

# 國立交通大學

## 電機與控制工程學系

### 博士論文

影像處理與電腦視覺技術應用於複雜文件影像分析、夜間駕駛輔助、以及視訊監控系統之研究

A Study of Image Processing and Computer Vision Techniques for Complex Document Image Analysis, Nighttime Driver Assistance, and Video Surveillance Systems

研究生：陳彥霖

指導教授：吳炳飛 教授

中華民國九十五年十二月

影像處理與電腦視覺技術應用於複雜文件影像分析、夜間駕駛輔助、  
以及視訊監控系統之研究

A Study of Image Processing and Computer Vision Techniques for  
Complex Document Image Analysis, Nighttime Driver Assistance, and  
Video Surveillance Systems

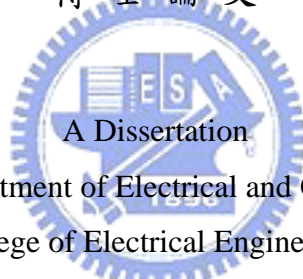
研究生：陳彥霖

Student : Yen-Lin Chen

指導教授：吳炳飛 教授

Advisor : Prof. 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  
National Chiao Tung University

in partial Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy  
in

Electrical and Control Engineering

December 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年十二月

# 影像處理與電腦視覺技術應用於複雜文件影像分析、夜間 駕駛輔助、以及視訊監控系統之研究

研究生：陳彥霖

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

## 中文摘要

影像處理與電腦視覺技術的發展，近十年來在各種應用領域的期刊文獻上，發表了許多針對不同目的所開發之系統，用於靜態影像切割、文件分析與辨識、智慧型運輸系統、視訊監控等許多應用上。本論文將針對這些應用需求，發展一系列以影像處理與電腦視覺技術為基礎的研究方法與應用系統。本論文主要分為六個章節，第一章我們對於影像處理與電腦視覺在各領域的應用作一簡要介紹。而第二章至第五章，則分別探討簡介本論文所提出之各個以影像處理與電腦視覺為基礎的研究方法與應用系統。

在第二章中，我們提出了一個快速且具高度可靠性的多重門檻值選擇機制，以針對含有多個物件的灰階影像，加以將其中內含之物件分離，以利於後續之處理與分析。這個機制包含了一個最佳化的分離度量測法則，可以確保所得之切割影像，具有統計特性上的最大分離度，以此獲得最佳的切割效果。目前現有的門檻值選取演算法，多是為了二值化切割而設計，少數具有多門檻值選取技術，其大多需要消耗相當多的運算能量，以及多是設計於在具有某些特定特性的影像上才具有較高可靠度。而這個研究提出了一個最佳化的分離度量測判定準則，據此所開發之自動多重門檻值選擇技術，可在多種不同特性的影像上，皆可以極低的運算時間，判定影像上的物件個數，並決定門檻值加以切割分離之，並經過多種實驗測試證明其高效率與可靠性。這個技術可應用於電腦與機器視覺相關之系統開發，以此技術將具意義之物件加以從影像分離後，以利其後續之分

析與辨識處理，如在文件影像分析與辨識、ITS 智慧型運輸系統之電腦視覺處理上之整合運用，皆需要將有意義之物件從影像中擷取出，以利進行進一步的分析處理工作。

我們在第三章則提出了一套多平面影像切割演算法，以將文字從彩色圖文交疊的複雜文件影像中完整萃取。由於在現今生活中常見的文件，因印刷排版技術的發展，使得多媒體複合文件的大量出現，此類文件之文字大多印刷在複雜的背景內涵下。將文字從文件影像中抽離是文件分析研究的重要一環，目前已經有許多學者在這個領域提出相關文字切割技術。然而，先前的技術大多無法解決複雜文件影像抽離文字的困難。這個技術可以解決許多由於文件影像背景日益複雜所衍生的相關問題。在實驗分析中，我們使用了書本封面、平面廣告和雜誌等進行處理，實驗結果證明本論文所提出的方法，能夠成功的將這些文件影像中的中英文文字字串加以成功萃取。這個研究的主要目的，是開發一套以區域性切割演算法為基礎的文件影像切割技術，針對彩色圖文交疊的複雜文件影像，將其中所包含的前景物件與背景物件分別分離，並再有一套文字萃取演算法，使其中之文字資訊能夠完整的加以萃取，即使他們都被印刷在緩慢變化或快速變化的高度複雜背景圖形中。

在第四章中，我們則提出了一個針對夜間行車駕駛輔助與車輛自動化駕駛需求的智慧型高速夜間車輛偵測與辨識系統。該系統透過 CCD 影像擷取設備，結合電腦視覺處理技術，以實現夜間車輛偵測、相對位置與距離判定、車輛標定與追蹤，並以此輔助駕駛獲得前方之交通狀況資訊。以此可以提供一個有效的機制，以自動操控車上的相關裝置設備，如運用於車輛頭燈的遠近光燈控制上，可以在偵測判斷前方車道的交通狀況時，自動將車輛頭燈之遠光燈與近光燈調整至最佳狀態，防止炫光而影響前方來車駕駛視線，避免因為遠光燈近距離照射造成目眩所困擾而導致之車禍危險性。並可以基於所偵測獲得之本車前方交通狀況，如本車與前方道路上所出現車輛之相對運動關係，以提供作為自動駕駛與自動巡航速度的上層控制機制。

第五章我們提出一個智慧型多通道錄影視訊監視控制系統，其對於影像壓縮速度的提升，在於達成多通道錄影即時視訊壓縮編碼的高標準要求，同時又能保持極高的壓縮影像品質與壓縮效率，加以為了達成現今軟體工程的主流，系統開發以實作微軟所提出之 ActiveX 系統元件模型完成，以利於多媒體應用、網際網路應用與快速應用軟體程式開發。再加上結合了網路伺服程式、影像擷取卡與 CCD 攝影機，發展成為一高效率的多錄影通道智慧型監控系統，並擁有高效能、低成本與功能強大的特質。最後，在第六章的部分，我們整理了本篇論文的結論與未來的研究展望。



# **A Study of Image Processing and Computer Vision Techniques for Complex Document Image Analysis, Nighttime Driver Assistance, and Video Surveillance Systems**

Student : Yen-Lin Chen

Advisor : Prof. Bing-Fei Wu

Department of Electrical and Control Engineering  
National Chiao Tung University

## **Abstract**




Image processing and computer vision are the studies of how computers can perceive and understand the interesting information about the world surrounding human beings by automatically extracting and analyzing observed images, image sets, or video sequences using theoretical and algorithmic computations. Object extraction and analysis is one of the important applications of image processing and computer vision. Among the applications of object extraction and analysis, document image analysis (DIA) is the one that provides many valuable applications in document analysis and understanding, such as optical character recognition, document retrieval, and compression. Vision-based techniques of driver assistance and autonomous vehicle navigation systems are emerging practical applications as well. It aims at detecting and recognition the vehicular objects in the road environment for driver assistance and autonomous vehicle guidance. As well as the security issues in modern life, digital video monitoring is also a promising application. In this dissertation, we will present several algorithmic, practical, and integrated methods and systems for the above-mentioned applications based on image processing and computer vision techniques.

Firstly, Chapter 2 presents an efficient automatic multilevel thresholding method for image segmentation. An effective criterion for measuring the separability of the homogenous objects in the image, based on discriminant analysis, has been introduced to automatically determine the number of thresholding levels to be performed. Then, by applying this discriminant criterion, the object regions with homogeneous illuminations in the image can be recursively and automatically thresholded into separate segmented images. This proposed method is fast and effective in analyzing and thresholding the histogram of the image. In order to conduct an equitable comparative performance evaluation of the proposed method with other thresholding methods, a combinatorial scheme is also introduced to properly reduce the computational complexity of performing multilevel thresholding.

In Chapter 3, we propose a new method, namely the multi-plane segmentation approach, for segmenting and extracting textual objects from various real-life complex document images. The proposed multi-plane segmentation approach first decomposes the document image into distinct object planes to extract and separate homogeneous objects including textual regions of interest, non-text objects such as graphics and pictures, and background textures. This proposed approach processes document images regionally and adaptively according to their respective local features. Hence detailed characteristics of the extracted textual objects, particularly small characters with thin strokes, as well as gradational illuminations of characters, can be well-preserved. Moreover, this way also allows background objects with uneven, gradational, and sharp variations in contrast, illumination, and texture to be handled easily and well.

Next, an effective method for detecting vehicles in front of the camera-assisted car during nighttime driving is presented in Chapter 4. This proposed method detects vehicles based on detecting and locating vehicle headlights and taillights by using techniques of image segmentation and pattern analysis. First, to effectively extract bright objects of interest, a fast bright object segmentation process based on automatic multilevel histogram thresholding is applied on the grabbed nighttime road-scene images. This automatic multilevel thresholding approach can provide robustness and adaptability for the detection system to be operated well

on various illuminated conditions at night. Then the extracted bright objects are processed by a rule-based connected-component analysis procedure, to identify the vehicles by locating and analyzing their vehicle light patterns, and estimate the distance between the detected vehicles and the camera-assisted car.

In Chapter 5, we present a wavelet-based approach to compressing video, with high speed, high image quality and high compression ratio. Using the sequential characteristics of surveillance images, this method applies the low-complexity zero-tree coding, which costs low memory, to develop an algorithm for encoding and decoding video, which significantly improves the speeds of compression and decompression and maintains images of high quality. Based on this low-complexity and low-memory-cost wavelet-based coding scheme and motion compression strategy, the proposed video codec achieves high vision quality, high compression speed and high compression ratio. Then the ActiveX COM component technique is also implemented and integrated with the proposed video codec to realize multimedia, internet applications and many other video-intensive applications. Furthermore, an intelligent surveillance system, which integrates the proposed wavelet-based video codec, computer peripherals and mobile communication, is also presented in this chapter. Finally, we give a brief conclusion and future works in Chapter 6.

## 誌 謝

回首這些年來的博士班研究過程，學生首先要誠摯的感謝恩師 吳炳飛教授。打從大三修習教授的線性控制系統課程，就深深為老師上課所展露的獨特風采與孜孜不倦的精神所吸引，從而加入了 CSSP 實驗室這個大家庭，展開了人生中最重要的一段學術研究訓練。因得力於恩師所給予我這樣的一個兼具深度與廣度、理論與實務的研究方向，使我得以一窺影像處理與電腦視覺研究領域的堂奧，給了我在這些年來的研究與學習的歷練中獲益匪淺。

感謝口試委員 蔡文祥校長、王聖智教授、曾定章教授、張志永教授與陸儀斌教授，給予了我在研究論文上許多相當寶貴的意見，從而讓本論文能夠更完整而嚴謹。此外，更讓我也瞭解到自己在進行研究時，常因過於著重於解決問題，以致過於著重 How，而時而忽略了對於研究課題本身的 Why 的思考的缺失，從而使我在對於自己往後所從事的研究，有了更深刻的了悟。此外，也很感謝 林源倍教授與黃有評教授，給予了我許許多多課業上與研究上的寶貴諮詢與幫助。

這些年來，相當的感謝重甫、忠哥、旭哥、暉哥學長們在我研究的過程中的提攜與照顧，也感謝從碩士班直到現在的嘴泡好哥們子偉同學，以及曾與我一起合作研究過的堯俊、元馨、至明學弟妹，這段一起在 CSSP 實驗室並肩打拼的日子是我最好的回憶。也要感謝所有與我一起在實驗室合作與相處的實驗室所有學長、同學、和學弟妹們，在這六年多的日子裡，和大家一起在實驗室裡共同的生活點滴、研究上的討論、嘴泡、與豪洩，組隊「團戰」的革命情感，還有因為睡過頭而偷偷低著頭閃進 Group Meeting... 等，這些酸甜苦辣的經驗，是我一生最難忘的回憶。

另外，也要感謝中華扶輪教育基金會在我博士修業期間所提供的豐厚獎學金，使我能夠毫無後顧之憂的專心完成我的研究。

最後，要感謝爸爸和媽媽一路上對我的支持與鼓勵，還有對我要「所費」時總是有求必應，使我的研究過程一直無後顧之憂的一往無前，而順利的完成學業。

謹以此文感謝這段期間許許多多曾經幫助提攜過我的人，謝謝你們。

陳彥霖 於交通大學電機與控制工程學系 CSSP 實驗室 2007/01/17

# Table of Contents

<b>Chapter 1. Introduction.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 The Proposed Approaches.....	6
1.2.1 Multi-level thresholding approaches for image segmentation.....	6
1.2.2 A multi-plane segmentation approach for text extraction in complex document images .....	7
1.2.3 Vision-based nighttime vehicle detection for driver assistance.....	8
1.2.4 Real-time wavelet-based video compression approach for video surveillance .....	9
1.3 Organization.....	10
<b>Chapter 2. Multi-level Thresholding Approaches for Image Segmentation.....</b>	<b>11</b>
2.1 Introduction.....	11
2.2 Review of Conventional Image Thresholding Criterion Functions.....	16
2.2.1 The between-class variance thresholding criterion.....	16
2.2.2 The maximum entropy thresholding criterion .....	19
2.2.3 The minimum error thresholding criterion .....	20
2.3 The Proposed Fast Combinatorial Scheme for Efficient Selection of Multiple Thresholds.....	21
2.3.1 The Combinatorial Search Scheme of Optimal Threshold Set Selection	21
2.3.2 Fast Implementation of Criterion Functions for Multilevel Thresholding Methods	27
2.4 The Proposed Automatic Multilevel Thresholding Method .....	33
2.5 Experimental Results .....	38
<b>Chapter 3. Multi-plane Segmentation Approach for Complex Document Images.....</b>	<b>53</b>
3.1 Introduction.....	54
3.2 Localized Histogram Multilevel Thresholding.....	61
3.3 Multi-plane Region Matching and Assembling Process.....	71
3.3.1 Initial Plane Selection Phase.....	72
3.3.2 Matching Phase.....	74
3.3.3 Plane Construction Phase.....	81
3.3.4 Overall Process .....	85
3.4 Text Extraction.....	88
3.5 Experimental Results .....	97
<b>Chapter 4. Nighttime Vehicle Detection for Driver Assistance .....</b>	<b>114</b>



4.1	Introduction.....	114
4.2	Bright Object Extraction .....	118
4.3	Spatial Clustering Process for Bright Objects .....	121
4.4	Rule-Based Vehicle Identification .....	126
4.5	Vehicle Distance and Position Estimation .....	127
4.6	Vehicle Tracking Process .....	131
4.7	Experimental Results .....	134
4.7.1	Implementation .....	134
4.7.2	Performance Evaluation.....	135
<b>Chapter 5. Real-Time Wavelet-Based Video Compression Approach to Video Surveillance Systems.....</b>		<b>139</b>
5.1	Introduction.....	139
5.2	The Proposed Fast Wavelet-Based Video Compression Technique.....	142
5.2.1	Forward/Inverse Component Transform.....	144
5.2.2	Forward/Inverse DWT .....	145
5.2.3	Quantization/De-quantization.....	147
5.2.4	Low-Complexity and Low-Memory Zero-tree Entropy Coder (LLZC Encoder/Decoder) .....	147
5.2.5	Inter-Frame Difference Extraction / Difference Compensation and Bidirectional Frame Interpolation.....	151
5.3	Experimental Results .....	153
5.4	Video Codec Software Component.....	159
5.5	Implementation of an Intelligent Video Surveillance System .....	162
<b>Chapter 6. Conclusions and Future Works .....</b>		<b>165</b>
6.1	Multi-level thresholding approaches for image segmentation.....	165
6.2	A multi-plane segmentation approach for text extraction in complex document images	166
6.3	Vision-based nighttime vehicle detection for driver assistance.....	168
6.4	Real-time wavelet-based video compression approach for video surveillance .	169
<b>References.....</b>		<b>170</b>

# List of Figures

Figure 2.1. Conventional search procedure of optimal $\psi(T)$ .....	22
Figure 2.2. The pseudo code of the <i>Twiddle</i> procedure .....	26
Figure 2.3. Improved search procedure of optimal $\psi(T)$ using the combinatorial scheme ..	26
Figure 2.4. Result of segmenting the test image 1, “Road” (image size=720×480).....	42
Figure 2.5. Result of segmenting the test image 2, “Lena” (image size=512×512) .....	46
Figure 2.6. Result of segmenting the test image 3, “Advertisement” .....	51
Figure 3.1. Block diagram of the proposed multi-plane segmentation approach .....	60
Figure 3.2. Example of the results by the localized multilevel thresholding procedure.....	70
Figure 3.3. An example of the test image, “Calibre”, and the object planes obtained by the multi-plane segmentation (image size = 1929 x 1019).....	87
Figure 3.4. Examples of the text location and extraction process .....	96
Figure 3.5. Representative color images of Figure 3.3(a) after performing Jain and Yu's method.....	97
Figure 3.6. Text extraction results of Fig. 3(a) by Jain and Yu’s method and Pietikainen and Okun’s method.....	98
Figure 3.7. Original images of the test images 2 and 3.....	100
Figure 3.8. Decomposed object planes of Figure 3.7(a) after performing the proposed multi-plane segmentation.....	101
Figure 3.9. Representative color images of Figure 3.7(a) after performing Jain and Yu's method.....	102
Figure 3.10. Text extraction results of Figure 3.7(a) by the proposed approach, Jain and Yu’s method, and Pietikainen and Okun’s method. ....	103
Figure 3.11. Decomposed object planes of Figure 3.7(b) after performing the proposed multi-plane segmentation.....	104
Figure 3.12. Representative color images of Figure 3.7(b) after performing Jain and Yu's method.....	105
Figure 3.13. Text extraction results of Figure 3.7(b) by the proposed approach, Jain and Yu’s method, and Pietikainen and Okun’s method. ....	106

Figure 3.14. Original images of the test images 4 - 6 .....	106
Figure 3.15. Text extraction results of Figure 3.14(a) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method. ....	108
Figure 3.16. Text extraction results of Figure 3.14(b) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method. ....	108
Figure 3.17. Text extraction results of Figure 3.14(c) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method. ....	109
Figure 3.18. Results of test image 7 (size: $1829 \times 2330$ ) .....	111
Figure 3.19. Results of test image 8 (size: $3147 \times 4536$ ) .....	111
Figure 3.20. Results of test image 9 (size: $1859 \times 2437$ ) .....	111
Figure 3.21. Results of test image 10 (size: $1344 \times 1792$ ) .....	112
Figure 3.22. Results of test image 11 (size: $2309 \times 2829$ ).....	112
Figure 3.23. Results of test image 12 (size: $2469 \times 3535$ ) .....	112
Figure 4.1. Block diagram of the proposed method.....	117
Figure 4.2. An example of nighttime road environment.....	119
Figure 4.3. Bright object plane extracted from Figure 4.2 after performing the bright object segmentation process .....	120
Figure 4.4. The processing area determined by the virtual horizon and the bright components of interest .....	123
Figure 4.5. The spatial clustering process of bright components .....	125
Figure 4.6. The illustration of the lateral positions of the detected vehicle bodies in the image coordinates .....	130
Figure 4.7. The experimental camera-assisted car – TAIWAN iTS-1 .....	134
Figure 4.8. The vision system mounted in the experimental car .....	135
Figure 4.9. Result of vehicle detection on the nighttime road scene with one oncoming vehicle.....	136
Figure 4.10. Result of vehicle detection on the nighttime road scene with both oncoming and preceding vehicles.....	136
Figure 4.11. Result of vehicle detection on the nighttime road scene comprised of vehicles and many other non-vehicle lights.....	137
Figure 5.1. Structure of the coding and decoding flow of the proposed method.....	143

Figure 5.2. Illustration of the 4:2:0 sampling process .....	144
Figure 5.3. Examples of Wavelet Transform .....	147
Figure 5.4. Tree structure of the distribution of wavelet coefficients .....	149
Figure 5.5. Coding procedure of LLZC .....	149
Figure 5.6. Frame sequence encoding in Fast Mode .....	152
Figure 5.7. Frame sequence encoding in Turbo Mode.....	152
Figure 5.8. Comparative results of JPEG and the proposed method .....	154
Figure 5.9. Results of compressing the standard testing clip – “Akiyo” (CIF format, 352*288) .....	156
Figure 5.10. Results of compressing the standard testing clip – “Foreman” (QCIF format, 176*144) .....	157
Figure 5.11. Comparison of compression speeds between MPEG-4 method and the proposed method.....	159
Figure 5.12. Video codec component and its applications.....	160
Figure 5.13. Visual Basic environment for developing surveillance applications.....	161
Figure 5.14. Example of web-surveillance application .....	161
Figure 5.15. Implementation of Intelligent surveillance system.....	163
Figure 5.16. Mobile carrier with a CCD camcorder .....	164
Figure 5.17. The controlling interface of the monitoring user.....	164

## List of Tables

Table 2.1. The values of the zeroth moment $Z(t_a, t_b)$ of intervals $t_a - t_b$ on the histogram ...	30
Table 2.2. The values of the first moment $F(t_a, t_b)$ of intervals $t_a - t_b$ on the histogram.....	30
Table 2.3. The values of the second moment $S(t_a, t_b)$ of intervals $t_a - t_b$ on the histogram...	31
Table 2.4. The values of the a priori entropy $E(t_a, t_b)$ of intervals $t_a - t_b$ on the histogram...	31
Table 2.5. Experimental data of performing thresholding on Figure 2.4(a), “Road”, by our proposed automatic multilevel thresholding method (AMT), Between-class variance method (BCV), Entropy method (ENT) and Minimum Error method (ME) .....	40
Table 2.6. Experimental data of performing thresholding on Figure 2.5(a), “Lena”, by our automatic multilevel thresholding method (AMT), Between-class variance method (BCV), Entropy method (ENT), and Minimum Error method (ME) .....	44
Table 2.7. Experimental data of performing thresholding on Figure 2.6(a), “Advertisement”, by our automatic multilevel thresholding method (AMT), Between-class variance method (BCV), Entropy method (ENT), and Minimum Error method (ME) .....	48
Table 3.1. Experimental data of Jain and Yu’s method and our proposed approach.....	109
Table 5.1. Classes of coefficients.....	151
Table 5.2. Comparisons of the need of storage space between the JPEG-Based system and the proposed method under similar visual quality (QVGA format, 320*240) .....	155
Table 5.3. Comparisons of performance between proposed method and MPEG-4 (Akiyo, CIF format, 352*288).....	158
Table 5.4. Comparisons of performance between proposed method and MPEG-4 (Foreman, QCIF format, 176*144) .....	158



# Chapter 1. INTRODUCTION

Image processing and computer vision are the studies of how computers can perceive and understand the interesting information about the world surrounding human beings by automatically extracting and analyzing observed images, image sets, or video sequences using theoretical and algorithmic computations [1]-[3]. Object extraction and analysis is one of the important applications of image processing and computer vision. Among the applications of object extraction and analysis, document image analysis (DIA) is the one that provides many valuable applications in document analysis and understanding, such as optical character recognition, document retrieval, and compression [4][5]. Vision-based techniques of driver assistance and autonomous vehicle navigation systems are emerging practical applications as well. It aims at detecting and recognition the vehicular objects in the road environment for driver assistance and autonomous vehicle guidance [6]-[9]. As well as the security issues in modern life, digital video monitoring is also a promising application. In this dissertation, we will present several algorithmic, practical, and integrated methods and systems for the above-mentioned applications based on image processing and computer vision techniques.

## 1.1 Motivation

Extraction of objects from observed images using image thresholding techniques is useful for in the early processing stages of a vision system [10][11]. To-date, many researchers have developed valuable thresholding techniques [12]-[26] for applications that include image segmentation, pattern recognition and document analysis, among others. The conventional concept of most of these methods is that the thresholding is carried out as a

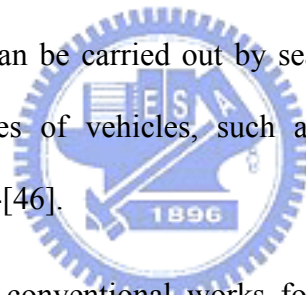
simple classification procedure that the pixels of the image are assigned to two classes, foreground object pixels and background pixels. Hence most of them were developed for effectively adoption on bi-level thresholding process. The surveys and comparative performance testing studies of these methods was presented in the remarkable works of Sahoo *et al.* [10] and Lee *et al.* [11]. Besides, in Trier and Jain's work [27][28], a goal-directed evaluation methodology has been proposed for performance testing on the thresholding methods based on judgment of the recognition performance of conducting OCR process on the binarization results obtained by these methods. The conventional thresholding techniques are all based on finding the threshold value which achieves the optimal condition of the criterion functions. Indeed, they are effective in bi-level thresholding. However, when the number of desired thresholds increases, the computation needed to obtain the optimal threshold values is substantially increased and the search to achieve the optimal value of the criterion functions is particularly exhaustive. Another problem associated with the conventional methods is that the number of segments, into which the image should be segmented, cannot be suitably and automatically determined. Thus, we intend to develop a computationally fast and effective automatic multilevel thresholding approach to overcome the above-mentioned image segmentation issues associated with the conventional methods.

For document image analysis, the interesting objects in a document image are textual objects. Extracting textual objects from document images provides many useful applications in document analysis and understanding, such as optical character recognition, document retrieval, and compression [4][5]. To-date, many techniques were presented for extracting textual objects from monochromatic document images [29]-[32]. In recent years, owing to advances in multimedia publishing and printing technology have led to an increasing number of real-life documents in which stylistic character strings are printed with pictorial, textured, and decorated objects and colorful, varied background components. However, most of

conventional approaches cannot work well for extracting textual objects from real-life complex document images. Compared to monochromatic document images, text extraction in complex document images brings many difficulties associated with the complexity of background images, variety and shading of character illuminations, superimposing characters with illustrations and pictures, as well as other decorated background components. As a result, there is an increasing demand for a system that is able to read and extract the textual information printed on pictorial and textured regions in both colored images as well as monochromatic main text regions.

Since most textual objects show sharp and distinctive edge features, methods based on edge information [33]-[36] have been developed. Such methods utilize an edge detection operator to extract the edge features of textual objects, and then use these features to extract characters from document images. Such edge-based methods are capable of extracting textual objects in different homogeneous illuminations from graphic backgrounds. However, when the characters are adjoined or touched with graphical objects, texture patterns, or backgrounds with sharply varying contours, edge-feature vectors of non-text objects with similar characteristics may also be identified as text, and thus the characters in extracted textual regions are blurred by those non-text objects. Several conventional color-segmentation-based methods for text extraction from color document images have been proposed [37]-[41]. These methods utilize color clustering or quantization approaches for determining the prototype colors of documents so as to facilitate the detection of textual objects in these separated color planes. However, most of these methods have difficulties in extracting characters which are embedded in complex backgrounds or that touch other graphical objects. This is because the prototype colors are determined in a global view, so that appropriate prototype colors cannot be easily selected for distinguishing textual objects from those touched graphical objects and complex backgrounds without sufficient contrast.

For vision-based systems for driver assistance and autonomous vehicle guidance, many researchers have also developed valuable techniques for recognizing interesting vehicles and obstacles from images of road environments outside the car [6]-[9], to facilitate applications on the camera-assisted system that assists drivers in understanding possible hazards on the road, and automatically controlling the apparatus of vehicles, such as headlights, windshield wipers, etc. A vision-based vehicle and obstacle detection system is aiming at identification of vehicles, obstacles, traffic signs and other patterns on the road from grabbed image sequences by means of image processing and pattern recognition techniques. Until recently, researchers in this field still open new questions and concepts [42][43]. By adopting different concepts and definitions on interesting objects on the road, different techniques are applied on the grabbed image sequences to detect them as vehicles or obstacles. For locating vehicles in an image sequence, the task can be carried out by searching for specific patterns on the images based on typical features of vehicles, such as shape, symmetrization, or their surrounding bounding boxes [44]-[46].



Until recently, most of the conventional works focused on detecting vehicles under daytime road environments. However, under bad-illuminated conditions in nighttime road environments, those obvious features of vehicles which are effective for detecting vehicles in daytime become invalid in nighttime road environments. Thus, most of the above-mentioned conventional techniques cannot work well under such nighttime road environments. At night, as well as under dark illuminated condition in general, the only visual features of vehicles are their headlights and taillights. Headlights and taillights are visible if a vehicle lies in the visible range of the CCD camera mounted on a camera-assisted car. However, there are also many other illuminant sources coexisted with the vehicle lights in nighttime road environments, such as street lamps, traffic lights, and road reflector plates on ground. These non-vehicle illuminant sources cause many difficulties for detecting actual vehicles in

nighttime road scenes.

As for the applications on video surveillance systems, it is an emerging application of video compression and communication for security issues in modern life. However, conventional monitoring systems are mostly analog systems, which exploit many tapes and human effort, to replace the tapes frequently. The recording time and image quality of systems cannot compete with those of digital monitoring systems. In digital monitoring systems, transform coding techniques are the most popular for video recording applications. At the beginning of the development of this field, DCT-based (Discrete Cosine Transform) coding techniques were commonly used, and have since become an element of the JPEG image compression standard. Accordingly, its application can be seen in many electronic devices today. Over recent years, researchers have demonstrated that DWT-based (Discrete Wavelet Transform) transform coding ([47]-[50]) outperforms DCT-based methods. Hence, newly emerging image compression methods such as the video compression method standard MPEG-4 [51][52] and the still image compression standard JPEG2000 are using DWT-based methods [53][54]. Since multiple CCD camera systems are continuously heavily loaded with sequences of images, so the speed of image compression is critical in such systems. Presently, DWT-based compression techniques suffer from high computational complexity, and so cannot support multi-channel video recording with a high frame rate. A new, highly efficient DWT-based technique, which yields images of high visual quality, is a significant demand for such application.



## **1.2 The Proposed Approaches**

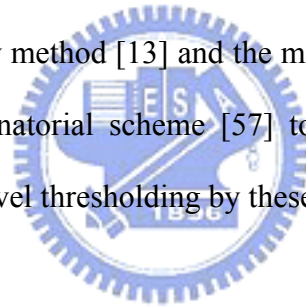
In this dissertation, we will present several algorithmic, practical, or integrated methods and systems based on image processing and computer vision techniques to deal with the above-mentioned issues, including multilevel thresholding techniques for low-level image segmentation, text extraction for complex document image analysis, nighttime vehicle detection for driver assistance, and multi-channel video surveillance. They are briefly introduced in the following sub-sections.

### **1.2.1 Multi-level thresholding approaches for image segmentation**

For segmenting objects from a given image, different objects with homogeneous illuminations must be separated into different segmented images. However, most of the conventional thresholding techniques [12]-[26] were developed for effectively applying on bi-level thresholding cases, and when the number of desired thresholds increases, the computation costs needed to obtain the optimal threshold values is substantially increased. Another problem associated with these conventional methods is that the number of segments, into which the image should be segmented, cannot be suitably and automatically determined. For this purpose, the discriminant criterion, for measuring separability among the segmented images with different objects, is described in this section. By evaluating the separability criterion, the number of objects, into which the image should be segmented, can be automatically determined. Hence, an automatic multilevel thresholding method, based on this criterion, will be presented in this dissertation.

The concept of using discriminant analysis for classification problems was first introduced by Fisher [55] and was applied on image thresholding by Otsu [12]. It is attractive for the simplicity in computation, with which it measures the separability among segmented

images. In Chapter 2, we will analyze the properties of discriminant analysis and then propose an automatic multilevel thresholding method [56]. The proposed method applies the discriminant criterion for analyzing the separability among the gray levels in the image to automatically determine the optimal number of thresholded classes that the gray levels should be partitioned. A fast recursive selection strategy is also introduced for determining the optimal thresholds to segment objects of interest in complex images into separate thresholded images in a computationally fast way. Each threshold determined by this recursive selection strategy is ensured to achieve the maximum separation on the resultant thresholded images, and hence satisfactory thresholded results can be accomplished by means of the smallest number of thresholding levels. To conduct an equitable performance evaluation of the proposed method, when compared to other criterion-based methods (i.e. the between-class variance method [12], the entropy method [13] and the minimum error method [14]), we also will introduce a efficient combinatorial scheme [57] to properly reduce the computation complexity of performing multilevel thresholding by these methods.



### **1.2.2 A multi-plane segmentation approach for text extraction in complex document images**

For extracting textual objects from complex document images involves several difficulties. These difficulties arise from the following properties of complex documents: 1) Character strings in complex document images may have different illuminations, sizes, and font styles, and are overlapped with various background objects with uneven, gradational, and sharp variations in contrast, illumination, and texture, such as illustrations, photographs, pictures or other background textures. 2) These documents may comprise small characters with very thin strokes as well as large characters with thick strokes, and may be influenced by image shading.

Hence, we will propose effective region-based approaches for extracting textual objects from these complex document images [58]-[62], and resolving the above issues associated with the complexity of their backgrounds. The document image is processed by the proposed multi-plane segmentation technique to decompose it into separate object planes. The proposed multi-plane segmentation technique comprises two stages: automatic localized histogram multilevel thresholding, and multi-plane region matching and assembling processing. After the multi-plane segmentation technique has been carried out, homogeneous objects including textual blocks, other non-text objects, and background textures are separated into individual object planes. The text extraction process is then performed on the resultant planes to detect and extract textual objects with different characteristics in the respective planes. The document image is processed regionally and adaptively according to local features by the proposed method. This allows detailed characteristics of the extracted textual objects to be well-preserved, especially the small characters with thin strokes, as well as the gradational illuminations of characters. This also allows for characters adjoined or touched with graphical objects and backgrounds with uneven, gradational, and sharp variations in contrast, illumination, and texture to be handled easily and well.

### **1.2.3 Vision-based nighttime vehicle detection for driver assistance**

For the issues of nighttime driver assistance and the development of autonomous camera-assisted vehicles, an efficient technique for effectively detection and recognition of moving vehicles in nighttime road-scene image sequences is practically a necessary demand. Besides, this way provides beneficial information for the driver to perceive surrounding traffic conditions outside the vehicle during nighttime driving, and can also be applied to a versatile control scheme for the apparatus of vehicles. For example, the use of high-beam and low-beam states of headlights can be intelligently controlled according to the detection results

of presence of oncoming and preceding vehicles, and thus many hazards during nighttime driving, such as headlight dazzler, can be efficiently prevented.

Therefore, we will present an effective nighttime vehicle detection method [63]-[65] for identifying vehicles by locating and analyzing their headlights and taillights. This proposed method comprises of the following processing stages. First, a fast bright object segmentation process based on automatic multilevel histogram thresholding is performed to extract pixels of the bright objects from the grabbed image sequences of nighttime road scenes. The advantage of this automatic multilevel thresholding approach is its robustness and adaptability for dealing with various illuminated conditions at night. Then a connected-component analysis procedure is applied on the bright pixels obtained by the previous bright object segmentation stage, to locate the connected-components of these bright objects. These bright components are then grouped by a projection-based spatial clustering process to obtain potential pairing headlights of oncoming vehicles, and taillights of preceding vehicles. Accordingly, a set of identification rules are applied on each group of bright objects to determine whether it represents an actual vehicle. Finally, the distance between each of the detected vehicles and the camera-assisted car can be estimated and reported.

#### **1.2.4 Real-time wavelet-based video compression approach for video surveillance**

For the purpose of developing a digital surveillance system fulfilling the requirements of real-time multi-channel video compression, and ensuring the high quality of restored images and the efficiency of compression and decompression of images, we will present a real-time wavelet-based video compression technique and an intelligent multi-channel surveillance system [66]. Based on the low-complexity and low-memory-cost wavelet-based coding

scheme and motion compression strategy, the proposed video codec achieves high vision quality, high compression speed and high compression ratio. Then the ActiveX COM component technique is also implemented and integrated with the proposed video codec to realize multimedia, internet applications and many other video-intensive applications. Furthermore, an intelligent surveillance system, which integrates the proposed wavelet-based video codec, computer peripherals and mobile communication, is also developed in this study. Therefore, the future e-Home with controlled home electronics, managed video/audio systems and home security will be realized.

### **1.3 Organization**

The rest of this dissertation is organized as follows. Chapter 2 presents an automatic multilevel thresholding method for image segmentation. A region-based segmentation approach for text extraction from complex document images will be proposed in Chapter 3. In Chapter 4, we will propose a vision system to detect and recognize of vehicles for nighttime driver assistance. A real-time wavelet-based video codec for intelligent multi-channel monitoring system will be presented in Chapter 5. Finally, some conclusions and future research perspectives will be stated in Chapter 6.



## **Chapter 2. MULTI-LEVEL THRESHOLDING APPROACHES FOR IMAGE SEGMENTATION**

In this chapter, we will present a combinatorial scheme for reducing the computation timings of fixed-level multilevel thresholding process, and an efficient automatic multilevel thresholding method for image segmentation. As for fixed-level multilevel thresholding, by applying the proposed combinatorial scheme on conventional criterion-based multilevel thresholding, not only the redundant evaluation of threshold sets can be effectively avoided, but the computation cost for each evaluation of each potential threshold set can also be substantially suppressed, and thereby the computation timings for obtaining the optimal set of thresholds can be significantly reduced. Besides, this proposed combinatorial scheme can also achieve the parameterization of the desired number of thresholds. For automatic multilevel thresholding, an effective criterion for measuring the separability of the homogenous objects in the image, based on discriminant analysis, has been introduced to automatically determine the number of thresholding levels to be performed. Then, by applying this discriminant criterion, the object regions with homogeneous illuminations in the image can be recursively and automatically thresholded into separate segmented images. Both the two proposed multilevel approaches are fast and effective in analyzing and thresholding the histogram of the image.

### **2.1 Introduction**

In image processing, objects must usually be segmented from an image in order to facilitate further processing. Accordingly, many researchers have developed valuable thresholding techniques [12]-[26] for applications that include image segmentation, pattern

recognition and document analysis, among others. The conventional concept of most of these methods is that the thresholding is carried out as a simple classification procedure that the pixels of the image are assigned to two classes, foreground pixels and background pixels. Hence most of them were developed for effectively adoption on bi-level thresholding process. Most of these methods were developed for adoption on bi-level thresholding. The surveys and comparative performance testing studies of these methods was presented in the remarkable works of Sahoo *et al.* [10] and Lee *et al.* [11]. Besides, in Trier and Jain's work [27][28], a goal-directed evaluation methodology has been proposed for performance testing on the thresholding methods based on judgment of the recognition performance of conducting OCR process on the binarization results obtained by these methods.

The concept of using discriminant analysis for classification problems was first introduced by Fisher [55] and was firstly applied on the gray-level histogram distributions for image thresholding by Otsu [12]. This method is carried out by maximizing the separability of the resultant thresholded classes using the between-class variance criterion associated with them. It is attractive for the simplicity in computation, with which it measures the separability among segmented images. Besides, from the alternative viewpoint of gray-level histogram distributions, Kittler and Illingworth [14] developed an approach that based on the assumption that the probability distributions of gray levels of objects in an image (i.e. foreground object and background) are Gaussianly distributed. Hence they proposed the minimum error thresholding method to determine the optimal threshold which minimizes the error rate of the resultant thresholded classes with the desired mixtures of Gaussian distributions. Selecting the optimal threshold using the information theoretic concept is another master stream of thresholding methods. Kapur *et al.* [13] developed a method using the concept of the maximum entropy principle of a histogram in order to determine a suitable threshold. This method is performed by separating the histogram of gray level probabilities

into two distributions, where one is associated with the foreground, and the other one with the background of the image. The entropy of the two distributions is then combined, and the gray value with the maximal combined entropy is selected as the threshold value. In the maximum entropic correlation method proposed by Chang *et al.* [15], the distributions of the object and background classes are formulated by a entropic correlation criterion function, and the optimal threshold is determined by maximizing the entropic correlation criterion between the two distributions. This method can provide relatively simplifier computation cost than that of Kapur *et al.*'s maximum entropy method. Similar to the concept of the above-mentioned two entropic methods, Sahoo *et al.* [16] proposed a threshold selection method using an alternative formulated entropic criterion function using Renyi's entropy. This entropic criterion function gives a generalization formulation form including the maximum entropy criterion and the maximum entropic correlation criterion, and can yield a more effective threshold for the image. However, this criterion function costs much more computation timing for determining the optimal threshold than those of the above-mentioned two entropic criterion functions.

Recently, based on the concept of the fuzzy set theory integrated with the maximum entropy principle, several fuzzy entropic thresholding methods are developed [17]-[19]. They are performed by selecting the optimal threshold which maximizes the fuzzy entropy of the distributions of the object and background classes. Cheng *et al.* [18] applied the concept of fuzzy set theory into the principle of maximizing the objective entropy function for determining the optimal threshold. Besides, a unified formulation for the criterion functions used in the methods of Otsu [12], Kittler and Illingworth [14], and Huang and Wang [17] was introduced in Yan's work [20]. Most of these above-mentioned criterion-based thresholding methods are mostly based on finding the threshold value which achieves the optimal condition of the criterion functions. Indeed, they are effective in bi-level thresholding.

However, when the number of desired thresholds increases, the computation needed to obtain the optimal threshold values is substantially increased, and the search for achieving the optimal value of the criterion functions becomes particularly exhaustive.

To perform multilevel thresholding in a more computationally frugal way, Tsai [21] developed a method using moment-preserving to threshold an image into a specific number of object images. This approach is fast and convincing when the number of thresholds is less than or equal to four. However, some complex numerical methods may be required when the number of thresholds exceeds four. In Reddi *et al.*'s work [22], an efficient implementation scheme for performing multilevel thresholding of the between-class variance criterion in a more computationally frugal way was presented. However, this method can be adequately performed only on exactly three or fewer thresholding levels. For segmenting dark characters in document images with multiple illuminated objects, Chereit *et al.* [23] extended Otsu's approach by repeatedly performing bi-level thresholding to segment the bright object away at each recursion, and leaving the darkest character-like objects in the given resultant image. This method is effective on extracting dark characters from document images with bright homogeneous background, especially for bank checks. Tsai [24] presented a method using Gaussian kernel smoothing to repeatedly perform on the histogram of the image until the histogram being divided into desired number of classes. When the desired number of classes is much lower than the number of peaks in the original histogram, the computation time to find the solutions of threshold values is expensive. Fleury *et al.* [26] proposed a running entropic method that can reduce the computation cost for performing multilevel thresholding of the maximum entropy criterion [13]. This method has also been implemented in a parallelization computation form on a parallel workstation. However, this method cannot adapt to the changes of different desired number of thresholds, that is, once the number of desired thresholds is changed, its computational implementation must be re-programmed for

performing on the new desired number of thresholding levels. Besides, the main problem associated with the aforementioned methods is that the number of segments, into which the gray-level image should be segmented, cannot be automatically determined.

In this study, we propose two approaches for fast and automatic implementation of multi-level thresholding process. First, we present a combinatorial scheme to reduce the computation cost of determining the optimal threshold values in multilevel thresholding. This proposed scheme comprises of two elements – a combinatorial search scheme for efficiently and sequentially producing sets of potential threshold values to be evaluated, as well as a fast implementation scheme of the criterion functions for speedy evaluation of the potential threshold sets. This scheme effectively avoids evaluating redundant threshold sets, substantially suppresses the computation cost for each evaluation of each potential threshold set, and thereby significantly reduces the computation timings of obtaining the optimal set of threshold values. Moreover, the proposed scheme also provides the parameterization for the number of desired thresholds, and hence the implementation of the proposed scheme can provide adaptation to be performed on any different number of thresholding levels. We will apply the proposed scheme on multilevel thresholding using the criterion functions of the three most well-known methods - Otsu's between-class variance criterion [12], Kapur *et al's* maximum entropy criterion [13], and the Kittler and Illingworth's minimum error thresholding criterion [14].

Furthermore, based on the extension of this concept, we analyze the properties of discriminant analysis and then propose an automatic multilevel thresholding method. This proposed method applies the discriminant criterion for analyzing the separability among the gray levels in the image to automatically determine the optimal number of thresholded classes that the gray levels should be partitioned. A fast recursive selection strategy is also introduced for determining the optimal thresholds to segment objects of interest in complex

images into separate thresholded images in a computationally fast way. Each threshold determined by this recursive selection strategy is ensured to achieve the maximum separation on the resultant thresholded images, and hence satisfactory thresholded results can be accomplished by means of the smallest number of thresholding levels. To conduct an equitable performance evaluation of the proposed method, when compared to other criterion-based methods which are improved by the combinatorial scheme as mentioned before. The results of these three modified criterion-based methods with combinatorial scheme, and this proposed method are compared using several images with different characteristics and complexity. The experimental results demonstrate the feasibility and computational efficiency of these two proposed methods on multilevel thresholding.

## **2.2 Review of Conventional Image Thresholding Criterion Functions**

In this section, we describe three well-known criterion functions utilized in the thresholding methods - the between-class variance criterion [12], the maximum entropy criterion [13], and the minimum error criterion [14]. The multilevel thresholding computation forms of these criterion functions are also in turn described.

### **2.2.1 The between-class variance thresholding criterion**

The classical method of discriminant analysis, to classify two class cases, was first introduced by Fisher [27], and extended to multiple class cases by Rao [67]. This method sought the set of variables which maximized the ratio of the between-class variance and within-class variance of the resultant classes. These discriminant criteria, used by image thresholding, to extract foreground objects from the background, were presented by Otsu [12]. In his work, three possible discriminant criterion functions, based on within-class,

between-class and total variance ratios were presented, all of which were equivalent, for evaluation of an optimal thresholding process. Among the above three criterion functions, the between-class variance is the simplest one to compute. This method is performed by finding an optimal threshold, for which the between-class variance between dark and bright regions of the image is maximized.

Let  $f_i$  denote the observed occurrence frequencies (histogram) of pixels in a given image  $I$ , with a given gray level  $i$ , and  $N$  denotes the total amount of pixels in the image  $I$ , and can be given by  $N = f_0 + f_1 + \dots + f_{L-1}$ , where  $L$  is the number of gray values in the histogram. Hence, the normalized probability  $P_i$  of one pixel having a given gray level  $i$  can be denoted as,

$$P_i = \frac{f_i}{N}, \quad (2.1)$$

$$\text{where } P_i \geq 0, \quad \sum_{i=0}^{L-1} P_i = 1 \quad (2.2)$$

For the bi-level thresholding process, an image  $I$  is classified into two classes,  $C_0$  and  $C_1$  (e.g. foreground and background) using an optimal gray-level threshold  $t$ , where  $C_0 = \{0, 1, \dots, t-1\}$  and  $C_1 = \{t, t+1, \dots, L-1\}$ . This method finds the optimal threshold for which the between-class variance is maximized. The between-class variance, denoted by  $v_{bc}$ , is an effective criterion for evaluating the separability of the two classes, and is defined as,

$$v_{bc}(t) = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 = w_0 w_1 (\mu_1 - \mu_0)^2 \quad (2.3)$$

The within-class variance, denoted by  $v_{wc}$ , of all segmented classes of pixels is computed as,

$$v_{wc}(t) = w_0 \sigma_0^2 + w_1 \sigma_1^2 \quad (2.4)$$

where  $w_0$  and  $w_1$  denotes the cumulative probability mass function of the classes  $C_0$  and  $C_1$ ,



respectively;  $\mu_0, \mu_1, \sigma_0,$  and  $\sigma_1$  denotes the mean and the standard deviation of pixels in class  $C_0$  and  $C_1$ , respectively. They are defined as

$$w_0 = \sum_{i=0}^{t-1} P_i, \text{ and } w_1 = \sum_{i=t}^{L-1} P_i = 1 - w_0 \quad (2.5)$$

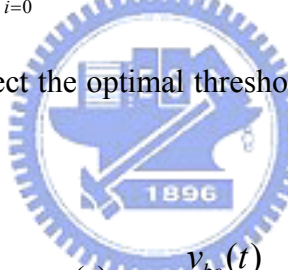
$$\mu_0 = \frac{\sum_{i=0}^{t-1} iP_i}{w_0}, \text{ and } \mu_1 = \frac{\sum_{i=t}^{L-1} iP_i}{w_1} \quad (2.6)$$

$$\sigma_0^2 = \frac{\sum_{i=0}^{t-1} P_i (i - \mu_0)^2}{w_0}, \text{ and } \sigma_1^2 = \frac{\sum_{i=t}^{L-1} P_i (i - \mu_1)^2}{1 - w_0} \quad (2.7)$$

And the total variance  $v_T$  and the overall mean  $\mu_T$  of pixels in the image  $\mathbf{I}$  are computed as,

$$v_T = \sum_{i=0}^{L-1} (i - \mu_T)^2 P_i, \text{ and } \mu_T = \sum_{i=0}^{L-1} iP_i \quad (2.8)$$

The criterion function used to select the optimal threshold for providing the best separability between the classes  $C_0$  and  $C_1$  is:



$$\psi_{BCV}(t) = \frac{v_{bc}(t)}{v_T} \quad (2.9)$$

The determination of the optimal threshold ( $t_{BCV}$ ) for achieving maximal separability between two classes, can be performed by maximizing the criterion function,

$$t_{BCV} = \text{Arg Max}_{0 \leq t < L} \psi_{BCV}(t). \quad (2.10)$$

We then extend this procedure to a multilevel thresholding case. For multilevel thresholding with  $k$  thresholds to partition the image into  $k+1$  classes, pixels of the image  $\mathbf{I}$  are segmented by applying a threshold set  $\mathbf{T}$ , which is composed of  $k$  thresholds, where  $\mathbf{T} = \{t_1, \dots, t_n, \dots, t_k\}$ . These classes are represented by  $C_0 = \{0, 1, \dots, t_1\}, \dots, C_n = \{t_n + 1, t_n + 2, \dots, t_{n+1}\}, \dots, C_k = \{t_k + 1, t_k + 2, \dots, L-1\}$ . First, the between-class variance can be

derived from Eq. (2.3) and is computed as,

$$v_{BC}(\mathbf{T}) = w_0(\mu_0 - \mu_T)^2 + \dots + w_n(\mu_n - \mu_T)^2 + \dots + w_k(\mu_k - \mu_T)^2 \quad (2.11)$$

Then the within-class variance for multilevel thresholding can also be derived from Eq. (2.4) and is computed as,

$$v_{WC}(\mathbf{T}) = w_0\sigma_0^2 + \dots + w_n\sigma_n^2 + \dots + w_k\sigma_k^2 \quad (2.12)$$

where  $k$  is the number of selected thresholds to segment pixels into  $k+1$  classes;  $w_n$  is the cumulative probability mass function of class  $C_n$ ;  $\mu_n$  and  $\sigma_n$  represent the mean and the standard deviation of pixels in class  $C_n$ , respectively. They are defined as,

$$w_0 = \sum_{i=0}^{t_1} P_i, \dots, w_n = \sum_{i=t_n+1}^{t_{n+1}} P_i, \dots, w_k = \sum_{i=t_k+1}^{L-1} P_i \quad (2.13)$$

$$\mu_0 = \frac{\sum_{i=0}^{t_1} iP_i}{w_0}, \dots, \mu_n = \frac{\sum_{i=t_n+1}^{t_{n+1}} iP_i}{w_n}, \dots, \mu_k = \frac{\sum_{i=t_k+1}^{L-1} iP_i}{w_k}, \quad (2.14)$$

$$\sigma_0^2 = \frac{\sum_{i=0}^{t_1} P_i(i - \mu_0)^2}{w_0}, \dots, \sigma_n^2 = \frac{\sum_{i=t_n+1}^{t_{n+1}} P_i(i - \mu_n)^2}{w_n}, \dots, \sigma_k^2 = \frac{\sum_{i=t_k+1}^{L-1} P_i(i - \mu_k)^2}{w_k} \quad (2.15)$$

where a dummy threshold  $t_0 = 0$  is utilized for simplifying the expression of equation terms. Thus, the optimal threshold set ( $\mathbf{T}_{BCV}$ ) can be determined by maximizing the following extended discriminant criterion function:

$$\mathbf{T}_{BCV} = \text{Arg Max}_{0 \leq T < L} \Psi_{BCV}(\mathbf{T}), \quad \Psi_{BCV}(\mathbf{T}) = \frac{v_{BC}(\mathbf{T})}{v_T} \quad (2.16)$$

## 2.2.2 The maximum entropy thresholding criterion

In entropy-based thresholding, the optimal threshold is obtained by applying information

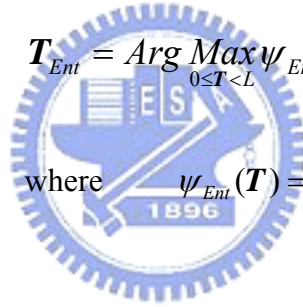
theory. As derived from Kapur *et al.*'s work [13], the original gray level distribution of the image is divided into a number of classes of probability distributions in the multilevel thresholding case. Then the entropies associated with these distributions are computed as,

$$H_n = - \sum_{i=t_n}^{t_{n+1}-1} (P_i/w_n) \log(P_i/w_n) \quad (2.17)$$

where  $n = 0, 1, \dots, k$ ; and the threshold set  $\mathbf{T}$ , and their corresponding probability mass functions  $w_n$  are computed similar to the ones utilized in the computation of the between-class variance criterion as described in the previous subsection. Then the optimal threshold set ( $\mathbf{T}_{Ent}$ ) is the threshold set which maximizes the entropy criterion, and is computed as,

$$\mathbf{T}_{Ent} = \underset{0 \leq \mathbf{T} < L}{\text{Arg Max}} \psi_{Ent}(\mathbf{T}), \quad (2.18)$$

$$\text{where } \psi_{Ent}(\mathbf{T}) = \sum_{n=0}^k H_n \quad (2.19)$$



### 2.2.3 The minimum error thresholding criterion

In the concept of minimum error thresholding [14], the gray level histogram of the image is thought of as an estimate of the probability density function  $p(i)$  of the mixture distribution, comprising the gray levels of several classes (i.e. objects and background). It is assumed that each of these class distributions  $p(i|n)$  of the mixture follows a normal distribution, with a class standard deviation of  $\sigma_n$ , a class mean of  $\mu_n$  and an *a priori* probability of  $w_n$ ; hence, the histogram can be approximated as,

$$p(i) = \sum_{n=0}^k \frac{w_n}{\sigma_n \sqrt{2\pi}} e^{-((i-\mu_n)^2/2\sigma_n^2)} \quad (2.20)$$

After re-arranging the nature log of both sides, the optimal threshold ( $\mathbf{T}_{ME}$ ) can be determined by solving the resultant quadratic equation with respect to  $i$ :

$$\begin{aligned} \frac{(i - \mu_0)^2}{\sigma_0^2} + 2 \log(\sigma_0) - 2 \log(w_0) &= \dots = \frac{(i - \mu_n)^2}{\sigma_n^2} + 2 \log(\sigma_n) - 2 \log(w_n) \\ \dots &= \frac{(i - \mu_k)^2}{\sigma_k^2} + 2 \log(\sigma_k) - 2 \log(w_k) \end{aligned} \quad (2.21)$$

However, the parameters  $w_n$ ,  $\mu_n$ , and  $\sigma_n$  of the mixture density function  $p(i)$  associated with the image are unknown. In order to overcome the difficulties of estimating the unknown parameters, Kittler and Illingworth [14] presented a criterion function  $\psi_{ME}(\mathbf{T})$ , which is given by,

$$\psi_{ME}(\mathbf{T}) = 1 + 2 \sum_{n=0}^k w_n [\log(\sigma_n) + \log(w_n)], \quad (2.22)$$

And  $w_n$ ,  $\mu_n$  and  $\sigma_n$  are respectively obtained as Eqs. (2.13), (2.14), and (2.15) which are similar to those in the computation of the between-class variance criterion. Hence, the optimal threshold set ( $\mathbf{T}_{ME}$ ) can be determined by minimizing the criterion function  $\psi_{ME}(\mathbf{T})$ :

$$\mathbf{T}_{ME} = \underset{0 \leq T < L}{\text{Arg Min}} \psi_{ME}(\mathbf{T}) \quad (2.23)$$

## 2.3 The Proposed Fast Combinatorial Scheme for Efficient Selection of Multiple Thresholds

### 2.3.1 The Combinatorial Search Scheme of Optimal Threshold Set Selection

To obtain the optimal threshold set  $\mathbf{T} = \{t_1, \dots, t_m, \dots, t_k\}$ , it is necessary to evaluate all the possible threshold sets to optimize the aforementioned criterion functions  $\psi(\mathbf{T})$ . Traditionally, one can obtain the optimal threshold set by a simple and iterative, but exhaustive search procedure. The values of all thresholds in  $\mathbf{T}$  are iteratively incremented and

evaluated till the maximum or minimum  $\Psi(\mathbf{T})$  is obtained, and is performed as the pseudo code procedure presented in Figure 2.1.

```

Max_ψ ← 0; (Min_ψ ← ∞; for using the Minimum Error criterion)
Optiaml_T ← 0
for (tl = 0 to L-1) do {
    .
    .
    for (tk = k-1 to L-1) do {
        Evaluate Ψ(T); (of ΨBCV(T), ΨEnt(T) or ΨME(T) )
        If Ψ(T) > Max_ψ then (using Ψ(T) < Min_ψ when using the Minimum Error
            criterion)
        {
            Max_ψ ← Ψ(T); (Min_ψ ← Ψ(T) when using the Minimum Error criterion)
            Optiaml_T ← T;
        }
    }
    .
    .
}

```

Figure 2.1. Conventional search procedure of optimal  $\Psi(\mathbf{T})$

It is obvious that the exhaustive search for finding the optimal  $\Psi(\mathbf{T})$  is very time consuming. In order to obtain the optimal threshold set with  $k$  thresholds,  $L \times (L-1) \times (L-2) \dots \times (L-k+1) = \frac{L!}{(L-k)!}$  possible threshold sets must be evaluated! Thus, for example, when the number of thresholds is  $k = 3$ , and the number of gray levels  $L = 256$ , the number of potential threshold sets for which the criterion function  $\Psi(\mathbf{T})$  must be evaluated is 16581120. This is because many equivalent threshold sets are repeatedly evaluated. In addition, this implementation is not scalable and parameterizable for variable amount of thresholds in multilevel thresholding, that is, the thresholding procedure cannot be

adapted to the changing of the number of desired thresholds  $k$ .

To overcome these above mentioned drawbacks which arise from extending bi-level thresholding to multilevel thresholding cases, we introduce a combinatorial scheme to suppress the computation complexity, and to accomplish the parameterization of a desired threshold number  $k$ . The iterative procedure of evaluating all potential threshold sets for determining the optimal threshold set can be viewed as a combinatorial problem [69]. So we implement an efficient combinatorial generation procedure to obtain all potential threshold sets  $T$ . This combinatorial scheme avoids duplicate evaluation of equivalent threshold sets, and significantly reduces the amount of computation required.

Looking at it as a combinatorial problem, the multilevel thresholding for selecting a set with  $k$  thresholds can be regarded as an “ $L$  choose  $k$ ” problem, i.e.  $\binom{L}{k}$ . The combination of a potential threshold set is denoted as a  $k$ -subset of the set of  $L$  gray levels. There are  $\binom{L}{k} = \frac{L!}{k!(L-k)!}$   $k$ -subsets possible in  $L$  gray levels. The task now of the multilevel thresholding is to find all these possible threshold combinations and determine the appropriate threshold set which achieves the optimal condition of the criterion function.

In this study, we employ the *Twiddle algorithm* [70] to obtain all candidate combinations of thresholds. The *Twiddle* algorithm is a simple and efficient method for generating a threshold set. It has several features: 1) at each stage, only one element of the combination, i.e.  $C[z]$  as mentioned below, is altered for generating a new successive combination of numbers (threshold values), 2) this algorithm is order preserving in the sense that the elements of each threshold combination generated in each stage are strictly increasing with respect to the order of the set of  $L$  gray levels.

We implement this algorithm using the C++ programming language, including several

necessary modifications for utilization in the C++ environment. Let the array  $G[0...L-1]$  store all elements of the set of  $L$  gray levels; and let the successive combinations be stored in the array  $C[0...k-1]$ . Consequently, one consecutive  $k$ -subset of combinations of thresholds is stored in this array in each stage of performing the *Twiddle*. Then the auxiliary integer array  $p[0...L+1]$  is utilized to store the states of the *Twiddle* procedure. The auxiliary array  $p$  is initialized as follows:  $p[0]$  is set equal to  $L+1$ ;  $p[1]$  through  $p[L-k]$  are set equal to  $0$ ;  $p[L-k+1]$  through  $p[L]$  are set equal to  $1$  through  $k$ , respectively, and  $p[L+1]$  is set equal to  $-2$ . When the *Twiddle* procedure is performed, three indexing numbers  $x, y, z$  are utilized to alter the states of the auxiliary array  $p$  and produce new successive combinations. A Boolean variable “*finish*” is employed as a return value of the *Twiddle* procedure to reflect if there are still new successive combination that can be produced. Initially, the “*finish*” is set equal to **false**, and the elements of the first combination  $C[0]$  through  $C[k-1]$  are set equal to  $G[L-k]$  through  $G[L-1]$ , respectively. Hence the *Twiddle* procedure is called upon, and is performed repeatedly to produce consecutive combinations by setting  $C[z]$  equal to  $G[x]$ , until the “*finish*” is toggled to be **true**. The pseudo code procedure of the *Twiddle* is presented in Figure 2.2.

```

procedure Twiddle(x, y, z, p, finish):
{
     $i \leftarrow 0, j \leftarrow 0$ , and  $k \leftarrow 0$ ;
    while ( $p[j] \leq 0$ ) do {
         $j \leftarrow j + 1$ ;
    }

    If ( $p[j-1] = 0$ ) then {
        For ( $i \leftarrow j-1$  step -1 until  $i = 1$ ) do {
             $p[i] \leftarrow -1$ ;
        }
         $p[j] \leftarrow 0$ ;
    }
}

```



```

     $x \leftarrow 0$ ,  $z \leftarrow 0$  and  $y \leftarrow j-1$ ;
     $p[1] \leftarrow 1$ ;
}
Else then {
    If ( $j > 1$ ) then {
         $p[j-1] \leftarrow 0$ ;
    }
    While ( $p[j] > 0$ ) do {
         $j \leftarrow j + 1$ ;
    }

     $k \leftarrow j - 1$ , and  $i \leftarrow j$ ;

    while ( $p[i] = 0$ ) do {
         $p[i] \leftarrow -1$ , then  $i \leftarrow i + 1$ ;
    }

    If ( $p[i] = -1$ ) then {
         $p[i] \leftarrow p[k]$ ;
         $z \leftarrow p[k]-1$ ;
         $x \leftarrow i-1$ , and  $y \leftarrow k-1$ ;
         $p[k] \leftarrow -1$ ;
    }

    Else then {
        If ( $i = p[0]$ ) then {
            finish  $\leftarrow$  true; (All combinations have been produced. Then the Twiddle
            procedure should be terminated by toggling "finish")
            Return (finish = true);
        }
        Else then {
             $p[j] \leftarrow p[i]$ ;
             $z \leftarrow p[i]-1$ ;
             $p[i] \leftarrow 0$ ;
             $x \leftarrow j-1$ , and  $y \leftarrow i-1$ ;
        }
    }
}
finish  $\leftarrow$  false; (there are still more successive combinations to be produced.)

```



```

return (finish = false);
}

```

Figure 2.2. The pseudo code of the *Twiddle* procedure

Hence, the optimal threshold set can be obtained by recursively performing the *Twiddle* procedure. The potential threshold sets are successively produced by *Twiddle*. They are then evaluated until the maximum or minimum  $\Psi(T)$  is achieved, and can be performed as the pseudo code procedure shown in Figure 2.3.

```

G[0...L-1] stores the set of L gray levels
Max_ψ ← 0; (Min_ψ ← ∞; for using the Minimum Error criterion)
Optiaml_T ← 0;
C[0]...[k-1] ← G[L-k] ...G[L-1], respectively; (Initialize combination array C [0...k-1])
x ← 0, y ← 0, and z ← 0;
p[0] ← L+1; p[1]...p[L-k] ← 0; p[L-k+1]...p[L] ← 1...k; p[L+1] ← -2; (Initialize auxiliary
array p[0...L+1])
finish ← false;
While (finish = false) do {
    Perform Twiddle (x, y, z, p, finish);
    C[z] ← G[x]; (to produce a new combination and store in array C)
    T ← C; (assign the combination to the current threshold set to be evaluated)
    Evaluate Ψ(T); (of ΨBCV(T), ΨEm(T) or ΨME(T) )
    If Ψ(T) > Max_ψ (using Ψ(T) < Min_ψ when performing the Minimum Error method)
    then
    {
        Max_ψ ← Ψ(T); (Min_ψ ← Ψ(T) when using the Minimum Error criterion)
        Optiaml_T ← T;
    }
}
}

```

Figure 2.3. Improved search procedure of optimal  $\Psi(T)$  using the combinatorial scheme

Consequently, the amount of evaluations of potential threshold sets that obtain the optimal threshold set using the combinatorial scheme is  $1/k!$  of those using the original exhaustive search scheme. For example, when performing in the case of quarter-level thresholding with three thresholds as previously mentioned, only 2763520 threshold sets are needed to be evaluated for determining the optimal one using the combinatorial scheme, which reduces  $3!=6$  times of evaluations compared with those using the exhaustive search scheme. Consequently the amount of evaluating values of the criterion function can be significantly reduced. Furthermore, this combinatorial scheme also parameterizes the number of desired thresholds. This means that by using this procedure the thresholding process can be adopted, without any changes, on a variable desired number of thresholds.



### **2.3.2 Fast Implementation of Criterion Functions for Multilevel Thresholding Methods**

As described in the previous section, it can be seen that finding one evaluation value of a certain criterion function ( $\psi_{BCV}(\mathbf{T})$ ,  $\psi_{Ent}(\mathbf{T})$  or  $\psi_{ME}(\mathbf{T})$ ) of one threshold set  $\mathbf{T}$  requires the computation of some class statistics of prospective threshold partitions, i.e.  $w_n$ ,  $\mu_n$ ,  $\sigma_n$ , and  $H_n$  in Eqs. (2.13), (2.14), (2.17), and (2.15), respectively. These class statistics involve summations which are performed on each position of gray values on the histogram to obtain a certain value of them. Most of calculations of these summations are repeatedly performed many times using their original computation forms in Eqs. (2.13), (2.14), (2.17), and (2.15), and most of these calculations are redundant and time consuming. It can be found that complete summations of the above-mentioned class statistics are not necessarily performed for obtaining each criterion function value of a given threshold set  $\mathbf{T}$ . To further reduce the

computation timing for obtaining one evaluation value of a certain threshold set  $\mathbf{T}$ , such redundant computations can be avoided by using pre-computed look-up tables of these summation terms of the class statistics involved in the thresholding criterion functions. By observing the computation equations of  $w_n$ ,  $\mu_n$ ,  $\sigma_n$ , and  $H_n$ , it can be seen that these class statistics comprise of summations of the zeroth moment, the first moment, the a priori entropy, and the second moment, respectively, as viewed in the following expansion forms of their computation equations. In here, the summation terms of the zeroth moment, the first moment, the second moment, and the a priori entropy which are calculated on an interval  $t_a - t_b$  formed by two thresholds of  $\mathbf{T}$ , i.e. the summation is performed from  $t_a$ -position to  $t_b$ -position on the histogram, are denoted as  $Z(t_a, t_b)$ ,  $F(t_a, t_b)$ ,  $S(t_a, t_b)$ , and  $E(t_a, t_b)$ , respectively. They are derived from aforementioned computation equations of  $w_n$ ,  $\mu_n$ ,  $\sigma_n$ , and  $H_n$ , and are defined as:

$$w_n = \sum_{i=t_n}^{t_{n+1}-1} P_i = Z(t_n, t_{n+1}-1), \text{ and } Z(t_a, t_b) = \sum_{i=t_a}^{t_b} P_i \quad (2.24)$$

$$\mu_n = \frac{\sum_{i=t_n}^{t_{n+1}-1} iP_i}{w_n} = \frac{F(t_n, t_{n+1}-1)}{w_n}, \text{ and } F(t_a, t_b) = \sum_{i=t_a}^{t_b} iP_i \quad (2.25)$$

$$\sigma_n^2 = \frac{\sum_{i=t_n}^{t_{n+1}-1} P_i (i - \mu_n)^2}{w_n} = \frac{\sum_{i=t_n}^{t_{n+1}-1} i^2 P_i}{w_n} - \mu_n^2 = \frac{S(t_n, t_{n+1}-1)}{w_n} - \mu_n^2, \quad (2.26)$$

$$\text{and } S(t_a, t_b) = \sum_{i=t_a}^{t_b} i^2 P_i$$

$$\begin{aligned}
H_n &= - \sum_{i=t_n}^{t_{n+1}-1} \frac{P_i}{w_n} \log \left( \frac{P_i}{w_n} \right) = \log(w_n) - \frac{\sum_{i=t_n}^{t_{n+1}-1} P_i \log(P_i)}{w_n}, \\
&= \log(w_n) - \frac{E(t_n, t_{n+1}-1)}{w_n},
\end{aligned} \tag{2.27}$$

$$\text{and } E(t_a, t_b) = \sum_{i=t_a}^{t_b} P_i \log(P_i)$$

As observed from above descriptions, it can be found that the values of  $Z(t_a, t_b)$ ,  $F(t_a, t_b)$ ,  $S(t_a, t_b)$ , and  $E(t_a, t_b)$  are repeatedly involved in obtaining each of the criterion function values. The re-calculation of the complete summations of them can be avoided in obtaining each criterion function value by using the following pre-computed tables for storing all potential values of  $Z(t_a, t_b)$ ,  $F(t_a, t_b)$ ,  $S(t_a, t_b)$ , and  $E(t_a, t_b)$  with respect to all possible intervals of  $t_a - t_b$  on the histogram, and hence the computation cost of these summations can be significantly saved. The pre-computed tables of  $Z$ ,  $F$ ,  $S$ , and  $E$  are depicted in Table 2.1 – Table 2.4. In here, the  $t_a - t_b$  intervals are bounded within  $0 \leq t_a \leq t_b \leq L-1$ .

Table 2.1. The values of the zeroth moment  $Z(t_a, t_b)$  of intervals  $t_a - t_b$  on the histogram

$t_a$	$t_b$					
	0	1	...	$b$	...	$L-1$
0	$Z(0,0)$	$Z(0,1)$	...	$Z(0,b)$	...	$Z(0,L-1)$
1	0	$Z(1,1)$	...	$Z(1,b)$	...	$Z(1,L-1)$
...	0	0	...	...	...	...
$a$	0	0	0	$Z(a,b)$	...	$Z(a,L-1)$
...	0	0	0	0	...	...
$L-1$	0	0	0	0	0	$Z(L-1,L-1)$

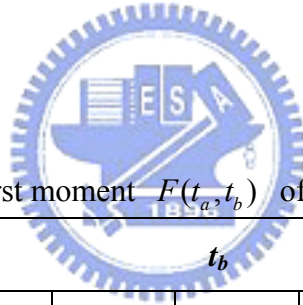


Table 2.2. The values of the first moment  $F(t_a, t_b)$  of intervals  $t_a - t_b$  on the histogram

$t_a$	$t_b$					
	0	1	...	$b$	...	$L-1$
0	$F(0,0)$	$F(0,1)$	...	$F(0,b)$	...	$F(0,L-1)$
1	0	$F(1,1)$	...	$F(1,b)$	...	$F(1,L-1)$
...	0	0	...	...	...	...
$a$	0	0	0	$F(a,b)$	...	$F(a,L-1)$
...	0	0	0	0	...	...
$L-1$	0	0	0	0	0	$F(L-1,L-1)$

Table 2.3. The values of the second moment  $S(t_a, t_b)$  of intervals  $t_a - t_b$  on the histogram

$t_a$	$t_b$					
	0	1	...	$b$	...	$L-1$
0	$S(0,0)$	$S(0,1)$	...	$S(0,b)$	...	$S(0,L-1)$
1	0	$S(1,1)$	...	$S(1,b)$	...	$S(1,L-1)$
...	0	0	...	...	...	...
$a$	0	0	0	$S(a,b)$	...	$S(a,L-1)$
...	0	0	0	0	...	...
$L-1$	0	0	0	0	0	$S(L-1,L-1)$

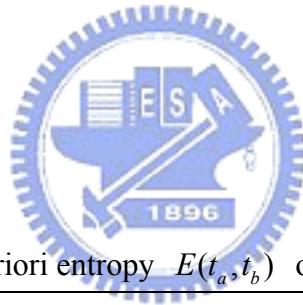


Table 2.4. The values of the a priori entropy  $E(t_a, t_b)$  of intervals  $t_a - t_b$  on the histogram

$t_a$	$t_b$					
	0	1	...	$b$	...	$L-1$
0	$E(0,0)$	$E(0,1)$	...	$E(0,b)$	...	$E(0,L-1)$
1	0	$E(1,1)$	...	$E(1,b)$	...	$E(1,L-1)$
...	0	0	...	...	...	...
$a$	0	0	0	$E(a,b)$	...	$E(a,L-1)$
...	0	0	0	0	...	...
$L-1$	0	0	0	0	0	$E(L-1,L-1)$



To rapidly construct the necessary tables for evaluating the criterion functions, we apply the recursively accumulative property of  $Z(t_a, t_b)$ ,  $F(t_a, t_b)$ ,  $S(t_a, t_b)$ , and  $E(t_a, t_b)$  for saving the amounts of “additions” for obtaining their summation terms during computing their values for consecutive  $(t_a - t_b)$  intervals. Their recursively accumulative computation forms can be obtained as,

$$Z(0, t_b) = \sum_{i=0}^{t_b-1} P_i + P_{t_b} = Z(0, t_b - 1) + P_{t_b} \quad , \quad (2.28)$$

$$\text{and } Z(t_a, t_b) = \sum_{i=0}^{t_b} P_i - \sum_{i=0}^{t_a-1} P_i = Z(0, t_b) - Z(0, t_a - 1) \quad (2.29)$$

Similarly, the computation forms of  $F$ ,  $S$ , and  $E$  can be obtained as

$$F(0, t_b) = F(0, t_b - 1) + t_b P_{t_b} \quad , \quad (2.30)$$

$$\text{and } F(t_a, t_b) = F(0, t_b) - F(0, t_a - 1) \quad (2.31)$$

$$S(0, t_b) = S(0, t_b - 1) + t_b^2 P_{t_b} \quad , \quad (2.32)$$

$$\text{and } S(t_a, t_b) = S(0, t_b) - S(0, t_a - 1) \quad (2.33)$$

$$E(0, t_b) = E(0, t_b - 1) + P_{t_b} \log(P_{t_b}) \quad , \quad (2.34)$$

$$\text{and } E(t_a, t_b) = E(0, t_b) - E(0, t_a - 1) \quad (2.35)$$

Hence, by using the above computation forms, the look-up tables of  $Z(t_a, t_b)$ ,  $F(t_a, t_b)$ ,  $S(t_a, t_b)$ , and  $E(t_a, t_b)$  can be rapidly constructed by using the following two steps:

**Step 1.** For the values of the first rows of Table 2.1 – Table 2.4, the recursively accumulative forms of in Eqs. (2.28), (2.30), (2.32), and (2.34) are utilized for obtaining the values of  $Z(0, t_b)$ ,  $F(0, t_b)$ ,  $S(0, t_b)$ , and  $E(0, t_b)$ , respectively.

**Step 2.** Then, for the rest rows of the tables, the values of  $Z(t_a, t_b)$ ,  $F(t_a, t_b)$ ,  $S(t_a, t_b)$ , and  $E(t_a, t_b)$  are obtained using Eqs. (2.29), (2.31), (2.33), and (2.35), respectively.

Hence, the class statistics  $w_n$ ,  $\mu_n$ ,  $\sigma_n$ , and  $H_n$  terms in the criterion functions,  $\psi_{BCV}(\mathbf{T})$  (applying  $Z$  and  $F$ ),  $\psi_{Ent}(\mathbf{T})$  (applying  $Z$  and  $E$ ), and  $\psi_{ME}(\mathbf{T})$  (applying  $Z$ ,  $F$ , and  $S$ ) can be rapidly computed by using index operations with the look-up tables of  $Z$ ,  $F$ ,  $S$ , and  $E$ , as depicted in Table 2.1 – Table 2.4 and Eqs. (2.24) – (2.27). By integrating the fast look-up table scheme with the combinatorial scheme based search procedure, the computation timing for obtaining the optimal threshold set  $\mathbf{T}$  can be dramatically reduced.

## 2.4 The Proposed Automatic Multilevel Thresholding Method

When segmenting objects from a given image, different objects with homogeneous illuminations must be separated into different segmented images. For this purpose, the discriminant criterion, for measuring separability among the segmented images with different objects, is described in this section. By evaluating the separability criterion, the number of objects, into which the image should be segmented, can be automatically determined. Hence, an automatic multilevel thresholding method, based on this criterion, is proposed as follows.

The discriminant criterion functions mentioned in Section 2.2 can be considered a measure of separability, among all existing classes, decomposed from the original image  $\mathbf{I}$ . We introduce this concept as a criterion of automatic image segmentation, denoted by the “separability factor” –  $\mathcal{SF}$  in this study, which is defined as,

$$\mathcal{SF} = \frac{v_{BC}(\mathbf{T})}{v_T} \quad (2.36)$$

where  $v_T$  is the total variance of the gray-level values of the image  $\mathbf{I}$  and serves as the

normalization factor in this equation. The  $SF$  value measures the separability among all existing classes, and the  $SF$  value lies within the range  $0 \leq SF \leq 1$ . Maximizing the  $SF$  value is the objective, to optimize the segmentation result. This property is more obvious when using an alternative expression of this measure:

$$SF = \frac{v_{BC}(\mathbf{T})}{v_T} = 1 - \frac{v_{WC}(\mathbf{T})}{v_T} \quad (2.37)$$

This can be supported by the property of the total variance is equivalent to the sum of the between class-variance and the within-class variance. This property is obtained by,

$$\begin{aligned} v_{BC}(\mathbf{T}) + v_{WC}(\mathbf{T}) &= \sum_{n=0}^k \left[ w_n (\mu_n - \mu_T)^2 + w_n \sigma_n^2 \right] \\ &= \sum_{n=0}^k \left[ (w_n \mu_n^2 - 2w_n \mu_T + w_n \mu_T^2) + \left( \sum_{k=t_n+1}^{t_{n+1}} i^2 P_i - w_n \mu_n^2 \right) \right] \\ &\quad \text{(cancel both } w_n \mu_n^2 \text{ terms and substitute } w_n \text{ by } \sum_{i=t_n+1}^{t_{n+1}} P_i) \\ &= \sum_{n=0}^k \left[ \sum_{k=t_n+1}^{t_{n+1}} P_i (i^2 - 2i \mu_T + \mu_T^2) \right] \\ &= \sum_{i=0}^{L-1} (i - \mu_T)^2 P_i = v_T \end{aligned} \quad (2.38)$$

In the above derivation, the dummy threshold  $t_0 = 0$  is utilized for convenience in simplifying the expression of equation terms. Through observation of the terms comprising  $v_{WC}(\mathbf{T})$ , if the pixels in each class are broadly spread, i.e. the contribution of the class variance  $\sigma_n^2$  is large, then the corresponding  $SF$  measure becomes small. Hence, when  $SF$  approaches 1.0, all classes of gray levels decomposed from the original image  $\mathbf{I}$  are ideally and completely separated. This property also satisfies the concept of uniformity of the segmented regions, as presented by Levein and Nazif [68].

Accordingly, based on this efficient discriminant criterion on measuring the separability of the object regions of homogenous gray levels, an automatic multilevel thresholding method can be developed to recursively segment homogeneous objects from the image  $I$ , regardless of the number of objects and the complexity of the image. The multilevel thresholding process can be recursively performed on the gray levels of the image  $I$  until the  $SF$  measure is large enough, i.e.  $SF$  approaches 1.0, to reflect that the appropriate discrepancy among the resultant classes of gray levels is achieved so that the homogeneous objects are completely segmented into separate thresholded images.

Through the aforementioned properties, this objective can be reached by minimizing the total within-class variance  $v_{wc}(\mathbf{T})$ . This can be achieved by the scheme that selects the class with the maximal contribution ( $w_n \sigma_n^2$ ) of the total within-class variance for performing the bi-level thresholding procedure to partition it into two more classes in each recursion. Thus, the  $SF$  measure will most rapidly achieve the maximal increment to satisfy sufficient separability among the resultant classes of gray levels. Furthermore, objects with homogeneous gray levels, will be well-separated.

Based on the above definitions, a new automatic multilevel thresholding method is developed. The details of the proposed method are presented below.

**Step 1:** In the beginning, compute the histogram of gray values of the image  $I$ , and all the gray values in  $I$  are assigned to one initial class  $C_0$ . Let  $q$  denote the number of currently determined thresholds in the threshold set  $\mathbf{T}$ , which classify the gray values into  $q+1$  classes; Initially,  $\mathbf{T}$  comprise of no thresholds and  $q = 0$ .

**Step 2:** In current recursion,  $q$  thresholds have been determined, i.e.  $\mathbf{T} = \{t_1, \dots, t_n, \dots, t_q\}$ , which partition the gray values of the image  $I$  into  $q+1$  classes ( $C_0, \dots, C_n, \dots, C_q$ ). Compute the class-mean  $\mu_n$ , the cumulative probability mass function  $w_n$ , and the standard deviation

$\sigma_n$  of each existing class  $C_n$  using Eqs. (2.13) – (2.15), respectively, where  $n$  denotes the index of the present classes and  $n = 0 \sim q$ .

**Step 3:** From all classes  $C_n$ , determine the class  $C_p$  with the maximal contribution ( $w_n \sigma_n^2$ ) of the total within-class variance  $v_{WC}$ , which is to be partitioned in the following step to achieve the maximal increment of  $SF$ .

**Step 4:** Determine the optimal threshold  $t_S^*$  to partition  $C_p: \{t_p + 1, t_p + 2, \dots, t_{p+1}\}$  into two classes  $C_{p0}$  and  $C_{p1}$  which comprise the subsets of gray values decomposed from  $C_p$ . The  $t_S^*$  is obtained by maximizing the between-class variance  $v'_{BC}$  of partitioned  $C_{p0}$  and  $C_{p1}$  with respect to  $t_S$ , and is computed as,

$$v'_{BC}(t_S^*) = \underset{t_p < t_S \leq t_{p+1}}{\text{Max}} v'_{BC}(t_S) \quad (2.39)$$

$$v'_{BC}(t_S) = w_{p0}(\mu_{p0} - \mu_p)^2 + w_{p1}(\mu_{p1} - \mu_p)^2, \quad (2.40)$$

$$w_{p0} = \sum_{i=t_p+1}^{t_S} P_i, \quad w_{p1} = \sum_{i=t_S+1}^{t_{p+1}} P_i \quad (2.41)$$

$$\mu_{p0} = \sum_{i=t_p+1}^{t_S} iP_i / w_{p0}, \quad \mu_{p1} = \sum_{i=t_S+1}^{t_{p+1}} iP_i / w_{p1} \quad (2.42)$$

$$w_p = \sum_{i=t_p+1}^{t_{p+1}} P_i, \quad \mu_p = \sum_{i=t_p+1}^{t_{p+1}} iP_i / w_p \quad (2.43)$$

where  $w_p$  and  $\mu_p$  are the class-probability and class-mean of  $C_p$ , respectively.

Then  $t_S^*$  is put into the threshold set  $\mathbf{T}$ , and is applied to partition  $C_p$  into  $C_{p0}$  and  $C_{p1}$ , i.e.

$C_{p0}: \{t_p + 1, t_p + 2, \dots, t_S^*\}$ , and  $C_{p1}: \{t_S^* + 1, t_S^* + 2, \dots, t_{p+1}\}$ . Hence the gray values of the image

$I$  are then divided into  $q+1$  classes. Then re-label all the thresholds in  $T$  and let  $q = q+1$ .

**Step 5:** Computed the  $SF$  measure of all currently obtained classes using Eq. (2.36). If the following “*Objective Condition*” is satisfied,

$$SF \geq Th_{SF} \quad (2.44)$$

then go to Step 6; otherwise, go back to Step 2 to perform further partition process on the obtained classes.

**Step 6:** Deliver the obtained threshold set  $T$ , and then classify the pixels into  $q+1$  separate classes  $C_0, C_1, \dots, C_q$  using to the resultant threshold values  $t_1, t_2, \dots, t_q$ ; and terminate the thresholding procedure.

As pointed out in the above-mentioned derivations in Eqs. (2.37)-(2.38), it can be seen that the intensity of the  $SF$  measure reflects whether if the obtained classes of gray levels (homogenous objects) have sufficient discrepancies, and are well-separated with each other. Hence, at each recursion in the proposed automatic multilevel thresholding procedure, one optimal threshold is determined to partition the class which comprises of the most divergent distribution of gray levels, i.e. with the maximal contribution ( $w_n \sigma_n^2$ ) of the  $v_{WC}(T)$ , to achieve maximum increment of the  $SF$  measure. This way can accomplish the satisfactory thresholded results by means of the smallest number of thresholding levels. In plain words, the number of thresholding levels is determined by the number of the necessary times of recursions for partitioning the gray levels of the image into appropriate number of classes so that the resultant  $SF$  measure can satisfy the above-mentioned *objective condition*. Here the parameter  $TH_{SF}$  is the pre-defined threshold to determine whether the thresholded objects are sufficiently separated to satisfy the objective condition. This study employs  $Th_{SF} = 0.92$ , which is determined from the training using numerous images, such that all existing classes

are satisfactorily separated. Consequently, we obtain  $q+1$  thresholded classes of gray levels,

$$I = C_0 \cup \dots \cup C_n \cup \dots \cup C_q$$

where each class  $C_n$  represents one homogeneous object decomposed from the original image  $I$ .

Besides, the proposed multilevel thresholding method can also be modified to be performed on the demand for thresholding the gray levels of the image into a desired number of classes. To accomplish this demand, the only necessary change is to replace the *objective condition* in Eq. (2.44) in Step 5 of the algorithm by the following one:

$$q = K \tag{2.45}$$

where  $K$  is the desired number of thresholds ( $K \geq 1$ ) to divide the gray levels of the image into  $K+1$  classes. Hence, the thresholds can be determined one by one in each recursion until the desired number of thresholding levels is achieved, and each determined threshold is ensured to achieve the maximum separation on the obtained thresholded images.

## 2.5 Experimental Results

In this section, the performance of the two proposed methods are evaluated and compared to the three most well-known and widely applied thresholding methods, which have been used in cases performing a supervised number of thresholding levels. We utilized two photographs - “Road” and “Lena” - and a real-life document image - “Advertisement” - as plotted in Figure 2.4(a)-Figure 2.6(a), respectively. They were transformed into gray-scale images with  $L=256$  gray-levels. The proposed combinatorial scheme is implemented on the multilevel versions of the three above-mentioned thresholding criterion functions –



$\Psi_{BCV}(\mathbf{T})$ ,  $\Psi_{Ent}(\mathbf{T})$ , and  $\Psi_{ME}(\mathbf{T})$ . Then multilevel thresholding experiments were performed, using the above-mentioned three thresholding criterion methods – the between-class variance method [12], the entropy method [13], and the minimum error method [14], the improved versions of these three criterion methods by the proposed combinatorial scheme, Fleury *et. al*'s running entropic method [26] for the maximum entropy criterion, and the proposed automatic multilevel thresholding method, in order to analyze and evaluate their performances. These methods were implemented on a Pentium-4-based personal computer using C++ programming language. As well as using subjective visual analysis, we adopted the “uniformity measure” from Levein and Nazif’s literature [68] to objectively evaluate the segmentation performance of these methods. If an image is well-segmented, the uniformity measure score should be close to 1. Besides, as for the experimental data, the computation timings of the proposed combinatorial scheme combined with the three thresholding criterion functions includes the construction time of their necessary corresponding class statistics tables.



The first test image, “Road”, a photograph taken in Hsinchu, is shown in Figure 2.4(a). After the proposed automatic multilevel thresholding method had been performed, two suitable thresholds were automatically determined, as listed in Table 2.5. Then the other three fixed-level methods were performed, by setting the number of desired thresholds to that determined by the proposed method. The performance results of these three fixed-level methods, as well as the proposed method, are shown in Figure 2.4 and Table 2.5. In this image, the light foreground objects, lanes and sky, are most interesting. As shown in Figure 2.4(b), (d), (f) and (h), only the proposed method and the between-class variance method correctly segments the lanes and sky, while the other two methods are unable to select adequate threshold values to obtain satisfactory thresholded images. In Figure 2.4(c), (e), (g) and (i), the gray levels (0, 127, and 255) have been utilized as representative gray levels of

the three segmented intervals to enhance the contrast of the combined thresholded images, derived by the three methods, respectively. Hence in this case, it is obvious that the proposed method and the between-class variance method obtain satisfactory thresholded images, as well as similar uniformity scores. As seen from the computational experimental information for each method listed in Table 2.5, the proposed automatic multilevel thresholding method saves a great deal of computational time achieves fastest computation performance among all methods, and it can also be seen that the proposed combinatorial scheme and Fleury *et. al*'s running entropic method save a significant amount of computation timings compared to the corresponding original methods of these criterion functions.

Table 2.5. Experimental data of performing thresholding on Figure 2.4(a), “Road”, by our proposed automatic multilevel thresholding method (AMT), Between-class variance method (BCV), Entropy method (ENT) and Minimum Error method (ME)

Criterion Method	Threshold levels	Selected threshold values	Uniformity measure	Computation time (seconds)		
				Original method	with Fleury <i>et al.</i> 's method	with our combinatorial scheme
AMT	3	96(2), 174(1)	0.942	0.00063	-	-
BCV	3*	96, 185	0.943	0.438	-	0.015
ENT	3*	44, 98	0.417	1.156	0.032	0.031
ME	3*	252, 253	0.738	0.656	-	0.047

“\*” denotes the number of thresholding levels is given by supervision

(n) denotes the order of the threshold been determined by the proposed method



(a) Original image



(b) The light object image derived by the proposed method



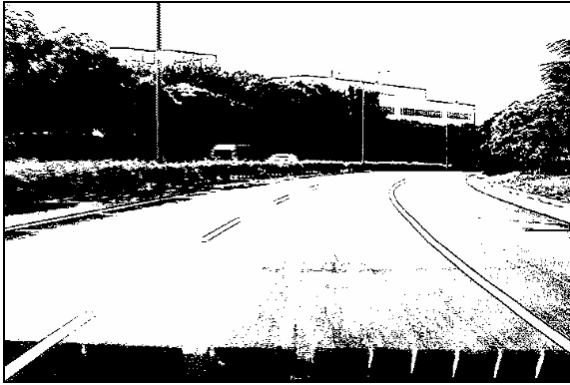
(c) The combined thresholded image derived by the proposed method



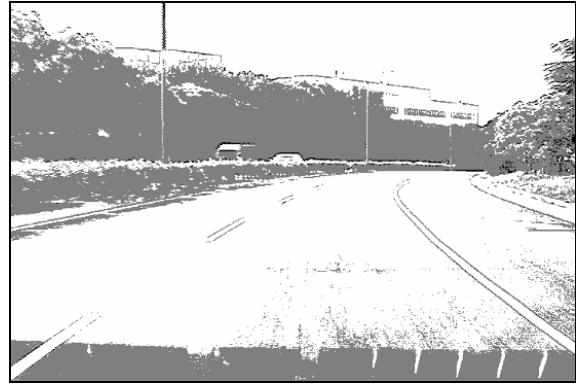
(d) The light object image derived by Between-class variance method



(e) The combined thresholded image derived by Between-class variance method



(f) The light object image derived by Entropy method



(g) The combined thresholded image derived by Entropy method



(h) The light object image derived by Minimum Error method



(i) The combined thresholded image derived by Minimum Error method

Figure 2.4. Result of segmenting the test image 1, "Road" (image size=720×480)

Figure 2.5 and Table 2.6 show the results for the second test image, “Lena”. After performing the proposed method, three thresholds were automatically determined, as listed in Table 2.6. Then the other three methods were performed by setting the desired number of thresholds the same as determined by the proposed method. In the “Lena” image, the dark foreground objects, the frame of the mirror, her hat decoration and her eyes and hair, are most interesting. As shown in Figure 2.5(b), (d), (f) and (h), the three fixed-level methods and the proposed method provided acceptable dark foreground object images with some differences. In Figure 2.5(c), (e), (g) and (i), the combined thresholded images, derived by the four methods, are plotted by utilizing the gray levels (0, 85, 170, and 255) as the representative gray levels. Hence, in this case, some of the fixed-level thresholding methods (such as the between-class variance method and the entropy method) can provide results as good as the proposed method, in both visual effect and uniformity scores. However, they require much more computational time to obtain the optimal threshold sets. From the corresponding information in Table 2.6, we can find that the proposed combinatorial scheme’s advantages on savings of computation timings on the thresholding criterion methods are more obvious as the number of desired thresholds is growing larger, and the proposed automatic multilevel thresholding method provides even more savings on computational costs.

Table 2.6. Experimental data of performing thresholding on Figure 2.5(a), “Lena”, by our automatic multilevel thresholding method (AMT), Between-class variance method (BCV), Entropy method (ENT), and Minimum Error method (ME)

Criterion Method	Threshold levels	Selected threshold values	Uniformity measure	Computation time (seconds)		
				Original method	with Fleury <i>et al.</i> 's method	with our combinatorial scheme
AMT	4	80( <b>3</b> ), 118( <b>1</b> ), 166( <b>2</b> )	0.926	0.00078	-	-
BCV	4*	83, 126, 169	0.930	42.192	-	0.328
ENT	4*	88, 129, 176	0.927	131.442	1.110	0.609
ME	4*	68, 246, 247	0.472	66.936	-	1.218

“\*” denotes the number of thresholding levels is given by supervision

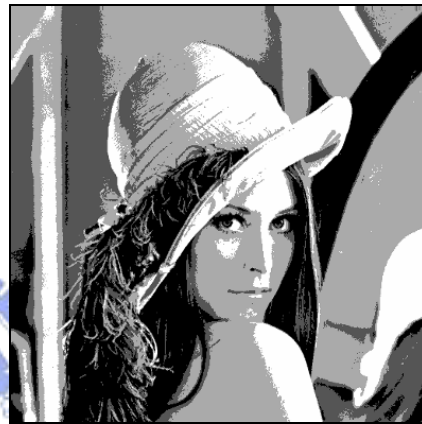
(**n**) denotes the order of the threshold been determined by the proposed method



(a) Original image



(b) The dark foreground object image derived by the proposed method



(c) The combined thresholded image derived by the proposed method



(d) The dark foreground object image derived by Between-class variance method



(e) The combined thresholded image derived by Between-class variance method





(f) The dark foreground object image derived by Entropy method



(g) The combined thresholded image derived by Entropy method



(f) The dark foreground object image derived by Minimum Error method



(g) The combined thresholded image derived by Minimum Error method

Figure 2.5. Result of segmenting the test image 2, "Lena" (image size=512×512)



For the document image analysis case, one should focus on the text extraction effect. Figure 2.6(a) shows a real-life, full-page, complex document image, “Advertisement”. It comprises several character strings, which are overlapped by a highly complex background, with several decorated objects and textures. By evaluating the processing results of this complex document image, the proposed method’s outstanding performance is more obviously demonstrated. The results of performing penta-level thresholding, for these four methods, are shown in Figure 2.6 and Table 2.7; the extraction of characters was the main focus in this experiment. As shown in Figure 2.6(b), after the proposed method had been performed, the dark characters were successfully and clearly segmented from the decorated non-text objects, with different illuminations and background textures, which they overlap. The other three methods were then performed, under the same number of thresholding levels as determined by the proposed method. Figure 2.6(d), (f) and (h) show the darkest thresholded image, which are expected to retain the characters, obtained by the other three fixed-level methods. However, only the between-class variance method can provide an acceptable thresholded result for characters extraction, as shown in Figure 2.6(d); the characters in Figure 2.6(f), obtained by the entropy method, suffer serious broken strokes, and, as shown in Figure 2.6(h), the minimum error method fails to separate characters from background objects. To observe the combined thresholded images derived by three methods, Figure 2.6(c), (e), (g) and (i) utilize the gray values (0, 63, 127, 191 and 255) as representative gray levels of the divided intervals of the thresholded images. Notably, the minimum error criterion method cannot obtain acceptable results, on multilevel thresholding, in most tests in this study. This is because the minimum error method assumes the histogram is comprised of a desired number ( $k$ ) of Gaussian distributions; hence, it fails when the actual histogram of the image cannot match this assumption. For the computational issues, as presented in Table 2.7, we can find that, although the uniformity score of the proposed automatic multilevel thresholding method is not the highest among all methods, it yields the best thresholded image for the extraction of

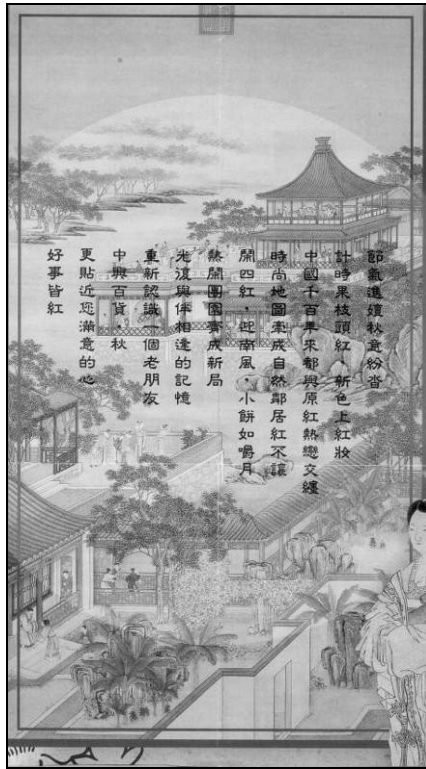
characters. Moreover, the computation time of the proposed automatic multilevel thresholding method just mildly increases for penta-level thresholding, while the computation costs of all other three fixed-level methods of the criterion functions tremendously increase. Herein, from Table 2.7, we can see that the proposed combinatorial scheme saves a very great deal of computation timings compared to the corresponding original methods of these criterion functions, and this proposed combinatorial scheme can also provide faster computation performance for the entropic method than Fleury *et. al.*'s running entropic method.

Table 2.7. Experimental data of performing thresholding on Figure 2.6(a), "Advertisement", by our automatic multilevel thresholding method (AMT), Between-class variance method (BCV), Entropy method (ENT), and Minimum Error method (ME)

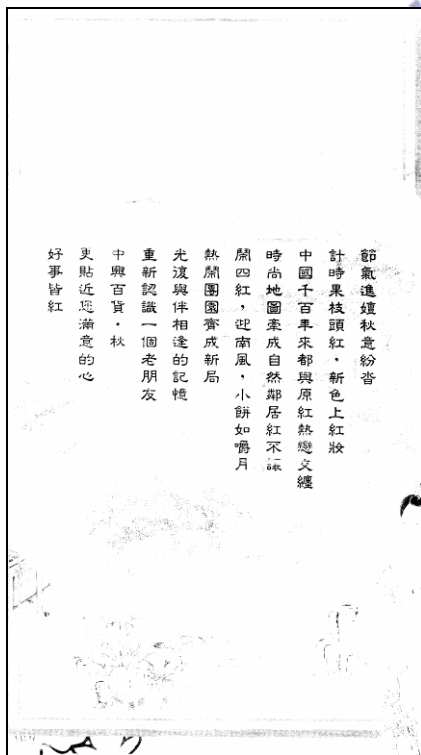
Criterion Method	Threshold levels	Selected threshold values	Uniformity measure	Computation time (seconds)		
				Original method	with Fleury <i>et al.</i> 's method	with our combinatorial scheme
AMT	5	59(4), 98(2), 148(1), 187(3)	0.925	0.00094	-	-
BCV	5*	82, 127, 163, 194	0.936	10342.68	-	31.641
ENT	5*	44, 89, 135, 180	0.913	29805.37	90.906	67.219
ME	5*	194, 195, 196, 197	0.412	16408.43	-	111.257

"\*" denotes the number of thresholding levels is given by supervision

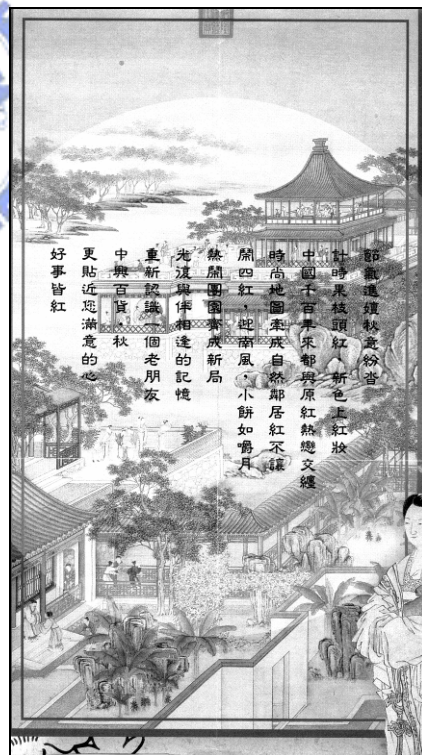
(n) denotes the order of the threshold been determined by the proposed method



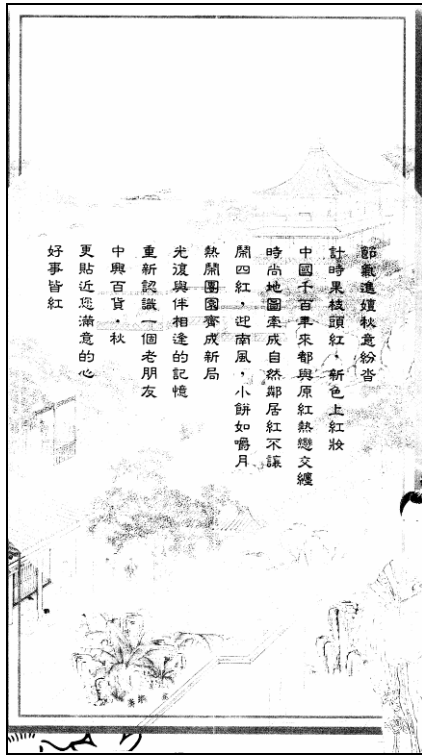
(a) Original image



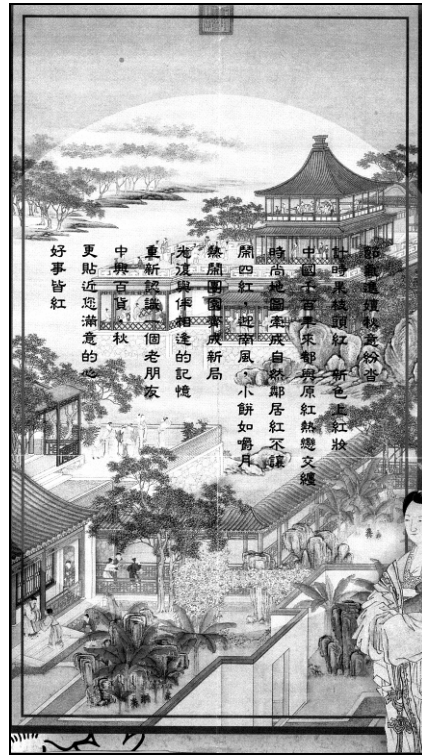
(b) The dark foreground object image derived by the proposed method



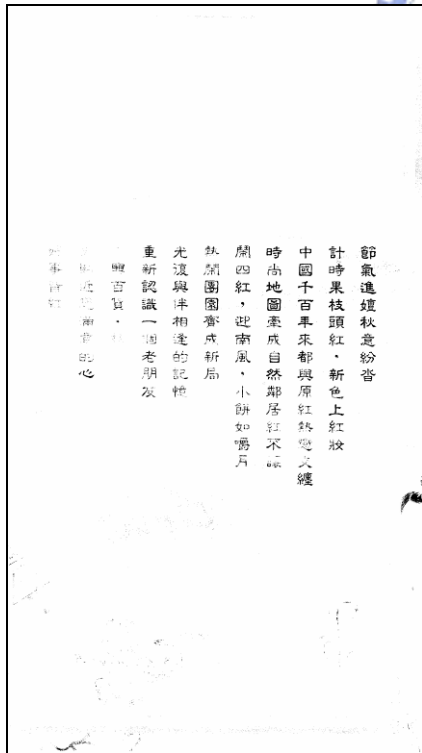
(c) The combined thresholded image derived by the proposed method



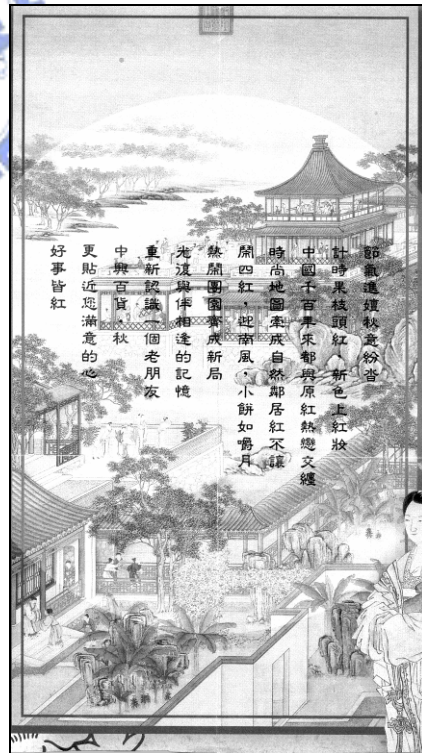
(d) The dark foreground object image derived by Between-class variance method



(e) The combined thresholded image derived by Between-class variance method

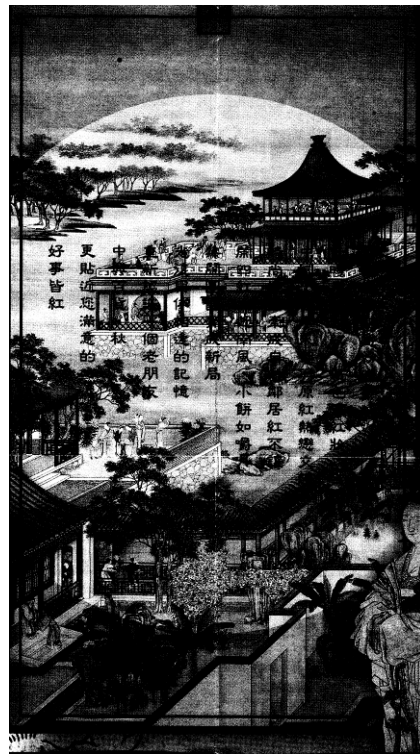
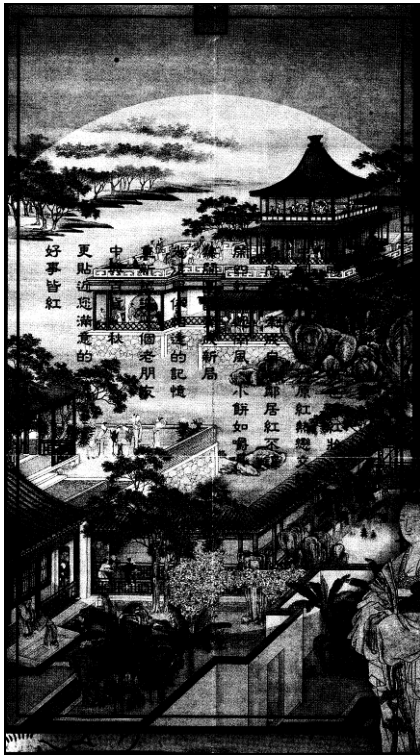


(f) The dark foreground object image derived by Entropy method



(g) The combined thresholded image derived by Entropy method

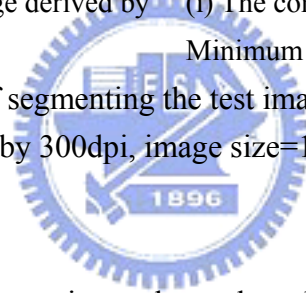




(h) The dark foreground object image derived by Minimum Error method

(i) The combined thresholded image derived by Minimum Error method

Figure 2.6. Result of segmenting the test image 3, “Advertisement”  
(scanned by 300dpi, image size=1842×3310)



As shown by the above experimental results, the proposed automatic multilevel thresholding method can automatically determines a suitable number of thresholding levels and selects appropriate threshold values that yield fine visual effects of thresholded images, with the segmented objects being sufficiently distinct for further processing, and the proposed objective uniformity evaluation also demonstrates its effectiveness. As well as the proposed combinatorial scheme can significantly reduce the computation timings of performing multilevel thresholding process in fixed-level cases based on the above-mentioned criterion functions.

As for the computational issues of the proposed combinatorial scheme, its computation timings in the tri-level thresholding cases are mostly dominated by the computation of construction of necessary class statistics tables. Therefore, the advantage on savings of the

computation timings of the proposed scheme becomes more obvious as the number of desired thresholds is growing larger. Additionally, it is notably that when the original computation methods of the criterion functions are performed, the computation time of the maximum entropy criterion is larger than the minimum error criterion; but when the proposed scheme is applied, the minimum error criterion's computation time is contrarily larger than that of the maximum entropy criterion. This is because when the proposed scheme is applied, the computation timings of the criterion functions are dominated by calculation of their non-summative terms, and hence the calculation of two logarithms and one square root evaluation for each class contribution of the minimum error criterion obviously costs more computation timing than the other criterion functions.

Notably, it can also be found that, under the same thresholding levels, the obtained thresholded results are as well as those obtained by the between-class variance method, but the computation time of the proposed automatic multilevel thresholding method is substantially faster. The computational time of the proposed automatic thresholding method increases moderately, when more thresholds are necessary as the complexity of the image increases. In contrast, the fixed-level criterion-based thresholding methods have to perform evaluations on all possible threshold sets. Thus, when the desired number of thresholding levels increases, their computational time increases exponentially; even though this is significantly improved by performing the proposed combinatorial scheme. From the experimental results, it can be concluded that the proposed automatic multilevel thresholding method performs satisfactorily in computational efficiency issues, as well as both subjective and objective evaluations of thresholded results, on all the tests in this study.

### **Chapter 3. MULTI-PLANE SEGMENTATION APPROACH FOR COMPLEX DOCUMENT IMAGES**

This chapter presents a new method, namely the multi-plane segmentation approach, for segmenting and extracting textual objects from various real-life complex document images. The proposed multi-plane segmentation approach first decomposes the document image into distinct object planes to extract and separate homogeneous objects including textual regions of interest, non-text objects such as graphics and pictures, and background textures. This process consists of two stages - automatic localized histogram multilevel thresholding, and multi-plane region matching and assembling. Then a text extraction procedure is applied on the resultant planes to detect and extract textual objects with different characteristics in the respective planes. The proposed approach processes document images regionally and adaptively according to their respective local features. Hence detailed characteristics of the extracted textual objects, particularly small characters with thin strokes, as well as gradational illuminations of characters, can be well-preserved. Moreover, this way also allows background objects with uneven, gradational, and sharp variations in contrast, illumination, and texture to be handled easily and well. Experimental results on real-life complex document images demonstrate that the proposed approach is effective in extracting textual objects with various illuminations, sizes, and font styles from various types of complex document images.

### 3.1 Introduction

Extraction of textual information from document images provides many useful applications in document analysis and understanding, such as optical character recognition, document retrieval, and compression [4][5]. To-date, many techniques were presented for extracting textual objects from monochromatic document images [29]-[32]. In recent years, owing to advances in multimedia publishing and printing technology have led to an increasing number of real-life documents in which stylistic character strings are printed with pictorial, textured, and decorated objects and colorful, varied background components. However, most of current approaches cannot work well for extracting textual objects from real-life complex document images. Compared to monochromatic document images, text extraction in complex document images brings many difficulties associated with the complexity of background images, variety and shading of character illuminations, superimposing characters with illustrations and pictures, as well as other decorated background components. As a result, there is an increasing demand for a system that is able to read and extract the textual information printed on pictorial and textured regions in both colored images as well as monochromatic main text regions.

Several newly developed global thresholding methods are useful in separating textual objects from non-uniform illuminated document images. Liu and Srihari [71] proposed a method based on texture features of character patterns, while Cheriet et al. [23] proposed a recursive thresholding algorithm extended from Otsu's optimal criterion [12]. Both these methods classify pixels in the original image as foreground objects (particularly textual objects of interest) or as background ones according to their gray intensities in a global view, and are attractive because of computational simplicity. However, binary images obtained by global thresholding methods are subject to noise and distortion, especially because of uneven illumination and the spreading effect caused by the image scanner. Parker [72] proposed a



local gray intensity gradient thresholding technique which is effective for extracting textual objects in badly illuminated document images. Because this method is based on the assumption of binary document images, its application is limited to extracting character objects from backgrounds no more complex than monotonically changing illuminations. A local and adaptive binarization method was proposed by Ohya et al. [73]. This method divides the original image into blocks of specific size, determines an optimal threshold associated with each block to be applied on its center pixel, and uses interpolation for determining pixel-wise thresholds. It can effectively extract textual objects from images with complex backgrounds on condition that the illuminations are very bright compared with those of the textual objects.

Some other methods support a different viewpoint for extracting texts by modeling the features of textual objects and backgrounds. Kamel and Zhao [74] proposed the logical level technique to utilize local linearity features of character strokes, while Venkateswarlu and Boyle's average clustering algorithm [75] utilizes local statistical features of textual objects. These methods apply symmetric local windows with a pre-specified size, and several pre-determined thresholds of prior knowledge on the local features, and so that characters with stroke widths that are substantially thinner or thicker than the assumed stroke width, or characters in varying illumination contrasts with backgrounds may not be appropriately extracted. Ye et al.'s hybrid extraction method [76] integrates global thresholding, local thresholding and the double-edge stroke feature extraction techniques to extract textual objects from document images with different complexities. The double-edge technique is useful in separating characters whose stroke widths are within a specified size from uneven backgrounds. Some recently presented methods [77][78] utilized the sub-image concepts to deal with the extraction of textual objects under different illumination contrasts with backgrounds. Dawoud and Kamel's [77] proposed a multi-model subimage thresholding

method that considers a document image as a collection of pre-determined regions, i.e. sub-images, and then textual objects contained in each sub-image are segmented using statistical models of the gray-intensity and stroke-run features. In Amin and Wu's multi-stage thresholding approach [78], Otsu's global thresholding method is firstly applied, and then a connected-component labeling process is applied on the thresholded image to determine the sub-images of interest, and these sub-images are undergone another thresholding process to extract textual objects. The extraction performance of the above two methods highly rely on the adequate determination of sub-image regions. Thus, in case of the textual objects overlapped on pictorial or textured backgrounds in poor and varying contrasts, suitable sub-images are hard to be determined to yield satisfactory extraction results.

Since most textual objects show sharp and distinctive edge features, methods based on edge information [33]-[36] have been developed. Such methods utilize an edge detection operator to extract the edge features of textual objects, and then use these features to extract texts from document images. Wu et al.'s Textfinder system [34] uses nine second-order Gaussian derivative filters to obtain edge-feature vectors of each pixel at three different scales, and applies the K-means algorithm on these edge-feature vectors to identify corresponding textual pixels. Hasan and Karam [35] introduced a method that utilizes a morphological edge extraction scheme, and applies morphological dilation and erosion operations on the extracted closure edges to locate textual regions. Edge information can also be treated as a measure for detecting the existence of textual objects in a specific region. In Pietikainen and Okun's work [36], edge features extracted by the Sobel operator are divided into non-overlapping blocks, and then these blocks are classified as text or non-text according to their corresponding values of the edge features. Such edge-based methods are capable of extracting textual objects in different homogeneous illuminations from graphic backgrounds. However, when the textual objects are adjoined or touched with graphical objects, texture patterns, or

backgrounds with sharply varying contours, edge-feature vectors of non-text objects with similar characteristics may also be identified as textual ones, and thus the characters in extracted textual regions are blurred by those non-text objects. Moreover, when textual objects do not have sufficient contrasts with non-text objects or backgrounds to form sufficiently strong edge features, such textual objects cannot be easily extracted with edge-based methods.

In recent years, several color-segmentation-based methods for text extraction from color document images have been proposed. Zhong et al. [37] proposed two methods and a hybrid approach for locating texts in color images, such as in CD jackets and book covers. The first method utilizes a histogram-based color clustering process to obtain connected-components with uniform colors, and then several heuristic rules are applied to classify them as textual or non-textual objects, as well as the second method locates textual regions based on their distinctive spatial variance. To more effectively detect textual regions, both methods are combined into a hybrid approach. Although the spatial variance method still suffers from drawbacks of the edge-based methods mentioned previously, the color connected-component method moderately compensates for these drawbacks. However, this approach still cannot provide acceptable results when the illuminations or colors of characters in large textual regions are shaded. Several recent techniques utilize color clustering or quantization approaches for determining the prototype colors of documents so as to facilitate the detection of character objects in these separated color planes. In Jain and Yu's work [38], a color document is decomposed into a set of foreground images on the RGB color space using a bit-dropping quantization and the single-link color clustering algorithm. Strouthopoulos et al.'s adaptive color reduction technique [39] utilizes an unsupervised neural network classifier and a tree-search procedure to determine prototype colors. Some alternative color spaces are also adopted to determine prototype colors for finding textual objects of interest.

Yang and Ozawa [40] make use of the HSI color space to segment homogenous color regions for extracting bibliography information from book covers, while Hase et al. [41] apply a histogram-based approach to select prototype colors on the CIE *Lab* color space to obtain textual regions. However, most of the aforementioned methods have difficulties in extracting texts which are embedded in complex backgrounds or that touch other pictorial and graphical objects. This is because the prototype colors are determined in a global view, so that appropriate prototype colors cannot be easily selected for distinguishing textual objects from those touched pictorial objects and complex backgrounds without sufficient contrasts. Besides, such problems also limit the reliability of such methods on handling uneven illuminated document images.

In a word, extracting texts from complex document images involves several difficulties. These difficulties arise from the following properties of complex documents: 1) Character strings in complex document images may have different illuminations, sizes, and font styles, and are overlapped with various background objects with uneven, gradational, and sharp variations in contrast, illumination, and texture, such as illustrations, photographs, pictures or other background textures. 2) These documents may comprise small characters with very thin strokes as well as large characters with thick strokes, and may be influenced by image shading. An approach for extracting black texts from such complex backgrounds to facilitate compression of document images has been proposed in our previous work [79].

In this study, we propose an effective method, namely the *multi-plane segmentation approach*, for segmenting and extracting textual objects of interest from these complex document images, and resolving the above issues associated with the complexity of their backgrounds. The proposed multi-plane segmentation approach decomposes the document image into separate object planes by applying the two processing stages: automatic localized histogram multilevel thresholding, and multi-plane region matching and assembling. Figure 1

illustrates a flow diagram of the proposed approach. In the first stage, distinct objects embedded in block regions are decomposed into separate “sub-block regions” by performing the localized histogram multilevel thresholding process. Afterward, in the second stage, the multi-plane region matching and assembling process is applied on these obtained sub-block regions to arrange them into homogeneous object planes. Consequently, homogeneous objects including textual regions of interest, non-text objects such as graphics and pictures, and background textures are extracted and separated into distinct object planes. The text extraction procedure is then performed on the resultant planes to detect and extract the textual objects with different characteristics in the respective planes. The proposed approach processes document images regionally and adaptively by means of their local features. This way allows detailed characteristics of the extracted textual objects to be well-preserved, especially the small characters with thin strokes, as well as characters in gradational and shaded illumination contrasts. Thus, textual objects adjoined or touched with pictorial objects and backgrounds with uneven, gradational, and sharp variations in contrast, illumination, and texture can be handled easily and well. Experimental results demonstrate that the proposed approach is capable of extracting textual objects with different illuminations, sizes, and font styles from different types of complex document images. As compared with other existing techniques, our proposed approach exhibits feasible and effective performance on text extraction from various real-life complex document images.

The rest of this chapter is organized as follows. In Section 3.2 and Section 3.3, the two stages of the proposed multi-plane segmentation approach, the localized histogram multilevel thresholding procedure, and the multi-plane region matching and assembling process, are respectively presented. Then, a simple text extraction procedure is described in Section 3.4. Next, Section 3.5 illustrates comparative performance evaluation results.

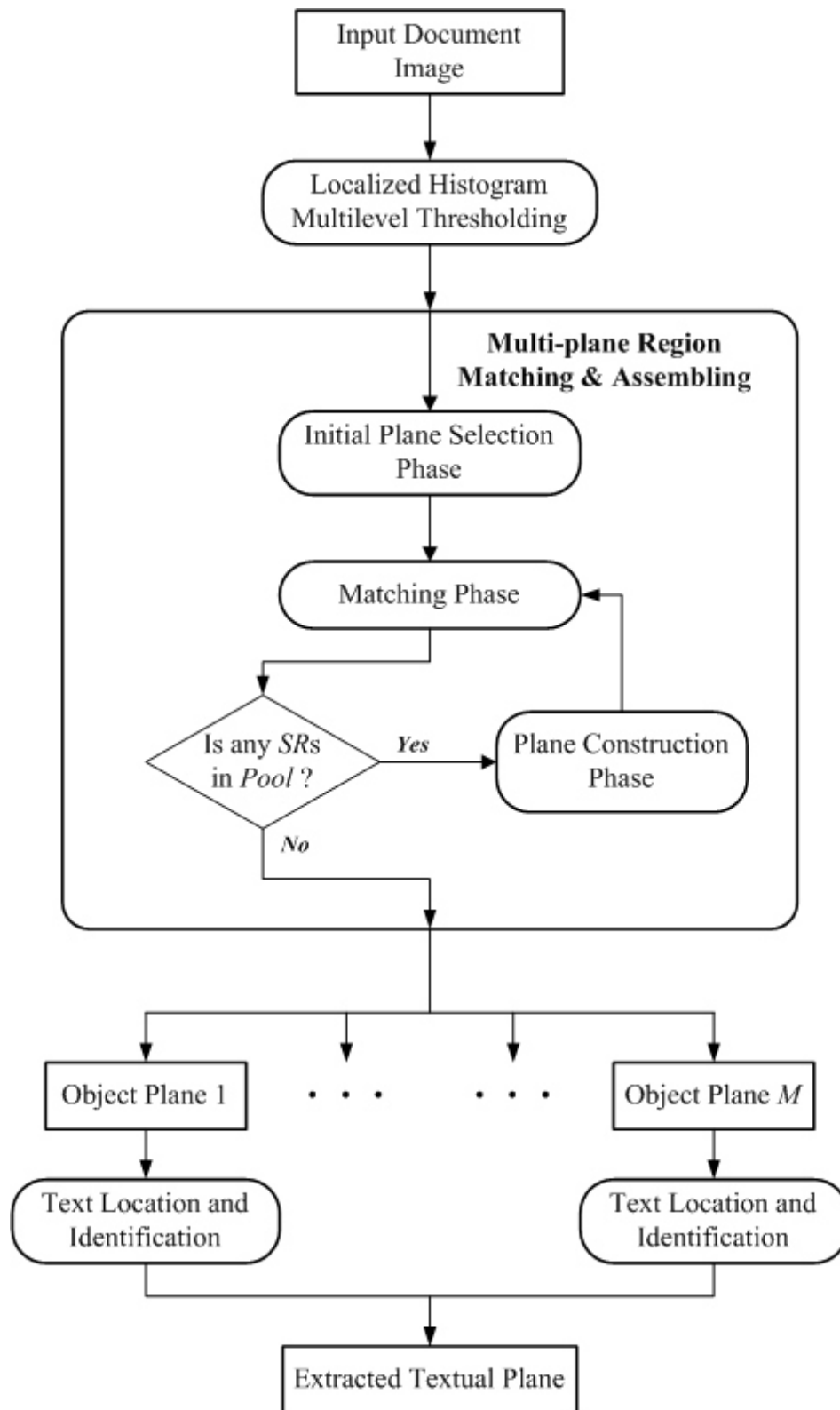


Figure 3.1. Block diagram of the proposed multi-plane segmentation approach

### 3.2 Localized Histogram Multilevel Thresholding

The multi-plane segmentation process, if necessary, begins by applying a color-to-grayscale transformation on the *RGB* components of image pixels in a color document image, to obtain its illumination image  $\mathbf{Y}$ . After the color transformation is performed, the illumination image  $\mathbf{Y}$  still retains the texture features of the original color image, as pointed out in [34], and thus the character strokes in their original color are still well-preserved. Then the obtained illumination image  $\mathbf{Y}$  will be divided into non-overlapping rectangular block regions with dimension  $M_H \times M_V$ , as shown in Fig. 2(a). Thus the mission is to extract objects with homogeneous characteristics from these rectangular block regions into different sub-block regions to facilitate further analysis in the following stage. For this purpose, the discriminant criterion is useful for measuring separability among the decomposed regions with different objects. Its application on bi-level global thresholding to extract foreground objects from the background was first presented by Otsu [12]. This method is ranked as the most effective bi-level threshold selection method [27][28]. However, when the number of desired thresholds increases, the computation needed to obtain the optimal threshold values is substantially increased and the search to achieve the optimal value of the criterion function is particularly exhaustive. To perform multilevel thresholding using discriminant criterion in a more computationally frugal way, Reddi *et al.* [22] presented an efficient method for thresholding an image into a specified number of objects. However, this method is adequately performed only when the image contains exactly three or fewer objects, and thus it cannot completely and effectively segment all objects of interest from a given image comprised of an uncertain number of objects.

Hence, an effective multilevel thresholding technique is needed for automatically determining the suitable number of thresholds for segmenting the block region into different decomposed object regions. By using the properties of discriminant analysis, we have

proposed an automatic multilevel global thresholding technique for image segmentation [56]. This technique extends and applies the concept of discriminant criterion on analyzing the separability among the gray levels in the image. It automatically determines the suitable number of thresholds, and utilizes a fast recursive selection strategy for selecting the optimal thresholds to segment the image into separate objects with similar characteristics in a computationally frugal way. In this study, we utilize this multilevel threshold selection technique and make necessary modifications to adapt it for segmenting block regions into different objects with similar characteristics. This process is described as follows.

Let  $f_g$  denote the observed frequencies (histogram) of gray illumination intensities of pixels in a certain block region, denoted by  $\mathfrak{R}$ , with a given gray illumination intensity  $g$ , and thus the total amount of pixels in  $\mathfrak{R}$  can be given by  $N = f_0 + f_1 + \dots + f_{U-1}$ , where  $U$  is the number of gray intensities in the histogram. Hence, the normalized probability of one pixel having a given gray intensity can be computed as,

$$P_g = \frac{f_g}{N}, \quad \text{and } P_g \geq 0, \quad \sum_{g=0}^{U-1} P_g = 1 \quad (3.1)$$

In order to segment textual objects, foreground objects and background components from a given region  $\mathfrak{R}$ , the pixels of  $\mathfrak{R}$  must be partitioned into a suitable number of classes. For multilevel thresholding, with  $n$  thresholds to partition the pixels in the region  $\mathfrak{R}$  into  $n+1$  classes, gray intensities of pixels in  $\mathfrak{R}$  are segmented by applying a threshold set  $\mathbf{T}$ , which is composed of  $n$  thresholds, where  $\mathbf{T} = \{t_k \mid k=1, \dots, n\}$ . These classes are represented by  $C_0 = \{0, 1, \dots, t_1\}$ ,  $\dots$ ,  $C_k = \{t_k + 1, t_k + 2, \dots, t_{k+1}\}$ ,  $\dots$ ,  $C_n = \{t_n + 1, t_n + 2, \dots, U - 1\}$ . The between-class variance of the gray intensities of pixels in the region  $\mathfrak{R}$ , denoted by  $v_{BC}$ , an effective criterion for evaluating segmentation results, is utilized to measure the separability among all classes, and is expressed as,



$$v_{BC}(\mathbf{T}) = \sum_{k=0}^n w_k (\mu_k - \mu_{\mathfrak{R}})^2, \quad \text{and} \quad \mu_{\mathfrak{R}} = \sum_{g=0}^{U-1} g P_g \quad (3.2)$$

where  $\mu_{\mathfrak{R}}$  is the overall mean of the gray intensities in  $\mathfrak{R}$ . Then the within-class variance and total variance, denoted by  $v_{WC}$  and  $v_{\mathfrak{R}}$ , respectively, of all segmented classes of gray intensities are respectively computed as,

$$v_{WC}(\mathbf{T}) = \sum_{k=0}^n w_k \sigma_k^2, \quad v_{\mathfrak{R}} = \sum_{g=0}^{U-1} (g - \mu_{\mathfrak{R}})^2 P_g \quad (3.3)$$

where  $w_k$  is the cumulative probability mass function of class  $C_k$ ;  $\mu_k$  and  $\sigma_k^2$  represent the mean and the standard deviation of the gray intensities in class  $C_k$ , respectively. They are defined as,

$$w_k = \sum_{g=t_k+1}^{t_{k+1}} P_g, \quad \mu_k = \frac{\sum_{g=t_k+1}^{t_{k+1}} g P_g}{w_k}, \quad \text{and} \quad \sigma_k^2 = \frac{\sum_{g=t_k+1}^{t_{k+1}} P_g (g - \mu_k)^2}{w_k} \quad (3.4)$$

Here, a dummy threshold  $t_0 = 0$  is utilized for the sake of convenience in simplifying the expression of equation terms.

The aforementioned criterion functions can be considered as a measure of separability among all existing classes decomposed from the original region  $\mathfrak{R}$ . We utilize this concept as a criterion of automatic segmentation of objects in a region, denoted by the ‘‘separability factor’’ –  $\mathcal{SF}$  in this study, which is defined as,

$$\mathcal{SF} = \frac{v_{BC}(\mathbf{T})}{v_{\mathfrak{R}}} = 1 - \frac{v_{WC}(\mathbf{T})}{v_{\mathfrak{R}}} \quad (3.5)$$

where  $v_{\mathfrak{R}}$  serves as the normalization factor in this equation. The  $\mathcal{SF}$  value represents the separability measure among all existing classes, and lies within the range  $\mathcal{SF} \in [0,1]$ ; the

lower bound is approached when the region  $\mathfrak{R}$  comprises a uniform gray intensity, while the upper bound is achieved when the region  $\mathfrak{R}$  consists of exactly  $n+1$  gray intensities. The objective is to maximize the  $SF$  value so as to optimize the segmentation result. This concept is supported by the property that  $v_{\mathfrak{R}}$  is equivalent to the sum of  $v_{BC}$  and  $v_{WC}$ . By observing the terms comprising  $v_{WC}(\mathbf{T})$ , if the gray intensities of the pixels belonging to most existing classes are widely distributed, i.e. the contribution values of their class variances  $\sigma_k^2$  are large, then the value of the corresponding  $SF$  measure becomes low. Accordingly, when  $SF$  approximates 1.0, all resultant classes of gray intensities  $C_k$  ( $k = 0, \dots, n$ ), which are decomposed from the original region  $\mathfrak{R}$ , are ideally and completely separated.

Therefore, based on this efficient discriminant criterion, an automatic multilevel thresholding can be applied for recursively segmenting the block region  $\mathfrak{R}$  into different objects of homogeneous illuminations, regardless of the number of objects and image complexity of the region  $\mathfrak{R}$ . It can be performed until the  $SF$  measure is large enough to show that the appropriate discrepancy among the resultant classes has been obtained. Through these aforementioned properties, this objective can be achieved by minimizing the total within-class variance  $v_{WC}(\mathbf{T})$ . This can be achieved by the scheme that selects the class with the maximal contribution ( $w_k \sigma_k^2$ ) to the total within-class variance for performing the bi-class partition procedure in each recursion. Thus, the  $SF$  measure will most rapidly reach the maximal increment to satisfy sufficient separability among the resultant classes of pixels. As a result, objects with homogeneous gray intensities will be well-separated.

The class having the maximal contribution of within-class variance  $w_k \sigma_k^2$  is denoted by  $C_p$ , and it comprises a subset interval of gray intensities represented by  $C_p: \{t_p+1, t_p+2, \dots, t_{p+1}\}$ . Then a simple effective *bi-class partition procedure*, as described in [56], is performed on each determined  $C_p$  in each recursion until the separability among all

classes becomes satisfactory, i.e. the condition where the  $SF$  measure approximates a sufficiently large value. The class  $C_p$  will be divided into two classes  $C_{p0}$  and  $C_{p1}$  by applying the optimal threshold  $t_S^*$  determined by the localized histogram based selection procedure as described in [56]. The resultant classes  $C_{p0}$  and  $C_{p1}$  comprise the subsets of gray intensities derived from  $C_p$  and can be represented as:  $C_{p0}: \{t_p+1, t_p+2, \dots, t_S^*\}$ , and  $C_{p1}: \{t_S^*+1, t_S^*+2, \dots, t_{p+1}\}$ . The threshold values determined by this recursive selection strategy is ensured to achieve maximum separation on the resultant segmented classes of gray intensities, and hence satisfactory segmentation results of objects can be accomplished by means of the smallest amount of thresholding levels.

Furthermore, if a region  $\mathfrak{R}$  is comprised of a set of pixels with homogeneous gray intensities, such as parts of a large background region, then it should not be partitioned and should keep its original components. For example, Figure 3.2(b) is the block region with these characteristics. Therefore, before performing the first partition procedure on the region  $\mathfrak{R}$ , an investigation of the homogeneity of  $\mathfrak{R}$  should be conducted in advance. This circumstance can be determined by evaluating the following two statistical features: 1) the bi-class  $SF$  measure, denoted as  $SF_b$ , which is the  $SF$  value obtained by performing the initial *bi-class partition procedure* on region  $\mathfrak{R}$ , i.e. the  $SF$  value associated with the determined threshold  $t_S^*$ ; and 2) the illumination variance,  $v_{\mathfrak{R}}$  of the pixels in the entire region  $\mathfrak{R}$ . According to the aforementioned properties, the  $SF_b$  value reflects the separability of the statistical distribution of gray intensities of pixels in the entire region  $\mathfrak{R}$ , and the lower the  $SF_b$  value is, the more indistinct the distribution is. The illumination variance  $v_{\mathfrak{R}}$  represents whether the distribution of gray intensities in  $\mathfrak{R}$  is widely dispersed or narrowly aggregated. Therefore, if both the  $SF_b$  and  $v_{\mathfrak{R}}$  features reveal lesser values, this means that the distribution of the region  $\mathfrak{R}$  is concentrated within a compact range, and thus the  $\mathfrak{R}$  comprises a set of homogeneous pixels that represent a simple object or parts thereof.

On the other hand, if  $S\mathcal{F}_b$  is small but  $v_{\mathfrak{R}}$  is large, the region  $\mathfrak{R}$  may consist of many indistinct object regions with low separability, and should undergo a repeated partition process to separate all objects. Based on the above-mentioned phenomenon, the following *homogeneity condition* is utilized for determining the situation where both the  $S\mathcal{F}_b$  and  $v_{\mathfrak{R}}$  features are sufficiently low:

$$S\mathcal{F}_b \leq \tau_{h0} \text{ , and } v_{\mathfrak{R}} \leq \tau_{h1} \quad (3.6)$$

where  $\tau_{h0}$  and  $\tau_{h1}$  are pre-defined thresholds. Therefore, if the homogeneity condition is satisfied, the region  $\mathfrak{R}$  is recognized as a homogeneous region, and does not need to undergo the partition process and hence keeps its pixels of homogeneous objects unchanged to be processed by the next stage. The values of the two thresholds  $\tau_{h0}$  and  $\tau_{h1}$  are chosen as 0.6 and 120, respectively. They are obtained from our experimental set of real-life complex documents to appropriately maintain the organized pixels of homogeneous block regions, and thus the integrity of associated larger characters or objects can be preserved well.

Based on the above-mentioned concepts, the localized automatic multilevel thresholding process is performed as the following steps:

**Step 1:** To begin, the illumination image  $\mathbf{Y}$  with size  $W_{img} \times H_{img}$  is divided into rectangular block regions  $\mathfrak{R}^{i,j}$  with dimension  $M_H \times M_V$ , as shown in Fig. 2(a). Here  $(i, j)$  are the location indices, and  $i = 0, \dots, N_H$  and  $j = 0, \dots, N_V$ , where  $N_H = (\lceil W_{img} / M_H \rceil - 1)$  and  $N_V = (\lceil H_{img} / M_V \rceil - 1)$ , which represent the numbers of divided block regions per row and per column, respectively. If the illumination image  $\mathbf{Y}$  cannot be exactly divided into rectangular regions such that all of them are with dimension  $M_H \times M_V$ , then the dimensions of the boundary rectangular regions, i.e. those in the right column and the bottom row, are with dimensions smaller than the other rectangular regions, as shown in Figure 3.3(b).

**Step 2:** For each block region  $\mathfrak{R}^{i,j}$ , compute the histogram of pixels in  $\mathfrak{R}^{i,j}$ , and then determine its associated illumination variance -  $v_{\mathfrak{R}}^{i,j}$  and the bi-class separability measure  $SF_b$ ; initially, there is only one class  $C_0^{i,j}$ ; let  $q$  represent the present amount of classes, and thus set  $q = 1$ . If the homogeneity condition, i.e. Eq. (3.6), is satisfied, then skip the localized thresholding process for this region  $\mathfrak{R}^{i,j}$  and go to step 7; else perform the following steps.

**Step 3:** Currently,  $q$  classes exist, having been decomposed from  $\mathfrak{R}^{i,j}$ . Compute the class probability  $w_k^{i,j}$ , the class mean  $\mu_k^{i,j}$ , and the standard deviation  $\sigma_k^{i,j}$ , of each existing class  $C_k^{i,j}$  of gray intensities decomposed from  $\mathfrak{R}^{i,j}$ , where  $k$  denotes the index of the present classes and  $n = 0, \dots, q-1$ .

**Step 4:** From all classes  $C_k^{i,j}$ , determine the class  $C_p^{i,j}$  which has the maximal contribution ( $w_k^{i,j} \sigma_k^{i,j 2}$ ) of the total within-class variance  $v_{WC}^{i,j}$  of  $\mathfrak{R}^{i,j}$ , to be partitioned in the next step in order to achieve the maximal increment of  $SF$ .

**Step 5:** Partition  $C_p^{i,j} : \{ t_p^{i,j} + 1, t_p^{i,j} + 2, \dots, t_{p+1}^{i,j} \}$  into two classes  $C_{p0}^{i,j} : \{ t_p^{i,j} + 1, t_p^{i,j} + 2, \dots, t_S^{i,j*} \}$ , and  $C_{p1}^{i,j} : \{ t_S^{i,j*} + 1, t_S^{i,j*} + 2, \dots, t_{p+1}^{i,j} \}$ , using the optimal threshold  $t_S^{i,j*}$  determined by the *bi-class partition procedure*. Consequently, the gray intensities of the region  $\mathfrak{R}^{i,j}$  are partitioned into  $q+1$  classes,  $C_0^{i,j}, \dots, C_{p0}^{i,j}, C_{p1}^{i,j}, \dots, C_{q-1}^{i,j}$ , and then let  $q = q+1$  update the record of the current class amount.

**Step 6:** Compute the  $SF$  value of all currently obtained classes using Eq. (3.5), if the *objective condition*,  $SF \geq \tau_{SF}$ , is satisfied, then perform the following Step 7; otherwise, go back to Step 3 to conduct further partition process on the obtained classes.

**Step 7:** Classify the pixels of the block region  $\mathfrak{R}^{i,j}$  into separate sub-block regions,  $SR^{i,j,0}, SR^{i,j,1}, \dots, SR^{i,j,q-1}$ , corresponding to the partitioned classes of gray illumination intensities,

$C_0^{i,j}$ ,  $C_1^{i,j}$ , ...,  $C_{q-1}^{i,j}$ , respectively, where the notation  $SR^{i,j,k}$  represents the  $k$ -th  $SR$  decomposed from the region  $\mathfrak{R}^{i,j}$ . Consequently, we obtain

$$\bigcup_{k=0}^{q-1} SR^{i,j,k} = \mathfrak{R}^{i,j}, \text{ and } SR^{i,j,k_1} \cap SR^{i,j,k_2} = \phi$$

Then, finish the localized thresholding process on  $\mathfrak{R}^{i,j}$  and go back to step 2 and repeat Steps 2~6 to process the remaining block regions; if all block regions have been processed, go to step 8.

**Step 8:** Terminate the segmentation process and deliver all obtained sub-block regions of the corresponding block regions.

The value of the separability measure threshold  $\tau_{SF}$  is chosen as 0.92 according to the experimental analysis described in [56] to yield satisfactory segmentation results on the block regions. With regard to the dimension parameters  $M_H \times M_V$  of each block region, in order for the localized thresholding process to be more adaptive on the steep gradation situation, and to extract the foreground objects in greater detail, smaller dimension block regions are desirable. In this way the small objects can be more clearly segmented, but at the cost of greater computation so as to yield the final results when performing the subsequent multi-plane region matching and assembling process. Therefore, suitable larger values of the parameters  $M_H$  and  $M_V$  should be chosen to moderately localize and accommodate the features of the allowable character size, and so that the contained textual objects in the images can be clearly segmented. Besides, given an input document image,  $M_H$  and  $M_V$  should also be automatically determined with respect to its scanning resolution  $RES$  (pixels per inch). Based on the analysis of typical characteristics of character sizes as described in [80] and the practice that typical resolutions for scanning most real-life document images

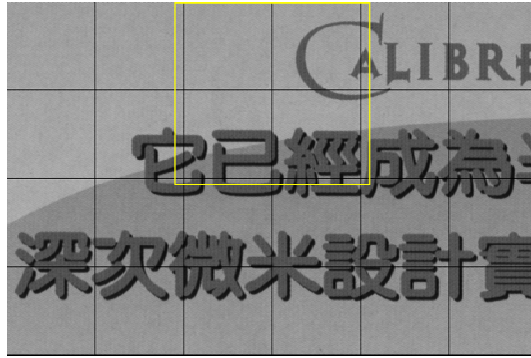
may range from 200 dpi to 600 dpi, the dimension parameters  $M_H$  and  $M_V$  are reasonably determined according to the typical allowable character sizes with respect to the scanning resolutions  $RES$ . They are obtained by,

$$M_H = M_V = 0.4 \cdot RES \quad (3.7)$$

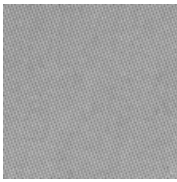
In this way, the dimension of each block region is determined as about  $10 \times 10 \text{ mm}^2$  in different scanning resolutions, such as  $M_H = M_V = 80$ ,  $M_H = M_V = 120$ , and  $M_H = M_V = 240$  in 200 dpi, 300 dpi, and 600 dpi scanning resolutions, respectively. These parameters are determined by conducting experiments involving numerous real-life document samples with various characteristics in our experimental set, so that nearly all foreground and textual objects in various document images can be appropriately separated in the preliminary experiments.

Figure 3.2 shows an example of performing the localized automatic multilevel thresholding procedure on several block regions. Figure 3.2(a) is part of the test sample image in Figure 3.3(a). Figure 3.2(b), (d), (g) and (l) are the four adjacent block regions chosen to illustrate the localized thresholding procedure. Figure 3.2(b) is a homogenous block region, and is properly detected by the homogenous conditions, and therefore its pixels are kept intact in Figure 3.2(c). Figure 3.2(d), (g) and (l) are the block regions comprised of two or three homogeneous objects. After the localized histogram multilevel thresholding procedure has been performed, different objects in these block regions are distinctly segmented into separate  $SRs$  from darkest to lightest, as shown in Figure 3.2(c), Figure 3.2(e)-(f), Figure 3.2(h)-(k), and Figure 3.2(m)-(o), respectively.

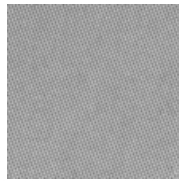




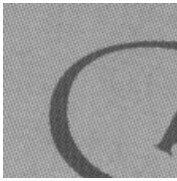
(a). Part of the partitioned block regions of the image “Calibre” in Figure 3.3(a), where the block regions enclosed by yellow ink are employed for the following examples of the localized multilevel thresholding procedure.



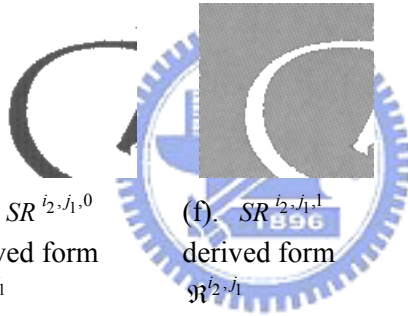
(b). The upper-left block region,  $\mathfrak{R}^{i,j_1}$



(c).  $SR^{i,j_1,0}$  derived form  $\mathfrak{R}^{i,j_1}$ , which is a homogenous block region

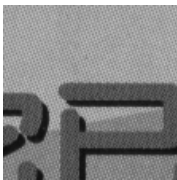


(d). The upper-right block region  $\mathfrak{R}^{i_2,j_1}$



(e).  $SR^{i_2,j_1,0}$  derived form  $\mathfrak{R}^{i_2,j_1}$

(f).  $SR^{i_2,j_1,1}$  derived form  $\mathfrak{R}^{i_2,j_1}$



(g). The Bottom-left block region  $\mathfrak{R}^{i,j_2}$

(h).  $SR^{i,j_2,0}$  derived form  $\mathfrak{R}^{i,j_2}$

(i).  $SR^{i,j_2,1}$  derived form  $\mathfrak{R}^{i,j_2}$

(j).  $SR^{i,j_2,2}$  derived form  $\mathfrak{R}^{i,j_2}$

(k).  $SR^{i,j_2,3}$  derived form  $\mathfrak{R}^{i,j_2}$



(l). Bottom-right block region  $\mathfrak{R}^{i_2,j_2}$

(m).  $SR^{i_2,j_2,0}$  derived form  $\mathfrak{R}^{i_2,j_2}$

(n).  $SR^{i_2,j_2,1}$  derived form  $\mathfrak{R}^{i_2,j_2}$

(o).  $SR^{i_2,j_2,2}$  derived form  $\mathfrak{R}^{i_2,j_2}$

Figure 3.2. Example of the results by the localized multilevel thresholding procedure



### 3.3 Multi-plane Region Matching and Assembling Process

After all block regions are segmented into several separate classes of pixels using the localized multilevel thresholding procedure introduced in the preceding section, various objects embedded or superimposed in different background objects and textures are respectively separated into relevant *SRs*, and then these obtained *SRs* are collected into a hypothetical “*Pool*” for further classifying and assembling process. Based on the contemporary concepts of region-based image segmentation [1][81], we present a multi-plane region matching and assembling process, to classify and assemble these obtained *SRs* to compose a set of object planes of homogeneous features, especially textual regions of interest. This proposed multi-plane region matching and assembling process is conducted by recursively performing the following three phases – the *initial plane selection phase*, the *matching phase* and the *plane construction phase*, as described in the following subsections.

To facilitate the matching and assembling process of the *SRs* obtained from the previous procedure, several concepts and definitions on statistical and spatial features of these *SRs* are introduced as follows. One *SR* may comprise several connected object regions of pixels decomposed from its associated block region  $\mathfrak{R}$ . Thus the pixels that belong to the object regions of a certain *SR* are said to be *object pixels* of this *SR*, while other pixels in this *SR* are *non-object pixels*. The set of the object pixels in one *SR* is defined as follows,

$$OP(SR^{i,j,k}) = \{g(SR^{i,j,k}, x, y) \mid \text{The pixel at } (x, y) \text{ is an object pixel in } SR^{i,j,k}\},$$

where  $g(SR^{i,j,k}, x, y)$  is the gray intensity of the pixel at location  $(x, y)$  in  $SR^{i,j,k}$ , and the range of  $x$  is within  $[0, M_H - 1]$  and  $y$  is within  $[0, M_V - 1]$ . As well as the total number of object pixels in  $SR^{i,j,k}$ , i.e. the amount of object pixels in  $OP(SR^{i,j,k})$ , is represented by

$$N_{op}(SR^{i,j,k}).$$

The concept *4-adjacent* refers to the situation in which each *SR* has four sides that border the top, the bottom, the left or the right boundary of its adjoining *SRs*. The *SRs* which are comprised of objects with homogeneous features are assembled to form an object plane, denoted by  $\mathcal{P}$ . An object plane  $\mathcal{P}$  represents a set of matching *SRs*, and for each pair of *SRs* in  $\mathcal{P}$ , there are some finite chains of *SRs* that connect them so that each successive pair of *SRs* is *4-adjacent*. We describe further details of the multi-plane region matching and assembling process in the following subsections.

### 3.3.1 Initial Plane Selection Phase

In the first processing phase, for the purpose of improving the speed and accurateness of the final convergence of the multi-plane region matching and assembling process, the mountain, or subtractive clustering technique [82][83] can be applied to determine the *SRs* with the most prominent and representative gray intensity features, and these *SRs* are selected as seeds to establish a set of initial planes. The mountain method is a fast, one-pass algorithm, which utilizes the density of features to determine the most representative feature points. Here we consider *SRs* as feature points in the mountain method.

First, given the localized multilevel thresholding process to segment the image into  $r$  *SRs* in the *Pool*, the mean  $\mu(SR^{i,j,k})$  associated with each of these obtained *SRs* is also obtained. Here  $\mu(SR^{i,j,k})$  is the mean of gray intensities of object pixels comprised by  $SR^{i,j,k}$ , and is equivalent to  $\mu_k^{i,j}$  obtained in the localized multilevel thresholding process. Then the region dissimilarity measure, denoted by  $D_{RM}$ , between two *SRs*, denoted by  $SR^{i_1,j_1,k_1}$  and  $SR^{i_2,j_2,k_2}$ , can be computed as,

$$D_{RM}(SR^{i_1,j_1,k_1}, SR^{i_2,j_2,k_2}) = \left\| \mu(SR^{i_1,j_1,k_1}) - \mu(SR^{i_2,j_2,k_2}) \right\|, \quad (3.8)$$

The range of the  $D_{RM}$  is within  $[0, 255]$ . The lower the computed value of  $D_{RM}$ , the stronger the similarity among two  $SR$ s. Therefore, the mountain function at an  $SR$  can be computed as,

$$M_0(SR^{i',j',k'}) = \sum_{\forall SR^{i,j,k} \in Pool} e^{-\alpha D_{RM}(SR^{i,j,k}, SR^{i',j',k'})} \quad (3.9)$$

where  $\alpha$  is a positive constant. A higher value of the mountain function reflects that  $SR^{i',j',k'}$  possesses more homogenous  $SR$ s in its vicinity. Therefore, it is sensible to select a  $SR^{i',j',k'}$  with a sufficiently high value of mountain function as a representative seed to establish a plane. Let  $M_0^*$  be the maximal value of the mountain function values, and  $SR_0^*$  be the  $SR$  whose mountain value is  $M_0^*$ :

$$M_0^*(SR_0^*) = \max_{SR^{i',j',k'}} [M_0(SR^{i',j',k'})] \quad (3.10)$$

Thus,  $SR_0^*$  is selected as the first seed of the first initial plane.

After computing the mountain function of each  $SR$  in the  $Pool$ , the following representative seeded  $SR$ s are determined by respectively destructing the mountains. Since the  $SR$ s whose gray intensity features close to  $SR_0^*$  also have high mountain values, it is necessary to eliminate the effects of the identified seeded  $SR$ s before determining the follow-up seeded  $SR$ s. Toward this purpose, the updating equation of the mountain function, after eliminating the previous  $(m-1)$ th seeded  $SR$  -  $SR_{m-1}^*$ , is computed by,

$$M_m(SR^{i',j',k'}) = M_{m-1}(SR^{i',j',k'}) - M_{m-1}^*(SR_{m-1}^*) e^{-\beta D_{RM}(SR^{i,j,k}, SR_{m-1}^*)} \quad (3.11)$$

where the parameter  $\beta$  determines the neighborhood radius that provide measurable reductions in the updated mountain function. Accordingly, through recursively performing the discount process of the mountain function given by Eq. (3.11), new suitable seeded  $SR$ s can be determined in the same manner, until the level of the current maximal  $M_{m-1}^*$  falls bellow a

certain level compared to that of the first maximal mountain  $M_0^*$ . The terminative criterion of this procedure is defined as,

$$\left(M_{m-1}^*/M_0^*\right) < \delta \quad (3.12)$$

where  $\delta$  is a positive constant less 1. Here the parameters are selected as  $\alpha = 5.4$ ,  $\beta = 1.5$  and  $\delta = 0.45$  as suggested by Pal and Chakraborty [84]. Consequently, this process converges to the determination of resultant  $N$  seeded  $SRs$ :  $\{SR_m^*, m = 0:N-1\}$ , and they are utilized to establish  $N$  initial planes for performing the following *matching phase*.

### 3.3.2 Matching Phase

In the matching phase, each of the unclassified  $SRs$  in the *Pool* is respectively compared their connectedness and similarity with the already existing planes, to determine its best belonging plane. For this purpose, we employ two forms of "matching grades", the *single-link matching grade*, and the *centroid-link matching grade*, to evaluate their related connectedness and similarity. The single-link matching grade examines the degree of connectedness between a pair of two neighboring  $SRs$ , an unclassified  $SR$  and its neighboring classified  $SRs$  that already have belonging planes; while the centroid-link matching grade represents the degree of similarity between an unclassified  $SR$  and an already existing plane. Then the two matching grades are combined to provide an effective criterion to determine the best belonging plane for this unclassified  $SR$  among the existing planes. If an unclassified  $SR$  can obtain its best belonging plane during the current matching phase recursion, i.e. this  $SR$  reflects sufficient similarity and connectedness with one of the existing planes after examining their mutual matching grade, then this  $SR$  is classified and assembled into this best belonging plane and removed from the *Pool* afterward; otherwise, if there is no suitable

matching plane for an unclassified  $SR$  at this time, then this  $SR$  will remain in the *Pool*. Since new potential planes will be created in the following recursion of the plane constructing phase,  $SR$ s which cannot find matching planes in the current matching phase recursion will be re-analyzed in subsequent recursions until their best matching planes are determined.

The single-link matching grade is utilized for examining the degree of connectedness between an unclassified  $SR^{i,j,k}$  and an already existent plane in a local manner. It is determined by applying a connectedness measure on  $SR^{i,j,k}$  and its 4-adjacent  $SR$ s already belonging to this existent plane. To facilitate the computation of the single-link matching grade, two measures for evaluating the continuity and similarity between two 4-adjacent  $SR$ s – the side-match measure, denoted as  $D_{SM}$ , and the region dissimilarity  $D_{RM}$ , as computed using Eq. (3.8), are employed. Then both  $D_{SM}$  and  $D_{RM}$  measures are jointly considered to determine the single-link matching grade of a pair of two 4-adjacent  $SR$ s.

The side-match measure -  $D_{SM}$ , which reveals the dissimilarity of the touching boundary between the two 4-adjacent  $SR$ s, is described as follows. Suppose that two  $SR$ s are 4-adjacent, they may have one of the two types of touching boundaries: 1) a vertical touching boundary mutually shared by two horizontally adjacent  $SR$ s, or 2) a horizontal boundary shared by two vertically adjacent  $SR$ s. For a pair of two horizontally adjacent  $SR$ s –  $SR^{i_1,j_1,k_1}$  on the left, and  $SR^{i_2,j_2,k_2}$  on the right, the pixel values on the rightmost side of  $SR^{i_1,j_1,k_1}$  and the leftmost side of  $SR^{i_2,j_2,k_2}$  can be described as:  $g(SR^{i_1,j_1,k_1}, M_H - 1, y)$  and  $g(SR^{i_2,j_2,k_2}, 0, y)$ , respectively.

The sets of object pixels on the rightmost side and the leftmost side of an  $SR$ , denoted by  $RS(SR^{i,j,k})$  and  $LS(SR^{i,j,k})$ , respectively, are defined as follows,

$$RS(SR^{i,j,k}) = \left\{ g(SR^{i,j,k}, M_H - 1, y) \mid g(SR^{i,j,k}, M_H - 1, y) \in OP(SR^{i,j,k}), \text{ and } 0 \leq y \leq M_V - 1 \right\},$$

$$\text{and } LS(SR^{i,j,k}) = \left\{ g(SR^{i,j,k}, 0, y) \mid g(SR^{i,j,k}, 0, y) \in OP(SR^{i,j,k}), \text{ and } 0 \leq y \leq M_V - 1 \right\}$$

To facilitate the following descriptions of the side-match features, the denotations of  $SR^{i_1, j_1, k_1}$  and  $SR^{i_2, j_2, k_2}$  are simplified as  $SR^l$  and  $SR^r$ , respectively. The vertical touching boundary of  $SR^l$  and  $SR^r$ , denoted as  $\mathbf{VB}(SR^l, SR^r)$ , is represented by a set of side connections formed by pairs of object pixels that are symmetrically connected on their associated rightmost and leftmost sides, and is defined as follows,

$$\mathbf{VB}(SR^l, SR^r) = \left\{ \left( g(SR^l, M_H - 1, y), g(SR^r, 0, y) \right) \mid g(SR^l, M_H - 1, y) \in \mathbf{RS}(SR^l), \text{ and } g(SR^r, 0, y) \in \mathbf{LS}(SR^r) \right\}$$

Similarly, in the case that  $SR^{i_1, j_1, k_1}$  and  $SR^{i_2, j_2, k_2}$  are vertically adjacent (suppose that  $SR^{i_1, j_1, k_1}$  is on the top and  $SR^{i_2, j_2, k_2}$  is on the bottom, and their denotations are also simplified as  $SR^t$  and  $SR^b$ , respectively), their horizontal touching boundary can be represented as,

$$\mathbf{HB}(SR^t, SR^b) = \left\{ \left( g(SR^t, x, M_v - 1), g(SR^b, x, 0) \right) \mid g(SR^t, x, M_v - 1) \in \mathbf{BS}(SR^t), \text{ and } g(SR^b, x, 0) \in \mathbf{TS}(SR^b) \right\}$$

where  $\mathbf{BS}(SR^t)$  and  $\mathbf{TS}(SR^b)$  represent the bottommost side and the topmost side of  $SR^t$  and  $SR^b$ , respectively, and are defined as,

$$\mathbf{BS}(SR^{i, j, k}) = \left\{ g(SR^{i, j, k}, x, M_v - 1) \mid g(SR^{i, j, k}, x, M_v - 1) \in \mathbf{OP}(SR^{i, j, k}), \text{ and } 0 \leq x \leq M_H - 1 \right\},$$

$$\text{and } \mathbf{TS}(SR^{i, j, k}) = \left\{ g(SR^{i, j, k}, x, 0) \mid g(SR^{i, j, k}, x, 0) \in \mathbf{OP}(SR^{i, j, k}), \text{ and } 0 \leq x \leq M_H - 1 \right\}$$

Also, the number of side connections of the touching boundary, i.e. the amount of connected pixel pairs in  $\mathbf{VB}(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2})$  or  $\mathbf{HB}(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2})$ , should also be considered for the connectedness of the two 4-adjacent SRs, and is denoted by  $N_{sc}(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2})$ . Therefore, based on the above-mentioned side-match features of two 4-adjacent SRs, the side-match measure,  $D_{SM}$ , of them when they are horizontally adjacent and vertically adjacent can be respectively computed as,

$$D_{SM}(SR^l, SR^r) = \frac{\sum_{(g(SR^l, M_H-1, y), g(SR^r, 0, y)) \in \mathbf{VB}(SR^l, SR^r)} \|g(SR^l, M_H-1, y) - g(SR^r, 0, y)\|}{N_{sc}(SR^l, SR^r)},$$

$$\text{and } D_{SM}(SR^t, SR^b) = \frac{\sum_{(g(SR^t, x, M_v-1), g(SR^b, x, 0)) \in \mathbf{HB}(SR^t, SR^b)} \|g(SR^t, x, M_v-1) - g(SR^b, x, 0)\|}{N_{sc}(SR^t, SR^b)} \quad (3.13)$$

The range of  $D_{SM}$  values is within  $[0, 255]$ . If the  $D_{SM}$  value of two 4-adjacent  $SR$ s is sufficiently low, then these two  $SR$ s are homogeneous with each other, and should belong to the same object plane  $\mathcal{P}$ .

Accordingly, the  $D_{SM}$  measure can reflect the continuity of two 4-adjacent  $SR$ s, and the  $D_{RM}$  value, as obtained by Eq. (3.8), assesses the similarity between them. Hence the homogeneity and connectedness of two 4-adjacent  $SR$ s can be evaluated by considering the dominant effect of the  $D_{SM}$  and the  $D_{RM}$  values. Therefore, based on the above definitions, the single-link matching grade of two 4-adjacent  $SR$ s, denoted by  $m_s$ , is determined as,

$$m_s(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2}) = \frac{\max(D_{SM}(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2}), D_{RM}(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2}))}{\max(\sigma(SR^{i_1, j_1, k_1}) + \sigma(SR^{i_2, j_2, k_2}), 1)} \quad (3.14)$$

where  $\sigma(SR^{i, j, k})$  is the standard deviation of gray intensities of all object pixels associated with  $SR^{i, j, k}$ , and is equivalent to  $\sigma_k^{i, j}$  obtained in the localized histogram multilevel thresholding process. Here the denominator term  $\max(\sigma(SR^{i_1, j_1, k_1}) + \sigma(SR^{i_2, j_2, k_2}), 1)$  in Eq. (3.14) serves as the normalization factor. Besides, it must be noted that the  $D_{SM}$  measure becomes invalid when  $N_{sc}(SR^{i_1, j_1, k_1}, SR^{i_2, j_2, k_2}) = 0$ . Therefore, in the determination of the single-link matching grade in Eq. (3.14), the  $D_{SM}$  can be disabled by setting the  $D_{SM}$  to zero using the “max” operation, so as to allow the  $D_{RM}$  having the dominant effect.

Next, we describe the centroid-link matching grade to assess the degree of similarity between  $SR^{i, j, k}$  and an already existing plane  $\mathcal{P}_q$  in a global manner. Let  $\mu(\mathcal{P}_q)$  and



$\sigma^2(\mathcal{P}_q)$  denote the mean and variance of the existing plane  $\mathcal{P}_q$ , respectively, and they are given by,

$$\mu(\mathcal{P}_q) = \frac{\sum_{SR_q^{i',j',k'} \in \mathcal{P}_q} N_{op}(SR_q^{i',j',k'}) \cdot \mu(SR_q^{i',j',k'})}{N_{op}(\mathcal{P}_q)}, \quad (3.15)$$

and

$$\sigma^2(\mathcal{P}_q) = \frac{\sum_{SR_q^{i',j',k'} \in \mathcal{P}_q} N_{op}(SR_q^{i',j',k'}) \cdot \|\mu(SR_q^{i',j',k'}) - \mu(\mathcal{P}_q)\|^2}{N_{op}(\mathcal{P}_q)} \quad (3.16)$$

where  $N_{op}(\mathcal{P}_q)$  denotes the amount of pixels in  $\mathcal{P}_q$ , and is given by,

$$N_{op}(\mathcal{P}_q) = \sum_{SR_q^{i',j',k'} \in \mathcal{P}_q} N_{op}(SR_q^{i',j',k'}) \quad (3.17)$$

Accordingly, the centroid-link matching grade of  $SR^{i,j,k}$  and  $\mathcal{P}_q$  can be computed by,

$$m_c(SR^{i,j,k}, \mathcal{P}_q) = \frac{\|\mu(SR^{i,j,k}) - \mu(\mathcal{P}_q)\|}{\max(\sigma(SR^{i,j,k}) + \sigma(\mathcal{P}_q), 1)} \quad (3.18)$$

If  $SR^{i,j,k}$  is finally determined to be merged into the plane  $\mathcal{P}_q$ , then the mean  $\mu(\mathcal{P}_q)$  and variance  $\sigma^2(\mathcal{P}_q)$  of  $\mathcal{P}_q$  should be updated after taking in  $SR^{i,j,k}$ . The new mean and variance of  $\mathcal{P}_q$  are respectively computed by,

$$\mu(\mathcal{P}_q^{new}) = \frac{(N_{op}(\mathcal{P}_q^{prev}) \cdot \mu(\mathcal{P}_q^{prev}) + N_{op}(SR^{i,j,k}) \cdot \mu(SR^{i,j,k}))}{(N_{op}(\mathcal{P}_q^{prev}) + N_{op}(SR^{i,j,k}))} \quad (3.19)$$

and

$$\sigma^2(\mathcal{P}_q^{new}) = \frac{\left[ N_{op}(\mathcal{P}_q^{prev}) \cdot \sigma^2(\mathcal{P}_q^{prev}) + N_{op}(SR^{i,j,k}) \cdot \left\| \mu(SR^{i,j,k}) - \mu(\mathcal{P}_q^{new}) \right\|^2 + N_{op}(\mathcal{P}_q^{prev}) \cdot \left\| \mu(\mathcal{P}_q^{new}) - \mu(\mathcal{P}_q^{prev}) \right\|^2 \right]}{\left( N_{op}(\mathcal{P}_q^{prev}) + N_{op}(SR^{i,j,k}) \right)} \quad (3.20)$$

where  $\mathcal{P}_q^{new}$  denotes the newly expanded plane  $\mathcal{P}_q$ , while  $\mathcal{P}_q^{prev}$  denotes the previous one; and  $\mu(\mathcal{P}_q^{new})$  and  $\sigma^2(\mathcal{P}_q^{new})$  represent the updated mean and variance of  $\mathcal{P}_q$ , respectively, while  $\mu(\mathcal{P}_q^{prev})$  and  $\sigma^2(\mathcal{P}_q^{prev})$  represent the previous ones.

The above-mentioned two matching grades are then combined to form a composite matching grade, denoted by  $\mathfrak{M}(SR^{i,j,k}, \mathcal{P}_q)$ , to complementarily evaluate the degree of connectedness and similarity of an unclassified  $SR$  and an already existing plane in both local and global manners. As a result, this composite matching grade can provide a more effective criterion for determining the best belonging plane for each of the unclassified  $SR$ s. Accordingly, in each recursion of the matching phase, each of the unclassified  $SR$ s, i.e.  $SR^{i,j,k}$  in the *Pool*, is analyzed by evaluating the composite matching grade of  $SR^{i,j,k}$  associated with each of its neighboring existent planes  $\mathcal{P}_q$ , to seek for the best matching plane into which  $SR^{i,j,k}$  should be grouped.

Since the evaluating process of the composite matching grades of  $SR^{i,j,k}$  is performed on its neighboring planes, a plane  $\mathcal{P}_q$  must have at least one of its own  $SR$ s 4-adjacent to  $SR^{i,j,k}$ , to compete for the belonging of  $SR^{i,j,k}$ . Here the  $SR$ s which have already been grouped into one of the current existing planes are denoted as  $SR_q^{i',j',k'}$ , where the subscript  $q$  represents that  $SR_q^{i',j',k'}$  belongs to the  $q$ -th plane  $\mathcal{P}_q$ . Therefore, to facilitate the computation of the composite matching grade of  $SR^{i,j,k}$  and a plane  $\mathcal{P}_q$ , the processing set

$AS(SR^{i,j,k}, \mathcal{P}_q)$  is utilized for storing the  $SR_{q,s}$  which are belonging to  $\mathcal{P}_q$  and 4-adjacent to  $SR^{i,j,k}$  as well, and is defined by,

$$AS(SR^{i,j,k}, \mathcal{P}_q) = \{SR_q^{i',j',k'} \in \mathcal{P}_q \mid SR_q^{i',j',k'} \text{ is 4-adjacent to } SR^{i,j,k}\}$$

Then the composite matching grade  $\mathfrak{M}$  of  $SR^{i,j,k}$  associated with the plane  $\mathcal{P}_q$ , which reveals how well  $SR^{i,j,k}$  matches with  $\mathcal{P}_q$ , can be determined by,

$$\mathfrak{M}(SR^{i,j,k}, \mathcal{P}_q) = w_c \left( m_c(SR^{i,j,k}, \mathcal{P}_q) \right) + w_s \left( \min_{\forall SR_q^{i',j',k'} \in AS(SR^{i,j,k}, \mathcal{P}_q)} m_s(SR^{i,j,k}, SR_q^{i',j',k'}) \right) \quad (3.21)$$

where  $w_c$  and  $w_s$  are the weighting factors to control the weighted contributions of the centroid-linkage and single-linkage strengths of the composite matching grade, respectively, and  $w_c + w_s = 1$ . By applying the weighting factors  $w_c$  and  $w_s$  in the composite matching grade, the centroid-linkage and single-linkage can be combined by taking advantage of their related strengths. Because textual regions mostly reveal obvious spatial connectedness, we reasonably strengthen the single-linkage weight of the composite matching grade, and thus the values of the weighting factors are chosen as  $w_c = 0.45$  and  $w_s = 0.55$ , respectively. Besides, if  $SR^{i,j,k}$  has no neighboring  $SR_{q,s}$  in  $\mathcal{P}_q$ , i.e.  $AS(SR^{i,j,k}, \mathcal{P}_q) = \emptyset$ , then  $\mathcal{P}_q$  is excluded from the consideration for matching with  $SR^{i,j,k}$ , that is, the evaluation process of their composite matching grade is skipped.

As a result, the best candidate belonging plane for  $SR^{i,j,k}$ , i.e. the plane having the lowest composite matching grade associated with  $SR^{i,j,k}$  among all existing planes, denoted by  $\mathcal{P}_m$ , can be determined by,

$$\mathfrak{M}(SR^{i,j,k}, \mathcal{P}_m) = \min_{\forall \mathcal{P}_q} \mathfrak{M}(SR^{i,j,k}, \mathcal{P}_q) \quad (3.22)$$

If the determined value of  $\mathfrak{M}(SR^{i,j,k}, \mathcal{P}_m)$  is too large,  $SR^{i,j,k}$  is not likely to have sufficient connectedness and similarity to  $\mathcal{P}_m$ . The following *matching criterion* is applied to check whether the currently selected candidate plane  $\mathcal{P}_m$  and  $SR^{i,j,k}$  are sufficiently matched, and then the suitability of  $SR^{i,j,k}$  for belonging to  $\mathcal{P}_m$  can be determined well. This matching criterion is defined as follows,

$$\mathfrak{M}(SR^{i,j,k}, \mathcal{P}_m) \leq \tau_m \quad (3.23)$$

where  $\tau_m$  is a predefined threshold which represents the acceptable tolerance of dissimilarity for  $SR^{i,j,k}$  to be grouped into  $\mathcal{P}_m$ . The matching criterion has a moderate effect on the number of resultant object planes, and the value choice of  $\tau_m = 1.2$  is experimentally determined to obtain sufficiently distinct planes and avoid excessive splitting of planes.

Accordingly, if  $SR^{i,j,k}$  and its associated  $\mathcal{P}_m$  satisfy the matching criterion, then  $SR^{i,j,k}$  is merged into  $\mathcal{P}_m$ , and removed from the *Pool*. If the matching criterion cannot be satisfied, this reflects that  $SR^{i,j,k}$  is distinct from all its existent adjoining planes, and there is no appropriate belonging plane for  $SR^{i,j,k}$  during this current matching phase recursion. Therefore,  $SR^{i,j,k}$  will remain in the *Pool*, until its suitable matching plane emerges or it begins its own plane in the following recursions of the plane construction phase. After a belonging determination has been made for  $SR^{i,j,k}$ , the matching process is in turn applied on the subsequent unclassified *SRs* in the *Pool*, until all the rest unclassified *SRs* have been processed one time in the current matching phase recursion.

### 3.3.3 Plane Construction Phase

After performing the previous matching phase recursion, if there are unclassified *SRs* remaining and the *Pool* is not drained as well, these unclassified *SRs* must be analyzed to

determine whether it is necessary to establish a new object plane to assemble the *SRs* with such features into this new plane to form another homogeneous object region. The textual regions and homogeneous objects may contain several connected regions. However, all similar and connected *SRs* comprised of them may not be completely assembled into appropriate planes in the previous matching phase recursion. This situation can be resolved by detecting and expanding the existent planes having sufficient close distances in gray intensity and spatial location features with unclassified *SRs* to take in such unclassified *SRs*. Therefore, in the next matching phase recursion, the follow-up unclassified *SRs* having homogeneous features with these planes can be successively assembled into them to prevent the split of homogeneous object regions. The plane construction phase determines whether to 1) create and initialize a new plane by selecting the unclassified *SR* “farthest away” from all existing planes as an initial seed, or 2) enlarge one suitably selected plane by merging one unclassified *SR* “closest” to this plane. The determination is made according to the analysis of the following gray intensity and spatial location features.

The dissimilarity between one unclassified *SR*, which is not adjoined to any one of currently existing planes in the previous matching phase recursion, and a certain object plane  $\mathcal{P}_q$ , can be determined by their value of the centroid-link matching grade, as computed by Eq. (3.18). Then the plane which is most similar to  $SR^{i,j,k}$  in gray intensity among all presently existing planes, i.e. the plane has the least value of the centroid-link matching grade associated with  $SR^{i,j,k}$ , denoted by  $\mathcal{P}_s(SR^{i,j,k})$ , is obtained by,

$$m_c(SR^{i,j,k}, \mathcal{P}_s(SR^{i,j,k})) = \min_{\forall \mathcal{P}_q} (m_c(SR^{i,j,k}, \mathcal{P}_q)) \quad (3.24)$$

Here  $m_c(SR^{i,j,k}, \mathcal{P}_s(SR^{i,j,k}))$  also represents the measure of the least dissimilarity between  $SR^{i,j,k}$  and all already existing planes. If  $SR^{i,j,k}$  can find a plane  $\mathcal{P}_s(SR^{i,j,k})$  having

sufficiently low dissimilarity with  $SR^{i,j,k}$  in gray intensity, and they are also locatively closed as well, then this condition reveals that  $SR^{i,j,k}$  is sufficiently homogeneous with  $\mathcal{P}_S(SR^{i,j,k})$ , even if it is not currently 4-adjacent to  $\mathcal{P}_S(SR^{i,j,k})$ .

To determine this situation, the locative distance between  $SR^{i,j,k}$  and a plane  $\mathcal{P}_q$ , denoted as  $D_E(SR^{i,j,k}, \mathcal{P}_q)$ , is computed by the Euclidean distance between  $SR^{i,j,k}$  and its closest  $SR_q$  among all  $SR_{q,S}$  associated with the plane  $\mathcal{P}_q$ ; and is determined as,

$$D_E(SR^{i,j,k}, \mathcal{P}_q) = \min_{\forall SR_q \in \mathcal{P}_q} D_e(SR^{i,j,k}, SR_q^{i',j',k'}), \quad (3.25)$$

$$\text{where } D_e(SR^{i,j,k}, SR_q^{i',j',k'}) = \sqrt{(i-i')^2 + (j-j')^2} \quad (3.26)$$

If  $SR^{i,j,k}$  and its  $\mathcal{P}_S(SR^{i,j,k})$  are homogeneous in gray intensity and also locatively close to each other, i.e. both  $m_c(SR^{i,j,k}, \mathcal{P}_S(SR^{i,j,k}))$  and  $D_E(SR^{i,j,k}, \mathcal{P}_S(SR^{i,j,k}))$  values are sufficiently low, then  $SR^{i,j,k}$  should join the plane  $\mathcal{P}_S(SR^{i,j,k})$ , rather than establish a new independent plane, so as to prevent a textual region or homogeneous object to be split into more than one plane. Otherwise, if no such planes are found, a new plane should be created to aggregate those  $SR$ s with distinct features.

In order to ensure that the new plane contains distinct features with the current existent planes, a scheme for selecting a suitable  $SR$  as the representative seed for constructing a new plane is given as follows,

$$m_c(SR_{NP}, \mathcal{P}_S(SR_{NP})) = \max_{\forall SR^{i,j,k} \in Pool} m_c(SR^{i,j,k}, \mathcal{P}_S(SR^{i,j,k})) \quad (3.27)$$

In this way, this determined seeded  $SR$ , denoted by  $SR_{NP}$ , is the one which is most dissimilar in gray intensities to any already existing planes, and thus  $SR_{NP}$  will begin its own new

plane to aggregate those  $SR$ s whose features are distinct from other existing planes but homogenous with  $SR_{NP}$ .

By means of the definitions given above, the plane construction phase is performed according to the following steps:

**Step 1:** First, the unclassified  $SR$ s which have sufficiently low  $m_c(SR^{i,j,k}, \mathcal{P}_S(SR^{i,j,k}))$  values are selected into the set  $SR_S$  using the following operation:

$$SR_S = \left\{ SR^{i,j,k} \in Pool \mid m_c(SR^{i,j,k}, \mathcal{P}_S(SR^{i,j,k})) \leq \tau_S \right\} \quad (3.28)$$

where  $\tau_S$  is a predefined threshold for determining whether one  $SR$  is sufficiently homogeneous with any one of existing planes. If none of the unclassified  $SR$ s satisfy the above condition to be selected into  $SR_S$ , i.e.  $SR_S = \phi$ , then go directly to **Step 3** for constructing a new plane; otherwise, perform the following **Step 2**.

**Step 2:** The set  $SR_S$  now contains the  $SR$ s which are significantly homogeneous with the already existing planes, but are not 4-adjacent with them, and thus remain unclassified in the previous matching phase recursion. The  $SR$  locatively nearest to its associated  $\mathcal{P}_S(SR^{i,j,k})$ , denoted by  $SR_p$ , is determined as follows,

$$D_E(SR_p, \mathcal{P}_S(SR_p)) = \min_{\forall SR^{i,j,k} \in SR_S} D_E(SR^{i,j,k}, \mathcal{P}_S(SR^{i,j,k})) \quad (3.29)$$

If  $SR_p$  and its associated  $\mathcal{P}_S(SR^{i,j,k})$  are sufficiently close to each other, i.e. the condition  $D_E(SR_p, \mathcal{P}_S(SR_p)) \leq \tau_L$  is satisfied, then  $SR_p$  is determined to be merged with  $\mathcal{P}_S(SR^{i,j,k})$  to enlarge its influential area on nearby  $SR$ s, and proceeds to **Step 4**. Otherwise, perform **Step 3** for constructing a new plane.

**Step 3:** The  $SR_{NP}$ , the  $SR$  most dissimilar to any currently existing planes, is determined by



using Eq. (3.27). Thus,  $SR_{NP}$  is employed as a seeded  $SR$  to establish a new plane  $\mathcal{P}_{new}$ , and then continues to perform **Step 4**.

**Step 4:** Finish the plane construction phase, and then conduct the next matching phase recursion.

The threshold  $\tau_s$  utilized in Eq. (3.28) moderately influences the number of resultant planes. If the value of  $\tau_s$  is low, then the number of resultant planes will be increased and a homogeneous region may be disjoined into more than one plane, although its influence on text extraction is not serious. If the value is large however, then the number of planes is reduced, and some objects may be merged to a certain degree. Reasonably, its value should be tighter than the value of  $\tau_m$ , which is utilized in the matching criterion in Eq. (3.23), to ensure that the determined  $SR_p$  is sufficiently homogeneous with its associated  $\mathcal{P}_s(SR^{i,j,k})$ , and thus  $\mathcal{P}_s(SR^{i,j,k})$  can appropriately attract homogeneous  $SR$ s near the extended influential area benefited from participation with  $SR_p$ . Therefore, in our experiments, the value of  $\tau_s$  is chosen as  $\tau_s = 0.8 \cdot \tau_m$ . Normally, text-lines or text-blocks usually occupy perceptible area of the image, and thus their width or height should be in appreciable proportion to those of the whole image. Therefore,  $\tau_L = \min(N_H, N_V)/4$  is used for experiments, where  $N_H$  and  $N_V$  are the numbers of block regions per row and per column, respectively.

### 3.3.4 Overall Process

Based on the three above-mentioned processing phases, the region matching and assembling process begins by applying the initial plane selection phase on all unclassified  $SR$ s in the *Pool* to determine the representative seeded  $SR$ s  $\{SR_m^*, m = 1:N\}$  for setting up

$N$  initial planes. Thus, the matching phase can subsequently apply on the rest of  $SR$ s in the  $Pool$  and the initial planes  $\mathcal{P}_0, \dots, \mathcal{P}_{N-1}$ . Then the matching phase and the plane construction phase are recursively performed in turns on the rest of unclassified  $SR$ s in the  $Pool$  and emerging planes, until each  $SR$  has been classified and associated with a particular plane, and the  $Pool$  is eventually cleared. As a result, after completing the multi-plane region matching and assembling process, the whole illumination image  $\mathbf{Y}$  is segmented into a set of separate object planes  $\{\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_{L-1}\}$ , each of which consists of homogenous objects with connected and similar features. Consequently, we obtain,

$$\bigcup_{q=0}^{L-1} \mathcal{P}_q = \mathbf{Y}, \quad \text{with} \quad \mathcal{P}_{q_1} \cap_{q_1 \neq q_2} \mathcal{P}_{q_2} = \phi$$

where  $L$  is the number of the resultant planes obtained.

We use Figure 3.3 as an example of the proposed multi-plane segmentation approach. The sample image in Figure 3.3(a) consists of three different colored textual regions printed on a varying and shaded background. Moreover, the black characters are superimposed on the white characters. First, as shown in Figure 3.3(b), the original image is transformed into grayscale, and is divided into block regions. The block regions are then processed by the localized multilevel thresholding procedure, and are decomposed into  $SR$ s according to the number of contained objects and their feature complexity. Next, these obtained  $SR$ s are processed by the initial plane selection phase, and four initial planes having most representative features are created, and they forms the resultant planes  $\mathcal{P}_0 - \mathcal{P}_4$ , as shown in Figure 3.3(c)-(f). Then the rest  $SR$ s in the  $Pool$  and the initial planes  $\mathcal{P}_0 - \mathcal{P}_4$  are analyzed and assembled by recursively applying the matching phase and the plane construction phase. As a result, there are seven major resultant object planes  $\mathcal{P}_0 - \mathcal{P}_6$  (while those insignificant planes are discarded) obtained after performing the multi-plane segmentation process, as depicted in

Figure 3.3(c)-(i). Within these planes, the planes  $\mathcal{P}_1$ ,  $\mathcal{P}_3$  and  $\mathcal{P}_4$  in Figure 3.3(d), (f), and (g), respectively, contains textual objects of interest.

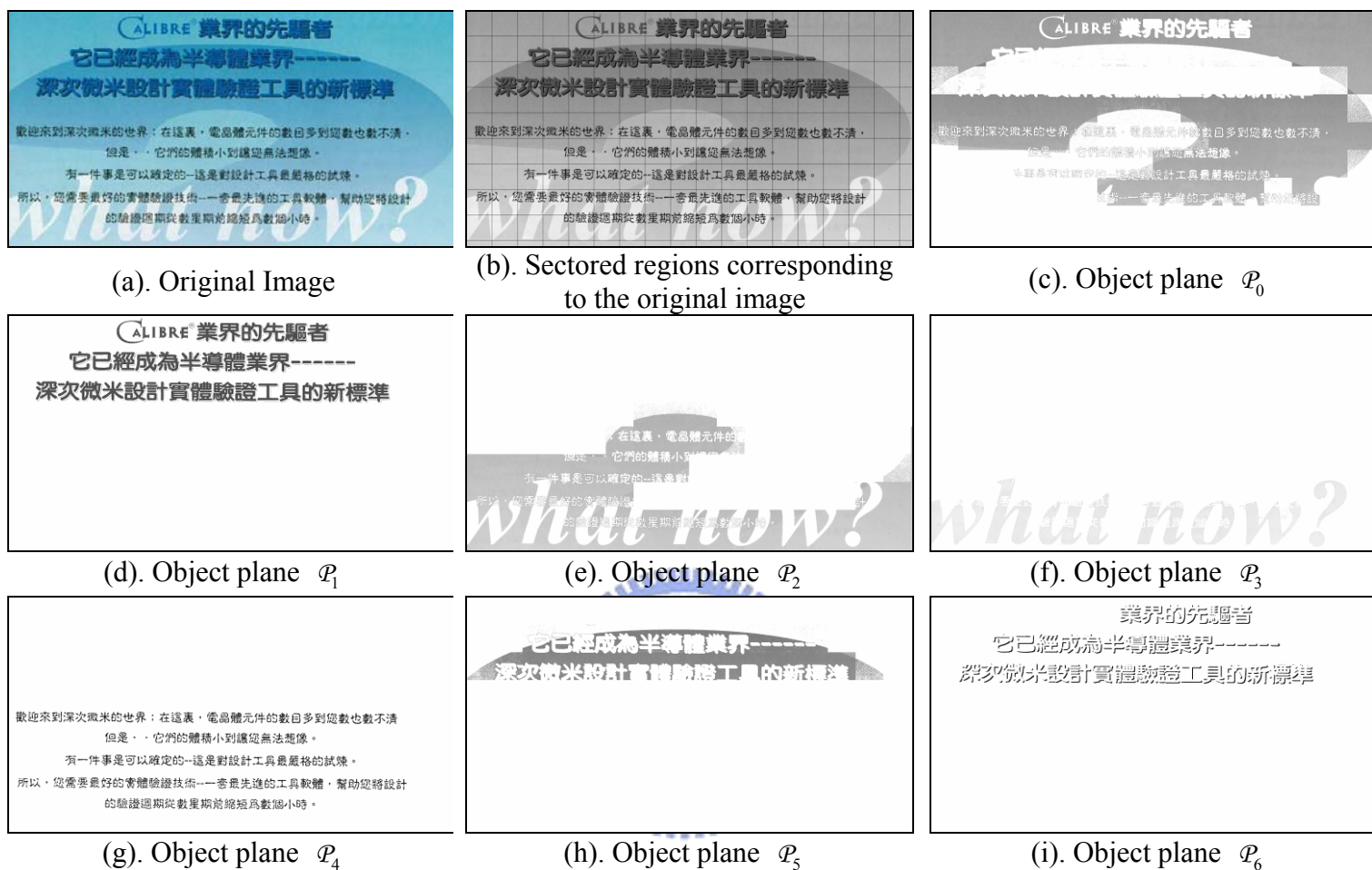


Figure 3.3. An example of the test image, “Calibre”, and the object planes obtained by the multi-plane segmentation (image size = 1929 x 1019)

Accordingly, the homogeneous objects in which all textual objects and background textures are segmented into several separate object planes can be effectively analyzed in detail. By observing these obtained planes, we can see that three textual regions with different characteristics are distinctly separated. Thus, extraction of textual objects from each binarized plane in which objects are well separated can be easily performed by some contemporarily developed text extraction techniques. The following section describes a simple procedure for locating and extracting textual objects from these resultant object planes.

### 3.4 Text Extraction

After performing the multi-plane segmentation, the entire image is decomposed into various object planes. Each object plane may consist of various considerable objects, such as characters, graphical and pictorial objects, background textures or other objects. Each individual object plane  $\mathcal{P}_q$  will be binarized by setting its object pixels to black, and setting other non-object pixels to white, and hence a “*binarized plane*”, denoted as  $\mathcal{BP}_q$ , is created corresponding to each plane  $\mathcal{P}_q$ . The text location and extraction process will be performed on each individual binary plane  $\mathcal{BP}_q$  to obtain the textual objects of interest. To obtain the character-like components from each binary plane  $\mathcal{BP}_q$ , a fast connected-component extraction technique [85] is first carried out to locate the connected-components of the black pixels in  $\mathcal{BP}_q$ . These connected-components may represent the character components, graphical and pictorial objects, or background textures. By extracting the connected-components, the location and dimension of each connected-component are obtained as well. The location and dimension of a connected-component are represented by the bounding box enclosing it.

In this study, based on the concepts of the recursive XY-cut techniques for connected-component projections [86], [87], we develop a recursive XY-cut spatial clustering process for grouping the connected-components into meaningful sets in each of the binarized planes. We are interested in looking for horizontal text-lines, and hence the XY-cut spatial clustering process is conducted to cluster the connected-components into several sets in horizontal direction. A resultant set of connected-components may comprise a character string, a larger graphical object, or a group of isolated background components inside the character strings. Each of these connected-component sets, denoted as **CS**, is then processed by the text

identification process to determine whether they are actual text-lines.

First, the definitions used in the text detection and extraction process are:

(a).  $C_i$  denotes a connected-component of the current  $\mathcal{BP}_q$ , and the bounding box which encloses  $C_i$  is denoted as  $B(C_i)$ .

(b).  $\mathbf{CS}_j$  denotes a group of connected-components,  $\mathbf{CS}_j = \{C_i : i = 0, 1, 2, \dots, N_{cc}(\mathbf{CS}_j)\}$ , where the number of connected-components contained in  $\mathbf{CS}_j$  is denoted as  $N_{cc}(\mathbf{CS}_j)$ , and the bounding box which enclose all the connected-components belonging to  $\mathbf{CS}_j$  is denoted as  $B(\mathbf{CS}_j)$ . A group  $\mathbf{CS}_j$  represents a preliminary general text line. Note that the simplified denotation  $B$  represents a bounding box of a connected-component  $C_i$ , or a group of connected-components  $\mathbf{CS}_j$ , respectively in the following descriptions for the spatial clustering process.

(c). The location of the bounding boxes of  $C_i$  or  $\mathbf{CS}_j$  employed in the spatial clustering process are their top and left coordinates, and they are denoted as  $t(B)$  and  $l(B)$ , respectively.

(d). The width and height of the bounding boxes are denoted as  $W(B)$  and  $H(B)$ , respectively.

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

$$D_h(B_i, B_j) = \max[l(B_i), l(B_j)] - \min[r(B_i), r(B_j)], \quad (3.30)$$

$$\text{and } D_v(B_i, B_j) = \min[b(B_i), b(B_j)] - \max[t(B_i), t(B_j)] \quad (3.31)$$

If the two bounding boxes are overlapping in the horizontal or vertical direction, then the

value of  $D_h(B_i, B_j)$  or  $D_v(B_i, B_j)$  will be a negative value.

(f). The measures of overlap between the horizontal and vertical projections of the two bounding boxes are defined as

$$P_h(B_i, B_j) = \frac{-D_h(B_i, B_j)}{\min[W(B_i), W(B_j)]} \quad (3.32)$$

$$\text{and } P_v(B_i, B_j) = \frac{-D_v(B_i, B_j)}{\min[H(B_i), H(B_j)]} \quad (3.33)$$

Based on the functions and notation defined above, the connected-component clustering process is detailed as follows. The XY-cut clustering process is conducted by recursively performing the horizontal clustering procedure, *X-cut* procedure, and the vertical clustering procedure, *Y-cut* procedure on the bounding boxes of the contained connected-components of the current processing binary plane  $\mathcal{BP}_q$ .

The horizontal clustering procedure - *X-cut*( $\mathbf{CS}_{in}$ ) (where the subscript “in” refers to “the original input group of connected-components”) is applied as follows:

*Step 1:* Project all the bounding boxes of the connected-components contained in  $\mathbf{CS}_{in}$  horizontally onto the vertical y-axis.

*Step 2:* Sort all the connected-components  $C_i$  in the  $\mathbf{CS}_{in}$  with respect to their corresponding  $t(C_i)$ , where all  $C_i \in \mathbf{CS}_{in}$ . Then scan the horizontal projections of their bounding boxes on the y-axis, and determine their “projection overlapping segments” on the y-axis. The bounding boxes that are said to share the same projection overlapping segment must be comprised of bounding boxes that overlap on the y-axis when projected horizontally. For example, if two components  $C_1$  and  $C_2$  overlap with each other, then they

should satisfy the overlapping condition, i.e.  $P_v(B(C_1), B(C_2)) > 0$ .

*Step 3:* For the connected-components sharing the same projection overlapping segments, cluster the connected-components which are aligned with each other into respective groups  $\mathbf{CS}_j$ . Since for character components in a text line, their corresponding bounding boxes of connected-components should be well-aligned with each other, we use the alignment-condition to determine the alignment among the connected-components which share the same projection overlapping segments, and the alignment-condition is defined as,

$$\frac{H(B_s) - H(B_s \cap B_r)}{H(B_s)} < T_a \quad (3.34)$$

where  $B_s$  is the shorter among the two bounding boxes  $B(C_1)$  and  $B(C_2)$ , and  $B_r$  is the taller one;  $T_a$  is a pre-defined threshold with the value 0.33 being used in this study. The left term of this condition can be simplified as,

$$\frac{H(B_s) - H(B_s \cap B_r)}{H(B_s)} = 1 - P_v(C_1, C_2) \quad (3.35)$$

In other words, if the non-overlapping part of the projection (i.e. the part of the projection which does not overlap with the taller bounding box) of the shorter bounding box is less than one-third of its height, then it should be merged into the same group.

*Step 4:* After the above steps are performed, several groups of connected-components,  $\mathbf{CS}_k$ , are obtained; where  $k = 0, 1, 2, \dots, K-1$ , and  $K$  is the number of resultant groups obtained in this recursion of the *X-cut* procedure. If only one resultant group  $\mathbf{CS}_0$  is obtained, then stop the procedure; otherwise, for each group  $\mathbf{CS}_k$ , conduct the vertical clustering procedure *Y-cut*( $\mathbf{CS}_k$ ).

Then the vertical clustering procedure *Y-cut*( $\mathbf{CS}_{in}$ ) is conducted as follows:

*Step 1:* Project all the bounding boxes of the contained connected-components of the input



group  $\mathbf{CS}_{in}$  vertically onto the x-axis.

*Step 2:* Sort all the connected-components  $C_i$  in the  $\mathbf{CS}_{in}$  with respect to their corresponding  $l(C_i)$ . Then, scan the vertical projections of their bounding boxes onto the x-axis and determine the projection overlapping segments. The connected-components whose vertical projections on the x-axis share the same overlapping segment can be detected when the overlapping condition  $P_h(B(C_1), B(C_2)) > 0$  is satisfied.

*Step 3:* For each projection overlapping segment, cluster the connected-components that are covered by the same overlapping segment into the associated group  $\mathbf{CS}_j$ .

*Step 4:* For each determined group  $\mathbf{CS}_j$ , if the following horizontal space condition among adjacent groups is satisfied, then merge them into a single group. The horizontal space condition of two adjacent groups  $\mathbf{CS}_{k1}$  and  $\mathbf{CS}_{k2}$  is defined as:

$$D_h(B(\mathbf{CS}_{k1}), B(\mathbf{CS}_{k2})) < \max\left(\frac{W(B(\mathbf{CS}_{k1}))}{N_{cc}(\mathbf{CS}_{k1})}, \frac{W(B(\mathbf{CS}_{k2}))}{N_{cc}(\mathbf{CS}_{k2})}\right) \quad (3.36)$$

where the term  $\frac{W(B(\mathbf{CS}))}{N_{cc}(\mathbf{CS})}$  reflects the average width of all connected-components that belong to the group  $\mathbf{CS}$ . The horizontal space condition states that if the horizontal space between the adjacent groups is sufficiently small, then the adjacent groups  $\mathbf{CS}_{k1}$  and  $\mathbf{CS}_{k2}$  are joined.

*Step 5:* After the above steps have been performed, several resultant groups  $\mathbf{CS}_l$ , are obtained, where  $l = 0, 1, 2, \dots, L-1$ , and  $L$  is the number of resultant groups obtained in this recursion of *Y-cut* procedure. If only one resultant group  $\mathbf{CS}_0$  is obtained, then stop the procedure; otherwise, for each group  $\mathbf{CS}_l$ , perform the horizontal clustering procedure *X-cut* ( $\mathbf{CS}_l$ ).

Initially, the initial group  $\mathbf{CS}_{in}$  comprises all the connected-components of the current processing binary plane  $\mathcal{BP}_q$ . Accordingly, based on the above-mentioned two procedures, the connected-component clustering process is started by performing the *X-cut* procedure on the initial group  $\mathbf{CS}_{in}$ . If the first recursion of performing the *X-cut* procedure cannot divide the initial group  $\mathbf{CS}_{in}$  into more than one group, then perform the *Y-cut* procedure on  $\mathbf{CS}_{in}$ . The clustering process is performed by recursively applying the *X-cut* and *Y-cut* procedures until the resultant connected-component groups cannot be divided into more sub-groups.

Figure 3.4 depicts the text extraction process. As shown in Figure 3.4(a), for the corresponding connected-components of characters in the binary plane  $\mathcal{BP}_4$  (corresponding to the plane  $\mathcal{P}_4$  in Figure 3.3(g)), there are five resultant  $\mathbf{CS}$ s obtained after the *X-cut* procedure is performed. The *Y-cut* procedure is in turn performed on these five  $\mathbf{CS}$ s. For instance, as shown in Figure 3.4(b), the *Y-cut* procedure is performed on the  $\mathbf{CS}$  at the top of the five  $\mathbf{CS}$ s obtained from the *X-cut* procedure, and then one resultant  $\mathbf{CS}$  is obtained. This is because the connected-components in Figure 3.4(b) are all close to each other, and hence are clustered into a single resultant  $\mathbf{CS}$ . After the *XY-cut* connected-component clustering process on a binary plane is completed, several final  $\mathbf{CS}$ s are obtained, representing candidates of actual text-lines, as shown in Figure 3.4(c). Accordingly, the  $\mathbf{CS}$ s associated with the remaining binary planes are also obtained after the *XY-cut* process is in turn performed on all binary planes.

The text identification process is then conducted to distinguish whether each one of these obtained  $\mathbf{CS}$ s comprises actual text-lines or non-text objects. Before distinguishing and extracting text-lines, we first identify halftone pictorial objects and background regions using the normalized correlation features [88]. For each one of these  $\mathbf{CS}$ s, its associated normalized

correlation features are computed on the bounding box region covered by its contained components. If these normalized correlation features of one **CS** meet the discrimination rules of halftone pictorial objects as suggested in [88], then it is determined to be a pictorial object or a background region.

After pictorial objects and background regions are identified and eliminated, the text identification is then performed on the rest of **CSs**. If a **CS** actually comprises a text-line, it may have the following distinguishing characteristics: 1) its contained connected-components should be respectively aligned, and the number of them should also be in proportion to the width of the **CS**; 2) the contained object pixels in the enclosing region of this **CS** show distinctive spatial variation. For the first characteristic, the identification strategies of the statistical features of connected-components [38] can be applied on each of the **CSs**, as well as the second characteristic can be determined by applying the discrimination rules of transition features on the object pixels contained in each of the **CSs** [89]. Both are independent of font types, lengths and sizes of text strings.

Based on the above-mentioned concepts, the text identification process employs the following heuristic rules  $R_1 - R_5$  determines whether the **CSs** contain text lines or non-text objects. A **CS** is identified as a real text-line if all of the following decision rules are satisfied. First, the ratio of the width  $W$  and the height  $H$  of the enclosing box of the **CS** must satisfy the condition,

$$R_1: W/H \geq 2.0 \quad (3.37)$$

The number of contained connected-components  $N_{cc}$  of the **CS** must satisfy the conditions,

$$R_2: 0.5(W/H) \leq N_{cc} \leq 8.0(W/H), \quad \text{and} \quad N_{cc} \geq 2 \quad (3.38)$$

The ratio of the total area of the bounding boxes of the **CS's** contained connected-components of to the area of its enclosing box must satisfy the condition,

$$R_3: 0.5 \leq \frac{\sum_{C_i \in \text{CS}} A(C_i)}{W \times H} \leq 0.95 \quad (3.39)$$

where  $A(C_i)$  is the area of the bounding box of the  $i$ -th contained connected-component of the **CS**. Then the identification rules based on the statistical features of the contained pixels of the **CS** are introduced as follows. Considering that “0” represents object pixels and “1” background pixels, the number of transition pixels  $T_p$  in the enclosing box of the **CS** is determined by calculating the number of “0” to “1” and “1” to “0” transmissions. Hence the horizontal transition pixel ratio of the **CS** must satisfy the condition,

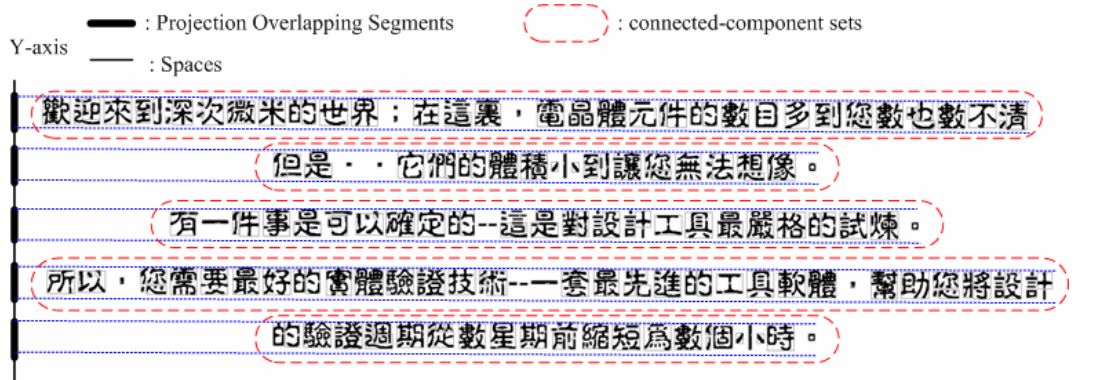
$$R_4: 1.2 \leq T_p / N_{Col} \leq 3.6, \quad (3.40)$$

where  $N_{Col}$  is the number of the column lines in which the object pixels are present. In addition, the density of object pixels in the **CS** must also satisfy the condition,

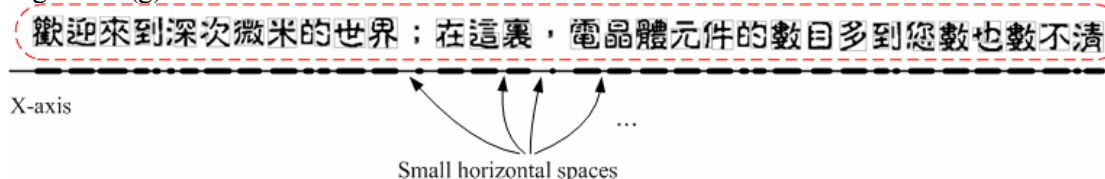
$$R_5: 0.2 \leq \frac{\sum_{C_i \in \text{CS}} O_p(C_i)}{W \times H} \leq 0.8 \quad (3.41)$$

where  $O_p(C_i)$  is the number of object pixels of the  $i$ -th connected-component of the **CS**.

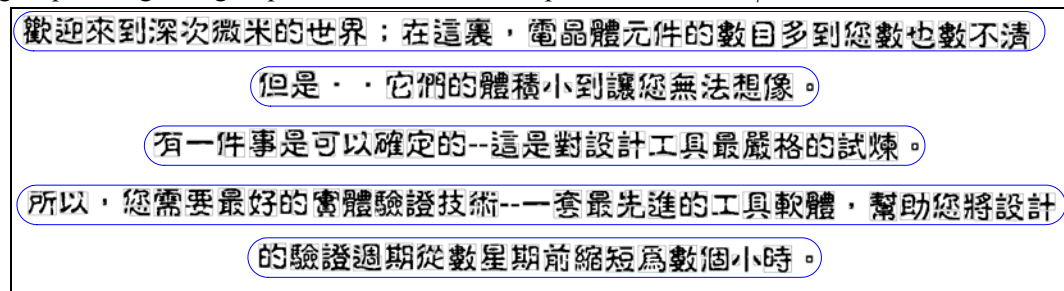
Accordingly, a **CS** is identified as an actual text-line if it satisfies both of the above characteristics. The above-mentioned decision rules are obtained by analyzing many experimental results of processing document images having text strings with various types, lengths and sizes. The constant values utilized under the above decision rules are determined experimentally and yield good performance in most general cases. After the text identification process has been conducted on all object planes, the text-lines extracted from these planes are then collected into a resultant text plane, as shown in Figure 3.4(d).



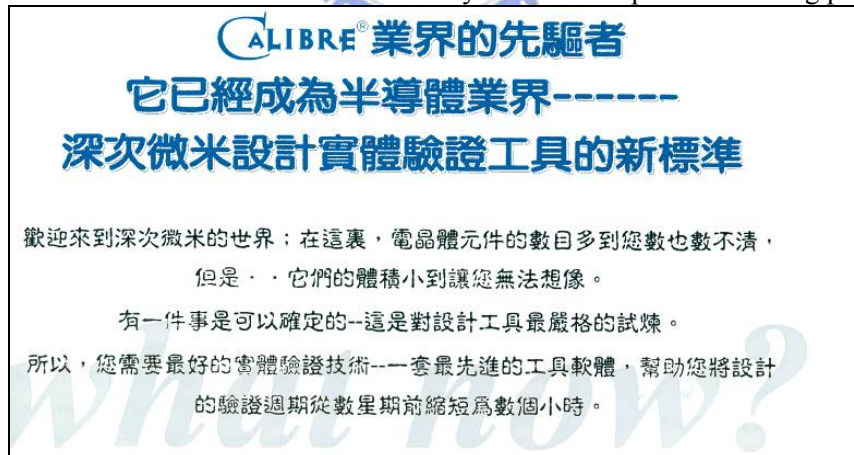
(a). Example of performing *X-cut* on connected-components in the binary plane  $\mathcal{BP}_4$  of Figure 3.3(g).



(b). Example of performing *Y-cut* on the top connected-component group, which is the first group among five groups obtained from *X-cut* procedure on  $\mathcal{BP}_4$ .



(c). The resultant candidate text-lines obtained by the XY-cut spatial clustering process.



(d). The resultant text plane obtained by performing text extraction process on all planes derived from Figure 3.3(a)

Figure 3.4. Examples of the text location and extraction process

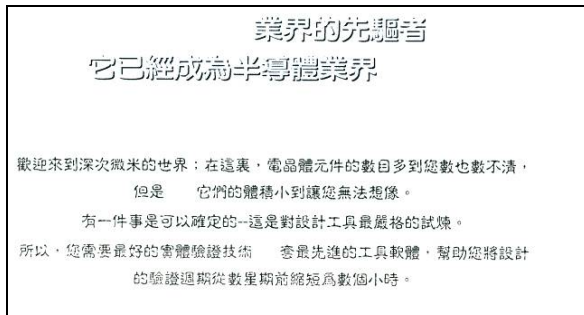
### 3.5 Experimental Results

In this section, the performance of the proposed multi-plane approach is evaluated and compared to several other well-known text extraction techniques, namely Jain and Yu's color-quantization-based method [38], and Pietikainen and Okun's edge-based method [36]. A set of 54 real-life complex document images was employed for experiments on performance evaluation of text extraction. These test images include a variety of book covers, book and magazine pages, advertisements, and other real-life documents at the scanning resolution of 200 dpi to 300 dpi. These images are comprised of textual objects in various colors or illuminations, font styles and sizes, including sparse and dense textual regions, adjoined or overlapped with pictorial, watermarked, textured, shaded, or uneven illuminated objects and background regions.

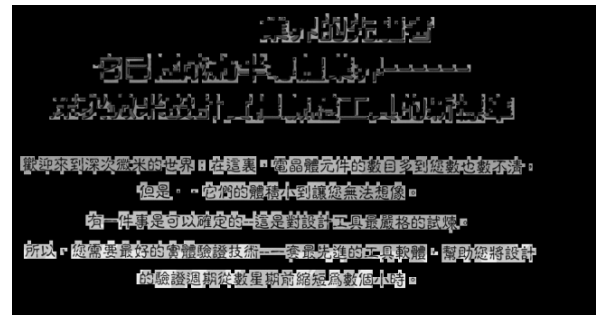


Figure 3.5. Representative color images of Figure 3.3(a) after performing Jain and Yu's method.





(a). Text extraction results by Jain and Yu's method



(b). Text extraction results by Pietikainen and Okun's method

Figure 3.6. Text extraction results of Fig. 3(a) by Jain and Yu's method and Pietikainen and Okun's method.

First, the comparative experiments are conducted on the aforementioned sample image of Figure 3.3(a). Figure 3.5 and Figure 3.6 show the processing results and text extraction results produced by Jain and Yu's color-quantization-based method [38], and Pietikainen and Okun's edge-based method [36]. Here the text extraction results of Pietikainen and Okun's method depicted in Figure 3.6(b) and the later figures are converted into masked images where the black mask was adopted to display the non-text regions. As a comparative experiment of document image decomposition, the decomposition results depicted in Figure 3.5(a)-(d) are four representative color images after performing Jain and Yu's color quantization method [38]. As can be seen from the second representative color image in Figure 3.5(b), the caption characters superimposed on the shaded background are blurred and cannot be appropriately separated. Besides, as shown in Figure 3.5(d), the bottom text-line "what now?" is occluded in the fourth representative color image by reason of the insufficient contrast for color quantization process. As a result, these two textual regions are missed in the resultant text extraction results, as shown in Figure 3.6(a). As seen from Figure 3.6(b), Pietikainen and Okun's method extracts most characters of the body text, but many caption characters are fragmented and characters of the text-line "what now?" are also lost due to the low contrast with the background.



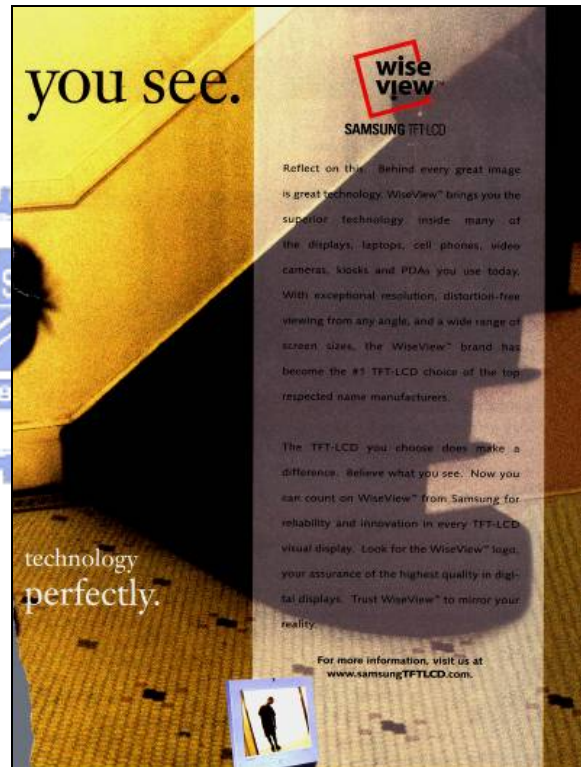
Figure 3.7(a) and Figure 3.7(b) are two typical test images of A4 full-size complex scanning documents. The test image shown in Figure 3.7(a) contains background objects with sharp illumination variations across textual regions, and some of these also possess similar colors and illuminations to those characters touched with them, so that their illuminations are influenced and have gradational variations due to the scanning process; while the test image in Figure 3.7(b) has a large portion of the main body text printed on a large shaded and textured background region, and thus the contrasts between the characters and this textured background region is extremely degraded.

Figure 3.8 and Figure 3.9 depict the decomposition results of Figure 3.7(a) produced by the proposed multi-plane approach and Jain and Yu's color-quantization-based method. As shown in Figure 3.8(a) – (h), the proposed approach clearly segment the homogenous objects into respective object planes. These planes comprise of the textual objects of interest including the large bright characters near the gray boundary blocks in Figure 3.8(b) and (e), the characters "SIEMENS" below the man in black in Figure 3.8(c), the white main body text close to the mobile phone's shell in Figure 3.8(d), and the rest of small characters in Figure 3.8(g) and (h). Comparatively, textual objects of the caption and the main body text in Figure 3.9(b), 9(e) and 9(f) of the representative color images decomposed by Jain and Yu's method are visibly fade or blurred with pictorial objects due to the influence of those background objects during the color quantization process. Figure 3.10(a) – (c) illustrate the text extraction results of Figure 3.7(a) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method. As shown in Figure 3.10(a), the text extraction results by the proposed approach demonstrate that the majority of the textual objects are successfully extracted from the sharply varying backgrounds. By comparison, as shown in Figure 3.10(b), Jain and Yu's method is unsuccessful to extract the large caption characters and many characters in the main body text by reason of the above-mentioned unsatisfactory decomposition results of the

color quantization process. Pietikainen and Okun's method extracts most textual objects except some broken large characters and several missed small characters, as shown in Figure 3.10(c); however, several pictorial objects with sharply varying contours are also identified as textual objects, and thus the characters in extracted textual regions are blurred.

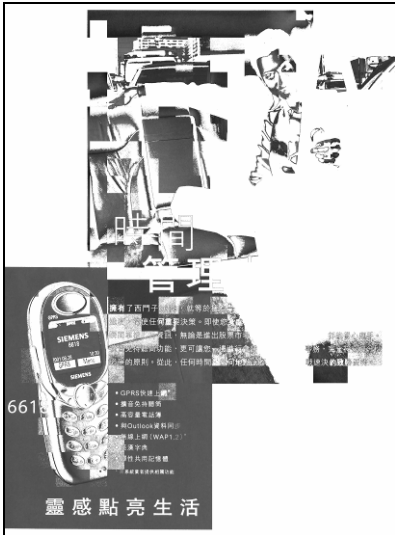


(a). Test image 2 (size: 2333 × 3153)



(b). Test image 3 (size: 2405 × 3207)

Figure 3.7. Original images of the test images 2 and 3



(a). Decomposed object plane 1



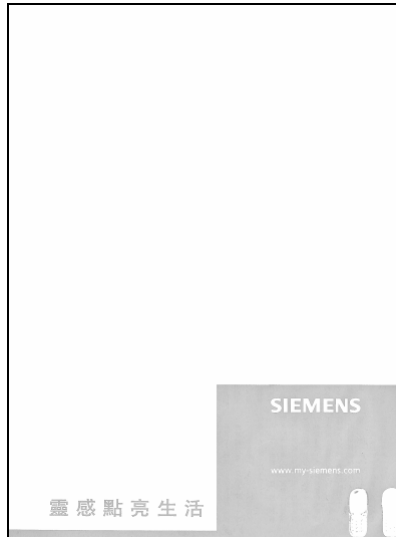
(b). Decomposed object plane 2



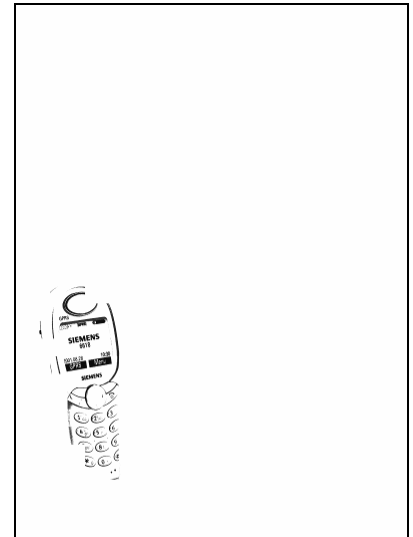
(c). Decomposed object plane 3



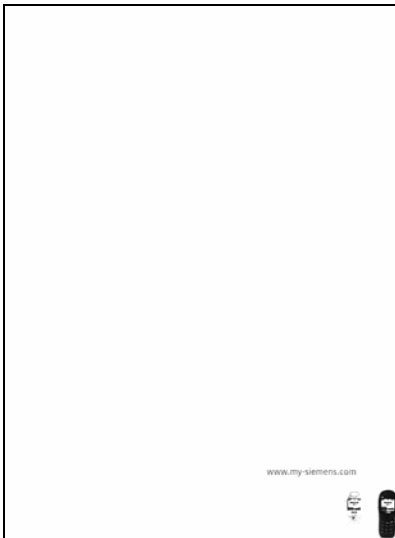
(d). Decomposed object plane 4



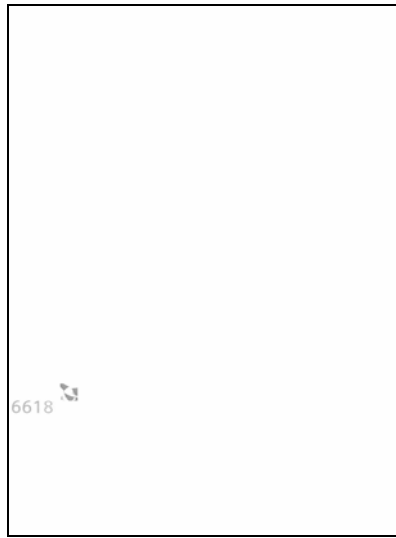
(e). Decomposed object plane 5



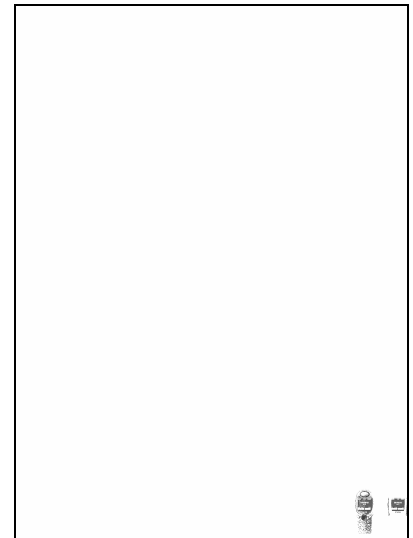
(f). Decomposed object plane 6



(g). Decomposed object plane 7

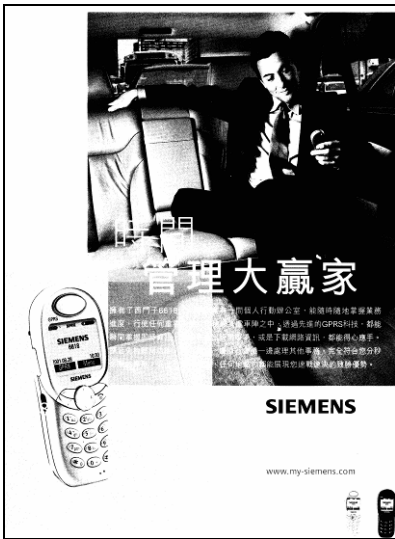


(h). Decomposed object plane 8



(i). Decomposed object plane 9

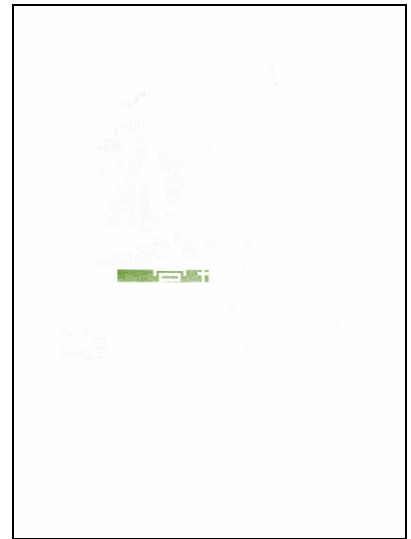
Figure 3.8. Decomposed object planes of Figure 3.7(a) after performing the proposed multi-plane segmentation



(a). Representative color image 1



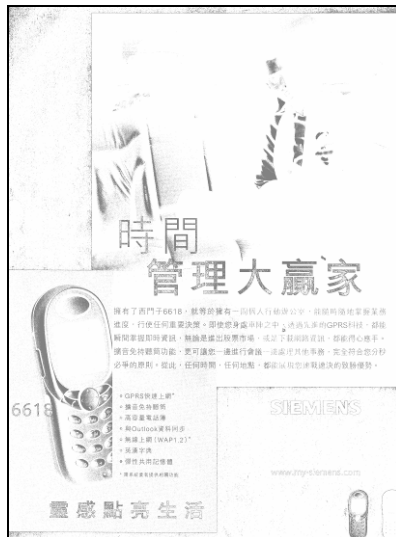
(b). Representative color image 2



(c). Representative color image 3



(d). Representative color image 4

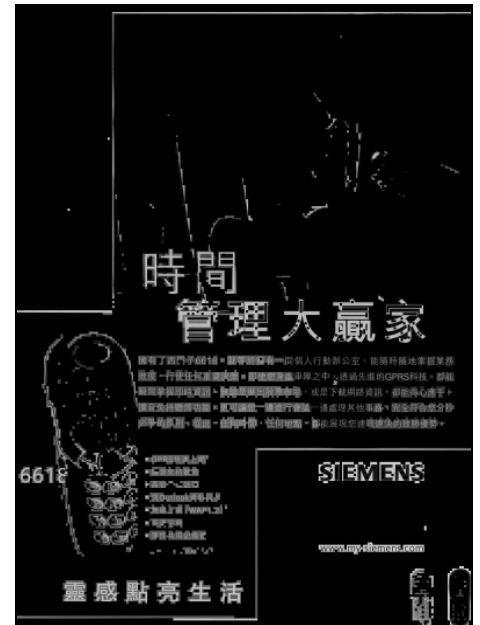
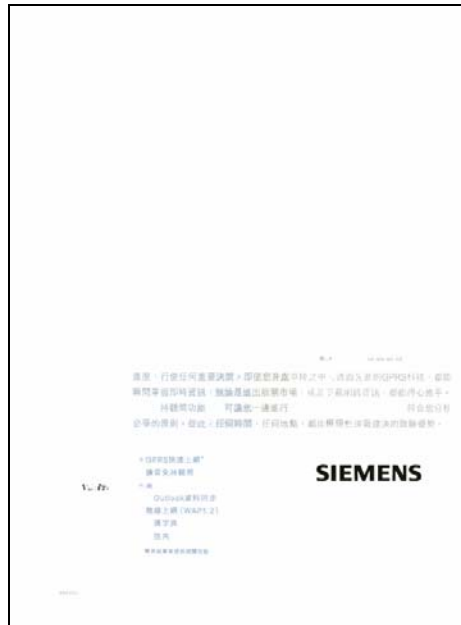
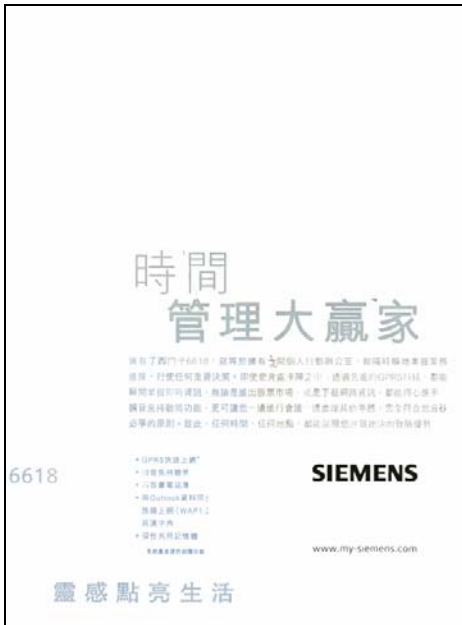


(e). Representative color image 5



(f). Representative color image 6

Figure 3.9. Representative color images of Figure 3.7(a) after performing Jain and Yu's method.



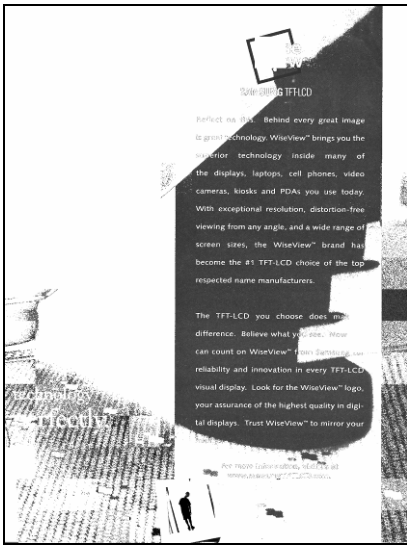
(a). Text extraction results by the proposed approach

(b). Text extraction results by Jain and Yu's method

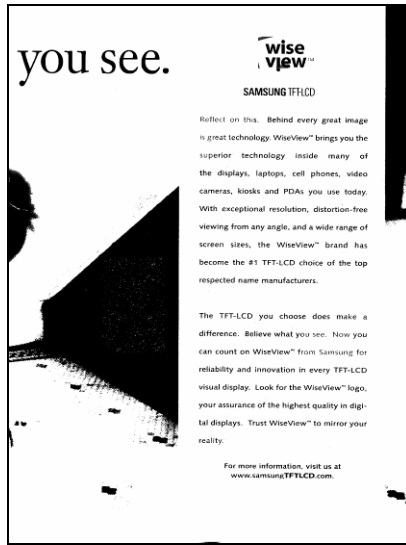
(c). Text extraction results by Pietikainen and Okun's method

Figure 3.10. Text extraction results of Figure 3.7(a) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method.

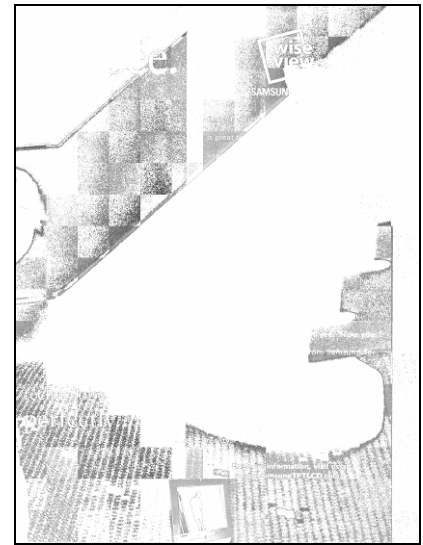
In the test image shown in Figure 3.7(b), the textual region of interest are the caption characters "you see" on the top-left, the main body texts on the right, and the white characters "technology perfectly" on the bottom-left. Figure 3.11 - Figure 3.13 illustrate the decomposition and text extraction results on the test image in Figure 3.7(b) obtained by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method. As shown in Figure 3.11(b), the proposed approach correctly separate the main body texts printed on shaded and textured background regions in highly degraded contrasts. By comparison, textual regions of the main body texts in Figure 3.12(a) of the representative color image obtained by Jain and Yu's method are apparently smeared with the background regions. Accordingly, as can be seen from Figure 3.13(a), the characters in three different textual regions are successfully extracted by the proposed approach; whereas both Jain and Yu's method, and Pietikainen and Okun's method could not perform well on extracting textual objects from the shaded and textured backgrounds in degraded contrasts, as shown in Figure 3.13(b) and Figure 3.13(c), respectively.



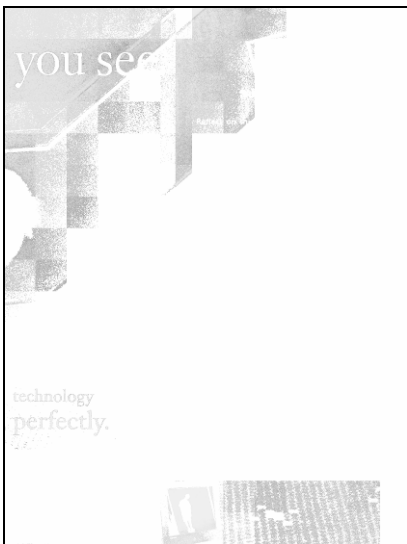
(a). Decomposed object plane 1



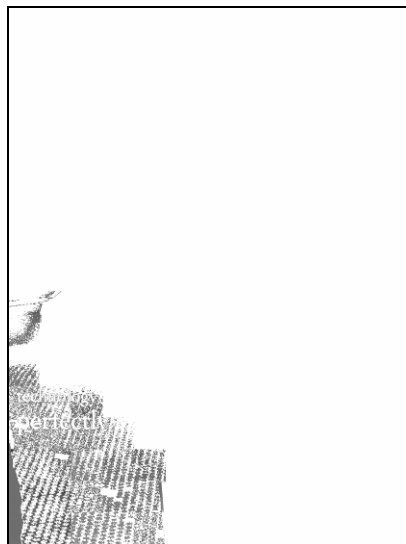
(b). Decomposed object plane 2



(c). Decomposed object plane 3



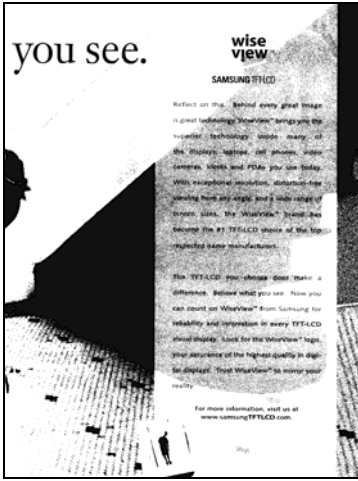
(d). Decomposed object plane 4



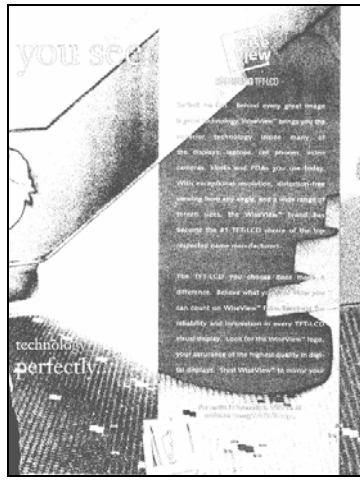
(e). Decomposed object plane 5

Figure 3.11. Decomposed object planes of Figure 3.7(b) after performing the proposed multi-plane segmentation

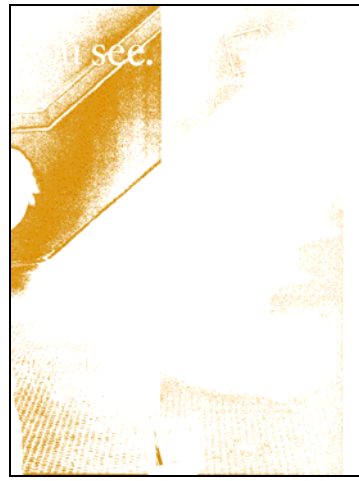




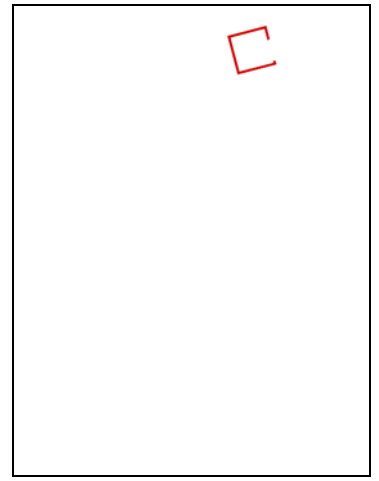
(a). Representative color image 1



(b). Representative color image 2



(c). Representative color image 3



(d). Representative color image 4



(e). Representative color image 5



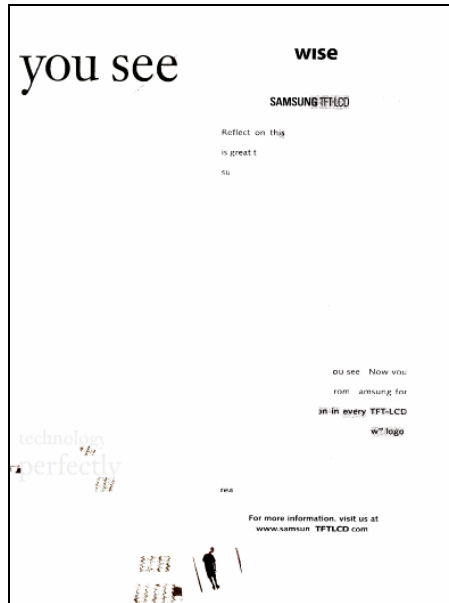
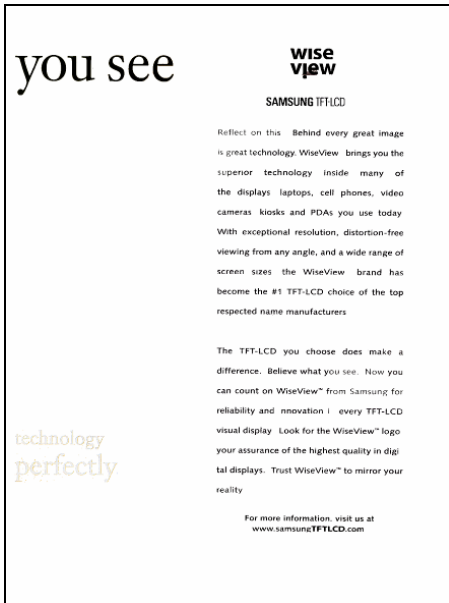
(f). Representative color image 6



(g). Representative color image 7

Figure 3.12. Representative color images of Figure 3.7(b) after performing Jain and Yu's method.



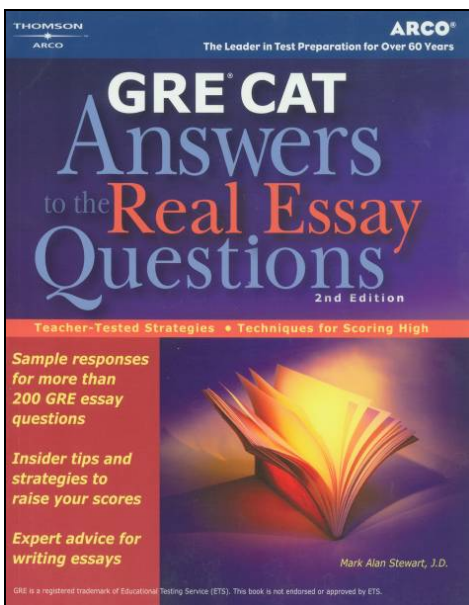


(a). Text extraction results by the proposed approach

(b). Text extraction results by Jain and Yu's method

(c). Text extraction results by Pietikainen and Okun's method

Figure 3.13. Text extraction results of Figure 3.7(b) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method.



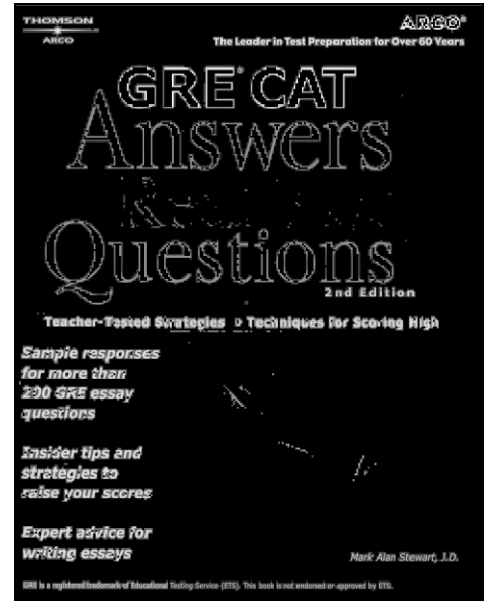
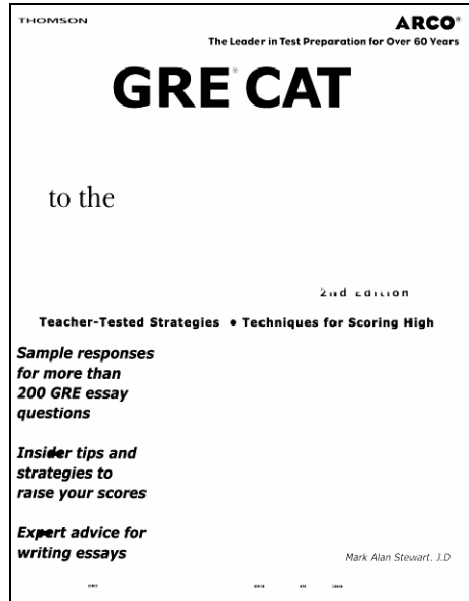
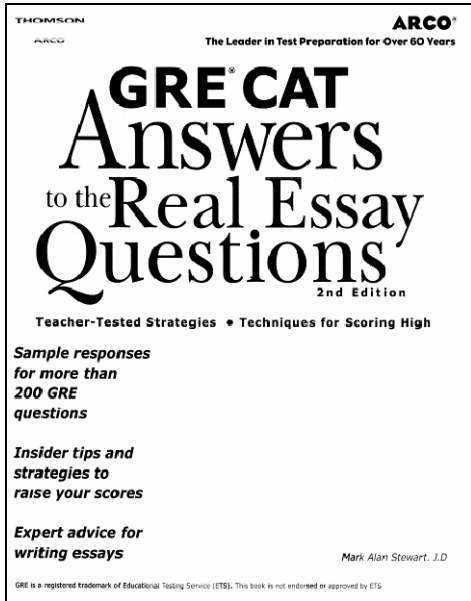
(a). Test image 4 (size: 2864 × 3658)

(b). Test image 5 (size: 2427 × 3166)

(c). Test image 6 (size: 2469 × 3535)

Figure 3.14. Original images of the test images 4 - 6

Figure 3.14(a) – (c) are three test images with several notable characteristics. The test image in Figure 3.14(a) has multiple-colored text-lines printed on several shaded background regions in indistinct contrasts, while the test images in Figure 3.14(b) and (c) comprise textual regions overlapped with numerous character-like objects with similar contrasts and textural features to those of actual textual objects. To facilitate the visual observation of bright characters, the text extraction results of Jain and Yu's method and the proposed approach in Figure 3.15(a)-(b), Figure 3.16(a)-(b), and Figure 3.17(a)-(b) are illustrated in the binarized form. Figure 3.15(a), Figure 3.16(a), and Figure 3.17(a) exhibit that the proposed approach correctly segment and extract the textual objects with different sizes, types, and colors under various difficulties associated with the complexity of background images. As shown in Figure 3.15(b), Figure 3.16(b), and Figure 3.17(b), Jain and Yu's method could not perform well on extracting several text-lines of interest, and some extracted textual regions are also blurred or degraded. As illustrated in Figure 3.15(c), Figure 3.16(c), and Figure 3.17(c), Pietikainen and Okun's method can extract most textual objects, but some shaded textual objects such as the caption characters "to the Real Essay" in Figure 3.15(c) are missed, and many background textures and contoured objects are also identified as textual objects, and so that many extracted textual regions are blotted by these spurious detections.



(a). Binarized text extraction results by the proposed approach

(b). Binarized text extraction results by Jain and Yu's method

(c). Text extraction results by Pietikainen and Okun's method

Figure 3.15. Text extraction results of Figure 3.14(a) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method.

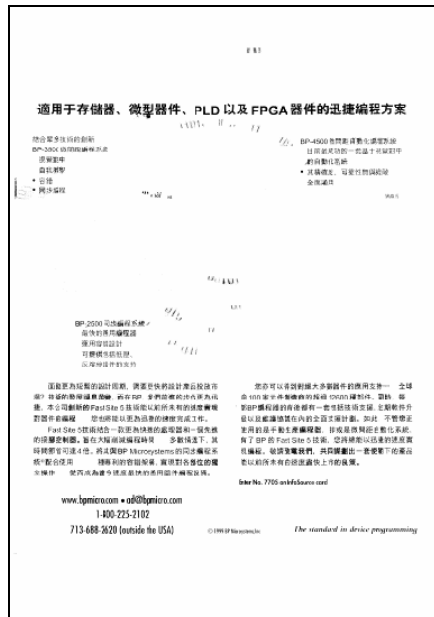
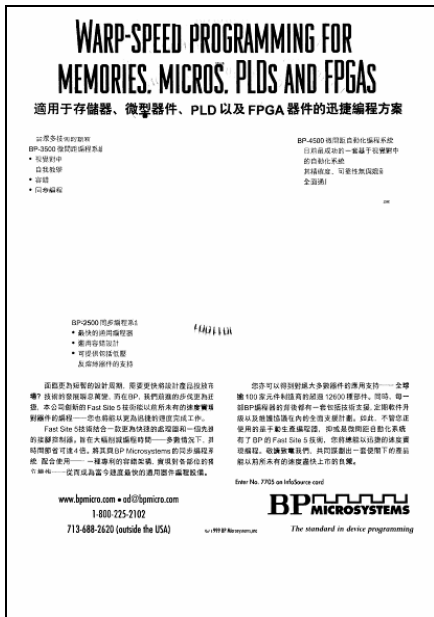


(a). Binarized text extraction results by the proposed approach

(b). Binarized text extraction results by Jain and Yu's method

(c). Text extraction results by Pietikainen and Okun's method

Figure 3.16. Text extraction results of Figure 3.14(b) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method.



(a). Binarized text extraction results by the proposed approach

(b). Binarized text extraction results by Jain and Yu's method

(c). Text extraction results by Pietikainen and Okun's method

Figure 3.17. Text extraction results of Figure 3.14(c) by the proposed approach, Jain and Yu's method, and Pietikainen and Okun's method.



Table 3.1. Experimental data of Jain and Yu's method and our proposed approach

Method	Recall Rate	Precision Rate
Jain and Yu's method	79.8%	95.2%
Our approach	99.1 %	99.3%

For the quantitative evaluation of text extraction performance, two measures, the recall rate and the precision rate, which are commonly used for evaluating performance in information retrieval, are adopted. They are respectively defined as,

$$\text{recall rate} = \frac{\text{No. of correctly extracted characters}}{\text{No. of actual characters}}, \text{ and} \quad (3.42)$$

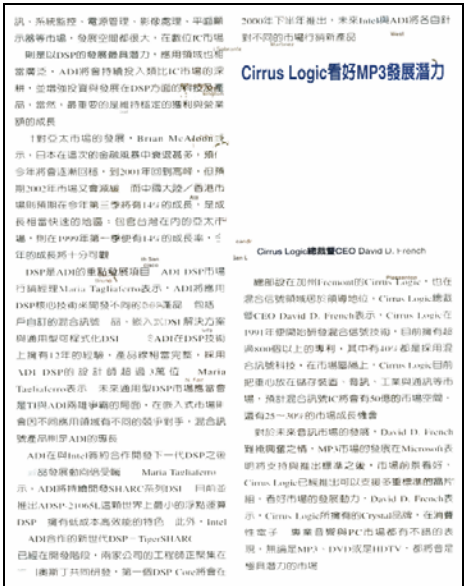
$$\text{precision rate} = \frac{\text{No. of correctly extracted characters}}{\text{No. of extracted character-like components}} \quad (3.43)$$

We compute the recall and precision rates for text extraction results of test images in this study by manually counting the number of actual characters of the document image, total extracted character-like connected-components, and the correctly extracted characters, respectively. The experiments of quantitative evaluation were performed on our test database of 54 complex document images with totaling 22791 visible characters. From the text extraction viewpoint, the recall rate reveals the percentage of correctly extracted characters as opposed to all actual characters within each processed document image, while the precision rate represents the percentage of correctly extracted characters as opposed to all extracted character-like connected-components. Since these quantitative evaluation criteria are performed on the extracted connected-components, the results of Pietikainen and Okun's method is inappropriate for evaluation using these criteria, and were not involved in the quantitative evaluation. Table 1 depicts the results of quantitative evaluation of Jain and Yu's method and the proposed approach. By observing Table 1, we can see that the proposed approach provides better text extraction performance as compared to that of Jain and Yu's method.





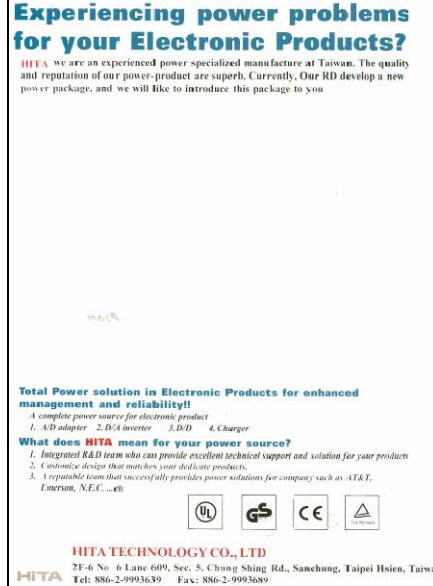
(a). Original Image



(b). Text extraction results by the proposed approach  
Figure 3.18. Results of test image 7 (size: 1829 × 2330)



(a). Original Image



(b). Text extraction results by the proposed approach  
Figure 3.19. Results of test image 8 (size: 3147 × 4536)



(a). Original Image



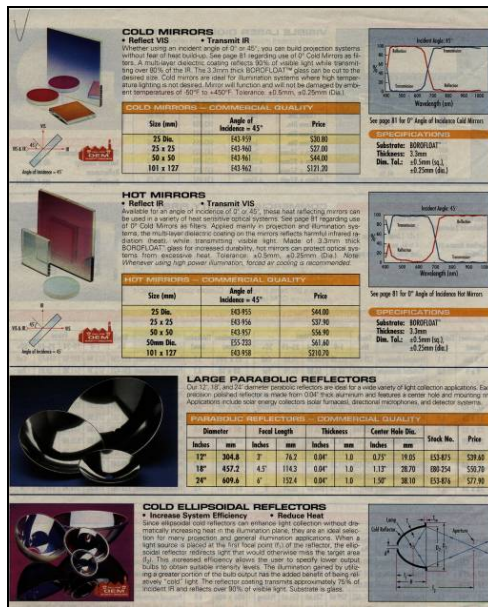
(b). Text extraction results by the proposed approach  
Figure 3.20. Results of test image 9 (size: 1859 × 2437)



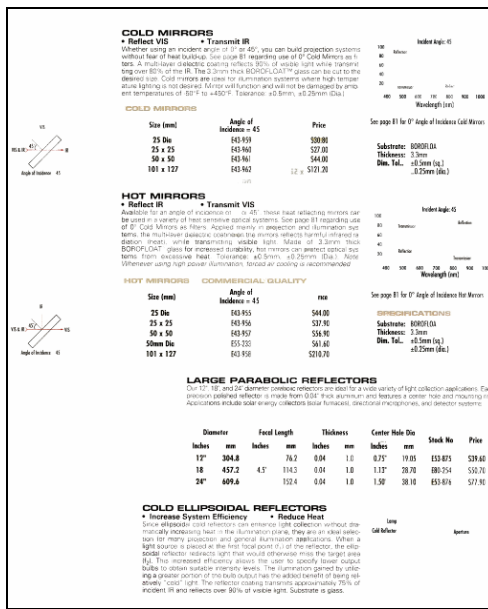
(a). Original Image



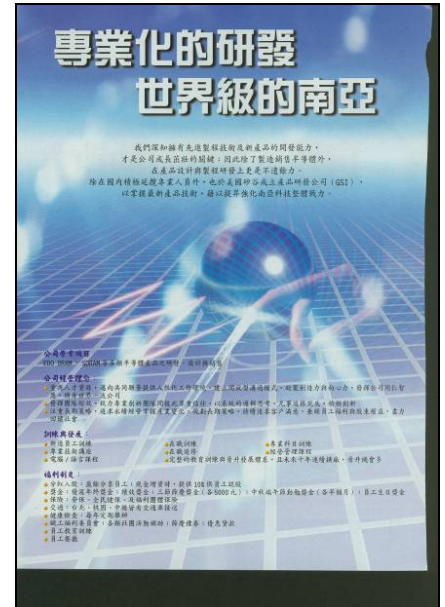
(b). Text extraction results by the proposed approach  
Figure 3.21. Results of test image 10 (size: 1344 × 1792)



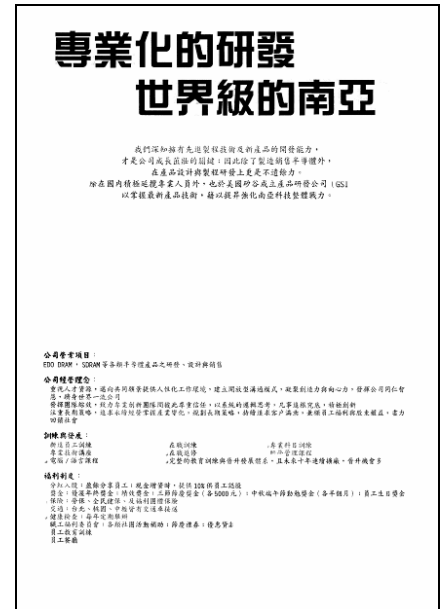
(a). Original Image



(b). Text extraction results by the proposed approach  
Figure 3.22. Results of test image 11 (size: 2309 × 2829)



(a). Original Image



(b). Binarized text extraction results by the proposed method  
Figure 3.23. Results of test image 12 (size: 2469 × 3535)



Figure 3.18 – Figure 3.23 show some further examples of the proposed approach on extracting textual objects from complex document images, and more results of test samples in the experimental set are shown in. Although a few non-text components with character-like characteristics are detected as textual objects, and a few small punctuation marks are missed because of their small sizes and non-alignment with other characters contained in text-lines, the overwhelming majority of the textual objects are correctly obtained. By observing the results obtained, even if textual objects comprised of various illuminations, sizes, and styles, are overlapped with pictorial objects and backgrounds with uneven, gradational, and sharp variations in contrast, illumination, and texture, almost all the textual objects are effectively detected and extracted by the proposed approach.

The proposed approach was implemented on a 2.4 GHz Pentium-IV personal computer using C++ programming language. The computation time spent on processing an input document image depends on the size and complexity of the image. Most of the computation time was spent on the multi-plane region matching and assembling process. For a typical A4-sized document page scanned at 300 dpi resolution, the average image size is 2408 pixels by 3260 pixels, with an average of 1.12 seconds processing time.

## **Chapter 4. NIGHTTIME VEHICLE DETECTION FOR DRIVER ASSISTANCE**

This chapter presents an effective method for detecting vehicles in front of the camera-assisted car during nighttime driving. The proposed method detects vehicles based on detecting and locating vehicle headlights and taillights by using techniques of image segmentation and pattern analysis. First, to effectively extract bright objects of interest, a fast bright object segmentation process based on automatic multilevel histogram thresholding is applied on the grabbed nighttime road-scene images. This automatic multilevel thresholding approach can provide robustness and adaptability for the detection system to be operated well on various illuminated conditions at night. Then the extracted bright objects are processed by a rule-based connected-component analysis procedure, to identify the vehicles by locating and analyzing their vehicle light patterns, and estimating the distances, relative positions, and tracking the relative motion between the detected vehicles and the camera-assisted car.

### **4.1 Introduction**

A vision-based system for detecting the road environment for driver assistance and autonomous vehicle guidance is an emerging research area. Accordingly, many researchers have developed valuable techniques for recognizing interesting vehicles and obstacles from images of road environments outside the car [6]-[9], to facilitate applications on the camera-assisted system that assists drivers in understanding possible hazards on the road, and automatically controlling the apparatus of vehicles, such as headlights, windshield wipers, etc.

A vision-based vehicle and obstacle detection system is aiming at identification of

vehicles, obstacles, traffic signs and other patterns on the road from grabbed image sequences by means of image processing and pattern recognition techniques. Until recently, researchers in this field still open new questions and concepts [42][43]. By adopting different concepts and definitions on interesting objects on the road, different techniques are applied on the grabbed image sequences to detect them as vehicles or obstacles. For locating vehicles in an image sequence, the task can be carried out by searching for specific patterns on the images based on typical features of vehicles, such as shape, symmetrization, or their surrounding bounding boxes [44]-[46]. Until recently, most of these works focused on detecting vehicles under daytime road environments.

However, under bad-illuminated conditions in nighttime road environments, those obvious features of vehicles which are effective for detecting vehicles in daytime become invalid in nighttime road environments. Thus, most of the above-mentioned techniques cannot work well under such nighttime road environments. At night, as well as under dark illuminated condition in general, the only visual features of vehicles are their headlights and taillights. Headlights and taillights are visible if a vehicle lies in the visible range of the CCD camera mounted on a camera-assisted car. However, there are also many other illuminant sources coexisted with the vehicle lights in nighttime road environments, such as street lamps, traffic lights, and road reflector plates on ground. These non-vehicle illuminant sources cause many difficulties for detecting actual vehicles in nighttime road scenes.

Recently, some techniques based on optical sensor are presented [90][91], for detecting the lights of preceding and oncoming vehicles during nighttime driving. For this purpose, an optical sensor array system is set up on the vehicle, then lighting objects appeared in the viewable area ahead of the vehicle are imaged on the lighting sensor array. Then a set of pre-determined thresholds are utilized to retrieve and label pixels of bright spots having gray intensities above the first threshold value, subsequently another threshold value is applied to

determine whether a bright spot is a light of the preceding or oncoming vehicle, shining reflection of a vehicle's lighting, or a lighting source of other circumstances other than that of a vehicle light. However, although such techniques can effectively detect the appearance of incoming vehicles approaching ahead or preceding vehicles driving on the same lane, yet it is still unable to further determine and demarcate their relative positions and the amount of the preceding and incoming vehicles to gain more interesting information of traffic conditions ahead. Furthermore, because these techniques use a set of fixed threshold values configured beforehand, they are unable to adaptively adjust the selection of threshold values aimed at different conditions of nighttime lighting, so the reliability of them on handling the circumstance where road environments have different lighting conditions is limited.

Therefore, an efficient technique for identification of actual vehicle lights to correctly detect and locate moving vehicles in nighttime road-scene image sequences is practically a necessary demand for issues of driver assistance and the development of autonomous camera-assisted vehicles. Besides, this way provides beneficial information for the driver to perceive surrounding traffic conditions outside the vehicle during nighttime driving, and can also be applied to a versatile control scheme for the apparatus of vehicles. For example, the use of high-beam and low-beam states of headlights can be intelligently controlled according to the detection results of presence of oncoming and preceding vehicles, and thus many hazards during nighttime driving, such as headlight dazzler, can be efficiently prevented.

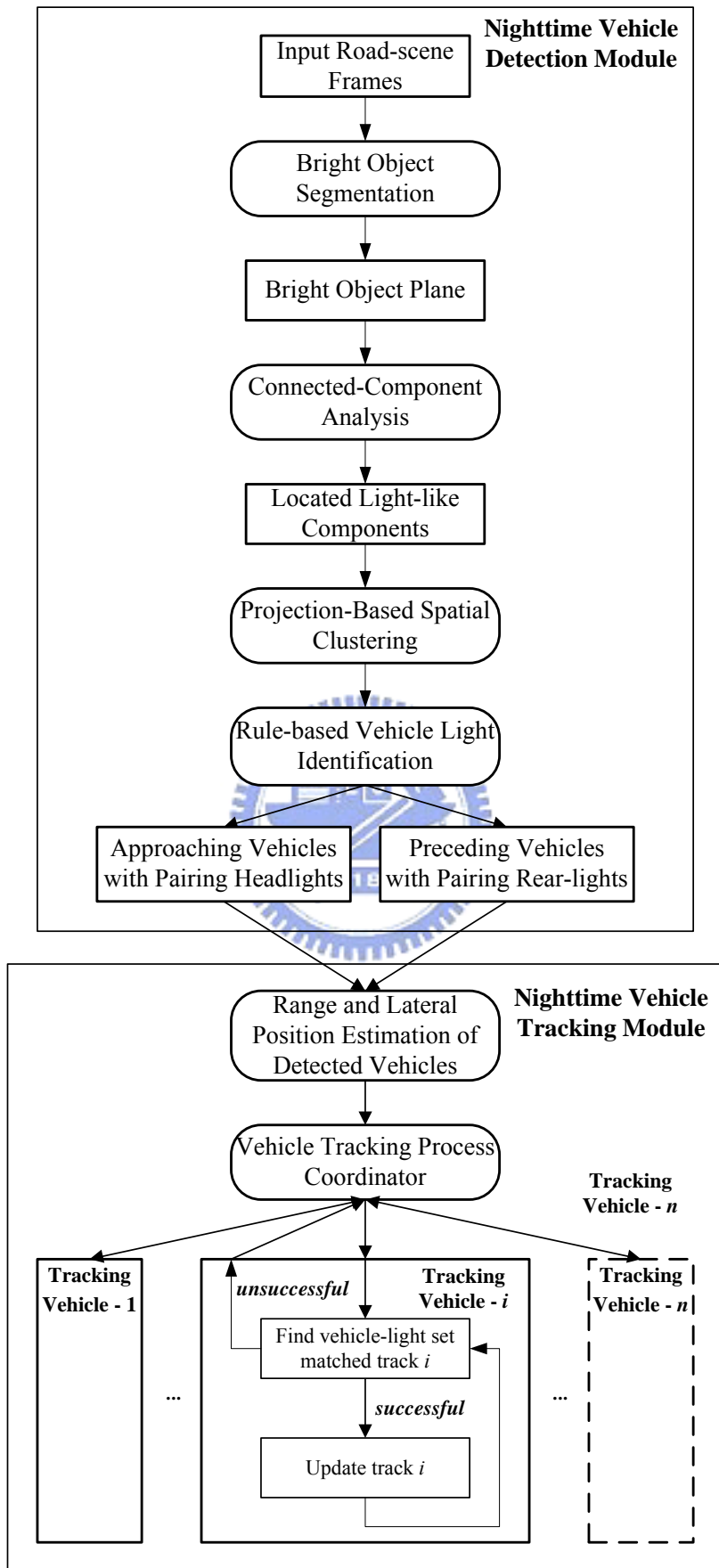


Figure 4.1. Block diagram of the proposed method

In this study, we propose an effective nighttime vehicle detection method for identifying vehicles by locating and analyzing their headlights and taillights. This proposed method comprises of the following processing stages. First, a fast bright object segmentation process based on automatic multilevel histogram thresholding is performed to extract pixels of the bright objects from the grabbed image sequences of nighttime road scenes. The advantage of this automatic multilevel thresholding approach is its robustness and adaptability for dealing with various illuminated conditions at night. Then a connected-component analysis procedure is applied on the bright pixels obtained by the previous bright object segmentation stage, to locate the connected-components of these bright objects. These bright components are then grouped by a projection-based spatial clustering process to obtain potential pairing headlights of oncoming vehicles, and taillights of preceding vehicles. Accordingly, a set of identification rules are applied on each group of bright objects to determine whether it represents an actual vehicle. Next, the distances, relative positions, and motion directions between each of the detected target vehicles appeared ahead and the camera-assisted car are estimated and tracked. Figure 4.1 sketches the flow diagram of the proposed nighttime vehicle detection method. Experimental results demonstrate that the proposed method is feasible and effective on vehicle detection in various nighttime road environments.

## **4.2 Bright Object Extraction**

The input image sequences grabbed from the vision system, which is mounted behind the windshield inside the camera-assisted car. These grabbed frames reflect nighttime road environments appeared in front of the car. The image sequences are grabbed with the 720x480 resolution with 24-bit true colors. Figure 4.2 shows one sample nighttime road

scene taken from the vision system. In this sample scene, there are two vehicles appeared on the road, where the left one is approaching in the opposite direction on the neighboring lane, and the right one is moving in the same direction with the camera-assisted car. The left approaching car shows its bright headlights, while the front moving one shows its smaller and slightly gloomier taillights. In addition to the headlights and taillights of the vehicles, some lamps, traffic lights and signs are also the visible illuminant appeared in the image sequences of the nighttime environment.



Figure 4.2. An example of nighttime road environment



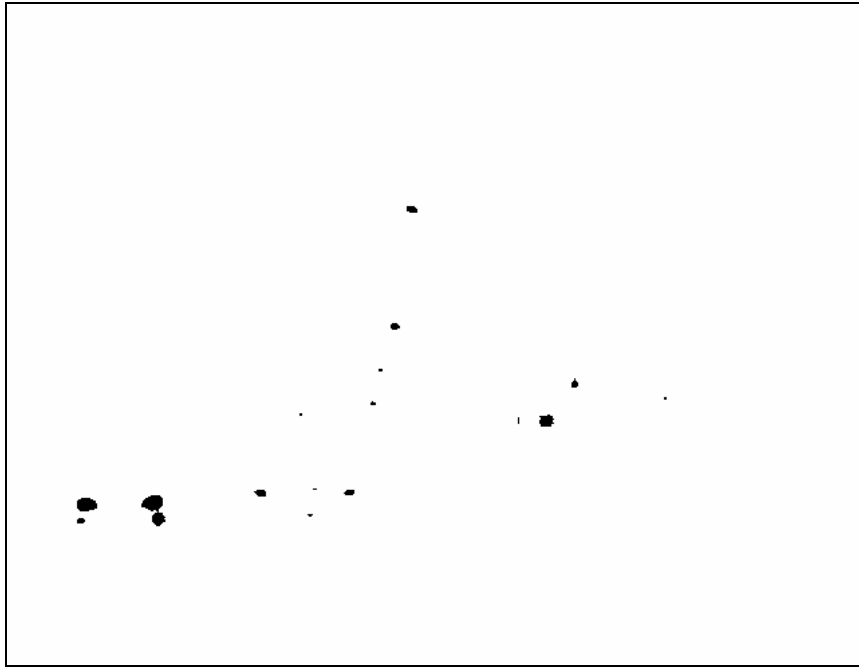


Figure 4.3. Bright object plane extracted from Figure 4.2 after performing the bright object segmentation process



Hence, the first task is to extract these bright objects from the road scene image to facilitate further rule-based analysis. To save the computation cost on extracting bright objects, we firstly extracted the grayscale image, i.e. the Y-channel, of the grabbed image by performing a RGB to Y transformation. For extracting these bright objects from a given transformed gray-intensity image, pixels of bright objects must be separated from other object pixels of different illuminations. Thus, an effective multilevel thresholding technique is needed for automatically determining the appropriate number of thresholds for segmenting bright object regions from the road-scene image. For this purpose, we have already proposed an automatic multilevel thresholding technique for image segmentation in Chapter 2. This technique extends and adopts the properties of discriminant analysis on multilevel thresholding. By evaluating the separability using the discriminant criterion, the number of homogeneous objects of interest, into which a road-scene image should be segmented, can be

automatically determined. As a result, bright objects can be appropriately extracted from other objects contained in the nighttime road-scene. As shown in Figure 4.3, pixels of bright objects in Figure 4.2 are successfully separated into the thresholded object plane after performing the segmentation process.

### 4.3 Spatial Clustering Process for Bright Objects

To obtain potential vehicle-light components from the bright obtained object plane, a connected-component extraction process [85] is then performed on the bright object plane to label and locate the connected-components of the bright objects. By extracting the connected-components, meaningful features of the location, dimension, and pixel distribution associated with each connected-component are also obtained as well. The location and dimension of a connected-component can be represented by the bounding box which encloses it. We are interested in looking for the horizontal-aligned vehicle lights; hence a spatial clustering process is applied on the connected-components to cluster them into several meaningful groups. A resultant group is comprised of a set of connected-components, and it may be consisted of vehicle-lights, traffic lights, road signs, and some other illuminant objects which are commonly appeared in nighttime road environments. These connected-component groups are then processed by the vehicle light identification process to obtain the actual moving vehicles.

First, the definitions used in the projection-based spatial clustering process are described as follows,

- 1).  $C_i$  denotes one certain bright connected-component to be processed.
- 2).  $CG_k$  denotes a group of bright components,  $CG_k = \{C_i, i = 0, 1, \dots, p\}$ , the total

number of connected-components contained in  $CG_k$  is denoted as  $N_{cc}(CG_k)$ .

3). The location of the bounding boxes of a certain component  $C_i$  employed in the spatial clustering process are their top, bottom, left and right coordinates, and they are denoted as  $t(C_i)$ ,  $b(C_i)$ ,  $l(C_i)$ , and  $r(C_i)$ , respectively.

4). The width and height of a bright component  $C_i$  are denoted as  $W(C_i)$  and  $H(C_i)$ , respectively.

5). The horizontal distance  $D_h$  and vertical distance  $D_v$  between two bright components are defined as,

$$D_h(C_i, C_j) = \max[l(C_i), l(C_j)] - \min[r(C_i), r(C_j)] \quad (4.1)$$

$$D_v(C_i, C_j) = \max[t(C_i), t(C_j)] - \min[b(C_i), b(C_j)] \quad (4.2)$$

If the two bright components are overlapping in the horizontal or vertical direction, then the value of the  $D_h(C_i, C_j)$  or  $D_v(C_i, C_j)$  will be a negative value.

6). Hence the measure of overlapping between the vertical projections of the two bright components can be computed as,

$$P_v(C_i, C_j) = \frac{-D_v(C_i, C_j)}{\min[H(C_i), H(C_j)]} \quad (4.3)$$

To preliminarily screen out non-vehicle illuminant objects such as street lamps and traffic lights, we firstly filter out the bright components which are located above the one-third of the vertical  $y$ -axis, i.e. only the bright components located under the constraint line as shown in Figure 4.4 will be taken into account. This is because the vehicles which are located at the distant place on the road become very small light “points”, and will “converge” into a virtual horizon. Hence we utilize the constraint line in Figure 4.4 as this virtual horizon.

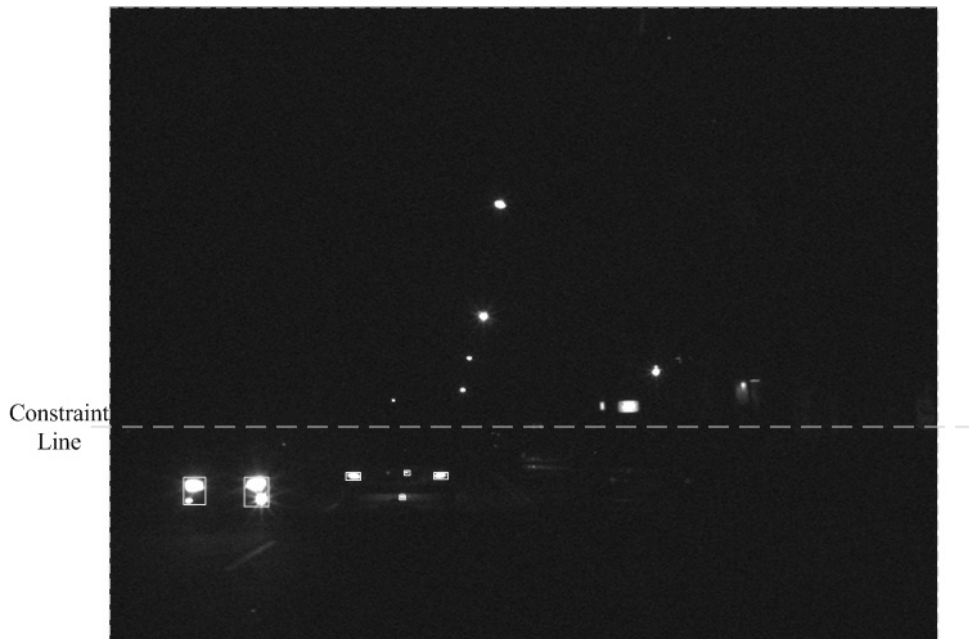


Figure 4.4. The processing area determined by the virtual horizon and the bright components of interest

Next, for the purpose of determining the moving directions of the detected vehicles, it is necessary to distinguish the bright components into potential headlights and taillights for performing respective analysis. Since the headlights have much more variation in colors and sizes than those of the taillights, so that we can utilize some distinguishable characteristics of taillights to distinguish them from potential vehicle lights. The distinguishable characteristic of the taillights is that they mostly are red illuminated lights. However, when the preceding vehicles are near to the camera-assisted car, i.e. within 30 meters, their taillights cause “blooming effects” in CCD cameras, and are usually too bright to appear as white objects in the grabbed images. This is because all  $R$ ,  $G$ ,  $B$  intensities of central pixels in these objects are very close to the full intensities, i.e. 255. As a result, only the pixels that are located around the components of potential taillights have distinguishable red appearance. Hence the following red-light criterion is utilized to check if a bright component contains a potential

taillight object, and this criterion is determined by,

$$R_p(C_i) - T_{red} > \text{both } G_p(C_i) \text{ and } B_p(C_i) \quad (4.4)$$

where  $T_{red}$  is a pre-determined threshold;  $R_p(C_i)$ ,  $G_p(C_i)$ , and  $B_p(C_i)$  respectively represent the average intensities of the  $R$ ,  $G$ , and  $B$  color frames of the pixels which are located at the peripheral of a given bright component  $C_i$ . Here the value of  $T_{red}$  is chosen as 10, to appropriately discriminate the potential taillight components and other bright components. If a bright component  $C_i$  satisfies the red-light criterion, then  $C_i$  is tagged as a red-light component; otherwise, it is tagged as a non-red-light component.

Based on the functions and notations addressed above, the connected-components of bright objects are recursively merged and clustered into bright component groups  $CG_k$  if they have the same light tags, are horizontally close to each other, vertically overlapped, and aligned. In other words, if two neighboring bright components satisfy the following conditions, they are merged with each other and clustered as the same group  $CG$ :

1). They have the same tags, i.e. both of them are red-light components, or both are non-red-light components.

2). They are horizontally close to each other, i.e.:

$$D_h(C_i, C_j) < T_d \times \max(H(C_i), H(C_j)) \quad (4.5)$$

3). They are highly overlapped in vertical projection profiles, i.e.:

$$P_v(C_i, C_j) > T_p \quad (4.6)$$

4). They have similar heights, i.e.:

$$H(C_S)/H(C_L) > T_h \quad (4.7)$$

where  $C_s$  is the one with the smaller height among the two bright components  $C_i$  and  $C_j$ , while  $C_L$  is the larger one.

Here  $T_d$ ,  $T_p$ , and  $T_h$  are the pre-determined thresholds to respect the pairing characteristics of vehicle lights, and the values of them are reasonably chosen as 3.0, 0.8, 0.7, respectively.

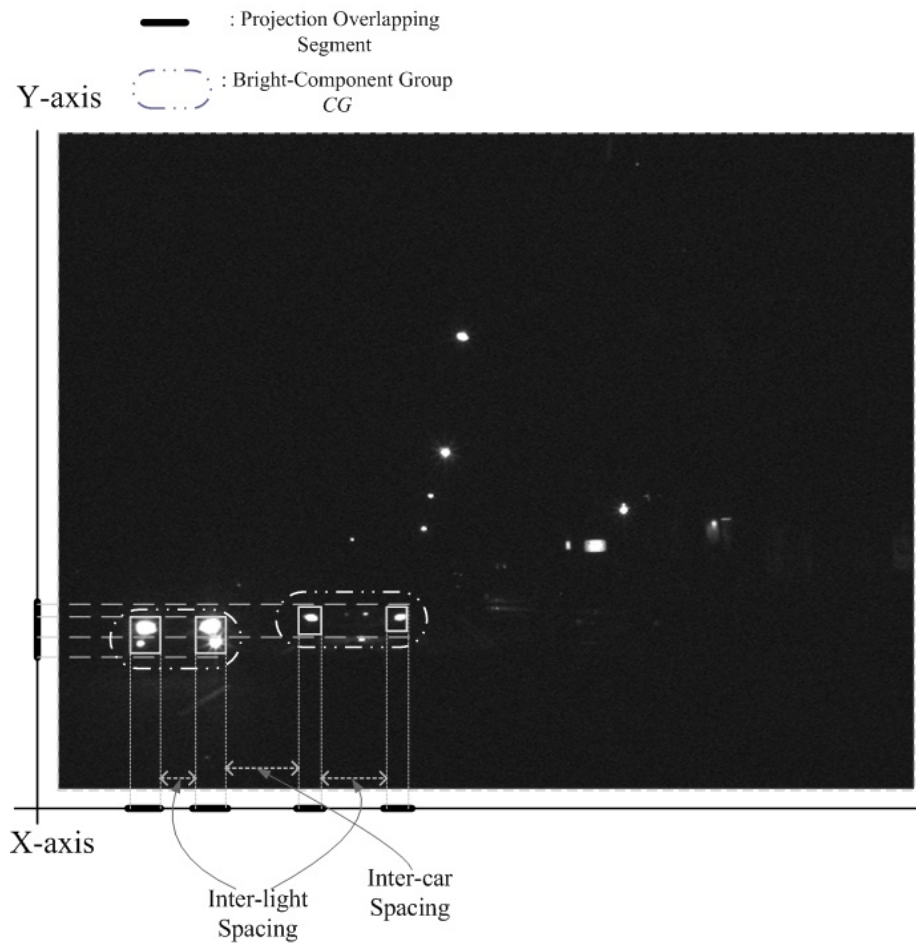


Figure 4.5. The spatial clustering process of bright components

Figure 4.5 illustrates the spatial clustering process of bright connected-components. After performing the spatial clustering process, several groups of bright components are

obtained, and they are called candidate vehicle light groups. As shown in Figure 4.5, the meaningful bright components are grouped into two candidate vehicle light groups, the left one contains the headlights of the oncoming vehicle, and the right one contains the taillights of the preceding vehicle. Then a rule-based vehicle identification process stated in the following section is conducted on these candidate groups to identify and locate actual vehicles appeared in road scene images.

#### 4.4 Rule-Based Vehicle Identification

A rule-based identification process is then applied on each of the candidate vehicle-light groups (i.e. the obtained  $CG_k$ s) to determine whether it comprises of actual vehicle lights or other illuminated objects. This identification process is based on the statistical features of their contained bright components. If a certain candidate group  $CG_k$  contains a set of actual vehicle lights representing a vehicle, then the following heuristic discriminating rules must be satisfied,

1). The enclosing bounding box of the candidate vehicle light group must form a horizontal rectangular shape, i.e. the size-ratio feature of the enclosing bounding box of  $CG_k$  must satisfy the following condition,

$$W(CG_k)/H(CG_k) \geq \tau_r \quad (4.8)$$

where the threshold  $\tau_r$  on the size-ratio condition is selected as 2.0 for suitably reflecting rectangular-shaped appearance of pairing vehicle lights.

2). The number of its contained bright components should also be within a reasonable range, because the vehicle lights are mostly appeared in symmetrical pairs, and some types of compound vehicular light set may comprise of at most four lights, so that the light amount



condition is defined as,

$$\tau_{n1} \leq N_{cc}(CG_k) \leq \tau_{n2} \quad (4.9)$$

where the values of  $\tau_{n1}$  and  $\tau_{n2}$  are chosen as 2 and 4, respectively, to appropriately reflect the pairing characteristic of vehicle lights.

3). Moreover, its contained bright components should be well-aligned, and thus the number of these components should be in reasonable proportion to the size of the size-ratio feature of its enclosing bounding box, thus, the following alignment condition must be satisfied,

$$\tau_{a1} \left( \frac{W(CG_k)}{H(CG_k)} \right) \leq N_{cc}(CG_k) \leq \tau_{a2} \left( \frac{W(CG_k)}{H(CG_k)} \right) \quad (4.10)$$

where the thresholds  $\tau_{a1}$  and  $\tau_{a2}$  are determined as 0.4 and 2.0, respectively, according to our analysis of typical visual characteristics of most vehicles during nighttime driving.


The above-mentioned discriminating rules are obtained by analyzing many experimental results of videos on real nighttime road environments having vehicle lights appeared in different shapes, sizes, directions and distances. The values of thresholds utilized for these discriminating rules are determined to yield good performance in most general cases of nighttime road environments.

## 4.5 Vehicle Distance and Position Estimation

After obtaining each vehicle position represented by vehicle-light group, on the basis of its approximate y coordinate height location of the vehicle body on the image, applying a distance estimation rule with perspective image modeling as base, to carry out vehicle real

space distance and position determining procedure, so as to gain estimation of its corresponding  $Z$ -distance of the coordinate system on imaginary and real world.

To estimate the real-world distance between the camera-assisted car and detected vehicles, we apply the perspective range estimation model of the CCD camera as introduced in [92]. The origin of the virtual vehicle coordinate system is placed at the central point of the camera lens. The  $X$  and  $Y$ -coordinate axes of the virtual vehicle coordinate are parallel to the  $x$  and  $y$ -coordinates of the grabbed images, and the  $Z$ -axis is placed along the optical axis and perpendicular to the plane formed by the  $X$  and  $Y$  axes. A vehicle on the road at a distance  $Z$  in front of the camera-assisted car will project to the image at a vertical coordinate  $y$ . Thus a perspective range estimation model can be utilized for estimating the  $Z$ -distance in meters between the camera-assisted car and one detected vehicle by using the equation,

$$Z = k \cdot \frac{f \cdot H}{y} \quad (4.11)$$


where  $k$  is a given factor for converting from pixels to millimeters for the CCD camera which is mounted on the car at the height  $H$ , and  $f$  is the focal length in meters.

Then the real vehicle body width  $W$  of the detected target vehicle, by the way of the perspective imaging model using the above-mentioned  $Z$ -distance value, can be respectively converted and computed. Let the pixel width that appeared in the image, of the detected target vehicle at time  $t$  be represented by  $w(t)$ , then its corresponding relation using the perspective image model and  $Z$ -distance,  $Z(t)$  at that time can be computed by,

$$\frac{W}{w(t)} = \frac{Z(t)}{k \cdot f}, \quad \text{and} \quad w(t) = k \cdot \frac{f \cdot W}{Z(t)} \quad (4.12)$$

Wherein, the pixel width of the target vehicle  $w(t) = x_r(t) - x_l(t)$ ,  $x_l(t)$  and  $x_r(t)$  are separately the location of the pixel coordinates at time  $t$  of the preceding detected target

vehicle's left-hand boundary (the vehicle light at left hand boundary) and right-hand boundary (the vehicle light at right-hand boundary) in the image. Hence, at a certain time slot  $\Delta t = t_1 - t_0$ , the relative motion velocity  $v$  of the vehicle in concern and a detected target vehicle in front can be gained by way of derivation operation below,

$$\begin{aligned}
 v &= \frac{\Delta Z}{\Delta t} = \frac{Z(t_1) - Z(t_0)}{t_1 - t_0} = \frac{\frac{k \cdot f \cdot W}{w(t_1)} - \frac{k \cdot f \cdot W}{w(t_0)}}{t_1 - t_0} \\
 &= \frac{k \cdot f \cdot W \cdot \frac{w(t_0) - w(t_1)}{w(t_0) \cdot w(t_1)}}{t_1 - t_0} = \frac{Z(t_0) \cdot \frac{w(t_0) - w(t_1)}{w(t_1)}}{\Delta t}
 \end{aligned} \tag{4.13}$$

Hence, if desired to compute the relative velocity  $v$  between the vehicle in concern and the detected target vehicle in front, it can be obtained by way of the product relationship of the Z-distance,  $Z(t_0)$  detected at a certain point of time  $t_0$ , and the changing rate of the pixel width  $w$  of the detected target vehicle in front, i.e.  $(w(t_0) - w(t_1))/w(t_1)$ .

By the way of the perspective model, from the corresponding relationship between the pixel's coordinate location  $x_l(t)$  and  $x_r(t)$  (as shown in Figure 4.6) of the preceding detected target vehicle's left-hand boundary and right-hand boundary in the image and Z-distance  $Z(t)$ , the real relative lateral positions  $X_l(t)$  and  $X_r(t)$  between it and the vehicle in concern on the lane can be respectively derived and computed. Suppose at time  $t$ , a certain position  $X(t)$  is at a distance  $Z(t)$  meters from the vehicle in concern on the lane, its corresponding relative position  $X(t)$  of the pixel coordinates in the image will have the corresponding conversion relationship,

$$\frac{X(t)}{x(t)} = \frac{Z(t)}{k \cdot f}, \text{ and } X(t) = \frac{x(t) \cdot Z(t)}{k \cdot f} \tag{4.14}$$

Based on the above mentioned equations it can be learned that the left-hand boundary,

$X_l(t)$  and right-hand boundary,  $X_r(t)$  of the preceding detected target vehicle can be respectively computed as,

$$X_l(t) = \frac{x_l(t) \cdot Z(t)}{k \cdot f}, \text{ and } X_r(t) = \frac{x_r(t) \cdot Z(t)}{k \cdot f} \quad (4.15)$$

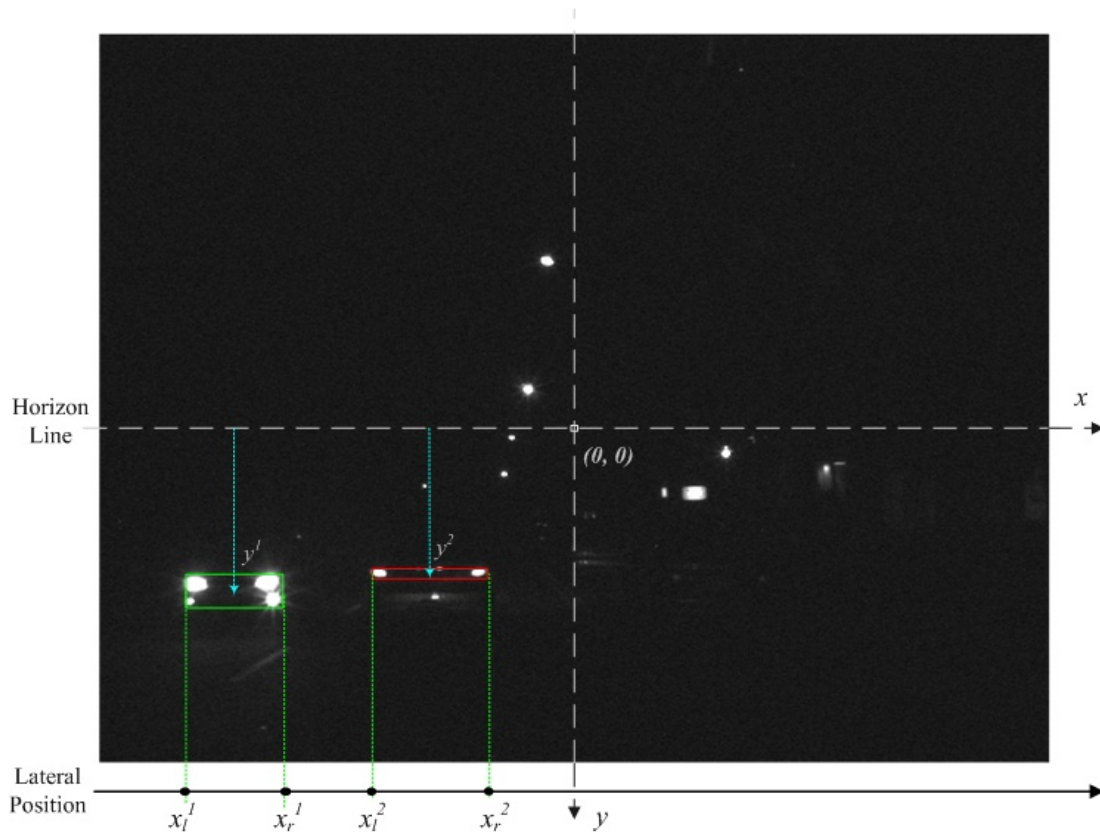


Figure 4.6. The illustration of the lateral positions of the detected vehicle bodies in the image coordinates

By means of the above-mentioned equations, the motion information about distances, relative velocities, and relative lateral positions et cetera between the vehicle in concern and the preceding detected target vehicle on the lane can be obtained. In this way, the driver can be assisted for learning about the information of relative positions and motion between the vehicle in concern and preceding vehicles so as to adopt correct driving actions and prevent nighttime vehicle accidents from happening, more further application of this detection

information is to use it as an automated control mechanism of vehicle cruise velocity and driving route, so as to increase the safety of nighttime driving.

## 4.6 Vehicle Tracking Process

After obtaining the motion information of the nominated vehicle light object groups in each consecutive image frame, a tracking procedure of these vehicle light groups can be applied, aimed at the vehicle light groups discriminated in each continuous image frames, with regard to the direction they are advancing, to track and detect so as to appropriately determine the information of their moving directions and actions, position, and relative velocity of every vehicle entering the area under surveillance, in this way the driver can be more perfectly assisted to determine the traffic conditions ahead of the vehicle.

With regard to the vehicle light object groups appearing at a length, as demarcated in the video, they respectively represent oncoming vehicles on the opposite lane and preceding vehicles on the same lane appearing on the preceding lane of the vehicle in concern, after conducting the above-mentioned procedure of real vehicle distances and positions determination to compute and determine the relative space positions ( $Z$ -distance  $Z(t)$ , positions of left-hand boundary  $X_l(t)$  and positions of right-hand boundary  $X_r(t)$ ) of them on the real lane, the motion trajectories of the detected target vehicles are analyzed and looked for in a series of images, until any of them disappear in the line of sight ahead of the vehicle in concern. When a target vehicle  $i$  at time  $t$  ( $t^{\text{th}}$  frame of the video image) appears at the preceding space position in front of the vehicle is represented by  $P_i^t$ , and is defined as:

$$P_i^t = (X_i(t), Z_i(t)) \quad (4.16)$$

where  $X_i(t)$  represents the horizontal midpoint position of the target vehicle  $i$  at time  $t$

appearing on the lane.  $X_i(t)$  can be obtained through the operation given below,

$$\frac{((X_i(t) + X_r(t)))}{2} \quad (4.17)$$

Accordingly, adopting the smallest path coherence function algorithm to compute and obtain the trajectory of each target vehicle appearing in each image frame, and on this account to compute the information of relative direction of motion, relative position, relative velocity et cetera of each vehicle appearing in front of the vehicle in concern at each point of time.

Firstly, a trajectory vector  $T_i$  is used to represent the tracking trajectory vector of vehicle  $i$ , it represents a sequence of positions of the target vehicle appeared in the visible area in front of the vehicle in concern, the trajectory formed by space position of the whereabouts in order of the  $i^{th}$  target vehicle in a continuous time slot  $0-t$  ( $0^{th}$  to  $t^{th}$  image), and is defined as:

$$T_i = \langle P_i^0, P_i^1, \dots, P_i^t, \dots, P_i^n \rangle \quad (4.18)$$

Thus, let  $d_i^t$  represents the path deviation value of  $i^{th}$  vehicle at  $t^{th}$  image frame in time, which is,


$$d_i^t = \phi(P_i^{t-1}, P_i^t, P_i^{t+1}) = \phi(\overline{P_i^{t-1}P_i^t}, \overline{P_i^tP_i^{t+1}}) \quad (4.19)$$

Wherein the function  $\phi$  is the path coherence function and vector  $\overline{P_i^{t-1}P_i^t}$  represents the position changing vector of vehicle  $i$ 's motion from  $P_i^{t-1}$  to  $P_i^t$ . The path coherence function  $\phi$  can be gained by calculations from the relationship formula between motion vectors  $\overline{P_i^{t-1}P_i^t}$  and  $\overline{P_i^tP_i^{t+1}}$ , the path coherence function  $\phi$  has two main terms, the former term represents the deviation of motion direction formed by  $\overline{P_i^{t-1}P_i^t}$  and  $\overline{P_i^tP_i^{t+1}}$ , the latter term

represents its change in velocity of motion, the concept is based mainly on the preservation of a definite smoothness by the motion trajectory, hence its direction of motion and velocity of motion should react a definite standard of smoothness, for this reason the path coherence function can be derived and computed as,

$$\begin{aligned}
& \phi(P_i^{t-1}, P_i^t, P_i^{t+1}) \\
&= w_1(1 - \cos \theta) + w_2 \left[ 1 - 2 \left( \frac{\sqrt{d(P_i^{t-1}, P_i^t) \cdot d(P_i^t, P_i^{t+1})}}{d(P_i^{t-1}, P_i^t) + d(P_i^t, P_i^{t+1})} \right) \right] \quad (4.20) \\
&= w_1 \left( 1 - \frac{\overline{P_i^{t-1} P_i^t} \cdot \overline{P_i^t P_i^{t+1}}}{\| \overline{P_i^{t-1} P_i^t} \| \cdot \| \overline{P_i^t P_i^{t+1}} \|} \right) + w_2 \left[ 1 - 2 \left( \frac{\sqrt{\| \overline{P_i^{t-1} P_i^t} \| \cdot \| \overline{P_i^t P_i^{t+1}} \|}}{\| \overline{P_i^{t-1} P_i^t} \| + \| \overline{P_i^t P_i^{t+1}} \|} \right) \right]
\end{aligned}$$

Hence, the path deviation of vehicle  $i$  corresponding to its trajectory vector, denoted as  $D_i(T_i)$ , can be computed and obtained by,



$$D_i(T_i) = \sum_{t=2}^{n-1} d_i^t \quad (4.21)$$

Going a step further, when  $m$  number of vehicles appears within the video image in a time slot, the overall trajectory deviation  $\mathbf{D}$  of the motion trajectory vector of these  $m$  number of vehicles can be obtained from the calculations below,

$$\mathbf{D} = \sum_{i=1}^m D_i \quad (4.22)$$

By means of the evaluation process of the overall trajectory deviation  $\mathbf{D}$  defined above, through finding a minimal value of overall trajectory deviation, to obtain the optimal multiple-vehicle tracking trajectory, and then appropriately obtain the information of relative motion directions, relative positions, relative velocities et cetera of the detected target vehicles appearing ahead of the vehicle in concern.



## 4.7 Experimental Results

In this section, we describe the implementation of the proposed method on our experimental camera-assisted car, and conduct various representative real-time experiments to make the performance evaluation of the proposed method.

### 4.7.1 Implementation

The proposed system is implemented on a Pentium-4 2.4 GHz platform which is set up on our experimental camera-assisted car – TAIWAN iTS-1, as shown in Figure 4.7. The vision system for acquiring input image sequences of road environments, as shown in Figure 4.8, is mounted behind the windshield inside the experimental camera-assisted car. The frame rate of the proposed vision system is 30 frames per second and the size of each frame of grabbed image sequences is 720 pixels by 480 pixels per frame.



Figure 4.7. The experimental camera-assisted car – TAIWAN iTS-1



Figure 4.8. The vision system mounted in the experimental car

#### 4.7.2 Performance Evaluation



The proposed system has been tested on several videos of real nighttime road scenes in various conditions. Figure 4.9 – Figure 4.11 exhibit the most representative ones of the experimental samples on performance evaluation. As shown in Figure 4.9, the oncoming vehicle is correctly detected by locating its headlight pair, although some other non-vehicle illuminated objects also coexist with the vehicle in this scene. The distance between this oncoming vehicle and the camera-assisted car is estimated as about 21 meters by the proposed system, which is close to the actual distance obtained by manual measurement.



Figure 4.9. Result of vehicle detection on the nighttime road scene with one oncoming vehicle



Figure 4.10. Result of vehicle detection on the nighttime road scene with both oncoming and preceding vehicles



Figure 4.11. Result of vehicle detection on the nighttime road scene comprised of vehicles and many other non-vehicle lights

Figure 4.10 exhibits a sample of the condition when two vehicles appeared in the scene. Here the headlight set comprised of four headlights at the left is determined as an oncoming vehicle, and its distance to the experimental car is estimated as 9.4 meters. Although some other illuminant objects are coexisted with the taillights of the right vehicle on its body, the taillight pair is still correctly located and identified as a preceding vehicle, and its distance is estimated as 10.7 meters. As shown in Figure 4.11, a more complicated scene is illustrated. The vehicle lights of the two vehicles are very close to each other in this scene. As well as a series of lamps appears above the left oncoming vehicle and many small illuminated light objects occur above and near to the right preceding vehicle. Although interfered by many non-vehicle illuminant objects in this scene, the proposed method still successfully detect these two vehicles by locating their vehicle light pairs. The distances of the left oncoming

vehicle and the right preceding vehicle to the experimental car are estimated as about 23 meters and 10 meters, respectively.

We utilize 4150 real nighttime road-scene images for evaluating the vehicle detection performance of our proposed system. The detection rate in vehicle detection of the proposed system is 93.8% within 40 meters. The computation time spent on processing one input frame depends on the road scene complexity of the frame. Most of the computation time is spent on the connected-component analysis and the projection-based spatial clustering process on bright objects. For an input video sequence with 720x480 pixels per frame, the proposed system takes an average of 24 milliseconds processing time per frame. Thus, this frugal computation cost ensures that the proposed system can effectively satisfy the demand of real-time processing with 30 frames per second. As can be seen from the experimental results of our numerous outdoor tests on many different road environments at night, the proposed system demonstrates that it can provide fast, real-time, and effective nighttime vehicle detection performance.



## **Chapter 5. REAL-TIME WAVELET-BASED VIDEO COMPRESSION APPROACH TO VIDEO SURVEILLANCE SYSTEMS**

In this chapter, we will present a wavelet-based approach to compressing video, with high speed, high image quality and high compression ratio. Using the sequential characteristics of surveillance images, this method applies the low-complexity zero-tree coding, which costs low memory, to develop an algorithm for encoding and decoding video, which greatly improves the speeds of compression and decompression and maintains images of high quality. The method provides good quality and smoothness even under multi-channel surveillance, and so is of great value to companies that develop multi-channel surveillance systems. The ActiveX technique is used to implement the algorithm to take advantage of multimedia, the internet and visual rapid-application-development. The versatile and intelligent surveillance system includes peripheral computer hardware and mobile communication. Incorporating IA, this system is not just a surveillance system but is, rather, an intelligent home manager that can control electronic appliances, video/audio systems and home security in an “e-Home”.

### **5.1 Introduction**

Real-time video compression and transmission techniques have been very popular research topics in recent years. MSN video chat real-time is one of the most interesting applications. Security issues also become more important in modern life, so the surveillance system is an emerging application of video compression and communication. However, the conventional surveillance system is an analog system, which uses many tapes and human effort, to replace the tapes frequently. The recording time and image quality of systems



cannot compete with those of digital surveillance systems. The real-time wavelet-based video compression method and the intelligent multi-channel surveillance system presented in this work fulfill the requirements of real-time multi-channel video compression, while ensuring the high quality of restored images and the efficiency of compression and decompression of the images. The ActiveX technique, from the main stream of software engineering, is used to implement the system. ActiveX supports it's the rapid development of multi-media and internet applications.

The transform coding technique is the most popular method for compressing images. At the beginning of the development of this field, DCT-based (Discrete Cosine Transform) coding was commonly used, and has since become an element of the JPEG image compression standard. Accordingly, its application can be seen in many electronic devices today. Over recent years, researchers have demonstrated that DWT-based (Discrete Wavelet Transform) transform coding ([47]-[50]) outperforms DCT-based methods. Hence, newly emerging image compression methods such as the video compression method standard MPEG-4 [51][52] and the still image compression standard JPEG2000 are using DWT-based methods [53][54].

The lifting scheme [53] almost halves the time taken to perform the calculations required in DWT, and so it has been proposed for incorporation into the DWT-based image compression techniques. The zero-tree coding method ([49][50][93]) is the most efficient and simplest approach to DWT-based transform coding.

In 1993, Shapiro proposed the earliest zero-tree wavelet algorithm [49], which was named the embedded zero-tree wavelet algorithm. This algorithm offers several advantages. For example, it does not require a pre-loaded tanning table, or statistical information about the image, and it can generate fully embedded codes. It can decompress a coded image at various bit-rates according to varying transmission bandwidth from an image compressed at a



single bit-rate. Said and Pearlman's SPIHT method [50], "Set Partitioning In Hierarchical Trees", further improves upon the execution performance of EZW. SPIHT provides high compressive quality and speed, and has been adopted as part of the MPEG-4 standard [52] in visual texture mode.

Multi CCD camera systems are continuously heavily loaded with sequences of images, so the speed of image compression is critical in such systems. Presently, DWT-based compression techniques suffer from high computational complexity, and so cannot support multi-channel video recording with a high frame rate. A new, highly efficient DWT-based algorithm, which yields images of high visual quality, must be developed. Su and Wu [95], as well as Zhao, Chan and Gao [96] developed suitable zero-tree entropy coding methods with low-complexity and low-memory characteristics for compressing images. This work simplifies and improves upon these algorithms, and combines them with the motion picture interpolation technique to meet the aforementioned high performance requirements.

Software component techniques are commonly used in current software engineering. Microsoft developed the COM [97] technique to solve the software development problems associated with version control, cross-platform conflicts, the reusability of cods, and other issues. Microsoft Windows is built on COM components. The COM components that support "Automation" are called ActiveX Controls [98], and can be easily reused by other programmers to develop their own applications. The application developers can directly and seamlessly embed the ActiveX COM video codec component into their developed video-intensive software and web-based application. This video codec component accelerates technical transfer across various applications under development.

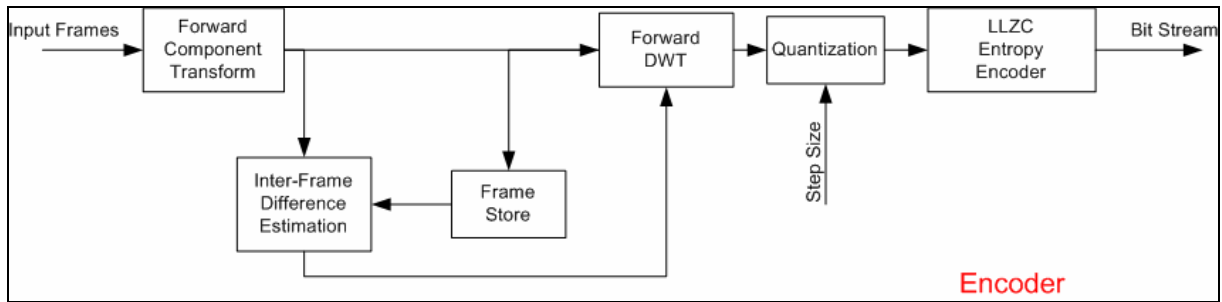
By combining the aforementioned techniques, and with the help of an internet server program, a video capture adapter and CCD cameras, a powerful multi-channel intelligent surveillance system is constructed at low cost and with excellent performance. The rest of

this chapter is organized as follows. In Section 5.2, a high speed wavelet-based video compression method is introduced; experimental results of the proposed method are shown in Section 5.3; the ActiveX video codec component implementation is described in Section 5.4; and the implementation on intelligent surveillance system with mobile carriers is presented in Section 5.5.

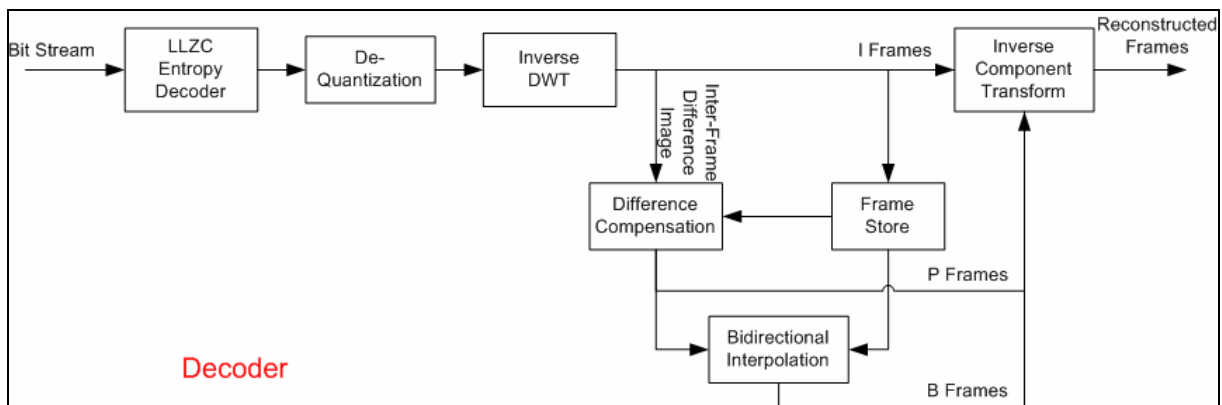
## 5.2 The Proposed Fast Wavelet-Based Video Compression Technique

Most surveillance systems are normally used to survey a designated area for a long continuous time. Using a perfect still image compression method, such as JPEG2000 [53] at all times wastes much storage space. Using standard motion image compression techniques, such as MPEG-4 [51][52], involves huge computational complexity. Neither approach can meet the high frame rate requirement of a multi-channel surveillance system. Hence, this work proposes a simplified video compression method with a high compression ratio and speed.

Figure 5.1 depicts the structure of encoding and decoding flow of the proposed video compression method. Since the decoding flow is the inverse operation of the encoding flow, hence the encoder and decoder can be described together. As shown in Figure 5.1(a) and (b), the function blocks of the encoding flow (Figure 5.1(a)) and the decoding flow (Figure 5.1(b)): forward/inverse component transform, forward/inverse DWT, coefficient quantization and de-quantization, LLZC (low-complexity and low-memory zero-tree entropy coder) encoding and decoding, inter-frame difference extraction, difference compensation and bidirectional frame interpolation. The step size parameter is the size of the step applied by the quantizer. Its value controls the compression ratio of the encoded frame.



(a). Encoding flow



(b). Decoding flow

Figure 5.1. Structure of the coding and decoding flow of the proposed method

The inter-frame correlations in a sequence of surveillance images are crucial to reducing the storage space and computational cost of coding. The residue obtained by taking the difference between two successive frames includes information about the motion of objects. Higher quantity of residue implies the larger image changes, and contains much information. So, more bits are required to encode these changes. The inter-frame difference characteristics are considered to optimize the allocation of bits by the proposed method, and to save transmission bandwidth in channels, without sacrificing the quality of the images.

### 5.2.1 Forward/Inverse Component Transform

The forward component transformation eliminates the correlation between color components, and improves the compression ratio. The compression method involves the reversible color transform, like that of JPEG2000 [53]. This transform is defined as,

$$Y_0 = \left\lfloor \frac{R + 2G + B}{4} \right\rfloor, \quad Y_1 = B - G, \quad \text{and} \quad Y_2 = R - G \quad (5.1)$$

where the symbol  $\lfloor x \rfloor$  represents the nearest integer which is lower than the floating number  $x$ .

The inverse transform of RCT is defined as,

$$G = Y_0 - \left\lfloor \frac{Y_2 + Y_1}{4} \right\rfloor, \quad R = Y_2 + G, \quad \text{and} \quad B = Y_1 + G \quad (5.2)$$

The obtained color components of each of the three color planes are sampled after the RCT transformation is applied to enhance the processing speed of compression. The 4:2:0 sampling method is applied in this study. As depicted in Figure 5.2, the color components of the three color planes are processed by taking the four-point-average in the  $Y_1$  and  $Y_2$  planes and keeping  $Y_0$  unchanged yields a data size ratio,  $Y_0:Y_1:Y_2$ , of 4:1:1.

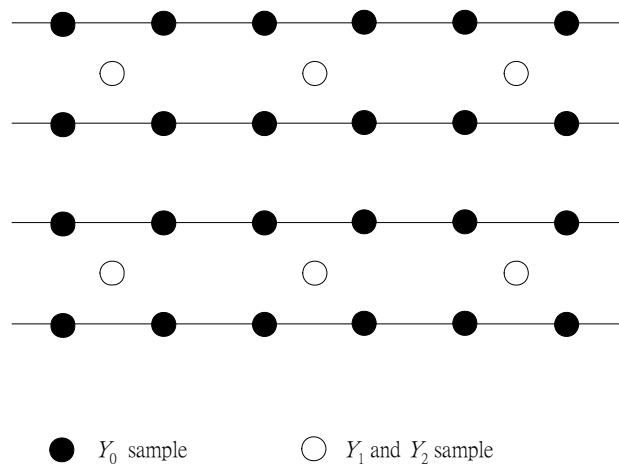


Figure 5.2. Illustration of the 4:2:0 sampling process

### 5.2.2 Forward/Inverse DWT

In image processing, most of the power associated with natural image signals tends to be in the low frequency band. Accordingly, the analysis of the low frequency band must be more extensive than that of the high frequency band. In practical applications, the low frequency band, decomposed from DWT is further analyzed through second level DWT processing to yield more detail of the analysis signal at the lower frequency band. Such analysis is referred to as multi-resolution. The DWT calculation has two parts - analysis and synthesis. The analysis part decomposes the signals into two frequency band using Eqs. (5.3) and (5.4). Figure 5.1(a) depicts the corresponding block diagram. The synthesis part reconstructs these two frequency bands back into the original signal, in a process called inverse DWT (IDWT), using Eq. (5.5). Figure 5.1(b) depicts the corresponding block diagram.

$$a_1[n] = \sum_k h[k] a_0[2n - k]; \quad (5.3)$$

$$d_1[n] = \sum_k g[k] a_0[2n - k]; \quad (5.4)$$

$$\hat{a}_0[n] = \sum_k \tilde{h}[n - 2k] a_1[k] + \sum_k \tilde{g}[n - 2k] d_1[k]. \quad (5.5)$$

where  $a_0[n]$ : the original input signals,  $0 \leq n \leq N$ , and  $N$  is the length of the input signal;  $a_1[n]$  and  $d_1[n]$  denote the low and high frequency bands of  $a_0[n]$  after DWT is performed, respectively, and  $h[n]$  and  $g[n]$  represent the low-pass and high-pass filter of FDWT, respectively. In Eq. (5.5),  $\tilde{h}[n]$  and  $\tilde{g}[n]$  denote the low-pass and high-pass filter of IDWT, respectively, and  $\hat{a}_0[n]$  represents the signal reconstructed from  $a_1[n]$  and  $d_1[n]$ .

Haar's Wavelet Transform is used to increase the speed of execution of the wavelet transform. The two-dimensional DWT is applied as a one-dimensional DWT in the horizontal

direction and then another in the vertical direction. The IDWT is then performed in the reverse order – in the vertical direction followed by the horizontal direction.

The one-dimensional Haar DWT is,

The equation of transformation to decompose the low frequency band is,

$$a_1[n] = \frac{1}{\sqrt{2}} a_0[2n] + \frac{1}{\sqrt{2}} a_0[2n+1]; \quad (5.6)$$

and, to decompose the low frequency band, is,

$$d_1[n] = \frac{1}{\sqrt{2}} a_0[2n] - \frac{1}{\sqrt{2}} a_0[2n+1] \quad (5.7)$$

The inverse transformations are,

$$\hat{a}_0[2n] = \frac{1}{\sqrt{2}} (a_1[n] + d_1[n]), \quad (5.8)$$

$$\text{and } \hat{a}_0[2n+1] = \frac{1}{\sqrt{2}} (a_1[n] - d_1[n]) \quad (5.9)$$

Figure 3(a) plots the corresponding locations of the images of the frequency bands decomposed by 2-D DWT. Figure 3(c) shows the results obtained using the “Lena” image, displayed in Fig. 3(b), after three levels of FDWT processing.

The computational effort of encoding and decoding is an important factor in concerning the compression speed. Hence, the above equations include multiplications by  $1/\sqrt{2}$ , applied after horizontal and vertical DWT, as well as a right-shifting operation instead of real multiplication operations, to reduce the calculation time.

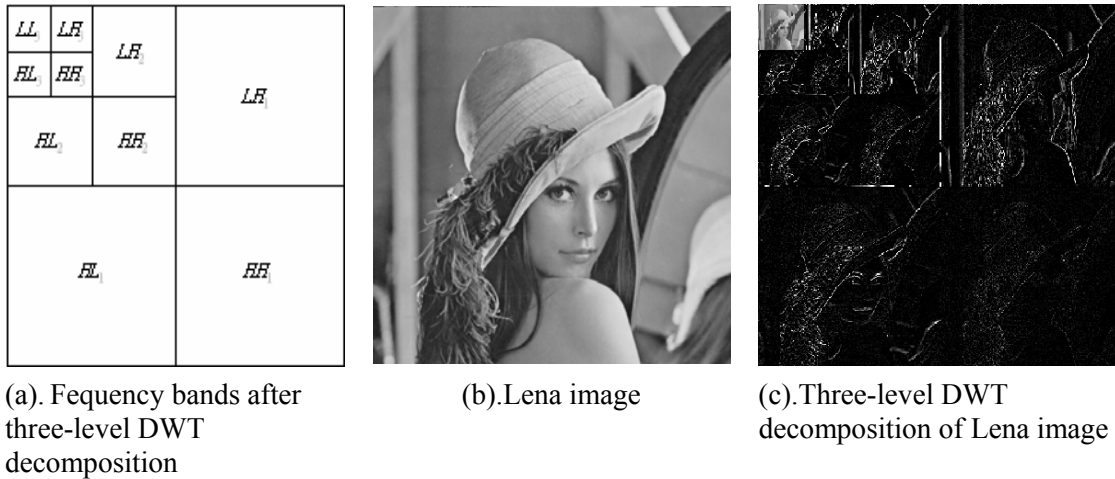


Figure 5.3. Examples of Wavelet Transform

### 5.2.3 Quantization/De-quantization

The proposed quantization is scalar with a dead-zone. The only parameter of the quantizer is the step size. The step size determines the error and the quality of the reconstructed images. A larger step size leads to poorer visual quality of the reconstructed image, but a smaller file size and shorter encoding/decoding period. The coefficients obtained following quantization are encoded using a lossless compression algorithm, the low-complexity and low-memory zero-tree entropy coder, as described in the following subsection. Consequently, the de-quantized values can be obtained directly by multiplying the decoded coefficients by the step size.

### 5.2.4 Low-Complexity and Low-Memory Zero-tree Entropy Coder (LLZC Encoder/Decoder)

Single-pass zero-tree coding and Golomb-Rice code [96] are the two key elements of this encoder. Single-pass zero-tree coding can identify the region in which significant coefficients are distributed; the G-R code encodes these coefficients. G-R code is one type of



variable length code. For applications of DCT-based transform coding, LLZC coding can replace look-up-table Huffman coding procedure in JPEG because the coding involves simpler computations. In DWT-based transform coding, the LLZC coding method is much faster than the SPIHT method.

The tree structure is defined before the coding algorithm is further elucidated. Figure 5.4 plots the three-level, two-dimensional frequency-band coefficient distribution after the wavelet transformation has been applied. Two letters, L and H, and a subscript specify the frequency-band. The first letter represents the horizontal frequency-band, and the second represents the vertical frequency-band. The subscript indicates the  $n$ th level of DWT processing. For example,  $HL_2$  means that the image-band has a high frequency-band in the horizontal direction, a low frequency-band in the vertical direction, and is generated by second-level DWT processing.  $LL_3$  is the lowest frequency-band, thus its coefficients can represent as the dc coefficients of the image after DWT processing, as shown in the top-left shaded area of Figure 5.4. The root of the tree structure is corresponding to the highest level of frequency-bands generated by DWT processing, ex.  $HL_3$ ,  $LH_3$ ,  $HH_3$  are the roots of three-level DWT transform as shown in Figure 5.4, and the frequency-bands with same letters and located in the same direction, ex.  $HL_3$ ,  $HL_2$  and  $HL_1$ , forms a branch of the tree structure. The directions of the arrows in the figure indicate the directions of the descendants. Each node has four direct descendants except those located in the level one, as depicted on the grids shown in  $HH_3$  and  $HH_2$  in Figure 5.4; and then Figure 5.5 presents the pseudo codes of the coding procedure.

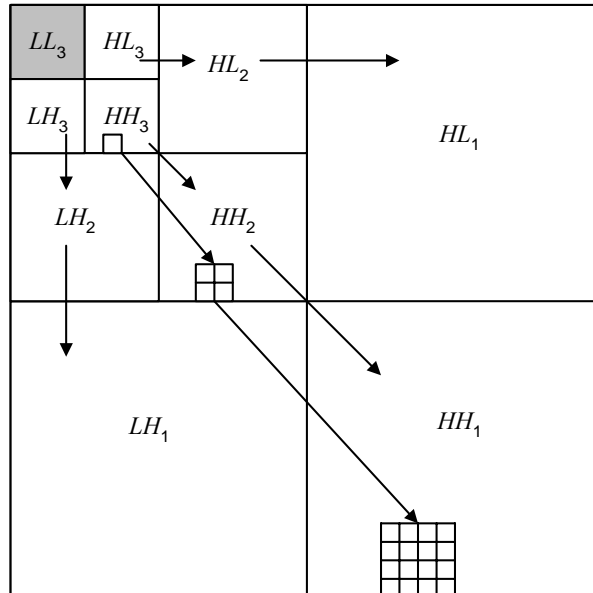
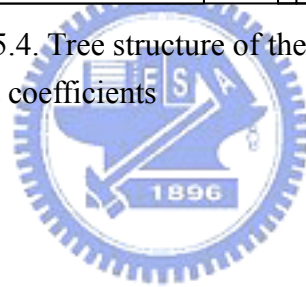


Figure 5.4. Tree structure of the distribution of wavelet coefficients



```

(1) Use DPCM scheme to code the DC coefficients.
(2) For each tree  $k$ , EncodeTree( $k$ ) {
    G-R_FS( $k$ );
    If at least one descendant of  $k$  is not zero {
        Output 1 in 1 bit;
        For each sub-tree of  $k$ :  $s$ , EncodeTree( $s$ );
    }
    else
        Output 0 in 1 bit;
}

```

Figure 5.5. Coding procedure of LLZC

The DPCM scheme [96] encodes the first dc value using G-R\_FS coding, and then encodes the dc value and current dc value minus previous dc value, for the subsequent coefficients. The magnitudes of these coefficients are categorized into several specified classes, as listed in Table 5.1. Then the G-R\_FS coding method is as follows.

- (1) Step 1: Determine the class of coefficient,  $x$ , with reference to Table 5.1. If  $x$  is in class  $n$ , then output will be  $(n+1)$  bits, where the first  $n$  bits are 0 and the last bit is 1.
- (2) Step 2: If  $n > 0$ , the output code is the last  $(n-1)$  bits of the binary code from the absolute value of  $x$ . Finally, the sign bit of  $x$  is the output - 0 for positive and 1 for negative.

For example, the value of  $x$  is -12. In step 1, the output is 00001, because -12 is in class 4. In step 2, first, output the coded bits 100, which is the last 3 bits of the binary code for 12 (1100), the absolute value of  $x$ , and then append the sign bit, 1 for negative value. Combining the aforementioned two steps, the output code for the coefficient value -12 is 000011001.

Decoding is the inverse of encoding. First, the class of  $x$  is identified by the G-R\_FS inverse operation. The class of  $x$  can be obtained by counting 0s until 1 is encountered using the bit-by-bit reading from the bit stream. Then, the absolute value of  $x$  is the value of the next  $(n-1)$  bits plus  $2^{n-1}$ . Finally, the sign bit is read, and the coefficient  $x$  is recovered. For example, when the aforementioned code for -12 is decoded, four zeros are read before the 1 is encountered. Accordingly, the class of the code is 4. Reading 3 more bits reveals the code 100, implying the decimal value 4. Adding  $2^3$  to 4 yields 12. Finally, the sign bit, 1, is read so the final value of the coefficient is determined to be -12.

Table 5.1. Classes of coefficients

class	Coefficients
0	0
1	-1,1
2	-3~-2, 2~3
3	-7~-4, 4~7
4	-15~-8, 8~15
5	-31~-16, 16~31
6	-63~-32, 32~63
7	-127~-64, 64~127
8	-255~-128, 128~255
9	-511~-256, 256~511
10	-1023~-512, 512~1023
11	-2047~-1024, 1024~2047
12	-4095~-2048, 2048~4095
13	-8191~-4096, 4096~8191

### 5.2.5 Inter-Frame Difference Extraction / Difference Compensation and Bidirectional Frame Interpolation

The inter-frame correlations of a series of continuous surveillance frames are essential to reducing the computational complexity of coding and the storage space. The residue is the difference of the power associated with two consecutive frames. These residues contain information about the motion of objects. A higher power associated with the residue frame implies greater changes among frames, and therefore more informational content. Hence, more bits must be allocated to encode the contents. This method utilizes inter-frame differences. A series of frames are divided into a series of frame groups, each of which

contains ten consecutive frames. The ten consecutive frames are further converted into I, an initial frame, P, residual frames, and B, bidirectional interpolation frames. One of two modes of the frame group encoding procedure can be applied – fast mode and turbo mode – as determined by the required rate of compression. The two modes of the encoding procedure are applied to process the frames in each frame group as depicted in the following diagrams, Figure 5.6 and Figure 5.7.

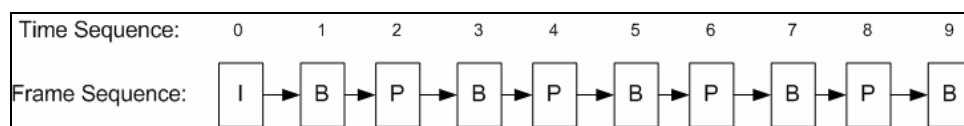


Figure 5.6. Frame sequence encoding in Fast Mode

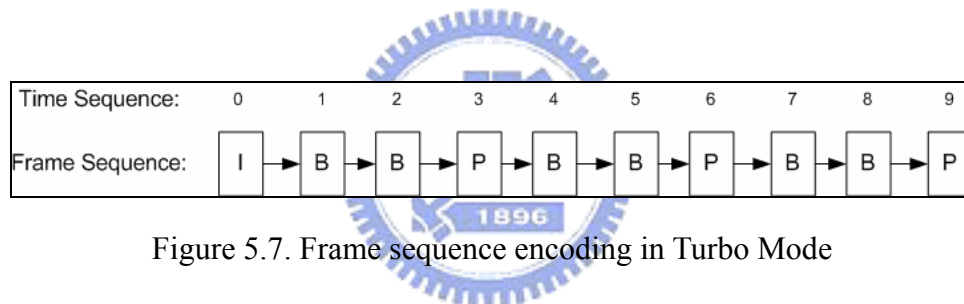


Figure 5.7. Frame sequence encoding in Turbo Mode

I-frame is a reference frame and so is directly encoded referred to the original input frame, and thus more bits are allocated for the I-frame to preserve its information. The P-frame is generated by calculating the inter-frame difference using the frame store buffer. These residue frames are then encoded and sent to receivers consecutively. In the decoding process, the I-frame is directly decoded from the first frame received. As shown in corresponding flow in Figure 5.1(b), the P-frame is then decoded by performing the inter-frame difference compensation with the current inter-frame difference image and the previous I-frame buffered in the frame store. Then, inter-frame difference compensation and bidirectional interpolation are applied to reconstruct the B-frame. The transmission band width can be reduced using this decoding sequence, while maintaining the good quality of the image frames.

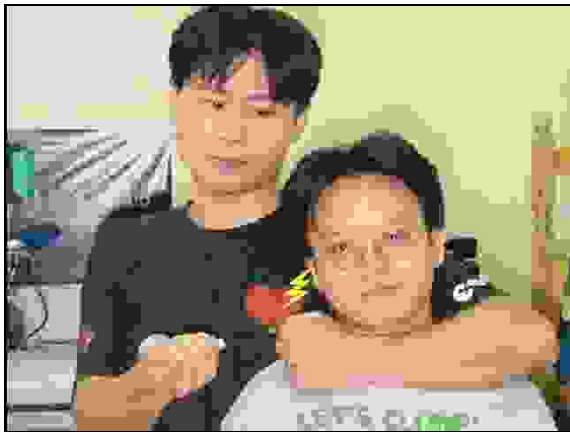
### 5.3 Experimental Results

This section uses a set of real surveillance-recorded images and standard testing images of different sizes to evaluate the performance of the proposed system. Two well-known commercial video compression methods and the proposed method are tested on a PC-based system with a 1.5 GHz AMD Athlon CPU and 512 MB memory. Image quality, compression ratio and compression speed are emphasized in evaluating the performance of all tested compression methods.

First, the quality of an image compressed using the commercially available JPEG-based surveillance video recording system was compared to that of one compressed using the proposed method. Figure 5.8(a) displays a representative surveillance image with 24-bit true color, which is of a crime evident in a house. The only evidence of the crime is the recorded surveillance image. The characteristics of the thief's face must be recognized to enable him to be arrested. Figure 5.8(b) to (e) show the results of compressing Figure 5.8(a) by the JPEG-based system and the proposed method with different compression ratios. However, in Figure 5.8(b), the facial characteristics of the thief are severely blurred, so the surveillance image cannot serve as good evidence in solving the crime. In contrast, even though the image was compressed at a high compression ratio by the proposed method, Figure 5.8(c) preserves distinct and recognizable facial characteristics. These experimental results indicate that for a given same compression ratio, the proposed method has a higher PSNR and yields images of much better visual quality.



(a). Original image (size: 640 x 480)



(b). Compression result by JPEG  
(compression ratio: 100 and PSNR: 27.25 dB)



(c). Compression result by the proposed method  
(compression ratio: 100 and PSNR: 29.09 dB)



(d). Compression result by JPEG (compression  
ratio: 50, and PSNR: 31.12 dB)



(e). compression result by the proposed method  
(compression ratio: 50, and PSNR: 34.78 dB)

Figure 5.8. Comparative results of JPEG and the proposed method



The results shown in Figure 5.8 also reveal that although the image is compressed by the proposed method at more than double of the compression ratio of that by the JPEG-based system, the visual quality of the compressed image by the proposed method (Figure 5.8(c)) can still retain similar visual quality with that by the JPEG-based system (Figure 5.8(d)). Hence we can find that under the requirement of similar visual quality of the compressed surveillance video sequence, the proposed method only needs less than a half of the needed storage capacity for storing similar time period of the surveillance video sequence using the JPEG-based system. For instance, we utilize one hard disk with 120GB capacity to compare the usage of storage space between the JPEG-based system and the proposed method. For the uncompressed QVGA format ( $320 \times 240$  pixels) true color (three bytes per pixel), real-time (30 frames per second) video sequence, the storage space needed per second is:  $320 \times 240 \times 3 \times 30 = 6912000$  bytes, and the storage period needed per day in seconds is:  $60 \times 60 \times 24 = 86400$  seconds, and hence total storage space needed per day for uncompressed QVGA, true color, real-time video sequence is 597 giga-bytes. Hence, as shown in Table 5.2, the proposed method can store double time period of surveillance video sequence compared with the JPEG-based system under similar visual quality.

Table 5.2. Comparisons of the need of storage space between the JPEG-Based system and the proposed method under similar visual quality (QVGA format,  $320 \times 240$ )

Method	Compression Ratio	120GB storage space can store
JPEG	100	20 days
Proposed method	200	40 days



(a). Original image “Akiyo”



(b). Compressed image by MPEG-4 with compression ratio: 184



(c). Compressed image by proposed method of Fast mode with compression ratio: 176



(d). Compressed image by proposed method of Turbo mode with compression ratio: 257

Figure 5.9. Results of compressing the standard testing clip – “Akiyo” (CIF format, 352\*288)

MPEG-4 [51][52] was also compared with the proposed system by application to two standard testing video clips, “Akiyo” and “Foreman”, from the MPEG organization. The “M9073” version of the MPEG-4 standard evaluation program, developed by National

Chiao-Tung University, is used as a reference system to test and compare its compression performance with the proposed method. The standard tested video clips “Akiyo” and “Foreman” are 24-bit true-color and 100 frames long; “Akiyo” is of CIF format (352 × 288 pixels) and “Foreman” is of QCIF format (176 × 144 pixels). The data are measured on the same testing platform. Table 5.3 and Table 5.4 compare the compression performance of MPEG-4 and the proposed method.

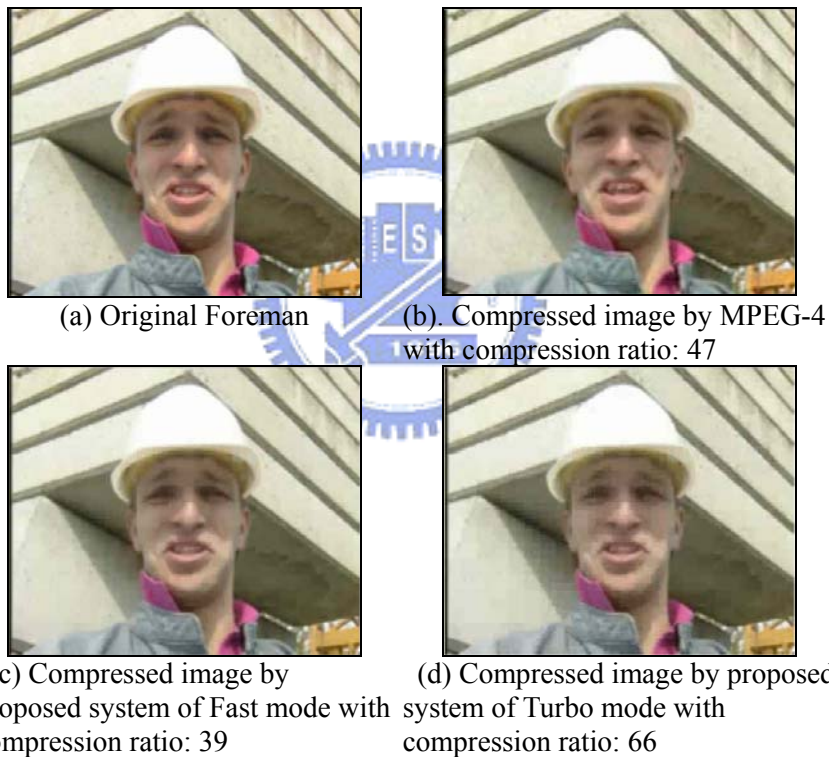


Figure 5.10. Results of compressing the standard testing clip – “Foreman” (QCIF format, 176\*144)

Table 5.3. Comparisons of performance between proposed method and MPEG-4 (Akiyo, CIF format, 352\*288)

Method	Settings	Frame rate (fps)	Compression Ratio
MPEG-4	Bit-rate: 384K bits	19.9	184
Proposed method	Fast mode	99	176
Proposed method	Turbo mode	122	257

Table 5.4. Comparisons of performance between proposed method and MPEG-4 (Foreman, QCIF format, 176\*144)

method	Settings	Frame rate (fps)	Compression Ratio
MPEG-4	Bit-rate: 384K bits	65	47
Proposed method	Fast mode	394	39
Proposed method	Turbo mode	491	66

Figure 5.11, Table 5.3, and Table 5.4 reveal that, for a given compression ratio and arbitrary video format with images of various sizes, the proposed method achieves a frame rate that is about five times higher than that of the MPEG-4 system. Therefore, the proposed method exhibits extremely high efficiency of compression. Figure 5.9 and Figure 5.10 show that the two systems yield very similar visual results. Restated, the proposed method of compression provides is effective for developing digital surveillance systems, and can also yield smoothly flowing images in multi-channel surveillance systems.

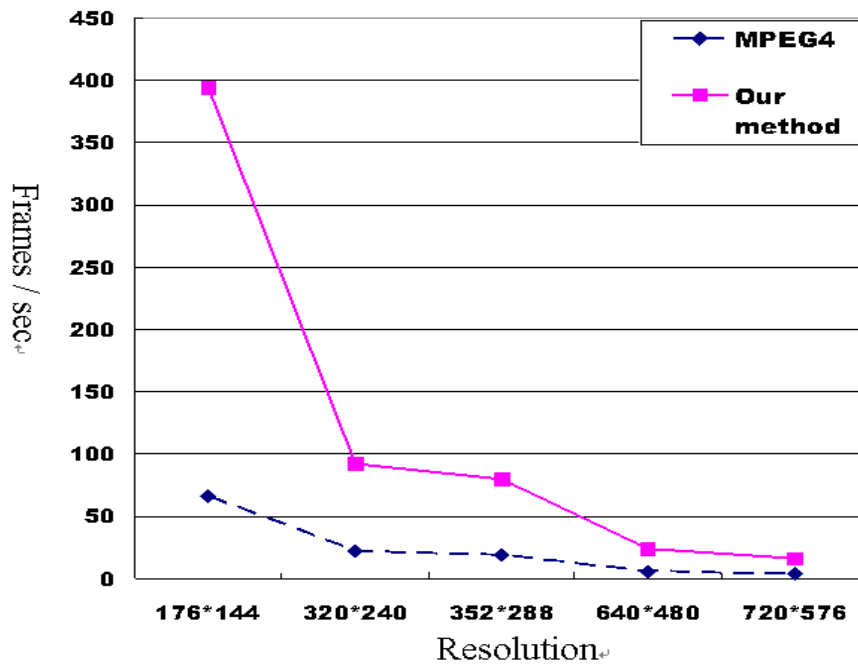


Figure 5.11. Comparison of compression speeds between MPEG-4 method and the proposed method



## 5.4 Video Codec Software Component

The video codec is implemented as an ActiveX COM [97] component so that it can be efficiently and easily embedded into various multimedia applications. This component is implemented with dual interfaces - IDispatch and IVCodecX. These properties, and the application of methods rapidly to develop application software, including network applications, greatly facilitate development. Using IPictureDisp image pointer, created by Microsoft, each frame is encoded and decoded according to the compAddFrame and decompExtFrame methods in the proposed ActiveX COM-based video codec component. A series of codes from compressed frames is then stored by the compToFile method. The codes played back from the storage media using the decompFromFile method. The storage and

playback processes also include two data stream methods, `compToArray` and `decompFromArray`, for network surveillance system applications on the server side and the client side. Besides, some properties are provided for performance tuning and frame synchronization. Figure 5.12 illustrates the above methods and their corresponding applications.

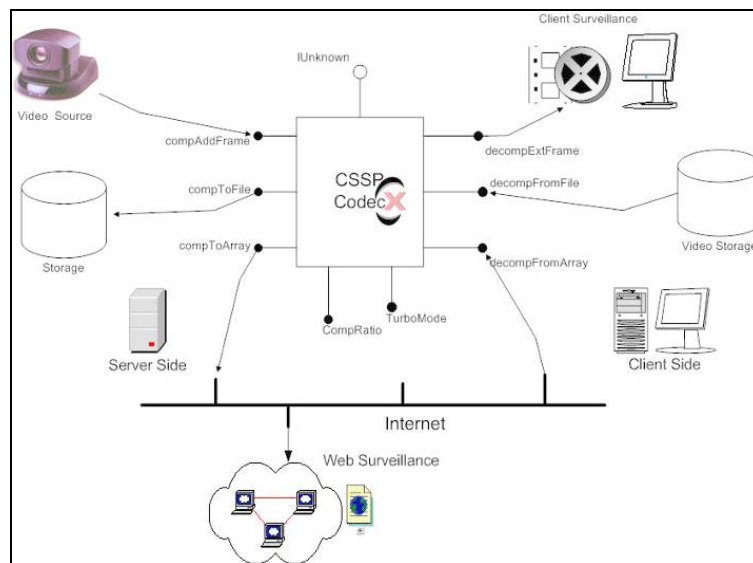


Figure 5.12. Video codec component and its applications

In this work, the video codec component is implemented using Visual C++ .Net using Microsoft ATL [98]. The main advantage of ATL is its lightweight, particularly crucial to the distribution of the COM component via the Internet. The component can be embedded into web pages so the web surveillance can be easily performed. Besides, the ActiveX component itself can be easily used in a visual software developing environment, including Visual Basic. Figure 5.13 presents the Visual Basic environment for developing a simple surveillance application.

The smallness of this component allows it to be downloaded fast over the Internet. For

example, as presented in Figure 5.14, a web-based surveillance system can be developed by integrating this codec component with Microsoft Internet application development tools. It can thereby be applied in surveillance systems, video conferencing, video phones, and long-distance education.

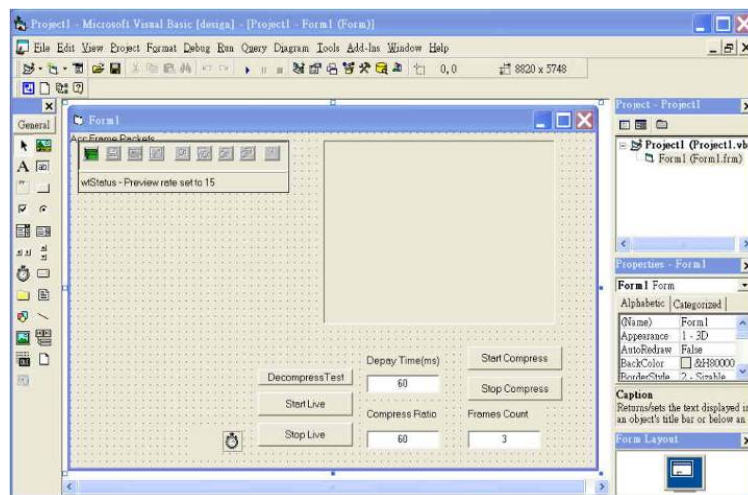


Figure 5.13. Visual Basic environment for developing surveillance applications



Figure 5.14. Example of web-surveillance application



## 5.5 Implementation of an Intelligent Video Surveillance System

The proposed video codec, which was developed for surveillance systems and mobile carriers, as shown in Figure 5.16, is integrated with our intelligent surveillance system to extent its range of applications. As presented in Figure 5.15, a multi-channel surveillance system was developed using the proposed video compression codec, to capture four channels of video in real-time and to record a total of 120 frames (with CIF format) per second at full-speed video compression on a system with a 2.4 GHz Pentium CPU and 512 MB RAM. As presented in Figure 5.15(c), this system also supports surveillance recording and random time-slot playback.

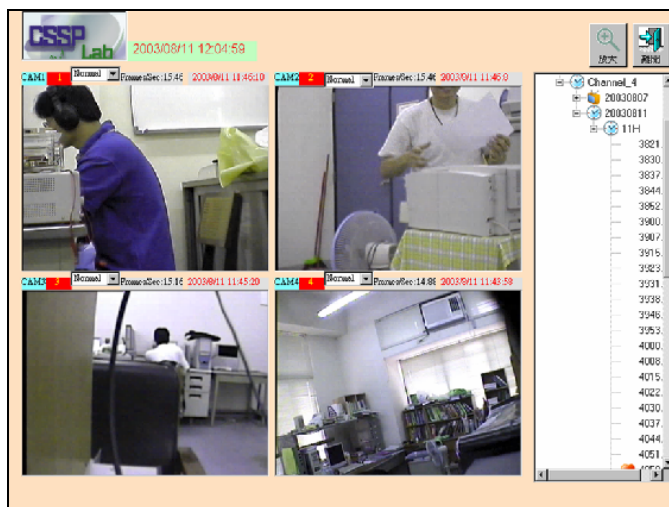
Figure 5.16 shows that the mobile carrier incorporates a CCD camcorder and wireless image transmission capability using RF module. The mobility of the carrier ensures that an image can be captured of the entire surveillance area. This system can also perform remote monitoring via the Internet, and can remotely control the mobile carrier in the same way, to capture images of the area of interest. The control signal is transmitted via a wireless system from the server side of the surveillance system to the mobile carrier. The monitoring user controls the forward, backward and rotational motion of the carrier using buttons on the web page. The CCD camcorder is placed on a rotating mechanism. The user can change the camcorder's viewing angle by controlling this mechanism. Selecting viewing angle is especially important when some of the surveyed area is obscured by furniture, as depicted in Figure 5.17.



(a). Prototype of our intelligent surveillance system



(b). Operation on real-time multi-channel surveillance



(c). Operation on random time-slot playback

Figure 5.15. Implementation of Intelligent surveillance system



Figure 5.16. Mobile carrier with a CCD camcorder

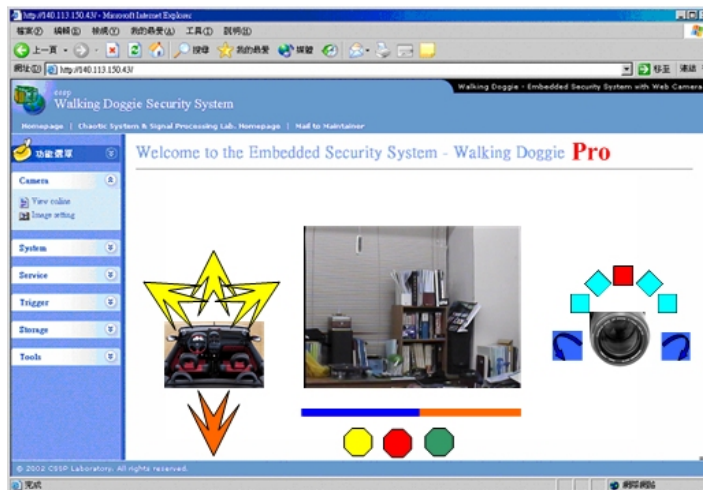


Figure 5.17. The controlling interface of the monitoring user

## **Chapter 6. CONCLUSIONS AND FUTURE WORKS**

In this chapter, we state a brief conclusion of this dissertation and present some directions for future works on the research studies presented in this dissertation.

### **6.1 Multi-level thresholding approaches for image segmentation**

This study has developed an efficient combinatorial scheme for reducing the computation timings and parameterizing the desired number of thresholds in multilevel thresholding, and an effective automatic multilevel thresholding method for image segmentation. This proposed combinatorial scheme can effectively avoid evaluating redundant threshold sets, substantially suppress the computation timings for obtaining each criterion function value of the potential threshold sets, and thereby significantly reduces the computation timings of obtaining the optimal set of threshold values. We apply the proposed combinatorial scheme on multilevel thresholding using three well-known thresholding criterion functions - the between-class variance criterion, the maximum entropy criterion, and the minimum error thresholding criterion, and then compare the objective and subjective evaluations, and computation costs of their performance. For automatic multilevel thresholding, an effective discriminant-analysis-based measure is utilized for determining the separability among the thresholded classes of gray levels, and is able to simplify the problem of determining the number of object images that should be segmented. The image is then recursively thresholded into more detailed segmented object images, until the separability measure is satisfied. Accordingly, all the objects of interest can be thresholded into separate segmented images, and they are ensured to be well-separated. The proposed automatic multilevel thresholding method is found to be effective in analyzing and thresholding the

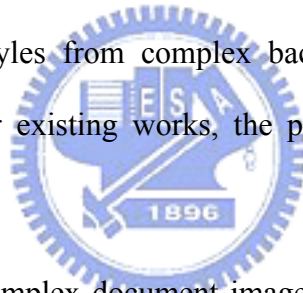
histogram of the image. It is also computationally simple and efficient for automatically determining the appropriate number of thresholding levels and performing thresholding on the image. Moreover, when applied to real-life complex document images, the proposed automatic multilevel thresholding method can successfully extract text strings, with various illuminations from overlaying non-text objects or complex backgrounds. This can also facilitate further text extraction for document analysis. From the experimental results, and our comparisons to other fixed-level criterion-based methods, we have demonstrated the effectiveness and advantages of the proposed multilevel thresholding approaches.

To achieve more effective and satisfactory performance for image segmentation issues, the future research directions of this study may be: (1) to extend the proposed algorithms on color models for applying on color image segmentation, it might be useful to integrate color space reduction concepts, such as the use of the moment-preserving techniques [99], and color contrast information for better fitting of human's visual perception [100]. (2) To more accurately determine the number of homogeneous objects for segmentation, it may be of use to analyze and detect the modes on the histogram of images in a more accurate way by applying the concepts of mean-shift approaches [101][102].

## **6.2 A multi-plane segmentation approach for text extraction in complex document images**

In this study, a new technique for segmenting and extracting textual objects from real-life complex document images is presented in this study. The proposed approach first segregates textual regions, non-text objects such as graphics and pictures, and background textures from the document image by decomposing it into distinct object planes. This decomposition process consists of two stages: automatic localized histogram multilevel thresholding, and multi-plane region matching and assembling. The first stage applies the localized histogram multilevel thresholding procedure to discriminate different textual

objects, non-textual objects, and background components in each block region into separate sub-block regions. In the second stage, the multi-plane region matching and assembling process organizes these obtained sub-block regions into object planes according to their respective features. A text extraction procedure is then applied on the resultant planes to extract textual objects with different characteristics in the corresponding planes. The document image is processed regionally and adaptively according to its local features, and thus detailed characteristics of extracted textual objects can be well-preserved, especially small characters with thin strokes. It also allows textual objects that touch graphical and pictorial background objects with uneven, gradational, and sharp variations in contrast, illumination, and texture to be well handled. When applied to real-life complex document images, the proposed approach exhibit its robustness on extracting textual objects of various illuminations, sizes, and font styles from complex backgrounds. From the experimental results and comparisons to other existing works, the proposed approach demonstrates its effectiveness and advantages.



To enhance our proposed complex document image analysis system on more versatile applications, future works on this study may focus on: (1) develop a more sophisticated text layout analysis technique to handle text-lines with various shapes, orientations, and layout, and thus more extensive types of real-life documents can be handled. (2) To process the color complex document images, it is necessary to extend the proposed multi-plane segmentation approach to adopt on color document images by adopting the concepts of color space analysis techniques [103], or fuzzy connectedness concepts on the space of color features on segmented block regions [104]. (3) To develop a versatile document content analysis and archive system, it is needed to develop and integrate the document content storage and retrieval techniques.



### **6.3 Vision-based nighttime vehicle detection for driver assistance**

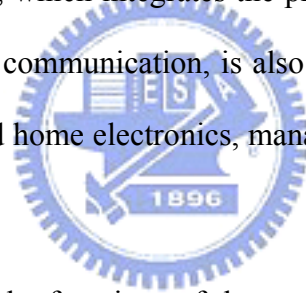
This study has presented an efficient nighttime vehicle detection method for identifying the vehicles by locating their pairing headlights and taillights in nighttime road environments. An effective object segmentation technique based on automatic multilevel histogram thresholding is applied for extracting illuminant bright objects of interest, especially vehicle lights. This technique demonstrates its robustness and adaptability for dealing with various illuminated conditions at night. Then, by utilizing the symmetrization, alignment and pairing characteristics of vehicle lights, discriminating pairing headlights and taillights to locate actual vehicles can be achieved by the projection-based spatial clustering and rule-based identification methods. Finally, the distances, relative positions, and motion directions between each of the detected target vehicles appeared ahead and the camera-assisted car are estimated and tracked. Experimental results demonstrate that the proposed system can effectively detect vehicles in front of the camera-assisted car in various nighttime road environments. This system provides beneficial information for assisting the driver to perceive surrounding traffic conditions outside the car during nighttime driving, and can also be applied to a versatile control scheme for the apparatus of vehicles to facilitate autonomous driving.

To enhance our proposed system to be a more efficient vision-based intelligent driver assistance system, future studies on this study may possibly focus on: (1) for the purpose of autonomous driving, it is essential to develop a intelligent control scheme of in-vehicle apparatus for autonomous driving, collision avoidance and automatic cruise speed control systems. (2) An automatic calibration approaches for CCD cameras of the vision system is also necessary for the effectiveness and feasibility for estimating distances, positions and relative motion information of the detected target vehicles.



## **6.4 Real-time wavelet-based video compression approach for video surveillance**

In this study, we have presented a new real-time wavelet-based video compression method for use in intelligent video surveillance systems. Based on the low-complexity and low-memory-cost wavelet-based coding scheme and motion compression strategy, the proposed video codec achieves high vision quality, high compression speed and high compression ratio. Experimental results on video compression performance demonstrate the effectiveness and efficiency of the proposed video codec. Then the ActiveX COM component technique is also implemented and integrated with the proposed video codec to realize multimedia, internet applications and many other video-intensive applications. Furthermore, an intelligent surveillance system, which integrates the proposed wavelet-based video codec, computer peripherals and mobile communication, is also developed in this study. Therefore, the future e-Home with controlled home electronics, managed video/audio systems and home security will be realized.



For the purpose of enhance the functions of the proposed video surveillance study, our future works should focus on: (1) develop and integrate the object detection, classification, recognition, and tracking techniques. (2) To achieve more effectively content-based coding of the surveillance video frames, an efficient region-of-interest ROI coding techniques is necessary for adopting on interesting objects, such as human beings for home security applications, and target vehicles for traffic surveillance applications.

## REFERENCES

- [1] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision vol. I*, Addison-Wesley Co., Inc., 1992.
- [2] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing: Analysis and Machine Vision*, 2<sup>nd</sup> Ed., Thomson-Engineering, 1998.
- [3] Linda G. Shapiro, George C. Stockman, *Computer Vision*, Prentice Hall, 2001.
- [4] D. Doermann, "The indexing and retrieval of document images: a survey," *Comput. Vision Image Understand.*, vol. 70, pp. 287-298, 1998.
- [5] H. Bunke and P.S.P. Wang (Eds.), *Handbook of Character Recognition and Document Image Analysis*, World Scientific, Singapore, 1997.
- [6] I. Masaki (Ed.), *Vision-based Vehicle Guidance*, New York: Springer-Verlag, 1992.
- [7] M. Bertozzi and A. Broggi, "Vision-based vehicle guidance", *IEEE Comput.*, vol. 30, pp. 49-55, 1997.
- [8] A. Broggi, M. Bertozzi, A. Fascioli, G. Conte, *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*, Singapore: World Scientific, 1999.
- [9] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection: A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694-711, 2006.
- [10] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen, "A survey of thresholding techniques," *Computer Vis., Graph. Image Process.*, vol. 41, pp. 233-260, 1988.
- [11] S. U. Lee and S. Y. Chung, "A comparative performance study of several global thresholding techniques for segmentation," *Computer Vis., Graph. Image Process.*, vol. 52, pp. 171-190, 1990.
- [12] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-8, pp. 62-66, 1978.
- [13] J. Kapur, P.K. Sahoo, A.K.C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vis., Graph. Image Process.*, vol. 29, pp. 273-285, 1985.
- [14] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognit.*, vol. 19, pp. 41-47, 1986.

- [15] J.C. Yen, F.J. Chang and S. Chang, "A new criterion for automatic multilevel thresholding," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 370-378, 1995.
- [16] P. Sahoo, C. Wilkins and J. Yeager, "Threshold selection using Renyi's entropy," *Pattern Recognit.*, vol. 30, no. 1, 71-84, 1997.
- [17] L.K. Huang and M.J. Wang, "Image thresholding by minimizing the measure of fuzziness," *Pattern Recognit.*, vol. 28, pp. 41-51, 1995.
- [18] H.D. Cheng, J.R. Chen, and J. Li, "Threshold selection based on fuzzy c-partition entropy approach," *Pattern Recognit.*, vol. 31, no. 7, pp. 857-870, 1998.
- [19] H.D. Cheng, Y.H. Chen, and Y. Sun, "A novel fuzzy entropy approach to image enhancement and thresholding," *Signal Process.*, vol. 75, pp. 277-301, 1999.
- [20] H. Yan, "Unified formulation of a class of optimal image thresholding techniques," *Pattern Recognit.*, vol. 29, no. 12, pp. 2025-2032, 1996.
- [21] W.-H. Tsai, "Moment-preserving thresholding: A new approach", *Comput. Vis., Graph. Image Process.*, vol.29, pp.277-393, 1985.
- [22] S.S. Reddi, S.F. Rudin, and H.R. Keshavan, "An optimal multiple threshold scheme for image segmentation," *IEEE Trans. Syst. Man Cybernet.*, vol. SMC-14, pp. 661-665, 1984.
- [23] M. Cheriet, J.N. Said, and C.Y. Suen, "A recursive thresholding technique for image segmentation," *IEEE Trans. Image Process.*, vol. 7, no. 6, pp. 918-921, 1998.
- [24] D.M. Tsai, "A fast thresholding selection procedure for multimodal and unimodal histograms", *Pattern Recognit. Lett.*, vol.16, no.6, pp.653-666, 1995.
- [25] L. Cao, Z.K. Shi, and Cheng E.K.W., "Fast automatic multilevel thresholding method", *Electronics Lett.*, vol.38, no.16, pp.868-870, 2002.
- [26] M. Fleury, L. Hayat, A.F. Clark, "Parallel entropic auto-thresholding," *Image Vis. Comput.*, vol. 14, pp. 247-263, 1996.
- [27] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 312-314, 1995.
- [28] O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 1191-1201, 1995.
- [29] L. O' Gorman, and R. Kasturi, *Document Image Analysis*, IEEE Computer Society Press, Silver Spring, MD, 1995.

- [30] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 910-918, 1988.
- [31] J. L. Fisher, S. C. Hinds and D. P. D'Amato, "Rule-based system for document image segmentation," in *Proc. 10<sup>th</sup> Int. Conf. Pattern Recognit.*, pp. 567-572, 1990.
- [32] F. Y. Shih, S. S. Chen, D. C. D. Hung and P. A. Ng, "Document segmentation, classification and recognition system," in *Proc. IEEE Int. Conf. Syst. Integr.*, pp. 258-267, 1992.
- [33] Q. Yuan and C.L. Tan, "Text extraction from gray scale document images using edge information," in *Proc. 6th Int'l Conf. Document Analysis and Recognit.*, pp. 302-306, 2001.
- [34] V. Wu, R. Manmatha, and E.M. Riseman, "Textfinder: an automatic system to detect and recognize text in images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1224-1229, 1999.
- [35] Y. M. Y. Hasan, and L. J. Karam, "Morphological text extraction from images," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1978-1983, 2000.
- [36] M. Pietikinen and O. Okun, "Edge-based method for text detection from complex document images," in *Proc. 6th Int'l Conf. Document Analysis and Recognit.*, pp. 286-291, 2001.
- [37] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," *Pattern Recognit.*, vol. 28, no. 10, pp. 1523-1535, 1995.
- [38] A. K. Jain, and B. Yu, "Automatic text location in images and video frames," *Pattern Recognit.*, vol. 31, no. 12, pp. 2055-2076, 1998.
- [39] C. Strouthopoulos, N. Papamarkos, and A. E. Atsalakis, "Text extraction in complex color documents," *Pattern Recognit.*, vol. 35, pp. 1743-1758, 2002.
- [40] H. Yang, and S. Ozawa, "Extraction of bibliography information based on the image of book cover," *IEICE Trans. Info. Syst.*, vol. E82-D, no. 7, pp. 1109-1116, 1999.
- [41] H. Hase, M. Yoneda, S. Tokai, J. Kato, and C. Y. Suen, "Color segmentation for text extraction," *Int'l. J. Doc. Anal. Recognit.*, vol. 6, no. 4, pp. 271-284, 2004.
- [42] A. Broggi, M. Bertozzi, A. Fascioli, C.G.L. Bianco, A. Piazzzi, "Visual perception of obstacles and vehicles for platooning", *IEEE Trans. Intell. Transport. Syst.*, vol. 1, pp. 164-176, 2000.

- [43] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, T. Graf, "High accuracy stereo vision for far distance obstacle detection", in *Proc. IEEE Intell. Vehicle Symp.*, pp. 292-297, 2004.
- [44] U. Franke, and S. Heinrich, "A study on recognition of road lane and movement of vehicles using vision system", in *Proc. SICE Annual Conf.*, Japan, pp. 38-41, 2001.
- [45] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle", *Mach. Vision Appl.*, vol. 12, pp. 69-83, 2000.
- [46] B.-F. Wu and C. T. Lin, "A fuzzy vehicle detection based on contour size similarity", in *Proc. IEEE Intell. Vehicle Symp.*, pp. 496-501, 2005.
- [47] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. Image Process.*, vol. 1, pp. 205-220, 1992.
- [48] S. Li, and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding", *IEEE Trans. Circuit. and Syst. Video Tech.*, vol. 10, pp. 725-743, 2000.
- [49] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients", *IEEE Trans. Signal Process.*, vol. 41, pp. 3445-3462, 1993.
- [50] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE Trans. Circuit. and Syst. Video Tech.*, vol. 6, no. 3, pp. 243-250, 1996.
- [51] R. Koenen (Ed.), *Overview of the MPEG-4 Version 1 Standard*, ISO/IEC JTC1/SC29/WG11 N1909," MPEG97, 1997.
- [52] T. Sikora, "The MPEG-4 video standard verification model", *IEEE Trans. Circuit. Syst. Video Tech.*, vol. 7, pp. 19-31, 1997.
- [53] ISO/IEC, *ISO/IEC 15444-1: Information Technology-JPEG2000 image coding system*, 2000.
- [54] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview", *IEEE Trans. Consumer Electron.*, vol. 46, pp. 1103-1127, 2000.
- [55] R. A. Fisher, "The use of multiple measurements in taxonomic problems", *Annals of Eugenics*, vol.7, pp.179-188, 1936.
- [56] B.-F. Wu, Y.-L. Chen, and C.-C. Chiu, "A discriminant analysis based recursive automatic thresholding approach for image segmentation", *IEICE Trans. Info. Syst.*, vol. E88-D, no. 7, pp. 1716-1723, 2005.

- [57] B.-F. Wu, Y.-L. Chen, and C.-C. Chiu, "Efficient implementation of several multilevel thresholding algorithms using a combinatorial scheme", *Int'l J. Computer. Appl.*, vol. 28, no. 3, pp. 259-269, 2006.
- [58] B.-F. Wu, Y.-L. Chen, and C.-C. Chiu, "A new region-based segmentation method for complex document image analysis", *Int'l. J. Comput. Sci, Eng.*, vol. 1, no. 1, pp. 34-44, 2005.
- [59] Y.-L. Chen, C.-C. Chiu, and B.-F. Wu, "Complex document image segmentation using localized histogram analysis with multi-layer matching and clustering", in *Proc. of IEEE Conf. on Syst., Man Cybernet.*, vol. 4, pp. 3063-3070, Netherlands, 2004.
- [60] B.-F. Wu, Y.-L. Chen, and C.-C. Chiu,, "Multi-layer segmentation of complex document images", *Int'l. J. Pattern Recognit. Artificial Intell.*, vol. 19, no. 8, pp. 997-1025, 2005.
- [61] Y.-L. Chen, and B.-F. Wu, "Text extraction from complex document images using the multi-plane segmentation technique", in *Proc. IEEE Conf. on Syst., Man Cybernet.*, pp. 3540 – 3547, Taiwan, 2006.
- [62] Y.-L. Chen, and B.-F. Wu, "A multi-plane segmentation approach for text extraction from complex document images", submitted for publication in *Comput. Vision Image Understand.*
- [63] B.-F. Wu, Y.-L. Chen, and Y.-H. Chen, "A fast intelligent nighttime vehicle-light recognition system based on computer vision", in *Proc. 14th Automation Tech. Conf.*, vol. 2, pp. J-29-34, Taiwan, June 2006.
- [64] Y.-L. Chen, Y.-H. Chen, C.-J. Chen, and B.-F. Wu, "Nighttime vehicle detection for driver assistance and autonomous vehicles", in *Proc. 18th IAPR Int'l Conf. Pattern Recognit.*, vol. 1, pp. 687 – 690, Hong Kung, 2006.
- [65] B.-F. Wu, Y.-L. Chen, C.-M. Hsieh, Y.-H. Chen, and C.-J. Chen, "Real-Time image segmentation and analysis for vehicle light detection on a moving vehicle for nighttime driving", submitted for publication in *Int'l J' Robotics & Automation.*
- [66] B.-F. Wu, Y.-L. Chen, C.-J. Chen, C.-C. Chiu and C.-Y. Su, "A real-time wavelet-based video compression approach to intelligent video surveillance systems", *Int'l J. Computer Appl. in Tech.*, vol. 25, no. 1, pp. 50-64, 2006.
- [67] Rao, C. R., "The utilization of multiple measurements in problems of biological classification", *J. of the Royal Statistic. Soc. series B*, vol.10, pp.159-203, 1948.
- [68] M.D. Levine, and A.M. Nazif, "Dynamic measurement of computer generated image segmentation", *IEEE Trans. Pattern Anal. Machine Intell.*, vol.7, pp.155-164, 1985.



- [69] J.H. Conway and R.K. Guy, Choice Numbers, In *The Book of Numbers* (New York: Springer-Verlag, pp. 67-68, 1996.
- [70] P.J. Chase, "Algorithm 382: Combinations of M out of N Objects [G6]", *Comm. of ACM*, vol. 13, no. 6, pp. 368, 1970.
- [71] Y. Liu and S. N. Srihari, "Document image binarization based on texture features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 540-544, 1997.
- [72] J. R. Parker, "Gray level thresholding in badly illuminated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 813-819, 1991.
- [73] J. Ohya, A. Shio, S. Akamatsu, "Recognizing characters in scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 214-220, 1994.
- [74] M. Kamel and A. Zhao, "Extraction of binary character/graphics images from grayscale document images," *CVGIP : Graph. Models and Image Process.*, vol. 55, no. 3, pp. 203-217, 1993.
- [75] N. B. Venkateswarlu and R. D. Boyle, "New segmentation techniques for document image analysis," *Image Vis. Comput.*, vol. 13, no. 7, pp. 573-583, 1995.
- [76] X. Ye, M. Cheriet, C. Y. Suen, "Stroke-model-based character extraction from gray-level document images," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1152-1161, 2001.
- [77] A. Dawoud and M. Kamel, "Iterative multi-model sub-image binarization for handwritten character segmentation," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1223-1230, 2004.
- [78] A. Amin and S. Wu, "A robust system for thresholding and skew detection in mixed text/graphics documents," *Int'l. J. Image Graph.*, vol. 5, no. 2, pp. 247-265, 2005.
- [79] B.-F. Wu, C.-C. Chiu, and Y.-L. Chen, "Compound document compression algorithms for text/background overlapping images," *IEE Proc. Vis. Image Signal Process.*, vol. 151, no. 6, pp. 453- 459, 2004.
- [80] R. Kasturi and M. M. Trivedi, *Image Analysis Applications*, Marcel Dekker, New York, 1990.
- [81] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, vol. 2, second Ed., Academic Press, New York, 1982.
- [82] R.R. Yager and D.P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst. Man Cybern.*, vol. 24, no. 8, pp. 1279-1284, 1994.



- [83] S.L. Chiu, "Extracting fuzzy rules for pattern classification by cluster estimation," in *Proc. 6<sup>th</sup> Int'l. Fuzzy Syst. Assoc. World Congr.*, pp. 1-4, 1995.
- [84] N.R. Pal and D. Chakraborty, "Mountain and subtractive clustering method: improvements and generalization," *Int'l. J. Intell. Syst.*, vol. 15, pp. 329-341, 2000.
- [85] K. Suzuki, I. Horiba, and N. Sugie, "Linear-time connected-component labeling based on sequential local operations," *Comput. Vision Image Understand.*, vol. 89, pp. 1-23, 2003.
- [86] J. Ha, R.M. Haralick, and I. Phillips, "Document page decomposition by the bounding-box projection technique," in *Proc. Third Int'l Conf. Document Analysis and Recognit.*, pp. 1119-1122, 1995.
- [87] J. Ha, R.M. Haralick, and I. Phillips, "Recursive X-Y Cut using bounding boxes of connected components," in *Proc. Third Int'l Conf. Document Analysis and Recognit.*, pp. 952-955, 1995.
- [88] T. Pavlidis and J. Zhou, "Page segmentation and classification," *Comput. Vis. Graph. Image Process.*, vol. 54, no. 6, pp. 484-496, 1992.
- [89] F. Y. Shih and S. S. Chen, "Adaptive document block segmentation and classification," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 26, no. 5, pp. 797-802, 1996.
- [90] J. S. Stam, "Headlamp control to prevent glare", U.S. Patent No. 6,861,809, 2005.
- [91] J. S. Stam, M. W. Pierce, H. C. Ockerse, "Image processing system to control vehicle headlamps or other vehicle equipment", U.S. Patent No. 6,868,322, 2005.
- [92] G. P. Stein, O. Mano and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy", in *Proc. IEEE Intell. Vehicle Symp.*, pp. 120-125, 2003.
- [93] C.-Y. Su and B.-F. Wu, "Image coding based on embedded recursive zerotree", in *Proc. Int'l Symp. Multi-Tech. Info. Process.*, pp. 387-392, Taiwan, 1997.
- [94] D. Taubman, "High performance image scalable image compression with EBCOT", *IEEE Trans. Image Process.*, vol. 9, pp. 1158-1170, 2000.
- [95] C.-Y. Su and B.-F. Wu, "A low memory embedded zerotree coding", *IEEE Trans. on Image Process.*, vol. 12, no.3, pp. 271-282, 2003.
- [96] D. Zhao, Y. K. Chan, and W. Gao, "Low-complexity and low-memory entropy coder for image compression", *IEEE Trans. Circuit. Syst. Video Tech.*, vol. 11, no. 10, pp. 1140-1145, 2001.

- [97] D. Box, *Essential COM*, Addison-Wesley publishers, 1997.
- [98] T. Armstrong and R. Patton, *ATL developer's guide*, 2<sup>nd</sup> Ed., M&T Books, IDG Books Worldwide, Inc, 2000.
- [99] C.-K. Yang, and W.-H. Tsai, "Reduction of color space dimensionality by moment-preserving thresholding and its application for edge detection in color images," *Pattern Recognit. Lett.*, Vol. 17, pp. 481-490, 1996.
- [100] H.-C. Chen, W.-J. Chien, and S.-J. Wang, "Contrast-based color image segmentation," *IEEE Signal Process. Lett.*, vol. 11, no. 7, pp. 641-644, 2004.
- [101] Y. Cheng, "Mean shift, mode seeking, and clustering", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, 1995.
- [102] H. Wang, and D. Suter, "False-Peaks-Avoiding Mean Shift Method for Unsupervised Peak-Valley Sliding Image Segmentation," in *Proc. 7<sup>th</sup> Digital Image Comput.: Tech. & Appl.*, pp. 581-590, 2003.
- [103] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation", in *Proc. IEEE Conf on Computer Vision and Pattern Recognit. (CVPR'97)*, pp.750-757, 1997.
- [104] A.S. Pednekar, and I. Kakadiaris, "Image segmentation based on fuzzy connectedness using dynamic weights", *IEEE Trans. Image Process.*, vol.15, no. 6, pp. 1555-1562, 2006.

## CURRICULUM VITAE



博 士 生：陳彥霖(Yen-Lin Chen)

指導教授：吳炳飛(Bing-Fei Wu)

論文題目：影像處理與電腦視覺技術應用於複雜文件影像分析、夜間駕駛輔助、以及視訊監控系統之研究 (A Study of Image Processing and Computer Vision Techniques for Complex Document Image Analysis, Nighttime Driver Assistance, and Video Surveillance Systems)

### Educations

1. 82 年 9 月～85 年 6 月                      高雄市立高雄高級中學
2. 85 年 9 月～89 年 6 月                      國立交通大學電機與控制工程學系
3. 89 年 9 月～now                              國立交通大學電機與控制工程學研究所博士班

### Publications

#### Referred Journal Papers:

- Bing-Fei Wu, Yen-Lin Chen, and Chung-Cheng Chiu, “**Efficient implementation of several multilevel thresholding algorithms using a combinatorial scheme**”, *International Journal of Computers and Applications*, Vol. 28, No. 3, pp. 259-269, 2006.
- Bing-Fei Wu, Yen-Lin Chen, and Chung-Cheng Chiu, “**Multi-Layer Segmentation of Complex Document Images**”, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 19, No. 8, pp. 997-1025, 2005.
- 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**”, *International Journal of Computer Applications in Technology*, Vol. 25, No. 1, pp. 50-64, 2006.
- Bing-Fei Wu, Yen-Lin Chen, and Chung-Cheng Chiu, “**A Discriminant Analysis Based Recursive Automatic Thresholding Approach for Image Segmentation**”, *IEICE Transactions on Information and Systems*, Vol. E88-D, No.7, pp.1716-1723, 2005.

- Bing-Fei Wu, Yen-Lin Chen, and Chung-Cheng Chiu, “**A New Region-Based Segmentation Method for Complex Document Image Analysis**”, *International Journal of Computational Science and Engineering*, Vol. 1, No. 1, pp. 34-44, 2005.
- Bing-Fei Wu, Chung-Cheng Chiu, and Yen-Lin Chen, “**Compound Document Compression Algorithms for Text/Background Overlapping Images**”, *IEE Proceedings Vision, Image and Signal Processing*, Vol. 151, No. 6, pp. 453- 459, 2004.
- Bing-Fei Wu, Yen-Lin Chen, and Chung-Cheng Chiu, “**Recursive Algorithms for Image Segmentation Based on a Discriminant Criterion**”, *International Journal of Signal Processing*, Vol. 1, pp. 55-60, 2004.

#### Domestic Journal Papers:

- 吳炳飛, 陳彥霖, “**以視覺為基礎的即時夜間車輛偵測與駕駛輔助系統**”, *影像與識別*, Vol. 12, No. 2, pp. 89-113, 2006.

#### Submitted Journal Papers:

- Yen-Lin Chen and Bing-Fei Wu, “**A Multi-plane Segmentation Approach for Text Extraction from Complex Document Images**”, submitted for publication in *Computer Vision and Image Understanding*.
- Bing-Fei Wu, Yen-Lin Chen, Chih-Ming Hsieh, Yuan-Hsin Chen, and Chao-Jung Chen, “**Real-Time Image Segmentation and Analysis for Vehicle Light Detection on a Moving Vehicle for Nighttime Driving**”, submitted for publication in *International Journal of Robotics and Automation*.
- Yen-Lin Chen and Bing-Fei Wu, “**Nighttime Multiple Vehicle Detection and Tracking for Driver Assistance and Autonomous Driving**”, to be submitted to *IEEE Transactions on Vehicular Technology*.

#### Conference Papers:

- Yen-Lin Chen, and Bing-Fei Wu, “**Text Extraction from Complex Document Images Using the Multi-plane Segmentation Technique**”, in *Proceedings of the 2006 IEEE Conference on Systems, Man and Cybernetics*, pp. 3540 – 3547, Taipei, Taiwan, 2006.
- Yen-Lin Chen, Yuan-Hsin Chen, Chao-Jung Chen, and Bing-Fei Wu, “**Nighttime Vehicle Detection for Driver Assistance and Autonomous Vehicles**”, in *Proceedings of the 18<sup>th</sup> IAPR International Conference on Pattern Recognition (ICPR 2006)*, Vol. 1, pp. 687 – 690, Hong Kung, 2006.
- Bing-Fei Wu, Yen-Lin Chen, and Yuan-Hsin Chen, “**A Fast Intelligent Nighttime Vehicle-Light Recognition System Based on Computer Vision**”, in *Proceedings of the 2006 14<sup>th</sup> Automation Technology Conference*, Vol. 2, pp. J-29-34, Changhua, Taiwan, June 2006.

- Bing-Fei Wu, Yen-Lin Chen, Yuan-Hsin Chen, Chao-Jung Chen, and Chuan-Tsai Lin, “**Real-Time Image Segmentation and Rule-Based Reasoning for Vehicle Head Light Detection on A Moving Vehicle**”, in *Proceedings of the 7<sup>th</sup> IASTED International Conference on Signal and Image Processing (SIP 2005)*, pp. 388-393, Honolulu, Hawaii, USA, Aug. 2005.
- Yen-Lin Chen, Chung-Cheng Chiu, and Bing-Fei Wu, “**Complex Document Image Segmentation using Localized Histogram Analysis with Multi-Layer Matching and Clustering**”, in *Proceedings of the 2004 IEEE Conference on Systems, Man and Cybernetics*, Vol. 4, pp. 3063-3070, Hague, Netherlands, Oct. 2004.
- 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 Intelligent Video Surveillance Systems”, in *Proceedings of the 13<sup>th</sup> Automation Technology Conference*, pp.1123-1130, Taipei, Taiwan, June 2004.
- Bing-Fei Wu, Yen-Lin Chen, Chih-Hsu Yen, and Chung-Cheng Chiu, “**The Study on High Efficient Defense Information Compression and Encryption System**”, in *Proceedings of the 2004 Symposium on National Defense Industries*, Tainan, Taiwan, Nov. 2004 (in Chinese).
- Bing-Fei Wu, Yen-Lin Chen, Chung-Cheng Chiu and Chorng-Yann Su, “**A Novel Image Segmentation Method for Complex Document Images**”, in *Proceedings of the 16<sup>th</sup> IPPR Conference on Computer Vision, Graphics, and Image Processing (CVGIP2003)*, Kinmen, Taiwan, pp. 646-654, 2003.
- Bing-Fei Wu, Yen-Lin Chen, and Chung-Cheng Chiu, “**Multi-Layers Segmentation Method for Complex Document Images**”, in *Proceedings of the 5<sup>th</sup> International Conference on Computer Vision, Pattern Recognition and Image Processing*, North Carolina, USA, pp. 647-650, 2003.

#### Patents:

- 「**使用二維離散小波轉換的影像壓縮之系統**」, 吳炳飛、陳彥霖、陳昭榮、瞿忠正、蘇崇彥, 中華民國發明專利, 發明第 I 241074 號
- 「**一個基於電腦視覺的智慧型高速夜間車燈辨識系統**」, 吳炳飛、陳彥霖、陳元馨、陳昭榮, 中華民國發明專利, 申請號: 第 95116809 號
- 「**Real-time nighttime vehicle detection and recognition system based on computer vision**」, 吳炳飛、陳彥霖、陳元馨、陳昭榮, US Patent 美國發明專利, 申請號: 第 11/500,141 號

## Honors / Awards

- 2006 中華民國第十四屆自動化科技研討會(ATC2006)會議最佳論文獎入圍 - "一個基於電腦視覺的智慧型即時夜間車燈辨識系統"
- 2006 獲得 2005-2006 年度中華扶輪博士獎學金 16 萬元
- 2005 第一屆機動車輛創新設計競賽 銀質獎 - "具有肇事現場重建功能的影音行車紀錄器"
- 2004 獲得交通大學電機學院電機與控制工程研究所 93 學年度成績優異博士研究獎學金
- 2003 第十七屆龍騰知識經濟論文獎 資訊科技及應用類 金質獎 - "以小波為基礎之高速影像壓縮技術與其在智慧型影像監控系統的應用"
- 2003 研華文教基金會第五屆 TIC100 科技創新競賽 最佳科技創新事業銀質獎 - "數位影像監控系統"
- 2002 教育部 91 學年度 微電腦系統設計應用 大學組 優等獎 - "Walking Doggie-結合行動通訊之嵌入式網路門禁監控系統"

