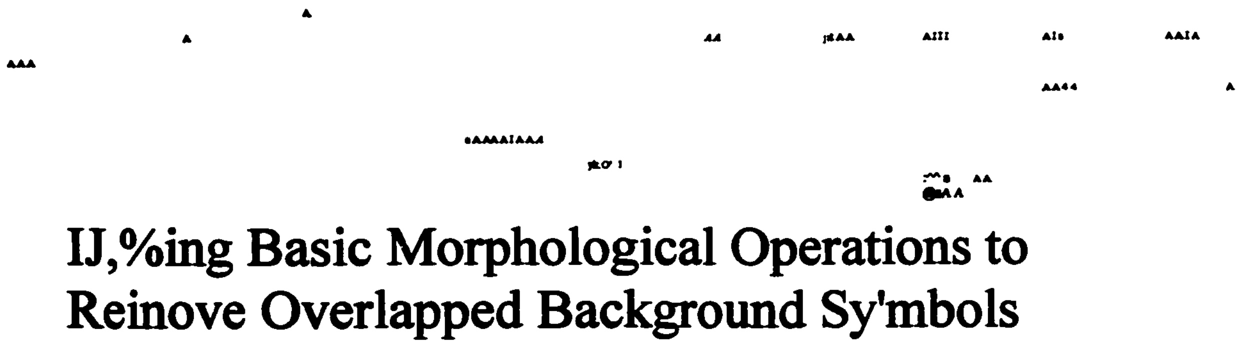
....
..
——————————————— ——————
——
—————— ————————
.... n
.... @@ @@ @

(Jsl'ng Ba-,sic N/lorpbologica-I 0 rat()ns to
Remove Overlapped Ilackground Sviiil)ri@ls

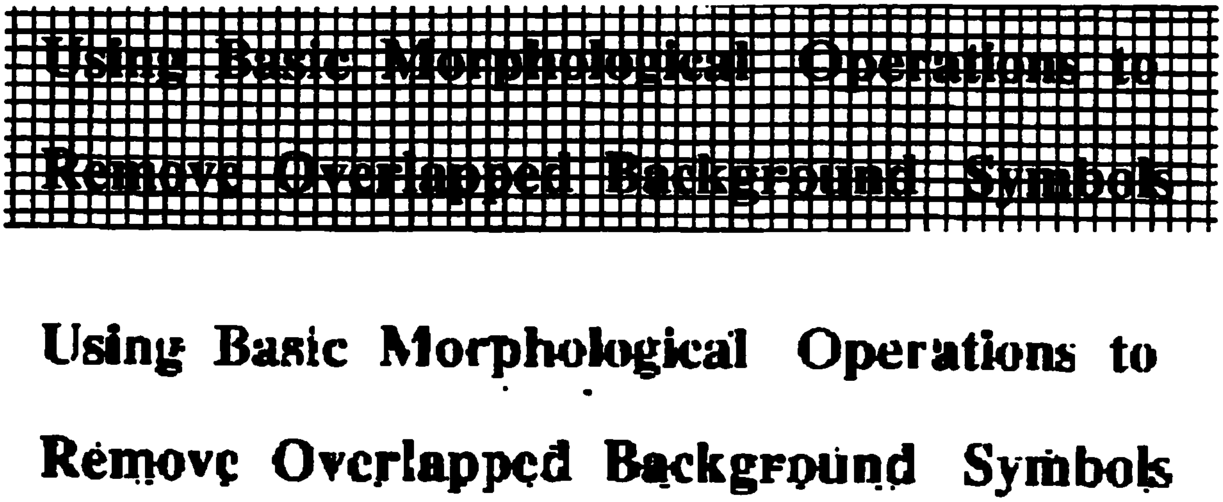
(h)


Using Basic Morphological Operations to
Remove Overlapped Background Symbols
Using Basic Morphological Operations to
Remove Overlapped Background Symbols

(i)


IJ,%ing Basic Morphological Operations to
Reinove Overlapped Background Sy'mbols

(j)



(k)

··········· «11311111111111 .111111111111111111111111

AII I1

mom Sim mill I II - Sir

IM -F "61 %r *ARAI I* I I -T IF *arllwin* I I

tjsing Basic Iklorphok)gicLil Oper:Ation;s to
R@move Overtappcdkgiround SyibboLs

(l)

Figure 2.15 The character images extracted from different text/background images
(a) (c) (e) (g) (i) (k) Text images with and without overlapping periodic background symbols
(b) (d) (f) (h) (j) (l) Recognition results of commercial OCR of (a) (c) (e) (g) (i) (k)

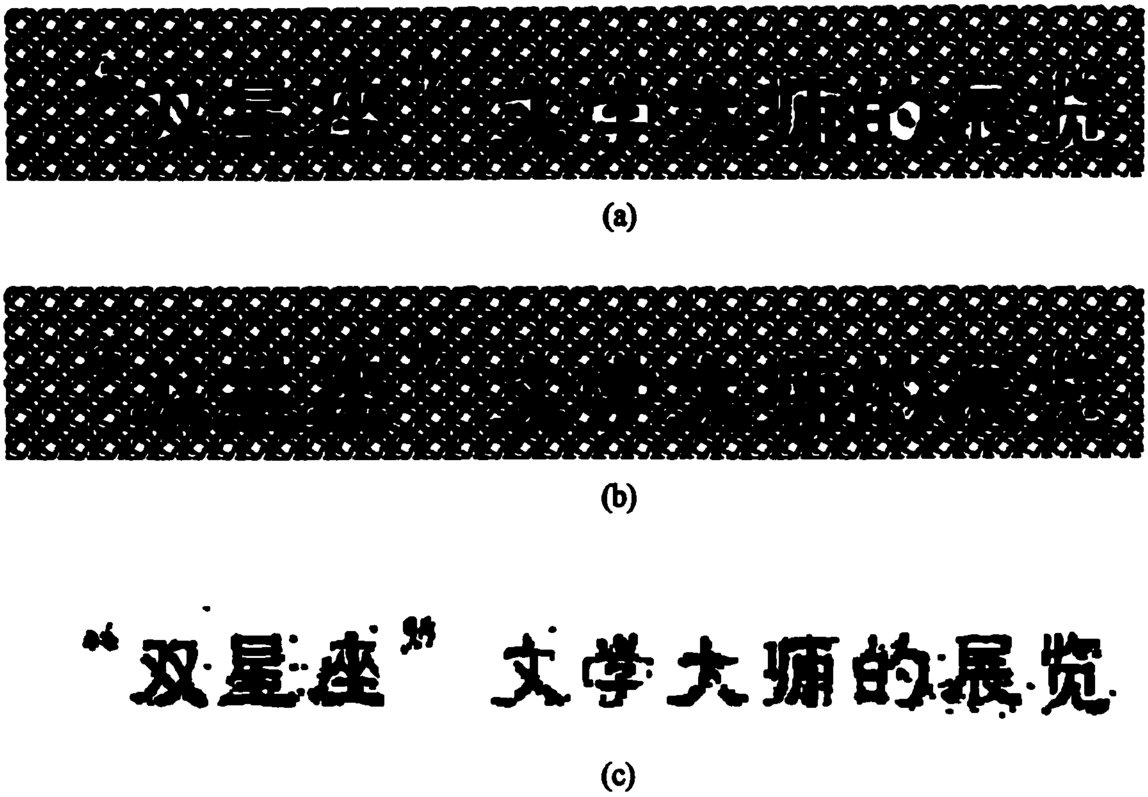


Figure 2.16. The character images extracted from a text/background headline image scanned from a Chinese newspaper

(a) Original headline image

(b) Modified headline image

(c) Final text extraction

2.5 Conclusions

A new algorithm has been developed for text string extraction from overlapping text/background images. The performance of the new scheme was tested on six artificial images and one real image, each with a different style of periodically distributed background symbols. Provided that the images conform to several constraints, the algorithm is effective and reliable. This new method is appropriate for implementation in document analysis systems. We focus only on the extraction of characters from the regular periodic overlapping backgrounds in this thesis, since these kinds of background symbols can be easily generated by computer publishing devices.

The uniform backgrounds can be considered as one sort of texture. However, most texture analysis techniques, such as statistical approaches, structural approaches, and spectral approaches [17], are effective and suitable for the grey level images. The above described algorithm, based on binary mathematical morphology, would be very effective for binarized document images.

Chapter III

Segmentation of Handwritten Interference Marks Using Multiple Directional Stroke Planes and Reformalized Morphological Approach

3.1 Introduction

Distorted document images are generated due to the physics of the apparatus for printing and imaging, including diffraction and aberration in the optical system, spreading and flaking of ink/toner, low print contrast, noise in electronic components, etc. Many effective algorithms for restoring distorted images have been developed so that high accuracy recognition rate may be achieved. However, document images that are frequently degraded due to human induced interference strokes that often cut across words are inadequate for processing by Optical Character Readers (OCR). For example, text overlapping with non-text strokes poses another kind of serious problem to automatic text recognition in many commercial applications. Human beings have little difficulty in differentiating meaningful boundary information and rejecting the strokes which are perceptually implausible. However, it is not easy for a computer to distinguish the ambiguous parts of text superimposed by interference strokes. The ruined text (Fig. 3.1) usually becomes unreadable and is rejected by text recognition systems. Therefore, it is necessary to perform a preprocessing procedure to extract text strings prior to recognition.

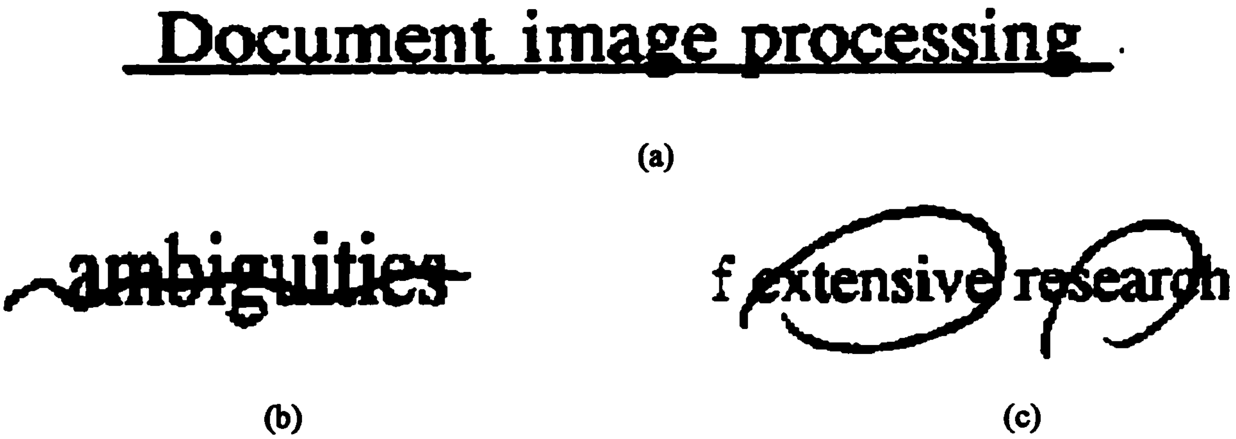
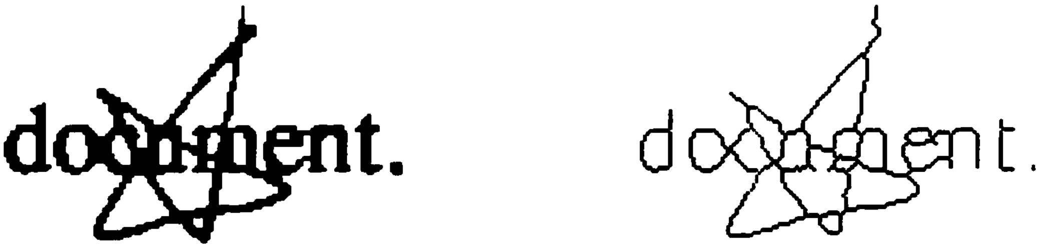


Figure 3.1 Interference strokes overlapping with machine printed text

In Chapter II, we have described a new approach to extract text strings from regular periodic overlapping text/background images with morphological tools. Govindaraju et al [43] discussed a preprocessing procedure to clean non-text strokes. They have assumed that after applying thinning algorithm, the curvatures of interference non-text strokes would be very small. However, obtaining reliable measures of curvature at a point in a thinned digital image is difficult because these skeleton images tend to be locally ragged. Yamada et al [77][78] developed directional morphological operations to calculate and extract the geometric lines on maps, such as roads and railways. However, one can still find some extracted curves with touching characters due to the difficulty of choosing suitable iterations of morphological operations. Guillevic et al[33] used morphological opening operations to remove underlines in automatic cheque reading systems. However, these simple opening operations will be ineffective when the cheque images are skewed.

To extract interference lines and curves from a binary image, two processes, thinning and extraction of orientation attribute of segments, are of importance in pattern analysis. The thinning must preserve the connectivity and the topological features of the original image so that the geometric features such as critical points and strokes can be extracted from the skeletal image. However, after thinning, estimating the smoothness of a curve is quite difficult. However, the thinning process introduces artifacts which would severely distort the shape characteristic of the image (Fig. 3.2) Therefore, using thinning technique alone will fail to accomplish successful segmentation of interference strokes.



(a) (b)

Figure 3.2 Thinned line images with serious distortion

Extraction of directional segments [79] can be based on the original images without the thinning process. Distortion due to the thinning process can then be eliminated. However, this technique excludes the connectivity property of the components. In this Chapter, a new algorithm which combines two processes to effectively segment all "natural" lines, including both straight lines and smooth curves, is developed. Here, a natural line implies that it possesses smooth curvatures and a long length.

When dealing with binary images, a principal application of morphology is extraction of connected components in the images. For example, let Y represent a connected component and assume that a single point p of Y is known. A is a binarized raster image that contains Y and p. Then the following iterative expression yields all of the elements of Y:

$$X_{k+1} = (X_k \oplus B) \cap A \qquad k = 0, 1, 2, 3, \ldots \tag{3.1}$$

The dilation described by Eq. (3.1) can be explained as the following process. Let $X_0 = p$ denote the initial condition. The single point X_0 is dilated using structuring element B to generate a subset binary image ($X_0 \oplus B$). The dilated elements of X_0 (i.e. $X_0 \oplus B$) is then intersected with the binary image A (i.e. $(X_0 \oplus B) \cap A$) to generate the dilated partial elements denoted as X_1. Similarly, given a dilated image set X_k after k dilation iterations, the resulting binary image, X_{k+1}, is the intersection of A and the dilated binary image $X_k \oplus B$. The dilation process continues until the entire connected component Y contained in A is extracted. Figure 3.3 shows an example of connected component extraction. The process is similar to the "flooding water". When a dam is broken, the water will pour down into the low ground in all directions to cause flooding unless there is a watershed or waterway to prevent flooding.

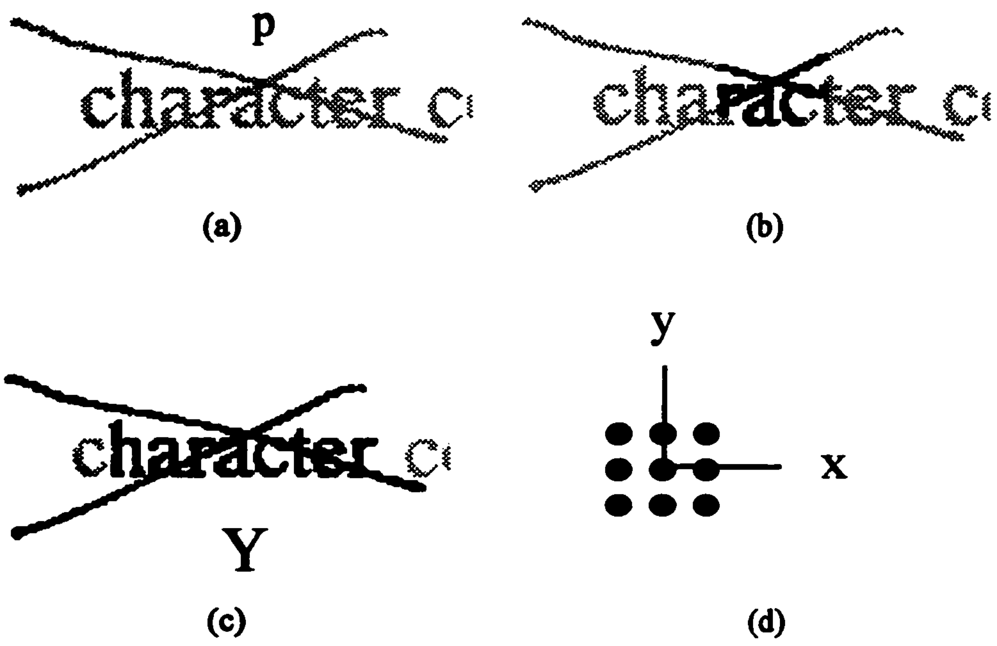


Figure 3.3 An example of connected component extraction using conventional dilations
(a) a black pixel p of image A
(b) conditional dilation after 20 iterations with a 3x3 square structuring element
(c) conditional dilation after 90 iterations with a 3x3 square structuring element
(d) a 3x3 square structuring element B

Morphological operations based on multiple direction planes and skeleton images are used to successfully prevent the "flooding water" effect of conventional morphological operations. In the Section 3.4, we will discuss the construction of a "waterway" in the touching text/interference stroke images. The effectiveness of the proposed algorithms is demonstrated by tests on several experimental images that possessed degrading interference strokes. The system flow chart is given in Figure 3.4. Detailed algorithms will be described in the following sections.

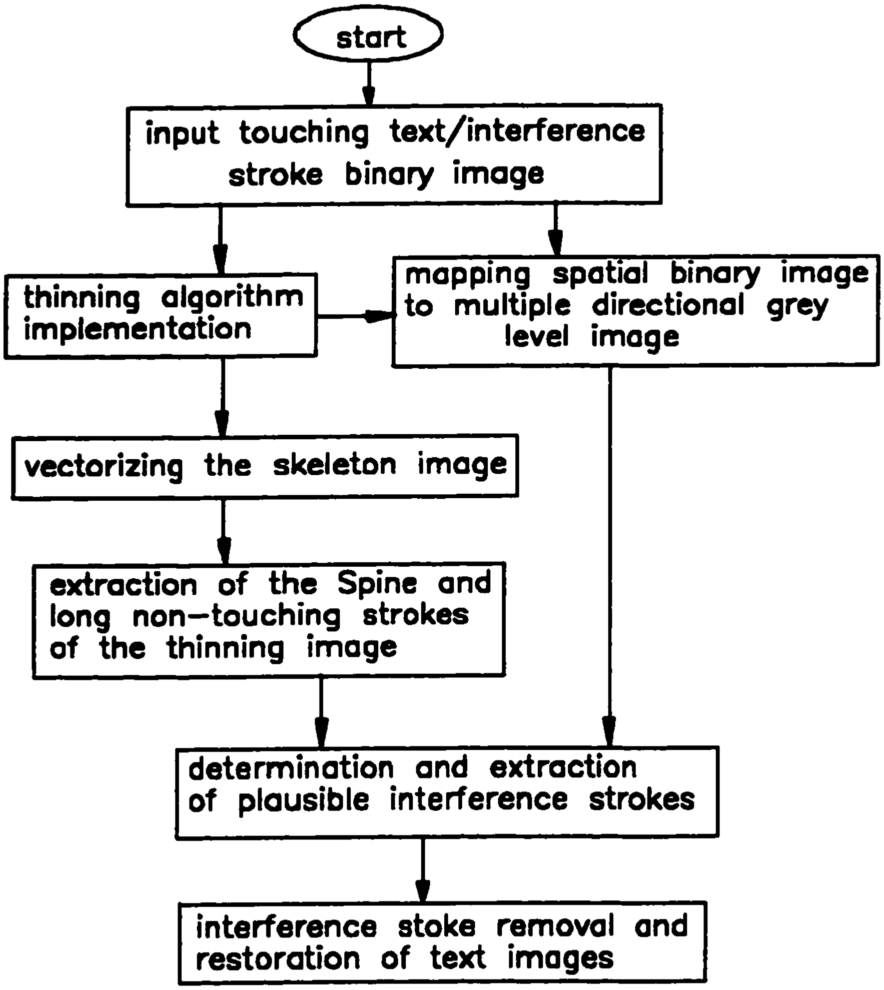

start
input touching text/interference stroke binary image
thinning algorithm implementation
mapping spatial binary image to multiple directional grey level image
vectorizing the skeleton image
extraction of the Spine and long non–touching strokes of the thinning image
determination and extraction of plausible interference strokes
interference stoke removal and restoration of text images

Figure 3.4 The system diagram

3.2 Generation of Topological Features

The skeleton of a region preserves the connectivity of the binarized raster images. Therefore, we utilize the thinning process to vectorize the topological structures of the original text images. A significant aspect of the algorithm is to identify and vectorize line or curve strokes in binary skeleton images that are represented by nodes connected by paths. For example, one thinned curve or straight line can be extracted by two known end nodes with a distinct path without any shape information of the curve or straight line. The methods used to generate a skeleton vary from one thinning algorithm to another[80][81]. An effective algorithm developed by Chu and Suen et al[80] was chosen for this study and the results are founded to be quite smooth and noise insensitive. Figure 3.2 shows the result after applying the thinning algorithm. To simplify the description of our algorithms, we introduce the following definitions:

Definitions:

End-node: A pixel X is an end-point if it has a value of 1 and has only one 8-neighbour point with value 1.

Junction-node: A pixel x is a junction-node if it has a value of 1 and has at least three 8-neighbour points with value 1.

Path: A path from node A to node B is a sequence of branches that one must pass in order to go from A to B.

Degree: The degree of a node equals the number of branches incident to it.

Spine: The Spine of a skeleton image is a longest length from end-point A to end-point B for one connected component.

3.2.1 Extraction of the Spine of a Thinned Image

Figure 3.5 illustrates the above definitions in a given thinned image. The Spine of thinned images is considered as most possible components of the interference strokes. Therefore, extraction of the Spine of a thinning image is a very critical step in removal of interference strokes.

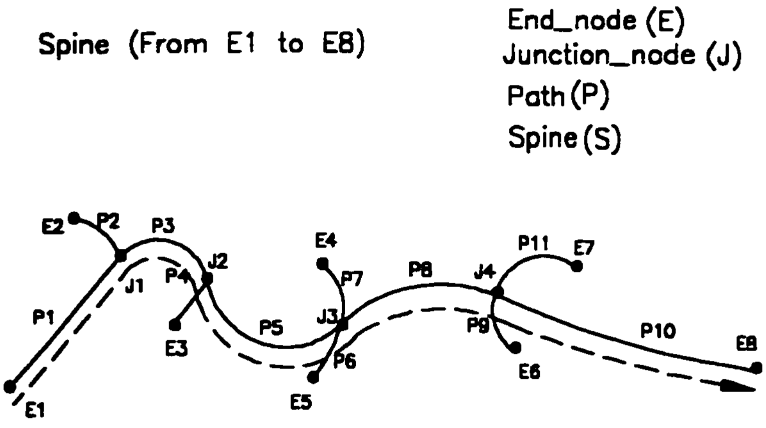


Figure 3.5 Illustrations of End-nodes, Junction-nodes, Paths, and a Spine

The extraction of the spine of a thinned image can be described as following steps:

Extr_Spine(X):

step 1: Generating the Junction-nodes and End-nodes for a specific thinned image X based on above node definitions.

step 2: Label_All_Path(X), labelling all paths of thinned image X including the paths from Junction-nodes to Junction-nodes and Junction-nodes to End-nodes.

step 3: Cal_Jun_End(A,B), calculating the distance from each junction-node to those end-nodes connected to it; A and B denote an Junction node and its End-node respectively. After the labelling procedure, calculation of distances between nodes to nodes can be easily implemented by counting the same labelling numbers starting from one node and ending at another.

step 4: Del_Jun_End(A,B), removing the paths between the junction-node to its end-nodes if the distances are less than a threshold T1; A and B denote a Junction node and its End-node respectively. The threshold T1 is an empirical value determined by the information related to the specific document.

step5: Extr_Spine(Y), extracting the spine of the thinned image Y by merging the remaining branches; Y denotes the thinned image after applying step 3 and step 4. After merging the remaining labels in the binary image, the Spine for a given connected component having the same labelling numbers is then extracted.

one interference strokes as shown in Figure 3.3. The mapping of those multiple interference strokes located in the same text region into several binary images in terms of their directional attributes in the original binarized raster image is a issue that will be discussed in section 3.3.

3.2.2 Extraction of the Long Paths of a Thinning Image

The removal of handwritten interference strokes is based on assumptions that interference strokes have long lengths and smooth curvatures. The algorithm (Extr_Spine(X)) discussed in the previous section is effective to extract those longest skeletal interference strokes (i.e. Spine). However, we cannot preclude cases that the interference strokes may not start from or terminate at End-nodes as the constraints set by Extr_Spine(X). As a result, those interference strokes are ignored by the algorithm Extr_Spine(X). Therefore, it is necessary to develop an algorithm to extract those large branches featured as interference strokes contained in thinned images. Figure 3.6(f) shows the extraction result of a long path generated only by two Junction-nodes. The algorithm for extracting the large branches of a given thinned image can be described by the following steps:

Extr_Long_path(X):

step 1: Generating the Junction-nodes and End-nodes for a specific thinned image X based on the node definitions.

step 2: Label_All_Path(X), labelling paths of a thinned image; X denotes a thinned image.

step 3: Cal_Nodes(A, B), calculating the distance from each node to all nodes connected to it; A and B denote the Junction-node or End-node.

step 4: Del_Path(A, B), removing the path between the node A to node B if the distance is less than threshold T2. The threshold T2 is an empirical value determined by the information related to the specific document.

step5: Extr_Long_Path(Z), extracting the large branches of the thinning image Z; Z denotes the thinned image after applying step 3 and step 4.

The algorithms Extr_Spine(X) and Extr_Long_Path(X) described in this section are based on the hypothetical thinned image containing only one Spine to illustrate the whole procedure in a more understandable manner. In practice. multiple interference strokes might be located in the same text region. In order to prevent the failure of our algorithms, those multiple interference strokes in the same text regions should be separated or mapped to different binary images. How do our algorithms use the directional attributes of the binary image to implement such a mapping process? Meanwhile, based on the discussion so far, we still have not provided a method to overcome the "flooding effects" of basic morphological operations by directly dilating on those successfully extracted skeletal interference strokes. These questions will be answered in section 3.3.

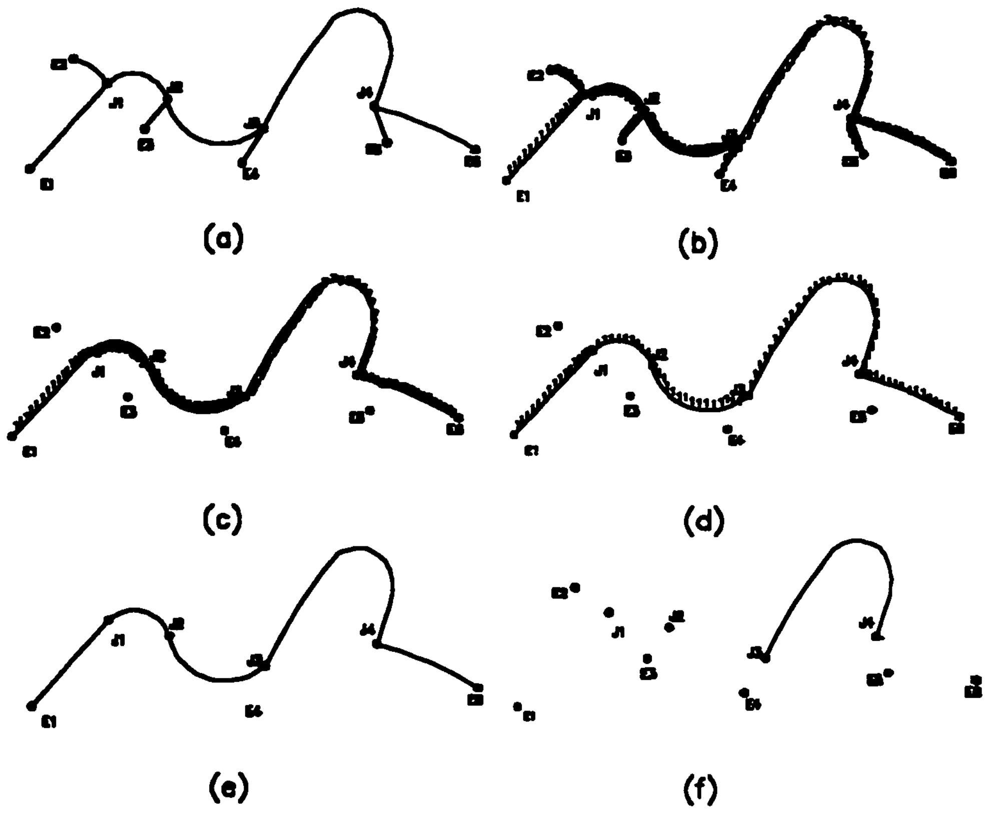


Figure 3.6 Illustrations of the algorithms Extr_Spine(X) and Extr_Long_Path(X)
(a) An original thinning image
(b) Labeling branches based on End-nodes and Junction-nodes
(c) Removing short End-node related branches
(d) Merging the labels for extracting the Spine of the thinning image
(e) The 'Spine' extracted by the Extr_Spine(X)
(f) Extracting the long paths based on End-nodes and Junction-nodes

3.3 Mapping Spatial Binary Image to Multiple Directional Grey Level Image

As mentioned in the previous section, the Spines of thinning images are not adequate to represent the multiple interference strokes in a specific text region. In this section, we will discuss how to map those multiple interference strokes located in the same text region into multiple binary images containing only one Spine according to their orientation attributes.

3.3.1 Orientation Angle

Define an orientation line of a pixel p(r,c) as a straight line that passes through p(r,c) of a connected component and intersects two boundary points along the kth quantized orientation. $D_k(r, c)$ is the length of the orientation line that crosses p (r, c) in direction k between two boundary points, where $k = 1, 2, 3, ..., M$. M is the number of directions that generates a uniform partition of the [0, 180°) range. For example, $M = 60$ implies that $k = 1$ is equivalent to the direction of 3° and $k = 8$ is equivalent to the direction of 24°. The orientation line at the pixel p(r, c) $D_m(r,c)$ is defined as a largest value of $D_k(r, c)$. If $D_m(r,c) > D_k(r, c)$, $\forall$ k $\neq$ m, it means $D_m(r, c)$ is the largest length of all the orientation lines across p(r, c) point. The orientation angle m with respect to the vertical axis is defined as Orientation Angle of p(r, c). Figure 3.7(a) gives an example for illustration of the definition.

3.3.2 Ambiguous Points

An ambiguous point can be defined as a nonzero point a(r, c) which has at least two large values (empirically specified) of $D_k(r, c)$, k = 1,2,3,...M. The following equations are used to analytically define ambiguous points:

$$\text{Max\_}D_m(r, c) = D_m(r, c) > D_k(r, c), \ \forall\, k \neq m \tag{3.2}$$

$$\text{Sub\_Max\_}D_n(r, c) = D_n(r, c) > D_k(r, c), \ \ \forall\, k \neq m \text{ and } \forall\, k \neq m \tag{3.3}$$

where Max_$D_m(r, c)$ and Sub_Max_$D_n(r, c)$ are the maximum value and sub-maximum value of $D_k(r, c)$ at the point a(r, c) respectively. The point a(r, c) is called an ambiguous point if the following conditions are satisfied:

$$\text{Max\_}D_m(r, c) > \alpha \tag{3.4}$$

$$\text{Sub\_Max\_}D_n(r, c) > \beta \tag{3.5}$$

$$|n - m| > \Theta \tag{3.6}$$

where n and m are two integers that denote the angles with respect to Max_D_m(r, c) and Sub_Max_D_n.(r, c). α, β and Θ are suitable predefined thresholds.

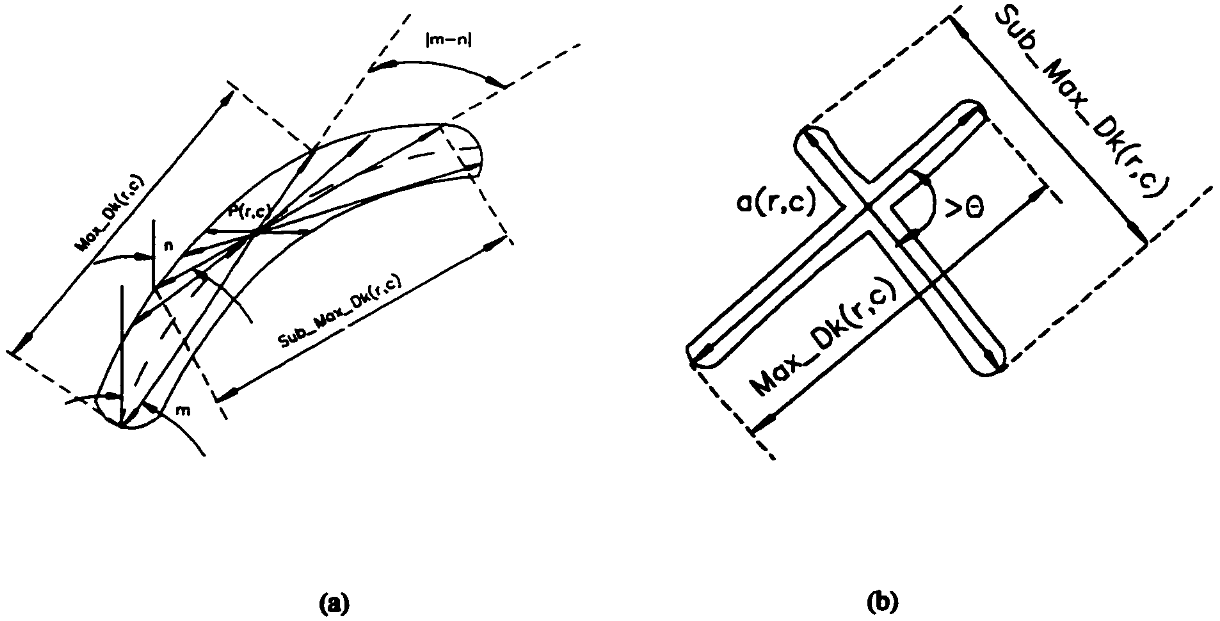


Figure 3.7 (a) The illustration of an orientation angle m at point p(r, c)
(b) The illustration of an ambiguous point a(r,c)
(Θ is a predefined constant)

Figure 3.7(b) illustrates the determination of an ambiguous point. Ambiguous points are generated at the intersections between interference strokes and characters or at the intersections inherently belonging to characters. These ambiguous points provide very critical information for recovering the characters crossed by the interference strokes. Accordingly, a continuous curve or line that cut across other connected components often generate several ambiguous segment points. These ambiguities provide clues for patching the broken characters or they provide auxiliary information for usage of higher level character analysis and recognition. However, not all ambiguous points (ambiguous points belonging only to characters) include useful information for restoration of broken characters. In section 3.5, we will discuss how to extract the critical ambiguous points to patch the broken characters.

3.3.3 Mapping to grey level images

This section describes the construction of an orientation map for a binary raster image. The orientation map is a grey-level image of the same size as that of the binary raster image. For each none-zero pixel p(r, c) in the binary raster image, we set the corresponding pixel g(r, c) in the orientation map to m assuming that the orientation angle of p(r, c) is m.

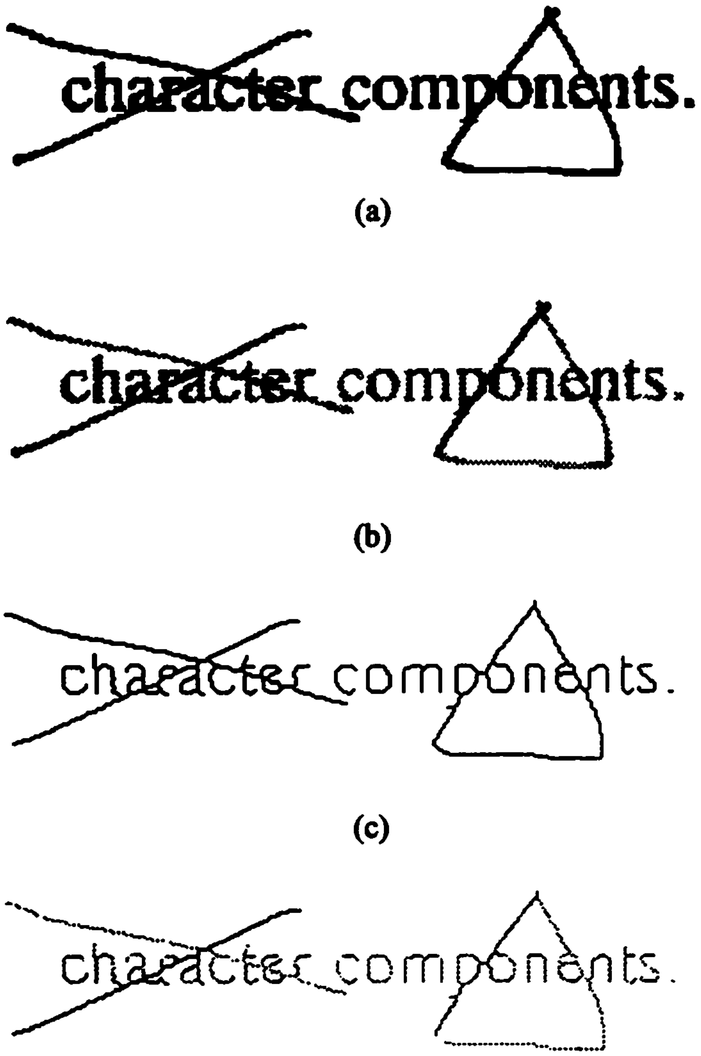
(i. e. g(r, c) = m, see Fig. 3.7(a)). Consequently, the binary image is then mapped to an image with pseudo grey levels. The value of g(r, c) indicates the orientation contribution of the connected component corresponding to the original binary raster image. For example, assuming that pixel p(5, 7) of the binary raster image has a non-zero value, and the orientation angle is m = 10. The corresponding pixel g(5, 7) of the orientation map is set to 10. Note that the maximum grey levels of the orientation map g is M which is the number of directions that generate a uniform partition of [0, 180°) range.

To calculate the grey level map of a thinned image, the following procedure is adopted:

1) Overlay the thinned image on the original image to which the thinned process is applied.

2) At each point t(r, c) on the thinned image, compute the orientation angle m of the original binary image p(r, c) and set the grey level map g_t (r, c) to m.

Figure 3.8(b) and 3.8(d) display the pseudo grey level images of both the original and thinned images.



(a)

(b)

(c)

(d)

Figure 3.8. (a) The original binary image
(b) The pseudo grey level image of Fig. 3.8 (a)
(c) The thinned image of Fig. 3.8(a)
(d) The pseudo gery level thinned image of Fig. 3.8(c)

3.4 Determination of Plausible Interference Strokes

Thinned images provide significant stroke information in terms of the connectivity property, since all skeleton branches can be easily vectorized by traversing these branches based on junction-nodes, end-nodes, and paths. Therefore, the use of thinned images facilitates the extraction of the 'Spine' components and the removal of some irrelevant noise branches. Moreover, the algorithms only take into account the length of the strokes. The advantage of the proposed algorithms using thinned image is insensitivity to stroke thickness.

After applying the above mapping algorithm to the thinned image, we obtain a pseudo grey level thinning image which includes directional information of the strokes. The total number of the grey levels of a specific orientation map depends on the orientation number M that uniformly partitions the [0, 180°) range. The M chosen as 60 in our study have provided sufficient resolution to determine most interference strokes. For a specific text block, the handwritten interference strokes only have a small number of orientations typically much less than 60. Since we assume that the interference strokes have smooth curvatures, the interference strokes distribute their orientation angles in a narrow range. Therefore, it is possible to cluster these grey levels into several ranges which can be used to binarize the grey level image into binary images. The thresholded images preserve the connectivity and orientation strokes of the original image. Obviously, using a fixed number of thresholds and projection planes is not an effective approach to obtaining satisfactory results, since apriori knowledge about direction of the interference strokes is not available in most cases. Therefore, the first procedure is to calculate the distribution of the grey levels of the thinned pseudo grey level image, and then locate the local maximal peaks of grey levels in the histogram. The grey level ranges (tl_i, tr_i, i = 1, 2, 3..) for binarizing the pseudo grey level images can be determined using the histogram maximum peaks (> t) by extending a constant range d in both side of the peaks. (see Figure 3.9) These parameters, tl_i and tr_i, for i = 1, 2, 3,.., are treated in modulo M, and will be utilized as tags for multiple binary image dilations later. In fact, multiple thresholding can be defined as the process of grouping parts of an image into coherent units that share the similar directional features. By correctly thresholding, large clusters of the direction branches will group into several specific binary images. (see Figure 3.10.)

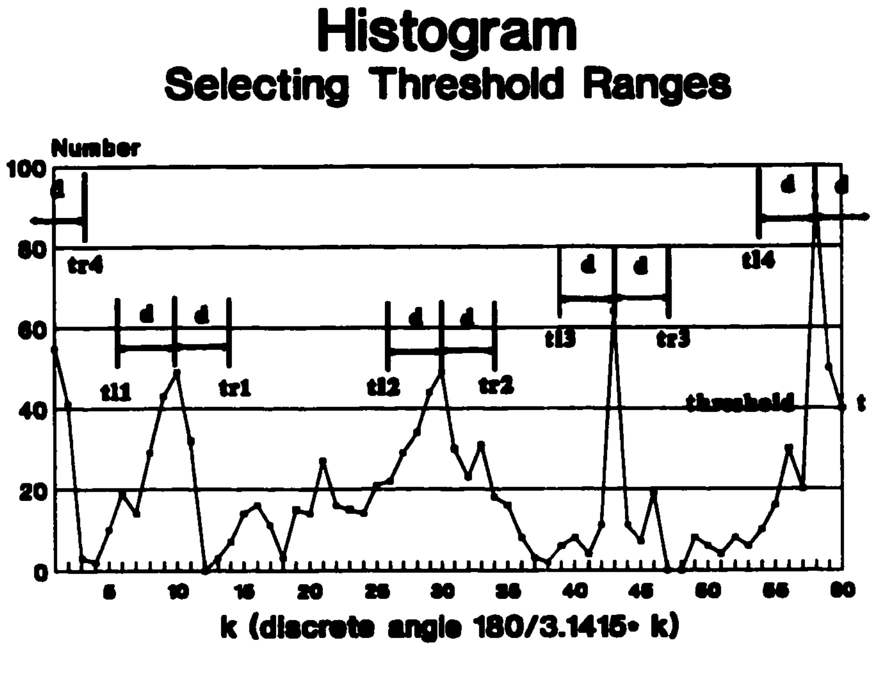


Figure 3.9 The histogram of the pseudo grey level of the thinning image Fig. 3.8(d)

The algorithm Extr_Spine(X) described in section 3.2 is applied to these binary images based on junction-nodes, end-nodes, and paths to extract the 'Spine'. However, some strokes which do not cross the character strings may have no junction points. For instance, the horizontal stroke in Figure 3.10(a) is ignored by the Extr_Spine(X) due to the lack of the junction-nodes. However, after applying the long stroke extraction algorithm Extr_long_path(X), this horizontal skeleton can be detected as an interference stroke according to the length of the stroke. Figure 3.10(j) gives the final results of extraction of the skeleton interference strokes after superimposing the interference strokes in different direction projection planes by using the algorithms Extr_Spine(X) and Extr_Long_Path(X).

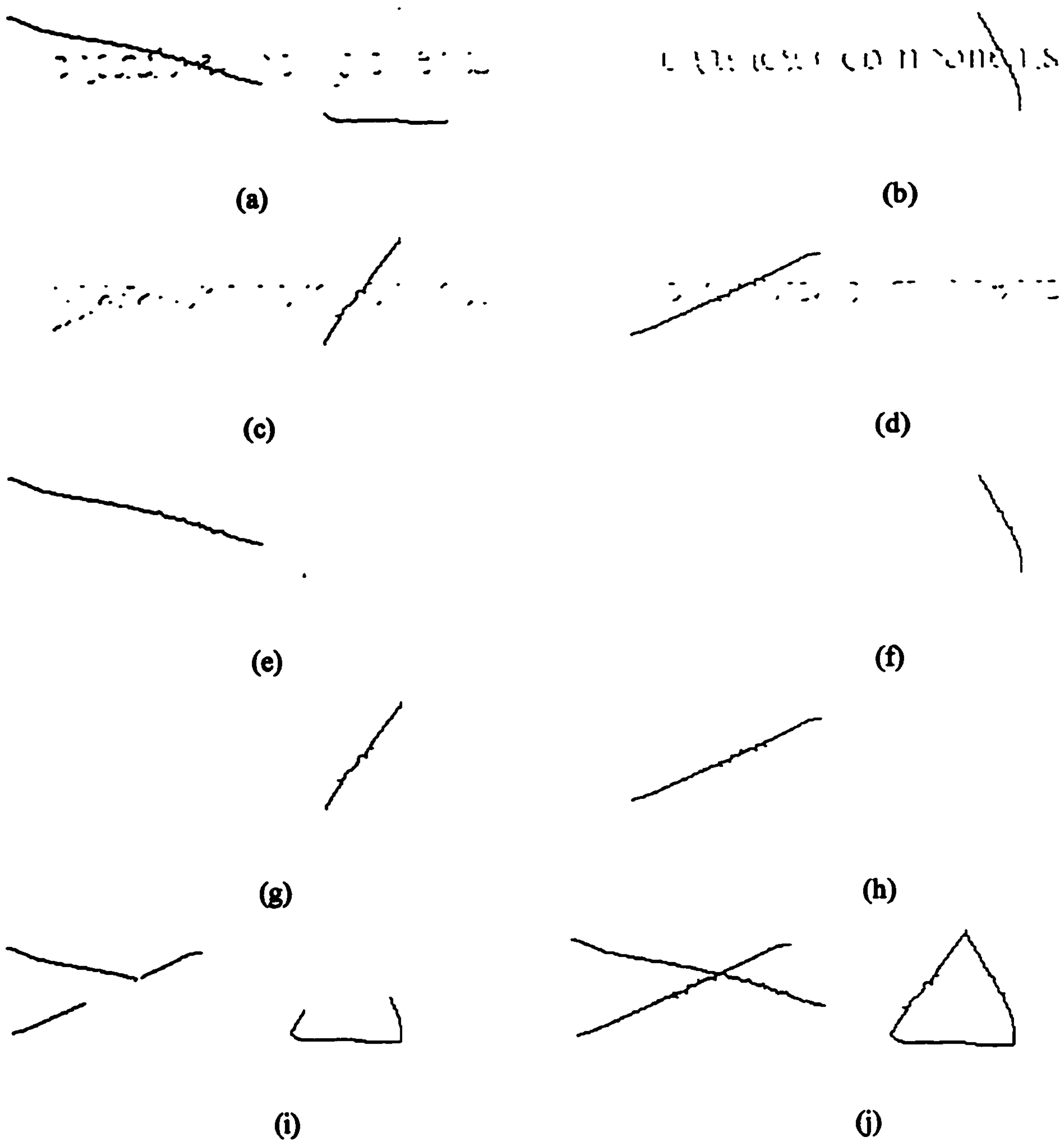


Figure 3.10 The procedures of extracting thinning interference strokes

(a)-(d) The thresholded pseudo grey level image Fig. 3.8(d) using tl_i and tr_i, $i = 1,2,3,4$

(e)-(h) The results after applying the algorithm Extr_Spine(X) to the images Figs. 10(a)-10(d)

(i) The result after applying the algorithm Extr_Long_Path(X) to the thinning image Fig. 3.8(c)

(j) The final result of extraction of the interfere strokes

3.5 Extraction of Interference Strokes

After applying the algorithms developed, all skeleton interference strokes, which are featured by long lengths and smooth curvatures, are successfully extracted. Then, the next procedure is the extraction of the interference stroke components using morphological operations. As mentioned previously in section 3.1, the conventional dilation operations have a 'flooding water' effect. Dilating an image causes the pixels to propagate in all directions and expands too many pixels to the regions of touching characters (see Figure 3.4(b) and 3.4(c)).

In order to perform a restricting dilation for a specific region, the thresholding tags (tl_i and tr_i, for i = 1, 2, 3,...) used to generate the binary thinned interference stroke images are also assigned to those specific skeletons to identify the ranges of the grey level where they are located. For example, the orientation map of the binary 'Spine' skeletons fall into the grey level range (20, 30), then dilation is applicable only on the pixels of the original binary image having grey level between 20 and 30 at the same locations of the orientation map. This constraint will prevent the dilation process from 'flooding' into the character areas that have different grey level ranges in their orientation maps. Figure 3.11(b) shows the dilation result with 20 iterations with a 3x3 square structuring element. Comparing the result of the Figure 3.4(b), we find that our 'waterway' successfully prevented the 'flooding' effects of conventional dilation operations.

3.6 Restoration of Damaged Characters

When an interference curve or line crosses a region containing text, the portions of the text which are overlapping with the interfering strokes are damaged. Therefore, removing interference stokes from the image creates ambiguities in the interpretation of the remaining part of the characters. The loss of the information is irreversible. The goal of restoration is to minimize the distortion after removing the interference strokes. In chapter II, we have described an effective algorithm to patch the character gaps with basic morphological operations in terms of the properties of periodic background symbols. We utilize the ambiguous points defined in section 3.2 instead of the background components to compensate for the loss of the character string pixels. The algorithm is implemented in following steps:

Step 1: Removal of the interference strokes from the image

The interference stroke removal operation (exclusive-OR (XOR) operation)

$$Y = XOR(X, I) \tag{3.7}$$

is performed on the original image set X (Figure 3.11(a)) and the interference stroke image set I (Figure 3.11(b)). The result Y denotes the character string image with interference stroke removal (Figure 3.11 (c)). Directly applying closing operation on the image Y to fill internal gaps of the character strings produces serious shape distortion. (see chapter II)

Step 2: Extraction of the critical ambiguous points

To improve the results of the morphological operations, the ambiguous points are used before the morphological closing operation is applied. Not all ambiguous points contain the valuable information to recover the character shapes. Only the ambiguous points located in both the character zone (Figure 3.11(d)) and the interference strokes are critical and are extracted by the following equation (Figure 3.11(e))

$$W = (A \cap I) \cap (A \cap C) \qquad (3.8)$$

Here A, I and C denote the ambiguous point set, interference stroke set and the character zone respectively.

Step 3: Operation for text string extraction

$$R = (W \cup Y) \bullet B \qquad (3.9)$$

B is a 3 X 3 square structuring element that is used to fill the narrow crack between the ambiguous points (in Figure 3.11(e)) and the characters with internal gaps (Figure 3.11(c)). R represents the final image (Figure 3.11(h)). Those text character pixels that do not overlap with interference stroke components should remain unchanged if the character shape distortion is to be minimized.

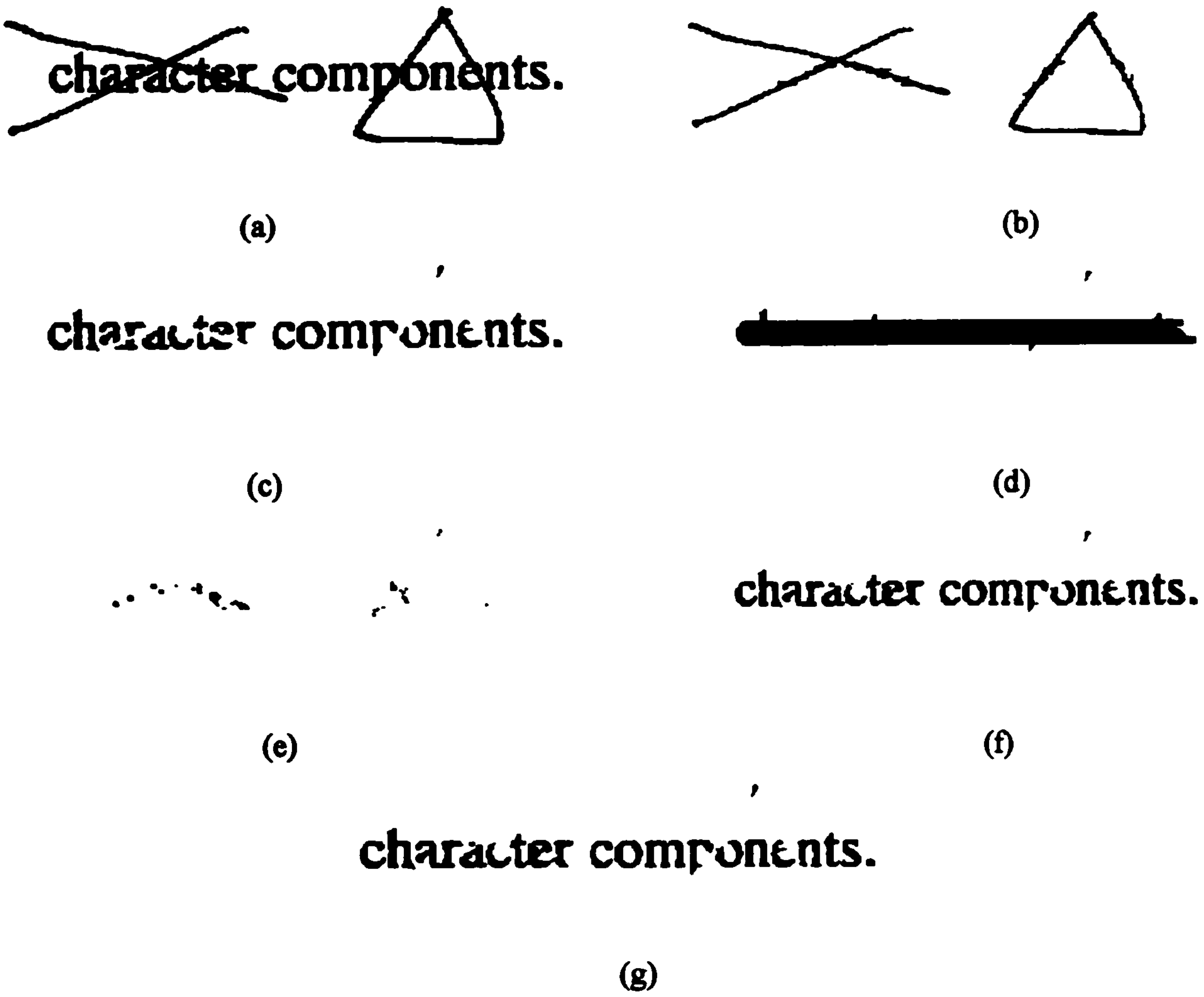


Figure 3.11. The procedures for extracting the text strings
(a) the original image (X)
(b) extraction of the interference strokes (I)
(c) the text image after removal of the interference strokes (Y)
(d) the character zone (C)
(e) the ambiguous points inside the character zone and interference strokes (W)
(f) patching the text strings with ambiguous points $(Y \cup W)$
(g) closing the text string images with small gaps inside (R)

3.7 Experimental Results

Twenty test images were obtained by a scanner interfaced to a Sun SPARC 10 workstation to test and evaluate the performance of the algorithms. The spatial resolution was set to 150 dpi. The progams were written in C language. We categorize the interference strokes into the following classes:

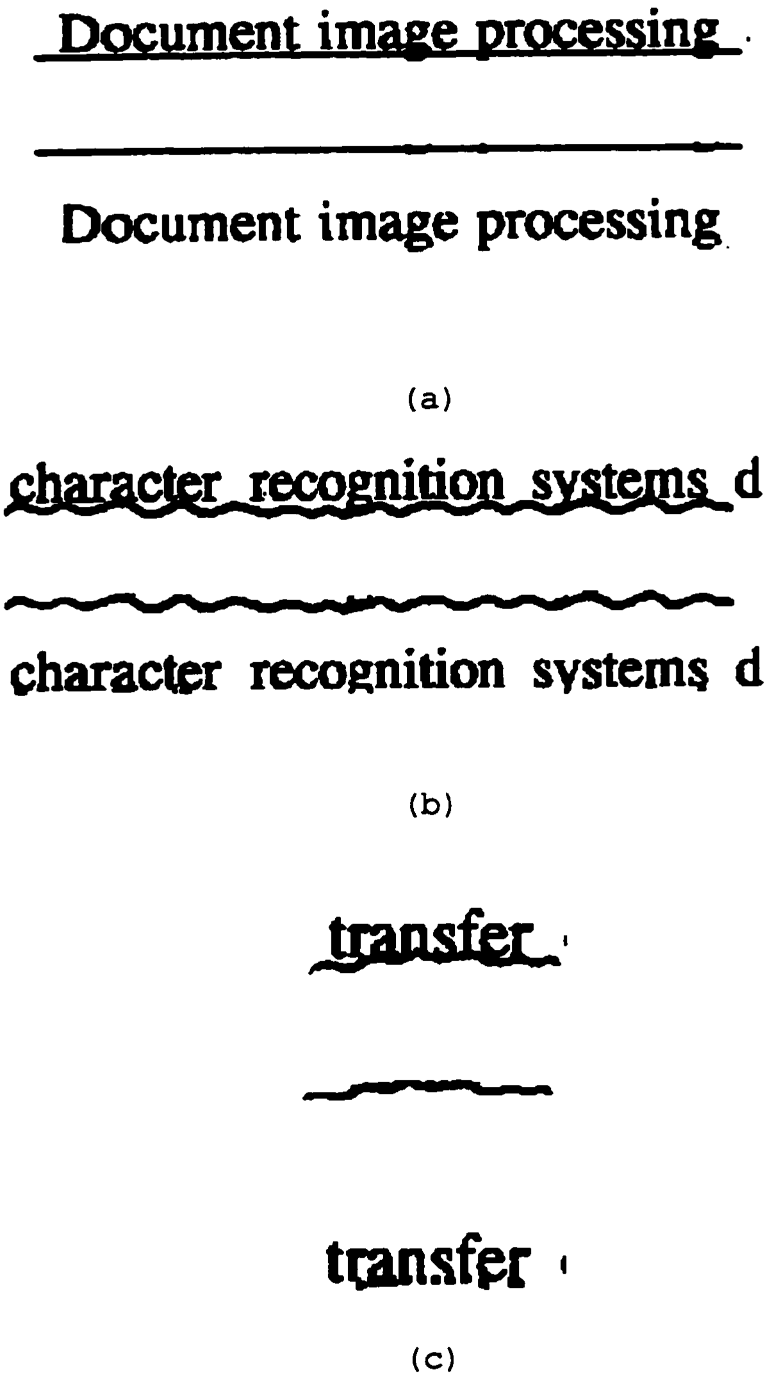
1) horizontal touching underlines which are either machine printed or hand drawing
2) straight lines with arbitrary angles.
3) curve strokes across words.
4) curves enclosing words.

Figure 3.12 shows the illustrative experimental results in terms of above interference classes. These test images demonstrate the algorithm's ability to extract text strings from touching text/interference stroke images. The commercial OCR software (HP DeskScan II WordScan 3.0) is applied to those test images. The OCR is able to recognize the machine printed underlined text, while total failures occur when interference strokes cut across characters. For example, for the text images in Figures 3.12(b), 3.12(c), 3.12(f) and 3.12(g), all text cut across by interference strokes are rejected by the OCR engine. After removal of the interference strokes, the recognition results turn out to be 'character recojznition systems d', 'ttunsfrr', 'ambiguities' and 'C@nmoy -B, Bose @ sh' in Figs 3.12(b), Fig 3.12(c), Fig 3.12(f) and Fig 3.12(g) respectively. We have compared the recognition results with and without the interference removal. The recognition rate 87% shows significant improvements after removing interference strokes based on conducted sample test images. The word "ambiguities" in Fig

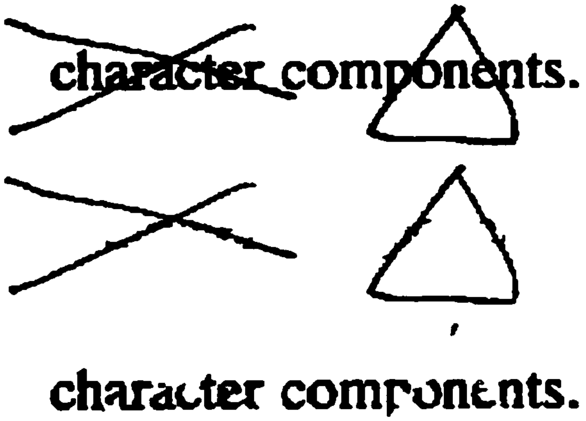
3.12(f) is correctly recognized by OCR with the built-in dictionary even when the characters 'a' and 'u' are broken after applying the proposed algorithm.

As mentioned in Chapter 2, morphological operations are time and memory consuming. For simplicity, similar discussions are omitted here. The memory requirement of the proposed algorithm include six two-dimensional arrays (6 x image_row x image_column bytes) to store the original image, the thinned image, the grey level images of both original and thinned images, two image buffers for morphological operations and the program itself. In the experiment conducted, the test image size is 100 pixels by 250 pixels obtained with 150 dpi and 5 x 5 structuring elements were used in the algorithm. It took 40 seconds to finish the interference stroke removal algorithm on SUN SPARC 10 work station.

The proposed algorithm is applicable to a class of interference strokes characterized by a relatively large perimeter and small curvatures. Under these conditions, the experimental results show that the proposed algorithm is effective. However, the algorithm may not yield satisfactory results, if the interference curve is nearly circular. Also if the size of the interference curve is small, the algorithm may fail. Meanwhile, since the interference strokes may occur anywhere in small regions on the document image, detection and extraction of those interference strokes can be implemented by widely used image processing techniques such as connected component labeling to reduce processing time in the application of document analysis.

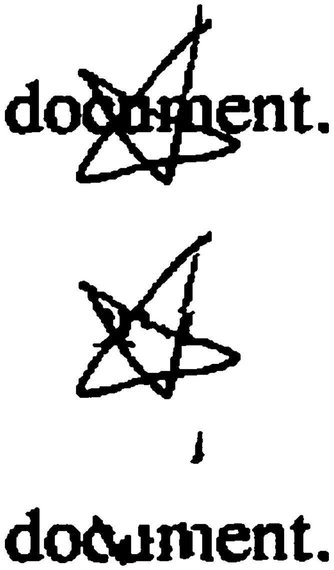


Document image processing
Document image processing
(a)
character recognition systems d
character recognition systems d
(b)
transfer
transfer
(c)



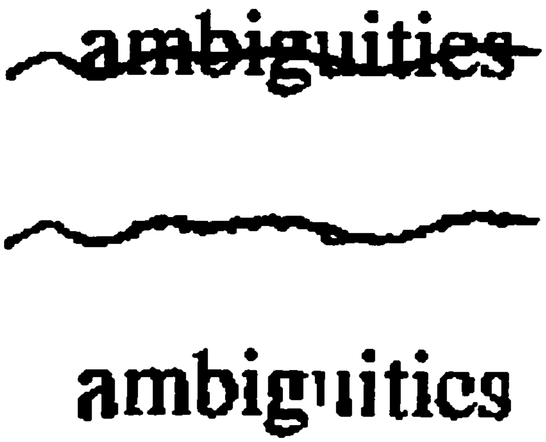
character components.
character components.

(d)

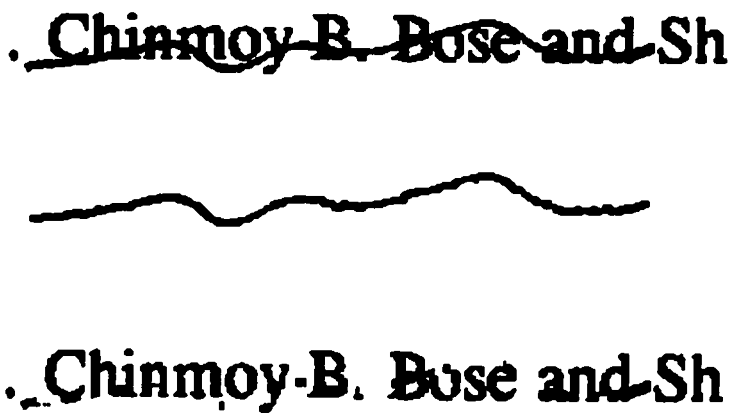


document.
document.

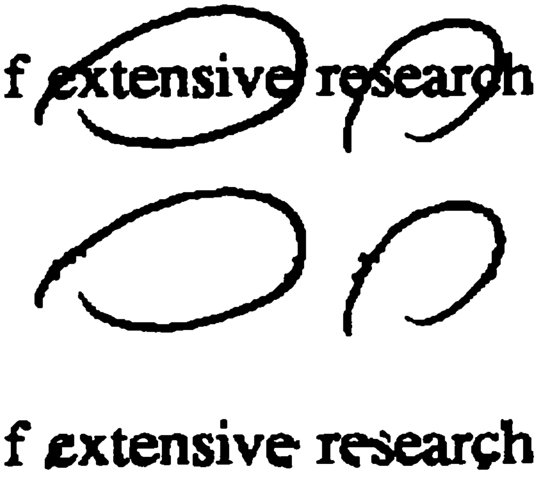
(e)


ambiguities
ambiguitics

(f)


Chinmoy B. Bose and Sh
Chinmoy B. Bose and Sh

(g)



(h)

Figure 3.12 (a)-(c) the horizontal touching underlines with machine printed and manual drawing
(d)(e) the straight touching lines with arbitrary angles
(f)(g) the smooth touching curves across words
(h) the enclosing touching curve

3.8 Conclusions

To extract interference strokes from a binary image, we have developed a new methodology that effectively segments not only straight lines but also smooth arbitrarily oriented curves. This segmentation technique combines the two processes of thinning and detection of directional strokes. These two processes operate complementarily, making the algorithms preserve both connectivity and directional characteristics. The morphological operations based on multiple direction projection planes and skeleton images are developed to successfully prevent the "flooding water" effect of conventional morphological operations. With improvements of the algorithm in terms of processing speed and efficiency of data representation, the algorithm will be highly appropriate for use in document analysis systems.

Chapter IV

SEGMENTATION OF MACHINE PRINTED TOUCHING CHARACTERS

4.1 Introduction

Today, character recognition systems dramatically facilitate the transfer of information into computer systems without intensive manual keying. Kahan, Pavlidis and Baird [54] suggested that in practical applications, a document recognition system is required to read texts accurately with at least a 99.9% recognition rate. It is not difficult to design a character recognition system that recognizes well-formed and well-spaced printed characters. However, it is a challenge to develop a system which can maintain such a high recognition rate, regardless of the quality of the input documents and the character fonts. Presently, most recognition errors are due to character segmentation errors [2][7][12][83][84]. Very often, even in printed text, adjacent characters are touching, and may exist in an overlapped field. Therefore, it is essential to segment a given word correctly into its character components. Any failure or error in this segmentation step can lead to a critical loss of information from the document.

Several investigators have attempted to develop techniques for properly segmenting words into their character components. Kahan, Pavlidis, and Baird [83] discussed the segmentation of touching characters. Tsujimoto and Asada [83] constructed a decision tree for resolving ambiguities in segmenting touching characters. Casey and Nagy [60] proposed a recursive segmentation algorithm for touching characters. Chinmoy B. Bose and Shyh-shiaw Kuo [85] applied the Hidden Markov Model to the touched and degraded text recognition.

In this Charpter, we propose a dynamic recursive segmentation algorithm for segmentation of touching characters based on the method proposed by Casey and Nagy [60]. In this process, several candidate cutting points are determined from the proposed discrimination functions; the algorithm then iteratively implements a forward segmentation or a backward merge procedure based on the outputs of a character classifier which operates on the components generated by segmentation algorithms. Using contextual information including a spell checker, further improvement in word recognition are achieved. Extensive studies with different text documents have attested to the feasibility of the proposed algorithm. The algorithm and test results are discussed in the following sections.

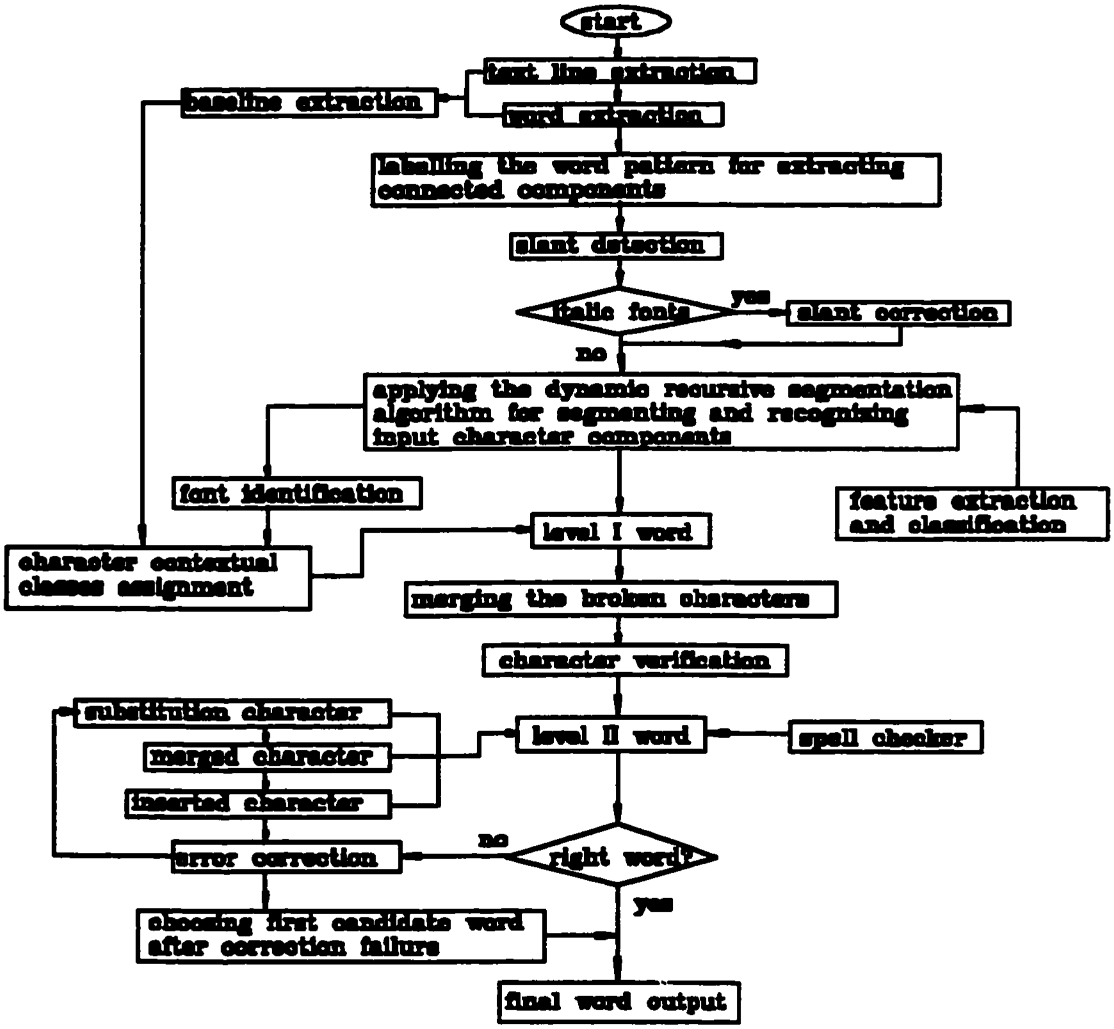

start
text line extraction
baseline extraction
word extraction
labelling the word pattern for extracting connected components
slant detection
italic fonts
yes
slant correction
no
applying the dynamic recursive segmentation algorithm for segmenting and recognizing input character components
feature extraction and classification
font identification
character contextual classes assignment
level I word
merging the broken characters
character verification
substitution character
merged character
inserted character
level II word
spell checker
no
right word?
error correction
yes
choosing first candidate word after correction failure
final word output

Figure 4.1 Recognition system diagram

```
                               111                    222
                              11111                222222
                              11111                 222222
                               1111                 222222
                                1           3       22222
                                           333       22222
                                          3333       22222
                                          3333       2222
                                  4      333333      22222  55555
6666666   7777777     8888     44444    333333333    22222222222222
6666666   7777777     8888   4444444    3333333      2222222 222222
666666     77777      8888   4444444     33333       22222     2222
 66666     77777      8888    444444     33333       22222     22222
 66666      7777      888      44444     33333       2222      22222
  6666      77777     888      44444     33333       2222      22222
  66666     77777     88       44444     33333       2222      22222
   6666     77777    888       44444     33333       2222      22222
   66666    777777   888       44444     33333       2222      22222
   66666   7777777   88        44444     33333       2222      22222
    6666  777777777 888        44444     33333       2222      22222
    66666 777 77777777         44444     33333       22222     22222
    66666666   7777777         44444     33333       22222     22222
     6666666   7777777         44444     33333      22222      22222
     6666666    77777          44444     33333      222222     22222
      66666     77777          44444     33333      222222     22222
      66666     7777           44444     33333   9  222222     222222
      6666       777           444444    333333 9999999999    2222222
       666       777         444444444    3333333333333333333333333333
       66         7           44444444     333333    333         33
                                            33
```

(a)

```
                               111                    222
                              11111                222222
                              11111                 222222
                               1111                 222222
                                1           2       22222
                                           222       22222
                                          2222       22222
                                          2222       2222
                                  4      222222      22222  22222
6666666   6666666     6666     44444    222222222    22222222222222
6666666   6666666     6666   4444444    2222222      2222222 222222
666666     66666      6666   4444444     22222       22222     2222
 66666     66666      6666    444444     22222       22222     22222
 66666      6666      666      44444     22222       2222      22222
  6666      66666     666      44444     22222       2222      22222
  66666     66666     66       44444     22222       2222      22222
   6666     66666    666       44444     22222       2222      22222
   66666    666666   666       44444     22222       2222      22222
   66666   6666666   66        44444     22222       2222      22222
    6666  666666666 666        44444     22222       2222      22222
    66666 666 66666666         44444     22222       22222     22222
    66666666   6666666         44444     22222       22222     22222
     6666666   6666666         44444     22222      22222      22222
     6666666    66666          44444     22222      222222     22222
      66666     66666          44444     22222      222222     22222
      66666     6666           44444     22222   2  222222     222222
      6666       666           444444    222222 2222222222    2222222
       666       666         444444444    2222222222222222222222222222
       66         6           44444444     222222    222         22
                                            22
```

(b)

```
                               111                     222
                              11111                 222222
                              11111                  222222
                               1111                  222222
                                1            2       22222
                                            222       22222
                                           2222       22222
                                           2222       2222
                                  1       222222      22222  22222
6666666   6666666     6666     11111     222222222    22222222222222
6666666   6666666     6666   1111111     2222222      2222222 222222
666666     66666      6666   1111111      22222       22222     2222
 66666     66666      6666    111111      22222       22222     22222
 66666      6666      666      11111      22222       2222      22222
  6666      66666     666      11111      22222       2222      22222
  66666     66666     66       11111      22222       2222      22222
   6666     66666    666       11111      22222       2222      22222
   66666    666666   666       11111      22222       2222      22222
   66666   6666666   66        11111      22222       2222      22222
    6666  666666666 666        11111      22222       2222      22222
    66666 666 66666666         11111      22222       22222     22222
    66666666   6666666         11111      22222       22222     22222
     6666666   6666666         11111      22222      22222      22222
     6666666    66666          11111      22222      222222     22222
      66666     66666          11111      22222      222222     22222
      66666     6666           11111      22222   2  222222     222222
      6666       666           111111     222222 2222222222    2222222
       666       666         111111111     2222222222222222222222222222
       66         6           11111111      222222    222         22
                                             22
```

(c)

Figure 4.2 (a) First scan of component labelling ('w' has 3 labels, 'i' has 2 labels and 'th' has 4 labels)

(b) Second scan of component labelling ('w' has 1 label, 'i' has 2 labels and 'th' has 1 label)

(c) Final result of the labelled "with" ('w', 'i' and 'th' have 1 label)

4.2 Separation of Touching Characters

A typical text recognition system is shown in the flow chart in Figure 4.1. Since the work described here deals with segmentation of touching characters, it is assumed that a line of text with the appropriately labelled connected components in the line have already been extracted using appropriate techniques [23][34][86]. Figure 4.2 shows the connected components and their labels for the word 'with'. It is interesting to observe (Figure 4.2c) that the letters 't' and 'h' are touching.

4.2.1 The Modified Discrimination Function for Touching Characters

Among the many articles that have been published concerning the segmentation of touching characters [54][60][83][85], we were especially interested in the algorithm proposed by Kahan, Pavlidis and Baird [54]. They detected touching characters by using the ratio of the second order difference of the vertical pixel projection to the value of the vertical projection as an objective function. The cutting points for touching characters were obtained in the horizontal positions where the segmenting objective function was maximized. This method is able to cut most lightly touching characters; however, it was unable to separate heavily touching characters, particularly such as 'oo', 'oe' and 'od' etc, because the pixel projection waveforms for these kinds of touching characters vary gradually due to the lack of vertical strokes near the touching points. Figure 4.3 shows an example of touching characters 'oo'.

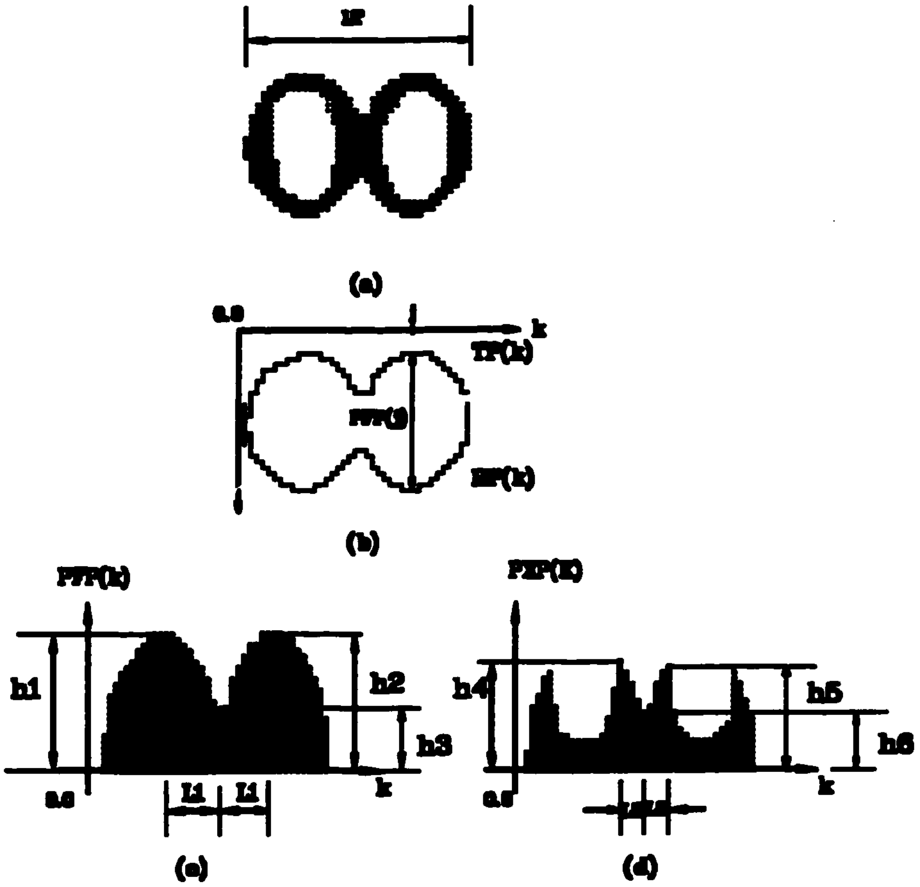


Figure 4.3 (a) Touching component 'oo'
(b) Top and bottom profiles
(c) The profile projection PFP(k)
(d) The pixel projection PXP(k)

In order to improve the segmentation process, the authors propose two discrimination functions based on pixel and profile projections. The pixel projection and profile projection are described as follows:

(1) The pixel projection is defined as

{ PXP(k), k = 1,2, ..., LT }. It consists of the total number of black "1" pixels in each vertical column. LT is the length of the touching characters.

(2) The profile projection is defined as

{PFP(k) = TP(k) - BP(k), k = 1, 2, ..., LT}. TP(k) is the top profile of the external contour of the touching characters as seen from the top; BP(k) is the bottom profile of the external contour of the touching characters as seen from the bottom.

Figure 4.3 illustrates these projections for the component 'oo'. Two segmentation discrimination functions based on the profile and pixel projections are then defined as follows:

$$F_1^{\alpha}(k) = \left(\frac{PFP(k+L1) - 2PFP(k) + PFP(k-L1)}{PFP(k)} \right)^{\alpha} \tag{4.1}$$

$$F_2^{\alpha}(k) = \left(\frac{PXP(k+L2) - 2PXP(k) + PXP(k-L2)}{PXP(k)} \right)^{\alpha} \tag{4.2}$$

where L1 and L2 in Eqs.(1) and (2) denote the distances between the current column and the adjacent columns L1 and L2. L1 and L2 are determined empirically based on the size of the characters in the documents. As an example for text with 'times' font of size 10, L1 and L2 were chosen as 8 and 4 respectively. α which represents the power of F1 and F2 is typically chosen to be an integer larger than 1. Figure 4.4 illustrates the effect of a on the values of the discrimination functions. Fig. 4.4(c) and Fig. 4.4(d) show the discrimination functions for $\alpha = 1$ and $\alpha = 2$ respectively. It is evident that the values of the discrimination functions are increased significantly with $\alpha = 2$, especially when F1 and F2 are larger than 2. Using suitable threshold candidate cutting points are derived for further processing. Fig. 4.4(c) and Fig. 4.4(d) show four candidate cutting points for the component 'oo' with a threshold of 11. It is noted that in this case, the correct cutting location is determined by the discrimination function $F_1^2(k)$ based on the profile projection. Figure 4.5 illustrates the process for the component 'th'. In this case, it is clearly seen that the discrimination function $F_2^2(k)$ based on the pixel projection yields candidate cutting points that contain the correct cutting location. We therefore define the discrimination function as follows:

$$F(k) = \begin{cases} F_2^2(k), & \text{if } F_1^2(k) > T \ \& \ F_2^2(k) > T \\ F_2^2(k), & \text{if } F_1^2(k) < T \ \& \ F_2^2(k) > T \\ F_1^2(k), & \text{if } F_1^2(k) > T \ \& \ F_2^2(k) < T \\ 0, & \text{if } F_1^2(k) < T \ \& \ F_2^2(k) < T \end{cases} \qquad (4.3)$$

Several cutting point candidates, including some false cutting points, were obtained at the locations where the values of the discrimination function were greater than a specific threshold. The optimal cutting points were found after several iterations by applying the dynamic recursive segmentation algorithm. Some broken characters caused by false cutting points can be merged using layout context information. This will be discussed in the following sections.

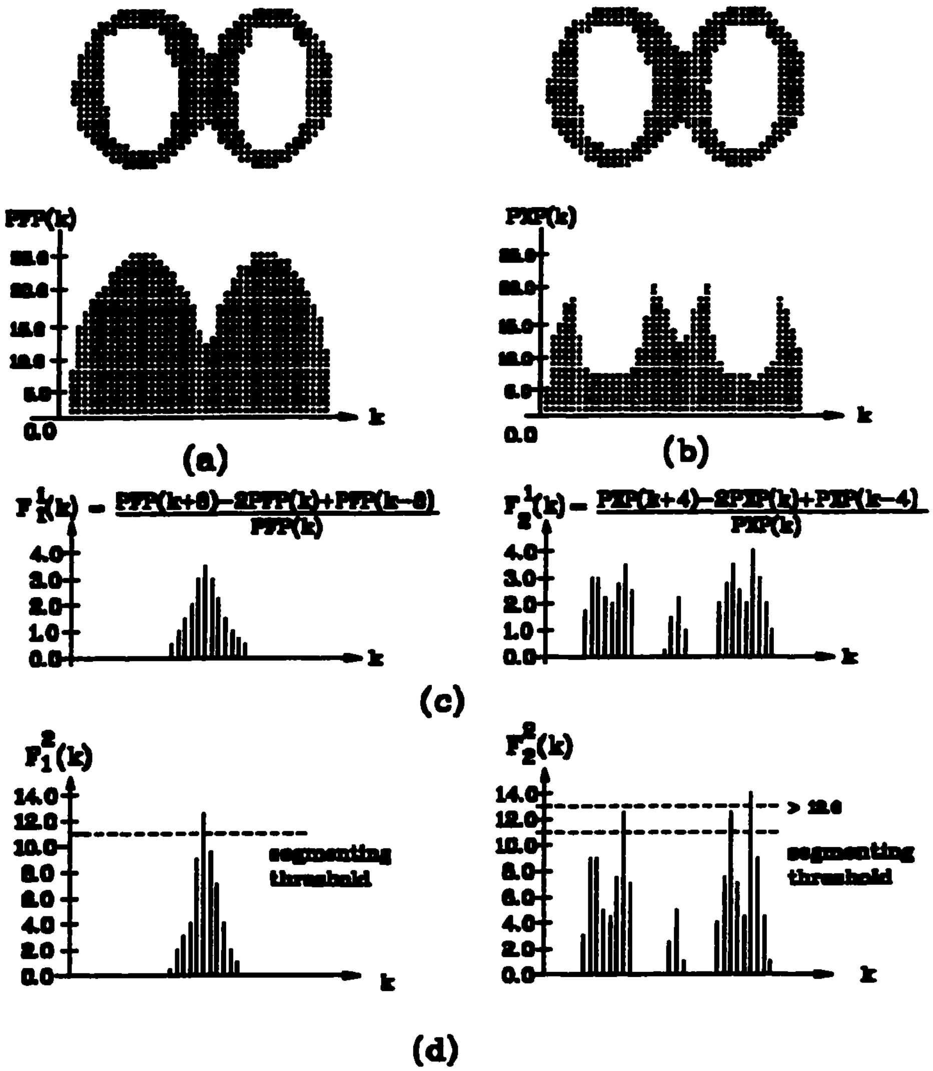
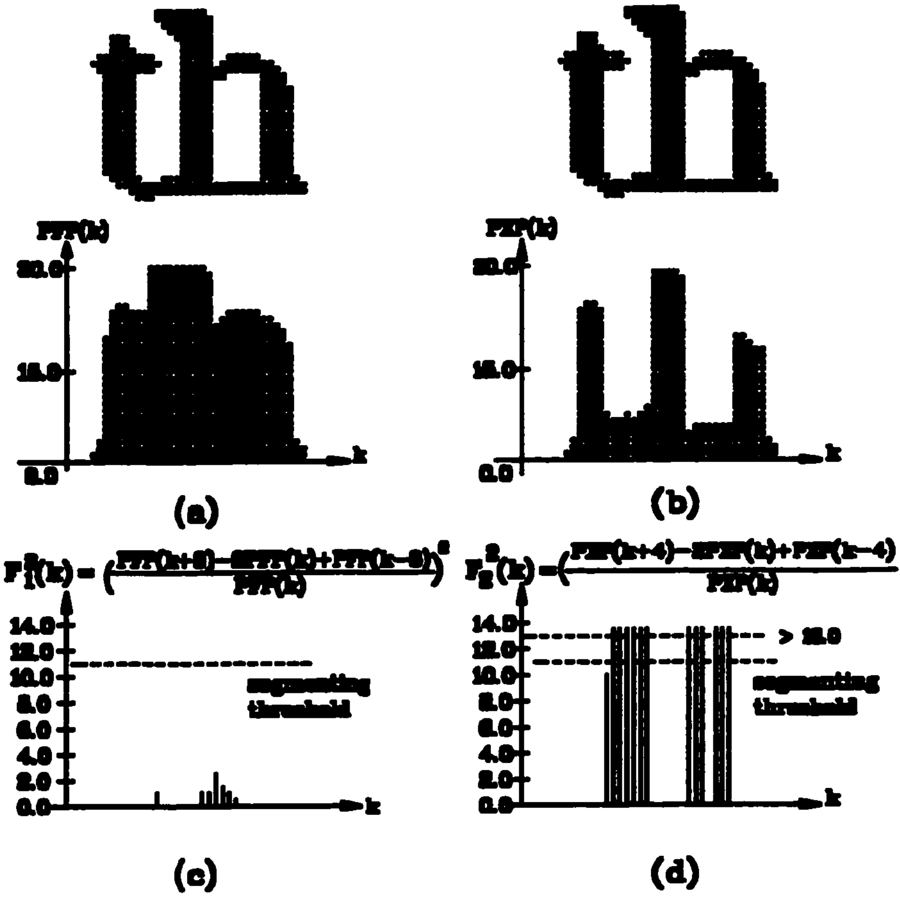


Figure 4.4. (a) The double-o touching character and its profile projection
(b) The double-o touching character and its pixel projection
(c) The discrimination function values based on the profile projection nd the pixel projection with $\alpha = 1$
(d) The discrimination function values based on the profile projection and the pixel projection with $\alpha = 2$



(a)

(b)

(c)

(d)

Figure 4.5 The serif touching character 'th'

(a) The profile projection

(b) The pixel projection

(c) The discrimination function based on the profile projection

(d) The discrimination function based on the pixel projection

4.2.2 The Dynamic Recursive Segmentation Algorithm

The recursive segmentation algorithm with adaptive windows, developed by Casey and Nagy [60], performed template matching after a suitable window was chosen. According to Casey and Nagy, if the classification failed, a different partitioning of the input pattern was tried again. However, even for a specific font, proportional spacing characters make it difficult to choose a suitable window on the first trial. Tsujimoto and Asada used a *break cost* to segment touching characters [83]. They constructed a decision tree and a set of additional rules to obtain the character component sequences. However, their algorithm required intensive computation to build a decision tree and search for a correct path; the more characters in a word, the larger the decision tree would be. Also their algorithm tended to accept several paths in the tree structure concurrently.

In our approach, we propose a dynamic recursive segmentation algorithm that implements a forward segmentation or a backward merge procedure based on the output of a character classifier operating on the components generated by the cutting points. This procedure terminates when the segments generated by a specific points are recognized by the character classifier with high confidence. Figure 4.6 and Figure 4.7 illustrate the algorithm for two types touching characters: the double-O 'oo' and the serif touching pair 'th', and the triplet 'thi'. Figure 4.6a shows the candidate cutting points determined by the discrimination function F(k). The first cutting point yields segments P1 and P2 which are sent to the character classifier. The classifier recognizes P1 as a valid 'c' while P2 is rejected as an invalid character. P2 is then segmented by the second candidate cutting point yielding P3 and P5. P3 is rejected by the classifier, while P5 is recognized as a valid 'o'. The segment P3 is then merged with segment P1 and the resulting segment P4 is correctly classified as a valid 'o'. At this point both the left and right segments are recognized with high confidence as valid characters ('o' and 'o' in this example). Figs. 4.6b, 4.6c, 4.6d illustrate the various stages of this technique.

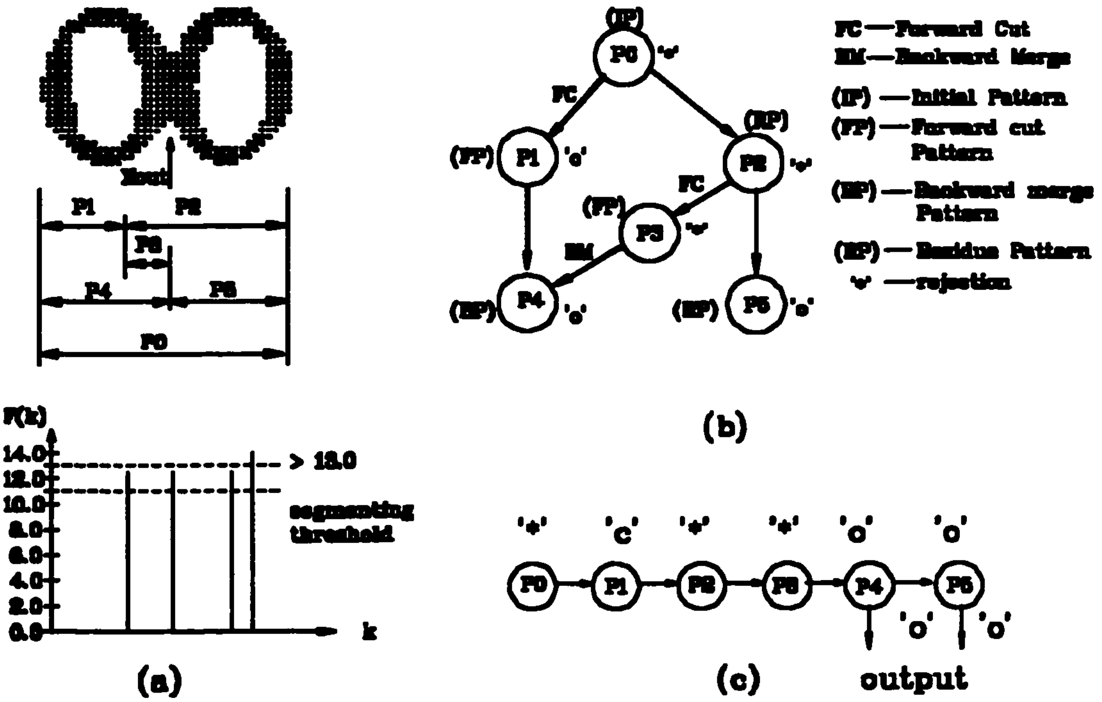


Figure 4.6 (a) The segmented patterns
(b) Graphic representation
(c) Time sequence of segmentation

Figure 4.5 shows 12 candidate cutting points for splitting component 'th' by using the discrimination function based on the pixel projection. In the segmentation process, even though cutting points sometimes deviate from the correct locations, the optimal cutting points X_{cut} are found after several forward and backward cutting iterations. Figure 4.7 illustrates the proposed algorithm for two sets of touching characters 'th' and 'thi'. The first case involving 'th' with two possible cutting locations is very similar to the case of 'oo' discussed earlier. The second case is interesting and brings out the effectiveness of the proposed algorithm. Due to the figure space limitation, only five candidate cutting points (Figs. 4.7c and 4.7d) are generated by the discrimination function:

(1) Initially the triplet labelled P0 is segmented into two components P1 and P2 of which only P1 is accepted as a valid character 'l' by the classifier.

(2) P2 is further segmented into P3 and P5, both of which are rejected by the classifier.

(3) P3 is merged with P1 yielding P4 which is accepted by the classifier as 't'.

(4) P5 is further split into P6 and P7 of which only P6 is accepted by the classifier as 'l' which P7 is rejected.

(5) P7 is split into two components P8 and P10, both of which are rejected by the classifier.

(6) P8 is merged with P6 yielding P9 that is accepted by the classifier as a valid 'l'.

(7) P10 is split into P11 and P12, both of which are accepted as valid characters 'I' and 'i'.

At this stage we have a segmentation that yields four characters 't', 'l', 'I' and 'i'. It is obvious that this segmentation is erroneous as the correct segments should have been 't', 'h' and 'i'. Touching characters that contains three or more characters, broken characters may appear in the sequence of the recognized characters. The error can only resolved using contextual classes and other contextual information. This is discussed in a later sections.

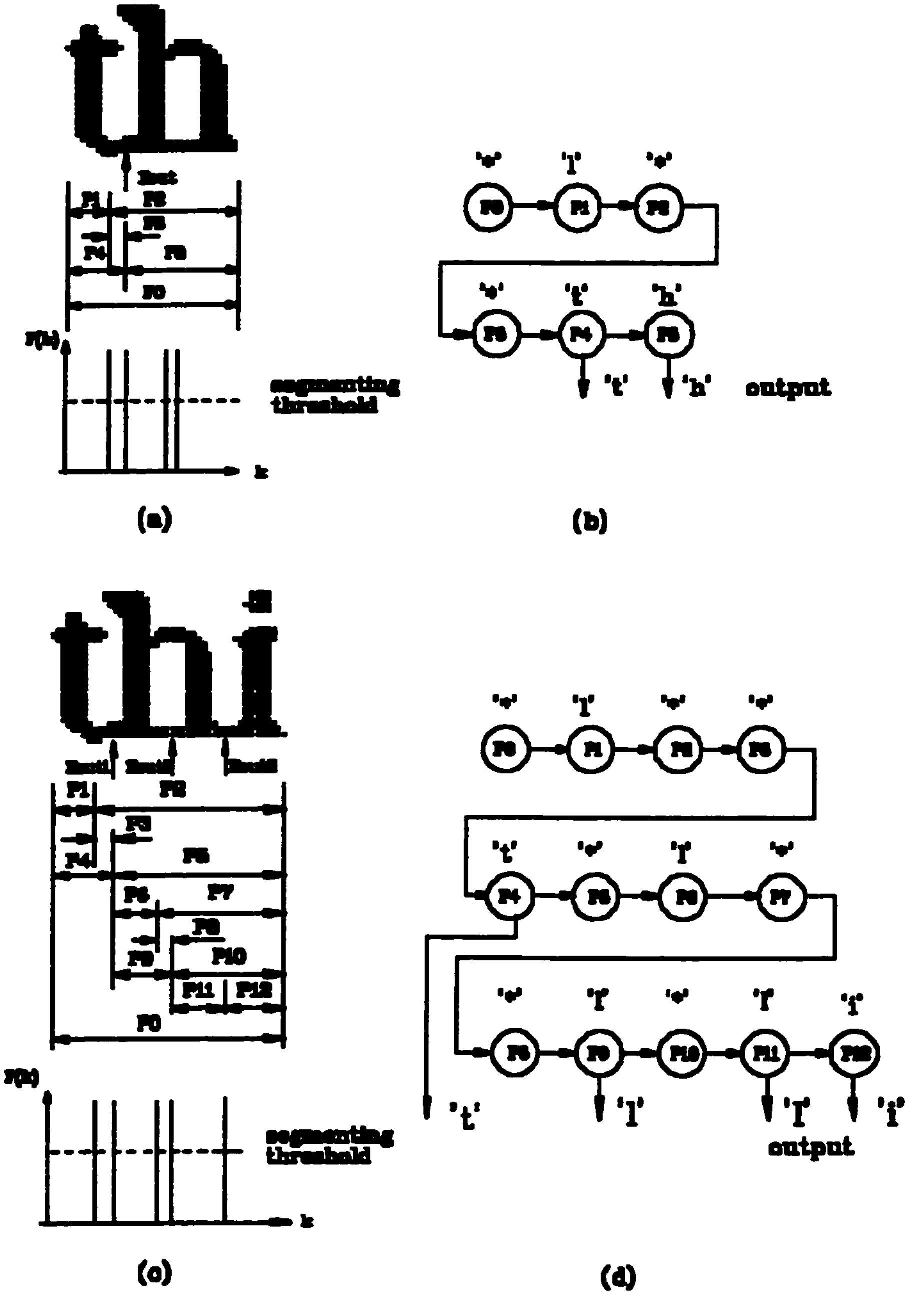


Figure 4.7 (a)(b) The serif touching character 'th' and its time sequence of the segmentation
(c)(d) The serif touching character 'thi' and its time sequence of the segmentation

4.3 Segmentation Touching Characters of Italic Fonts

Two additional steps, *slant detection and slant correction*, are taken prior to the segmentation of italic touching characters. The method described below is simple and requires a small amount of computation. For expression simplicity, the following definitions are given.

Definition:

(1) Let x(r, c) be a given nonzero pixel in the word image. The nonzero pixels, x(r+1, c+1) and x(r+1, c-1), are called the Negative Slant Contribution (NSC) and Positive Slant Contribution (PSC) respectively.

(2) The Relative Slant Ratio (RSR) is defined as:

$$\mathbf{RSR} = (\,\mathbf{PSC} - \ \mathbf{NSC}\,)\,/\,\mathbf{TP} \times 100\% \tag{4.4}$$

where **TP**, **PSC**, and **NSC** are respectively the total numbers of pixels, the total sum of PSCs, and the total sum of NSCs in the word image. Both the NSC and the PSC can be calculated with a 3 X 3 window (Figure. 4.8). The centre '1' pixels in the two masks represent locations of the scanned '1' pixels in the binary word image. The '1' pixels in the bottom-left corner of the PSC mask and the bottom-right corner of the NSC mask denote the reference points. The symbol "*" denotes "don't care". The NSC and PSC in the whole word image are computed by counting the total number of the pixels whose eight-neighbour pixels match the two masks. The **RSR** is then calculated based on the eq. (4.4); it is a relative quantity, therefore, it is obviously size-invariant.

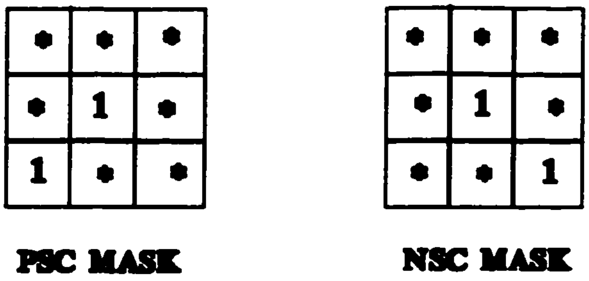


Figure 4.8 The NSC mask and the PSC mask

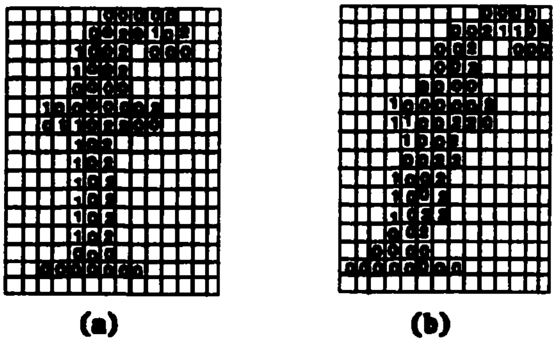


Fig.	No. of NSC '1'	No. of PSC '2'	Total pixels	RSR(%)
(a)	12	13	71	1.4
(b)	9	15	73	8.2

Note: '0's belong to both NSC and PSC

Figure 4.9 An example of computing the NSC, PSC and RSR based on the character 'f' and its italic form

The RSR exhibits consistency in the stroke orientation of the input word image. Figure 4.9 illustrates an example of computing the value of the NSC and the PSC on the single character 'f' and its italic form. In Figure 4.9, the symbols '0', '1' and '2' in both 'f' characters respectively indicate the pixels which do not affect the RSR value, decrease the RSR value, and increase the RSR value. For example, the values of the RSR of the character 'f' and its italic form are approximately 1.4% and 8.1%. Hence, any large positive values of RSR (> 5%) indicates that the character components are positively slanted and should be extracted for slant correction. The slant-corrected word image was obtained by applying the following transformations proposed by Bozinovic and Srihari [89] to all image pixel with coordinates (X, Y) in the original image:

$$X' = X - Y \times \tan(m)\,, \quad Y' = Y \tag{4.5}$$

m is the slant angle with respect to the vertical direction. This angle in italic font families is set to 12° [90]. The slant removed word image is shown in Figure 4.10 (b).

After slant removal, the segmentation discrimination function was calculated based on the new word image; it would have otherwise been difficult to separate the italic touching character '*de*' as shown in Figure 4.10 (a). However, the slant removed word image can sometimes have serious shape distortion due to round off error; therefore, the following transform

$$X_{s\_cut} = X_{v\_cut} - Y \times \tan(m) + \beta \tag{4.6}$$

was applied to the original italicized font image to cut the touching characters at an angle m = 12° (Figure 4.10(c)). β is an adjustable constant, X_{v_cut} is a cutting point in the new image (Fig. 4.10(b)), and X_{s_cut} denotes the slant line in terms of X_{v_cut}, Y, and m.

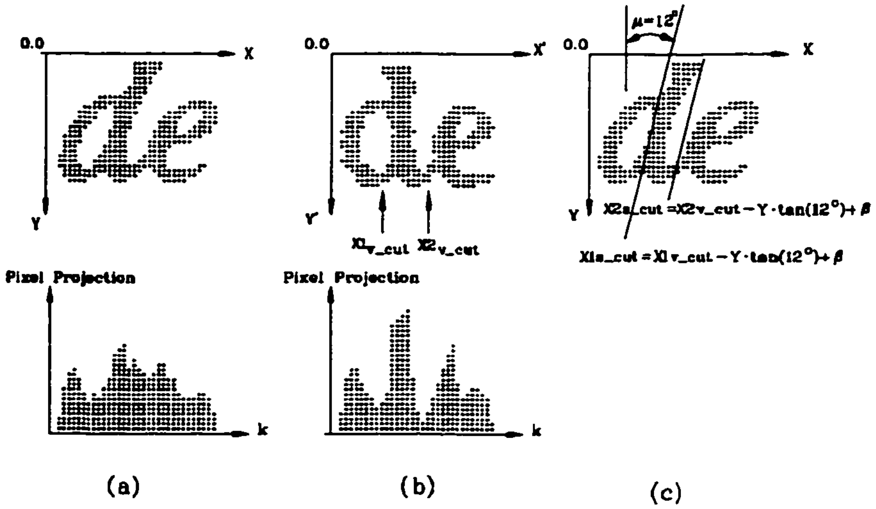
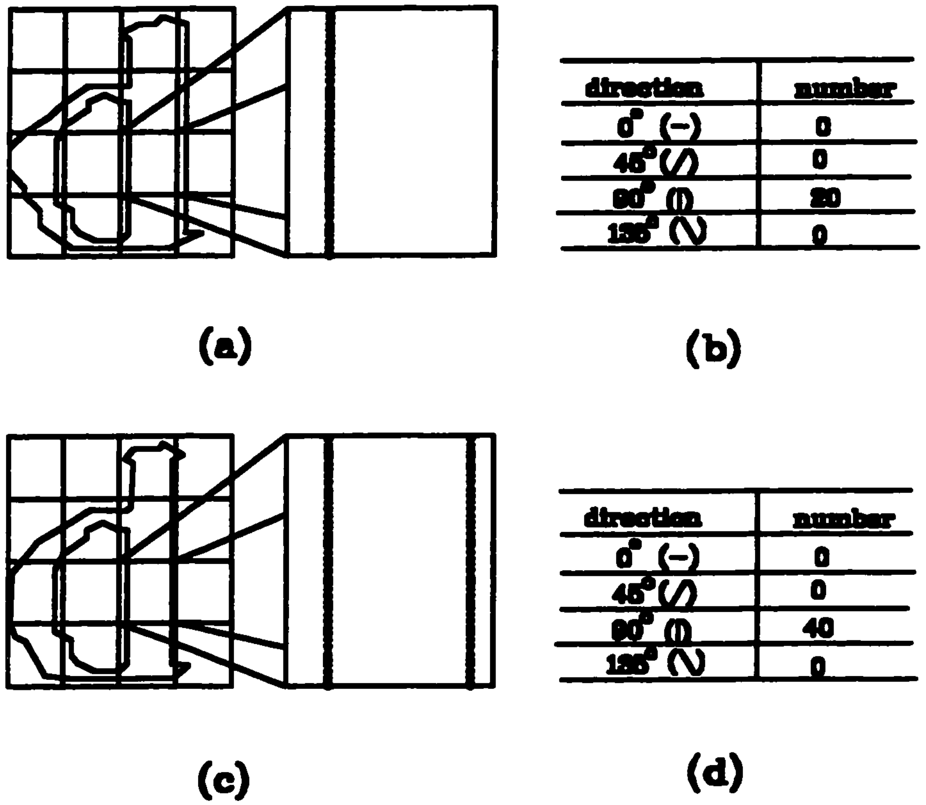


Figure 4.10 (a) The original slant touching character 'de' and its pixel projection
(b) The slant removal 'de' and its pixel projection
(c) The slantly cutting based on the original 'de'

4.4 Feature Extraction and Classification

Even in machine-printed documents, shape discrepancy among characters belonging to the same prototype is sometimes quite large because of the poor quality and low resolution of document images. Particularly, when touching characters are segmented, the noise blobs near the cutting points overlap both sides of the characters, possibly resulting in a large dissimilarity between the input pattern and the corresponding sample class. Image processing techniques such as border tracing or component labelling are able to remove noise blobs that are not touching the character component; however, it is impossible to remove touching noise blobs by simple techniques. We have developed a new feature, the *Overlap-Neighbour-Direction-Feature*, that is largely based on the border chain code histogram proposed by Kimura and Shridhar [53]. They independently calculated a local histogram of the quantized chain codes (0 for '-', 1 for '/', 2 for '\', 4 for '|') in 4x4 rectangular zones (Figure 4.11)(see [53] for detail). The modified feature is then extracted as illustrated in Figure 4.11, where there are six overlap regions across the interface of each sub-rectangle; three vertical strips with width m2, and three horizontal strips with width m1. When a local chain code histogram is calculated, the direction chain codes in the overlap regions contribute their direction values to the histograms of the related local sub-rectangles.



(a)

direction	number
0° (–)	0
45° (/)	0
90° (|)	20
135° (\\)	0

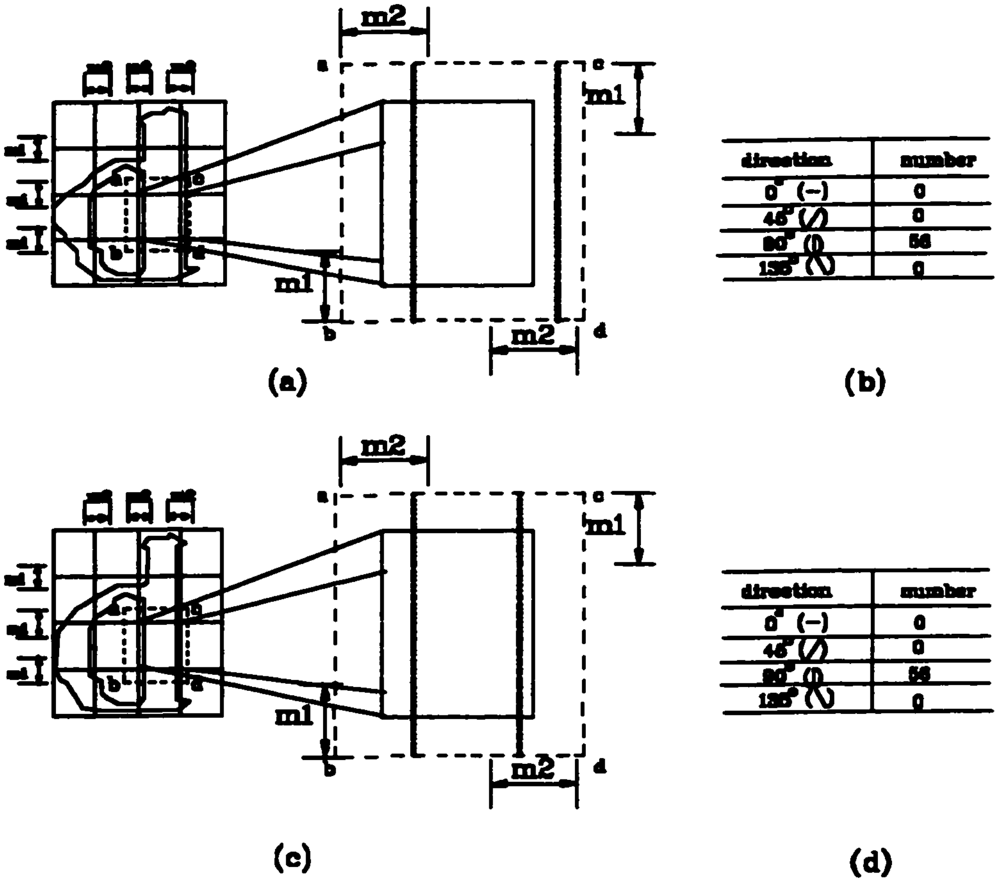
(b)

(c)

direction	number
0° (–)	0
45° (/)	0
90° (|)	40
135° (\\)	0

(d)

Figure 4.11. The direction chain code feature
(a)(c) Normalized character 'd's and their chain codes in a sub-rectangle
(b)(d) Direction chain codes in a sub-rectangle



direction	number
0° (—)	0
45° (/)	0
90° (|)	56
135° (\\)	0

direction	number
0° (—)	0
45° (/)	0
90° (|)	56
135° (\\)	0

Figure 4.12. The Overlap Neighbour Direction feature
(a)(c) Normalized character 'd's and their chain codes in an
extensive sub-rectangle
(b)(d) The direction chain codes in an extensive sub-rectangle

Kimura's feature extraction method created a large distance between the two 'd's shown in Fig. 4.11(a) and 4.11(b) because the rightmost vertical stroke of two 'd's occupy different sub-rectangles. However, the Euclidian distance between the feature vectors of the two input patterns in Fig. 4.12(a) and 4.12(b) decreases with the modified features by properly choosing two parameters, m1 and m2. Our experiments indicate that with the *Overlap-Neighbour-Direction-Feature*, the Euclidian distance between the input pattern and the stored prototype of the same class decreases by approximately 23% on the average when m1 = 10 and m2 = 8 in the 64W X 80H pixel unit rectangular frame.

The minimum-distance classifier [87][88] is an effective technique for classification problems in which the pattern classes exhibit a reasonably limited degree of variability. For a specific and clear machine printed text, the pattern of each class tends to cluster tightly about a typical or representative pattern for that class. Under these conditions, a minimum-distance classifier can be a very effective approach to the classification problems.

Consider M pattern classes and assume that they are represented by the mean feature vectors of the training classes, i.e., m_1, m_2, ..., m_m. The Euclidian distance between an input feature vector X and the ith class is given by

$$D_i = \| X - \mu_I \| = (X - \mu_I)^T (X - \mu_I) \qquad (3.7)$$

In these experiments, X is a 64-dimensional feature vector, i.e. $X = (x_1, x_2, ..., x_{64})^T$, where the superscript T denotes the transpose. Each component x_i of X, for i=1, 2, ... 64, is composed of a local histogram of overlap-neighbour direction chain codes of a character shape boundary based on 4 X 4 subsquares of a normalized binary pattern. A minimum-distance classifier based on a feature extraction approach computes the distance from a input pattern to the mean vector of each class and assigns the pattern to the category of the closest mean, i.e., X is classified to the class i if $D_i < D_j$, for all $j \neq i$.

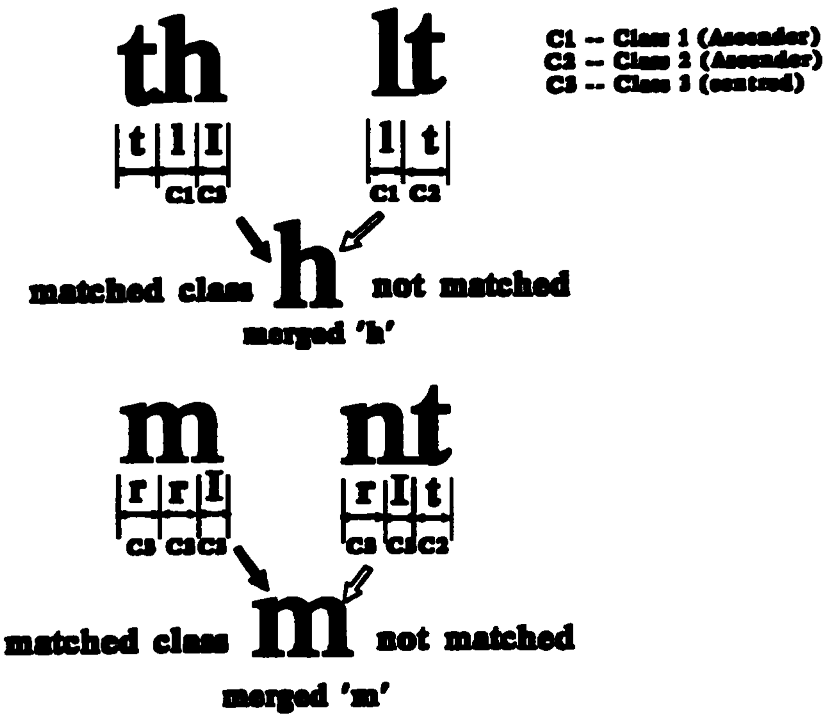


Figure 4.13 Merging broken characters with the knowledge of the character contextual classes

Figure 4.14 Baseline detection in a segmented text line

4.5 Word Analysis Based on Contextual Knowledge

4.5.1 Character Contextual Classes

Using contextual information to analyze a recognized word can improve the performance of the text recognition system. Characters occurring in a text line are generally related to one of the classes based on their location with respect to the baselines [91].

Table 4.1 Character Contextual Classes

class	description	example
1	Ascender	All Capital letters, numerals, and b, d, f, h, k, l
2	Ascender	i, t (not for all fonts)
3	Centred	a, c, e, m, n, o, r, s, u, v, w, x, z
4	Descender	g, p, q, y
5	Full	j

In Table 4.1, the classification of character contextual classes is summarized for the fonts used in our application. To improve the flexibility and applicability of our document recognition system, we have made use of size and font information to achieve an accurate contextual class description of the character components in a particular text line. The 't' and 'i' are aggregated into a specific contextual class, because they have a smaller number of pixels above the upper baseline than the other Ascender character's in a specific font. This eliminates the ambiguities that easily occur between ('i' 'I'), ('i' 'l'), ('t' 'f'), etc. Character contextual classes define character sets as statistically disjoint from the results of classification. The incorrect segmentation achieved earlier for the connected component 'thi' can be resolved by using contextual classes. As shown in Figure 4.13, the combination 'th' which can often be split into three valid characters 't', 'l' and 'I' can be corrected by recognizing that 'l' and 'I' do not belong to the same contextual class and merging the two segments 'l' and 'I' into one segment that is accepted by the classifier as a valid 'h'. Thus using character contextual classes, the component 'thi' is correctly split into 't' and 'h' and 'i'.

To determine the contextual features of character components, the baselines of words are required. The most straightforward method to determine these baselines is to find the character locations with respect to the maximal value of the first order difference of the pixel projection of the text line image in the horizontal direction [89]. Unfortunately, naturally skewed text lines always appear in practical document images, resulting in incorrect character contextual class assignments. Projection of the segmented text lines can be used to relieve this problem (see Figure 4.14).

We took advantage of a two-phase detection of baselines. The first phase was based on the baselines extracted from isolated words; failure of this phase prompted the activation of the second phase which extracted the baselines from segmented text lines. These baselines were used to assign a contextual class to an input character component. This two-phase strategy makes the process more accurate and effective.

4.5.2 Merging Broken Characters

The dynamic recursive segmentation algorithm described in section 4.2 may decompose one character into two or three valid but incorrect character components. Here are some examples:

An 'm' is regarded as 'r' and 'n' or 'r','r' and 'I';

An 'n' is replaced by 'r' and 'I';

A 'B' is segmented into 'I' and '3';

An 'h' becomes 'l' and 'r';

A 'U' is separated into 'I' and 'J'; etc.

Therefore, a merge procedure based on the similarity calculation is performed to rejoin the broken characters in the following steps:

(1) If the characters such as 'l', 'I', 'J', 'r', 'n', 't', etc. appear adjacently in one word, they are considered to be candidate broken character components.

(2) An adjustable window with a maximum size of three character units is moved from left to right across the input word to check if two or three pieces of the character components mentioned in step (1) appear adjacently and simultaneously in the window. If any such character components are detected, a classifier with a specific rejection threshold, is applied to the image array corresponding to those detected character components. Any rejection indicates that the input array contains two or more distinct characters; otherwise, the composition of the character components probably is one character.

(3) The merge procedure is not applied if the character components to be merged are not in correct character contextual classes. For instance, the character 'm' is only composed of Class 3 (Centred) broken components. The character 'h' may contain a Class 1 (Ascender) broken component in the left side and a Class 3 (Centred) broken component in the right side. The touching character 'It', even if accepted by the classifier, can be prevented from being merged to 'h', because the touching character 'It' is composed of one Class 1 (Ascender) component in the left side and one Class 2 (Ascender) component in the right side, violating the merge rule for the character 'h'. (Figure 4.13).

4.5.3 Character Verification

If the output of the classifier do not match to the character contextual classes to which the characters belong, the characters may be corrected according to their contextual classes. For example, both the lower and upper cases of the characters c, o, p, s, u, v, w, x, y, and z can be distinguished by applying contextual class information. Some characters, such as 'i' and 'j' or 't' and 'f', etc. may be misclassified after normalization because of their similarity in shape. The character verification procedure is applied to eliminate these ambiguities. If the classifier outputs for a given pattern does not match the contextual classes to which the pattern should belong, the second character candidate accepted by the classifier with matched contextual class will be considered as a possible substitution. For example, if the input component is recognized as 'i' and its contextual class is 5 (i.e., the number of the pixels above the upper baseline and below the lower baseline is greater than a specific threshold), then the 'i' will be replaced by an accepted candidate 'j' which is a class 5 character (Table 4.1). Since the character verification process is not absolutely free of error due to baseline deviation, the final replacement is made only after spell checking.

4.6 Spelling Errors and Spelling Correction

The dictionary look-up method is the most reliable way to ascertain the character level context [63][92]. The main problem with dictionary look-up methods is the large size dictionary (at least 50,000 words) and consequent costs in memory size and searching time required to handle a realistic vocabulary [63]. Because speed is essential, the spelling correction program should be simple and efficient. Fortunately, the UNIX operating system provides a powerful spell command and a "system" or "popen" function in UNIX C that allows one to execute standard UNIX commands in C programs that makes "on line" spelling correction possible. We assumed that the dictionary in the UNIX system included most English words. To avoid making the correction procedure too complicated, we also assumed that spelling errors in a word came from one of three types of errors (substitution, inserted, and merged character errors), the total number of substitution and insert errors in one word did not exceed one, and the number of merged character errors did not exceed two. Generally, words with more than two different types of error can not be corrected without human intervention. The procedure consists of the two following steps:

(1) Spelling Errors

The words produced by the recognition system after the character verification phase are spell checked by the UNIX spell program. If the given word passes spelling checker test, it is assumed to be a correct word; otherwise, the error correction procedure is employed until a correct word is obtained. If the correction procedure fails to correct the error in the word, the system chooses the first candidate word as the final output. After merging broken characters and verifying characters with contextual class information, we found very few words to be rejected by the spell checker in our case. Only those words absent from the UNIX dictionary were referred to the error correction procedure.

(2) Spelling Correction

Correction of Substitution Character Errors

All characters in a word have corresponding sequential candidates. The number of character candidates corresponding to one character component is equal to the number of classes which have accepted Euclidian distances to the input pattern. These candidates are sorted in the order of their Euclidian distance values. The character candidate with the smallest Euclidian distance to the corresponding input pattern is placed at the top of the rating table. The algorithm outputs the first candidate word accepted by the spell checker.

Correction for Merged Character Errors

Touching characters which have a similar shape to some isolated characters are frequently recognized as specific characters. These touching characters are even confusing to human beings if they do not appear in a word level. For example, touching 'r' and 'n' are regarded as 'm' and touching 'n' and 'n' can be interpreted as 'n' and 'n' or 'r' and 'm'. It is impossible to eliminate these ambiguities without spelling tools. The correction method proposed here is based on the knowledge of those touching characters. If a word was again rejected by the spell checker after substitution error correction, we assumed that there was probably one or two merged character errors in the word. By using possible character compositions to substitute these merged characters, a series of word candidates were obtained. The first candidate to pass the spell checker became the output word.

Correction of Inserted Character Errors

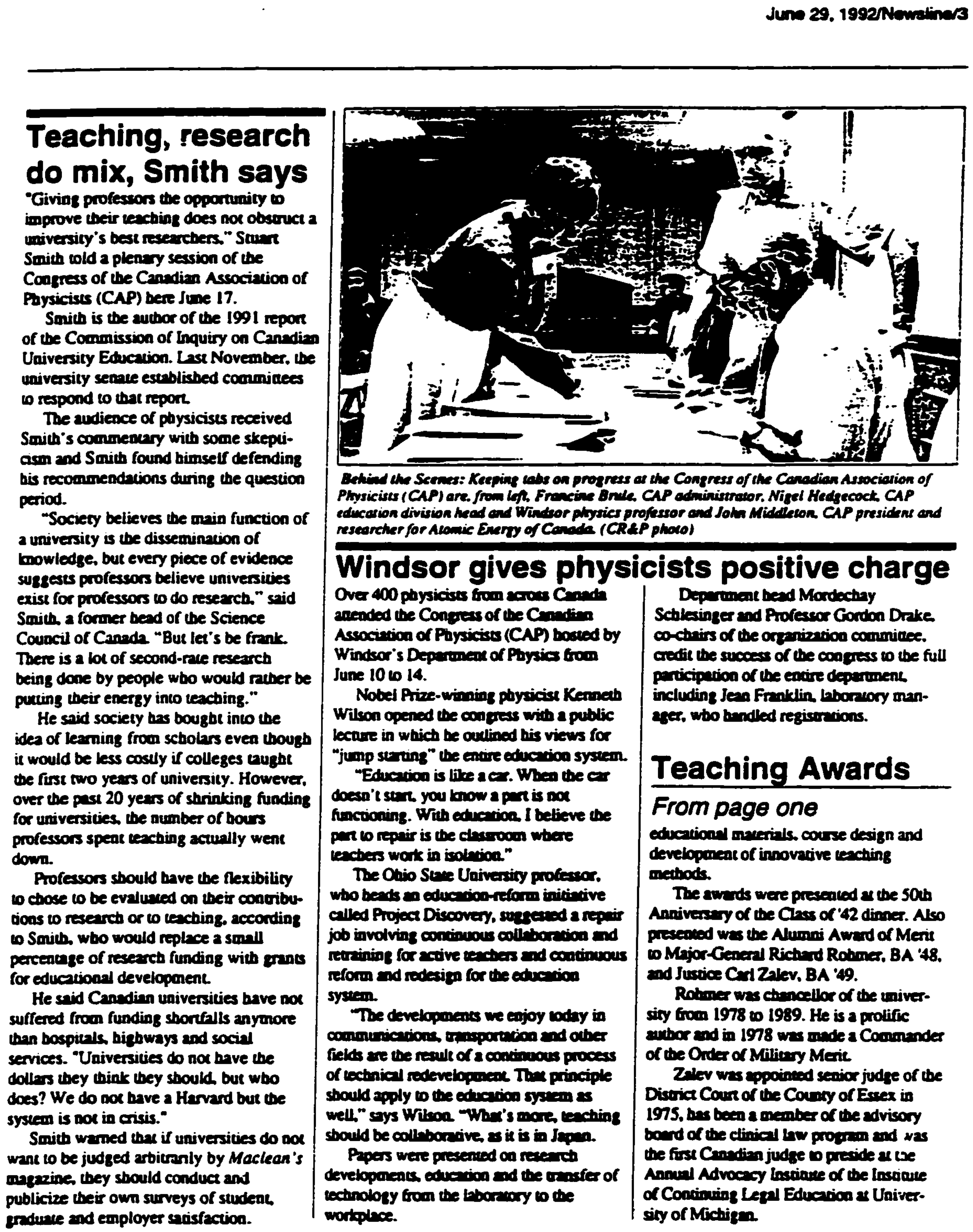
It is very difficult to delete insert errors without some contextual information. Insert errors are always caused by noise components which cannot be eliminated by a noise smoothing filter. Obviously, arbitrarily deleting characters in word strings is impractical. Generally, inserted character errors occur in situations where the character contextual classes of those inserted components do not match the recognition result of these components. For instance, the pixel numbers of inserted characters are always below the specific threshold for a given character height. Hence, using context information is an important clue to delete the insert characters with a least amount of risk.

4.7 Experimental Results

The documents used in our experiment were based on 12 copies of 'NEWS LINE', a *U. of Windsor publication* which contained with 40% touching characters. A typical document image is shown in Fig. 15. The document images were scanned with a 300 dpi scanner connected to an *IBM 386*. The algorithms were written in C, and executed on a **SUN SPARC 10** work station. We collected and constructed the character sample classes in one font individually drawn from the practical document images. The number of samples in each class ranges from 6 samples per class to 100 samples per class, depending on the characters available in the training documents. The clustering technique "K-Means" [88] was used to specify and partition the given data sample sets. After clustering our raw sample data, each sample class set was subdivided into at least one subclass based on a selected threshold. Text, pictures, and graphic blocks were identified, and non-text parts were removed with the block segmentation and text discrimination approaches proposed by Wong, Casey, and Wahl [34]. It is conceivable that both segmentation and recognition may be considerably improved by using contextual information. In our case, layout contextual knowledge dealing with baseline information of text lines and the location of the character components with respect to their neighbours was used to assign a specific class to all the character components in the text lines. We successfully merged the broken character components by using the character contextual classes.

The approaches proposed for segmenting touching characters utilize multiple techniques. To make our interpretation more clear, we use Figs. 4.16 - 4.19 to illustrate the application of various steps of each individual technique. For the purpose of demonstration, only a part of the results of the original document image Figure 4.15 are illustrated. Figure 4.16

shows the results of the dynamic recursive segmentation algorithm based on Figure 4.15. The curly braces were used to indicate the errors made by segmentation and recognition algorithms. Figure 4.17 shows the results after word contextual analysis including merging broken characters. Figure 4.18 shows the results after character verification. Figure 4.19 shows the results after spelling correction.



June 29, 1992/Newsline/3

Teaching, research do mix, Smith says

"Giving professors the opportunity to improve their teaching does not obstruct a university's best researchers." Stuart Smith told a plenary session of the Congress of the Canadian Association of Physicists (CAP) here June 17.

Smith is the author of the 1991 report of the Commission of Inquiry on Canadian University Education. Last November, the university senate established committees to respond to that report.

The audience of physicists received Smith's commentary with some skepticism and Smith found himself defending his recommendations during the question period.

"Society believes the main function of a university is the dissemination of knowledge, but every piece of evidence suggests professors believe universities exist for professors to do research," said Smith, a former head of the Science Council of Canada. "But let's be frank. There is a lot of second-rate research being done by people who would rather be putting their energy into teaching."

He said society has bought into the idea of learning from scholars even though it would be less costly if colleges taught the first two years of university. However, over the past 20 years of shrinking funding for universities, the number of hours professors spent teaching actually went down.

Professors should have the flexibility to chose to be evaluated on their contributions to research or to teaching, according to Smith, who would replace a small percentage of research funding with grants for educational development.

He said Canadian universities have not suffered from funding shortfalls anymore than hospitals, highways and social services. "Universities do not have the dollars they think they should, but who does? We do not have a Harvard but the system is not in crisis."

Smith warned that if universities do not want to be judged arbitrarily by *Maclean's* magazine, they should conduct and publicize their own surveys of student, graduate and employer satisfaction.

***Behind the Scenes:** Keeping tabs on progress at the Congress of the Canadian Association of Physicists (CAP) are, from left, Francine Brule, CAP administrator, Nigel Hedgecock, CAP education division head and Windsor physics professor and John Middleton, CAP president and researcher for Atomic Energy of Canada. (CR&P photo)*

Windsor gives physicists positive charge

Over 400 physicists from across Canada attended the Congress of the Canadian Association of Physicists (CAP) hosted by Windsor's Department of Physics from June 10 to 14.

Nobel Prize-winning physicist Kenneth Wilson opened the congress with a public lecture in which he outlined his views for "jump starting" the entire education system.

"Education is like a car. When the car doesn't start, you know a part is not functioning. With education, I believe the part to repair is the classroom where teachers work in isolation."

The Ohio State University professor, who heads an education-reform initiative called Project Discovery, suggested a repair job involving continuous collaboration and retraining for active teachers and continuous reform and redesign for the education system.

"The developments we enjoy today in communications, transportation and other fields are the result of a continuous process of technical redevelopment. That principle should apply to the education system as well," says Wilson. "What's more, teaching should be collaborative, as it is in Japan.

Papers were presented on research developments, education and the transfer of technology from the laboratory to the workplace.

Department head Mordechay Schlesinger and Professor Gordon Drake, co-chairs of the organization committee, credit the success of the congress to the full participation of the entire department, including Jean Franklin, laboratory manager, who handled registrations.

Teaching Awards

From page one

educational materials, course design and development of innovative teaching methods.

The awards were presented at the 50th Anniversary of the Class of '42 dinner. Also presented was the Alumni Award of Merit to Major-General Richard Rohmer, BA '48, and Justice Carl Zalev, BA '49.

Rohmer was chancellor of the university from 1978 to 1989. He is a prolific author and in 1978 was made a Commander of the Order of Military Merit.

Zalev was appointed senior judge of the District Court of the County of Essex in 1975, has been a member of the advisory board of the clinical law program and was the first Canadian judge to preside at the Annual Advocacy Institute of the Institute of Continuing Legal Education at University of Michigan.

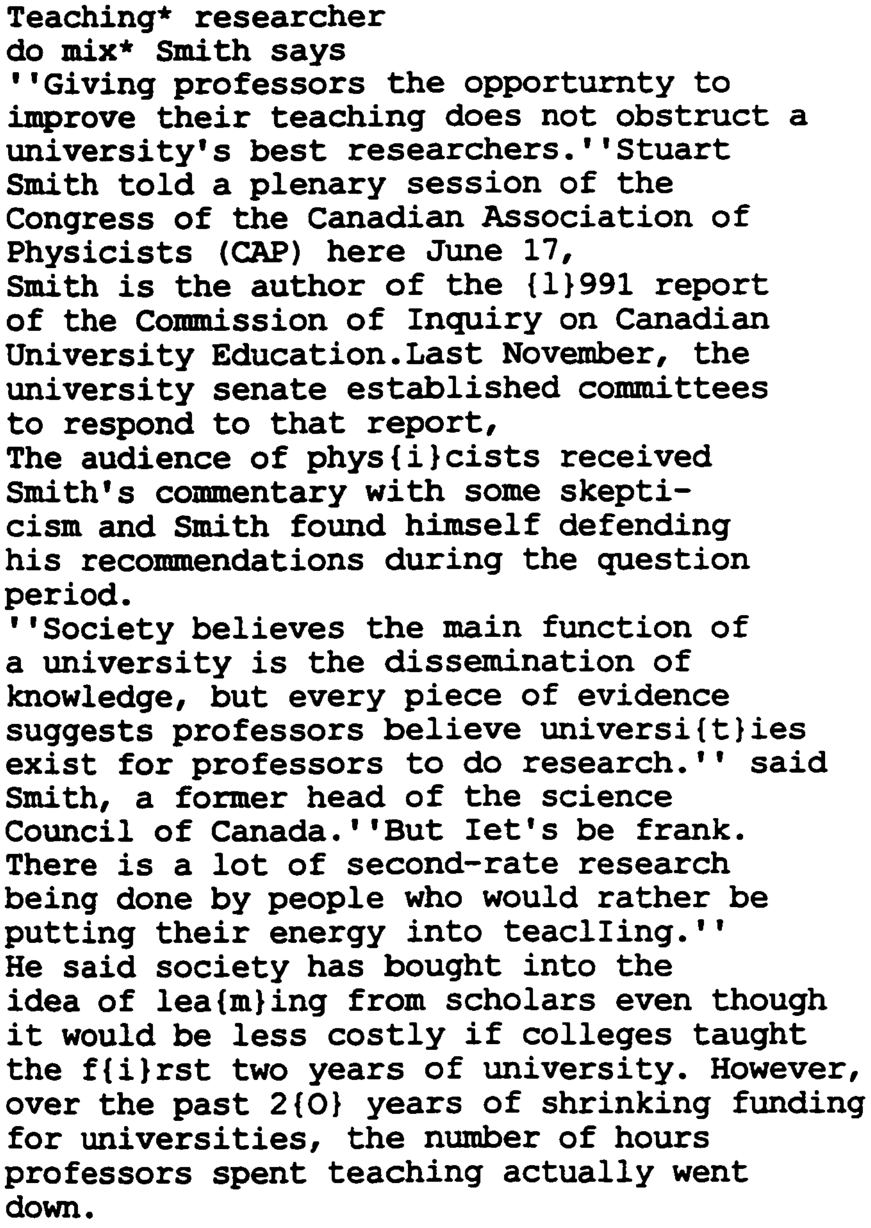
Figure 4.15 A sample document

```
Teaching* researcher
do mix* Smith says
''Giving professors the opportur{Ii}ty to
improve t{lI}eir teachi{rI}g does not obstruct a
u{rl}iversity's best researchers.''{s}tuart
S{rrl}ith told a plenary session of the
{c}o{rI}gress of the Canadian Association of
P{lI}ysicists ({c}AP) here June 17,
S{rrI}ith is the author of the {l}991 report
of the Co{rrI}{rrI}ission of Inquiry on Canadia{rI}
University Education.Last November, the
{II}{rI}iversity se{rl}ate established co{rrI}{rrI}ittees
to respond to that report,
The audience of phys{l}cists received
S{rn}ith's co{rrI}{rrI}entary with some skepti-
cism and {s}{rrl}ith found himself defending
his reco{rrI}{rrl}endations d{ll}ring the question
period.
''Society believes the {rrI}ain function of
a u{rI}iversity is the disse{rrl}i{rI}ation of
knowledge, but every piece of evidence
suggests professors believe u{rl}iversi{l}ies
exist for professors to do research.'' said
S{rrl}ith, a former head of the science
Council of Ca{rI}ada.''{13}ut Iet's be frank.
There is a lot of second-rate research
bei{rI}g done by people who would rather be
putting their energy into teaclIing.''
He said society has bought i{rl}to the
idea of lea{m}ing from scholars even though
it would be less costly if colleges taught
t{lI}e f{l}rst two years of university. However,
over the past 2{O} years of s{lI}ri{rl}king funding
for u{rI}iversities, the n{Il}{rn}ber of ho{lI}rs
professors spent teachi{rI}g actually went
down.
```

Figure 4.16. Results after applying the dynamic recursive segmentation algorithm

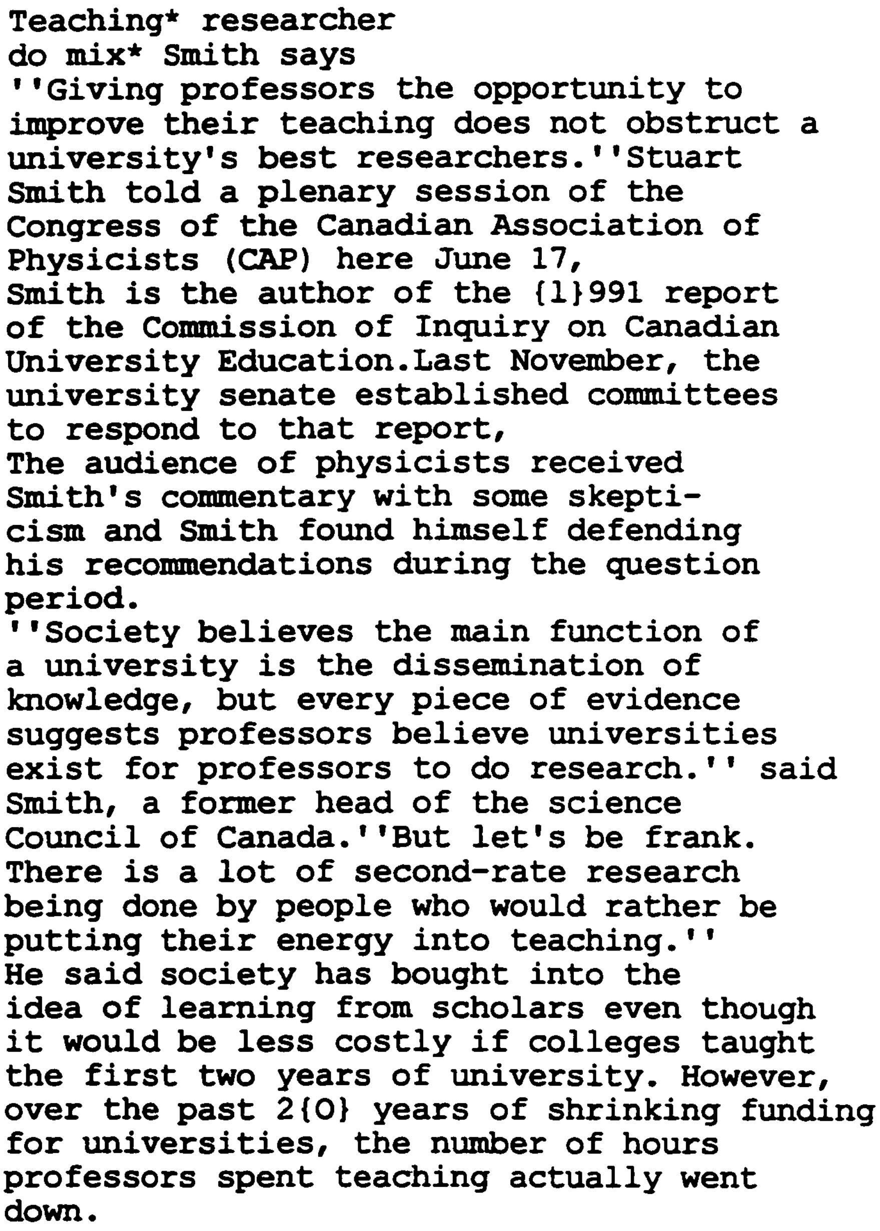
```
Teaching* researcher
do mix* Smith says
''Giving professors the opporturnty to
improve their teaching does not obstruct a
university's best researchers.''{s}tuart
Smith told a plenary session of the
{c}ongress of the Canadian Association of
Physicists ({c}AP) here June 17,
Smith is the author of the {l}991 report
of the Commission of Inquiry on Canadian
University Education.Last November, the
university senate established committees
to respond to that report,
The audience of phys{l}cists received
Smith's commentary with some skepti-
cism and {s}mith found himself defending
his recommendations during the question
period.
''Society believes the main function of
a university is the dissemination of
knowledge, but every piece of evidence
suggests professors believe universi{l}ies
exist for professors to do research.'' said
Smith, a former head of the science
Council of Canada.''But Iet's be frank.
There is a lot of second-rate research
being done by people who would rather be
putting their energy into teaclIing.''
He said society has bought into the
idea of lea{m}ing from scholars even though
it would be less costly if colleges taught
the f{l}rst two years of university. However,
over the past 2{O} years of shrinking funding
for universities, the number of hours
professors spent teaching actually went
down.
```

Figure 4.17. Results after merging broken characters



```
Teaching* researcher
do mix* Smith says
''Giving professors the opporturnty to
improve their teaching does not obstruct a
university's best researchers.''Stuart
Smith told a plenary session of the
Congress of the Canadian Association of
Physicists (CAP) here June 17,
Smith is the author of the {1}991 report
of the Commission of Inquiry on Canadian
University Education.Last November, the
university senate established committees
to respond to that report,
The audience of phys{i}cists received
Smith's commentary with some skepti-
cism and Smith found himself defending
his recommendations during the question
period.
''Society believes the main function of
a university is the dissemination of
knowledge, but every piece of evidence
suggests professors believe universi{t}ies
exist for professors to do research.'' said
Smith, a former head of the science
Council of Canada.''But Iet's be frank.
There is a lot of second-rate research
being done by people who would rather be
putting their energy into teaclIing.''
He said society has bought into the
idea of lea{m}ing from scholars even though
it would be less costly if colleges taught
the f{i}rst two years of university. However,
over the past 2{O} years of shrinking funding
for universities, the number of hours
professors spent teaching actually went
down.
```

Figure 4.18. Result after character verification



```
Teaching* researcher
do mix* Smith says
''Giving professors the opportunity to
improve their teaching does not obstruct a
university's best researchers.''Stuart
Smith told a plenary session of the
Congress of the Canadian Association of
Physicists (CAP) here June 17,
Smith is the author of the {1}991 report
of the Commission of Inquiry on Canadian
University Education.Last November, the
university senate established committees
to respond to that report,
The audience of physicists received
Smith's commentary with some skepti-
cism and Smith found himself defending
his recommendations during the question
period.
''Society believes the main function of
a university is the dissemination of
knowledge, but every piece of evidence
suggests professors believe universities
exist for professors to do research.'' said
Smith, a former head of the science
Council of Canada.''But let's be frank.
There is a lot of second-rate research
being done by people who would rather be
putting their energy into teaching.''
He said society has bought into the
idea of learning from scholars even though
it would be less costly if colleges taught
the first two years of university. However,
over the past 2{0} years of shrinking funding
for universities, the number of hours
professors spent teaching actually went
down.
```

Figure 4.19. Result after spelling correction

Most touching characters were successfully separated. The recognition accuracy was 99.6% for the 12 copies of the documents with 40% touching characters. The recognition errors did not take into account small punctuations such as ',' and '.', and those symbols that were not included in our sample sets.

Recognition errors in our system stemmed mainly from the following sources:

(1) Some substitution errors, merged character errors were not corrected by spelling correction because the correct words were absent from the system dictionary, e. g. names of people or cities.

(2) Some incorrect words were accepted by the spell checker before or while the correction procedure was applied.

(3) Incorrect words with more than one substitution or two merged character errors were not corrected.

Table 4.2 Summary of segmentation and recognition results

total number of characters	total number of errors	substitution errors (40%)	broken errors (50%)	merged errors (10%)	inserted errors(0%)	recognition rate
54000	185	74	92	19	0	99.6%

In Table 4.2, we list the number of errors and the error type distribution occurred in our recognition system. About 30% of the errors were broken errors. They were caused by deviation of the baseline detection and the reject threshold set for merging the broken characters. About 60% of the errors were substitution errors between 'l' and '1', '0' and 'O', and etc. About 10% of the errors came from merged errors that were not corrected by spelling correction because more than one type of error occurred in a word. By applying the commercial OCR (HP DeskScan II WordScan 3.0) to the same document images, a 99.1% recognition rate is obtained, which is lower than 99.6% recognition rate of the proposed algorithm. This demonstrates that our proposed algorithm is effective for segmenting and recognizing text images with many touching characters.

4.8 Conclusions

We have developed an effective segmentation strategy for segmenting a word into its character components. A new discrimination function is presented in this thesis for segmenting touching characters based on pixel and profile projections. A dynamic recursive segmentation algorithm is developed for effectively segmenting touching characters. Contextual information and spell checking are used to correct errors caused by incorrect recognition and segmentation. With multiple segmentation techniques, we have provided robustness to the proposed algorithms, and we feel that these techniques can be introduced into an overall document recognition system.

Chapter V

Summary and Conclusions

5.1 Recapitulation

The principle objective of the research work described in this thesis was to develop algorithms for restoration and segmentation of machine printed documents.

To achieve this goal, we first developed a morphological approach to text string extraction from regular periodic overlapping text/background images. Mathematical morphology, because of its ability to grasp the geometry and structure of images, is adopted to realize this new scheme. The underlying strategy of the algorithm is to maximize background component removal while minimizing the shape distortion of text characters by using appropriate morphological operations. The performance of the new scheme was tested on six artificial images and one real image, each with a different style of periodically distributed background symbols. The experimental results indicate that the algorithm is effective and reliable. This new method is appropriate for implementation in a document analysis system.

Algorithms were also developed to remove handwritten interference strokes which do not possess the property of periodicities. To extract interference strokes from a binary image, we have presented a new methodology that effectively segments not only straight lines but also smooth arbitrarily oriented curves. This segmentation technique combines two processes of thinning and detection of directional strokes. These two processes operate complementarily and make the algorithms preserve both connectivity and directional characteristics. Morphological operations based on orientation maps and skeleton images are developed to successfully prevent the "flooding water" effect of conventional morphological operations. The proposed algorithm successfully removed the interference strokes based on 20 test images. We have compared the recognition results with and without the interference removal to evaluate the performance of our new algorithm. The experimental test showed that the commercial OCR completely failed to recognize the text strings that were cut across by interference strokes. The experimental tests indicated a significant improvement in recognition rate after removal of interference strokes using the proposed algorithm.

Finally, we proposed a new algorithm for segmenting touching characters. Segmenting a word into its character components is one of the most critical steps in the document recognition process. In this thesis, a new segmentation discrimination function for segmenting touching characters based on the pixel projection and profile projections is presented. With the locations of cutting points detected by the discrimination function, a dynamic recursive segmentation algorithm executes a forward segmentation or a backward merging process dynamically based on the recognition result of the current input array and its neighbouring touching components. With contextual information and a spell checker, errors caused by incorrect recognition and segmentation (broken characters, substitution characters, merged characters, and inserted characters) are corrected in multiple phases. Based on 12 real documents, a 99.6% recognition accuracy has been achieved.

5.2 Future Research

In this final section, the author would like to reveal a number of problems worthy of future investigation. In this thesis, we developed new morphological operations to implement the removal of the interference background symbols including periodic overlapping background symbols and handwritten marks. As observed in our experiments, the morphological operations which are quite useful for image processing and analysis require a great deal of computation to implement. Because of the simplicity of morphological operations, VLSI technology for implementing these operations on large images in real time becomes possible and worthy of future investigation. Meanwhile, the algorithms for segmenting text with interference stroke background have indicated significant improvement in character recognition. The technique is applicable to a class of interference strokes characterized by a relatively large perimeter and small curvatures. Under these conditions, some threshold parameters used in our proposed algorithm can be determined from knowledge of character dimension in the document and digitized resolution of the scanned document images. It is very important to conduct more research on the automatic detection of those parameters in order to make the algorithms more practical.

Bibliography

1. S. Mori, C, Y. Suen, and K. Yamamoto, Historical Review of OCR Research and Development, Proceddings of the IEEE, Vol. 80, No. 7, July 1992, pp 1029-1058.

2. M. Bokser, Omnidocument Technologies, Proceedings of the IEEE, Vol. 80, No. 7, July, 1992, pp 1066-1078.

3. H. S. Baird, Anatomy of a Versatile Page Reader, Proceddings of the IEEE, Vol. 80, No. 7, July 1992, pp 1059-1065.

4. G. Nagy, Teaching a Computer to Reading, in 11th IAPR International Conference on Pattern Recognition, Hague, Netherlands, Aug. 30 - Sept. 3, 1992, pp 225-229.

5. Y. Y. Tang, C. Y. Suen, C. D. Yan, and M. Cheriet, Document Analysis and Understanding: A Brief Survey, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo,France, 1991, pp 17-31.

6. S. N. Shrihari, High-Performance Reading Machines, Proceedings of the IEEE, Vol. 80, No. 7, July, 1992, pp 1120-1132.

7. H. Fujisawa, Y. Nakano, and K. Kurino, Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis, Proceedings of the IEEE, Vol. 80, No. 7, 1992, pp 1079-1092.

8. R.G. Casey and C. R. Jih, A Processor-Based OCR System, IBM J. Res. Develop, Vol. 27, No. 4, July 1983, pp378-399.

9. S. N. Srihari and G. W. Zack, Document Image Analysis, 8th ICPR International Conference on Pattern Recognition, Paris, France, Oct. 27-31, 1986, pp 434-436.

10. S. Tsujimoto and H. Asada, Majar Components of the a Complete Text Reading System, Proceedings of the IEEE, Vol. 80, No. 7, July, 1992, pp 1133-1149.

11. G. Nagy, J. Kanai, and M. Krishnamoorthy, Two Complementary Techniques for digitized Document Analysis, Proc. ACM Conference on Document Processing System, 1988, pp 149- 159.

12. T. Bayer, J. Franke, U. Kressel, E. Mandler, M. Oberlander, and J. Schurmann, Towards the Understanding of Printed Documents, Structure Document Image Analysis, H.S. Baird, H. Bunke, and K. Yamamoto Eds, Springer-Verlag, 1992, pp 3-35.

13. J. Schurmann, N. Bartneck, T. Bayer, J. Franke, E. Mandler, and M. Oberlander, Document Analysis — From Pixels to Contents, Proceedings of the IEEE, Vol. 80, No. 7, July, 1992, pp 1101-1119.

14. R.C. Gonzales and R.E. Woods, Digital Image Processing, Addision-Wesley, 1992.

15. J.M. White and G.D. Rohrer, Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction, IBM J. Res. Develop, Vol. 27, No. 4, Nov. 1983, pp 400-410.

16. T. Taxt, P. J. Flynn, and K. Jain, Segmentation of Document Images, IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No. 12, Spet. 1989, pp 1322-1329.

17. H. Yamada, K. Yamamoto, T. Saito, K. Hosokawa, and H. Yanagisawa, Laser-Marked Alphanumeric Character Recogniton by Multi-Angled Parallel Matching Method, 11th IAPR International Conference on Pattern Recognition, Hague, Netherlands, Aug. 30 - Sept. 3, 1992, pp 326-349.

18. J. Kittler and J. Illingworth, Minimum Error Thresholding, Pattern Recognition, 29(1), 1986, pp 41-47.

19. R. M. Haralick, S. R. Sternberg, and X. Zhung, Image Analysis Using Mathematical Morphology, IEEE tans. on Pattern Analysis and Machine Intelligence, Vol. 9, No. 4, July, 1987, pp 532-550.

20. J. Serra, Image Analysis and Mathematical Morphology, Acdemic Press, New York, 1982.

21. T.Kanungo, R. Haralick, and I. Phillips, Global and Local Document Degradation Models, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 730-734.

22. Y.Y. Tang and C.Y. Suen, Nonlinear Shape Restoration by Transformation Models, 10th International Conference on Pattern Recognition, Vol 2, Altantic City, USA, 16-21 June, 1990, pp 14-19.

23. T. Akiyama and N. Hagita, Automated Entry System for Printed Documents, Pattern Recognition, Vol 23. No. 11, 1990, pp 1141-1154.

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, 10th International Conference on Pattern Recognition, Vol 2, Altantic City, USA, 16-21 June, 1990, pp 464-468.

25. A. Hashizume, P.S. Yeh, and A. Rosenfeld, A Method of Detecting the Orientation of aligned components, Pattern Recognition Letter, Vol. 4, pp 125-132, 1986.

26. H. Ozawa and T. Nakagawa, A Character Image Enhancement Method from Characters with Various Background Images, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 58-61.

27. D.S. Doermann and A. Rosenfeld, The Processing of Form Documents, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 497-501.

28. J.M. Gloger, Use of the Hough Transform to Separate Merged Text/Graphics in Forms, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 268-271.

29. S.H. Joseph, On the Extraction of Text Connected to Linework in Document Images, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo,France, 1991, pp 993-999.

30. J. P. Bixler, Tracking Text in Mixed-Mode Documents, ACM Conference on Document Processing Systems, Santa Fe, New Mexico, Dec. 5-9, 1988, pp 177-185.

31. D. Wang and S. N. Srihari, Analysis of Form Images, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo,France, 1991, pp 181-191.

32. L. Boatto, V. Consorti, M.D. Buono, V. Eramo, and A. Esposito, Detection and Separation of Symbols Connected to Graphics in Line Drawings, 11th IAPR International Conference on Pattern Recognition, Hague, Netherlands, Aug. 30 - Sept. 3, 1992, pp 545-548.

33. D. Guillevic and C. Y. Suen, Cursive Script Recognition: A fast Reader Scheme, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 311-314.

34. K.Y. Wong, R.G. Casey, and F.M. Wahl, Document Analysis System, IBM J. RES. DEVELOP, Vol. 26, No. 6, Nov. 1982, pp 647-656.

35. T. Pavlidis and J. Zhou, page Segmentation by White Streams, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo,France, 1991, pp 945-953.

36. T. Saitoh, M. Tachikawa, and T. Yamaai, Document Image Segmentation and text Area Ordering, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 323-329.

37. F. Hones, J. Lichter, Text string Extraction within Mixed-Mode Documents, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 655-659.

38. J. Wieser and A. Pinz, Layout and Analysis: Finding Text, Titles, and Photos in Digital Images of Newspaper Pages, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp774-777.

39. T. Saitoh and T. Pavlidis, Page Segmentation without Rectangle Assumption, 11th IAPR International Conference on Pattern Recognition, Hague, Netherlands, Aug. 30 - Sept. 3, 1992, pp 277-280.

40. H. Fujisawa and Y. Nakano, A Top-Down Approach to the Analysis of Document Images, Structure Document Image Analysis, H.S. Baird, H. Bunke, and K. Yamamoto Eds, Springer-Verlag, 1992, pp 99-114.

41. Image Analysis Application, edited by R. Kasturi and M. M. Trivedi, Marcel Dekker, Inc. 1990, pp 1-37.

42. R. Kasturi, Rajesh, C. Chennubhotla, and L. OGorman, An Overview of Techniques for Graphics Recognition, Structure Document Image Analysis, H.S. Baird, H. Bunke, and K. Yamamoto Eds, Springer-Verlag, 1992; pp 285-324.

43. V. Govindaraju and S. N. Srihari, Separating Handwritten Text from Interfering Strokes, From Pixels to Features III: Frontiers in Handwriting Recognition, S. Impedovo and J.C. Simon (eds), Elsevier Science Publisher B.V., 1992, pp 17-28.

44. J. Illingworth and J. Kittler, A Survey of the Hough Transform, Computer Vision, Graphics, and Image Proceedings 44, 1988, pp 87-116.

45. L. A. Fletcher and R. Rasturi, A Robust Algorithm for Text String Segmentation From Mixed Text/Graphics Images, IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, Nov. 1988, pp 910-1988.

46. C. Crowner and J. J. Hull, A Hierarchical Pattern Matching Parser and its Application to Word Shape Recognition, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo, France, 1991, pp 323-331.

47. T. K. Ho, J. J. Hull, and S. N. Srihari, Word Recognition with Multi-Level Contextual Knowledge, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo,France, 1991, pp 905-915.

48. C. H. Chen and J. L. DeCurtins, Word Recognition in a Segmentation-free Approach to OCR, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 573-576.

49. J.C. Simon and O. Baret, Cursive Words Recognition, From Pixels to Features III: Frontiers in Handwriting Recognition, S. Impedovo and J.C. Simon (eds), Elsevier Science Publisher B.V., 1992, pp 241-259.

50. H. S. Baird and R. Fossey, A 100-Font Classifier, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo, France, 1991, pp 333-340.

51. F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition, IEEE trans. on Pattern Analysis and Machine Intelligence, 9, 1987, pp 149-153.

52. M. Shridhar and A. Badreldin, A High-Accuracy Syntactic Recognition Algorithm for Handwritten Numerals, IEEE trans. on Syst. Man, and Cybern. 15, 1985, pp 152-158.

53. F. Kimura and M. Shridhar, Handwritten Numerical Recognition Based on Multiple Algorithm, Pattern Recognition 24, 1991, pp 969-983.

54. S. Kahan, T. Pavlidis, and H. S. Baird, On the Recognition of Printed Characters of Any Fonts and Size, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 2, Mar. 1987. pp 274-287.

55. H. S. Baird, Document Image Defect Models and Their Use, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 62-67.

56. S. Khoubyari and J. J. Hull, Keyword Location in Noise Document Images, Second Annual Symposium on Document Analysis and Information Retrieval, pp 217-231.

57. A. Kryzak, W. Dai, and C.Y. Suen, Unconstrained Handwritten Classification Using Modified Backpropagation Model, in Proc. Frontiers in Handwritten Recognition, 1990, pp 155-164.

58. K. Fukushima and N. Wake, Handwritten Alphanumeric Character Recognition by Neocognitron, IEEE trans. Neural Networks, Vol. 2, 1991, pp 355-365.

59. Y. Lu, On the Segmentation of Touching Characters, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993. pp 565-568.

60. R.G. Casey and G. Nagy, Recursive Segmentation and Classification of Composite Character Patterns, Proc. 6th Inter. Conf. Pattern Recognition, Munich, Germany, 1982, pp. 1023-1026.

61. R. M. K. Sinha, B. Prasada, G. F. Houle, and M. Sabourin, Hybrid Contextual Text Recognition with String Matching, IEEE trans. on Pattern Analysis and Machine Intelligence, Vol., 15, No. 9, Sept. 1993, pp 915-155.

62. R.M.K. Sinha and B. Prasada, Visual Text Recognition through Contextual Processing, Pattern Recognition, Vol. 21. No 5, 1988, pp 463-479.

63. C. J. Wells, L.J. Evett, P.E. White, and R.J. Whitrow, Fast Dictionary Look-up for Contextual Word Recognition, Pattern Recognition, Vol. 23, No. 5, 1990, pp 501-508.

64. R. M. K. Sinha, On Partitioning a Dictionary for Visual Text Recognition, Pattern Recognition, Vol. 23, No.5, 1990, pp 497-500.

65. H. Takahashi, N. Itoh, T. Amano, and A. Yamashita, A Spelling Correction Method and its Application to an OCR System, Pattern Recognition, Vol. 23, No. 3/4, 1990, pp 363-377.

66. N.Billawala, P.E.Hart, and M. Peairs, Image Continuation, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 53-57.

67. R.M. Lougheed, An Overview of Greyscale Morphological Filters, 23th Asilomar Conferece on Signals, Systems, and Computers, Vol.1, Oct. 30 - Nov. 1, Pacific Grove, California, 1989, pp 152-156.

68. Y. Liu, R. Fenrich, and S.N. Srihari, An Object Attribute Thresholding Algorithm for Document Image Binarization, Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp 278-281.

69. P. Maragos, A Representation Theory for Morphological Image and Signal Processing, IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No. 6, June, 1986, pp 586-599.

70. J. Serra, Introduction to Mathematical Morphology, Computer Vision, Graphics, and Image Processing 35, 1986, pp 283-305.

71. R. C. Vogt, Automatic Generation of Morphological Set Recognition Algorithms, Springer-Verlag, New York, 1989.

72. I. Pitas and A. N. Venetsnopoulos, Morphological Shape Representation, Pattern Recognition, Vol. 25, No. 6, 1992, pp 555-565.

73. C.R. Giardina and E.R. Dougherty, Morphological Methods in Methods in Image and Signal Processing, Prentice-Hall, Englewood Cliffs, N.J., 1988.

74. M. J. B. Duff, D. M. Watson, T. M. Fountain, and G.K. Shaw, A Cellular Logic Array for Image Processing, Pattern Recognition. 5, 1973, pp 229-247.

75. F.A. Gerristsen and L.G. Ardema, Design of Use of DIP-1: A Fast Flexibleand Dynamically Microprogrammable Image Processor, Pattern Recognition. 27, 1984, pp 115-123.

76. F. A. Gerritsen and P.W. Verbeek, Implementation of Cellular Logic Operation Using 3 X 3 Convution and Table Lookup Hardware, Computer Vision, Graphics, and Image Processing, 27. 1984, pp 319-330.

77. H. Yamada, K. Yamamoto, T. SAito, and S. Matsui, MAP: Multi-Angled Parallelism for Feature Extraction from Topographical Maps, Pattern Recognition, Vol. 24, No. 6, 1991, pp 479-488.

78. H. Yamada, K. Yamamoto, and K. Hosokawa, Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographics Maps, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 15. No. 4, April, 1993, pp 380-387.

79. Y.S. Chen and W.H. Hsu, An Interpretive Model of Line Continuation in Human Visual Perception, Pattern Recognition, Vol. 22. No. 5, 1989, pp 619-639.

80. Y.K. Chu and C. Y. Suen, An Algorithm Smoothing and Stripping Algorithm for Thinning Digital Binary Patterns, Signal Processing 11, 1986, pp 207-222.

81. T. Pavlidis, A Vectorizer and Feature Extractor for Document Recognition, Computer Vision, Graphics, and Image Processing 35, 1986, pp 111-127.

82. S. S. Wilson, Theory of Matrix Morphology, IEEE tans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 6, June, 1992, pp 636-652.

83. S. Tsujimoto and H. Asada, Resolving Ambiguity in Segmenting Touching Characters, 1st Int. Conf. on Document Analysis and Recognition, Saint-Malo,France, 1991, pp 701-709.

84. H. S. Baird, Calibration of Document Image Defect Models, Second Annual Symposium on Document Analysis and Information Retrieval, 1992, pp 1-16.

85. C. B. Bose and S. Kuo, Connected and Degraded Text Recognition Using Hidden Markov Model, Hague, Netherlands, Aug. 30 - Sept. 3, 1992, pp 116-119.

86. A. Rosenfeld and A.C. Kak, Digital Image Processing, 2nd Ed, Addision-Wesley, London, 1982, pp 347-349.

87. R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

88. J.T. Tou and R.C. Gonzales, Pattern Recognition Principles, Addison-Wesley, London, 1974.

89. R. M. Bozinovic, and S. N. Srihari, Off-Line Cursive Script Word Recognition, IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No. 1, Jan., 1989. pp 68-83.

90. D.E. Knuth, Computer Modern Typefaces, Addision-Wesley, 1986.

91. P.G. De Luca and A. Gisotti, Printed Character Preclassification Based on Word Structure, Pattern Recognition, Vol. 24, No. 7, 1991, pp 609-615.

92. H. Takahashi, N. Itoh, T. Amano, A. Yamashita, A Spelling Correction Method and its Application to An OCR System, Pattern Recognition, Vol. 23, No. 3/4, 1990, pp 363-377.

Appendix

Appendix 1:

The procedure for calculating PDH and PDV is described as follows in C-like format:

```
 Procedure for left edges extraction:
       for (r from 1 to Image_Row ; c from 1 to Image_Col)
             {
              image_left_edge[r][c] = image_text_background[r][c] $\Theta$ T1;
             }
 Procedure for top edges extraction:
      for( r from 1 to Image_Row ; c from 1 to Image_Col)
             {
             image_left_edge[r][c] = image_text_background[r][c]  $\Theta$ T2;
              }

 Procedure for calculating PDH:
      for( k from 1 to M){
             for( r from 1 to Image_Row ; c from 1 to Image_Col)
                    {
                    $imagebuff_k$ [r][c] = image_left_edge[r][c] $\Theta$ $B_i$ ;
                    $N_{k-1}$ = CL($imagebuff_{k-1}$ );
                    $N_k$ = CL($imagebuff_k$ );
                    $N_{k+1}$ = CL($imagebuff_{k+1}$ );
                    Iff( $|N_k - N_{k-1}|$ > threshold && $|N_k - N_{k+1}|$ > threshold)
                           PDH = k ;
                    }
                      }

 Procedure for calculating PDV:
      for( k from 1 to M){
             for( c from 1 to Image_Col; r from 1 to Image_Row ;)
                    {
                    $imagebuff_k$ [r][c] = image_left_edge[r][c] $\Theta$ $B_j$ ;
                    $N_{k-1}$ = CL($imagebuff_{k-1}$ );
                    $N_k$ = CL($imagebuff_k$ );
                    $N_{k+1}$ = CL($imagebuff_{k+1}$ );
                    iff( $|N_k - N_{k-1}|$ > threshold && $|N_k - N_{k+1}|$ > threshold)
                           PDH = k ;
                    }
                      }
```

Appendix 2:

The procedure for extracting background pixels is described as follows in C-like format:

```
 image_background$_1$[r][c] = original_image[r][c];
 for( i from 1 to Number_iteration)
        {
        for( r from 1 to Image_Row ; c from 1 to Image_Col)
                {
                image_background$_i$ [r][c] = image_background$_{i-1}$ [r][c] Θ S1;
                }
       if(image_background$_{i-1}$ [r][c]-image_background$_i$[r][c] < threshold)
procedure_end;

for( r from 1 to Image_Row ; c from Image_Col to 1)
        {
        image_background$_i$ [r][c] = image_background$_{i-1}$ [r][c] Θ S2;
        }
    if(image_background$_{i-1}$[r][c]-image_background$_i$[r][c] < threshold)
procedure_end;

for( r from 1 to Image_Row; c from 1 to Image_Col)
        {
        image_background$_i$ [r][c] = image_background$_{i-1}$ [r][c] Θ S3;
        }
    if(image_background$_{i-1}$ [r][c]-image_background$_i$ [r][c] < threshold)
procedure_end;

for( r from Image_Row to 1 ; c from 1 to Image_Col )
        {
        image_background$_i$ [r][c] = image_background$_{i-1}$ [r][c] Θ S4;
        }
    if(image_background$_{i-1}$ [r][c]-image_background$_i$ [r][c] < threshold)
procedure_end;
```

Appendix 3:

The procedure for text restoration is described as follows in C-like format:

```
for( r from 1 to Image_Row ; c from 1 to Image_Col)
    {
    if(image_text_background[r][c] == 1)
                image_background_removal[r][c] =
                        image_text_background[r][c] XOR image_background[r][c];
    }

for( r from 1 to Image_Row; c from 1 to Image_Col)
  do  {
        image_background_removal[r][c]
                = removal_isolate_pixels(image_background_removal);
      }
 do  {
        image_text1[r][c] = closing1(image_background_removal);
      }
 do  {
        image_intersect[r][c] = intersect(image_text1,image_background);
      }
 do  {
        image_text2[r][c]=
                union(image_backgd_removal,image_background_removal);
      }
 do  {
        image_text_final[r][c] = closing2(image_text2);
      }
```

Appendix 4:

The procedure for modification of headline images is described as follows in C-like format:

```
for( r from 1 to Image_Row ; c from 1 to Image_Col)
        {
         image_headline_modified[r][c] = image_headline[r][c];
        }

for( r from 1 to PDV; c from 1 to Image_Col)
        {
         k = PDV;
        while( k < Image_Row && k >= PDV)
                {
                if( image_headline[r][c]==1 && image_headline[k][c]==0)
                        image_headline_modified[k][c] = 1;
                k = k + PDV;
                 }
        }
```

VITAE AUCTORIS

SU LIANG

1962 Born in April, SuZhow, China

1983 Received B. A. Sc. degree in Electrical Engineering from Dalian Marine College, China.

1988 Received M. A. Sc. degree in Electrical Engineering from University of Science and Technology of China, China.

1996 Candidate for Ph. D. degree in Electrical Engineering at University of Windsor, Windsor, Ontario, Canada.

PUBLICATIONS

Journals

[1] Su Liang, M. Shridhar, and M. Ahmadi, "Segmentation of Touching Characters in Printed Document Recognition", **Pattern Recognition**, Vol. 27, No 6, pp 825-840, 1994.

[2] Su Liang, M. Ahmadi, and M. Shridhar, "Morphological Approach for Text String Extraction from Regular Overlapping Text/Background Images", **CVGIP: Graphical Models and Image Processing**, Vol. 56, No. 3, pp 402-413, Sept. 1994.

[3] Su Liang, M. Ahmadi, and M. Shridhar, "Segmentation of Handwritten Interference Marks Using Multiple Directional Stroke Planes and Reformalized Morphological Approach", To appear in IEEE Transaction on Image Processing. (Date of acceptance June 1996)

Conferences

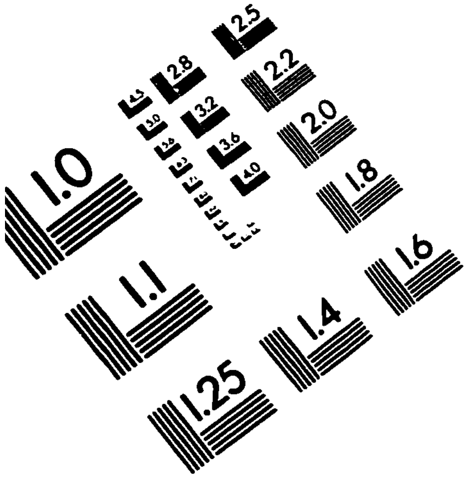
[1] Su Liang, M. Ahmadi, and M. Shridhar, "Segmentation of Handwritten Interference Marks Using Multiple Directional Stroke Planes and Reformalized Morphological Approach", Proceedings of International Conference on Document Analysis and Recognition (ICDAR 95), pp. 1161-1164, Montreal, Aug., 1995.

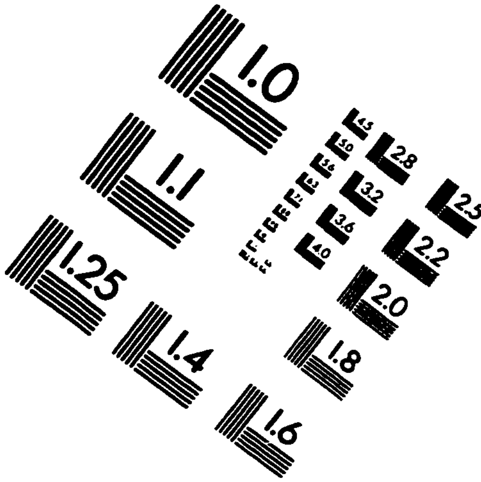
[2] Su Liang, M. Ahmadi, and M. Shridhar, "Morphological Approach for Text String Extraction from Regular Overlapping Text/Background Images", First IEEE Inter. Conf. on Image Processing, Austin, Texas, Nov. 13-16, 1994.

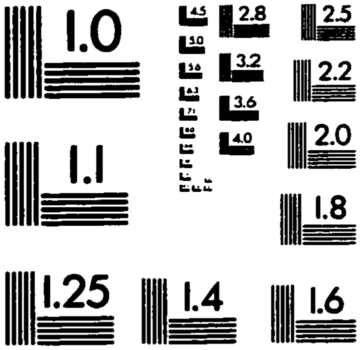
[3] Su Liang, M. Shridhar, and M. Ahmadi, "Efficient Algorithms for Segmentation and Recognition of Printed Characters in Document Processing", in Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, May 19-21, vol. 1, pp. 240-243, 1993.

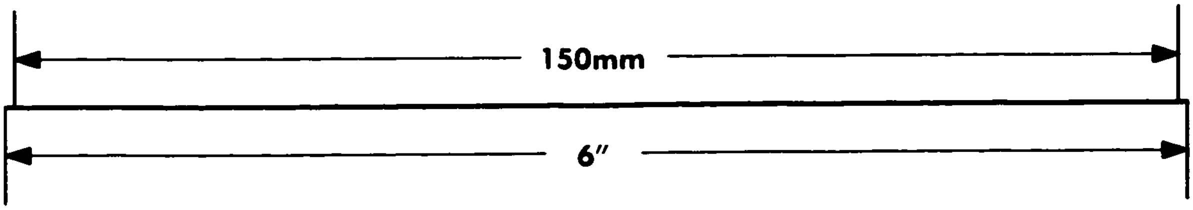
[4] Su Liang, M. Shridhar, and M. Ahmadi, "Segmentation of Touching Characters in Printed Document Recognition," in Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 569-572, Oct. 20-22, 1993.

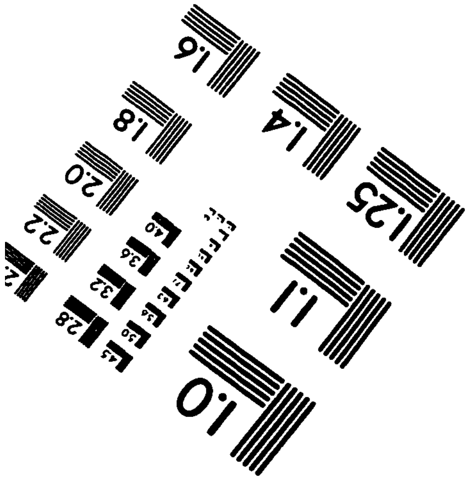
IMAGE EVALUATION
TEST TARGET (QA-3)

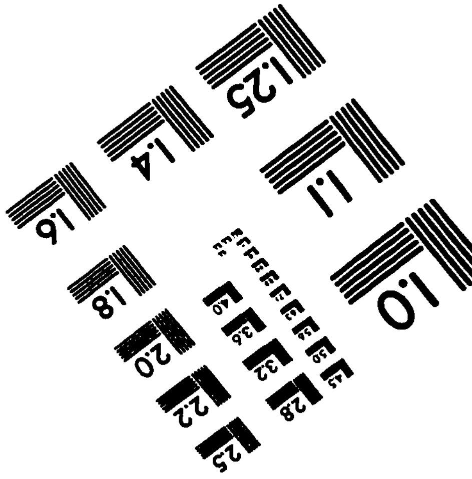












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989