

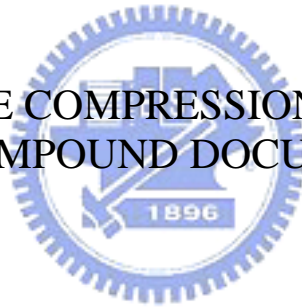
國立交通大學

電機與控制工程學系

博士論文

複雜型複合式文件影像壓縮方法之研究

THE STUDY OF THE COMPRESSION ALGORITHMS FOR
COMPLEX COMPOUND DOCUMENT IMAGES



研究生：瞿忠正

指導教授：吳炳飛 教授

中華民國九十三年十二月

複雜型複合式文件影像壓縮方法之研究
THE STUDY OF THE COMPRESSION ALGORITHMS FOR
COMPLEX COMPOUND DOCUMENT IMAGES

研究生：瞿忠正

Student : Chung-Cheng Chiu

指導教授：吳炳飛

Advisor : Bing-Fei Wu

國立交通大學
電機與控制工程學系
博士論文



Submitted to Department of Electrical and Control Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
In partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Electrical and Control Engineering

December 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年十二月

複雜型複合式文件影像壓縮方法之研究

學生：瞿忠正

指導教授：吳炳飛 教授

國立交通大學 電機與控制工程學系 博士班

摘 要

由於複合式文件影像中包含許多文字資訊，當文件影像以傳統壓縮方法壓縮時，文字資訊會產生大量的失真，文字和一些屬於高頻的資訊會變的模糊。所以，傳統壓縮方法並不適合拿來直接對複合式文件影像作壓縮處理，同時，壓縮後文件影像中的文字，也無法容易的被電腦辨識或被我們閱讀。因為文件中文字資訊的重要性，所以文件影像的文字切割技術已經發展了十多年，但是針對複合式文件影像的研究，仍是一個新鮮的研究課題。目前已有許多學者針對複合式文件影像研究文字切割的方法，但是這些方法依然不能適用於目前報章雜誌上圖文交疊、背景變化多端的複雜型複合式文件影像。像這類複雜型複合式文件影像的文字切割技術，可以說是文件影像處理的一大挑戰。如果可以從不同複雜程度的影像中，將文字切割出來，那就可以適用於所有的文件影像處理。本篇論文研究目標就是發展出一種可以解決複雜型複合式文件影像的文字切割方法，使文件影像壓縮可以達到更高的壓縮倍數與視覺品質。

本篇論文提出三個文字切割的方法，這三個方法所處理的複合式文件影像難度依章節順序增高。本文中提出的切割方法應用於文件影像壓縮，可以明顯的看出壓縮倍

數與視覺品質優於 JPEG 或 DjVu，而且在本文第三個切割方法(MLSM)中，提出新的區域性區塊特徵分離與拼圖式全圖整合的方法，在解決複雜型複合式文件影像的文字切割問題時，即使在同一張完整的文件影像中，包含各種不同程度的複雜狀況，也可以順利的將不同顏色、不同複雜背景與不同交疊程度的文字切割出來，提高各種複雜型複合式文件影像的壓縮品質。



THE STUDY OF THE COMPRESSION ALGORITHMS FOR COMPLEX COMPOUND DOCUMENT IMAGES

Student : Chung-Cheng Chiu

Advisor : Prof. Bing-Fei Wu

Department of Electrical and Control Engineering
National Chiao Tung University

ABSTRACT

Traditional image compression methods are not suitable for compound document images because such images include much text. These image data are high-frequency components, many of which are lost in compression. Text and the high-frequency components thus become blurred. Then, the text cannot be recognized easily by the human eye or a computer. The text contains most information, separating the text from a compound document image is one of the most significant areas of research into document images. Document image segmentation, which separates the text from the monochromatic background, has been studied for over ten years. Segmenting compound document images is still an open research field. Many techniques have been developed to segment document images. However, they are insufficient when the background includes sharply varying contours or overlaps with text. Finding a text segmentation method of complex compound documents remains a great challenge and the research field is still young. This dissertation presents three segmentation algorithms for compressing image documents, with a high compression ratio of both color and monochromatic compound document images. The

proposed algorithms greatly outperform the famous image compression methods, JPEG and DjVu, and enable the effective extraction of the text from a complex background, achieving a high compression ratio for compound document images.



ACKNOWLEDGEMENTS

第一次見到恩師 吳炳飛教授，就深深被教授的獨特氣質與淵博的學養所吸引，當下就決定進到 CSSP 實驗室展開人生一段非常重要的學術研究訓練。這一段學習的過程當然是多采多姿，研究的工作雖然辛苦，但是絕不會累！因為一路上有教授幫忙排除萬難；研究的路途雖然遙遠，但是決不孤獨！因為一路上有許多實驗室的夥伴陪伴。

能順利完成博士學位，需要感謝的人很多，首先要感謝的就是恩師 吳炳飛教授與師母，感謝教授為我開啟一扇智慧之門，讓我未來可以走的更遠更平順；感謝師母的協助，讓教授能有更多的時間來指導我們。感謝強哥與旭哥在我研究的過程一路陪伴，真懷念我們一起在 CSSP 實驗室並肩熬夜打拼的日子；尤其要感謝陳彥霖學弟，他從碩士班開始就跟著我做研究，並直攻到博士班，一路陪伴我完成博士學業；感謝實驗室所有學弟、妹們，與各位合作研究的經驗，是我這一生重要的回憶。

感謝口試委員 宋開泰教授、張志永教授、賈叢林教授、陸儀斌教授與蘇崇彥教授，給予我在研究上許多寶貴的意見。

更要感謝我的父親 瞿梓庭先生與母親 羅金妹女士，從小就給我一個模範的榜樣，感謝父親嚴厲的身教與言教，讓我能掌握人生的方向，謝謝您們！

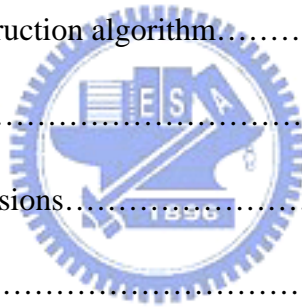
最後要感謝的是我的妻子 李明珍小姐，由於研究工作的繁重，這段時間感謝妳的陪伴與支持，兩個頑皮小子旭民與瑞宏也照顧的很出色，讓我無後顧之憂的將研究工作圓滿完成，謹將此成果與妳分享。

忠正 於國立交通大學電機與控制工程學系 CSSP 實驗室 2004/12/14

CONTENTS

ABSTRACT (Chinese)	ii
ABSTRACT (English)	iv
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
1. INTRODUCTION	1
1.1 Motivation.....	1
1.2 Organization of the dissertation.....	2
2. THE FUZZY-BASED TEXT SEGMENTATION METHOD	7
2.1 Introduction	8
2.2 The characteristics of coefficients in wavelet transform.....	9
2.3 Fuzzy picture-text segmentation algorithm.....	12
2.4 Color document images compression method.....	27
2.5 Experimental results.....	31
2.6 Concluding remarks.....	32
3. THE COMPRESSION ALGORITHMS FOR COMPOUND DOCUMENT IMAGES WITH LARGE TEXT/BACKGROUND OVERLAP	35
3.1 Introduction	36

3.2 Text segmentation algorithm.....	38
3.3 Document image compression algorithm.....	48
3.4 Experimental results.....	53
3.5 Concluding remarks.....	55
4. THE MULTI-LAYER SEGMENTATION METHOD FOR COMPLEX DOCUMENT IMAGES.....	68
4.1 Introduction.....	69
4.2 Multi-layer segmentation method.....	73
4.2.1 Block-based clustering algorithm.....	77
4.2.2 Jigsaw-puzzle layer construction algorithm.....	81
4.3 Text extraction algorithm.....	97
4.4 Experimental results and discussions.....	104
4.5 Concluding remarks.....	108
5. CONCLUSIONS AND PERSPECTIVE.....	116
REFERENCE.....	120
VITA.....	125
PUBLICATION LIST.....	127



LIST OF TABLES

Table 1	The dynamic range of UV-plane coefficient.....	20
Table 2	The fractal dimension influenced by the resolution of images.....	23
Table 3	Comparison the compression ratio and PSNR for the proposed methods to JPEG & DjVu.....	67



LIST OF FIGURES

Fig.1 Image after 2-level discrete wavelet transformation.....	10
Fig.2 The flowchart of Fuzzy picture-text Segmentation algorithm.....	13
Fig.3 The influence of threshold.....	14
Fig.4 Example of CRLA and region growing.....	17
Fig.5 The edge projection of text components in high frequency band.....	18
Fig.6 Two kinds of projection histogram.....	19
Fig.7 Images with text/picture component.....	23
Fig.8 The characteristics of three parameters.....	25
Fig.9 The membership functions of three parameters.....	25
Fig.10 Fuzzy Rule Table.....	26
Fig.11 Possible fuzzy quantization by triangle-sharp fuzzy number.....	27
Fig.12 The flowchart of proposed compression algorithm.....	28
Fig.13(a) The original image(512×1024, scanned by ScanMaker V600 at 200dpi).....	33
Fig.13(b) JPEG image(CR=104.1).....	33
Fig.13(c) Proposed method(CR=112.3).....	33

Fig.14(a) The original image(1024×512, scanned at 200dpi).....	34
Fig.14(b) JPEG image(CR=83.1).....	34
Fig.14(c) Proposed method(CR=84.31).....	34
Fig.15 The original full image (200 dpi).....	39
Fig.16 The planes after clustering algorithm.....	40
Fig.17 An example of text extraction algorithm (size=256×128).....	46
Fig.18 Document image compression format.....	50
Fig.19 Segmentation images of proposed algorithm CSSP-I.....	59
Fig.20 Compared with proposed algorithm CSSP-I & JPEG.....	61
Fig.21 Processed images of the DjVu.....	64
Fig.22 Processed images of the CSSP-II.....	66
Fig.23 An example of the results after the block-based clustering algorithm...	75
Fig.24 Flowchart of the decision procedure to construct or extend an object layer.....	89
Fig.25 An example of the MLSM (image size=1361x1333).....	96
Fig.26 The example of the text extraction algorithm of the Fig.25(d).....	101
Fig.27 Test image 1 (image size=2262x3263).....	110

Fig.28 Test image 2 (image size=1829x2330)..... 111

Fig.29 Test image 3 (image size=2462x3250)..... 112

Fig.30 Test image 4 (image size=2333x3153)..... 113

Fig.31 Test image 5 (image size=2469x3535)..... 114

Fig.32 Test image 6 (image size=2469x3535)..... 115



CHAPTER 1

INTRODUCTION

1.1 Motivation

As the electronic storage, retrieval and transmission of documents become faster and cheaper, documents are becoming increasingly digitized. The image size of a magazine page at 300 dpi is 3300 pixels high and 2500 pixels wide, it occupies about 25 Mbytes of memory in uncompressed form. The volume of data greatly prolongs the transmission time and makes the storage cost high. Therefore, the document images should be compressed before transmission or storage.

A typical digital image encoder initially converts the input image data into coefficients by means of one of the transform procedures, such as DCT and FFT. The obtained coefficients are then encoded using scalar or vector quantization followed by one of entropy-based coders which is a Huffman coder in a majority of cases. The combination of discrete wavelet transform [1],[2] and zerotree coding [3],[4] was proposed to compress a nature image.

However, using those methods on color document images as advertisements existed in our daily life has achieved poor performance on text. Since the

characteristics of text and pictures are different, it is not suitable to compress them by the same method like JPEG [5] or discrete wavelet transform. Digitized images of printed documents typically consist of a mixture of texts, pictures, and graphics elements, which have to be separated for further processing and efficient representation. Because text captures the most information, how to segment the text from printed document images becomes an important step in document analysis.

So far, there are more and more documents printed with gorgeous styles such as various color texts and background objects. They must be segmented from an image to facilitate further processing. Therefore, many researchers have developed valuable segmentation techniques for applications that include document analysis, image segmentation, image compression, and pattern recognition.



1.2 Organization of the dissertation

In this dissertation, three segmentation methods for document images proposed to extract texts from compound document images.

In the Chapter 2, a compression method for color document images based on the wavelet transform and fuzzy picture-text segmentation is presented. This approach addresses a fuzzy picture-text segmentation method, which separates pictures and texts by using wavelet coefficients from color document images. Two components, text strings and pictures, are generated and processed by different compression

algorithms.

The fuzzy picture-text segmentation method separates the text and picture from the monochromatic background. However, the rapid development of multimedia technology has led to increasing numbers of real-life documents, including stylistic text strings with decorated objects and colorful, slowly or highly varying background components. These documents overlap the text strings with the background images. Therefore, the fuzzy picture-text segmentation method cannot effectively segment all important objects. It is insufficient when the background includes sharply varying contours or overlaps with text.

Therefore, Chapter 3 proposes a new segmentation algorithm for separating text from a document image with a complex background. However, the image of text cannot easily be directly separated from the background image because the difference between the gray values is too small. Therefore, two phases are used to accomplish the desired purpose. In the first phase, which involves color transformation and clustering analysis, the monochromatic document image is partitioned into three planes, the dark plane, the medium plane, and the bright plane. The color of the text is almost all the same, so the variance of the text's grayscale is small. Therefore, all the text can be grouped in the same plane. When the text is black, the text and some of the background with a gray value close to that of the text is put in the dark plane. In contrast, the text is put into another plane if it is not black. Thus, the text and some

noise are coarsely separated from the background. In the second phase, an adaptive threshold is determined to refine the text by adaptive binarization and block extraction. Then, two compression algorithms that yield a high compression ratio are also proposed.

The segmentation algorithm focuses on processing the images whose texts are overlap to the complex background. The study is powerful in extracting texts from complex backgrounds. However, we can find many advertisements or magazines whose background images contain many different cases including 1) monochromatic background with/without texts, 2) slowly varying background with/without texts, 3) highly varying background with/without texts and 4) complex varying background with/without different color texts. It is hard to extract the texts when all of the cases spread in a compound document image, especially. Furthermore, the color of texts may be more than three. Therefore, the segmentation algorithm in the Chapter 3 may be insufficient to extract the text from document images in all cases. The text segmentation method of those complex images becomes a great challenge and still a novel research field.

To conquer this challenge, we present a text segmentation algorithm for various document images in Chapter 4. The proposed segmentation algorithm incorporating with a new multi-layer segmentation method (MLSM) can separate the text from various compound document images, regardless of whether the text and background

overlap. This method solves various problems associated with the complexity of background images.

The MLSM provides an effective method to extract objects from different complex images. The complex image includes many different objects such as difference color texts, figures, scenes and complex backgrounds. Those objects could be overlapped or non-overlapped by each others. Because those objects have different features, the image can be partitioned into many object-layers by means of the features of objects embedded in it. Then the block-based clustering algorithm can be performed on those layered image sub-blocks and cluster them to form several object layers. Consequently, different text, non-text objects and background components are segmented into separate object layers. The proposed method can separate or objects from 8-bit grayscale or 24-bit true-color images, no matter the objects overlap a simple, slowly or highly varying background. The block-based clustering algorithm decomposes the sub-block image into different layered sub-block images, *LSBs*, in the order of darkest to lightest corresponding to the original sub-block image. In the jigsaw-puzzle layer construction algorithm, some statistical and spatial features of adjacent *LSBs* are introduced to assemble all *LSBs* of the same text paragraph or object.

The different text and non-text objects and background components are clearly

segmented into several independent object layers for further extraction process. When applied to real-life, complex document images, the proposed method can successfully extract text strings with various colors and illuminations from overlaying non-text objects or complex backgrounds, as determined experimentally. Experimental results obtained using different document images scanned from book covers, advertisements, brochures, and magazines reveal that the proposed algorithm can successfully segment Chinese and English text strings from various backgrounds, regardless of whether the texts are over a simple, slowly varying or rapidly varying background texture.



CHAPTER 2

THE FUZZY-BASED TEXT SEGMENTATION METHOD

This chapter presents a compression method for color document images based on the wavelet transform and fuzzy picture-text segmentation. This approach addresses a fuzzy picture-text segmentation method, which separates pictures and texts by using wavelet coefficients from color document images. The number of colors, the ratio of projection variance, and the fractal dimension are utilized to segment the pictures and texts. By using the fuzzy characteristics of these parameters, a fuzzy rule is proposed to achieve the purpose of picture-text image segmentation. Two components, text strings and pictures, are generated and processed by different compression algorithms. The picture components and the text components are encoded by zerotree wavelet coding and by the modified run-length Huffman coding, respectively. Experimental results have shown that the work has achieved promising performance on high compression ratio for color document images.

2.1 Introduction

Digitized images of printed documents typically consist of a mixture of texts, pictures, and graphics elements, which have to be separated for further processing and efficient representation. Because text captures the most information, how to segment the text from printed document images becomes an important step in document analysis. Accordingly, various techniques have been developed to segment document images. Many approaches devoted to process monochrome document have been proposed in the past years. Wahl *et al.* [6] designed a prototype system for document analysis and a constrained run length algorithm (CRLA) for block segmentation. Nagy *et al.* [7] presented an expert system with two tools: the X-Y tree and formal block-labeling schema, to accomplish document analysis. Fletcher and Kasturi [8] proposed a robust algorithm, which uses the Hough transform to group connected components into local character strings, to separate text from mixed text/graphics document images. Kamel and Zhao [9] presented two new extraction techniques: a logical level technique and a mask-based subtraction technique. Tsai [10] proposed an approach to automatic threshold selection using the moment-preserving principle. Some other systems based on the prior knowledge of some statistical properties of various blocks [11]-[15], or texture analyses [16],[17] have also been successively developed. Those systems all focus on processing monochrome document. In contrast, few approaches have been proposed for dealing with color document. Suen and Wang

[19] presented a text string extraction algorithm, which uses the edge-detection technique and text block identification to extract the text string. Haffner et al. [20] proposed an image compression technique called “DjVu” that is specially geared toward the compression of document image in color.

In this chapter, we present a compression method for color document images by using a new fuzzy picture-text segmentation algorithm. This proposed segmentation algorithm separates the text from color document images in frequency domain by using the coefficients derived from the discrete wavelet transform. The coefficients are used to separate the text/picture components by using the fuzzy classification method. Then, the coefficients of picture components are encoded by the coding method of zerotree, and Modified Run-Length Huffman Code (MRLHC) encodes the coefficients of text components.

2.2 The characteristics of coefficients in wavelet transform

The basic theory of the wavelet transform is to represent any arbitrary function f as a superposition of wavelets. After the first level of two-dimensional discrete wavelet transform, the image was arranged placing the lowest-frequency band in the left upper corner, the highest-frequency band in the right down corner, and the middle-frequency band in the right upper corner and left down corner. The coefficients in lowest-frequency band have further correlation than the ones in other

bands, therefore, making the image decomposed into the second level of wavelet transform, and then seven frequency bands as Fig.1 are obtained. The coefficients of seven bands obtained from the original image by wavelet transform are treated as the textures of different frequency.

LL2	LH2	LH1
HL2	HH2	
HL1		HH1

Fig.1 Image after 2-level discrete wavelet transformation.

In Fig.1, the LL2 band has the coefficients of the lowest-frequency. LHi, HLi and HHi (i=1, 2) bands indicate the edge information in the original images. The LL2 band is very similar to the original image but with only the size of 1/16. The cost of calculation can be reduced by using the LL2 band for picture-text segmentation. For picture components, the signals in LH1, HL1 and HH1 bands are not sensitive to human eyes, and they could be directly discarded. However, for text components, those frequency bands include prominent edge information which should be coded to preserve the text information.

After the wavelet transform, the coefficients extracted from text components

appear more edge information than the coefficients from picture components. That is, the coefficients of text components show higher frequency characteristics than the coefficients of picture components. As such, the characteristics of wavelet transform coefficients are different between text components and picture components. Therefore, the edge feature is a good parameter for segmenting picture-text components.

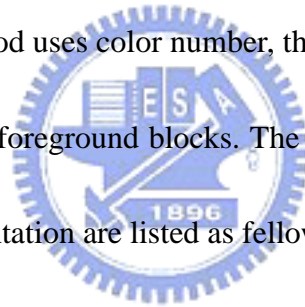
The number of colors, also a useful feature, can be used for color-document image segmentation. Because the coefficients of LL2 band are very similar to the original image after wavelet transform, the color number can be obtained by counting the color number of UV-plane from the coefficients of LL2 band. However, it is difficult to obtain the number of color from the color document images directly. In this chapter, a new algorithm is proposed to extract the number of colors from the coefficients of LL2 band.

The fractal dimension indicates the complexity of images. In addition, it is found that the fractal dimension [22] between text components and picture components is quite different. Because the picture components are more complicated than the text components, the picture components have higher fractal dimension than the text components. The fractal dimension in original image and in low-resolution image is similar. Therefore, the coefficients of LL2 band are applied only to obtain the fractal dimension from text components and picture components. In this way, the processing time to compute fractal dimension can be reduced.

2.3 Fuzzy picture-text segmentation algorithm

As mentioned in previous section, the new document image segmentation utilizes the coefficients from wavelet transform to extract the features of picture components and text components which can be further separated by extracted features using a fuzzy algorithm. We use the technique of spreading and region growing to mark all the foreground blocks including picture-images, text-images and other kinds of images on LL2 band, and these blocks are then segmented to text components or picture components (non-text components).

The segmentation method uses color number, the energy of edge projection, and fractal dimension to segment foreground blocks. The reasons why we use those three parameters to perform segmentation are listed as follows :



(1)Color number: Since picture components are more colorful than text components, color number can distinguish them.

(2)The energy of edge projection: It shows the distribution of edge projection in a block. In general, the variation of edge projection is regular in text components, and irregular in picture components.

(3)Fractal dimension (FD): It shows the complexity of images. In most cases, the pixels in text components distribute more uniformly and the fractal dimension is lower than picture components.

The three parameters, color number, the energy of edge projection and fractal dimension, are used in the same time to reduce misjudgment. For example, if the reliability of the three parameters are $9/10$, $19/20$ and $4/5$, the decision error of picture-text segmentation would diminish to $1/1000$ ($1/10 \times 1/20 \times 1/5$) when we consider the three parameters in the same time appropriately. The Fuzzy Rule calculation [23] is very suitable to analyze this kind of variables. Therefore, we propose a fuzzy picture-text segmentation algorithm to separate text components and picture components from color document images.

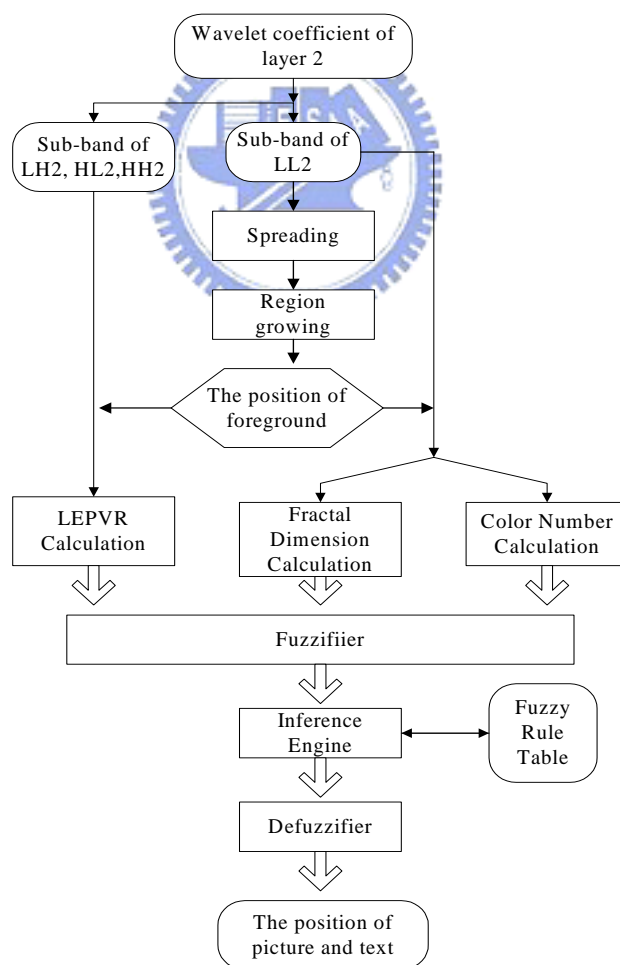


Fig.2 The flowchart of fuzzy picture-text segmentation algorithm.

The flowchart of the algorithm is shown in Fig.2. Details of the algorithm are explained in the following subsections.

A. Spreading and region growing for blocks extraction

The coefficients of LL2 band are used to perform block extraction. The proposed block extraction method is to divide the foreground of document images into text components and picture components. Before the process, we need to convert the coefficients of LL2 band into bi-level data, and use the thresholding method to decide the location of foreground and background. However, the pixel numbers of background are more than foreground's. In order to make the boundary of foreground more obvious, the algorithm uses a threshold value from the mean value. The threshold value is $(Mean - Variance)$. We can realize the influence of bias in Fig.3.

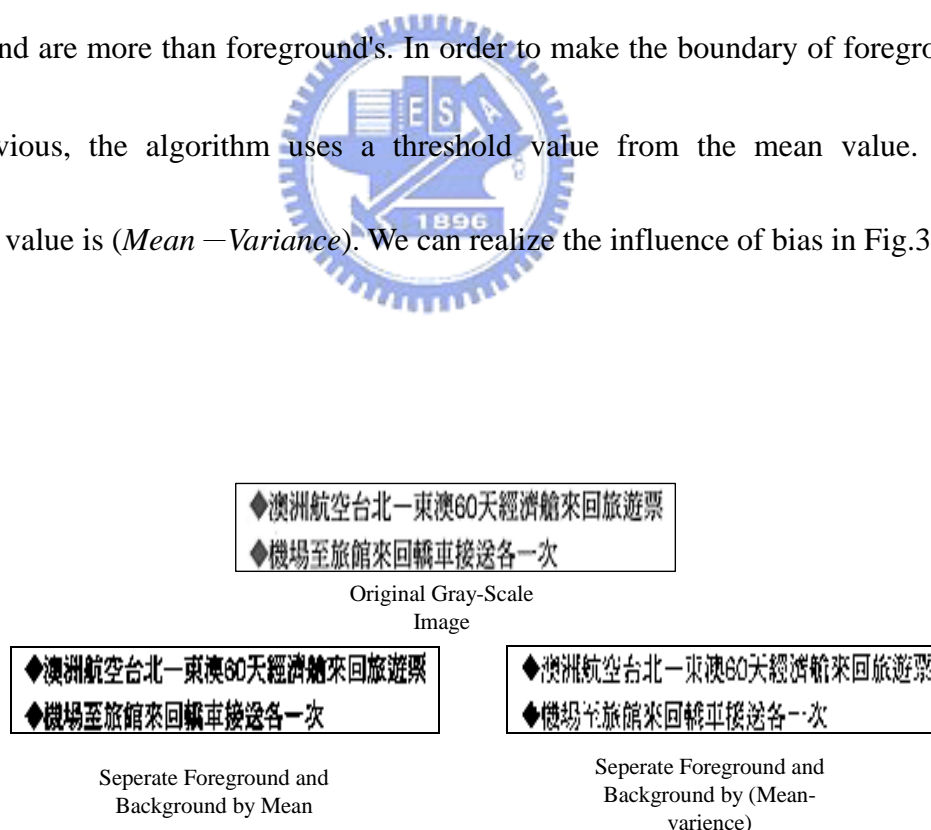


Fig.3 The influence of threshold.

The pixels of foreground and noise will be all extracted by a thresholding method in the same time. Therefore, we use the Constrained Run Length Algorithm (CRLA) to remove noise pixels. The algorithm was proposed by Wahl *et al.* [6] to preserve the pixel when it comes from the valid continuous pixels. For example, there is a binary string, 11001000001000011, with a constraint $C=4$ to the run length of 0s, if the number of consecutive 0s is less than or equal to C , these 0s must be replaced with 1s; otherwise, they are reserved. As a result, the above binary string is converted into the sequence, 11111000001111111. Some noises are eliminated by the method.

The CRLA is performed in horizontal and vertical directions, and the bi-level images, " M_v " and " M_h ", are obtained, respectively. Then, we apply the "OR" operator on M_v and M_h pixel by pixel, and get a bi-level spreading image, M_{hv} , which merges the neighboring pixels in both direction.

Therefore, the methods of thresholding, CRLA and logic operation are called the spreading process. After the spreading process, the bi-level spreading image M_{hv} is processed by the region growing method to gather the foreground pixels into rectangle blocks. The steps of region growing method are described below:

Step 1. Collect the foreground pixels of image M_{hv} row by row.

Step 2. Compare the foreground pixels collected from Step 1 with the current blocks. If there exists any overlap between the foreground pixels and

blocks, the foreground pixels and blocks are merged into the same block.

If there is no overlap, make a new block for the foreground pixels.

Step 3. After region growing, every block will be checked. If the block is neither growing bigger nor being a new one, stop the block's growing and regard it as an isolated block.

Step 4. Check if there is any overlap between blocks or not. Merging the overlapping blocks into the same block.

Step 5. If Mh_v comes to the last row, then go to Step 6; if not, return to Step 1.

Step 6. Change all existing blocks into isolated blocks. If there is any overlap between blocks, merge the overlapping blocks into the same block.

Step 7. Delete those smaller noise blocks.

Step 8. The end.

After the processes of spreading and region growing, we got all foreground blocks of the image Mh_v . An example is shown in Fig.4.

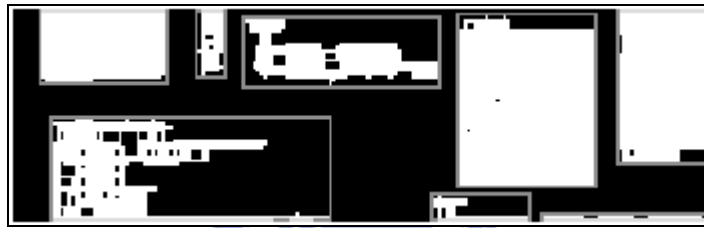
We can calculate the local edge projection variance ratio, the color number, and the fractal dimension from each foreground block.



(a) Original document image.(200 dpi, image size=768×256)



(b) The binary image of sub-band LL2.(image size=384×128)



(c) The binary image of sub-band LL2 after CRLA and region growing.

(image size=384×128)

Fig.4 Example of CRLA and region growing.

Those calculating methods are described as follows.

B. The calculation of local edge projection variance ratio

It is assumed that texts are written in horizontal or vertical direction. When the edge information is projected toward the vertical direction of text strings, the projection histogram would variation regularly. In addition, the projection magnitudes of text components are larger than those of picture components. If the edge

information is not projected on the vertical direction of text strings, the variation of the projection histogram will be irregularly. This property can be used to decide the direction of text strings. The horizontal or vertical edge projection is used to decide the direction of text strings.

Furthermore, since the variation of edge projection is different between text components and picture components, it can be applied to distinguish text components or picture components from foreground blocks. After discrete wavelet transform, the edge projection in high frequency bands (LH, HL, and HH) is more obvious than the one in low frequency band (LL). Therefore, the edge projection is calculated from the binary image which combines the binary images of high frequency bands (LH, HL, and HH) using logical *OR* operator. Fig.5 shows the vertical and horizontal edge projection of text-image in high frequency band.

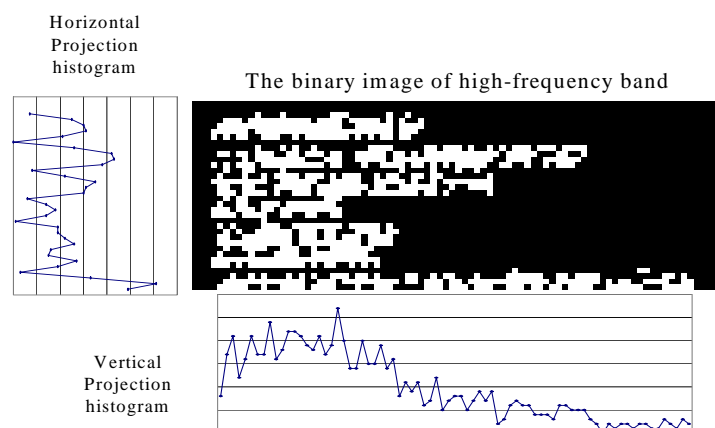


Fig.5 The edge projections of text components in high frequency band.

The variation of edge projection is regular in text components, and irregular in picture components. The edge projection variance ratio is defined by

$$\text{Edge projection variance ratio (EPVR)} = \frac{1}{\text{Mean}} \times \sqrt{\sum_i (P(i) - \text{Mean})^2} \quad (1-1)$$

in which $P(i)$ is the magnitude of i th projection and Mean is the average of the projection histogram.

The edge projection variance ratio shows the variation in projection histogram. By considering those two histograms shown in Fig.6, the left one is the projection histogram of text component and the right one is the projection histogram of picture component. We find these two projection histograms share the same EPVR. However, only the left diagram reveals the property of text.



(a)The projection of text component (b)The projection of picture component

Fig.6 Two kinds of projection histogram.

So we modify the equation (1-1) below:

$$\text{Local edge projection variance ratio (LEPVR)} = \frac{1}{\text{Mean}} \times \sqrt{\sum_i (P(i) - \text{LMean}(i))^2} \quad (1-2)$$

, where $P(i)$ is the magnitude of i th projection, $Mean$ is the average of projection, and

$$LMean(i) = \frac{1}{N} \times \sum_{k=i-N/2}^{i+N/2-1} P(k), \quad N \text{ is the width of the sliding window.}$$

$LEPVR$ uses the local mean to compute the local standard deviation used to substitute the global one. Therefore, the $LEPVR$ of text components is larger than that of picture components.

C. The calculation of color number

In general, the RGB-plane of a color document image is transformed to the YUV-plane before further processing. Because all color information can be collected in the UV-plane, the color number will be calculated in the UV-plane of LL2 band. Because the wavelet transform 9/7 filter is adopted in this work, the dynamic range of UV-coefficients, shown in Table 1, would be amplified 4 times after the wavelet transform.

Table 1. The dynamic range of UV-plane coefficient

	Original dynamic range	First level	Second level
U -plane	± 112	± 448	± 1792
V-plane	± 158	± 632	± 2528

In Table 1, the coefficients in second-level wavelet transform could be enlarged to 16 times at maximum, but it would not achieve the maximum value in most cases. However, if we reduce the coefficients 16 times, the range would be too small.

Therefore, minimizing 8 times is more reasonable in current situation.

The extraction of color number from foreground blocks is performed as follows:

Step 1. Calculating the histogram of UV-plane.

Step 2. Set LB as the threshold value, calculate the effective color range

($ColorRange$) whose pixel number is larger than LB .

$$LB = \begin{cases} 10, & \text{If } 1000 \leq TotalPixel \\ \frac{TotalPixel}{3}, & \text{If } 400 \leq TotalPixel < 1000 \\ 100, & \text{Otherwise} \end{cases} \quad (1-3)$$

Step 3. Define the effective mean value ($MeanPixel$) of pixels as:

$$MeanPixel = \frac{TotalPixel}{ColorRange} \quad (1-4)$$

Step 4. Set $\frac{MeanPixel}{2}$ as the threshold value. Calculate the number of colors

whose peak values of histogram are larger than the threshold value.

This number of colors is called *color number* in UV-plane.

We can calculate the color numbers of A and B in UV-plane. We define the value of $A \times B$ to be the color number of the foreground block.

D. The calculation of fractal dimension

The fractal dimension indicates the complexity of images. The more complex image is, the larger the fractal dimensions. However, images are two-dimensional, so the maximum fractal dimension value is 2 theoretically.

There are many ways to calculate the fractal dimension (FD). Here, we use the Two Dimension Box Counting Method [25] which is easier and faster to calculate compared to others. The steps of Two Dimension Box Counting Method are listed below:

Step 1. Assume all boxes are square. Set the minimum value (Box_{min}) and the maximum value (Box_{max}) of box, and let Len equal to the width of Box_{min} . To partition the image into blocks, the width of block is the value of Len .

Step 2. From the upper left side of image, search whether there exists any foreground pixel in the box whose width is the value of Len . If yes, plus 1 to the counter ($Count(Len)$). Apply this method to the whole image from left to right and up to down. After the process, the number of boxes contained the foreground pixel is obtained.

Step 3. If $Len \leq Box_{max}$, Len is increased by 1, and go to Step 2, else go to Step 4.

Step 4. Draw the logarithm figure of $\frac{1}{Len}$ and $Count(Len)$.

Step 5. Use the Least Square Error Method to get the line which is the nearest to the curve of the logarithm figure in step 4. Then define the slope of the curve as the fractal dimension of the image.

Theoretically, fractal dimension would not change in any scale, meaning that the characteristics of fractal dimension will be the same when enlarge or reduce the size of image. However, the property will be slightly changed because of the limited resolution in computer display. Therefore, we need to know the variation when we

lower down the resolution of image. Fig.7 illustrates the images with text component and picture component. The FD difference between the original image and the image in LL2 band is shown in Table 2.

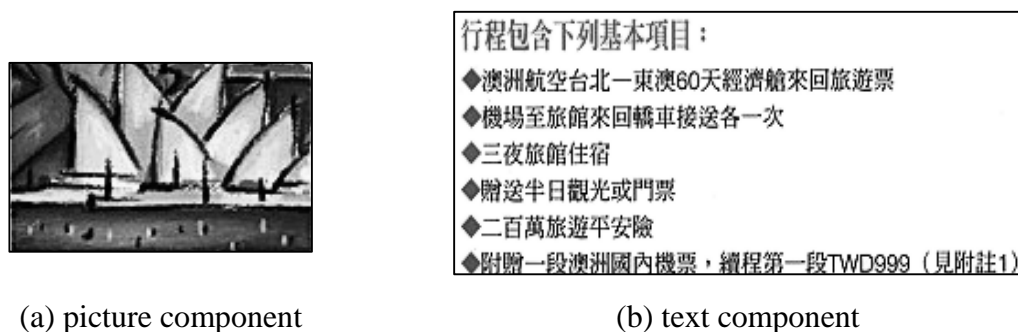


Fig.7 Images with text/picture component.

Table 2. The fractal dimension influenced by the resolution of images.

	FD of original image	FD of LL2 image
Fig.7 (a)	1.809	1.861
Fig.7 (b)	1.24	1.178

There are two information concluded from Table 2.

- (1) The FD in original image is very close to the FD in the low-resolution image. The variation range is between ± 0.1 .
- (2) FD in text components is smaller than in picture components.

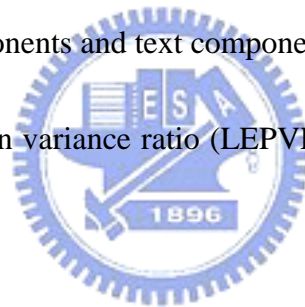
The information (1) confirms that we can calculate FD in LL2 band to replace FD in original image. The information (2) shows the difference of FD in picture and text components.

E. Fuzzy logic decision system

We can use the local edge projection variance ratio, color number, and fractal dimension to segment the foreground blocks to picture components or text components. In some cases, the local edge projection variance ratio, color number, and fractal dimension of text components would show the same characteristics as those of picture components do. In other words, those parameters have the fuzzy characteristic. Therefore, we can use the fuzzy theorem to classify the foreground blocks.

In general, picture components and text components have three characteristics:

- (1) If the local-edge projection variance ratio (LEPVR) is large, it is likely to be text components.
- (2) If the color number is large, it is likely to be picture components.
- (3) If the FD is large, it is likely to be picture components.



The three characteristics are showed in Fig.8. The fuzzy membership functions shown in Fig.9 are used for the testing of many images and the recognition of human eyes toward color document image.

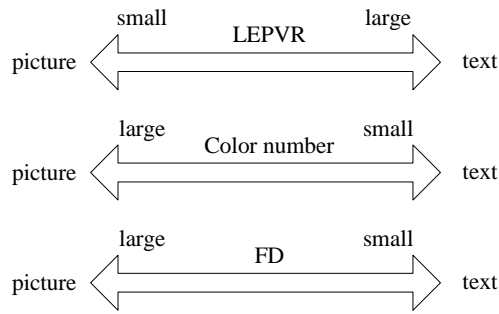


Fig.8 The characteristics of three parameters.

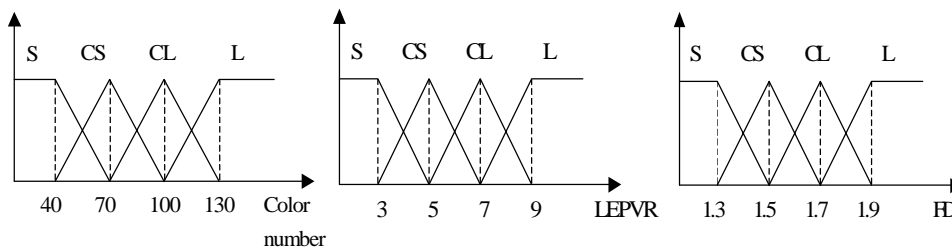
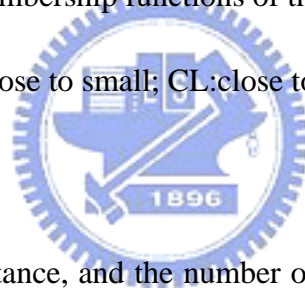


Fig.9 The membership functions of three parameters.

(S:small; CS:close to small; CL:close to large; L:large)



By using Hamming distance, and the number of reliability of three parameters, we set a Fuzzy Rule Table ($4 \times 4 \times 4$ cubic) as shown in Fig.10. First, filling into 0 (the most likely be picture components) in one edge of this cubic block, then fill into 18 (the most likely be text components) in the opposite position of 0. From 0 to 18, no matter which the route is, the Hamming distance is equal to 9. Then, filling number based on equal Hamming distance, we got an initial fuzzy rule table where the three parameters maintain the same degree of importance.

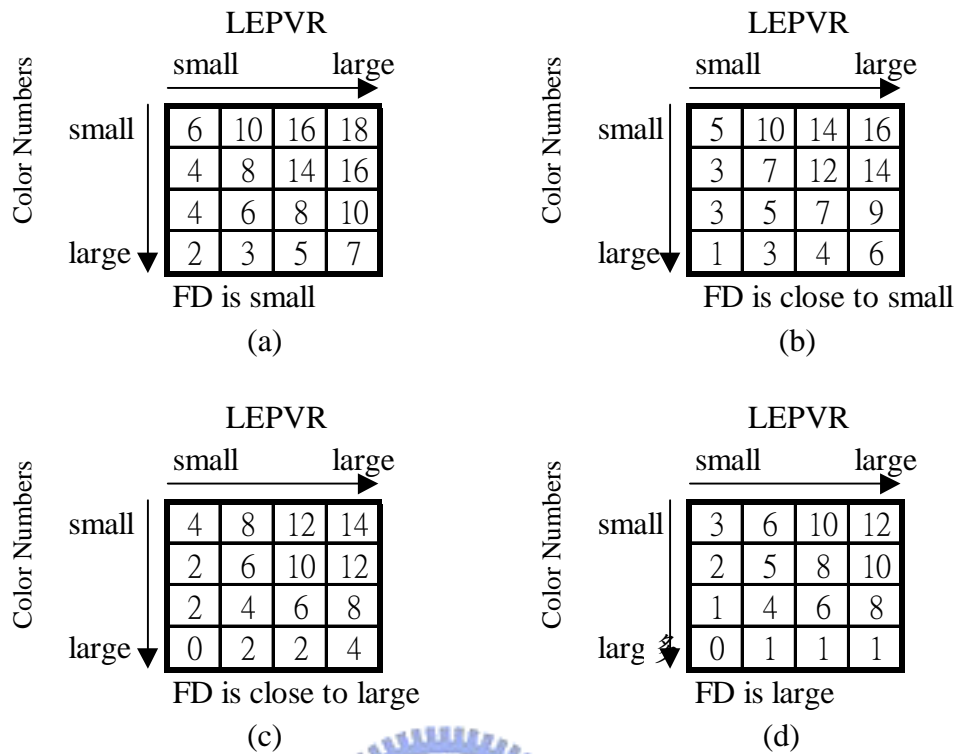


Fig.10 The fuzzy rule table.

In the fuzzy rule table, the large number is more likely to be the text components. On the contrary, the small number is more likely to be the picture components. Even though the text components, which contain only a color in the same text component, it would not include more color number than picture components do. Therefore, we can emphasize the weighting of “color number is large” for picture components. However, for gray or single color images, since there is only one color, this information (color number) is not reliable for text components and picture components. Under this circumstance, it is needed to rely on LEPVR and FD, so we need to put more emphasis on the weighting of these two parameters for gray or single color images.

Through these methods discussed above, we design the fuzzy rule table in Fig.10. The corresponding function in every rule is shown in Fig.11.

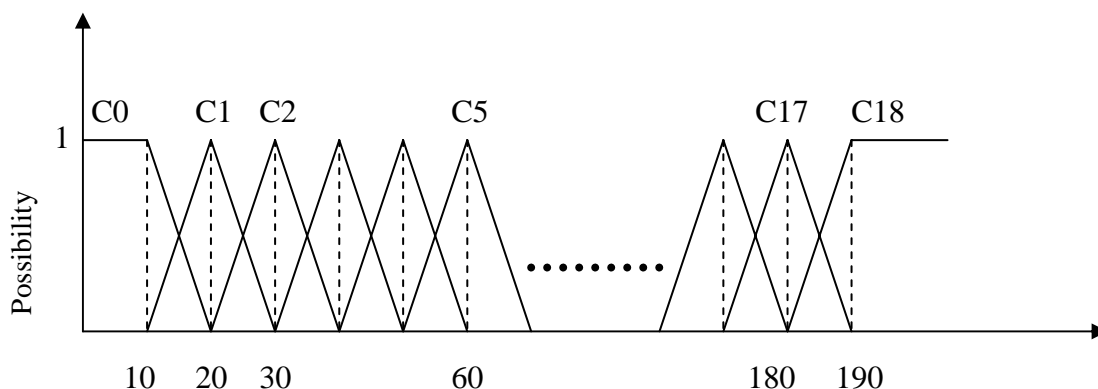


Fig.11 Possible fuzzy quantization by triangular-shaped fuzzy numbers

A number of defuzzification methods leading to distinct results were proposed in many literatures. Each method is based on some rationale. In this method, the center of area method [23] is selected to define the defuzzified value. The defuzzified value is set 100 : if the defuzzified value of foreground blocks is smaller than 100, it belongs to picture components; and if the defuzzified value of foreground blocks is larger or equal to 100, then it belongs to text components.

2.4 Color document images compression method

After fuzzy picture-text segmentation algorithm is applied, the document images are classified into text components and picture components. In this section, zerotree

coding method is used to compress the coefficients of wavelet transform for picture components, but the coefficients of text components need to be removed. For text-components, we have to extract the colors of text from the original document image. Each color of text components will form a single color plane. The color plane is compressed by the Modified Huffman code.

However, the run-length of blank pixels between text-lines can be very long, and may reduce the coding efficiency of Huffman code. To solve this problem, we use Modified Run-length code to deal with the long run-length, and Modified Huffman code codes the short run-length.

The flowchart of proposed compression algorithm is shown in Fig.12.

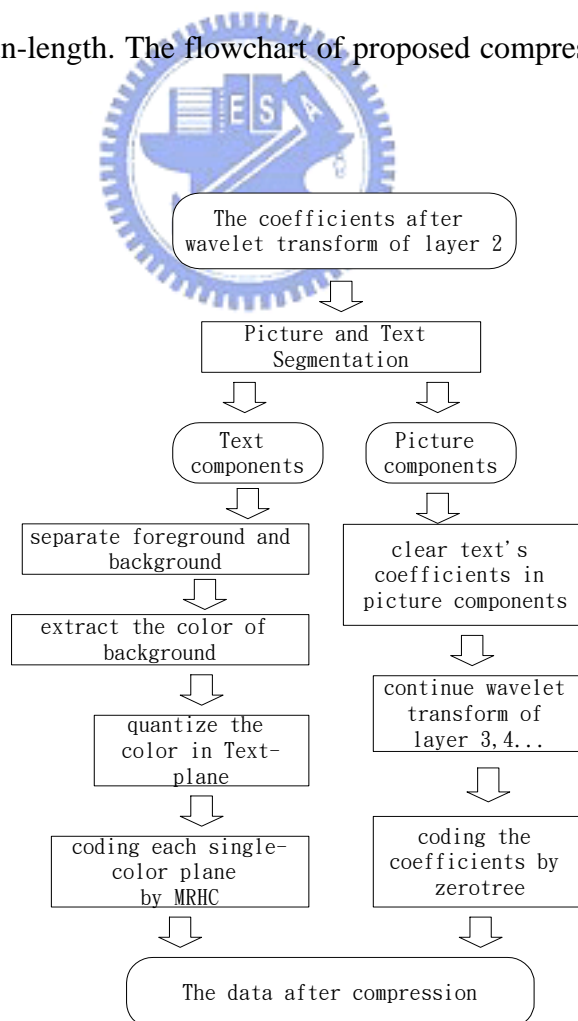


Fig.12 The flowchart of proposed compression algorithm.

A. Color quantization in text components

For the text components, they are separated into several single-color planes. Therefore, we have to decide the number of colors in the text components by calculating the foreground histogram ($His[i]$) from the UV-plane. The steps are listed as follow:

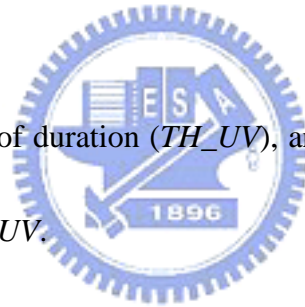
Step 1. Let $His[i]$ pass the low-pass filter, and set the result as $Fhis[i]$.

Step 2. Find the maximum values of $Fhis[i]$.

Step 3. Get rid of the maximum value of $Fhis[i]$ which is smaller than

$TH_LowBound$.

Step 4. Set a threshold of duration (TH_UV), and calculate the local-maximum between $\pm TH_UV$.



In Step 1, the purpose of low-pass filter is to eliminate high frequency noise. In Step 2, all the peak values in $FHis[i]$ can be found, and the smaller peaks in Step 3 are deleted. In Step 4, we can obtain the major colors of text components and use the color information to segment the text component into many single-color planes by difference colors. The meaning of TH_UV is the minimum difference of colors that human eyes are able to discern. Here the value of TH_UV is set to 15 and $TH_LowBound$ is set to 10.

B. Using Modified Run-Length Huffman Code (MRLHC) to encode text

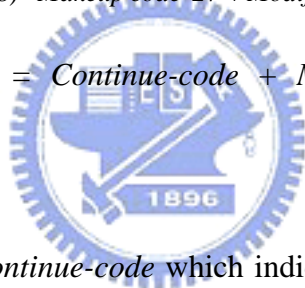
The text components are segmented into several single-color planes. Those planes are compressed by Modified Run-Length Huffman Code.

Modified Run-Length Huffman Code is especially designed to code the long run, so it is expected that the code has the ability to handle extreme long runs. The algorithm is designed as follows:

$$\text{Run-Length} = 1728 \times \text{Multi-code} + 64 \times \text{Makeup code} + \text{Terminate code}$$

$$\text{Run-length} = \begin{cases} 0 \sim 63 & \text{only Terminate code} \\ 64 \sim 1727 & \text{Makeup code '1' } \sim 26 + \text{Terminate code} \\ 1728 \sim (2^{16} \times 1728) & \text{Makeup code '27' } + \text{Modified Run Length code} \end{cases}$$

$$\text{Modified Run Length code} = \text{Continue-code} + \text{Multi-code} + \text{Makeup code} + \text{Terminate code}.$$



The first four bits are *Continue-code* which indicates the number of bits can be read in *Multi-code*. By using this approach, the maximum Run-Length is represented as long as $1728 \times 2^{16} = 27 \times 2^{22}$. The maximum Run-Length in A3-size paper scanned in 300dpi is approximate by 8×2^{22} . Therefore, Using Modified Run Length Code can represent an A4-size paper just in one code.

C. Compression method of picture components

When the blocks of texts are extracted from the color document image, many gaps are located at the color document image. The boundary of those gaps produces many high-frequency coefficients after wavelet transform. The zerotree coding

algorithm will waste bits to encode those high-frequency coefficients. In order to improve the efficiency of compression, those gaps must be compensated with appropriate coefficients. Because the fuzzy picture-text segmentation algorithm extracts the texts based on the coefficients of wavelet transform, it just need to compensate the coefficients of text components from the coefficients of wavelet transform. Our method is to directly compensate the text components by the average of neighboring data in the LL2 band, and set the coefficients of text components in HLi, LHi, and HHi ($i=1,2$) bands as zeroes. After compensating the text components in the sub-band coefficients, the wavelet transform is adopted for the coefficients of LL2 band continuously. Then, the zerotree coding algorithm is used to encode the coefficients of wavelet transform.

2.5 Experimental results



The proposed coding algorithm was simulated on Window 2000 (Pentium III 700, 128 MB RAM) with programs written in C++ language. In this study, we used the 24-bit true color image format and 200 dpi in processing. Each pixel in a 24-bit true color image is characterized by its R, G, and B color values, and 8 bits represent every value. Fig.13 and Fig.14 show the comparison of the JPEG and the new compression algorithm. The time of performing fuzzy picture-text segmentation in Fig.13(c) and Fig.14(c) are both 0.02 sec. Experimental results show that the compression algorithm based on fuzzy picture-text segmentation has achieved better and clearer quality of pictures and texts than those from JPEG.

2.6 Concluding remarks

Traditional color image-compression standards such as JPEG are inappropriate for document images. JPEG's application relies on the assumption that the high spatial frequency components in images can be essentially removed without much quality degradation. While this assumption holds for most pictures of natural scenes, it does not work for document images. The texts require a lossless coding technique to maximize readability. This chapter has proposed a new compression method with promising performance on color document images. The method uses different compression algorithms based on fuzzy picture-text segmentation for sub-images with different characteristics. The fuzzy picture-text segmentation algorithm is based on the coefficients of wavelet transform. It is fast to find out the text components and picture components from the coefficients of wavelet transform. We have also compared our method with JPEG. The results show that the new compression method has achieved better and clearer quality than those from JPEG.





“SPOX was the number-one factor in completing our GSM basestation design six months ahead of schedule.”

Andreas Wosqien , SIEMENS AG

Fig.13(a) The original image(512×1024, scanned by ScanMaker V600 at 200dpi)



“SPOX was the number-one factor in completing our GSM basestation design six months ahead of schedule.”

Andreas Wosqien , SIEMENS AG



“SPOX was the number-one factor in completing our GSM basestation design six months ahead of schedule.”

Andreas Wosqien , SIEMENS AG

Fig.13(b) JPEG image(CR=104.1)

Fig.13(c) Proposed method(CR=112.3)



Fig.14(a) The original image(1024×512, scanned at 200dpi)



Fig.14(b) JPEG image(CR=83.1)



Fig.14(c) Proposed method(CR=84.31)

CHAPTER 3

THE COMPRESSION ALGORITHMS FOR COMPOUND DOCUMENT IMAGES WITH LARGE TEXT/BACKGROUND OVERLAP

This chapter presents two algorithms for compressing image documents, with a high compression ratio of both color and monochromatic compound document images. The proposed algorithms apply a new segmentation method to separate the text from the image in a compound document in which the text and background overlap. The segmentation method classifies document images into three planes: the text plane, the background plane, and the text's color plane. Different compression techniques are used to process the text plane, the background and the text's color plane. The text plane is compressed using the pattern matching technique, called JB2. Wavelet transform and zerotree coding are used to compress the background plane and the text's color plane. Assigning bits for different planes yields high-quality compound document images with both a high compression ratio and well presented text. The proposed algorithms greatly outperform the famous image compression methods, JPEG and DjVu, and enable the effective extraction of the text from a complex background, achieving a high compression ratio for compound document images.

3.1 Introduction

A color page of A4 size at 200 dpi is 1660 pixels wide and 2360 pixels high; it occupies about 12 Mbytes of memory in an uncompressed form. The large amount of data prolongs transmission time and makes storage expensive. Texts and pictures cannot be compressed by a single method like JPEG since they have different characteristics. Digitized images of compound documents typically consist of a mixture of text, pictures, and graphic elements, which have to be separated for further processing and efficient representation. Rapid advances in multimedia techniques have enabled document images, advertisements, checks, brochures and magazines to overlap text with background images. Separating the text from a compound document image is an important step in analyzing a document.

Document image segmentation, which separates the text from the monochromatic background, has been studied for over ten years. Segmenting compound document images is still an open research field. Traditional image compression methods, such as JPEG, are not suitable for compound document images because such images include much text. These image data are high-frequency components, many of which are lost in JPEG compression. Text and the high-frequency components thus become blurred. Then, the text cannot be recognized easily by the human eye or a computer. The text contains most information, separating the text from a compound document image is one of the most significant areas of

research into document images. The traditional compression method cannot meet the needs of the digital world, because when compound documents are compressed at a high compression ratio, the image quality of the text part usually becomes unacceptable.

Many techniques have been developed to segment document images. Some approaches to processing monochromatic document images have already been proposed. Queiroz *et al.* proposed a segmentation algorithm based on block-thresholding, in which the thresholds were found in a rate-distortion analysis method [26]. Some other systems based on a prior knowledge of some statistical properties of the various blocks [11]-[15], or textual analyses [16],[17],[27] have also been subsequently developed. All these systems focus on processing monochromatic documents. In contrast, few approaches to analyzing color documents have been proposed. Suen and Wang [19] utilized geometric features and color information to classify segmented blocks into lines of text and picture components. Digipaper [28] and DjVu [20],[21] are two image compression techniques that are particularly geared towards the compression of a color document image. The basic idea behind Digipaper and DjVu is to separate the text from the background and to use different techniques to compress each of those components. The image of the text part in DjVu is encoded using a bi-level image compression algorithm called JB2, and the background image is encoded by a progressive, wavelet-based compression algorithm called IW44.

These methods powerfully extract text characters from a simple or slowly varying background. However, they are insufficient when the background includes sharply varying contours or overlaps with text. Extracting the text when the color of the overlapped background is close that of the text is especially difficult. Finding a text segmentation method of complex compound documents remains a great challenge and the research field is still young.

This chapter proposes a new segmentation algorithm for separating text from a complex compound document in 24-bit true-color or 8-bit monochrome. Two compression algorithms that yield a high compression ratio are also proposed. The technique separates text from background image after segmenting the text. Therefore, it has many applications, such as to color facsimiles and document compression. Moreover, the segmentation algorithm can be used to find characters in complex documents with a large text/background overlap.



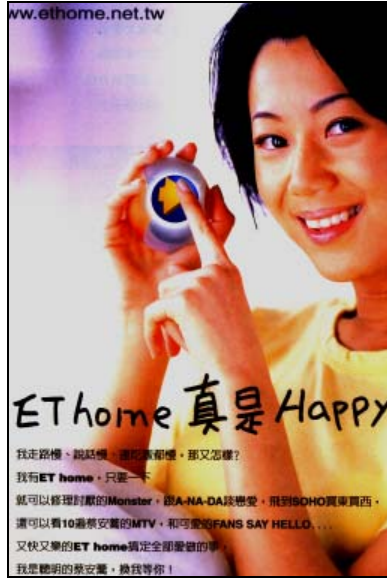
3.2 Text segmentation algorithm

When a document image is captured from a scanner, it can include several different components, including text, graphics, pictures, and others. The textual parts must be separated from a compound document image, whether effective compression or optical character recognition (OCR) is intended.

This section introduces a new extraction algorithm that can separate text from a complex, color or monochromatic compound document, as in Fig.15.



(a) Test image A
size=1024×1536



(b) Test image B
size=1024×1536



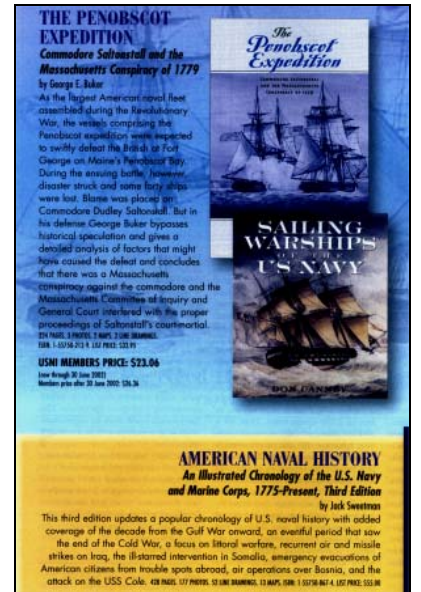
(c) Test image C
size=1024×1536



(d) Test image D
size=1024×1536

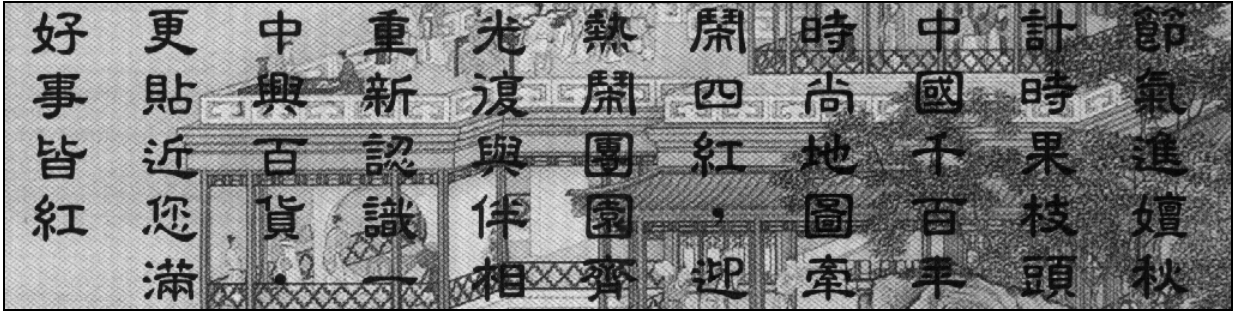


(e) Test image E
size=1344×1792

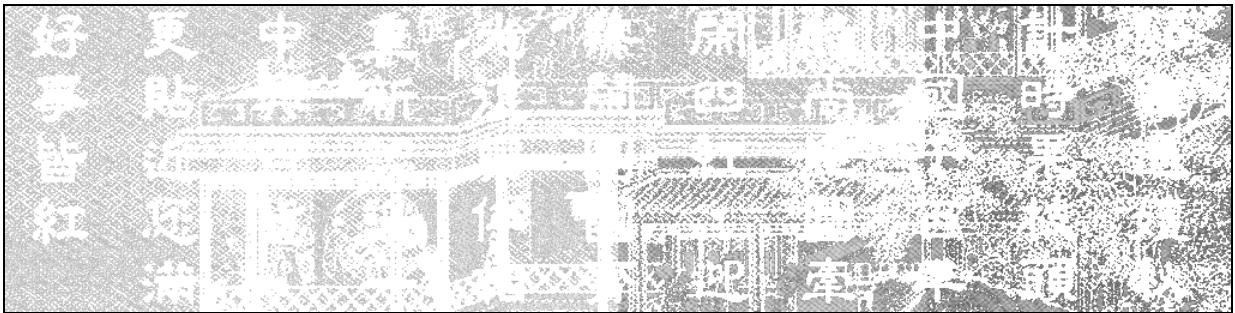


(f) Test image F
size=1024×1536

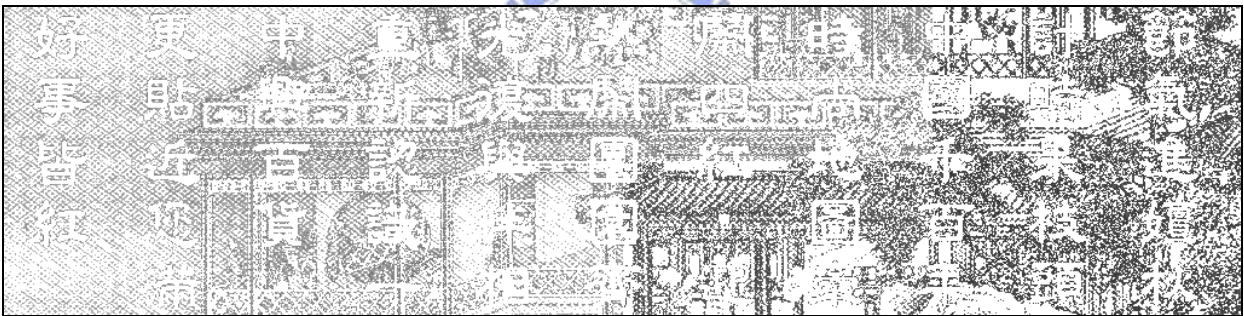
Fig.15 The original full images (200 dpi)



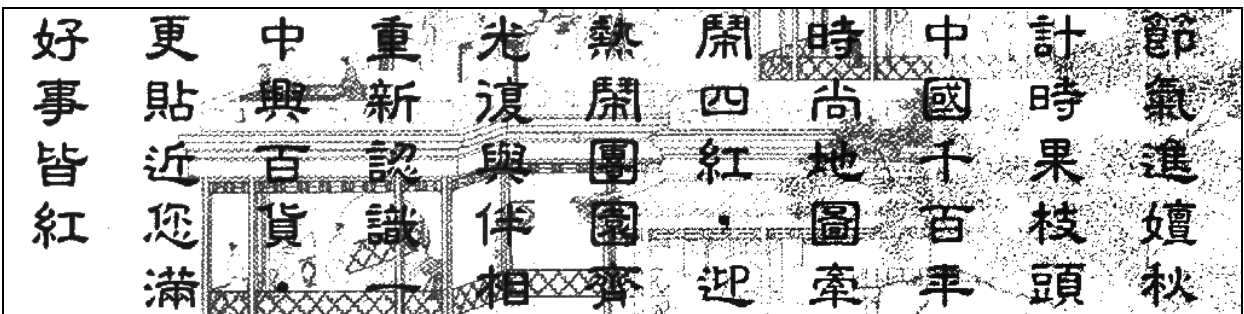
(a) Original Y-plane image



(b) Bright plane



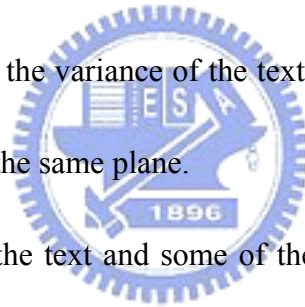
(c) Medium plane



(d) Dark plane

Fig.16 The planes after clustering algorithm

The features and texture of a complex compound document can be very complicated. In a pilot test, when a color document image was converted into a monochromatic image, the gray value of the image of text differed slightly from the gray value of the background image. However, the image of text cannot easily be directly separated from the background image because the difference between the gray values is too small. Therefore, two phases are used to accomplish the desired purpose. In the first phase, which involves color transformation and clustering analysis, the monochromatic document image is partitioned into three planes, the dark plane, the medium plane, and the bright plane, as depicted in Fig.16. The color of the text is almost all the same, so the variance of the text's grayscale is small. Therefore, all the text can be grouped in the same plane.



When the text is black, the text and some of the background with a gray value close to that of the text is put in the dark plane. In contrast, the text is put into another plane if it is not black. Thus, the text and some noise are coarsely separated from the background. In the second phase, an adaptive threshold is determined to refine the text by adaptive binarization and block extraction. The two phases in which the algorithm extracts the text from the background are shown below.

A. Color Transformation

The color transformation technique is used to transfer a color document image to the YUV plane and the Y-plane (grayscale image) is used to segment the text from the

complicated background image.

B. Clustering analysis

In general, after a color document image is converted into a monochromatic one, the textures of the original color image are still present in the converted grayscale image. The difference between the text's gray value and that of the overlapping background image is small. Thus, a clustering algorithm is used to split the grayscale images. Clustering analysis roughly separates text from a background image. First, it extracts as many as possible of the different textures of an image. The text is embedded in one of the planes.

The clustering algorithm is described below.

Step 1. Partition the $M \times N$ grayscale image $A(i, j)$ into p sub-block images $x_n(i, j)$.

Each sub-block $x_n(i, j)$ is of $K \times L$, where $n=1, 2, \dots, p$.

Step 2. Calculate the mean of gray value, m_n , and the standard derivation, σ_n , of each $K \times L$ sub-block image. For the n^{th} sub-block image $x_n(i, j)$, the mean and standard derivation are computed as

$$m_n = \frac{\sum_{i,j} x_n(i, j)}{K \times L} ; \quad (3-1)$$

$$\sigma_n = \sqrt{\frac{\sum_{i,j} [x_n(i, j) - m_n]^2}{K \times L}} . \quad (3-2)$$

Step 3. Split $x_n(i, j)$ according to mean and standard derivation. Define two centers,

C'_{n1} and C'_{n2} , by $C'_{n1} = m_n + 0.5 \times \sigma_n$ and

$$C'_{n2} = m_n - 0.5 \times \sigma_n. \quad (3-3)$$

Step 4. Calculate the absolute difference of each pixel of $x_n(i, j)$ to C'_{n1} and C'_{n2}

using

$$D'_{ij,1} = |x_n(i, j) - C'_{n1}| \text{ and}$$

$$D'_{ij,2} = |x_n(i, j) - C'_{n2}|. \quad (3-4)$$

Then, $x_n(i, j)$ partition into two clusters $\eta_k (k=1,2)$ according to

$$\eta_1 : \{x_n(i, j) | D'_{ij,1} \leq D'_{ij,2}\}, \text{ and}$$

$$\eta_2 : \{x_n(i, j) | D'_{ij,1} > D'_{ij,2}\} \quad (3-5)$$

Step 5. Calculate the mean m_{nk} and standard derivation σ_{nk} of the two clusters

$\eta_k (k=1,2)$ using Equations (3-1) and (3-2), respectively.

If $\sigma_{n1} > \sigma_{n2}$, then center $C_{n3} = m_n - 0.5 \times \sigma_n$, and compute the two new centers

C_{n1} and C_{n2} using $C_{n1} = m_{n1} + 0.5 \times \sigma_{n1}$, and

$$C_{n2} = m_{n1} - 0.5 \times \sigma_{n1}. \quad (3-6)$$

Else, (if $\sigma_{n1} < \sigma_{n2}$, then center $C_{n3} = m_n + 0.5 \times \sigma_n$, and compute the two new centers

C_{n1} and C_{n2} using $C_{n1} = m_{n2} + 0.5 \times \sigma_{n2}$, and

$$C_{n2} = m_{n2} - 0.5 \times \sigma_{n2}. \quad (3-7)$$

Step 6. After Step 5, three clustering centers $C_{nk} (k=1,2,3)$ are obtained. $x_n(i, j)$

is partitioned into three clusters ψ_k ($k=1,2,3$) according to,

$$\begin{aligned}\psi_1 &: \{x_n(i, j) | D_{ij,1} < D_{ij,2} \text{ and } D_{ij,1} < D_{ij,3}\}; \\ \psi_2 &: \{x_n(i, j) | D_{ij,2} < D_{ij,1} \text{ and } D_{ij,2} < D_{ij,3}\}; \\ \psi_3 &: \{x_n(i, j) | D_{ij,3} < D_{ij,1} \text{ and } D_{ij,3} < D_{ij,2}\};\end{aligned}\quad (3-8)$$

$$\begin{aligned}\text{where, } D_{ij,1} &= |x_n(i, j) - C_{n1}|, \\ D_{ij,2} &= |x_n(i, j) - C_{n2}|, \text{ and} \\ D_{ij,3} &= |x_n(i, j) - C_{n3}|.\end{aligned}\quad (3-9)$$

Repeat Steps 2 to 6 until all of the sub-block images $x_n(i, j)$ ($n=1,2,\dots,p$) have been processed.

The optimal partition of the images depends on the intensity distribution of background images and the lengths, sizes and layouts of the text strings. However, analyzing those parameters is very complex. Therefore, images are partitioned into equal sub-blocks for simplicity. This study used $K=256$ and $L=128$, and a value of p that depended on the image size. The constants were determined empirically to ensure good performance in general cases.

C. Adaptive binarization

After the first phase, the gray values of each plane become simpler than those of the original document image. Then, the text is extracted from the background image using the thresholding algorithm. Thresholding techniques can be categorized into

two classes, global and local. The global thresholding algorithm uses a single threshold, while a local thresholding algorithm computes a separate threshold for each local region. This work utilizes a local thresholding algorithm. Set the threshold value TH_n of the dark and bright planes of the sub-block images $x_n(i, j)$ ($n=1,2,\dots,p$) to,

$$TH_n = m_{f_n} \pm \left(\frac{m_{f_n}}{m_{b_n} - m_{f_n}} \right) \times \sigma_{f_n} \quad (3-10)$$

, where m_{f_n} is the mean value calculated from the pixels in one of the planes, $x_n(i, j)$; m_{b_n} is the mean value calculated from the pixels in the other $x_n(i, j)$ plane, and σ_{f_n} is the standard deviation calculated from the pixels in the same plane $x_n(i, j)$ as m_{f_n} .



Restated, m_{f_n} is determined by the foreground pixels of the plane that is being processed, while m_{b_n} relates to the background pixels of the other two planes. From Eq. (3-10), the value of TH_n can be adapted to the gray value of sub-block $x_n(i, j)$. The foreground pixels in the dark plane are darker than the background pixels. The

adaptive thresholding value can be calculated as, $TH_n = m_{f_n} - \left| \frac{m_{f_n}}{m_{b_n} - m_{f_n}} \right| \times \sigma_{f_n}$.

The threshold value TH_n is biased to the left of the mean value m_{f_n} . The foreground pixels in the bright plane are brighter than the background pixels. The adaptive

thresholding value can be calculated as, $TH_n = m_{f_n} + \left| \frac{m_{f_n}}{m_{b_n} - m_{f_n}} \right| \times \sigma_{f_n}$. The

threshold value TH_n is biased to the right of the mean value m_{f_n} . Figure 17 gives an

example of the algorithm.

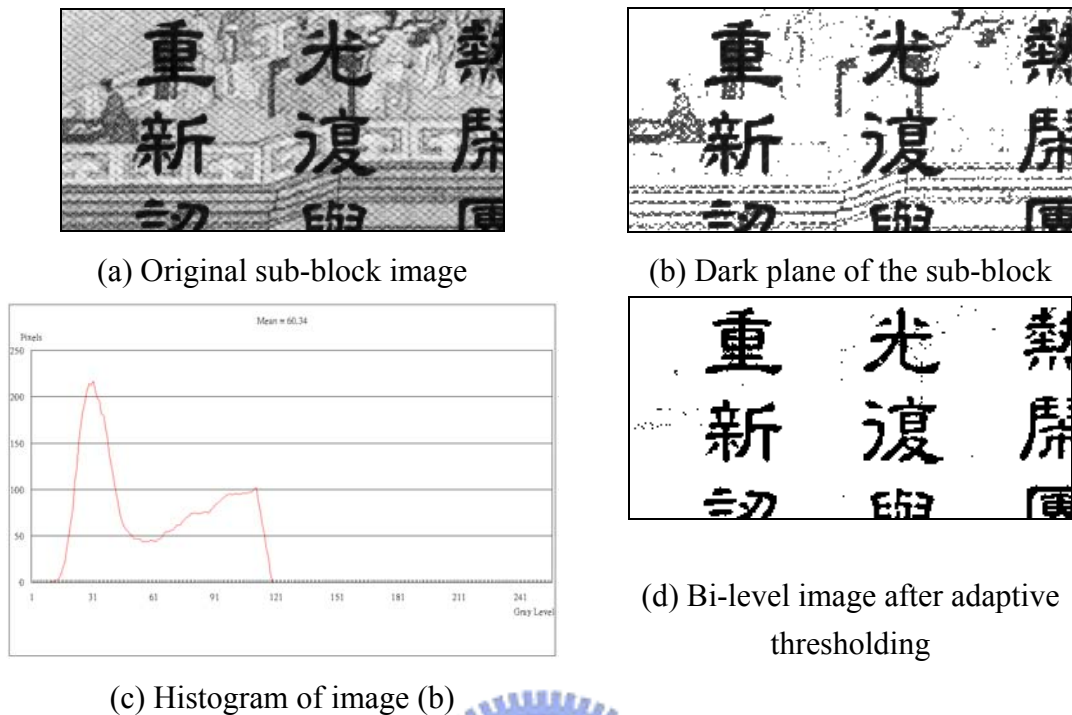


Fig.17 An example of text extraction algorithm image(size=256×128)

Figure 17(a) is the sub-block image $x_n(i, j)$, Figure 17(b) is the dark plane of the sub-block, and Fig. 17(c) shows the histogram of the sub-block image $x_n(i, j)$. The threshold value TH_n has a bias to the left of the mean value m_{f_n} to clarify the text. Figure 17(d) shows the bi-level image.

D. Spreading and region growing for block extraction

The pixels of foreground and noise are obtained simultaneously using the thresholding method. Accordingly, the isolated pixels are deleted and the Constrained Run Length Algorithm (CRLA) [7] is applied to remove noise pixels.

The CRLA is performed in horizontal and vertical directions, yielding the

binary images $Mh(i, j)$ and $Mv(i, j)$ ($1 < i < M$, $1 < j < N$), respectively. Then, the "AND" operator is applied to $Mh(i, j)$ and $Mv(i, j)$ pixel by pixel, and a binary image, $Mhv(i, j)$, which merges the neighboring pixels in both directions, is obtained.

The CRLA and the logic operation together constitute the spreading process, after which, the binary spreading image $Mhv(i, j)$ is processed using the region growing method to arrange the foreground pixels into rectangular blocks. The region growing method is described in the Chapter 2.

The processes of spreading and region growing yield the positions of the foreground blocks.

E. Distinguishing text from foreground blocks

The blocks that contain foreground pixels are extracted following the spreading and growing processes. The blocks that contain text strings must now be identified. In this study, three parameters, transition pixel ratio, foreground pixel ratio, and block size, are used to identify these blocks.

The transition pixel ratio is defined as,

$$T = \frac{\text{Total number of transition pixels in block}}{\text{Area of block}} \quad (3-11)$$

The transition pixel occurs at the boundary of foreground pixels. For example, in the following bi-level image,

0 0 0 1 1 0 1 0 0 1 1 1 0 0 0
 ▲ ▲ ▲ ▲ ▲

the marked pixels "▲" are the transition pixels, of which five are included. The foreground pixel ratio is defined as,

$$B = \frac{\text{Total number of foreground pixels in block}}{\text{Area of block}} \quad (3-12)$$

The ratio of the number of transition pixels to the area of the block indicates the complexity of the blocks, and the foreground pixel ratio reflects the density of the foreground pixels. The block size is defined as block width×block length. $T \leq 0.3$, $B \leq 0.5$ and $300 \leq \text{block size} \leq 30000$ are set to extract the text from foreground blocks.

3.3 Document image compression algorithm

As stated above, text and background images have different characteristics. Traditional image compression algorithms, such as JPEG, are unsuitable for document images. JPEG's use of local cosine transforms is based on the assumption that the high spatial frequency components in images can be removed without too much degradation of quality. While this assumption holds for most images of natural scenes, it does not hold for document images. A different compression method is required to code text accurately and efficiently to maximize its clarity. Text and the background image can be encoded by methods appropriate for bi-level and continuous-tone images, respectively.

The foreground/background representation was proposed in the ITU MRC/T.44

recommendation [31]. This prototype of document image representation is used in Xerox's XIFF image format, which presently uses CCITT-G4 to code the mask (text) layer, and JPEG to code the foreground (text color) and background layers. However, the compression ratio of MRC/T.44 is insufficient for document images. Thus, this foreground/background representation is used and two compression algorithms proposed for compound document images. In this work, pixels of text are extracted from a compound document image. The text plane is the mask layer.

Several gaps appear in the background image when pixels of text are extracted from it. The gaps are replaced by pixels with the average gray value of the neighboring pixels to improve the efficiency of compression. The foreground image is the color plane of the text. The color of the text can be obtained from the original image according to the position of the text. The pixels of the text are called used pixels, and the others are called unused pixels. Those unused pixels can be replaced by pixels of an appropriate color to enhance the compression. The color-filling algorithm is as follows.

Step 1. Mark the pixels in the mask as used pixels; the other pixels are unused pixels.

Step 2. Fill the gap with the color of the pixel which adjoins the used pixel, row by row. Mark the filled pixels as used pixels.

Step 3. Fill the gap with the color of the pixel next to the used pixel, column by

column. Mark the filled pixels as used pixels.

Step 4. Repeat the processes in Steps 2 and 3 until no unused pixels remain. The foreground plane is thus obtained.

Different planes are compressed using different compression methods.

(1) Mask plane: The text pixels, also called the mask, are represented by a single bit-plane. This bit-plane uses “1” to represent a text pixel and “0” to represent a background pixel. The text pixels are coded using JB2, which is a variation of AT&T’s proposal for the JBIG2 fax standard.

(2) Foreground plane: The text’s color, also called the foreground, is represented in a color plane. Neighboring text characters generally have identical color, so the color plane contains large areas of contiguous pixels of almost the same color. This color plane is coded using a wavelet-based compression algorithm [3].

(3) Background plane: The background image is coded by the same algorithm as that to code the foreground image.

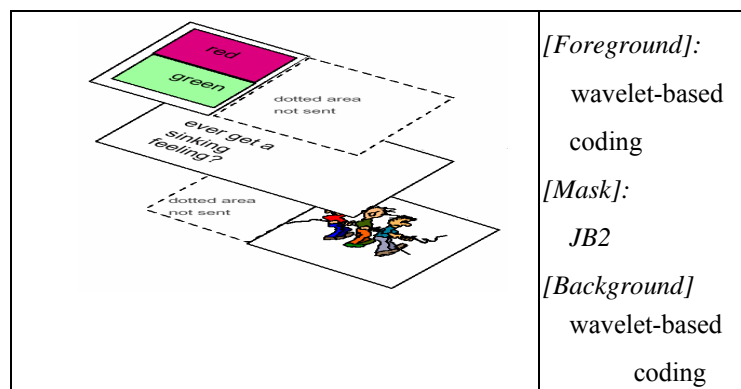


Fig.18 Document image compression format

Figure 18 depicts the compression format. This work proposes two compression

algorithms, CSSP-I and CSSP-II, respectively, to compress compound document images. Each component of the encoder is described below.

A. Method of compressing the mask plane

The mask image compression method uses JB2 [29]. JB2 is an algorithm proposed by AT&T's for the JBIG2 standard [30] for compressing fax and bi-level images. JB2 provides better compression than JBIG1 [32], which is often used for faxes, in both lossless and lossy compression of arbitrarily scanned images with scanning resolutions from 100 to 800 dpi. JB2 uses information in previously encountered characters and does not risk the introduction of the character substitution errors that are inherent in the use of OCR. The JB2 method has been proven to be approximately 20% more efficient than the JBIG1 standard in the lossless compression of bi-level images. By running JB2 in a controlled loss mode, this algorithm yields a compression ratio of about two to four times that provided by the JBIG1 method. In lossy mode, JB2 is four to eight times better than CCITT-G4 (which is lossless). It is also four to eight times better than GIF.

B. Method for compressing foreground/background plane

The combination of discrete wavelet transform and zerotree coding [3],[4],[33] has been proposed to compress a pure image with a high compression ratio. Such coding algorithms provide good image quality. This study uses the embedded zerotree wavelet (EZW) coding algorithm [3] as the algorithm for compressing

foreground/background images.

These compression methods are applied to mask, foreground and background images. The document image compression algorithm mentioned earlier is called compression algorithm CSSP-I.

Compression algorithm CSSP-I can extract the text plane from an overlapping background image and compress it using JB2. The compressed data thus obtained are approximately 50% of all the compressed data. Therefore, compression algorithm CSSP-II is proposed to improve the compression ratio.

Compression algorithm CSSP-II uses a downsampling method to reduce the data of the original document images. The downsampling method replaces each 2×2 pixel block by its mean value. This process of reducing the number of pixels is called downsampling. The size of the image is thus diminished to a quarter of the original. The text segmentation algorithm extracts the text plane. The processing time and the size of the text plane are reduced because the size of the image is a quarter of that of the original. After the segmentation algorithm is applied, the full-size background and quarter-size foreground are compressed using the wavelet-based compression algorithm, and quarter-size mask using the JB2.

In the decompression phase, the quarter-size mask is enlarged by upsampling. The upsampling method expands each pixel in the mask into a 2×2 pixel. After upsampling, text looks thinner than the original one, so the characters are expanded

by one pixel around the boundaries of it.

3.4 Experimental results

The proposed algorithms were simulated in Windows 2000 (Pentium III 700, 128 MB RAM) using programs written in C++ language. A 24-bit true color image and 200dpi processing were used. Each pixel in a 24-bit true color is characterized by R, G and B values, and every value is represented by 8 bits. The compression algorithms are applied to complex compound documents, so test images include text that overlaps the background. Two compression algorithms, JPEG and DjVu, are selected for comparison with the proposed algorithms. Figure 19 shows the mask images and the background images obtained using proposed algorithm CSSP-I.

The test images are processed using CSSP-I and JPEG, as shown in Fig.20. The color of the image compressed by JPEG is very seriously lost; the block effect is very obvious and the text is blurred. The visual quality obtained using CSSP-I is better than that obtained using the JPEG algorithm.

Figure 21 displays images processed by DjVu. CSSP-I and DjVu are based on the MRC format so the test images are divided into three – the mask image, the background image and the reconstruction image, for comparison.

(1) Mask image: From Figs. 19 and 21, text extracted by CSSP-I is clearer than that extracted by DjVu. The text can be extracted from the complex background using segmentation algorithm of CSSP-I, so the mask image to which CSSP-I is applied can

be put into post-processing, such as by OCR, more precisely than the mask of DjVu. Accordingly, the segmentation algorithm can also be applied to an OCR system to recognize text on a complex background.

(2) Background image: Figure 19 displays background images obtained by CSSP-I, and Fig. 21 shows those obtained by DjVu. Although DjVu is especially for extracting sharp edges, some parts of the text are missing. Clearly, the background images obtained using CSSP-I are more precise than those obtained using DjVu.

(3) Reconstruction image: Figure 20 displays the reconstruction images obtained of CSSP-I, and Fig. 21 displays those reconstructed by of DjVu.

Figure 22 shows the reconstruction images and the mask images obtained by CSSP-II. The latter are a quarter of the size obtained using CSSP-I. The mask images obtained using CSSP-II are not directly downsampled from those obtained using CSSP-I, but they are extracted from downsampled document images. Therefore, the processing time and the amount of memory used are reduced.

Table 3 presents the compression ratio and PSNR of the proposed methods, JPEG and DjVu. CSSP-I and CSSP-II yield better quality images than JPEG. Furthermore, the compression ratio and PSNR of the CSSP-I and CSSP-II are higher than those of JPEG. The average PSNR of the proposed methods is close to the average PSNR of DjVu, but the visual quality obtained using the proposed method is better than that obtained using DjVu. The total compression ratio of CSSP-II is higher

than that of DjVu.

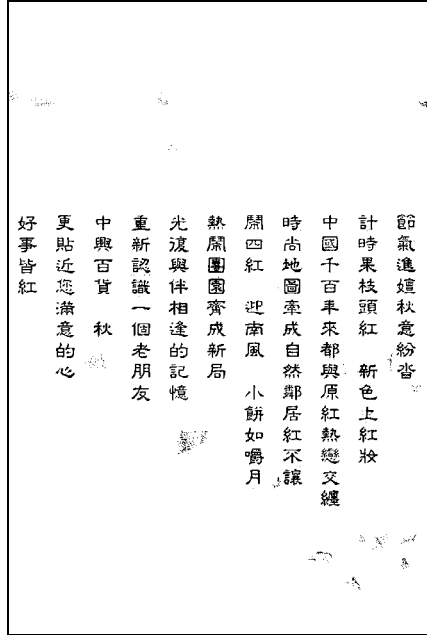
3.5 Concluding remarks

Document image segmentation has been studied for over ten years. Directly extracting text from a complex compound document is difficult because the text overlaps background. This chapter proposed a new segmentation method for separating text from compound document images with high text/background overlap. Based on the new segmentation method, two methods for compressing compound document images were presented. High-quality compound document images with both high compression ratio and a good presentation of text were thus obtained. The proposed compression algorithms were compared with JPEG and DjVu. The proposed methods perform much better.





Original image



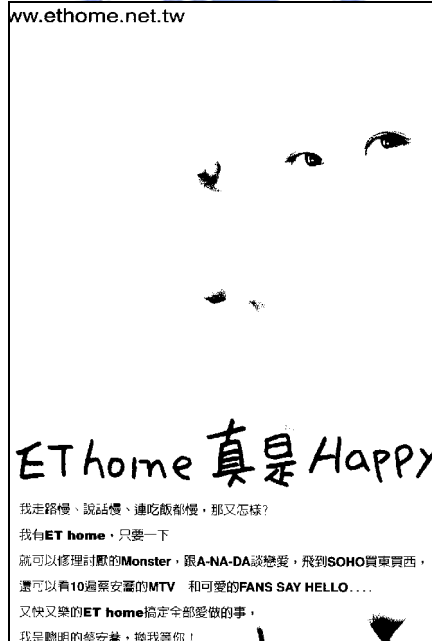
Mask image
(a) Test image A



Background image



Original image

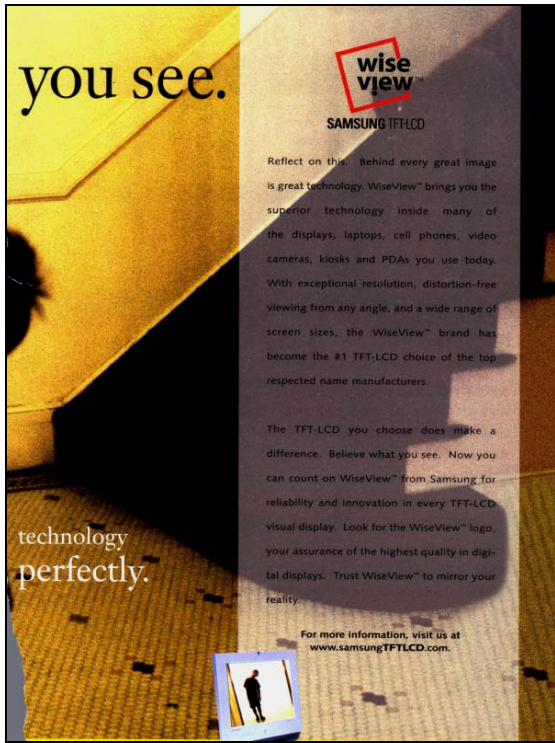


Mask image
(b) Test image B

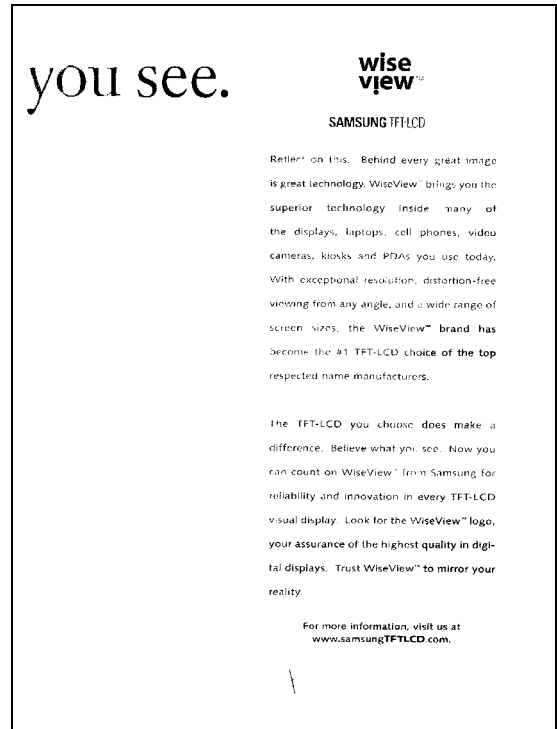


Background image

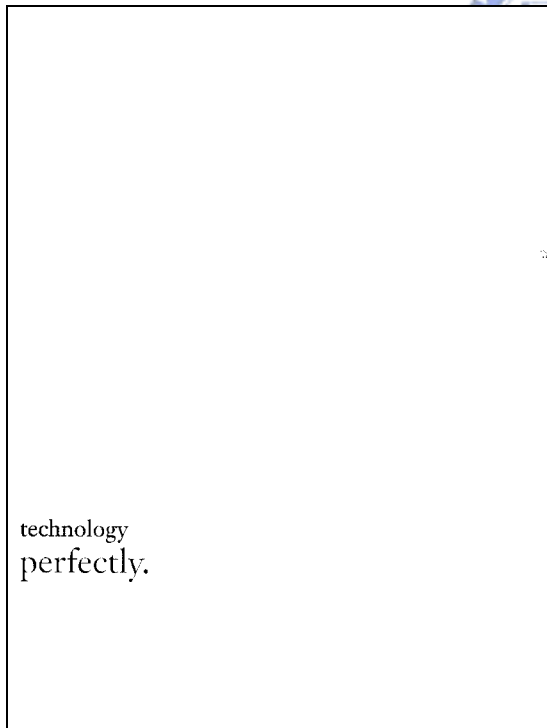
		
Original image	Mask image	Background image
(c) Test image C		
		
		
Original image	Mask image	Background image
(d) Test image D		



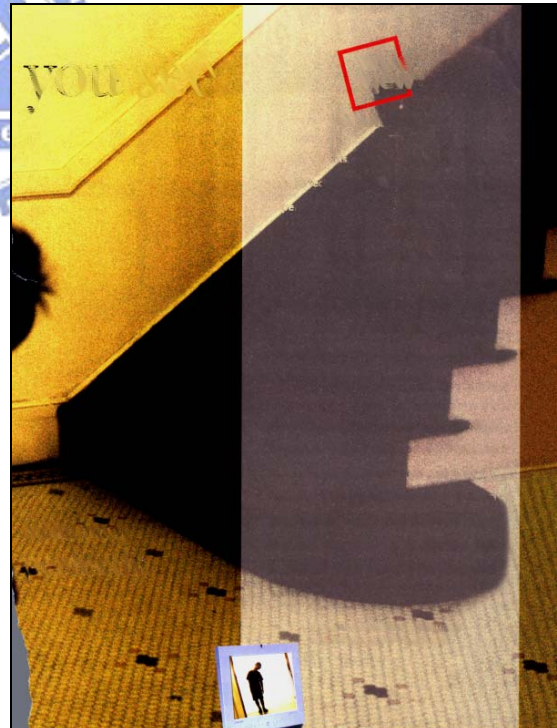
Original image



Mask image 1

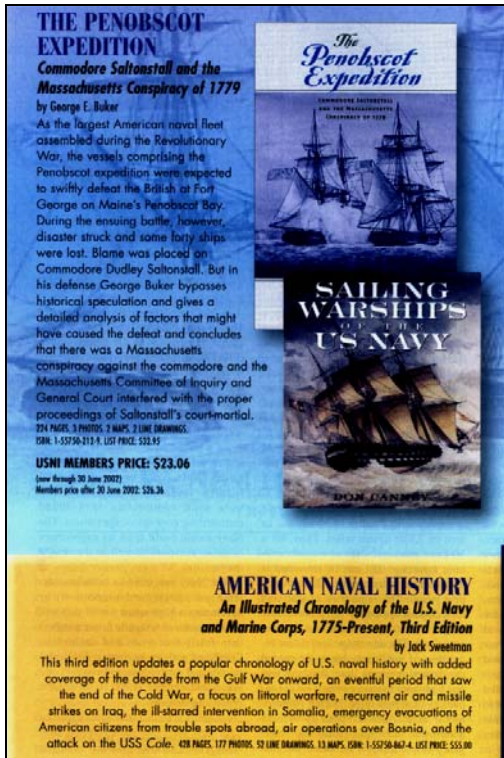


Mask image 2

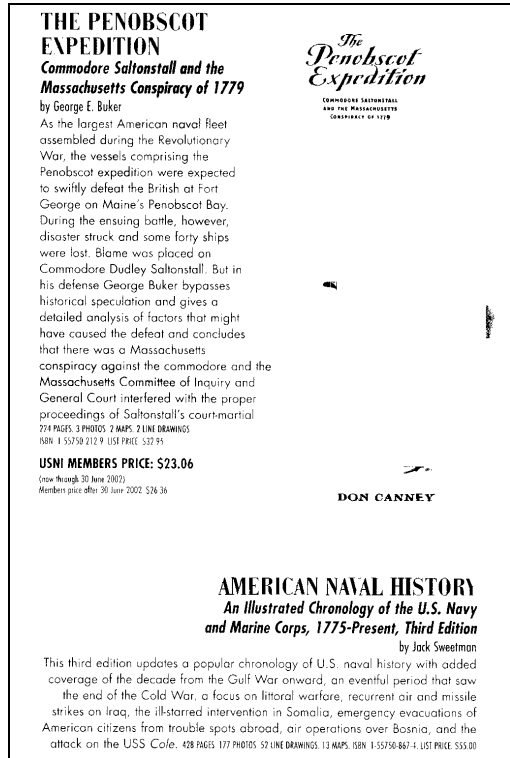


Background image

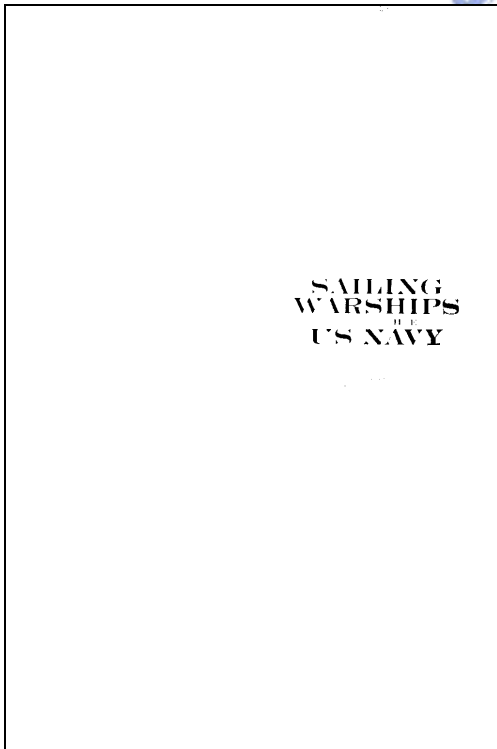
(e) Test image E



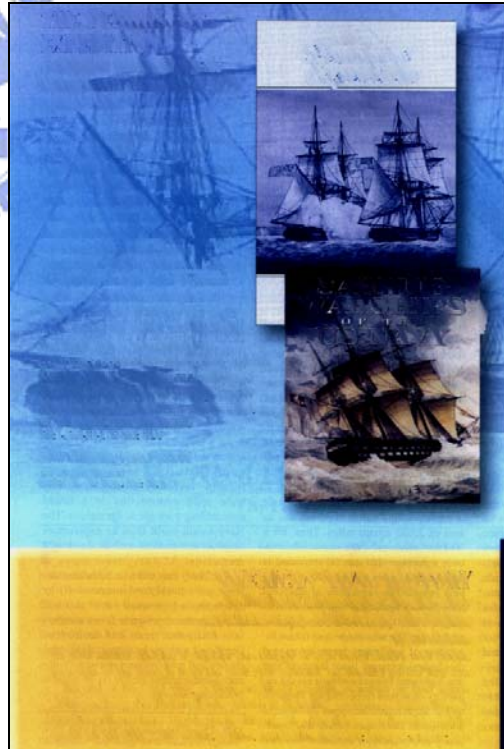
Original image



Mask image 1



Mask image 2



Background image

(f) Test image F

Fig.19 Segmentation images of proposed algorithm CSSP-I



CSSP-I

Mask 8,647 bytes
 Foreground 2,359 bytes
 Background 15,728 bytes
 Compression ratio=176.7



CSSP-I

Mask 7,083 bytes
 Foreground 2,359 bytes
 Background 15,728 bytes
 Compression ratio=187.5



CSSP-I

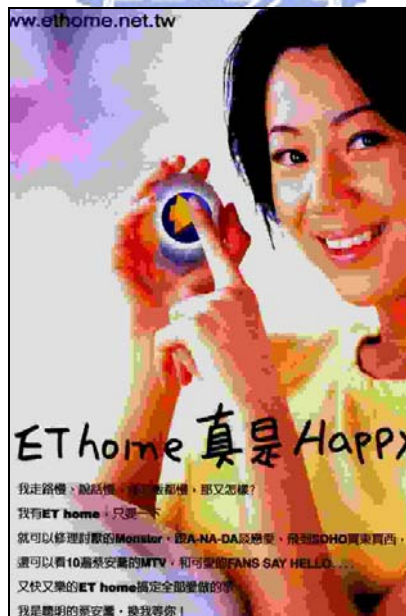
Mask 9,556 bytes
 Foreground 2,359 bytes
 Background 15,728 bytes
 Compression ratio=170.7



JPEG

Compression ratio=122.4

(a) Test image A



JPEG

Compression ratio=138.7

(b) Test image B



JPEG

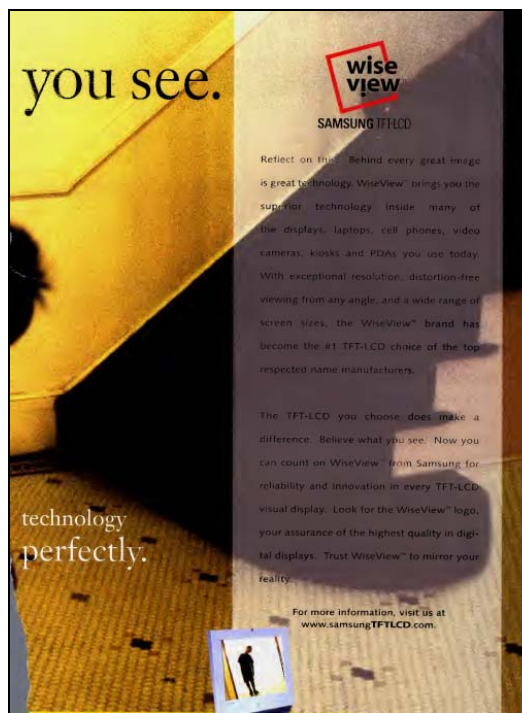
Compression ratio=123.9

(c) Test image C

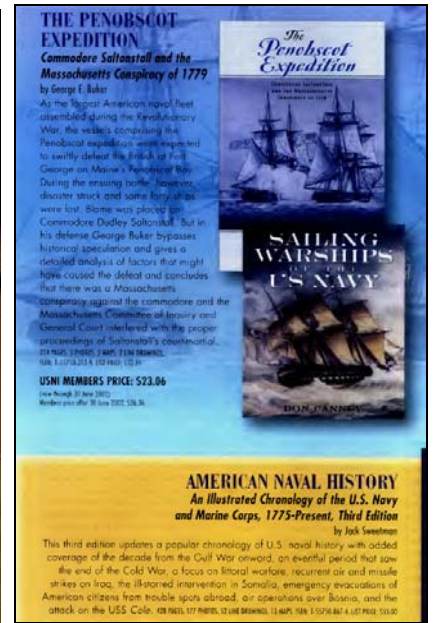


推出後廣受玩家的支持與肯定
 千禧年最值得珍藏的遊戲!!
 大宇的堅持...造就出RPG史上 新的里程碑
 融合異國風情如史詩般的傳奇故事
 氣勢磅礴的配樂音效
 中國式水墨畫風格的戰鬥場景.....
 讓你在不自覺中與遊戲裡的人物

CSSP-I
 Mask 6,995 bytes
 Foreground 2,359 bytes
 Background 15,728 bytes
 Compression ratio=188.1



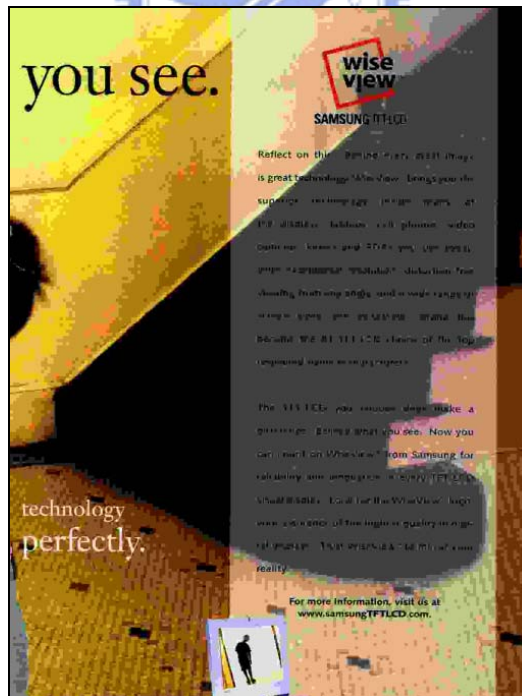
CSSP-I
 Mask 8,835+1,051bytes
 Foreground 3,612*2 bytes
 Background 24,084 bytes
 Compression ratio=175.4



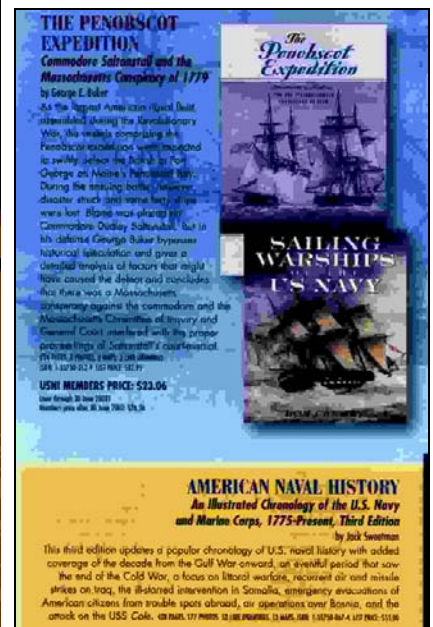
CSSP-I
 Mask 14,947+4,096 bytes
 Foreground 2,359*2 bytes
 Background 15,728 bytes
 Compression ratio=119.5



JPEG
 Compression ratio=120.5
 (d) Test image D



JPEG
 Compression ratio=160.4
 (e) Test image E

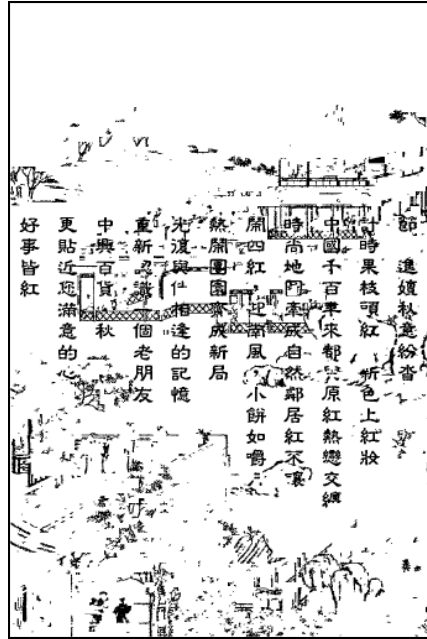


JPEG
 Compression ratio=117.4
 (f) Test image F

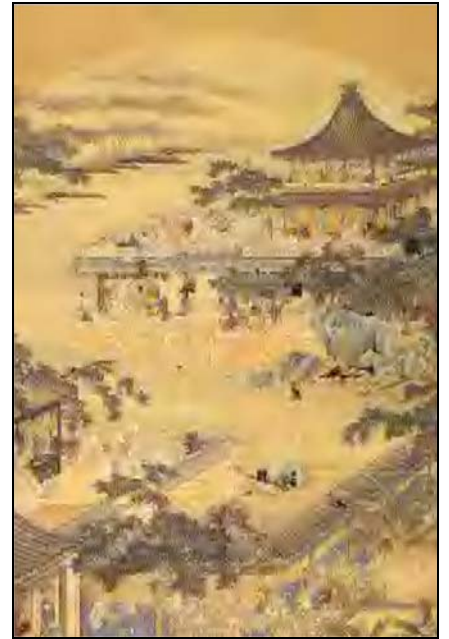
Fig.20 Compared with proposed algorithm CSSP-I & JPEG



Compression ratio=163



Mask plane

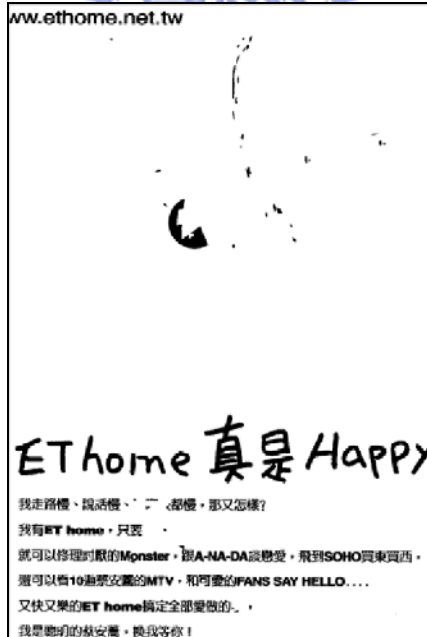


Background plane

(a) Test image A



Compression ratio=157.7



Mask plane

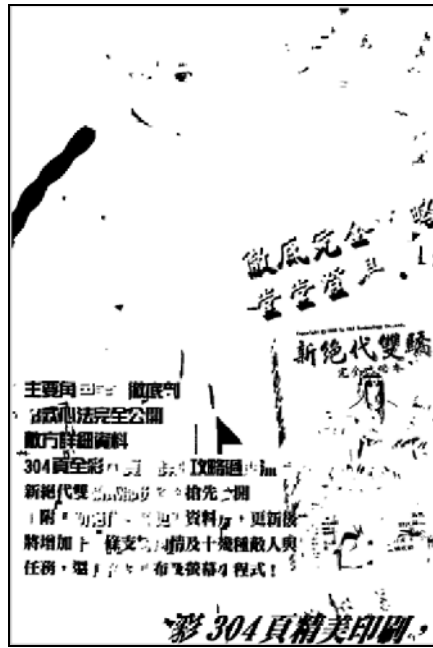


Background plane

(b) Test image B



Compression ratio=160.5



Mask plane



Background plane

(c) Test image C



Compression ratio=166.3



Mask plane

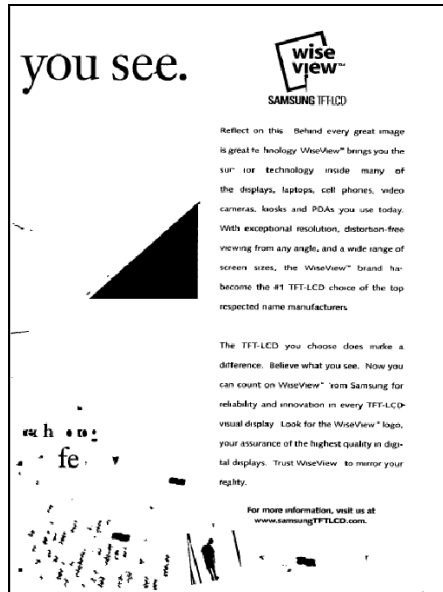


Background plane

(d) Test image D



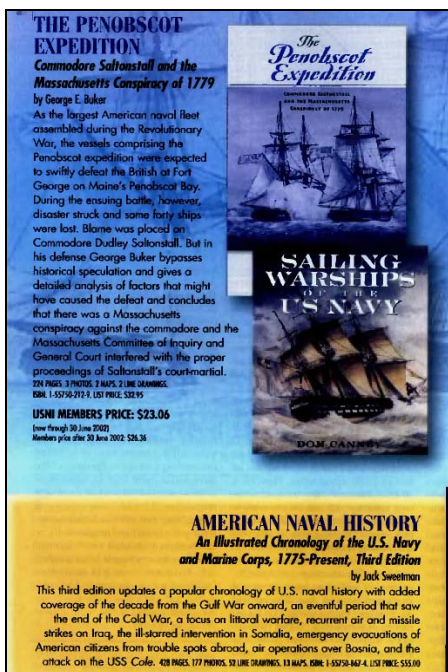
Compression ratio=166



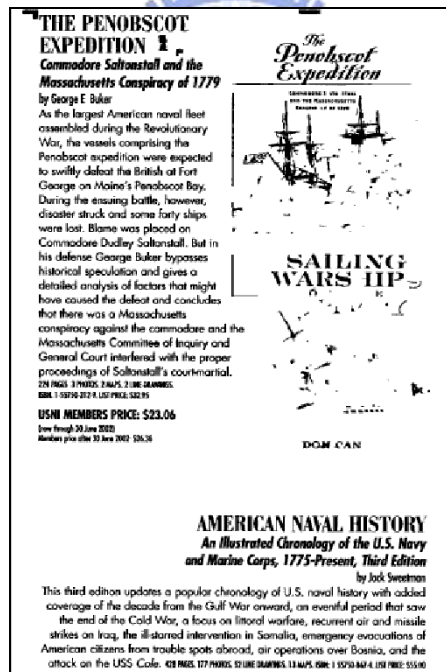
Mask plane
(e) Test image E



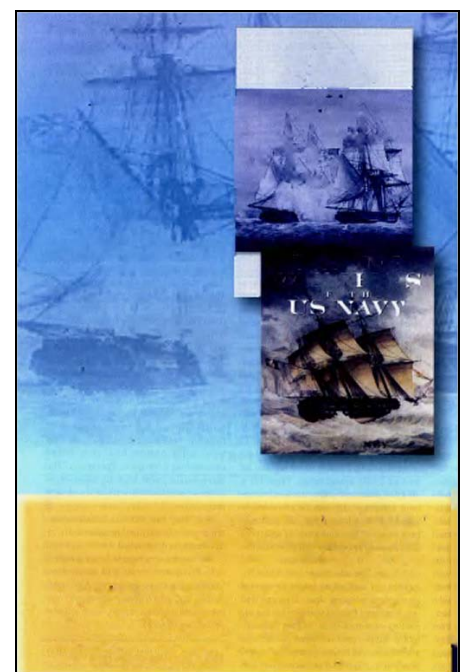
Background plane



Compression ratio=118.2



Mask plane
(f) Test image F



Background plane

Fig.21 Processed images by the DjVu



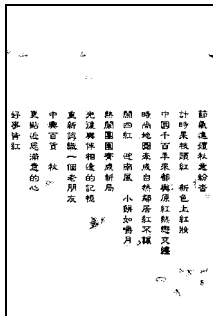
Reconstruction image



Reconstruction image



Reconstruction image



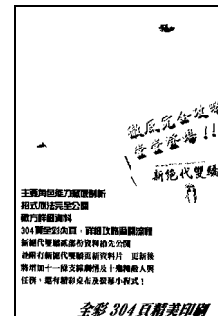
Mask plane

(a) Test image A
 Mask 4,362 bytes
 Foreground 590 bytes
 Background 15,728 bytes
 Compression ratio=228.2



Mask plane

(b) Test image B
 Mask 3,731 bytes
 Foreground 590 bytes
 Background 15,728 bytes
 Compression ratio=235.4

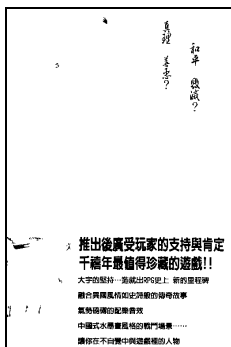


Mask plane

(c) Test image C
 Mask 4,936 bytes
 Foreground 590 bytes
 Background 15,728 bytes
 Compression ratio=222

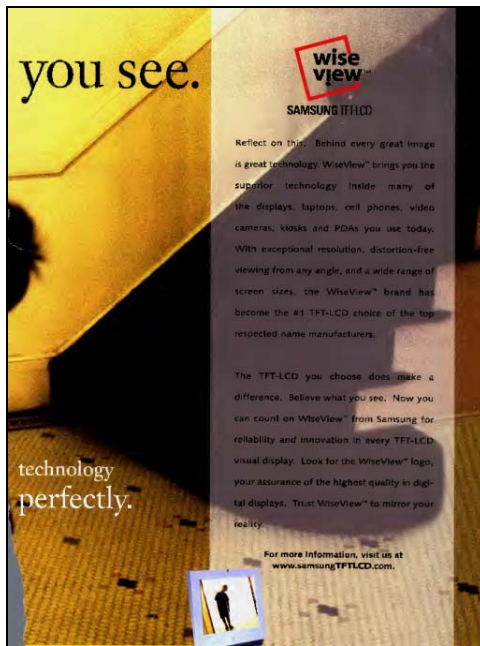


Reconstruction image

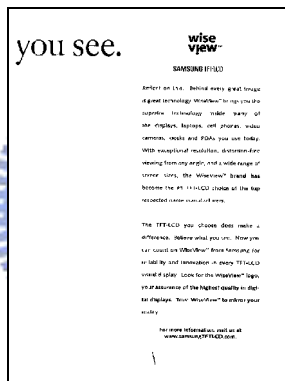


Mask plane

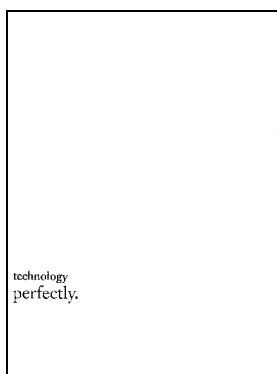
(d) Test image D
 Mask 3,572 bytes
 Foreground 590 bytes
 Background 15,728 bytes
 Compression ratio=237.2



Reconstruction image

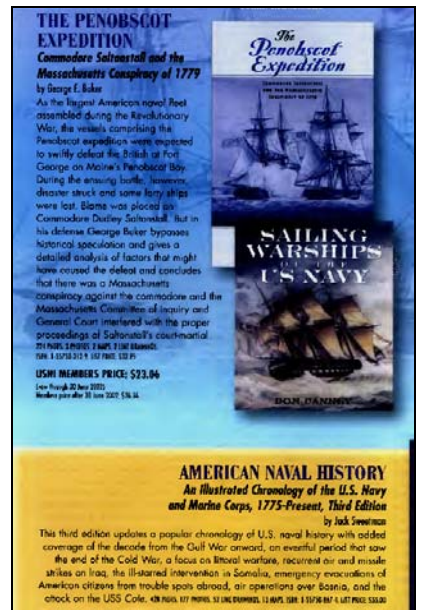


Mask plane 1

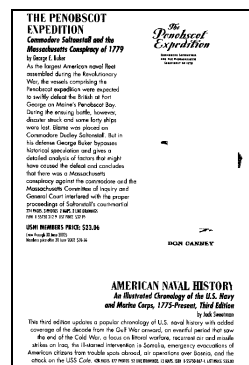


Mask plane 2

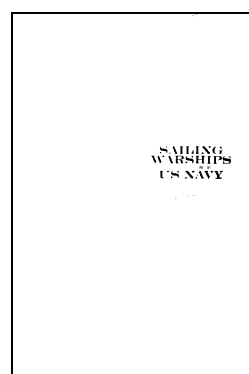
(e) Test image E
 Mask 4,616+ 671 bytes
 Foreground 3,612*2 bytes
 Background 24,084 bytes
 Compression ratio=197.5



Reconstruction image



Mask plane 1



Mask plane 2

(f) Test image F
 Mask 7,884+ 674 bytes
 Foreground 590*2 bytes
 Background 15,728 bytes
 Compression ratio=185.3

Fig.22 Processed images by the CSSP-II

Table 3 Comparison the compression ratio and PSNR for the proposed methods to JPEG & DjVu

Images		a	b	c	d	e	f	Average
Algorithm CSSP-I	Compression ratio	176.7	185.5	170.7	188.1	175.4	119.5	169.3
	PSNR	18.9	22.8	20.9	19.8	23.9	20.5	21.13
Algorithm CSSP-II	Compression ratio	228.2	235.4	222.2	237.2	197.5	185.3	217.6
	PSNR	18.8	22.7	20.9	20.3	23.7	20.3	21.1
JPEG	Compression ratio	122.4	138.7	123.9	120.5	160.4	117.4	130.6
	PSNR	17.6	21.5	20.0	19.8	21.3	19.5	19.95
DjVu	Compression ratio	163	157.7	160.5	166.3	166	118.2	155.3
	PSNR	18.7	23.6	20.6	20.2	23.6	20.3	21.17

CHAPTER 4

THE MULTI-LAYER SEGMENTATION METHOD FOR COMPLEX DOCUMENT IMAGES

Texts are frequently printed on complex backgrounds. Segmenting texts is an important topic in document analysis. Some methods of segmentation have been developed for texts with images. However, previous studies have not sufficiently addressed complex compound documents. This chapter proposes a text segmentation algorithm for various document images. The proposed segmentation algorithm incorporates a new multi-layer segmentation method to separate the text from various compound document images, regardless of whether the text and background overlap. This method solves various problems associated with the complexity of background images. Experimental results obtained using different document images scanned from book covers, advertisements, brochures, and magazines reveal that the proposed algorithm can successfully segment Chinese and English text strings from various backgrounds, regardless of whether the texts are over a simple, slowly varying or rapidly varying background texture.

4.1 Introduction

The complexity of background images is critical to the application of the text segmentation algorithm. Segmenting the texts from a complex compound document image is an important issue in document analysis. Document image segmentation, which separates the text from a monochromatic background, has been studied for over a decade. Some systems based on prior knowledge of some statistical characteristics of various blocks [11],[13], or texture analyses [27] have been successively developed. A text segmentation algorithm based on block-thresholding, which involves thresholds on rate-distortion has been proposed [26]. A system that focuses on the extraction and classification of bibliographical information from book covers has been developed [34]. Several other approaches use the features of wavelet coefficients to extract text [35]-[39].

All such systems focus on processing document images whose texts do not overlay a complexity background. These studies are effective in extracting characters from monochromatic backgrounds. However, they do not apply when backgrounds include sharply varying contours or overlap with texts. These background images include 1) monochromatic backgrounds with/without texts; 2) slowly varying backgrounds with/without texts; 3) highly varying background with/without texts and, 4) complex varying backgrounds with/without texts with different colors. Extracting

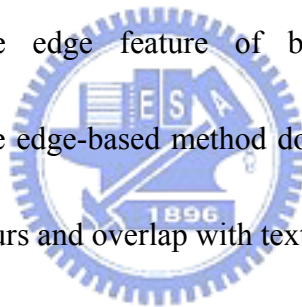
the texts is particularly difficult when the compound document image includes all of these backgrounds.

A text extraction algorithm was proposed to aim at WWW images [40]. The algorithm used the Euclidean minimum-spanning-tree (EMST) technique to cluster the R-G-B space of the input image into a number of color classes. For each color class, the bounding boxes of connected components were found by the connected-component labeling method and the shape and feature of the bounding boxes were used to classify the connected components as text-like or non-text-like. However, the global algorithm is not sufficiently to extract the texts in many document, advertisement, brochure, and magazine images. The texts in these A4-size images are widespread distributions. Because these images are captured by scanners, the pixel values of texts will be spread due to the optical property of scanners for the complex varying backgrounds overlap with texts with different colors. Hence, the texts in each color classes will be fragmented by the global algorithm.

A global segmentation method for color document was proposed [41], which uses spatial information of R-G-B color space to select the line segments by the author as initial clusters. Then, reduce the line segments that are close and use a predefined threshold to group the neighbor pixels of the remaining line segments. The method makes the assumption that for the documents under consideration the

background color for each frame is uniform over the whole frame. Hence, the method does not apply when images include rich and colorful backgrounds or little texts, because the line segments can not be selected correctly. Meanwhile, the texts in each cluster will be fragmented by the global algorithm.

Some edge-based methods were proposed to detect the texts from complex document images [42],[43]. These methods use Sobel or Canny operator to detect the edge features and calculates an edge-based feature to detect the texts. The edge-based methods can detect the edge feature and use the feature to extract the texts from document images. But, the edge feature of backgrounds will be detected simultaneously. Therefore, the edge-based method does not valid when backgrounds include sharply varying contours and overlap with texts.



A text detection method [44] used three second-order Gaussian derivative filters to calculate the edge-feature vector from three different image sizes, and the K-means algorithm (with $K=3$) was used to cluster the pixels based on the edge-feature vectors. One of the three clusters is labeled as text plane. Because the text plane contains many complex backgrounds and texture patterns, the refinement phase calculates the strokes, edge information, of the text plane and groups the strokes which have similar heights and are horizontally aligned into tight rectangular bounding boxes. Furthermore, the text plane clustered by the edge-feature vectors will be interfered by non-text edges

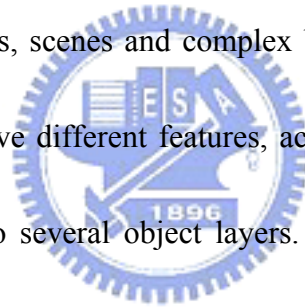
when the texts are connected with the complex texture background. Hence, the edge-based method is not useful when backgrounds include sharply varying contours and overlap with texts.

Recently, some text detection and tracking methods focus on digital video were proposed [45],[46]. The text detection methods use the wavelet transfer or the gradient image of R-G-B space to obtain the edge information of the images. The text extraction methods are edge-based and multi-scale methods. These methods are sufficient to detect and track texts from video frames. In document images, many texts are small and thin. The edge features of small texts and thin strokes will be diminished after the downsample process. Hence, the methods could be unsuitable to detect the texts from the document images. Furthermore, the edge-based text extraction method will also be interfered by the non-text edges when the texts are connected with the complex texture backgrounds which include sharply varying contours.

In the multi-layer segmentation method (MLSM) proposed by this chapter, it uses a block-based unsupervised clustering algorithm to cluster the pixels whose values are near. As we know, the disadvantage of the local method is that a lot of the structural information is lost in the process. Hence, a jigsaw-puzzle layer construction algorithm is presented to reconstruct the structural information based on different objects of the

processed images. Therefore, the MLSM can be used to solve various problems associated with the different document images scanned from book covers, advertisements, brochures, and magazines, regardless of whether the texts are over a simple, slowly varying or rapidly varying background texture. The MLSM focuses on the document images and can solve the disadvantages of the edge-based and downsample methods.

This chapter presents a good method to extract texts from different compound document images. The compound document image includes several objects, including different colored texts, figures, scenes and complex backgrounds. Such objects may overlap each others. They have different features, according to which the document image can be partitioned into several object layers. The MLSM can separate texts from 8-bit grayscale or 24-bit true-color document images, regardless of whether the texts overlay a simple, slowly or highly varying background.

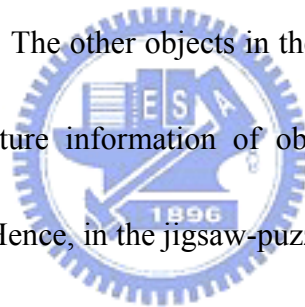


4.2 Multi-layer segmentation method

The multi-layer segmentation method uses two stages to segment different objects from various compound document images. The different objects in a document image are segmented into various object layers. The first stage of the method is the block-based clustering algorithm, which clusters distinct objects embedded in sub-block images into different "layered-sub-blocks" (*LSBs*). The second

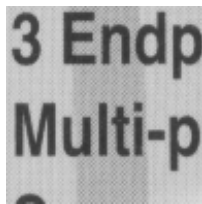
stage applies the jigsaw-puzzle layer construction algorithm to assemble the adjacent *LSBs* of a same object into an object layer.

In the block-based clustering algorithm, the document image is partitioned into many sub-block images, and each sub-block image is classified into different *LSBs*. Each *LSB* contains one object with similar value. The texts/objects with the same color are embedded in one of the *LSBs* of a sub-block image. Although the block-based clustering algorithm can extract the texts from different backgrounds, the texts in the same paragraph will be divided into many $K \times L$ blocks which are the *LSBs* of different sub-block images. The other objects in the document image also have the same problem that the structure information of objects is destroyed. This is the drawback of a local method. Hence, in the jigsaw-puzzle layer construction algorithm, some statistical and spatial features of adjacent *LSBs* are introduced to assemble all *LSBs* of the same text paragraph or object, as jigsaw puzzle. The structure information of different objects can be recovered by the jigsaw-puzzle layer construction algorithm. The construction algorithm uses the *LSBs* of two adjacent sub-block images to find the best match. The match process is described in the section of the jigsaw-puzzle layer construction algorithm. Figure 23 shows the results of two adjacent sub-block images after the block-based clustering algorithm.

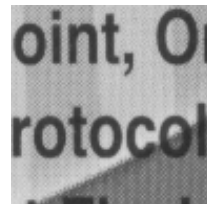




(a) A partial image of the Test image 6 in Fig.32



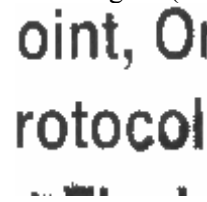
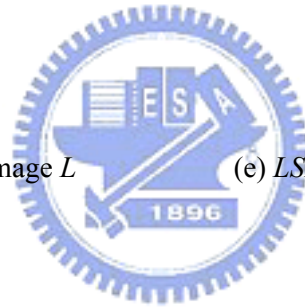
(b) Sub-block image L (size=96x96)



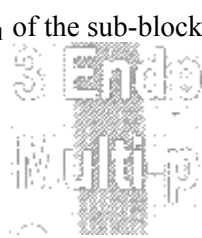
(c) Sub-block image R (size=96x96)



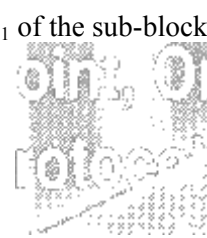
(d) LSB_1 of the sub-block image L



(e) LSB_1 of the sub-block image R



(f) LSB_2 of the sub-block image L



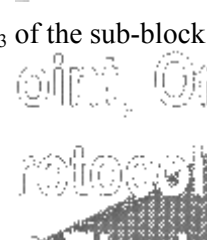
(g) LSB_2 of the sub-block image R



(h) LSB_3 of the sub-block image L



(i) LSB_3 of the sub-block image R

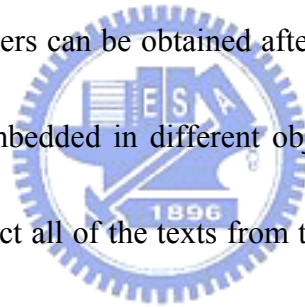


(j) LSB_4 of the sub-block image R

Fig.23 An example of the results after the block-based clustering algorithm.

Figure 23(a) is a partial image of the test image 6 (in Fig.32). Figure 23(b) and (c) are two sub-block images in Fig.23(a). Figure 23(d), (f), and (h) are the *LSBs* of Fig.(b). Figure 23(e), (g), (i), and (j) are the *LSBs* of Fig.23(c). Obviously, the sub-block image of Fig.23(b)/(c) contains three/four different objects. After the block-based clustering algorithm, it is segmented to three/four *LSBs* automatically.

The MLSM uses the block-based clustering algorithm and the jigsaw-puzzle layer construction algorithm to segment different objects in a document image to distinct object layers. For example, the Fig.23(d) & (e) will belong to the same object layer. The different object layers can be obtained after the MLSM is applied, and the texts in various colors are embedded in different object layers. Hence, the text line extraction algorithm can collect all of the texts from these object layers, regardless of whether they overlap a monochromatic background or a complex one. The number of object layers depends on the maximum number of the *LSBs* of the sub-block image.



The “joint division factor”(*JDF*) measures the separability between two adjacent clusters in the block-based clustering algorithm. When the number of the clusters is more than two, the average *JDF* is used to measure the separability of the clusters. The pool of layered sub-blocks, denoted by *Pool*, collects all undetermined *LSBs*. A cluster of the determined *LSBs* constitutes an object layer for further analysis.

4.2.1 Block-based clustering algorithm

A color transformation is applied to transfer a color document image to the YUV space and the Y-plane (grayscale image) of the original document image is applied for performing the segmentation method. There are two reasons for using the Y-plane to cluster the different objects. First, the processing speed can be fast, because the Y-plane is one-third of the color image (RGB-plane). Second, the method can be suitable for color and monochromatic images, because many document images are stored monochromatically.

After a color document image is transformed into a monochromatic one, the textures of the original color image remain present in the converted grayscale image. The difference between the gray value of the texts and that of the overlapping background image may still be small. Therefore, a block-based clustering algorithm is proposed to cluster the grayscale sub-block images. The clustering analysis is an unsupervised method for separating text from a sub-block image. The sub-block image is classified into different clusters. Each cluster contains one object with similar value.

The proposed block-based clustering algorithm is shown below.

Step 1. Partition the $M \times N$ grayscale image A into q sub-block images x_n . Each sub-block x_n is of size $K \times L$. When the full image A cannot be exactly divided into

sub-blocks, the sizes of the bounding blocks, located on the right column and the bottom row, are smaller than the sizes of the sub-block images.

Step 2. Calculate the mean value and the standard deviation of each $K \times L$ sub-block image. For the n^{th} sub-block image x_n , the mean and standard deviation are m_n and σ_n .

Step 3. If $\sigma_n < TH_\sigma$, terminate the clustering process. Else, if $\sigma_n > TH_\sigma$, then $x_n(i, j)$ is split according to mean and standard deviation of the processed sub-block image. Define two centers, C'_{n1} and C'_{n2} , by

$$\begin{aligned} C'_{n1} &= m_n + 0.5 \times \sigma_n \quad \text{and} \\ C'_{n2} &= m_n - 0.5 \times \sigma_n. \end{aligned} \quad (4-1)$$

Step 4. Calculate the Euclidean distance from each pixel $x_n(i, j)$ to C'_{n1} and C'_{n2} , using the following equalities, respectively.

$$\begin{aligned} D'_{ij,1} &= |x_n(i, j) - C'_{n1}| \quad \text{and} \\ D'_{ij,2} &= |x_n(i, j) - C'_{n2}| \end{aligned} \quad (4-2)$$

Then, $x_n(i, j)$ is partitioned into two clusters $\psi_k (k = 1, 2)$ as,

$$\begin{aligned} \psi_1 &: \{x_n(i, j) | D'_{ij,1} \leq D'_{ij,2}\}, \quad \text{and} \\ \psi_2 &: \{x_n(i, j) | D'_{ij,1} > D'_{ij,2}\}. \end{aligned} \quad (4-3)$$

Step 5. As stated in [47], let $\sigma_{B1,2}^2$ and $\sigma_{T1,2}^2$ be the within-class variance and the total variance, respectively. The joint division factor (*JDF*) between two adjacent

clusters is defined as,

$$JDF_{1,2} = \frac{\sigma_{B1,2}^2}{\sigma_{T1,2}^2}, \quad (4-4)$$

$$\sigma_{T1,2}^2 = \sum_{r \in (\psi_1 \cup \psi_2)} (r - \mu_{T1,2})^2 P_r, \quad \mu_{T1,2} = \frac{\sum_{r \in (\psi_1 \cup \psi_2)} r P_r}{\sum_{r \in (\psi_1 \cup \psi_2)} P_r}, \quad (4-5)$$

$$\sigma_{B1,2}^2 = \omega_1 \omega_2 (\mu_1 \mu_2)^2, \quad \omega_1 = \frac{\sum_{r \in \psi_1} P_r}{\sum_{r \in (\psi_1 \cup \psi_2)} P_r}, \quad \omega_2 = 1 - \omega_1, \quad (4-6)$$

$$\mu_2 = \frac{\mu_{T1,2} - \mu_1}{\omega_2}, \quad \mu_1 = \frac{\mu_t}{\omega_1}, \quad \mu_t = \frac{\sum_{r \in \psi_1} r P_r}{\sum_{r \in \psi_1} P_r}, \quad P_r = \frac{n_r}{n}, \quad (4-7)$$

where n_r is the number of pixels with gray-level r and n is the total number of pixels in ψ_1 and ψ_2 .

Step 6. Calculate the mean m_{nk} and standard deviation σ_{nk} of the two clusters ψ_k ($k=1,2$).

If $JDF_{1,2} < TH_{JDF}$ and $\sigma_{n_max} > TH_\sigma$, where σ_{n_max} is the maximum of the σ_{nk} ($k=1,2$), then split the cluster of σ_{n_max} .

Else, terminate the clustering process.

Step 7. Step 6 yields three clustering centers C_{nk} ($k=1,2,3$). x_n is partitioned into three clusters ψ_k ($k=1,2,3$).

Step 8. Calculate the mean value, m_{nk} , the standard deviation, σ_{nk} , of each cluster and the joint division factor, $JDF_{1,2}$, $JDF_{2,3}$, ..., $JDF_{k-1,k}$ among the clusters.

($k=1,2,\dots,x$ and $x>2$)

If $\sqrt{\frac{JDF_{1,2}^2 + JDF_{2,3}^2 + \dots + JDF_{k-1,k}^2}{k-1}} < TH_{JDF}$ and $\sigma_{n_max} > TH_\sigma$, then split the cluster of

maximum σ_{nk} into two clusters. Next, repeat Step 8.

Else, terminate the clustering process.

Repeat Steps 2~8 until all of the sub-block images x_n have been processed. This study employs $TH_{JDF}=0.9$, $TH_{\sigma}=14$ and $K=L=96$.

As for the values of K and L , if the size of the sub-block images grows smaller, there will be a more detailed segmentation result for each sub-block image. The small objects can be segmented more clearly, but it will cost more computation to get the final result by the following procedure, Jigsaw-Puzzle Layer Construction Algorithm.

Therefore, we have to select the maximum values of the parameters K and L that

objects in document images can be segmented clearly. The maximum values of the parameters K and L depend on the size of the smallest texts in the document images.

In this work, the $K=L=96$ are determined from the experiments using numerous image samples, such that all existing objects in the document images are almost completely separated. Consequently, all objects are segmented into individual clusters in the order of the darkest to the brightest.

After all sub-block images are clustered, several clusters are decomposed from each sub-block image. Each cluster contains a partial image of its original sub-block image. In each cluster, it could be contained more than one connected regions. For example, two letters, i and j , and many Chinese's characters contain more than one connected region. Therefore, if a sub-block image is divided into k clusters, it will

probably contain more than k connected regions. A specific analytical method - Jigsaw-Puzzle Layer Construction Algorithm - is applied to them. Furthermore, observation of a sub-block image and its resultant clusters generated from a block-based clustering algorithm reveals that the clusters look like “sub-layers” of the original sub-block image. Therefore, a cluster is called a “Layered-Sub-Block”, or *LSB*. All *LSBs* generated from the previous clustering process are collected into a “*Pool*”, to which the jigsaw-puzzle layer construction algorithm is applied.

4.2.2 Jigsaw-puzzle layer construction algorithm

A sub-block image may be composed by one or more object images with various intensity features. Those object images may be parts of a larger object, one or several character patterns with various intensities, or one piece of background texture components. The block-based clustering algorithm decomposes the sub-block image into different layered sub-block images, *LSBs*, in the order of the darkest to brightest corresponding to the original sub-block image. In the jigsaw-puzzle layer construction algorithm, some statistical and spatial features of adjacent *LSBs* are introduced to assemble all *LSBs* of the same text paragraph or object. This section introduces an algorithm for constructing the object layers from the *LSBs* generated from the block-based clustering algorithm introduced in the preceding section. Before the explanation of the algorithm, we describe some definitions in the

algorithm.

We define the *4-adjacent* that each *LSB* has four sides between the adjacent sub-block images that border on the top, bottom, left, or right side of the *LSB*. Each side of the *LSB* adjoins several adjacent *LSBs* derived from the adjacent sub-block image and matches only one adjacent *LSB*. An object layer is assembled by the *LSBs* which match their adjacent *LSBs*, and all the *LSBs* of the object image are recorded by a finite chain. Since text strings are mostly printed in horizontally or vertically in documents, the text strings of document images have a continuous relationship in horizontal or vertical. Hence, it is appropriate that the *4-adjacent* property is used to determine the connectedness of the *LSBs*. The continuity of the adjacent *LSBs* is used to match the *LSBs* with *4-adjacent*. The pixels of each *LSB* can be represented as a specified subset of all pixels in the corresponding sub-block image. A *LSB* may comprise several connected regions - these pixels of the connected regions are said the valid pixels and the rest pixels of the *LSB* are said the invalid pixels.

The parameter $LSB(i, j, k)$ is defined the k -th *LSB* decomposed from the sub-block image $x_n(i, j)$. If the $LSB(i, j, k)$ is matched to the object layer L_q , then it is denoted as $LSB_q(i, j, k)$, where the subscript q denote the q -th object layer. We denote the valid pixel value located on (x, y) in the $LSB(i, j, k)$ as $Pix(LSB, x, y)$, $x=0\sim(K-1)$ and $y=0\sim(L-1)$. Two measurements of the continuity among two *LSBs* are defined.

First, the mean distortion of all touched valid pixels at the boundary between two adjacent *LSBs*, called side-match distortion, is represented as the D_{SM} . For instance, there are two horizontally adjacent *LSBs* which have the dimension $K \times L$, and we denote the left one by the LSB_l and the right one by the LSB_r . Their pixel values on the horizontal touching boundary can be described by $Pix(LSB_l, K-1, y)$ and $Pix(LSB_r, 0, y)$, $y=0 \sim (L-1)$. Note that only the valid pixels are taken into account, and the values of the boundary pixels are taken into account for the D_{SM} . The horizontal touching boundaries between the adjacent *LSBs* form a vertical edge. The valid pixels that are symmetrically located on both sides of the vertical edge are considered as the valid side connection. The pair number of the valid pixels in the valid side connection is a factor that reflects the connectedness of the two adjacent *LSBs*, and is denoted by $N_{vs}(LSB_l, LSB_r)$. Hence the D_{SM} of two horizontally adjacent *LSBs* is computed as

$$D_{SM}(LSB_l, LSB_r) = \frac{\sum_{y=0}^{L-1} |Pix(LSB_l, K-1, y) - Pix(LSB_r, 0, y)|}{N_{vs}(LSB_l, LSB_r)}. \quad (4-8)$$

The D_{SM} means the average difference between the valid pixels in the valid side connection. If the value of the D_{SM} is small, the two adjacent *LSBs* will be the same object layer. The range of the D_{SM} value is 0~255.

Second, the difference between the average pixel values of two *LSBs*, called the *inter-LSB distortion*, is defined as the D_{LM} . Similarly, only the valid pixels of the

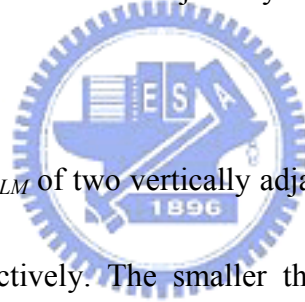
$LSBs$ are taken into account for the D_{LM} , and we denote the number of valid pixels of a specific LSB by the $N_{vp}(LSB)$. The D_{LM} is computed as

$$D_{LM}(LSB_l, LSB_r) = |m(LSB_l) - m(LSB_r)|, \quad (4-9)$$

where $m(LSB)$ denote the average of all valid pixels belonging to this LSB , and is computed as

$$m(LSB) = \frac{\sum_{x=0}^{K-1} \sum_{y=0}^{L-1} Pix(LSB, x, y)}{N_{vp}(LSB)}. \quad (4-10)$$

The D_{LM} means the average difference between two $LSBs$. If the value of the D_{LM} is small, the two $LSBs$ will be the same object layer. The range of the D_{LM} value is 0~255.



Similarly, the D_{SM} and D_{LM} of two vertically adjacent $LSBs$ can be deduced from Eq.(4-8) and Eq.(4-9), respectively. The smaller the value of the D_{SM} or D_{LM} is computed, the stronger the continuity or similarity of the adjacent $LSBs$ is presented.

According to the definitions described above, the match grade is defined as

$$\text{match grade} = \max(D_{SM}, D_{LM}), \quad (4-11)$$

where the D_{SM} and D_{LM} are calculated from the unclassified $LSB(i, j, k)$ and the representative $LSB_q(i', j', k')$. The D_{SM} of the Eq.(4-8) can be treated as the local average difference located on the both adjacent sides of the two $LSBs$ and the D_{LM} can be treated as the global average difference of the two $LSBs$. Therefore, the maximum value of the D_{SM} and D_{LM} is defined as the match grade. The best match of two $LSBs$

can be selected by the minimal value of the match grade.

The jigsaw-puzzle layer construction algorithm is constructed by two procedures, the decision procedure for constructing of a new object layer and the matching procedure. The proposed algorithm is given as the pseudo-code as follow:

Algorithm : Jigsaw-puzzle layer construction

Input : The *Pool* (The unclassified $LSB(i, j, k)$)

Output : The object layer planes

Initiation : $N \leftarrow 0$

$L_N \leftarrow LSB(0,0,0)$ (set up and initialize the first object layer.)

$N \leftarrow 1$ (N is current amount of existing object layers)

Begin

while the *Pool* is not empty do

{ DP()
MP()
}

End

DP() / The decision procedure/

{ for each unclassified $LSB(i, j, k)$ in the *Pool* do

{ Find the *SID*, *LID* and *SLD* of the unclassified *LSB*

}
if the smallest $SID(LSB(i, j, k)) \leq Th_{SI}$

{ $LSB[s], s \leq 5 \leftarrow$ The unclassified *LSBs* with the smallest *SID*

$LSB_{min_SLD} \leftarrow$ The unclassified *LSB* with smallest *SLD* computed from the $LSB[s]$

Classify the LSB_{min_SLD} to its corresponding object layer L_{SI}

Remove the LSB_{min_SLD} from the pool

}
else

{ $LSB_{max_LID} \leftarrow$ The *LSB* which has the maximum *LID*

$L_N \leftarrow LSB_{max_LID}$

$N \leftarrow N+1$

Remove the LSB_{max_LID} from the pool

}
}



```

MP() /The matching procedure/
{
  for each unclassified  $LSB(i, j, k)$  in the Pool do
  {
    for each existing object layers  $L_N$  do
    {
      for each 4-adjacent neighboring  $LSB_N$ , all the  $LSB_N \in L_N$ ,
      of the unclassified  $LSB(i, j, k)$  do
      {
        if the  $LSB_N$  satisfy the pre-match condition
        {
          Mark the  $LSB_N$  as the representative  $LSB$  of the object layer
        }
      }
    }

    if there are one more representative  $LSBs$  in this process
    {
       $LSB_q' \leftarrow$  the representative  $LSB$  with the minimal match grade
      Found_flag  $\leftarrow$  1
    }
    else if there is one representative  $LSB$  in this process
    {
       $LSB_q' \leftarrow$  the representative  $LSB$ 
      Representative_flag  $\leftarrow$  1
    }
    else
    {
      Representative_flag  $\leftarrow$  0
    }
    if Representative_flag = 1
    {
      candidate_insert( $LSB_q'$ )
    }
  }
  candidate_decide()  $\rightarrow L_w$ 
  {
     $L_w \leftarrow LSB(i, j, k)$ 
  }
}
}

```

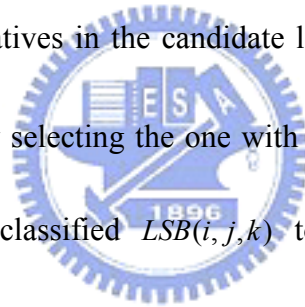
The proposed algorithm begins by analyzing the initially unclassified $LSBs$ in the *Pool*. The *Pool* will be analyzed several times until each of all unclassified $LSBs$ has been classified as a certain object layer. Before starting a new iteration of analyzing

the unclassified *LSBs* in the *Pool*, the algorithm will perform a procedure, called “decision procedure for constructing of a new object layer”, which will be described later, to determine a chosen seeded *LSB* whether to set up a new object layer or to belong to an existing object layer which is similar to the chosen seeded *LSB*. Then, the “matching procedure” will find the matched object layers of the unclassified *LSBs*. Once an unclassified *LSB* in the *Pool* has been classified to an object layer, then it will be removed from the *Pool*.

In the first time to analyze the *Pool*, a new object layer should be set up by the first unclassified $LSB(0,0,0)$, because there is no existing object layers initially. So, the $LSB(0,0,0)$ becomes a new object layer L_0 and is denoted as $LSB_0(0,0,0)$ in the decision procedure for constructing of a new object layer. In the matching procedure, we then scan each of the rest unclassified *LSBs* in the *Pool*. Whenever an unclassified $LSB(i, j, k)$ which is 4-adjacent to one or more existing object layers is detected, then we check the pre-match condition to find the reasonable object layers for the unclassified $LSB(i, j, k)$.

The pre-match condition is defined as $D_{LM}(LSB(i, j, k), LSB_q(i', j', k')) \leq Th_{LM}$, where $Th_{LM} = 14$ is a predefined threshold of gray level distance. When the condition is satisfied, the object layer L_q becomes a candidate for the unclassified $LSB(i, j, k)$ and the representative $LSB_q(i', j', k')$ of

the L_q will participate in the process of the match grade. The purpose of the pre-match is to filter out the unreasonable object layers in order to save the computation power. All representative $LSBs$ of the reasonable object layers will be found and inserted into the candidate list. Note that if there are more than one LSB of the same object layer L_q that are *4-adjacent* to the unclassified $LSB(i, j, k)$ and satisfy the pre-match condition, then we will choose the one with the minimal match grade as the representative $LSB_q(i', j', k')$ of the object layer L_q . After all representatives of the object layers are obtained, we calculate and compare the match grades between the unclassified $LSB(i, j, k)$ and all representatives in the candidate list, and then determine the best match representative LSB_w by selecting the one with the minimal value of the match grade, and thus classify the unclassified $LSB(i, j, k)$ to the L_w .



Now we return to the matching procedure, after the L_0 has been set up, there exists one object layer in this iteration. The $LSB(1,0,0)$ is currently analyzed, assume the $LSB_0(0,0,0)$ is *4-adjacent* and satisfies the pre-match condition with $LSB(1,0,0)$. Since there is only the $LSB_0(0,0,0)$ in the L_0 , so the $LSB_0(0,0,0)$ is selected as the representative LSB of the L_0 and inserted into the candidate list. Since no other object layers existing in this time, the L_0 is directly determined as the best match object layer, and thus $LSB(1,0,0)$ is classified to the L_0 and removed from the *Pool*. Then, repeat the two procedures until all unclassified $LSBs$ in the *Pool* have been analyzed once in

this iteration. The detail descriptions of the two procedures are explained below.

A. The decision procedure for constructing of a new object layer

The procedure determines a chosen *LSB*: 1) to set up and initialize a new object layer, or 2) to classify it into an existing object layer which is most similar to the chosen *LSB*. The decision procedure is performed to achieve an optimum decision to construct or extend an object layer. The decision is determined according to the analysis of following features and is depicted in Fig.24.

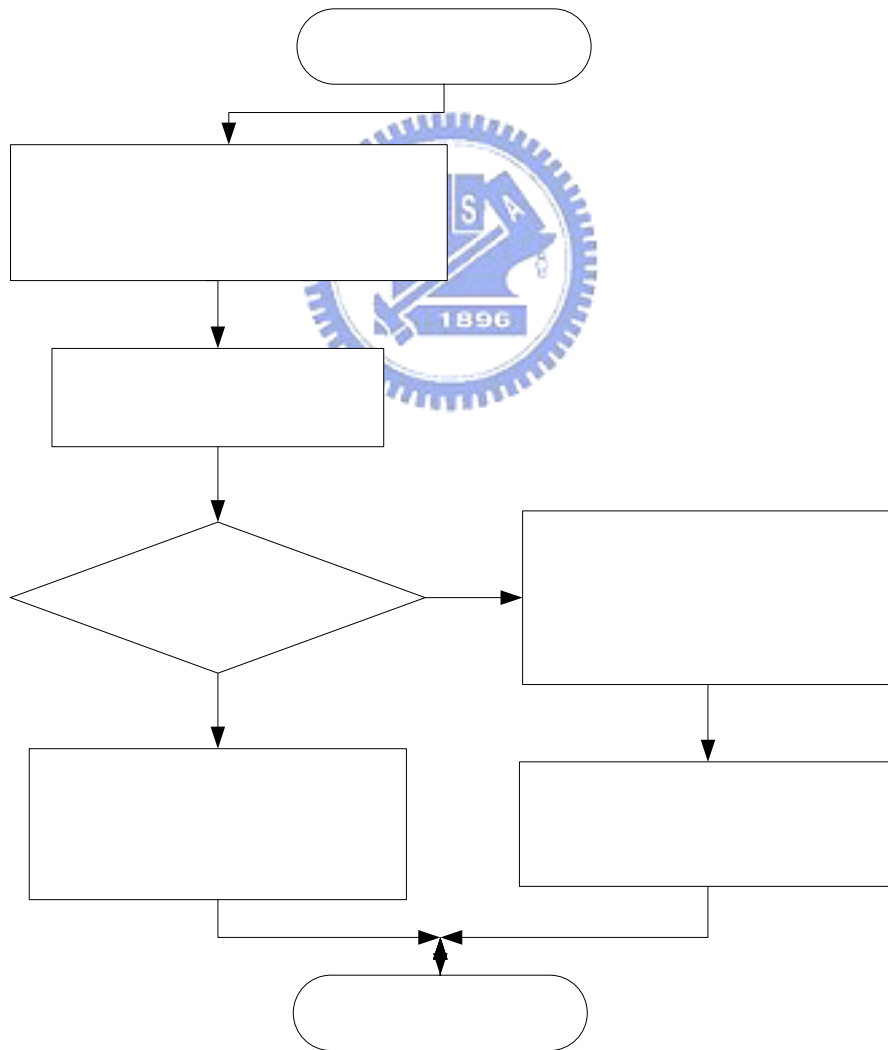


Fig.24 Flowchart of the decision procedure to construct or extend an object layer

Several definitions and measures must be stated before the details of this procedure are described. The minimum gray intensity distance between one unclassified $LSB(i, j, k)$ and the object layer L_p , denoted as $ID(LSB(i, j, k), L_p)$, is determined by the minimum intensity difference between the unclassified $LSB(i, j, k)$ and all $LSBs$ of the L_p , and is computed as

$$ID(LSB(i, j, k), L_p) = \min_{\forall LSB_p \in L_p} D_{LM}(LSB(i, j, k), LSB_p(i', j', k')), \quad (4-12)$$

where D_{LM} is defined in Eq.(4-9).

The Euclidean location distance between one unclassified $LSB(i, j, k)$ and the object layer L_p , denoted as $LD(LSB(i, j, k), L_p)$, is computed by the Euclidean distance between the unclassified $LSB(i, j, k)$ and the $LSB_p(i', j', k')$ of the L_p , and is computed as

$$LD(LSB(i, j, k), L_p) = \sqrt{(i - i')^2 + (j - j')^2}. \quad (4-13)$$

The smallest gray intensity distance between an unclassified $LSB(i, j, k)$ and all currently existing object layers is defined as

$$SID(LSB(i, j, k)) = \min_{\forall n} (ID(LSB(i, j, k), L_n)), \quad (4-14)$$

where n is the index of the existing object layers. The object layer with the smallest gray intensity distance determined by the $SID(LSB(i, j, k))$ is denoted as L_{SI} , and the LSB with the minimum gray intensity distance determined by the $ID(LSB(i, j, k), L_{SI})$ is denoted as LSB_{SI} which is the LSB of the L_{SI} . The value of the $SID(LSB(i, j, k))$ is

the smallest difference of the gray intensity between the unclassified $LSB(i, j, k)$ and all existing object layers. If the $SID(LSB(i, j, k))$ value of an unclassified $LSB(i, j, k)$ is very small, it reflects that the gray intensity of the unclassified $LSB(i, j, k)$ is very similar to the gray intensity of the LSB_{SI} in the object layer L_{SI} . Because the texts and the homogeneous objects have the same gray intensity, it means that the unclassified $LSB(i, j, k)$ may be part of the object layer L_{SI} . The unclassified $LSB(i, j, k)$ should not set up a new object layer to prevent the splitting of the texts or homogeneous objects into more than one object layer.

The largest gray intensity distance between an unclassified $LSB(i, j, k)$ and all currently existing object layers is defined as

$$LID(LSB(i, j, k)) = \max_{\forall n} (ID(LSB(i, j, k), L_n)). \quad (4-15)$$

The object layer with the largest gray intensity distance determined by $LID(LSB(i, j, k))$ is denoted as L_{LI} . The value of the $LID(LSB(i, j, k))$ is the largest difference of the gray intensity between the unclassified $LSB(i, j, k)$ and all existing object layers. If the $SID(LSB(i, j, k))$ values of all unclassified $LSB(i, j, k)$ are very large, it reflects that all the unclassified $LSB(i, j, k)$ are dissimilar to the existing object layers. Hence, the unclassified $LSB(i, j, k)$ which has the largest $LID(LSB(i, j, k))$ should be selected as the seeded LSB to set up a new object layer.

The minimum Euclidean location distance measured between an unclassified

$LSB(i, j, k)$ and all currently existing object layers is defined as

$$SLD(LSB(i, j, k)) = \min_{\forall n} (LD(LSB(i, j, k), L_n)), \quad (4-16)$$

and the object layer with minimum Euclidean location distance determined by the $SLD(LSB(i, j, k))$ is denoted as L_{SL} . The value of the $SLD(LSB(i, j, k))$ is the minimum Euclidean location distance between an unclassified $LSB(i, j, k)$ and all existing object layers.

In this procedure, all unclassified $LSBs$ in the *Pool* are processed to extract their corresponding $SIDs$, $LIDs$ and $SLDs$ with all existing object layers according to the definitions stated above. In order to classify the chosen LSB into an existing object layer which is most similar to the chosen LSB , the unclassified LSB which has the similar gray intensity and is closest to its corresponding object layer is chosen by the SID and SLD values. We select five unclassified $LSBs$ with the smallest SID value at most, which must satisfy the condition - $SID(LSB(i, j, k)) \leq Th_{SI}$, and compute the $SLDs$ between the five unclassified $LSBs$ and their corresponding L_{SL} . Because the texts or the homogeneous objects may contain many different connected regions, each unclassified LSB is part of its corresponding L_{SL} . Then, the unclassified LSB with the smallest SLD will be the seeded LSB and classified to its corresponding L_{SL} .

If the SID values of all unclassified $LSBs$ are larger than Th_{SI} , $SID(LSB(i, j, k)) > Th_{SI}$, so that none unclassified LSB is similar to any of the existing

object layers, then it is appropriate to set up a new object layer by determining the seeded *LSB* with the largest *LID* from the *Pool* to initialize a new object layer.

The Th_{SI} is the predefined threshold and set as 14. The setting of the Th_{SI} value will influence the number of resultant object layers. If the value is too small, then the number of object layers will increase and some homogeneous region may split into more than one object layers, such as broken text lines; while the value is too large, then there may some different object regions be merged into the same object layer.

B. The matching procedure

We now present the matching procedure that assigns each unclassified *LSB* into an existing object layer to which the unclassified *LSB* should belong. The matching procedure analyzes the unclassified *LSBs* from darkest to lightest, left side to right side, and top side to bottom side. Hence, all unclassified *LSBs* are put in a “*Pool*”, and they will be analyzed following the order described above.

The algorithm uses a list to keep track of the representative *LSBs* of the object layers and to determine which object layer the unclassified *LSB* should belong. The representative LSB_q of the object layer L_q must be out of the *LSBs* which are belonging to the L_q and *4-adjacent* to the current unclassified *LSB*. When an unclassified *LSB* is analyzed to determine which object layer is the best match, there may be several object layers to choose. The list stores the representative *LSBs* of the

candidate object layers, where each candidate object layer provides one representative *LSB*. Then we can calculate the match grades between the unclassified *LSB* and all representative *LSBs*, which are *4-adjacent* to the current unclassified *LSB*, in the list to determine which object layer is the best match for the unclassified *LSB*. The match grade is a criterion utilized to calculate how well the unclassified $LSB(i, j, k)$ match with an candidate object layer L_q . Before the computation of the match grade, the pre-match condition is firstly used as to determine whether the representative $LSB_q(i', j', k')$, which represents the object layer L_q , is a candidate for the unclassified $LSB(i, j, k)$.



The match grade is determined by analyzing the D_{LM} and D_{SM} values of the unclassified $LSB(i, j, k)$ and the representative $LSB_q(i', j', k')$. Because some noise pixels will influence the valid pixels in the valid side connection, the D_{SM} could be invalid when the D_{SM} value is small. Hence, the D_{SM} is computed under two cases. 1) When N_{vs} value is larger enough to reflect that the side information of the two adjacent *LSBs* is appropriate, the D_{SM} is taken into consideration for the match grade, i.e.

$$N_{vs}(LSB(i, j, k), LSB_q(i', j', k')) \geq Th_{vs}, \quad (4-17)$$

where Th_{vs} is a predefined threshold. 2) Otherwise, the D_{SM} factor is disabled by setting D_{SM} to zero. Considering the cases of the two adjacent *LSBs* which contains

character patterns with thin strokes cross the side of them, so that the reasonable value of the Th_{vs} can be defined as 5% of the average of K or L values. Since we use $K=L=96$ experimentally as described before, so the $Th_{vs}=5$ is obtained and used in this work.

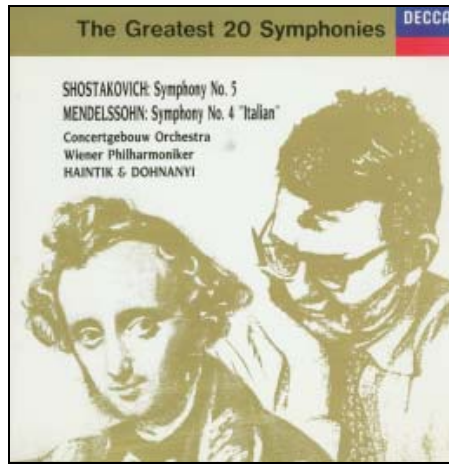
We define two operations for the candidate list:

i) *candidate_insert*($LSB_q(i', j', k')$): which inserts a representative $LSB_q(i', j', k')$ of the object layer L_q into the candidate list.

ii) *candidate_decide*() $\rightarrow L_w$: which computes the match grades of all representative $LSB_q(i', j', k')$ in the candidate list and then finds the best match object layer L_w which has the minimal match grade among all candidates. Hence, the current unclassified $LSB(i, j, k)$ will be classified to the object layers L_w .

After the proposed MLSM algorithm is performed, all $LSBs$ are classified into appropriate object layers. Consequently there are N object layers, L_0, L_1, \dots, L_{N-1} are created. Each object layer possesses a set of the $LSBs$. An object image is created by all pixels belonging to the object layer. Figure 25(a) displays the image of a CD cover. Figures 25(b), (c), and (d) are the object images derived from Fig.25(a) after the MLSM. A detailed analysis of those object images in which all character patterns, foreground objects and background components are well separated, can be easily

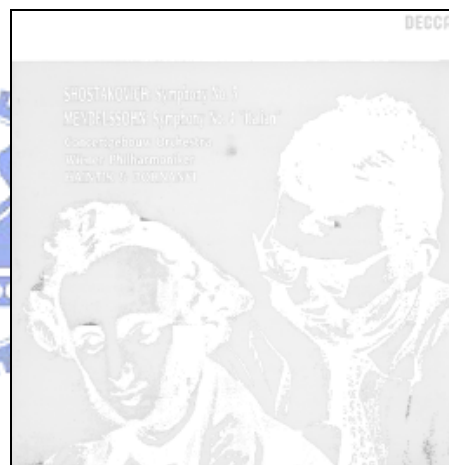
performed. The text-lines will be extracted from each object layer in the text extraction algorithm presented in the following section.



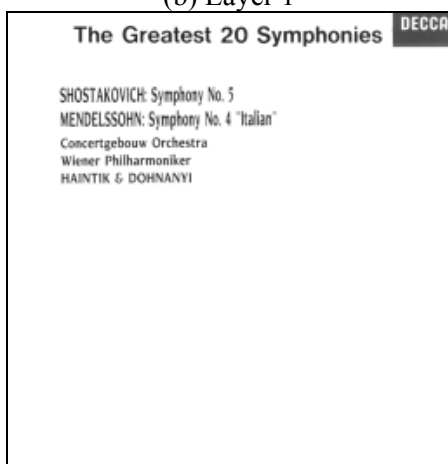
(a) Original image



(b) Layer 1



(c) Layer 2



(d) Layer 3

Fig. 25 An example of the MLSM (image size=1361x1333)

4.3 Text extraction algorithm

After MLSM is performed, the whole document image is decomposed into various object layers. Each object layer may include significant information about characters, foreground objects, background textures or some other objects. Then, each object layer will be binarized by setting the valid pixels of the object layer to “1” and setting the invalid pixels to “0”. Note that in this work only text lines are interested.

The bounding boxes of all connected-components are extracted by the connected-component extraction step. The blocks that contain characters must be identified and they must be organized to form text lines or text regions. The

connected-component-based projection profile method is applied to separate all bounding boxes into different “general text lines”, *GTLs*. Each *GTL* contains a group of bounding boxes.



The following notation is defined:

- (a). CC_i is the i -th connected-component of the binarized object layer.
- (b). CG is a group of connected-components, $CG = \{CC_i, i=0,1,2,\dots,p\}$
- (c). The connected-component CC_i has the top, left, bottom and right coordinates denoted by $t(CC_i)$, $l(CC_i)$, $b(CC_i)$ and $r(CC_i)$, respectively, where $t(CC_i) < b(CC_i)$ and $l(CC_i) < r(CC_i)$.
- (d). The width and height of CC_i are denoted as $W(CC_i)$ and $H(CC_i)$, respectively.

(e). The horizontal and vertical distances between two bounding boxes are defined

as

$$D_h(CC_i, CC_j) = \max[l(CC_i), l(CC_j)] - \min[r(CC_i), r(CC_j)], \quad (4-18)$$

$$\text{and } D_v(CC_i, CC_j) = \min[b(CC_i), b(CC_j)] - \max[t(CC_i), t(CC_j)]. \quad (4-19)$$

If the two bounding boxes are overlapping in the horizontal or vertical direction, the value of the $D_h(CC_i, CC_j)$ or $D_v(CC_i, CC_j)$ will be a negative one.

(f). The horizontal and vertical projection overlap measures of the two bounding

boxes are defined as

$$P_h(CC_i, CC_j) = \frac{-D_h(CC_i, CC_j)}{\min[W(CC_i), W(CC_j)]} \quad (4-20)$$

$$\text{and } P_v(CC_i, CC_j) = \frac{-D_v(CC_i, CC_j)}{\min[H(CC_i), H(CC_j)]} \quad (4-21)$$

Using the functions and notations defined above, the details of the text extraction method are introduced as follows. The method includes two procedures:

The horizontal segmentation procedure ***H-seg(CG_{in})*** (The subscript “*in*” indicates “the input *CG*”) is performed as follows:

- (1). Project all the bounding boxes of the *CCs* in the *CG_{in}* horizontally onto the vertical y-axis.
- (2). Sort all the *CCs* in the *CG_{in}* according to their corresponding $t(CC_i)$, where all the $CC_i \in CG_{in}$. Then scan the horizontal projections of these *CCs* on the y-axis

and determine the “shadow segments” of these CCs on the y -axis. The CCs that are said to be sharing the same shadow segment must have their horizontal projections of bounding boxes overlap on the y -axis, and can be detected when $P_v(CC_i, CC_j) > 0$.

(3). For each shadow segment, group the CCs which are covered by the same shadow segment into an individual CG .

(4). After the above steps are performed, there are many CGs produced, CG_K , where $K = 0, 1, 2, \dots, k-1$. For each CG_K , perform the vertical segmentation procedure $V\text{-seg}(CG_K)$.



The vertical segmentation procedure $V\text{-seg}(CG_K)$ is performed as follows:

- (1). Project all the bounding boxes of CCs of the CG_K vertically onto the x -axis.
- (2). Sort all the CCs in the CG_K according to their corresponding $l(CC_i)$, where all the $CC_i \in CG_K$. Then scan the vertical projections of these CCs on the x -axis and determine their shadow segments. The CCs sharing the same shadow segment of their vertical projections on the x -axis can be detected when $P_h(CC_i, CC_j) > 0$.
- (3). For each shadow segment, group the CCs which are covered by the same shadow segment into individual CGs .

(4). Determine the two merge conditions of the adjacent CGs, CG_{K1} and CG_{K2} ,

which are: i) whether the horizontal space between the two adjacent CGs is sufficiently small, that is,

$$\min D_h(CC_i, CC_j) < \max(Avg_W(CG_{K1}), Avg_W(CG_{K2})), \quad (4-22)$$

where the $\min D_h(CC_i, CC_j)$ is the minimal horizontal distance between the two CGs, the $CC_i \in CG_{K1}$ and the $CC_j \in CG_{K2}$, and $Avg_W(CG)$ is the average width of all CCs belonging to this CG; ii) whether the average heights of the CCs belonging to the two CGs are similar, that is, the ratio of the two average heights should be within a reasonable range

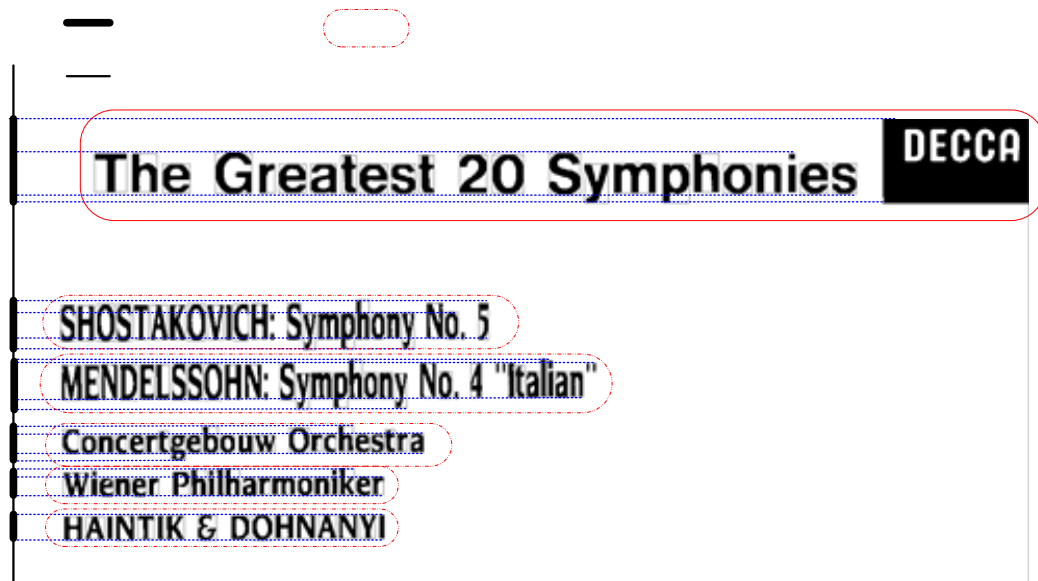
$$0.67 \leq Avg_H(CG_{K1}) / Avg_H(CG_{K2}) \leq 1.5. \quad (4-23)$$

If the above two conditions are satisfied, then merge the two adjacent CGs.

(5). After the above steps are performed, there are many CGs of CCs produced, CG_L , where $L = 0, 1, 2, \dots, l-1$. If only one resultant CG_0 is obtained, then terminate the segmentation procedure; otherwise, for each CG_L , perform $H-seg(CG_L)$.

As defined above, the text extraction procedure is performed on all CCs in processing a certain object image by recursive segmentation, involving $H-seg$ and $V-seg$ procedures. The sets of all the CCs extracted from the processed object image are defined as the CGs, the processes of the text extraction algorithm and the results

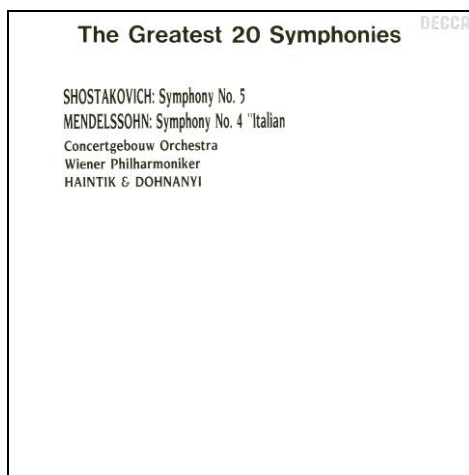
are illustrated in Fig.26.



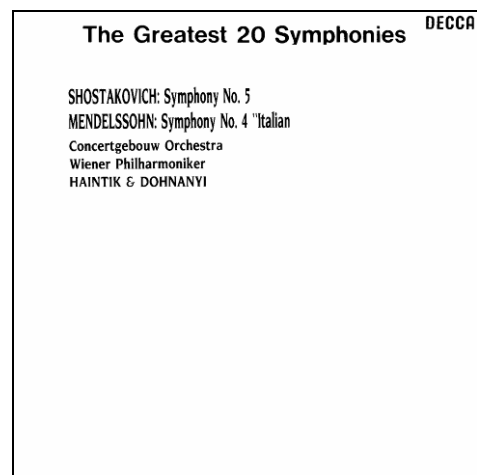
(a). The result of the *H-seg* procedure on the *CCs* of the Fig.25(d).



(b). The result of the *V-seg* procedure of the first *CG* of the Fig.26(a).



(c). The text plane after the text extraction algorithm



(d). The binary image of the text plane

Fig.26 The example of the text extraction algorithm of the Fig.25(d)

We denote the resultant *CGs* as the “general text lines”, *GTLs*. Figure 26(a) shows the result of the *H-seg* procedure on the *CCs* of the Fig.25(d), and five *CGs* are obtained. Then the five *CGs* are performed the *V-seg* procedure in turn. We take the first *CG* for example and the result is shown in Fig.26(b). Then, the *CGs* obtained from the *V-seg* procedure is divided into two *CGs* according the condition 4-23. The two *CGs* obtained from the *V-seg* procedure will be performed the *H-seg* procedure in turn and both cannot be divided into more *CGs*. Hence the two *CGs* are the resultant *GTLs* and then are checked by the text-line decision rules.

We state a set of knowledge-based decision rules to determine whether each *GTL* is a text line or a non-text region. If one *GTL* meets the requested rules of a text-line, it will be identified as a text-line. The shape and contents of the bounding box are determined from the *GTL*, and the features of the text line, such as the transition pixel ratio, the foreground pixel density ratio, and the block size, are considered. A “1” represents a valid pixel and a “0” represents an invalid pixel. The transition pixel is at the boundary of the foreground pixels.

The horizontal transition pixel ratio of the *GTL* block is defined as

$$T_h = \frac{\text{Total number of the transition pixels of the } GTL}{Col_N}, \quad (4-24)$$

where the Col_N is the number of the pixel columns in which the valid pixels are present.

The valid pixel density of the *GTL* is defined as,

$$B = \frac{\text{Total number of valid pixels of the } GTL}{A}, \quad (4-25)$$

where the A is the area of the bounding box of the *GTL*.

The width and height of the bounding box of the *GLT*, and the number of the *CCs* belonging to the *GTL* are W , H and N_c , respectively.

Using these features defined above, a *GTL* block is identified as a text-line block if all of the following decision rules are met.

$$(i). \quad 1.1 < T_h < 4.0 \quad (4-26)$$

$$(ii). \quad 0.2 < B < 0.7 \quad (4-27)$$

$$(iii). \quad W/H \geq 2.0 \quad (4-28)$$

$$(iv). \quad 0.5(W/H) \leq N_c \leq 8.0(W/H) \quad \text{and} \quad 2 \leq N_c \quad (4-29)$$

$$(v). \quad \frac{\sum A_i}{A} \geq 0.4, \quad \text{where } A_i \text{ is the area of the } i\text{-th } CC \text{ of the } GTL \quad (4-30)$$

The ratio of the number of the transition pixels used in the condition (i) is to evaluate the complexity of the area of the *GTL*. The valid pixel density utilized in the condition (ii) measures the density of the valid pixels. The conditions (iii)-(v) determine whether the *CCs* in the *GTL* are well aligned, that is, if a series of the *CCs* is a text line, they should be well aligned. The above decision conditions are determined from analyzing many experimental results of processing document images which are with text strings with various types, lengths and sizes. The constant

values utilized in the above decision conditions are determined experimentally and achieve good performance in most general cases.

After all object layers have been processed to extract all text-lines from them, all text lines are extracted and collected as the final segmentation result, as depicted in Fig. 26(c). Figure 26(d) is the binarized text image of Fig. 26(c).

4.4 Experimental results and discussions

This study illustrates 24-bit true color or 8-bit monochromatic document images at 300dpi, full page. The proposed method for automatic text segmentation has been tested on numerous magazine images, cover images and advertisement images. Figures 27(a)~32(a) display parts of the test images. The background images in Figs. 27(a) to 32(a) include the following features. 1) Monochromatic background with/without text; 2) slowly varying background with/without text; 3) highly varying background with/without texts, and 4) complex varying background with/without text of various colors.

Figures 27(b)~32(b) present the text planes in Figs. 27(a)~32(a) after the proposed text segmentation method is implemented. Figures 27(c) to 32(c) show parts of the object layers in Figs. 27(a) to 32(a). The ratio of success of the proposed text segmentation method is,

$$\text{Ratio of success} = \frac{\text{Number of texts extracted}}{\text{Total number of texts}} \% \quad (4-31)$$

The ratios of success in Figs. 27(b)~32(b) are 100%, 98.5%, 99.2%, 98.7%, 100%, and 97%, respectively. The proposed text segmentation method can be successfully applied to extract texts with different typefaces or sizes, as well as those spread in a compound document image with monochromatic, slowly varying, highly varying and complex varying backgrounds.

The MLSM decomposes the document image into several object layers. All of the texts are spread into different object layers, according to their colors. The text extraction algorithm extracts the text from all of the object layers. Different object layers may contain text-like blocks in a particular position, so the text extracting algorithm could make the wrong decision. Consequently, the text extraction algorithm can be further improved. For instance, although most of the text in Fig.28(a) overlays a complex varying background - a map - all of the text that overlaps the map is segmented into one of the object layers in Fig. 28(c). Although the ratios of success in Figs. 28(b), 29(b), 30(b) and 32(b) are not 100%, the MLSM successfully segments all of the texts.

According to our results, the texts can be extracted from different backgrounds, regardless of whether the texts overlap a simple, slowly or rapidly varying background. This method overcomes various issues raised by the complexity of

background images. Consequently, the multi-layer segmentation algorithm constitutes an effective solution for extracting text from various document images.

In block-based clustering algorithm, the parameters TH_{JDF} and TH_{σ} are the threshold values to decide which cluster is convergence when the conditions, $JDF > TH_{JDF}$ and $\sigma < TH_{\sigma}$, are met. The JDF value measures the separability between two adjacent clusters in the block-based clustering algorithm. The JDF value may lies within the range $0 \leq JDF \leq 1$. Maximizing the JDF value can be utilized as an objective function, to optimize the segmentation result. Hence, when the JDF approximates 1.0 , the two adjacent clusters are ideally and completely separated. When the number of the clusters is more than two, the average JDF is used to measure the separability of the clusters. This study employs $TH_{JDF}=0.9$.

The standard deviation, σ , measures the compactness of the pixel values of each clusters. Ideally, the σ approximates zero for a monochromatic object. In a pilot experiment, we analyze the widespread distributions, caused by the scanner or the original document, of the pixel values of monochromatic texts in different document images. The average variation of the monochromatic texts with different size or style is around $0\sim 50$. In general, the $TH_{\sigma}=25$ can obtain good preservation of the texts, but it is insufficient for our needs. Therefore, this study employs $TH_{\sigma}=14$ to obtain better outcome, when the texts overlap a background with rapidly varying texture and


similar grayscale. When the TH_σ is below 25, the extracted texts are thinner than original texts and the boundary of the texts are clustered to different object layer, as the Fig.23(j).

Because the value of the TH_σ is set as 14, the standard deviation, σ , of each *LSB* will less than 14. In other words, if two *LSBs* belong to the same object layer, the difference of the average values between the two *LSBs* will be less than 14. The threshold values of the Th_{LM} and Th_{SI} are used to judge whether the two *LSBs* are belong to the same object layer or not by the difference of the average values. Therefore, the threshold values of the Th_{LM} and Th_{SI} are set as 14. In the decision procedure for constructing of a new object layer, we use the Th_{SI} to determine which unclassified *LSB* should be merged with an existing object layer or set up a new object layer. In the pre-match condition of the matching procedure, the Th_{LM} is used to filter out the unreasonable object layers in order to save the computation power.

The segmentation method proposed by the chapter has experimented on a large number of different document images, scanned from book covers, advertisements, brochures, and magazines. We find that the monochromatic objects, text or non-text, can be successfully separated from a document image by the MLSM, nevertheless, a few texts could be failed to extract when the pixel values of the texts are multicolor, gradual change, or too close to the pixel values of the background. A multicolor or

gradual change text brings the text fragmented and distributed to different clusters. A text could be merged with its background when the values of the text are too close to the values of its background. Although, decreasing the parameter TH_{σ} (below 14) can separate the text and the overlapped background, whose values are too close to the text, to solve the merged problem, it will cause the text fragmented and distributed to different clusters. Therefore, an adaptive threshold TH_{σ} is the future work to solve the merged problem.

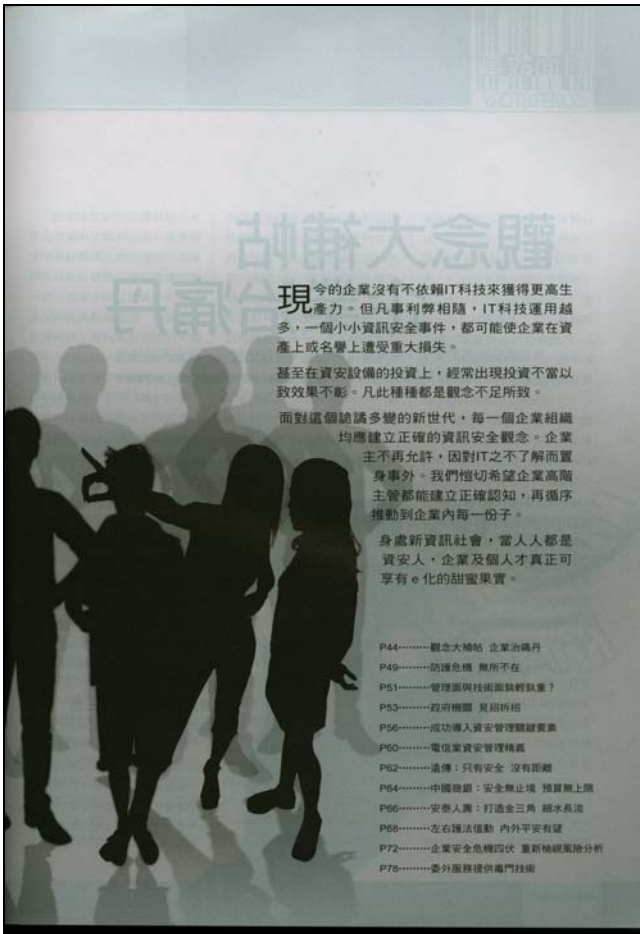
4.5 Concluding remarks



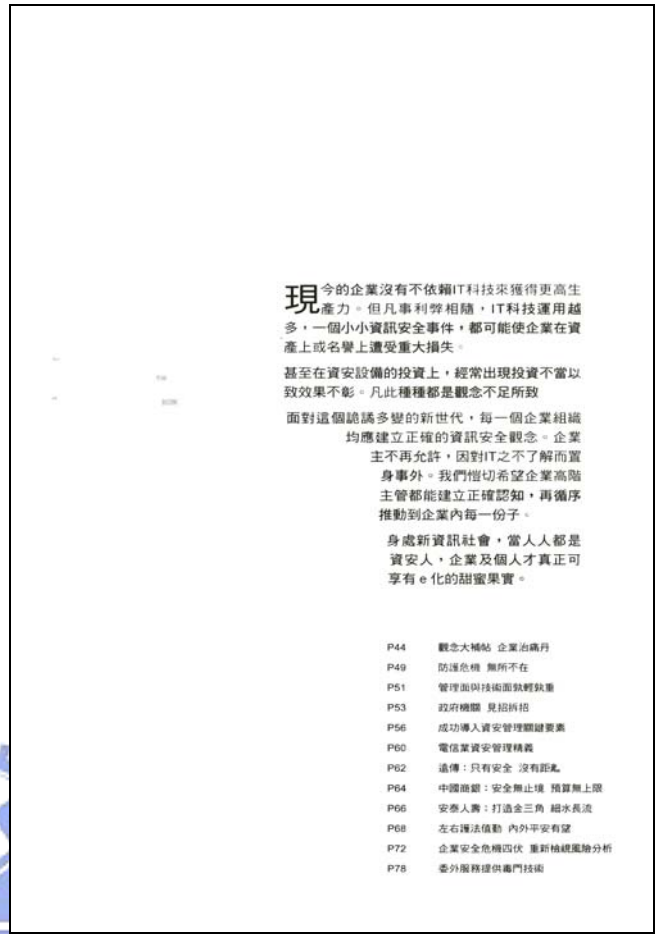
This study presents a viable method for extracting texts from a complex compound document image in which texts overlay various background images. The proposed segmentation algorithm uses a multi-layer segmentation method to segment the texts from various compound document images, regardless of whether the texts overlap the background. This method overcomes various issues raised by the complexity of the background images. Experimental results obtained with various document images reveal that the proposed algorithm can successfully segment Chinese and English text strings from various backgrounds, regardless of whether the texts overlap a simple, slowly or rapidly varying background. The method can be used to improve the effectiveness of compression; the technique has many applications, including compressing color faxes and documents. Moreover, the segmentation

algorithm can be used in Optical Character Recognition (OCR) to search for characters in complex documents strong text/background overlap.





(a) Original image

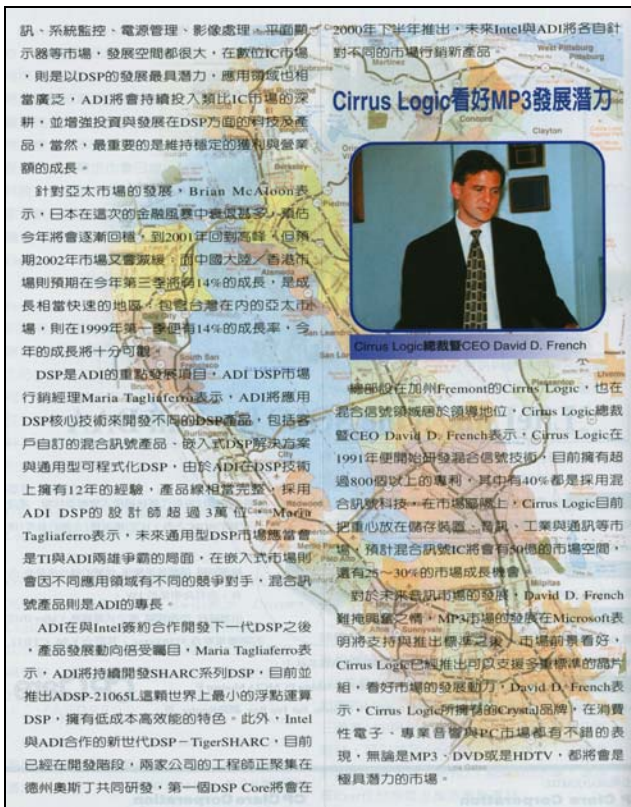


(b) Text plane



(c) Parts of layer planes

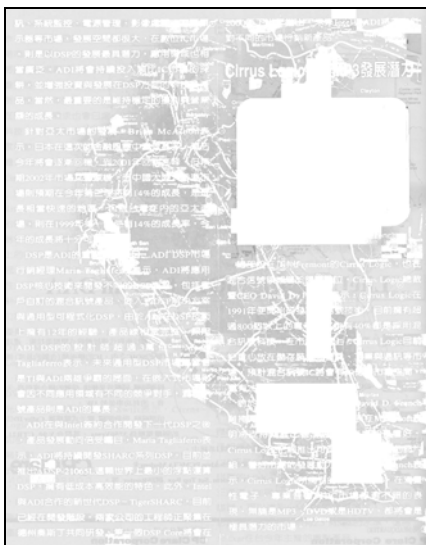
Fig.27 Test image 1 (image size=2262x3263)



(a) Original image

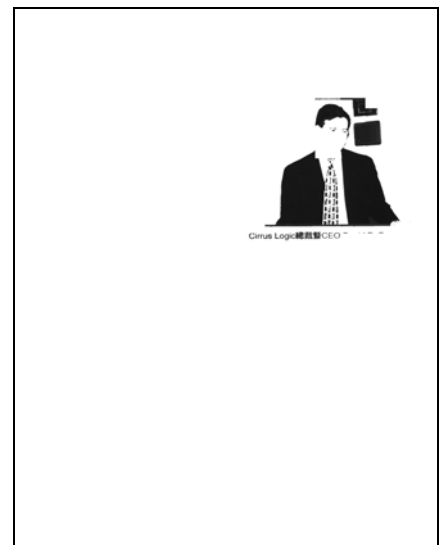


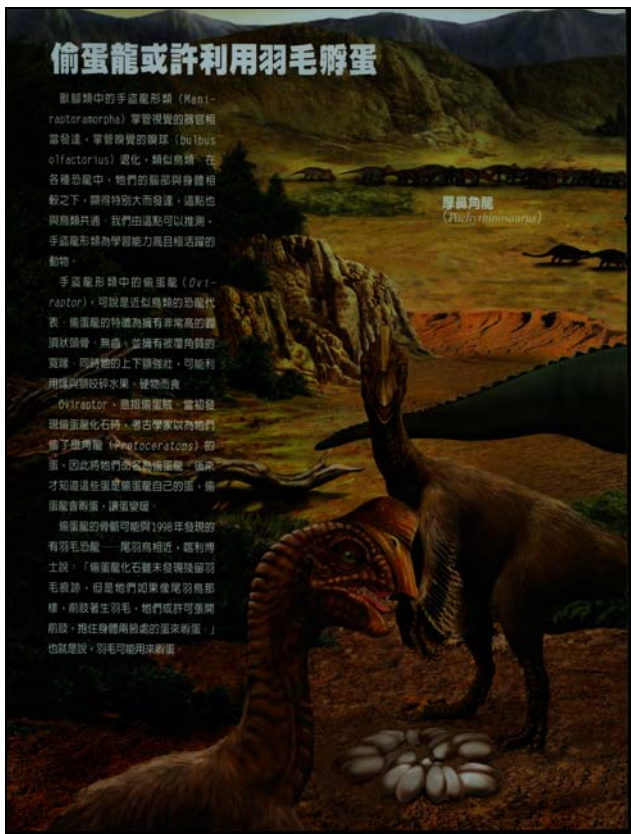
(b) Text plane



(c) Parts of layer planes

Fig.28 Test image 2 (image size=1829x2330)





(a) Original image

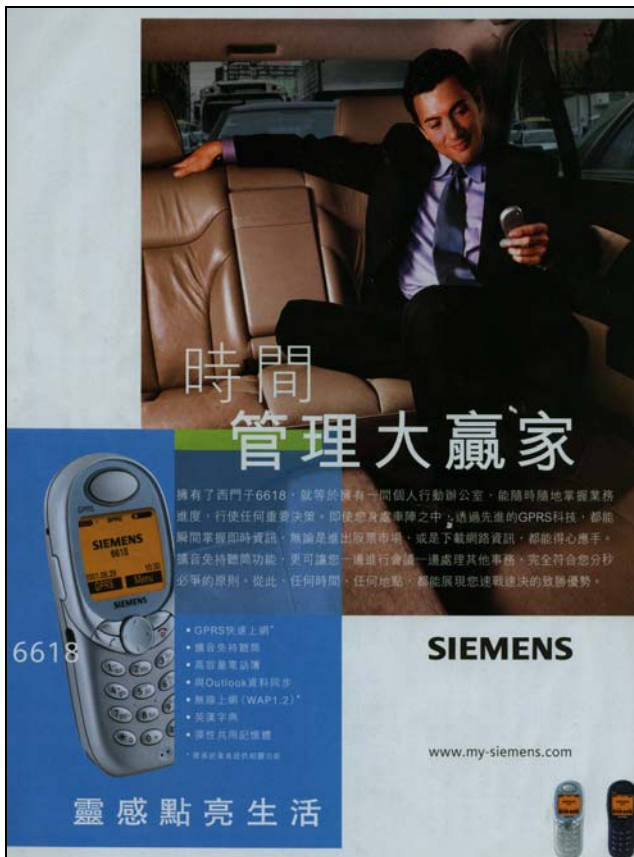


(b) Text plane



(c) Parts of layer planes

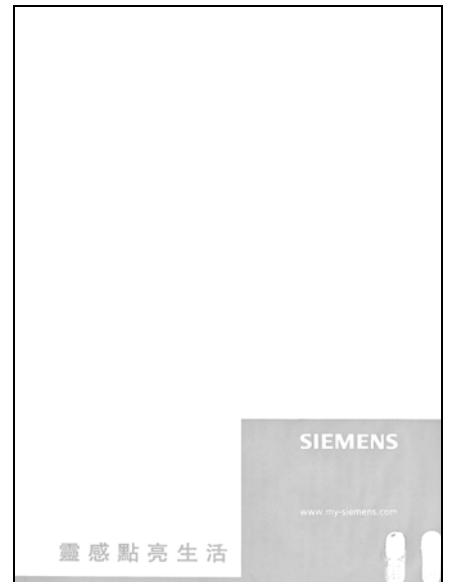
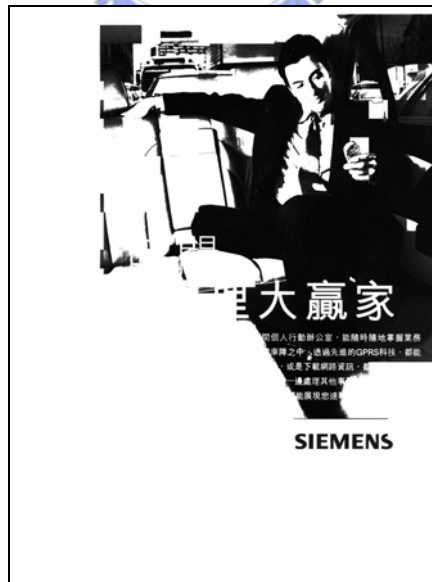
Fig.29 Test image 3 (image size=2462x3250)



(a) Original image

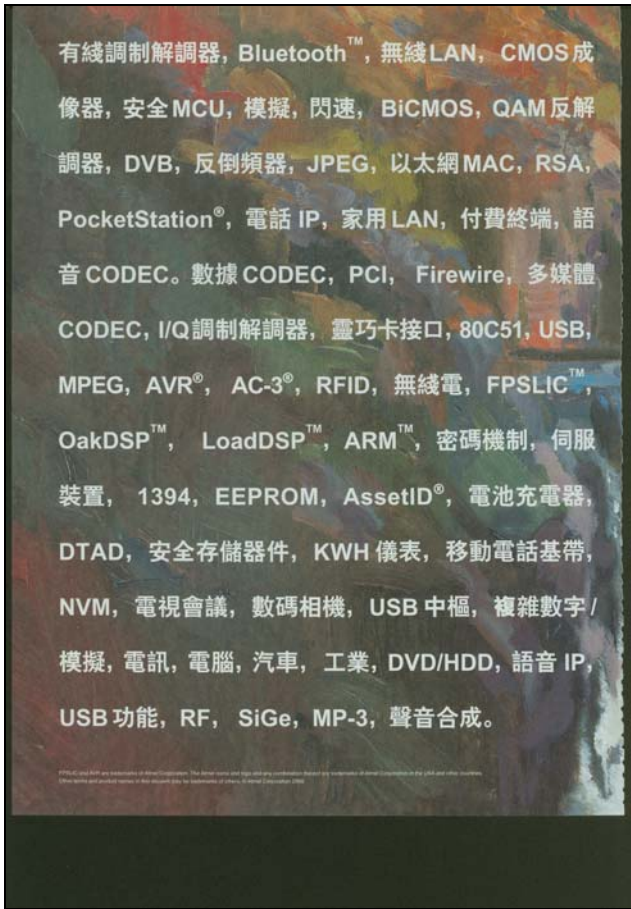


(b) Text plane



(c) Parts of layer planes

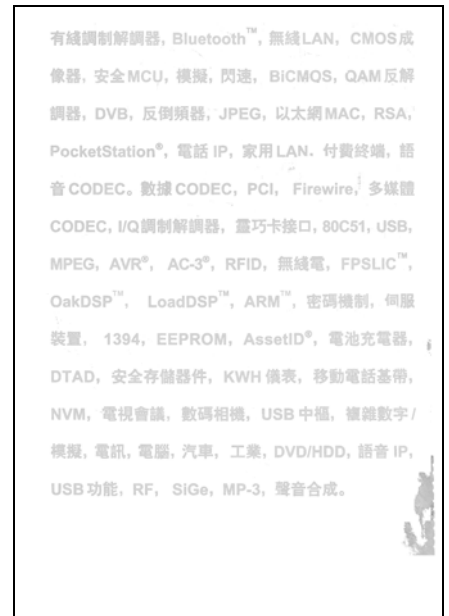
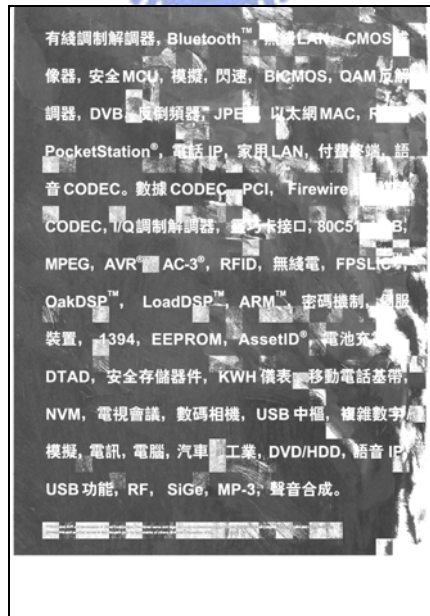
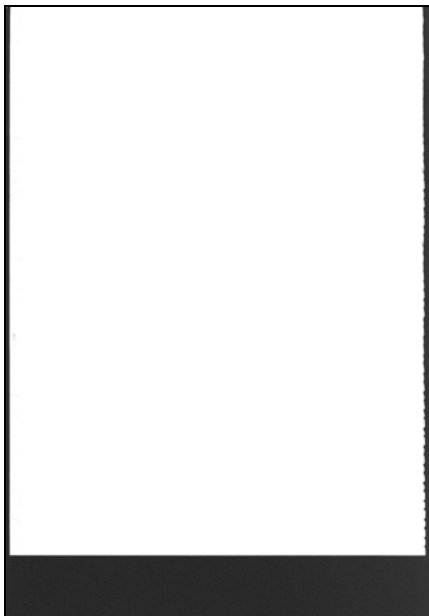
Fig.30 Test image 4 (image size=2333x3153)



(a) Original image



(b) Text plane



(c) Parts of layer planes

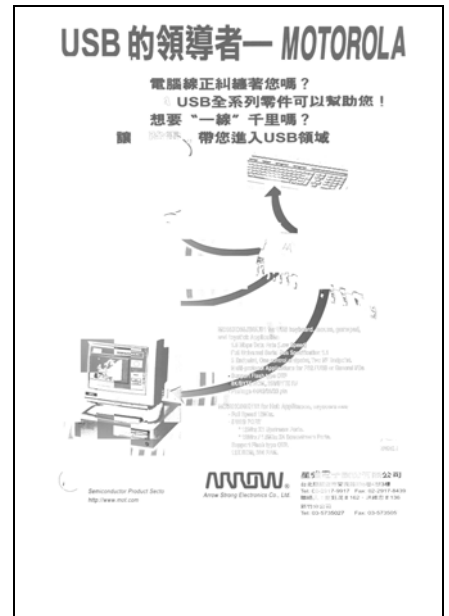
Fig.31 Test image 5 (image size=2469x3535)



(a) Original image



(b) Text plane



(c) Parts of layer planes

Fig.32 Test image 6 (image size=2469x3535)

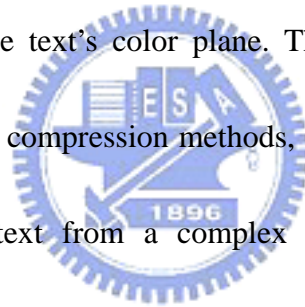
CHAPTER 5

CONCLUSIONS AND PERSPECTIVE

This dissertation presents three segmentation methods for document image compression. In the Chapter 2, a compression method for color document images based on the wavelet transform and fuzzy picture-text segmentation was presented. This approach addresses a fuzzy picture-text segmentation method, which separates pictures and texts by using wavelet coefficients from color document images. The number of colors, the ratio of projection variance, and the fractal dimension are utilized to segment the pictures and texts. By using the fuzzy characteristics of these parameters, a fuzzy rule is proposed to achieve the purpose of picture-text image segmentation. Two components, text strings and pictures, are generated and processed by different compression algorithms. The picture components and the text components are encoded by zerotree wavelet coding and by the modified run-length Huffman coding, respectively.

However, the fuzzy picture-text segmentation method does not suitable for the document images whose texts are overlap with a complex background. Therefore, two

algorithms for compressing image documents with large text/background overlap are proposed in Chapter 3. The proposed algorithms apply a new segmentation method to separate the text from the image in a compound document in which the text and background overlap. The segmentation method classifies document images into three planes: the text plane, the background (non-text) plane, and the text's color plane. Different compression techniques are used to process the text plane, the background and the text's color plane. The text plane is compressed using the pattern matching technique, called JB2. Wavelet transform and zerotree coding are used to compress the background plane and the text's color plane. The proposed algorithms greatly outperform the famous image compression methods, JPEG and DjVu, and enable the effective extraction of the text from a complex background, achieving a high compression ratio for compound document images.



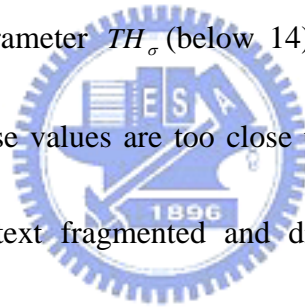
Although the segmentation method in the Chapter 3 outperforms the famous image compression methods, JPEG and DjVu, it does not apply when backgrounds include sharply varying contours or overlap with texts. These background images include 1) monochromatic backgrounds with/without texts; 2) slowly varying backgrounds with/without texts; 3) highly varying background with/without texts and, 4) complex varying backgrounds with/without texts with different colors. Extracting the texts is particularly difficult when the compound document image includes all of

these backgrounds.

In Chapter 4, a viable method for extracting texts from a complex compound document image in which texts overlay various background images is presented. The proposed segmentation algorithm uses a multi-layer segmentation method (MLSM) to segment the texts from various compound document images, regardless of whether the texts overlap the background. The MLSM method overcomes various issues raised by the complexity of the background images. Experimental results obtained with various document images reveal that the proposed algorithm can successfully segment Chinese and English text strings from various backgrounds, regardless of whether the texts overlap a simple, slowly or rapidly varying background. The method can be used to improve the effectiveness of compression; the technique has many applications, including compressing color faxes and documents. Moreover, the segmentation algorithm can be used in Optical Character Recognition (OCR) to search for characters in complex documents strong text/background overlap.

According to our results, the texts can be extracted from different backgrounds, regardless of whether the texts overlap a simple, slowly or rapidly varying background. This method overcomes various issues raised by the complexity of background images. Consequently, the multi-layer segmentation algorithm constitutes an effective solution for extracting text from various document images.

The MLSM method has experimented on a large number of different document images, scanned from book covers, advertisements, brochures, and magazines. We find that the monochromatic objects, text or non-text, can be successfully separated from a document image by the MLSM, nevertheless, a few texts could be failed to extract when the pixel values of the texts are multicolor, gradual change, or too close to the pixel values of the background. A multicolor or gradual change text brings the text fragmented and distributed to different clusters. A text could be merged with its background when the values of the text are too close to the values of its background. Although, decreasing the parameter TH_{σ} (below 14) can separate the text and the overlapped background, whose values are too close to the text, to solve the merged problem, it will cause the text fragmented and distributed to different clusters. Therefore, an adaptive threshold TH_{σ} is the future work to solve the merged problem.



REFERENCE

- [1] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. Image Processing*, Vol. 1, No. 2, pp. 205-220, 1992.
- [2] S. C. Mallet, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, 1989.
- [3] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelets Coefficients," *IEEE Trans. Signal Processing*, Vol. 41, No. 12, pp. 3445-3462, 1993.
- [4] A. Said and W. A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 6, No. 3, pp. 243-250, 1996.
- [5] A. Leger, T. Omachi, and G.K. Wallace, "JPEG Still Picture Compression Algorithm," *Optical Engineering*, Vol. 30, No. 7, pp. 947-954, 1991.
- [6] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents," *Computer Graphics and Image Processing*, Vol. 20, pp. 375-390, 1982.
- [7] G. Nagy, S. C. Seth and S. D. Stoddard, "Document Analysis with an Expert System", *Pattern recognition practice II*, pp. 149-159, 1986.
- [8] L. A. Fletcher and R. Kasturi, "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, pp. 910-918, 1988.
- [9] Mohamed Kamel and Aiguo Zhao, "Extraction of Binary Character/Graphics Images from Grayscale Document Images," *CVGIP: Graphical Models and Image Processing*, Vol. 55, No. 3, pp. 203-217, 1993.

- [10] Wen-Hsiang Tsai, "Moment-Preserving Thresholding : A New Approach," *Computer Vision, Graphics, and Image Processing*, Vol. 29, pp. 377-393, 1985.
- [11] J. L. Fisher, S. C. Hinds and D. P. D'amato, "A Rule-Based System for Document Image Segmentation," *Proceedings of 10th IEEE international conference on Pattern Recognition*, Vol.1, pp. 567-572, 1990.
- [12] T. Akiyama and N. Hagita, "Automated Entry System for Printed Documents," *Pattern Recognition*, Vol. 23, No. 11, pp. 1141-1154, 1990.
- [13] F. Y. Shih, S. S. Chen, D. C. D. Hung and P. A. Ng, "A Document Segmentation, Classification and Recognition System," *Proceedings of IEEE international conference on System integration*, pp.258-267, 1992.
- [14] T. Pavlidis and J. Zhou, "Page Segmentation and Classification," *CVGIP: Graph. Models Image Process.*, Vol. 54, No. 6, pp. 484-496, 1992.
- [15] A. A. Zlatopolsky, "Automated Document Segmentation," *Pattern Recognition Lett.*, Vol. 15, No. 7, pp. 699-704, 1994.
- [16] D. Wang and S. N. Srihari, "Classification of Newspaper Image Blocks using Texture Analysis," *Computer Vision Graph. Image Process.*, Vol. 47, pp. 327-352, 1989.
- [17] A. K. Jain and S. Bhattacharjee, "Text Segmentation using Gabor Filters for Automatic Document Processing," *Mach. Vis. Appl.*, Vol. 5, pp. 169-184, 1992.
- [18] C. Fortin, R. Kumaresan, W. Ohley and S. Hofer, "Fractal Dimension in The Analysis of Medical Images," *IEEE Engineering in Medicine and Biology Magazine*, Vol. 11, pp. 65 -71, 1992.
- [19] H. M. Suen and J. F. Wang, "Text String Extraction from Images of Colour-Printed Documents," *IEE Proc.-Vis. Image Signal Process.*, Vol. 143, No. 4, pp. 210-216, 1996.

- [20] P. Haffner, L. Bottou, P. G. Howard, P. Simard, Y. Bengio and Y. Le Cun, "Browsing Through High Quality Document Images with DjVu," *Proceedings. IEEE International Forum on Research and Technology Advances in Digital Libraries*, pp.309-318, 1998.
- [21] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio and Y. Le Cun, "High Quality Document Image Compression with DjVu," *Journal of Electric Imaging*, Vol.7, No.3, pp.410-425, 1998.
- [22] B. B. Chaudhuri and N. Sarkar, "Texture Segmentation using Fractal Dimension," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 72 –77, 1995.
- [23] George J. Klir et al., *Fuzzy Sets and Fuzzy Logic : Theory and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1995.
- [24] N. Sarkar and B. B. Chaudhuri, "An Efficient Differential Box-Counting Approach to Compute Fractal Dimension of Image," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, pp. 115 –120, 1994.
- [25] S. Buczkowski, S. Kyriacos, F. Nekka and L. Ccartililer, "The Modified Box-Counting Method: Analysis of Some Characteristic Parameters," *Pattern Recognition*, Vol. 31, No. 4, pp. 441-418, 1998.
- [26] R. L. Queiroz, Z. Fan and T. D. Tran, "Optimizing Block-Thresholding Segmentation for Multilayer Compression of Compound Images," *IEEE Transaction on Image Processing*, Vol. 9, No.9, pp.1461-1471, 2000.
- [27] Y. Lin and S. N. Srihari, "Document Image Binarization Based on Texture Features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No 5, pp. 540-544, 1997.

- [28] D. Huttenlocher, P. Felzenszwalb and W. Rucklidge, "DigiPaer: A Versatile Color Document Image Representation," *Proc. IEEE Intl. Conf. Image Proc.*, pp.219-223, 1999.
- [29] P. G. Howard, "Text Image Compression using Soft Pattern Matching," *Computer Journal*, Vol. 40, No. 2, pp.146-156, 1997.
- [30] "JBIG Committee FDIS Text," ISO/IEC International Standard 14492, 1999.
- [31] "MRC. Mixed raster content mode," ITU Recommendation T.44, 1997.
- [32] "JBIG. Progressive bi-level image compression," ITU recommendation T.82, ISO/IEC International Standard 11544, 1993.
- [33] D. Taubaman and A. Zakhor, "Multi-Rate 3D Subband Coding of Video," *IEEE Trans. Image Processing*, Vol. 3, No.5, pp. 572-588, 1994.
- [34] H. Yang, M. Kashimura, N. Onda and S. Ozawa, "Extraction of Bibliography Information Based on Image of Book Cover," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.14, No.7, pp.963-978, 2000.
- [35] J. Li and R. M. Gray, "Context-Based Multiscale Classification of Document Images Using Wavelet Coefficient Distributions," *IEEE Trans. on Image Processing*, Vol.9, No.9, pp.1604-1616, 2000.
- [36] H. Choi and R. G. Baraniuk, "Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models," *IEEE Trans. on Image Processing*, Vol.10, No.9, pp.1309-1321, 2001.
- [37] H. Cheng and C. A. Bouman, "Multiscale Bayesian Segmentation Using a Trainable Context Model," *IEEE Trans. on Image Processing*, Vol.10, No.4, pp.511-524, 2001.
- [38] Bing-Fei Wu, Chung-Cheng Chiu and Wen-Long Lin, "Wavelet-based Images Compression of Color Document by Fuzzy Picture-Text Segmentation," *Journal of The Chinese Institute of Engineers*, Vol. 26, No.1, pp.113-118, 2003.

- [39] M. Acharyya and M. K. Kundu, "Document Image Segmentation Using Wavelet Scale-Space Features," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.12, No.12, pp.1117-1127, 2002.
- [40] J. Zhou and D. Lopresti, "Extracting Text from WWW Images," *Proceedings of International Conference on Document Analysis and Recognition*, pp.248-252, 1997.
- [41] M. Worring and L. Todoran, "Segmentation of Color Documents by Line Oriented Clustering using Spatial Information," *Proceedings of International Conference on Document Analysis and Recognition*, pp.67-70, 1999.
- [42] M. Pietikinen and O. Okun, "Edge-Based Method for Text Detection from Complex Document Images," *Proceedings of International Conference on Document Analysis and Recognition*, pp.286-291, 2001.
- [43] Q. Yuan and C.L. Tan, "Text Extraction from Gray Scale Document Images using Edge Information," *Proceedings of International Conference on Document Analysis and Recognition*, pp.302-306, 2001.
- [44] V. Wu, R. Manmatha and E.M. Riseman, "Finding Text in Images," *Proceedings of 2nd ACM International Conference on Digital Libraries*, pp.3-12, 1997.
- [45] H.P. Li, D. Doermann and O. Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Trans. on Image Processing*, Vol.9, No.1, pp.147-156, 2000.
- [46] R. Lienhart and A. Wernicked, "Localizing and Segmenting Text in Images and Videos," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.12, No.4, pp.236-268, 2002.
- [47] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.8, pp.62-66, 1978.

VITA

博士候選人簡歷

姓名：瞿忠正

性別：男

生日：民國 56 年 1 月 18 日

出生地：台南市

論文題目：

中文：複雜型複合式文件影像壓縮方法之研究

英文：THE STUDY OF THE COMPRESSION ALGORITHMS FOR COMPLEX
COMPOUND DOCUMENT IMAGES

學經歷：

1. 75 年 9 月～79 年 7 月
2. 79 年 7 月～81 年 7 月
3. 81 年 9 月～83 年 7 月
4. 83 年 7 月～now
5. 87 年 9 月～now



中正理工學院電機工程學系
中正理工學院電機工程學系 助教
中正理工學院電子工程研究所
中正理工學院電機工程學系 講師
在職進修國立交通大學電機與控制工程研究所博士
學位

榮譽：

- 一、第五屆 TIC100 創業競賽 冬令營 冠軍
- 二、第五屆 TIC100 創業競賽 總決賽 銀質獎
- 三、第十七屆宏碁龍騰知識經濟論文獎 金質獎



第十七屆宏碁龍騰知識經濟論文獎 金質獎



第五屆 TIC100 創業競賽冬令營冠軍



第五屆 TIC100 創業競賽總決賽銀質獎

PUBLICATION LIST

博士候選人著作目錄

姓名：瞿忠正 (Chung-Cheng Chiu)

Journal

- [1] Bing-Fei Wu, Chung-Cheng Chiu and Wen-Long Lin, “Wavelet-Based Images Compression of Color Document by Fuzzy Picture-Text Segmentation,” *Journal of The Chinese Institute of Engineers*, Vol. 26, No.1, pp.113-118, 2003.
- [2] Bing-Fei Wu, Chung-Cheng Chiu and Yen-Lin Chen, “Algorithms for Compressing Compound Document Images with Large Text/Background Overlap”, accepted by *IEE Proceedings-Vision, Image, and Signal Processing*, June 2003.
- [3] Bing-Fei Wu, Yen-Lin Chen and Chung-Cheng Chiu, “Recursive Algorithms for Image Segmentation Based on a Discriminant Criterion,” accepted by *International Journal of Signal Processing*, Sep. 2004.
- [4] Bing-Fei Wu, Yen-Lin Chen, Chao-Jung Chen, Chung-Cheng Chiu and Chorng-Yann Su, “A Real-Time Wavelet-Based Video Compression Approach to Intelligent Video Surveillance Systems,” accepted by *International Journal of Computer Applications in Technology*, Sep. 2004.

Conference

- [1] Bing-Fei Wu, Wen-Long Lin and Chung-Cheng Chiu, “Wavelet-Based Color Document Compression by Fuzzy Graph-Text Segmentation,” *Int. Symposium on Multimedia Information Processing*, pp.356-361, Chung-Li, Taiwan, Dec. 14-16, 1998.
- [2] Chung-Cheng Chiu, Chia-Feng Lin and Bing-Fei Wu, “High Quality Image Compression in Compound Text and Image Documents,” *2000 13th IPPR Conference on Computer Vision Graphics and Image Processing*, No.2, pp.15-22, Aug. 20-22, 2000.
- [3] Bing-Fei Wu, Yen-Lin Chen, Chung-Cheng Chiu and Chorng-Yann Su, “A Novel Image Segmentation Method for Complex Document Images,” *Proceedings of the Symposium on Computer Vision, Graphics, and Image Processing, CVGIP2003*, pp.646~654, July, Kinmen, Taiwan.
- [4] Bing-Fei Wu, Yen-Lin Chen and Chung-Cheng Chiu, “Multi-Layers Segmentation Method for Complex Document Images,” *Proceedings of the 7th Joint Conference on Information Science, JCIS2003, and the 5th International Conference on Computer Vision, Pattern Recognition and Image Processing, CVPRIP03*, pp.647~650, Sept. 26 - 30, 2003, Cary, North Carolina.
- [5] Chao-Jung Chen, Chung-Cheng Chiu, Bing-Fei Wu, Shin-Ping Lin and Chia-Da Huang, “The Moving Object Segmentation Approach to Vehicle Extraction,” *2004 IEEE*

International Conference on Networking, Sensing and Control, ICNSC2004, vol.1,
pp.19-23, March 21-23, 2004, Taipei, Taiwan.

[6] Bing-Fei Wu, Yao-Chun Hung, Yen-Lin Chen, Chao-Jung Chen, Chung-Cheng Chiu
and Chorng-Yann Su, “A High-Speed Wavelet-Based Video Codec for Video
Surveillance Systems,” *13th Automation Technology Conference*, pp.1123-1130, June
17-18, 2004, Taipei, Taiwan.

[7] Bing-Fei Wu, Chung-Cheng Chiu, Chao-Jung Chen, Wen-Chen Wu, Jau-Woei Perng
and Tsu-Tian Lee, “An Intelligent Vision-Based Real-Time Integration System on
Autonomous Vehicles,” *International Conference on Intelligent Systems and Control,*
ISC 2004, pp. 51-57, Aug. 23-25, 2004, Honolulu, Hawaii, USA.

[8] Bing-Fei Wu, Chao-Jung Chen, Chung-Cheng Chiu and Tze-Chiuan Lai, “A Real-Time
Robust Lane Detection Approach for Autonomous Vehicle Environment,”
International Conference on Signal and Image Processing, SIP 2004, pp.518-523, Aug.
23-25, 2004, Honolulu, Hawaii, USA.

[9] Yen-Lin Chen, Chung-Cheng Chiu and Bing-Fei Wu, “Complex Document Image
Segmentation using Localized Histogram Analysis with Multi-Layer Matching and
Clustering,” *2004 IEEE International Conference on Systems, Man and Cybernetics,*
pp. 3063-3070, Oct. 10-13, 2004, Hague, Netherlands.

Submitted paper

- [1] Bing-Fei Wu, Chung-Cheng Chiu and Yen-Lin Chen, " Multi-Layer Segmentation of Complex Document Images," Submitted to *International Journal of Pattern Recognition and Artificial Intelligence*, Aug. 2003, Revised May 2004.
- [2] Bing-Fei Wu, Yen-Lin Chen and Chung-Cheng Chiu, "A Novel Region-Based Segmentation Method for Complex Document Image Analysis," Submitted to *International Journal of Computational Science and Engineering*, Mar. 2004.
- [3] Bing-Fei Wu, Shin-Ping Lin and Chung-Cheng Chiu, "Extracting Characters from Real Vehicle License Plates Out-of-doors," Submitted to *IEE Proceedings-Vision, Image, and Signal Processing*, Aug. 2004.
- [4] Bing-Fei Wu, Yen-Lin Chen and Chung-Cheng Chiu, "Efficient Implementation of Several Multilevel Thresholding Algorithms Using A Combinatorial Scheme," Submitted to *International Journal of Computers and Applications*, Aug. 2004.
- [5] Bing-Fei Wu, Yen-Lin Chen and Chung-Cheng Chiu, "A Discriminant Analysis Based Recursive Automatic Thresholding Approach for Image Segmentation," Submitted to *IEICE Transactions on Information and Systems*, Sep. 2004.

書籍著作

- 一、「JPEG 2000 壓縮編碼技術」，吳炳飛、胡益強、瞿忠正、蘇崇彥，全華科技圖書，2003.

發明專利

- 一、「彩色文件圖文分割方法」，吳炳飛、林文隆、瞿忠正，中華民國專利證書發明第一一三三〇一號。中華民國 89 年 7 月 21 日。
- 二、「文字與圖形交疊之彩色文件圖文分離方法」，吳炳飛、瞿忠正，中華民國專利證書發明第一四七四七三號。中華民國 91 年 4 月 18 日。
- 三、「玩具看護」，吳炳飛、陳昭榮、瞿忠正，中華民國專利證書新型第 M 二四二九五〇號。中華民國 93 年 9 月 1 日。

