

南 華 大 學

資訊管理學系碩士論文

植基於決策樹建構法之二維多重分閥值法

Two-Dimensional Multilevel Thresholding
Based on Decision Tree Construction



研 究 生：郭 晉 銘

指 導 教 授：吳 光 閔

洪 明 輝

中 華 民 國 九 十 二 年 六 月

南 華 大 學

碩 士 學 位 論 文

資 訊 管 理 學 系

植基於決策樹建構法之二維多重分閥值法

研 究 生：鄧晉銘

經考試合格特此證明

口試委員：

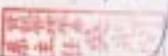
許威之
吳裕平
楊弘章

指導教授：



洪明輝

所 長：



口試日期：中華民國九十二年六月二日

誌 謝

經過二年的研究生涯能夠完成此論文，首先要感謝洪明輝教授。在研究過程有疑難時，皆能給予我明確的指導和適當的幫助，使我在專業領域上獲益良多。另外也謝謝楊弘章教授，對於我功課和論文上面的指導。同時亦感激吳光閔和許成之教授，對於本篇論文的建議與指教，使得本篇論文更加充實。

再者，謝謝黃小娟學姐在我的論文和程式寫作上給予我很大幫助。也謝謝同個實驗室中的伙伴，東坡、克寧及雅雯，大家相互討論，彼此關心與成長，讓我這二年的研究生活更加順遂。至於同窗二年的眾多好友們，真的很幸運能跟你們當同學，有你們陪伴的日子真是多采多姿啊！我相信未來大家的關心永遠在。

最後，更要感謝我的家人，父母無怨無尤的付出及時時的鼓勵與關懷，我才能無後顧之憂地專注於研究，並順利地取得學業。也謝謝曾經關心過我的許多朋友，我的成果來自於你們的幫助。

在此，僅以此論文，獻給所有支持我的人，並致以最衷心的感謝，我的喜悅將和你們一起擁有。

植基於決策樹建構法之二維多重分閥值法

學生：郭晉銘

指導教授：吳光閔
洪明輝

南 華 大 學資訊管理學系碩士班

摘 要

在本論文中，我們提出以決策樹建構法為基礎的二維多重分閥值法來解決影像分割的問題。此法使用非監督的叢聚樹演算法，將二維灰階統計直方圖當作資料空間，來分析像素值之間的相似程度並予以分群。這個方法包含了二個步驟：叢聚樹的建構以及修剪。在叢聚樹建構的過程中，我們利用資訊增益為準則，採取各個擊破策略法於二維灰階統計直方圖中分別求取各維度的最佳切割點，直到每個節點之資料量或相對密度皆符合自訂的條件為止。為了簡化叢聚樹以得到有意義的群數，因此使用區域與其相鄰區域之相對密度來進行修剪，以獲得最適當的分割結果。為了證明本方法的有效性，我們進行多組影像的實驗，其中包含了人造與真實影像。分割結果與 L. Cao 所提的快速自動多重分閥值法作比較。由實驗結果顯示，我們的方法確實能夠獲得較好的分割結果。

關鍵字：決策樹建構法、多重分閥值法、影像分割、叢聚樹、二維灰階統計直方圖

Two-Dimensional Multilevel Thresholding Based on Decision Tree Construction

Student : Chin-Ming Kuo

Advisors : Dr. Guang-Ming Wu
Dr. Ming-Huwi Horng

Department of Management Information
Nan-Hua University

ABSTRACT

In this thesis, we propose a two-dimensional multilevel thresholding based on decision tree construction for image segmentation. In this method, we use unsupervised cluster tree method to discriminate the different objects of the two-dimensional gray level histogram. Our method consists of two steps: cluster tree construction and cluster tree pruning. In the cluster tree construction step, we use the gain criterion to select the appropriate cut in the two-dimensional gray-level histogram until every node's number of data points or relative density conform to user-specify parameters. In order to simplify the tree to find meaningful clusters, we prune tree according to similarity between the region's relative density and adjacent regions' relative density. In experiments, we take many samples, including real and synthetic images, to demonstrate the effectiveness of our proposed method. The results compare with the ones of L. Cao et al. that propose the fast automatic multilevel thresholding method. Experimental results reveal that our method is superior to L. Cao's work.

Keyword: Decision tree construction, Multilevel thresholding, Image segmentation, Two-dimensional gray level histogram

目 錄

論文口試委員審定書	i
誌謝	ii
中文摘要	iii
英文摘要	iv
目錄	v
表目錄	vii
圖目錄	viii
第一章 導論	1
第一節 研究動機	1
第二節 研究目的	5
第三節 論文的組織	6
第二章 文獻探討	7
第一節 二維灰階統計直方圖	7
第二節 最大分割影像資訊分閾值法	9
第三節 二維熵值分割	10
第四節 快速自動多重分閾值法	12
第五節 決策樹	14
第六節 CLTree 演算法	20
第三章 植基於決策樹建構法之二維多重分閾值法	30
第一節 植基於決策樹建構法之二維多重分閾值法	30

第二節	叢聚樹的建立	33
第三節	叢聚樹的修剪	43
第四章	實驗結果與討論	52
第一節	實驗設計	52
第二節	人造影像之實驗結果與討論	52
第三節	真實影像之實驗結果與討論	54
第五章	結論與未來展望	62
第一節	結論	62
第二節	未來展望	62
參考文獻	64

表 目 錄

表 3-1	演算法變數名稱說明	32
表 4-1	停止建樹條件和修剪條件	54

圖 目 錄

圖 1-1	雙峰形態直方圖	2
圖 1-2	多峰形態直方圖	3
圖 2-1	胡椒：256×256 灰階影像	8
圖 2-2	胡椒影像的灰階統計直方圖	8
圖 2-3	胡椒影像的散佈圖	8
圖 2-4	分割出四個象限	12
圖 2-5	分閥線分割出二群	12
圖 2-6	資料空間的切割與其相對應的決策樹	16
圖 2-7	資訊增益的估算	18
圖 2-8	ID3 版本的決策樹演算法	19
圖 2-9	(A) 資料在資料空間中的分布狀況	21
	(B) 加入均勻分布的 N 點至資料空間 中以進行 CLTree 叢聚演算法	21
	(C) 不實際的加入 N 點, 而是 需要的時候才加以計算	21
圖 2-10	在不同測試節點加入不同數量的 N 點的考量	22
圖 2-11	加入不同數量的 N 點的法則	22
圖 2-12	加入不同數量的 N 點	23
圖 2-13	N 點數量的計算	24
圖 2-14	傳統資訊增益的問題	25
圖 2-15	決定最佳切割點示意圖	26
圖 2-16	決定最佳切割點演算法	26
圖 2-17	在此資料空間中應該有多少個叢聚呢?	28

圖 2-18	叢聚樹的修剪演算法	29
圖 3-1	演算法架構與流程	31
圖 3-2	(a) 麗娜原圖	34
	(b) 麗娜的灰階統計量直方圖	34
	(c) 去雜訊後之麗娜影像	34
	(d) 去雜訊後之灰階統計量直方圖	34
圖 3-3	二維灰階統計直方圖	35
圖 3-4	切割後的二維灰階統計直方圖	38
圖 3-5	麗娜影像之二維灰階統計直方圖分割過程	41
	(a) 第一次切割： $(129, 129)$	41
	(b) 第二次切割：區域 R1 切在 $(157, 156)$ ， 區域 R3 切在 $(90, 90)$	41
	(c) 第三次切割：區域 R11 切在 $(182, 181)$ ，區域 R33 切在 $(60, 61)$	41
圖 3-6	完整的叢聚樹	42
圖 3-7	分割為 16 群的影像	42
圖 3-8	初步修剪的叢聚樹	44
圖 3-9	6 群的影像，雜訊未處理	45
圖 3-10	相鄰區塊	45
圖 3-11	相鄰節點表格	46
圖 3-12	合併相鄰節點演算法	48
圖 3-13	(a) R113 合併到 R13 後之相鄰節點表格	49
	(b) R331 合併到 R31 後之相鄰節點表格	49
圖 3-14	修剪合併後的叢聚樹	50
圖 3-15	分割為 4 群之結果，雜訊未處理	50
圖 3-16	最終結果-共 4 群	51

圖 4-1	人造影像之實驗結果比較	53
圖 4-2	麗娜影像之實驗結果比較	57
圖 4-3	蝴蝶影像之實驗結果比較	58
圖 4-4	心血管影像之實驗結果比較	59
圖 4-5	青椒影像之實驗結果比較	60
圖 4-6	鸚鵡影像之實驗結果比較	61

第一章 導論

第一節 研究動機

隨著科技與網路的進步，愈來愈多的影像資料以數位化的方式出現，如醫學影像、衛星拍攝的照片，以及數位相片等等。鑑於數位影像之應用範圍日益廣闊，有關數位影像之處理技術，例如影像加強（Image Enhancement）、影像壓縮（Image Compression）、影像分割（Image Segmentation）、圖形識別（Image Recognition）、影像加密（Image Encryption）等等，也因而如雨後春筍般蓬勃發展起來[1][2][21]。

影像分割在數位影像處理領域裡是一個很重要的工作，通常必須從影像中萃取出我們想要辨識的主體以簡化後續影像分析工作的進行。例如可應用於自動目標識別、文字增強、電腦視覺中的工業應用，及生物醫學影像分析等。

而在這許多影像處理應用中，屬於物件的像素與屬於背景的像素其灰階度有相當大的不同。分閾值法（Thresholding）於是成為簡易但卻是有效的方法，因為只要找出影像中的灰階度之分閾值就可以將物件與背景區分出來。而分閾值法可以分為二類[16]：

一、二階分閾值法（Bilevel Thresholding）：

如果影像中的物件與背景之間的灰階值有明顯差異時，則利用單一的分閾值，即可把影像分割為物件和背景。在影像中，如果整個物件能從背景中區別出來，其灰階統計直方圖（Gray-Level Histogram）一定是呈現雙峰（bimodal）的形式，如圖 1-1 所示，而分閾值一定是發生在山谷（valley）的位置。因此影像中的每個像素其灰階度小於分閾值的即

為背景（物件），大於或等於分閾值為物件（背景）。雖然二階分閾值法很簡易，但當影像的灰階統計直方圖若不是呈現雙峰形式，則產生的分閾值不會有很有效的結果。

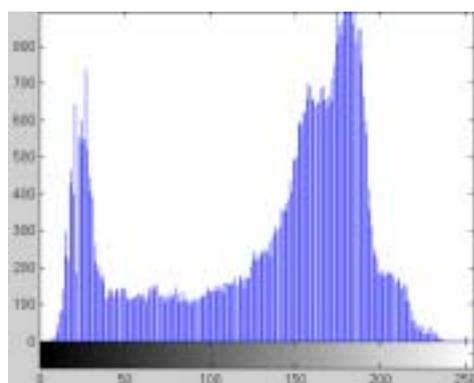


圖 1-1 雙峰形態直方圖

以數學的方式來表示。假設 $i(r,c)$ 為一張灰階影像， b_0 表示灰階值為 0， b_1 為 255， t 是灰階度的分閾值， $j(r,c)$ 表示分割結果的二階影像。

$$j(r,c) = \begin{cases} b_0 & \text{if } i(r,c) < t \\ b_1 & \text{if } i(r,c) > t \end{cases} \quad (1-1)$$

二、多重分閾值法（Multilevel Thresholding）：

當影像中包含不只一個物件，而是有多個物件時，即能利用這個方法把多個物件從背景中區分出來。數個不同的物件會使得灰階度直方圖呈現多峰（multimodal）的型態。如圖 1-2 所示，其分閾值也是發生於各個山谷中，因此它是二階分閾值法的延伸。

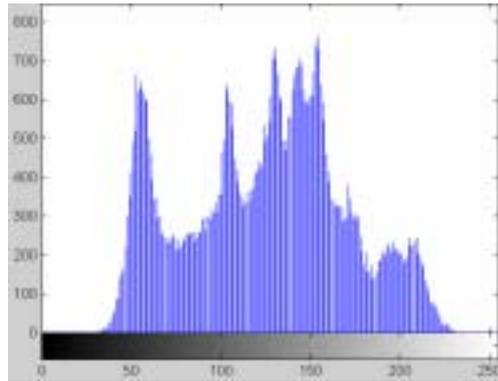


圖 1-2 多峰形態直方圖

而 Bulent Sankur[6]這位學者根據分閾值法所利用資訊的不同，把二階分閾值法分類為六種，分別是：

1. 基於直方圖形式的方法 (Histogram Shape-Based Methods): 利用直方圖的山峰、山谷，和其彎曲程度，這些屬性來進行分析。
2. 基於叢聚的方法 (Clustering-Based Methods): 利用叢聚分析將灰階度分為前景和背景二群。
3. 基於熵值的方法 (Entropy-Based Methods): 例如前景和背景區域熵值的使用，或利用原影像和其二階影像間交叉熵值來估算。
4. 基於物件屬性的方法 (Object Attribute-Based Methods): 尋找灰階影像和二階影像相似度的估算。例如模糊相似度，及物件的形式、邊界和數量等等。
5. 空間的方法 (The Spatial Methods): 利用前景像素和背景像素空間的資訊，也就是使用機率密度函式來考慮像素間的關係。
6. 局部的方法 (Local Methods): 分閾值主要是依據局部影像的特徵。

近二十年來，基於熵值的方法已經被很多學者提出來。Pun[18][22]這位學者是第一位提出的。其所提出的演算法是先了解灰階直方圖的事

前熵值 (Priori Entropy) 後，藉由最大化事後熵值 (Posteriori Entropy) 的上界來決定最佳的分閾值。這個方法有很多理論上的問題，關於這些問題的處理可參照文獻[11]。Kapur[11]這位學者之後提出了類似 Pun 的方法，它的方法是藉由物件和背景的事先熵值最大化來決定最佳分閾值。隨後 Kapur、Sahoo 和 Wong 這三位學者提出了一個觀點，”當有二張不同的影像但卻有相同的灰階度直方圖，因此也就有相同的分閾值，這個分閾值對二者都是合適的嗎？”因此他們建議加入第二個統計量或某些局部屬性到原來熵值的觀念以求取分閾值，對這個觀點會有更好的解決。Abutaleb[5]延伸了 Kapur 的方法到二維的熵值，利用像素灰階值和像素的局部平均灰階值來產生二維直方圖，藉由二維直方圖來獲得二維熵值。

以上這些學者所提出來的二階分閾值法皆能直接延伸到多重分閾值法，來解決影像中有多個物件的問題。然而，這些方法有二個主要的缺點[12]：

1. 影像灰階值應被分類成幾群很難以去決定，通常都是以監督方式來指定群數。
2. 計算時間非常耗時。

因此，我們提出了一個新的多重分閾值法。這個方法應用到資料採礦 (Data Mining) 中叢聚 (Clustering) 的觀念，以非監督方式，不需事先指定分群的數目。運用的演算法是改良自 IBM 研究中心 Buing Liu[7][8] 在 2000 年 3 月提出的一個嶄新的叢聚演算法的觀念為主，該演算法是以決策樹建構法為基礎，稱之為 CLTree (Clustering based on decision Trees) 本文所發展的多重分閾值法，以決策樹建構法為主，並導入 CLTree

的觀念，加以修改為能解決影像分割的問題。

第二節 研究目的

本文擬發展一套植基於決策樹建構法之二維多重分閥值法，來解決影像處理中分割多個物件的問題。這個方法包含了二個步驟：(1)叢聚樹的建構(2)叢聚樹的修剪。首先，在叢聚樹的建構中，我們是以決策樹建構法中的資訊增益 (Information Gain) 為建樹的準則。分別在二維的空間中，各在 X 軸和 Y 軸估算出分割點，即能切割出四個區域，以相同的方式遞迴去切割每個區域，直到符合停止建樹條件為止。每個切割的步驟都是同時在 X 軸和 Y 軸切割，因此所呈現出來的是一顆四元樹。因在切割過程中，可能會有過度切割 (over segmentation) 的情形。我們提出了三個修剪步驟。一開始先針對資料點數過小的區域先進行暫時地修剪，接下來再對餘留的區域做合併的動作，合併的動作會根據區域是否相鄰及相對密度是否相似來進行合併，最後的步驟再來處理暫時被修剪的區域。其結果將有效地將各個物件切割出來。本方法具有以下二種特點：

1. 其分割結果能形成最適當的群集。因為在分割影像的過程中，是以非監督方式產生群數，不需預先指定分割的群數。
2. 能有效地處理雜訊或離群值。雜訊幾乎都位於點數非常少的區域，我們的方法能夠把雜訊從影像中分割出來，並以我們提出來的的方法進行處理。

第三節 論文的組織

本論文共分為五個章節。除第一章導論，介紹本研究動機及目的外。在第二章文獻探討中，分為二部份。首先探討影像分割，包含了近年來相關的二維分閥值影像分割法、最大分割影像資訊分閥值法、及快速自動多重分閥值法；然後再介紹資料採礦（Data Mining）中的決策樹演算法和 CLTree 的相關研究。

第三章詳述我們提出的植基於決策樹建構法之二維多重分閥值法的完整步驟。

第四章為實驗結果與討論。以本文所提出的方法來處理影像分割的實驗結果並和 L. Cao. 等作者在 2002 年所提出的快速自動多重分閥值法作比較與結果討論。

最後第五章為結論與未來展望。

第二章 文獻探討

第一節 二維灰階統計直方圖

假如影像中包含了灰階度深的物體和灰階度淺的背景，則其一維直方圖即包含二個山峰，意味著物體和背景這二群的點數，而切割點則置於山谷上。因此分閥值只要靠近谷底的位置，物體和背景應該就能良好地區別開來，然而，谷底有時候是很難去找到的。

有很多方法已被提出來，主要的做法是將直方圖轉變，目的是使山谷的地方變得更深，這樣分閥值就變得更容易找到。而 Robert L. Kirby 和 Azriel Rosenfeld[18][20]在 1979 年美國電子電機工程師學會（the Institute of Electrical and Electronics Engineers, IEEE）中提出一個方法，它是基於灰階度和局部平均灰階度（Local Average Gray Level）在二維散佈圖的使用。在散佈圖（也稱為二維灰階統計直方圖）中，靠近對角線的像素值意味著這些點數的灰階度相當接近於其局部平均灰階度。因為具有同質性，這些像素點本質上有可能是影像中的物件或是背景。因此這些點數的灰階直方圖就應該有較深的谷底。換句話說，離對角線距離愈遠的像素值意味著這些點應該是置於物件或是背景的邊界，而這些點數的灰階直方圖就有明顯的山峰型式，來代替原來直方圖的山谷。

在圖 2-1 中，為一張 256×256 的灰階影像。圖 2-2 為圖 2-1 的灰階度直方圖，在直方圖中，左邊底端像素為 0 相當於黑色，右邊底端像素為 255 相當於白色。圖 2-3 為其二維灰階統計直方圖，灰階度位於 X 軸，灰階程度是由左到右遞增（0 255），而局部平均灰階度位於 Y 軸，灰階程

度是由下至上遞增 (0 255), 因此在右上方的點數意味著這些點數落在灰階度影像或局部灰階度影像其灰階度都是偏向 255, 而位於對角線的值代表這些點數為物體或背景, 離對角線的點數則為物體或背景的邊界值。



圖 2-1 胡椒：256×256 灰階影像

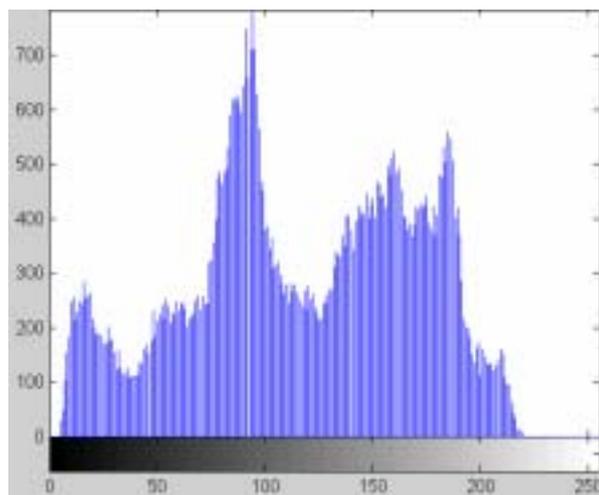


圖 2-2 胡椒影像的灰階統計直方圖

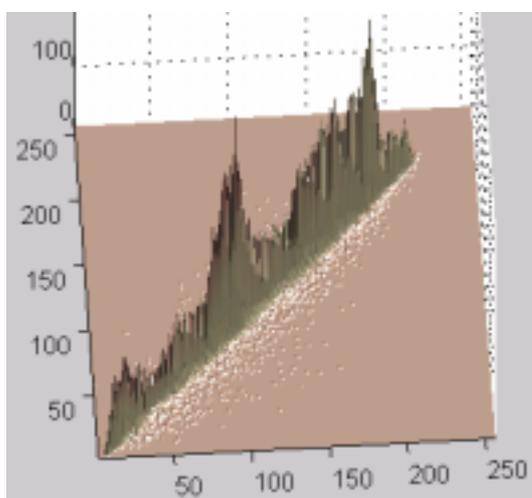


圖 2-3 胡椒影像的二維灰階統計直方圖

第二節 最大分割影像資訊分閾值法 (Maximum Segmented Image Information Thresholding)

Leung 和 Lam[9]利用資訊理論將影像分割從通訊上的觀點來考慮。影像分割處理被理解成一個資料處理 (data processing) 的步驟，因為主要是在灰階影像上做處理，來產生分割的影像。分割後的影像即包含了關於這個影像物件或景色 (scene) 某些資訊量。因此提出了一個分閾值準則，最大化其分割影像所包含的資訊量，以便獲得關於影像物件最大的資訊量。

在缺少任何的觀察下，也就是影像還沒分割前，物件的熵值可由公式(2-1)估算出來， X 為影像中隨機選擇的像素值， α 表示像素值屬於物件的機率，而 $1-\alpha$ 則表示像素值屬於背景的機率。

$$H(X) = -\alpha \log \alpha - (1-\alpha) \log (1-\alpha) \quad (2-1)$$

而在影像分割後，不確定的資訊量應會減少。假如其分閾值若為 g ，則 Gray-scale Image Information (GII) 定義為分割影像後關於像素值分類的不確定資訊量的減少，如公式(2-2)，類似資訊增益 (Information Gain) 的觀念。 $H(F_0)$ 、 $H(F_1)$ 、 $H(F)$ 分別代表前景像素值數量的機率密度函數的熵值、背景像素值數量的機率密度函數的熵值，及前景和背景混合的熵值。

$$\begin{aligned} \text{GII} &= \text{初始的不確性資訊量} - \text{剩餘的不確性資訊量} \\ &= H(X) - H(X | g) \\ &= H(X) - \alpha H(F_0) - (1-\alpha) H(F_1) \end{aligned} \quad (2-2)$$

因此一個好的分割結果，GII 的值應該愈大愈好，也就是愈趨近於

$H(X)$ 愈好，意味著剩餘的不確性資訊量愈少，分割出來的物件相似性愈高。

第三節 二維熵值分割 (Two-Dimensional Entropic Segmentation)

Prasanna K. Sahoo 於 1997 年提出一個新的二階分閥值法[10][17]，它是基於灰階統計直方圖分佈中前景熵值與背景熵值差異最小即為最佳的分閥值。

灰階統計統計圖大致上可以近似於密度函數 (Density Function)：

$$h(x) = \text{Prob}[f(m,n) = x] \quad (2-3)$$

$$\hat{h}(x) = \frac{g_x}{g_{total}} \quad (2-4)$$

在此， $f(m,n)$ 表示像素點在 (m,n) 這個位置上的灰階度， $h(x)$ 是像素值 x 的機率， g_x 是影像中像素灰階值 x 的數量，其中 x 的值由0到255。 g_{total} 為一張影像中所有像素的總數量，若一張影像的尺寸大小為 $M \times N$ ，則 g_{total} 等於 $M \times N$ ，而 $\hat{h}(x)$ 是用來估算 $h(x)$ 。

令 $p_i = \hat{h}(x) = n_i / MN$ ，為影像中灰階值的機率，此處的 n_i 為事件 $[f(m,n) = x]$ 中的元素個數。而分割前的熵值則定義為 H_T 。

就一整張影像而言，假設有兩個類別 (class) 的像素， b_0 記為黑色像素的類別 (也就是灰階度為 0)， b_1 記為白色像素的類別 (也就是灰階度為 255)，因此對每一類別的像素而言其分割前的熵值分別為 $H_{b_0}(t)$ 以及 $H_{b_1}(t)$ 。

$$H_T = -\sum_{i=0}^{255} p_i \ln p_i \quad (2-5)$$

$$H_{b_0}(t) = -\sum_{i=0}^t \frac{p_i}{p(b_0)} \ln \frac{p_i}{p(b_0)} \quad (2-6)$$

$$H_{b_1}(t) = -\sum_{i=t+1}^{255} \frac{p_i}{p(b_1)} \ln \frac{p_i}{p(b_1)} \quad (2-7)$$

where

$$p(b_0) = \sum_{i=0}^t p_i, p(b_1) = \sum_{i=t+1}^{255} p_i \quad (2-8)$$

Sahoo 認為最佳的臨界值 t^* 將發生在兩個熵值 $H_{b_0}(t)$ 以及 $H_{b_1}(t)$ 的差之平方之最小值時。也就是求最小的 $E(t)$ 中的 t 值，Sahoo 將之稱為 t^* 。

$$E(t) = [H_{b_0}(t) - H_{b_1}(t)]^2 \quad (2-9)$$

$$t^* = \arg \min E(t). \quad (2-10)$$

Sahoo 指出愈小的 $E(t)$ ，代表著 b_0 與 b_1 將愈具有同質性。而由於該方法是使用一維灰階統計直方圖的方式，因此又可以稱為一維交叉熵值法 (1-D Entropy Crossover Method)。但是 Sahoo 所提出的這一方法有一個缺點，也就是隱含了一個假設，即物件與背景兩類的分割前機率幾乎是相等的。但是當兩機率之差為遞增時，則分割後的品質會有下降的情況發生。

因此，他延伸熵值交叉法到二維的灰階統計直方圖。這樣不僅只依靠灰階統計直方圖來估算，也利用了像素間的空間資訊。第二個特徵值是使用局部平均灰階度，採用 3×3 的遮罩逐點估算其平均值，再結合影像灰階值形成二維灰階統計直方圖。如圖 2-4 所示， x 軸的分割點為利用基於灰階值的交叉熵值法所估算出來，而 y 軸則是基於局部平均灰階值估算出來，二點即能分割出四個象限。

這樣分割的形式，不在對角線的區域主要發生在第二及第四象限，而接近對角線的值都位於第一及第三象限。這意味著影像邊界值主要發生在第二及第四象限。傳統上，這些邊界值不是沒有被分類就是被忽略。然而，這些邊界值如果沒有被利用，有很多關於物件的重要資訊將會被摒棄。而置於第一象限的值歸類為前景，第三象限為背景。

因為這樣的缺點，他提出了分閥線（Threshold Line）的觀念，能夠產生最佳識別函式將前景和背景做切割。根據這個分閥線，如圖 2-5 所示，不在對角線的那些邊界值即能包含在分割區域中，分類錯誤的像素值數目即能大大地減少。

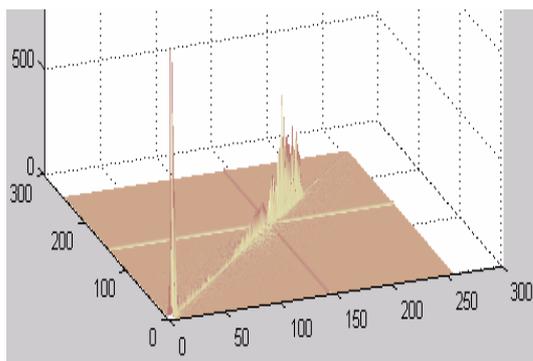


圖 2-4 分割出四個象限

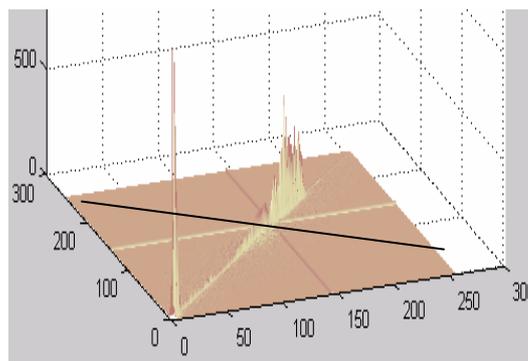


圖 2-5 分閥線分割出二群

第四節 快速自動多重分閥值法 (Fast Automatic Multilevel Thresholding Method)

L. Cao, Z.K. Shi 和 E.K.W. Cheng[15]於 2002 年提出了一個基於最大熵值定理（Maximum Entropy Theorem）的自動多重分閥值法，這個方法比一些傳統熵值的法的優點為：非常地簡易、更好的可靠度，及計算快速。

最大熵值原理：資訊理論 (Information Theory) 中的最大熵值定理是藉由分閥值法來分割影像的一個有用的方法。假設 X 是一個離散隨機變數，例如一個有限的範圍 $R=\{x_1, x_2, \dots\}$ ，令 $p_i = P\{X = x_i\}$ ， $i = 1, 2, \dots, n$ ，則 X 的夏農熵值 (Shannon Entropy) 為公式(2-11)所示。而 $p_i > 0$ ， $\sum_{i=1}^n p_i = 1$ ，對數的基底為任何的實數，假如 $p_i = 0$ ，則 $\sum_{i=1}^n p_i \log_a \frac{1}{p_i}$ 這一項是否等於零為不確定的，在這裡定義它為零。

$$H(X) \cong \sum_{i=1}^n p_i \log_a \frac{1}{p_i} = -\sum_{i=1}^n p_i \log_a p_i \quad (2-11)$$

在公式(2-11)中， $H(X)$ 定義了隨機變數 X 的熵值，意味著包含了 X 不確定性估算的資訊總量。 $H(X)$ 也意味著 p_1, p_2, \dots, p_n 的機密分佈函數 (Function of Probability Distribution)，定義如公式(2-12)所示。

$$H(X) = H(p_1, p_2, \dots, p_n) \cong -\sum_{i=1}^n p_i \log p_i \quad (2-12)$$

從公式(2-12)中，因為 $\sum_{i=1}^n p_i = 1$ ，我們可以利用拉格朗日乘數法 (Lagrange Multiplier Method) 得到的公式(2-13)。

$$H(p_1, p_2, \dots, p_n) = -\sum_{i=1}^n p_i \log p_i + u \left(\sum_{i=1}^n p_i - 1 \right) \quad (2-13)$$

令 $H(A) = H(p_1, p_2, \dots, p_n)$ ， $p(x_i) = p_i$ ， $i=1, 2, \dots, n$ ，在這裡 x_i 代表影像中像素的灰階值，而 $p(x_i)$ 意味著 x_i 這個灰階值的密度。公式(2-13)可變成公式(2-14)。要使公式(2-14)最大化，可從公式(2-15)獲得。

$$H(A) = -\sum_{i=1}^n p(x_i) \log p(x_i) + u \left(\sum_{i=1}^n p(x_i) - 1 \right) \quad (2-14)$$

$$\frac{\partial H(A)}{\partial p(x_i)} = -\log p(x_i) - 1 + u = 0 \quad (2-15)$$

我們就能夠從公式(2-15)獲得解答 $p(x_i) = e^{n-1} = c, i=1,2,\dots,n$ 。假如 $p_1 = p_2 = \dots = p_n = 1/n$ ，將使得 $H(A)$ 為最大。因此作者認為當 $p_1 = p_2 = \dots = p_n, \sum_{i=1}^n p_i = 1$ ，熵值將會達到最大。

在影像處理中，把影像視為資訊來源 A ，而把直方圖分佈視為機率 p_i 。因此作者提出了一個新的自動估算影像分閥值法。

定義：令 A 為影像像素值的集合， $p_i (i=0,1,\dots,255)$ 為第 i 個灰階值 ($p_i =$ 第 i 個灰階像素點數除以影像像素總點數) 的機率， N 為 A 分割後的群數， $A_j (j=0,1,\dots,N-1)$ 代表每個切割， $p(A_j)$ 則代表切割後 A_j 的機率。新的公式則如公式 (2-16) 所示。

$$H(A,N) = \sum_{j=0}^{N-1} \left\| p(A_j) - \frac{1}{N} \right\| \quad (2-16)$$

從公式(2-16)中，我們可以得知假如 $p(A_j) = 1/N$ 時， $H(A,N)$ 將會達到最小 0；而公式 (2-12) 中的熵值 $H(A)$ 也將會達到最大。因此作者認為將公式(2-16)達到最小，即能得到最佳分閥值。公式(2-12)和公式(2-16)在影像處理中是相等的。然而，這二個公式在計算的時間上有著相當大的差異。因為作者提出的方法，其目標函式 (Objection Function) 變成比較直覺與簡易，所以其處理時間會在 $N(N-1)$ 的範圍內，改善了傳統最大熵值分閥值法耗時的缺點。

第五節 決策樹

在資料探勘 (Data Mining) 技術中，決策樹演算法是屬於資料分類 (Classification) 的方法之一 [3,4,13,14,19]。而分類是屬於監督式的學習，其目的在於利用訓練資料 (training data) 的各種屬性進行訓練，找出隱

藏在這些資料中的分類規則，也就是希望找到該類別（Class）的特徵描述。

本研究所採用的決策樹演算法為由 Quinlan 於 1993 年所提出的 C4.5 學習演算法，此演算法為改良其於 1986 年所提出的 ID3 學習演算法。這二個演算法最大的不同在於，ID3 的版本所處理的資料型態必須是離散型的、或是文字型的（Categorical），若是連續型的資料型態都必須轉換為離散型態的資料才可以繼續進行；而 C4.5 是 ID3 的改良與延伸，它可以處理連續性數字型態的資料。

決策樹中的節點(Node)可以分為兩種：

1. 決策節點(Decision Node)：指出某一項屬性的測試。
2. 葉節點(Leaf Node)：指出結果屬於哪一類別。

以幾何的觀點來看，決策樹意味著資料空間的切割。而超矩形（Hyper-Rectangle）代表著從根節點到葉節點這一連續的測試（分割）。我們藉由下圖來說明決策樹與其資料空間切割的對應關係。下圖 2-6 為節錄自 IBM 研究報告 CLTree 論文[7][8]中的圖示。以該圖為例，在圖(A)中包含了四塊超矩形區域，而所產生的決策樹在圖(B)，其中 X 與 Y 代表測試節點，而 或 表示葉節點，分別代表兩種不同的類別。

在圖 2-6 中，(A)與(B)分別為空間切割與決策樹的對應情況。舉例來說，圖(B)決策樹中的一個路徑 $X > 3.5, Y > 3.5 \rightarrow$ ，對應至圖(A)則為右上半邊的部分，且其中的資料類別為 。

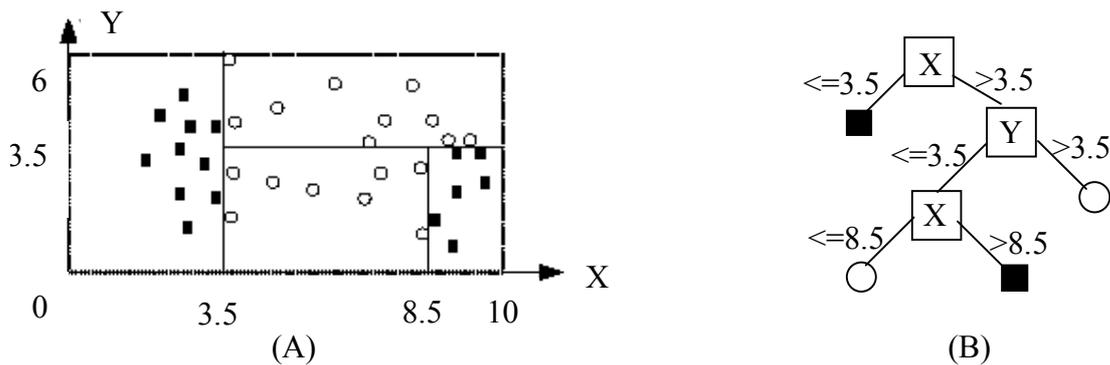


圖 2-6 資料空間的切割與其相對應的決策樹。

整個決策樹的運作包括兩大過程：

1. 決策樹的建構
2. 決策樹的修剪

以下我們將針對這二個過程作進一步的說明。

一、 決策樹的建構

建立決策樹的演算法一般是採用各個擊破 (divide-and-conquer) 的策略並遞迴地切割資料空間來建構決策樹。每個建樹的步驟是找出最佳的切割點並且將資料空間分成二部份，目的是為了獲得更為純粹的區域，也就是希望該區域中的資料類別越一致越好，並且使得這二部份之間的差異性最大。就如同上圖 2-6 所示[7]。

資訊增益 (Information Gain, IG) 為一般最常用的選擇最佳切割點準則，它是源自於資訊理論 (Information Theory)。資訊理論的概念認為訊息所具有的資訊是可以被量化的。假設訊息本身的機率為 P_i ，則該訊息於每位元中所傳達的資訊量可由 $-P_i \times \log_2 P_i$ 計算出來。而 $-P_i \times \log_2 P_i$ 也就是該訊息本身所具有的熵值 (Entropy)。

若在資料集 D 中有 q 個類別，分別為 C_1, \dots, C_q ；假設 x 為可能的

切割，且將資料集 D 切割為 m 個子資料集，分別為 D_1, \dots, D_m 。因為我們僅僅執行二元切割，所以 $m=2$ 。

假設 $info(D)$ 為在可能的切割之前資料集 D 所傳達的資訊量， $info_x(D)$ 為在某單一屬性 A_i 上 x 將 D 切割之後而資料集 D 所傳達的資訊量，而 $gain(X)$ 為 $info(D)$ 減去 $info_x(D)$ 的值，即為在整個切割前後的資訊量變化。也就是以分類前的資訊量與分類後的資訊量之差作為分類的條件。因此，在切割前的資訊量 $info(D)$ 不變的情況下，若切割後的資訊量 $info_x(D)$ 越少，則表示在整個切割前後的過程中所傳達的資訊量 $gain(X)$ 越多，所切割出來的資料空間類別更為純粹。

令 P_j 為由資料集 D 之中選出一筆資料，而該筆資料屬於某一類別 C_j 的機率。 P_j 的求法即如公式(2-17)所示，其中 $|D|$ 為資料集 D 之中的所有資料筆數， $freq(C_j, D)$ 為在資料集 D 之中屬於類別 C_j 的資料筆數。

$$P_j = \frac{freq(C_j, D)}{|D|} \quad (2-17)$$

而 P_j 本身所傳達的資訊量(熵)，則可由 $-P_j \times \log_2 P_j$ 計算出來。因此，在切割之前資料集 D 所傳達的熵 $info(D)$ ，為資料集 D 之中所有 P_j ($j=1$ 到 q) 的熵加總，即為公式(2-18)所示。

$$info(D) = - \sum_{j=1}^q \frac{freq(C_j, D)}{|D|} \times \log_2 \left(\frac{freq(C_j, D)}{|D|} \right) \quad (2-18)$$

假設 x 為可能的切割，根據 x 將資料集 D 切割為 m 個子資料集 D_i ，則其分類後所佔的資訊量為各個子資料集的資訊量乘上各個子資料集所佔的比例的總和。切割之後資料集 D 所傳達的熵 $info_x(D)$ 為公式(2-19)所示，參數中 $|D_i|$ 為子資料集 D_i 之中的資料個數。

$$info_x(D) = \sum_{i=1}^m \frac{|D_i|}{|D|} \times info_x(D_i) \quad (2-19)$$

最後，在整個切割前後的過程中的資訊量(熵)變化將以公式(2-20)來獲得。很明顯地，我們應該使得 $gain(X)$ 最大化，則表示所切割出來的資料空間類別最為純粹。

$$gain(X) = info(D) - info_x(D) \quad (2-20)$$

資訊增益估算的步驟如圖 2-7 所示。其在整個維度中，估算出每個可能的切割點，並在這些切割點中找到能使資訊增益為最大的，即為最佳切割點。

```

for each attribute  $A_i \in \{A_1, A_2, \dots, A_d\}$  do      /*  $A_1, A_2, \dots, A_d$  are attributes of  $D$  */
  for each value  $x$  of  $A_i$  in  $D$  do /*each value is considered as a possible cut */
    compute the information gain at  $x$ 
  end
end
select the cut that gives the best information gain to partition the space

```

圖 2-7 資訊增益的估算[7]。

建立決策樹的演算法以遞迴的方式在資料空間中找出最佳切割點以進行資料空間的切割，目的在於找出更為純粹的區域。當資料空間中的資料類別為一致時，或當該資料空間中無法提供切割點時，則停止繼續建立決策樹。其演算法如圖 2-8 所示。

Algorithm: Generate_decision_tree.

Input: samples, represented by discrete-valued attributes; the set of candidate attributes, attribute-list.

Output: A decision tree.

Method:

- (1) Create a node N;
 - (2) If samples are all of the same class C then
 - (3) return N as a leaf node labeled with the class C;
 - (4) If attribute-list is empty then
 - (5) return N as a leaf node labeled with the common class in samples
 - (6) Select test-attribute, the attribute among attribute-list with the highest information gain;
 - (7) Label node N with test-attribute;
 - (8) For each known value a_i of test-attribute //partition the samples;
 - (9) Grow a branch from node N for the condition test-attribute = a_i ;
 - (10) let s_i be the set of samples in samples for which test-attribute = a_i ; //a partition
 - (11) if s_i is empty then
 - (12) attach a leaf labeled with the most common class in samples;
 - else
 - (13) attach the node returned by Generate_decision_tree(s_i , attribute-list-test-attribute);
-

圖 2-8 ID3 版本的決策樹演算法。

二、決策樹的修剪

由於建構決策樹是以遞迴分割的方法，它會分割資料空間直到每個子資料集的資料都屬於同一種類別，或者是子資料集已經沒有任何資料可以改進。這樣的結果會使得決策樹有過度訓練（overfitting）的情況產生，以致於所產生的決策樹太過於複雜且沒有必要，因此建構決策樹後還需要做適當的修剪。一般可有兩種方法：

1. 停止(Stopping): 在建樹的時候，若判斷一些判斷法使得此區域沒有必要再做進一步分割，那麼就停止建立此節點的子樹。
2. 修剪(Pruning): 在整棵樹建立好之後，才修剪樹中的某些子樹。

最後，由整個決策樹的過程執行完畢後，則可由決策樹中產生出分類規則。經由測試資料驗證分類規則的正確性後，往後，使用者可依據這些分類規則以作為新資料的預測。

三、 小結

決策樹具有容易實作、結構簡單、分類規則的產生，以及雜訊資料的處理等優點。但是它也有缺點存在，例如決策樹為了逼近該真實的類別邊界，求取最大的資訊增益，而可能會產生許多的節點。而這樣的情況有可能使訓練階段產生過多的規則而造成訓練過度。為了避免這種情況，還必須對所得的決策樹作進一步的修剪。

第六節 CLTree 演算法

IBM 研究中心的 Buing Liu, Yiyuan Xia，於西元 2000 年提出了一個新的叢聚技術，以決策樹建構法為基礎，他們將之命名 CLTree(CLustering Based on Decision Trees)。而為了與傳統決策樹有所區別，他們將 CLTree 所產生的樹稱作為叢聚樹 (Cluster trees) [7][8]。

這篇論文中所討論的範圍限制在數值空間中，每一筆資料在數值空間中所呈現的都是一個點。作者主要的想法是利用決策樹將資料空間切割為資料密集叢聚的區域 (cluster region) 以及稀疏的區域 (sparse region)。可是決策樹並不能直接應用於叢聚分析上，因為叢聚分析是屬於非監督式的學習過程。因此作者提出解決這個問題的方法，做法是將原來資料記為 Y 類別 (Y class)，然後假設有另外一種的資料點，稱為不存在的點 (non-existing points)，並將之記為 N 類別 (N class)，這些虛擬

的點數均勻地導入散佈在資料空間中，之後用修改過的決策樹演算法來達到分群的目的。

這個做法理由是，假如在資料中有許多群組，則這些資料點不會一致散佈在整個空間。藉著導入均勻散佈的 N 點，即能將群與群之間孤立出來，因為在資料叢聚的區域中 Y 點一定會比 N 點來的多。而決策樹就是適用於解決這樣的問題，如圖 2-9(B)。

但因為實際地加入 N 點將會增加整體的資料量以及運算時間。因此作者希望不實際加入這些均勻分布的 N 點至資料空間中，而是需要 N 點時才去計算。也就是圖 2-9(C)的情況。

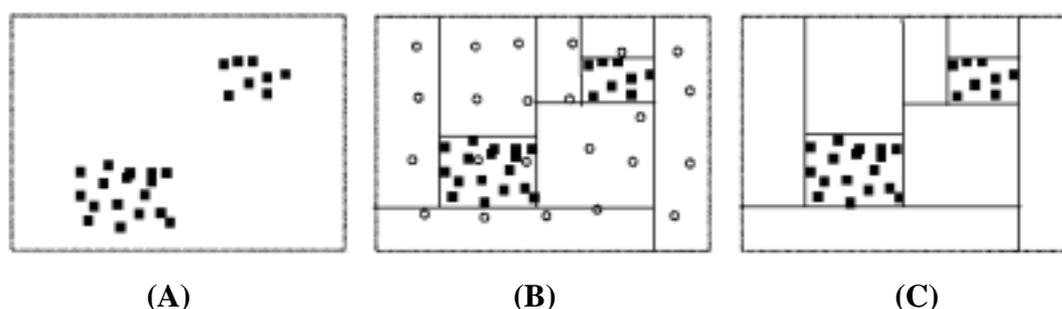


圖 2-9(A)資料在資料空間中的分布狀況。(B)加入均勻分布的 N 點至資料空間中以進行 CLTree 叢聚演算法。(C)不實際的加入 N 點，而是需要的時候才加以計算

由於不實際加入 N 點至資料空間中，以及傳統決策樹之的判斷標準無法符合需求，因此在建構叢聚樹時，作者提出一些適當的修改。

CLTree 演算法有兩個步驟：叢聚樹的建構，以及叢聚樹的修剪。

一、叢聚樹的建構

在叢聚樹建構的步驟中，不須預作任何假設以及給定任何參數，利用修改過的新的純粹函數來建構叢聚樹，以獲取資料分佈的描述。

(一) 在不同的測試節點 (test node) 加入不同數量的 N 點

作者認為若一開始就固定 N 點的數量，則分割到後來可能會有在某個區域中 N 點過少而無法將群與群之間作進一步的分割。如圖 2-10 所示，這個資料空間包含了 32 個資料點，經過二次切割後，所有的資料點皆在 region-1 裡。假如我們在 region-1 之中不增加 N 點數量，則此時的 N 點只剩下 $32/2^2=8$ 個。

因為資料空間的維度只有二維，所以這種情形還可以接受。假如是非常高的維度空間，在經過多次切割後，則 N 點的數量將會以指數方式激烈地下降。

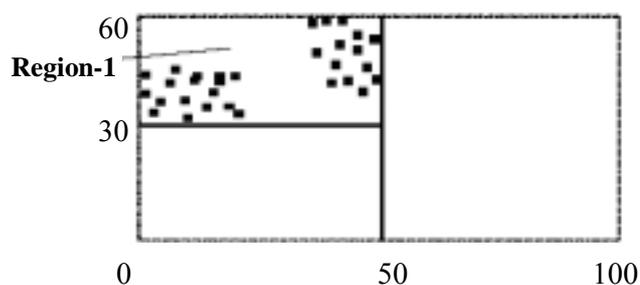


圖 2-10 在不同測試節點加入不同數量的 N 點的考量。

因此，在叢聚樹建立之時，作者考慮在不同的測試節點加入不同數量的 N 點，而依據的法則如下：

If the number of N points inherited from the parent node of E is less than the number of Y points in E **then**
the number of N points for E is increased to the number of Y points in E
else the number of inherited N points from its parent is used for E

圖 2-11 加入不同數量的 N 點的法則。

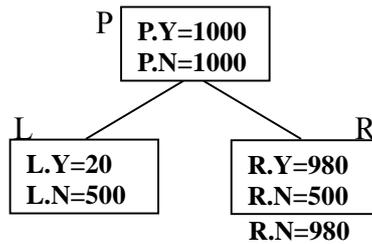


圖 2-12 加入不同數量的 N 點。

若 current node 為 N node (表示在此節點中 N 點的數量會比 Y 點數量還多) 的話, 則 N points 的數量不必減少, 因為若減少 N 點數量會造成局外者 (outliers) Y 點以形成 Y node, 如此一來叢聚區域 (cluster regions) 與非叢聚區域 (non-cluster regions) 就可能無法被區分開來了。

我們以圖 2-12 來說明。根節點 P 有 1000 個 Y 點, 因此其 N 點為 1000 點。假設經過分割後, 有二個子節點 L 和 R, 可以看到子節點 R 的 N 資料點少於 Y 資料點, 因此我們必須把 N 點的數量增加到與 Y 點相同的 980 點。而子節點 L 的 N 點大於 Y 點, 則不須改變。

(二) 計算資訊增益時, N 點的數量如何取得

由於不實際地加入 N 點到原有的資料空間中, 作者提出機動地計算 N 點的數量。以圖 2-13 為例, 資料空間中共有 25 個資料點 (Y), 以及 25 個 N 點。假設 $P=4$ 時為可能的切割點, 因此在 P 點左邊的區域中, N 點共有 $25 \times (4/10) = 10$ 個, Y 點共有 3 點; 而在 P 點右邊的區域中 N 點共有 $25 - 10 = 15$ 個, Y 點共有 $25 - 3 = 22$ 個。

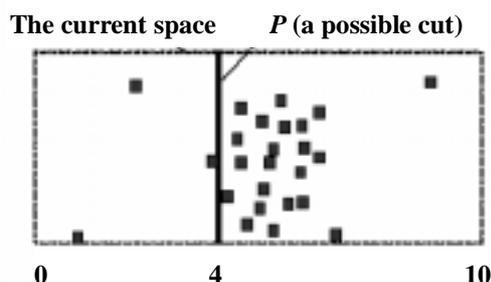


圖 2-13 N 點數量的計算。

(三) 決定最佳切割點

傳統決策樹用資訊增益來選擇最佳的切割點，但是對於叢聚而言是不足夠的，因為資訊增益所找出的最佳切割點往往會把一群資料點切割為密集集中與鬆散兩個區域。如圖 2-14 所示，在圖 2-14(A) 顯示出資料空間包含了二群，透過資訊增益的計算，分別在二個維度中求出 d_1_cut 和 d_2_cut 。很清楚地可看到，這二個切割點並不是合適的，因為它們把同一群的也切成二群。所以，在理想的情況下，以圖 2-14(B) 所示。我們應該是切在 d_2_cut2 或 d_2_cut3 而不是切在 d_1_cut 上。然而使用資訊增益的標準，並不能求出 d_2_cut2 或 d_2_cut3 。

因此，作者為了能夠找出更逼近叢聚的邊緣，於是考慮在鬆散的區域中更進一步的找到更適合的切割點，他們把這樣的判斷標準命名為 ”進一步的增益標準(the Lookahead Gain Criterion)”。

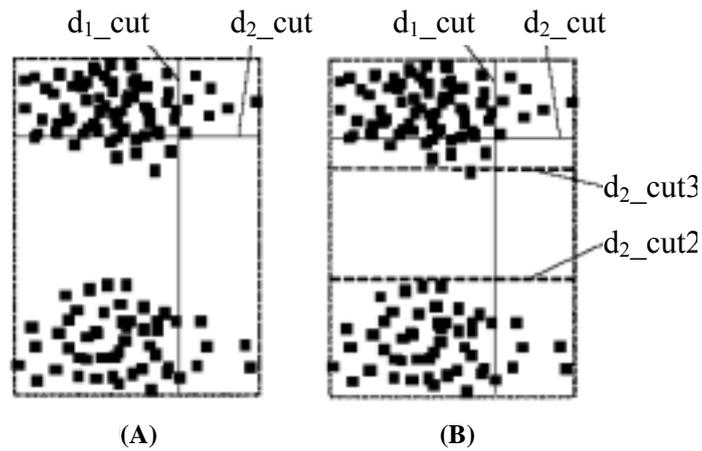


圖 2-14 傳統資訊增益的問題

相對密度的定義：

$$\text{一個區域的相對密度} = r.Y / r.N \quad (2-21)$$

$r.Y$ ：在區域 r 之中 Y 點的數目

$r.N$ ：在區域 r 之中 N 點的數目

而決定最佳的切割點的步驟可以分為三個步驟：

1. 對於每個維度 i ，都個別以資訊增益來找到初始的切割點 d_{i_cut1} 。
2. 任一維度 i 被 d_{i_cut1} 切割為兩個區域，相對密度較高者稱為 H_i ，相對密度較低者稱為 L_i ，然後每一個維度都在進一步地在 L_i 區域尋找最佳的切割點 d_{i_cut2} ，或甚至至多到 d_{i_cut3} 。而找出 d_{i_cut2} 仍是依據在 L_i 中資訊增益最大者。並記錄各維度最後的 L_i 的相對密度。
3. 在所有維度中，比較各 L_i 的相對密度 (Relative Density) 若為最小者，則該維的 d_{i_cut2} (或 d_{i_cut3}) 即為最佳的切割點。

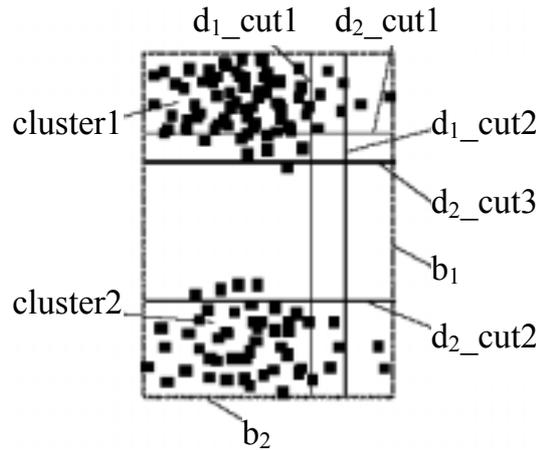


圖 2-15 決定最佳切割點示意圖

經由進一步的增益標準的計算，所找出的最佳切割點如圖 2-15 中的 d_{2_cut3} 。圖 2-16 為決定最佳切割點的演算法。

```

Algorithm evaluateCut(D)
For each attribute  $A_i \in \{A_1, A_2, \dots, A_d\}$  do
   $d_{i\_cut1}$  = the value (cut) of  $A_i$  that gives the best gain on dimension  $i$ ;
   $d_{i\_cut2}$  = the value (cut) of  $A_i$  that gives the best gain in the  $L_i$  region produced by
     $d_{i\_cut1}$ ;
  if the relative density between  $d_{i\_cut1}$  and  $d_{i\_cut2}$  is higher than that between
     $d_{i\_cut2}$  and  $b_i$ ;
     $r\_density_i = y_i/n_i$ , where  $y_i$  and  $n_i$  are the numbers of  $Y$  points and  $N$  points
      between  $d_{i\_cut2}$  and  $b_i$ ;
  else  $d_{i\_cut3}$  = the value (cut) that gives the best gain in the  $L_i$  region produced by
     $d_{i\_cut2}$ ; /*  $L_i$  here is the region between  $d_{i\_cut1}$  and  $d_{i\_cut2}$  */
     $r\_density_i = y_i/n_i$ ; where  $y_i$  and  $n_i$  are the numbers of  $Y$  points and  $N$  points in the
      region between  $d_{i\_cut1}$  and  $d_{i\_cut3}$  or  $d_{i\_cut2}$  and  $d_{i\_cut3}$  that has a
      lower proportion of  $Y$  points (or a lower relative density).
  end
end
 $bestCut = d_{i\_cut3}$  (or  $d_{i\_cut2}$  if there is no  $d_{i\_cut3}$ ) of dimension  $i$  whose  $r\_density_i$  is
the minimal among the  $d$  dimensions.

```

圖 2-16 決定最佳切割點演算法

二、叢聚樹的修剪

建立叢聚樹時所採用的遞迴切割方法會一直地切割資料空間直到每一個切割區域只包含單一類別，或直到此切割區域無法再做進一步地分割為止。這樣的結果通常會切割出沒有必要的區域，而導致一棵很複雜的樹。因此要簡化所得到的樹，一般可有兩種方法：

1. 停止 (stopping) : 在建樹的時候，若判斷此區域沒有必要再做進一步分割，那麼就停止建立此節點的子樹。但是作者認為這種判斷停止生長的法則因不同的應用領域而有所不同，難以客觀化。
2. 修剪 (pruning) : 在整棵樹建立好之後，才修剪樹中的某些子樹。作者所採用的是修剪的方法。

就某種程度來說，叢聚 (Clustering) 結果的好壞或是否具有意義，是依據使用者主觀的判斷以及在不同的應用領域中，因此分類 (Classification) 中的修剪方法 (pruning method) 就不能拿來在叢聚的問題中應用。

以圖 2-17 為例，原來的資料空間利用 CLTree 演算法切割為 14 個區域。究竟我們要把它看作是一群 (整個資料空間) 呢？還是兩群呢？如果我們把它看作為兩群的話，那究竟是 C1 與 C2 是兩大群呢？還是 S1 與 S2 是兩大群呢？因此，這答案是依據特定的應用而定，是個很主觀的問題。

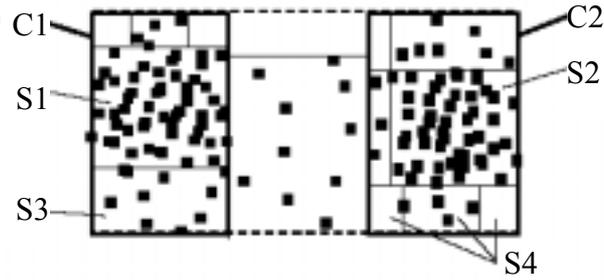


圖 2-17 在此資料空間中應該有多少個叢聚呢？

作者提出兩種互動式的方法來允許使用者瀏覽叢聚樹，以便找到具有意義的資料群 (Clusters)。

1. 瀏覽 (browsing) : 使用者可以簡單地瀏覽樹本身，以找出具有意義的叢聚。這並不是件很難的工作，因為最主要的叢聚可以在樹的上層辨認出來。
2. 使用者導向的修剪方式 (user-oriented pruning) : 利用兩種使用者自訂的參數來修剪。

(1) min_y :

此參數是指在一個區域中所應包含的 Y 點之最少數量， min_y 通常表示為佔資料總數 $|D|$ 的百分比，例如吾人可以設定 $min_y=1\%$ 。

在 CLTree 的節點之中，若該節點的 Y 點數量小於 $min_y \times |D|$ ，那麼此節點以下所生長出來的子樹可以被砍掉。

(2) min_rd :

若 N 區域 (節點) 稱作 E ; Y 區域 (節點) 稱作 F 。 min_rd 所考慮的是 E 是否應該與鄰接的 F 合併在一起已成為更大的叢聚區域 (cluster region)。 $E.Y$ 表示在 E 區域中 Y 點數量， $E.N$ 表示在 E 區域中 N 點數量。 如果 $(E.Y / E.N)$ 大於 min_rd ，則合併

E 與 F 兩個區域。

叢聚樹的修剪演算法是以遞迴的方式先向下地找到樹的葉節點，再一層一層的倒回決定是否由獨自一個 Y 節點自行形成一個叢聚或是與相鄰的節點合併成為一個叢聚，圖 2-18 即為其演算法。

```
Algorithm evaluatePrune(Node, min_y, min_rd)
  /*Node is the node being analyzed*/
  if Node is a leaf then Node.Stop = TRUE
  else LeftChild = Node.left;  RightChild = Node.right;
    if LeftChild.Y < min_y * |D| then LeftChild.Stop = TRUE
    else evaluatePrune(LeftChild, min_y, min_rd);
    end
    if RightChild.Y < min_y * |D| then RightChild.Stop = TRUE
    else evaluatePrune(RightChild, min_y, min_rd);
    end
    if LeftChild.Stop = TRUE then    /*We assume that the relative density*/
      if RightChild.Stop = TRUE then /* of LeftChild is always higher */
        /* than that of RightChild*/
          if RightChild.Y / RightChild.N > min_rd then
            Node.Stop = TRUE /*We can prune from Node either*/
            /* because we can join or because both*/
            /* children are N nodes.*/
          elseif LeftChild is an N node then Node.Stop = TRUE
          else Node.Stop = FALSE
          end
        else Node.Stop = FALSE
        end
      else Node.Stop = FALSE
      end
    end
  end
```

圖 2-18 叢聚樹的修剪演算法

第三章 植基於決策樹建構法之二維多重分閥值法

第一節 植基於決策樹建構法之二維多重分閥值法

就以影像分割來說，每張影像灰階值的分佈都不同，我們也無法事先得知應分割成幾群，結果才是最適當的。

在本論文中，我們提出一個新方法，以決策樹建構法為基礎，並加入叢聚樹（CLTree）的觀念加以修改使成為可以解決影像分割中多重分閥值的問題。在處理上是利用資訊增益在影像二維灰階統計直方圖中遞迴地求取最佳切割點，直到每一區域資料類別一致為止，並建構出叢聚樹。而判斷區域內資料類別是否一致的動作，是以我們提出的停止建樹條件（詳見本章第二節之步驟四）為準則。之後以我們提出的修剪條件（詳見本章第三節）對叢聚樹內各葉節點進行修剪，來達到最適當的分群結果。因此在整個演算法中主要分為兩階段，為叢聚樹的建構和叢聚樹的修剪。

圖 3-1 描繪出演算法架構與流程，往後章節中將針對各步驟做詳細的介紹。

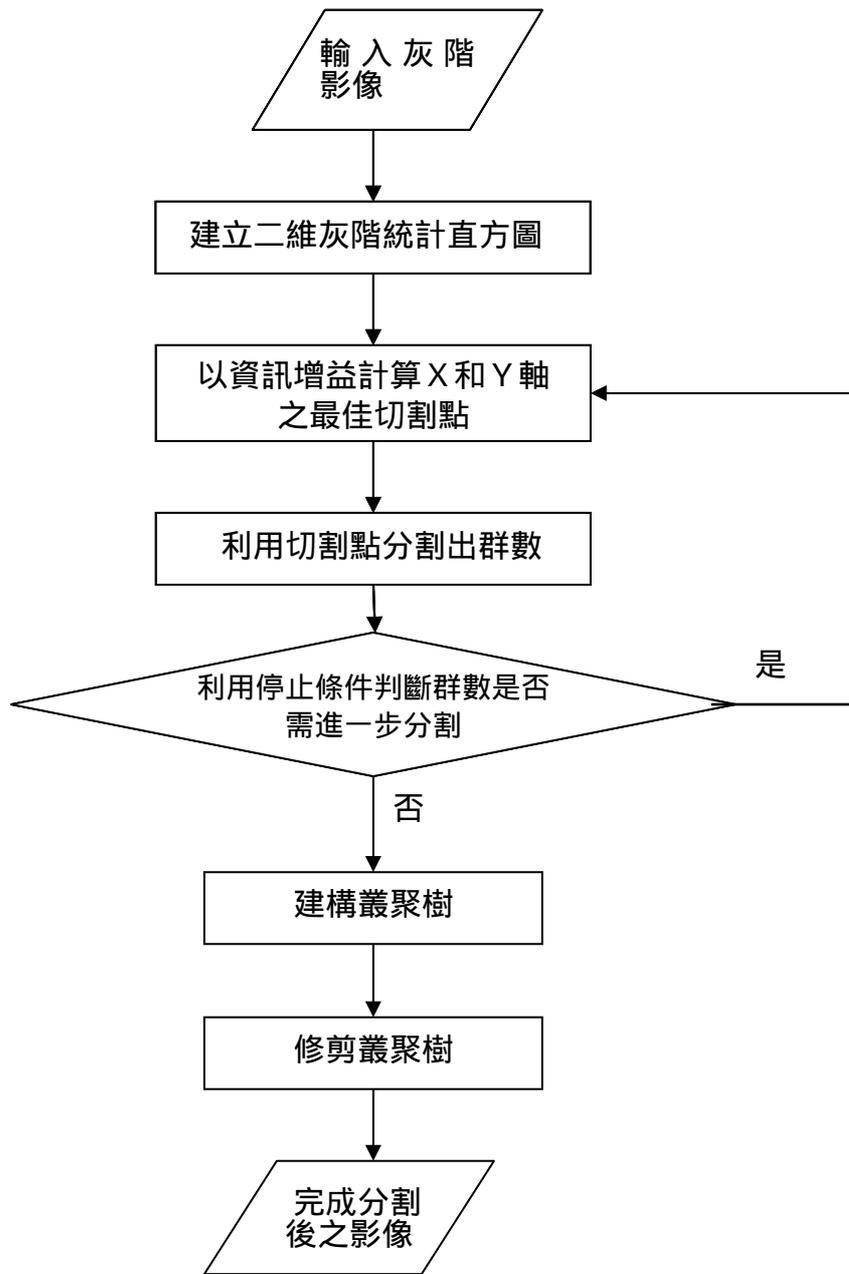


圖 3-1 演算法架構與流程

變數名稱	描述
Y	區域內原本資料點數量
N	虛擬加入的資料量數量
min_y	區域內所應包含總點數的百分比
min_rd	區域內所規定的相對密度條件
$ D $	資料空間原本資料點總數
P	相鄰區域合併的參數
Y_k	$k=1\sim 4$ ，代表分割後四群的 Y 類別數量
n_k	$k=1\sim 4$ ，代表分割後四群的 N 類別數量
x_l	雜訊值的集合
c_i	$i=1\sim j$ ，代表第 i 個叢聚的群心
L_i	叢聚樹的葉節點，i 的範圍為 $1\sim n$
A_{ij}	為 L_i 的相鄰節點，j 最多為 2

表 3-1 演算法變數名稱之說明

第二節 叢聚樹的建立

步驟一：產生二維灰階統計直方圖

首先輸入欲分割之灰階影像，為了得到整張影像的灰階次數分佈，因此求取其灰階統計直方圖統計量，以做為接下來二維灰階統計直方圖之第一個特徵值（即 X 軸）。並以中值濾波器（Median Filter）來消除影像中的雜訊，其為採用遮罩內的中值來取代原來的像素灰階度。因此新的灰階度稱為局部中值灰階度（Local Median Gray Level），以做為二維灰階統計直方圖之第二個特徵值（即 Y 軸）。

中值濾波器是採用遮罩（mask）逐點對影像做處理的方法，主要是用來降低雜訊。我們的方法是採用 3×3 的遮罩。首先把遮罩內所含蓋的像素灰階值由小而大排列，中值是指排序在中間的那一個值，此值即為濾波器的輸出，也就是局部中值灰階值。因為灰階值的雜訊值與其鄰近像素的灰階值有很大的差異，因此經排序後取中值的結果是強迫將此雜訊點變成與其鄰近的某些像素的灰階值一樣，達到去除雜訊的效果。

我們以麗娜這張像素大小為 256×256 的灰階影像來做說明，圖 3-2(a) 為麗娜的原圖，圖 3-2(b) 為其灰階統計直方圖；圖 3-2(c) 為經過中值濾波器的處理，已去除雜訊的灰階影像，圖 3-2(d) 為經過去雜訊處理後之影像灰階統計直方圖。

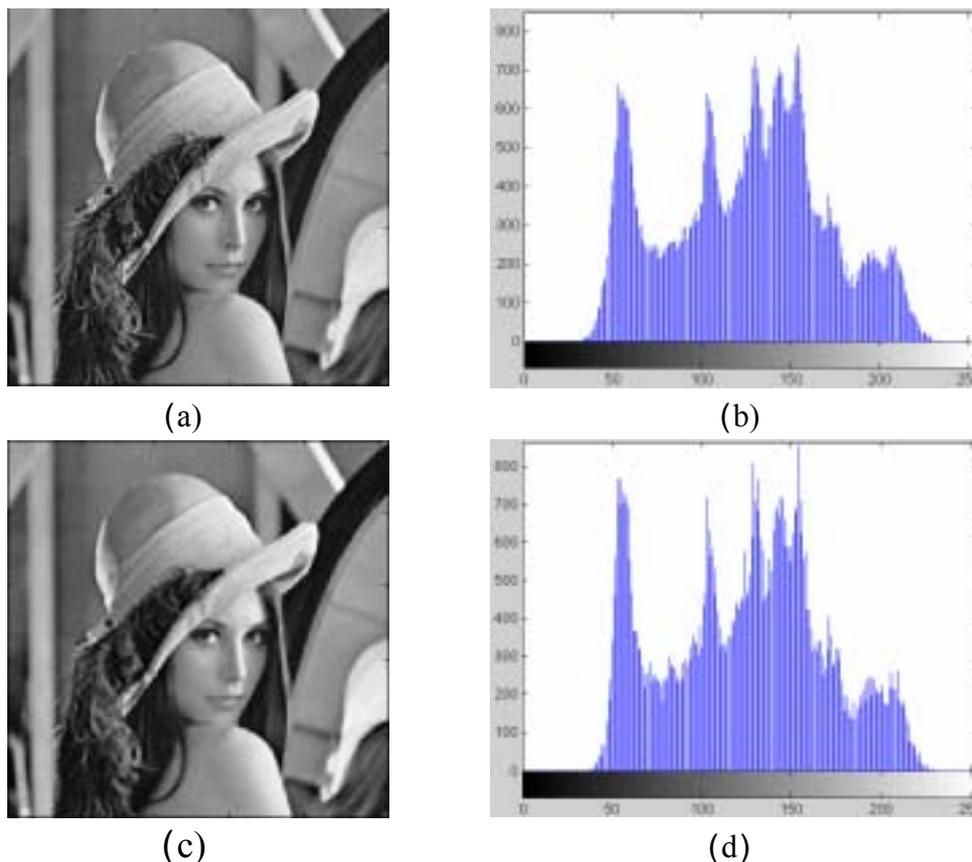


圖 3-2 (a) 麗娜原圖 (b) 麗娜的灰階統計量直方圖
(c) 去雜訊後之麗娜影像 (d) 去雜訊後之灰階統計量直方圖

因為在利用一維灰階統計量來處理影像分割的問題中，只有考慮到影像中灰階度的分佈，並沒有考慮到影像中像素與像素之間的空間關係。因此只基於一維灰階統計直方圖，當訊號雜訊比（Signal-to-Noise Ratio, SNR）減少時，則分割出來的影像將會大大的降低其效果。基於這個理由，本論文採用二維灰階統計直方圖的分割方法。

二維灰階統計直方圖是由原來影像的灰階值與其鄰近像素的局部中值灰階度這個二個特徵的聯合機率密度函數估計來的，一般通常以影像灰階度的相關矩陣（gray-level co-occurrence matrix）來表示，如圖 3-3 所示，灰階度為 X 軸，灰階程度是由左到右遞增(0 255)，而局部中值

灰階度為 Y 軸，灰階程度是由下至上遞增(0 255)。

在二維灰階統計直方圖中，靠近對角線的像素值意味著那些點不管置於原圖的一維灰階統計直方圖還是置於局部中值灰階度的一維灰階統計直方圖，其灰階統計量都幾乎一樣。因此這些點可能是影像中的物件或是背景，換句話說，離對角線距離愈遠的像素值意味著那些點應該是置於影像的邊界值或是雜訊值。

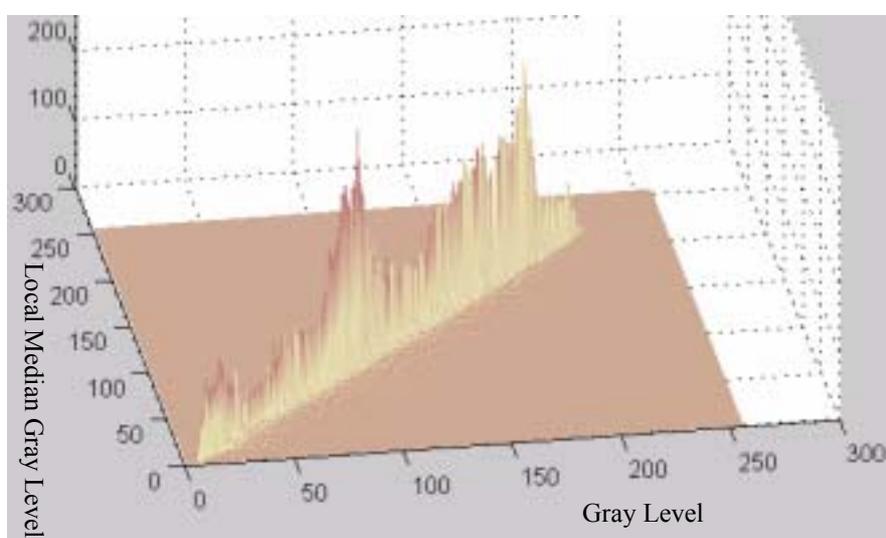


圖 3-3 二維灰階統計直方圖

步驟二：加入虛擬的 N 類別

假如一張像素大小為 256×256 之灰階影像，其像素點或稱為 Y 類別的總數即有 65536 個像素點，因此 N 類別的數量也應和 Y 類別數量一致，並且均勻地散佈在二維灰階統計直方圖上，才能將群與群之間區隔出來。

由於我們分割影像的準則是利用決策樹建構法中的資訊增益，因此在計算資訊增益時，必須至少要有兩種類別的資料才能運算。所以我們採用 IBM 研究中心所提出的基於決策樹之叢聚法的做法，把資料空間的

資料記為 Y 類別，並均勻地加入虛擬的 N 類別，才能成為一個分類的問題。因此我們把二維灰階統計直方圖視為資料空間，在二維灰階統計直方圖上的資料點記為 Y 類別，並導入虛擬的 N 類別。

然而我們估算 N 類別的方式並沒有和 IBM 研究中心所提出的基於決策樹之叢聚法的做法一樣。在基於決策樹之叢聚法中，若繼承於父節點之子節點其 N 類別數量小於 Y 類別數量的話，則必須將 N 類別數量調整到與 Y 類別數量一致；相反的，如果 N 類別數量大於 Y 類別數量的話，則不作任何的改變。此做法的動機是考慮到在如果是在高維度的空間上，經過多次切割後，則 N 類別的數量將會以指數方式激烈地下降。若 N 類別數量太少的話，對於未來若還必須做切割，其切割會變得很困難。

然而在我們的方法中，因為資料空間只有二維，所以在切割的過程中，區域中 N 類別的數量仍在可接受的範圍，並不會有太少的情形發生。而且若切割後，區域內 Y 類別數量若遠大於 N 類別數量的話，意味著其資料點數相當地相似，同質性非常高，則此區域也許不用再進行切割。因此我們並不採用調整 N 類別數量與 Y 類別數量一致這樣的做法，而是採用固定 N 類別數量的做法，即初始將 N 類別數量設定與 Y 類別數量一致，並均勻地加入資料空間後，在切割過程中，不論大於或小於 Y 類別數量皆不做任何地改變。

步驟三：求取最佳切割點

我們以決策樹演算法 C4.5 版本中的資訊增益方法為建樹準則，做法是分別在 X 軸和 Y 軸以資訊增益估算出各別的最佳切割點。因此即可得到二個分閥值，並利用這二點，分割出四塊區域。關於求取資訊增益計算方法，請參照於第二章所列出的公式(2-17) (2-18) (2-19) (2-20)。

因我們的主要目的是求取多重分閥值，所以必須在各個區域中判斷是否需要再進行分割，若是不符合我們停止分割的條件，則以遞迴的方式再做進一步的分割動作，也就是叢聚樹的建樹動作。

在建構叢聚樹中，基本上是先以我們提出的判斷建樹的停止條件，來判斷二維灰階統計直方圖上各區域內其資料點是否一致。若是不一致，則分別再利用資訊增益來計算子區域內 X 軸和 Y 軸的最佳切割點，並再分割為四個子區域，以遞迴方式切割到各個區域內的資料點為同一類別，全部符合建樹的停止條件為止。所以建出來完整的樹是一顆四元樹。

步驟四：判斷停止建樹條件

我們的想法是若是必須繼續分割的節點，一定是 Y 類別資料量夠多與相對密度較小，並且此分割必須能夠區分出相異的群數才作分割。因此在這裡定義了三個條件來判斷各區域是否停止建樹。分別是節點（二維灰階統計直方圖上各區域內）內 Y 類別的資料量、相對密度、與切割後是否有分割出相異的二群。如果節點符合任一個條件即停止分割，並且設為葉節點：

（一）利用參數 min_y ：

若區域內的資料點太少，則該節點以下所生長出來的節點將不是我們所感興趣的，因為其若再繼續分割，那麼將會使資料空間分割得太過細微，則分割出來的子區域將是沒有意義而且是多餘的。這樣的想法是由基於決策樹之叢聚法中修剪叢聚樹的概念而來的。因此 min_y 此參數是指在一個區域中所應包含的資料點之最少數量，表示為佔資料總數的百分比，如公式（3-1）所示，

Y_k , $k=1\sim 4$, 代表分割後四區域內的 Y 類別之資料總量, $|D|$ 代表整個資料空間的 Y 類別資料量總數。

$$Y_k < \min_y \times |D| \quad (3-1)$$

例如設定 $\min_y=5\%$, 則以一張 256×256 的灰階圖來說, 若區域中資料點數少於 3276 點, 則此區域就停止分割。在麗娜影像的實驗中則設定為 $\min_y=5\%$ 。

我們以圖 3-4 來說明, 在圖中 R14 這塊區域, 若所包含的點數只有 350 點, 佔不到整個空間資料總數 65536 點的百分之一, 若再繼續分割, 將會產生無意義的群數, 因此我們即停止分割動作。

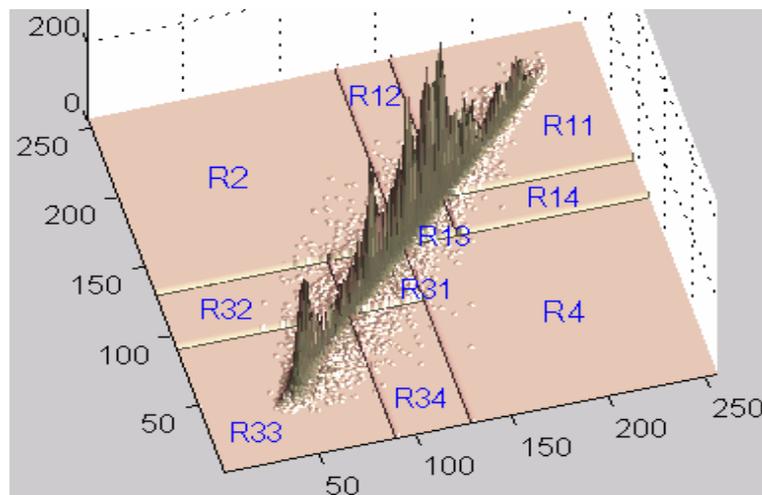


圖 3-4 切割後的二維灰階統計直方圖

(二) 利用參數 \min_rd :

若節點內的相對密度 (一個區域的相對密度 = Y 類別資料量除以 N 類別資料量) 太高, 我們就不再對其區域進行切割, 因為資訊增益所找出來的最佳切割點是將資料空間切割為資料密集與資料稀疏的區域, 所以我們計算資料密集區域裡的相對密度, 若

其相對密度大於 min_rd ，則此節點就停止建樹。如公式(3-2)所示， N_k ， $k=1\sim 4$ ，代表分割後四群的 N 類別之資料總量。因為在相對密度相當高的區域中，意味著區域內的資料同質性非常高，所以無須再進一步切割。

$$\frac{Y_k}{N_k} \geq min_rd \quad (3-2)$$

而在實驗中 min_rd 設定在 $[2, 4]$ 這個區間是較為理想的。我們以圖 3-4 來說明，在圖中 R31 這塊區域，若 Y 類別的點數有 15663 點， N 類別的點數有 1521 點，其相對密度高達 10.3，相當地密集，因此我們既視為其區域為同一群而不再繼續分割。

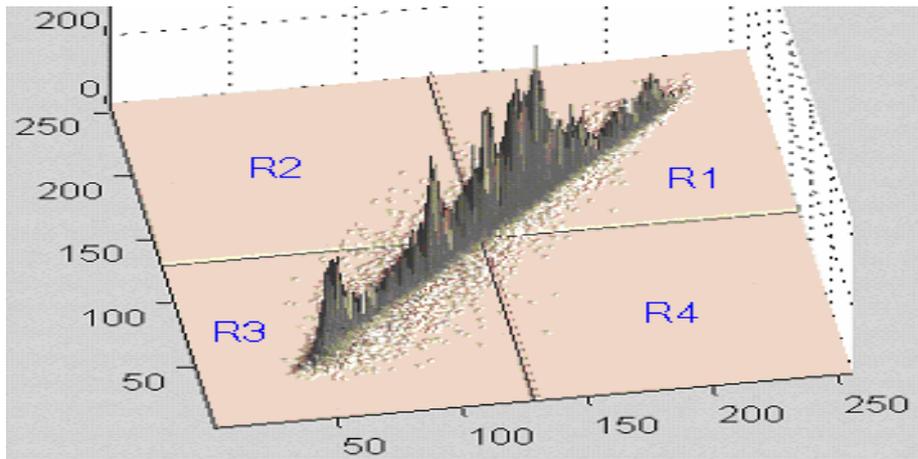
(三) 假設有一個區域 $R1$ ，利用資訊增益各在 X 軸和 Y 軸求出二個最佳切割點，並把區域切割成四塊子區域後。假如切割後只有一塊子區域或沒有任何一塊子區域的 Y 類別之資料量超過 $min_y \times |D|$ 的話， $|D|$ 為整個資料空間的 Y 類別之資料總量，表示並沒把 $R1$ 這個區域內的資料分成相異的二群，而是全都集中於一群。很顯然地此分割是多餘的，我們就判斷 $R1$ 這個區域不做分割。

至於為何 Y 類別之資料量大於 $min_y \times |D|$ 才當作是一群，因為 Y 類別之資料量小於 $min_y \times |D|$ 的區域，我們並沒有把它視為一群，而先把它視為雜訊，留待最後才處理，這個動作在本章節的後面有較詳述的說明。

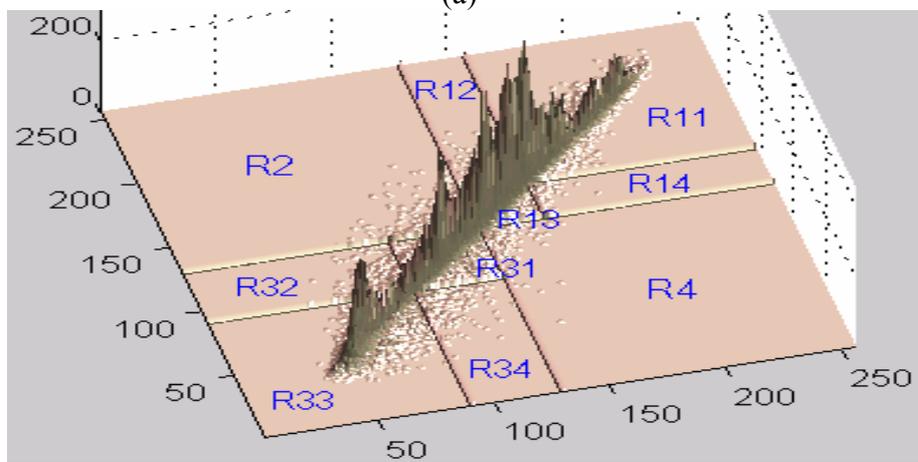
我們以麗娜灰階影像來進行分割，其分割後之每一區域內 Y 類別之資料量若少 3276 點或其相對密度小於 3，則停止分割，否則繼續分割。

以圖 3-9 來說明分割的過程。圖 3-5(a)為麗娜灰階影像的二維灰階統計直方圖之第一次切割。X 和 Y 軸的最佳切割點分別落在灰階度 129 及 129 的地方，並且切割為四群，記為 R1、R2、R3、R4。以停止建樹的判斷條件來判斷這四群是否須再進行分割，並以遞迴的方式直到所有群數皆符合停止條件為止。圖 3-5 (b)是其第二次切割。從圖中可得知，R2 和 R4 這二區域皆符合停止建樹的條件。而區域 R1 之切割點落在(157, 156)的地方。而區域 R3 之切割點落在(90, 90)的地方。圖 3-5 (c)為第三次切割。R11 和 R33 這二區域繼續被分割，各自切在(182, 181)和(60, 61)上。在第三次切割後，其所有群數皆符合停止建樹的條件，結果共分為 16 群。

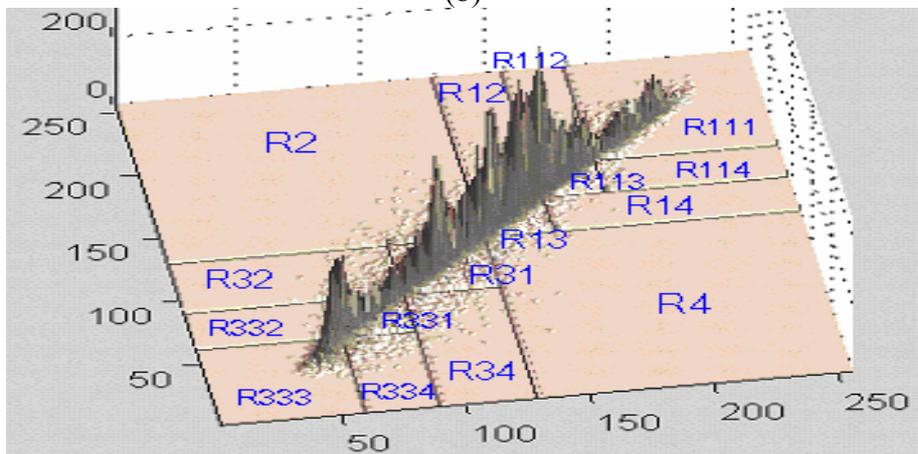
圖 3-6 為對應圖 3-5 所建構的叢聚樹。此叢聚樹共有四個階層，16 個葉節點。各節點內的 Y 和 N 分別代別區域內 Y 類別的資料量和 N 類別的資料量。圖 3-7 為麗娜灰階影像分割為 16 群的結果。



(a)



(b)



(c)

圖 3-5 麗娜影像之二維灰階統計直方圖分割過程(a) 第一次切割：(129, 129)

(b) 第二次切割：區域 R1 切在(157, 156)，區域 R3 切在(90, 90)

(d) 第三次切割：區域 R11 切在(182, 181)，區域 R33 切在(60, 61)

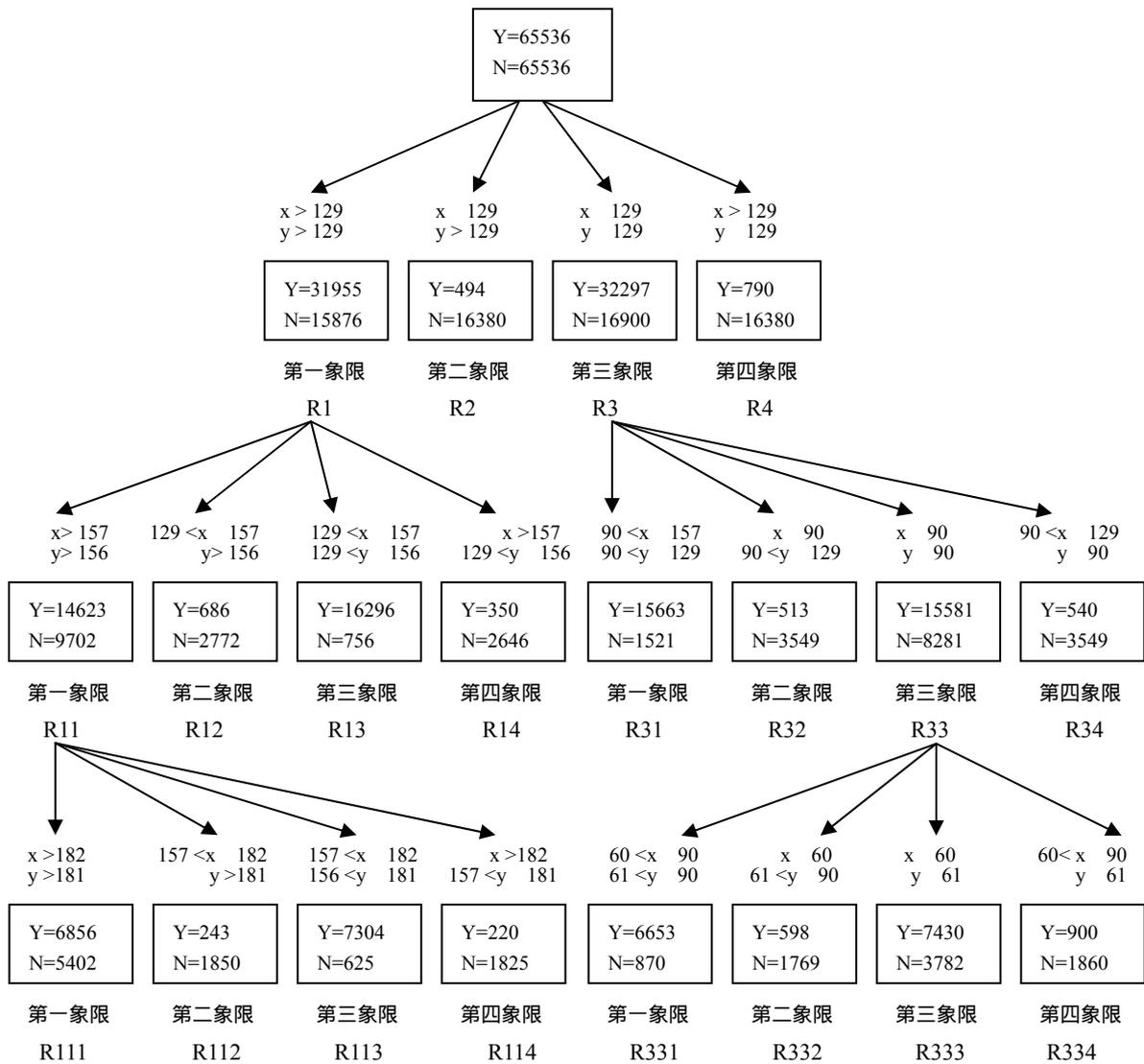


圖 3-6 完整的叢聚樹



圖 3-7 分割為 16 群的影像

第三節 叢聚樹的修剪

在叢聚樹建立好後，由於決策樹過度訓練 (overfitting)，會導致樹枝變得很複雜且沒有必要，因此需要修剪所得到的樹，才能產生有意義的分群。在修剪過程中，總共有三個步驟。步驟一為初步修剪，步驟二為合併相鄰區域，步驟三為雜訊的處理。

步驟一：叢聚樹的修剪 初步修剪

而在做修剪之前，為了使修剪動作更為簡單，我們把 Y 類別點數相當少的區域先進行初步地修剪，因為這些點數相當少的區域一定是位於二維灰階直方圖中離對角線較遠的位置，意味著這些區域內的資料點不是雜訊就是邊界值，因此若這些少量的資料各自皆形成一群的話，會變得群數太多，若不先進行刪除，會使得叢聚樹變得非常龐大，對於接下來樹的修剪會變的很煩雜，而且沒有意義。

因此我們在這裡以停止建樹中的參數 min_y 來判斷區域是否進行初步修剪，主要是以各區域內的資料點數來做判斷。若 Y 類別的總數小於 $min_y \times |D|$ ，則把此區域相對之葉節點暫時由叢聚樹切除不處理，直到整個修剪動作結束後才處理這些區域。

圖 3-8 是麗娜灰階影像初步修剪後的叢聚樹。原本為 16 個葉節點，經初步修剪，共有十個葉節點暫時切除不處理，使得叢聚樹剩下六個葉節點。這六個葉節點分別為 R13、R31、R111、R113、R331、R333，對應到圖 3-5 (c)，可發現這些葉節點皆位於對角線的區域。這代表這些區域為影像中的主要物件；而不在對角線上的區域，則可能是雜訊或是物件的邊界值。

圖 3-9 為分割為 6 群後的結果，而初步修剪的十群的資料點暫時以以

灰階度 255 取代。很顯然地這十群的資料點幾乎都置於邊界上，並不是主要的群數，故暫時刪除以簡化往後的過程。

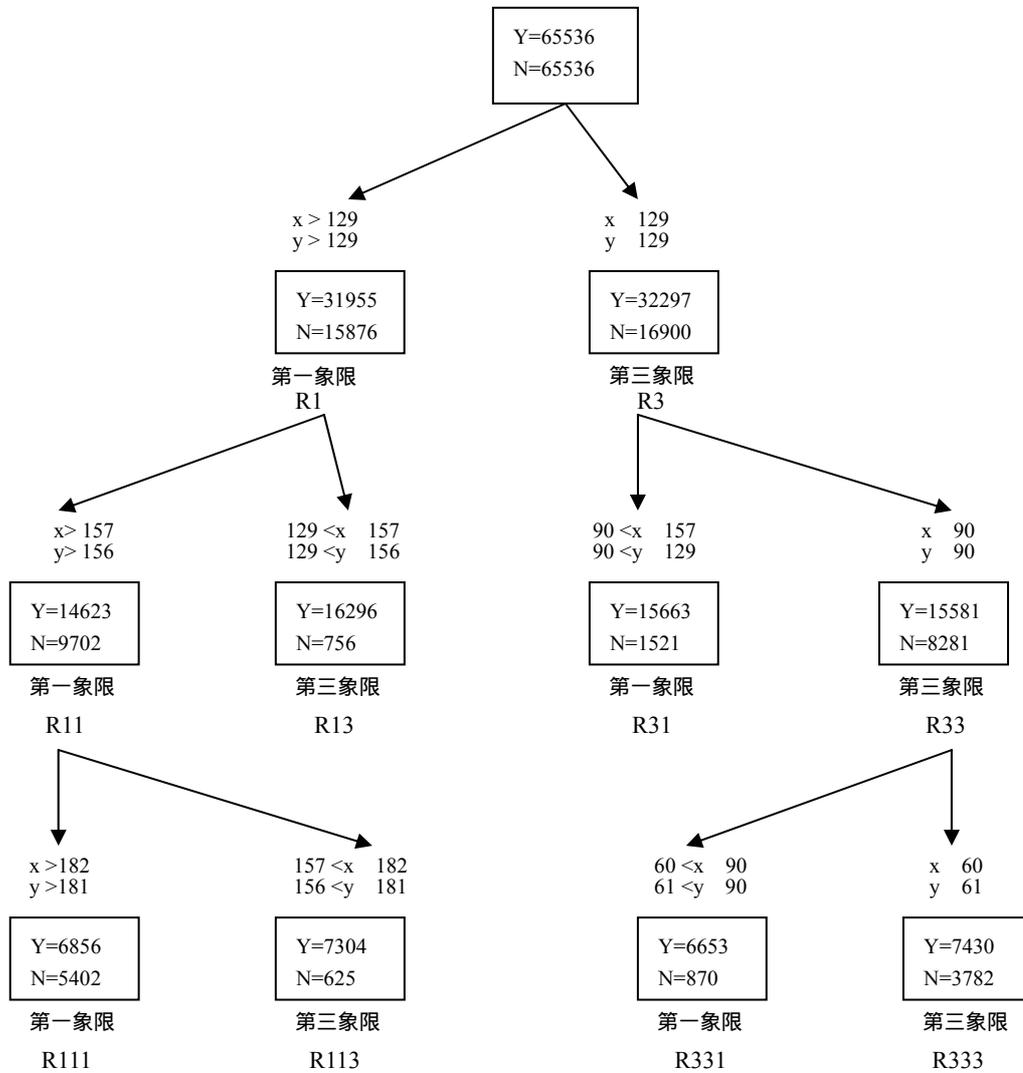


圖 3-8 初步修剪的叢聚樹



圖 3-9 6 群的影像，雜訊未處理

步驟二：叢聚樹的修剪 合併相鄰區域

1. 判斷相鄰區域：

一個區域的相鄰區域是其相鄰之上、下、左、右和其對角線之左上、右上、左下、右下這些區域，例如在圖 3-10 中，A 的相鄰區域即是 B、C、D、E、F、G、H 及 I 這八個區域。然而在初步修剪中，已先將點數過少的節點視為雜訊而暫時刪除。這些被刪除的節點皆位於二維灰階統計直方圖上非對角線的位置，因此這些節點皆不需做相鄰節點的判斷。所以只有沿對角線的區域必須做合併的動作，這樣會使得合併的動作更為簡易。

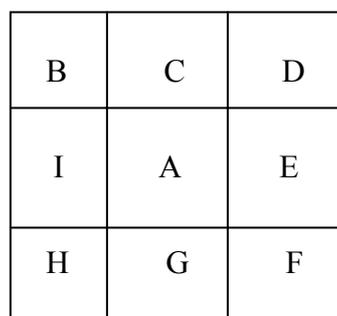


圖 3-10 相鄰區域

我們能以叢聚樹中每個葉節點在二維灰階統計直方圖上的 x 軸和 y 軸的範圍，來判斷各個葉節點的相鄰節點。以圖 3-8 為例，我們知道 R331 這個葉節點在二維灰階統計直方圖上是位於 x 軸小於及等於 90，大於 60、y 軸小於及等於 90，大於 61 的區域。因此，其左下對角的相鄰節點一定是位於 x 軸小於及等於 60 與 y 軸小於及等於 61 的區域。其右下對角的相鄰節點一定是位於 x 軸大於 90 與 y 軸大於 90 的區域。很明顯的，R333、R31 即是 R331 的左下對角與右下對角的相鄰區域。

所以使用者即能自行地去瀏覽叢聚樹以找出各個葉節點的相鄰節點。以這樣的方式，即能將叢聚樹各個葉節點的相鄰關係轉換成相鄰節點表格 (adjacent node's table) 來表示。如圖 3-11 所示，表示在圖 3-8 中六個葉節點與其相鄰節點的關係。因為只有沿著對角線的區域才要考慮是否合併，所以一個葉節點的相鄰節點最多只有二個節點。

葉節點	相鄰節點	相鄰節點
R111	R113	
R113	R111	R13
R331	R333	R31
R333	R331	
R13	R113	R31
R31	R331	R13

圖 3-11 相鄰節點表格

2. 合併條件：

合併的條件會根據區域是否相鄰及相對密度是否相似來進行合

併。這樣的想法的理由是，在二維灰階統計直方圖中相鄰之資料區域，若將之合併後其相對密度不會造成很大的變化，並且能與相對密度低的區域有所區別，那麼合併在一起是很合理的。

若要判斷葉節點 A 與其相鄰區域 B 是否應該合併，如公式(3-3)所示。先比較 B 和 A 的相對密度，取較大的來和合併後的相對密度做比較。若大於我們所設定的參數 P ，則將相對密度較小的區域合併到相對密度大的區域。

$$\frac{d(A+B)}{\max(d(A), d(B))} \geq P \quad (3-3)$$

$d(A)$ 表示區域 A 的相對密度 (Y 類別資料量除以 N 類別資料量)。而在實驗經驗中 P 範圍設定在 $[0.65, 0.99]$ 這個區間內，會有較好的表現，因為這表示合併後的相對密度和合併前的密度是差不多的，才能合併成一類。 P 愈高，合併條件愈嚴苛，表示二類的相似度要愈高，因此可能合併的群數也就變少。

3. 合併先後規則：

我們的修剪順序首先從樹最底層的最左葉節點開始，先考慮同階層中於二維灰階統計直方圖中相鄰資料區域是否應該合併。當同階層考慮完畢之後，才考慮另外一個相鄰區域，可能是同一子樹中上一階層位於二維灰階統計直方圖中相鄰之資料區域。而一旦有節點相合併後，因為合併後節點之相鄰節點將會改變，所以必須重新調整“相鄰節點表格”，並且刪除相對密度比較小的節點。同樣的做法一直以遞迴的方式由下往上來判斷各個葉節點與其相鄰節點是否合併，直到叢聚樹階層最小的葉節點為止。

圖 3-12 為合併相鄰區域演算法。 L_i 為葉節點， i 的範圍為 $1 \sim n$ 。

A_{ij} 為 L_i 的相鄰節點， j 最多為 2。

```

Algorithm evaluatePrune( $L_i, A_{ij}, P$ )
For  $L_i \in \{L_1, L_2, \dots, L_n\}$  do
  For  $A_{ij} \in \{A_{i1}, A_{i2}\}$  do
    if  $L_i$ 's relative density is higher than  $A_{ij}$ 's relative density
      newregion =  $L_i$ ;
    else newregion =  $A_{ij}$ ;
    end
    if  $d(L_i + A_{ij}) / d(\text{newregion}) > P$ 
      merge( $L_i, A_{ij}$ ) newregion; /* we can merge  $L_i$  and  $A_{ij}$  to
      adjust Adjacent region's table; produce only one cluster */
      delete  $\min(d(L_i), d(A_{ij}))$ ; /* if  $d(L_i) < d(A_{ij})$  the delete  $L_i$ 
    end
      else delete  $A_{ij}$  */
    end
  end
end
end

```

圖 3-12 合併相鄰節點演算法

合併的順序是由樹的最底層開始，然後以置於樹的最左節點 R111 開始做合併判斷。由左至右，當同一階層的葉節點皆修剪完，則繼續往上一層，直到階度最小的葉節點 R31 為止。因此，圖 3-11 由上而下即是此叢聚樹合併相鄰區域的順序。而各個葉節點在與相鄰節點做合併時，必須先和同階層的相鄰節點做判斷。例如 R331 這個葉節點，則應先與同階層的 R333 進行合併的判斷，接下來才與上一階層的 R31 做合併判斷。

我們將參數 P 設為我們設定的 75%，並舉個例子來說明。在圖 3-11 中，R113 的相鄰區域為右上角的 R111 和左下角的 R13 這二塊區域，其他的相鄰區域因在初步修剪時已暫時刪除故不考慮。在叢聚樹中 R113 和 R111 為同一階層，故先判斷是否與 R111 合併。由圖 3-8 叢聚樹可得知各個葉節點 Y 及 N 類別的資料量，因此即可估算相對密度。R113 和

R111 的相對密度各為 11.686、1.269，而這二群合併後之新相對密度為 2.726。利用公式(3-3)求出的結果為 23%，小於 75%，故這二群非常相異，故不合併。

接下來 R113 再與相鄰的上一階層區域 R13 做比較。R13 的相對密度為 21.556，合併後之新相對密度為 17.089，二群的相似度有 79%，大於我們設定的 75%。因此 R113 被合併到 R13，而 R113 這個節點則被刪除掉，並且 R13 與其相鄰節點則必須重新調整。如圖 3-13(a)為 R113 被合併到 R13 後，相鄰節點表格調整後的結果。

就以這樣的步驟，由下往上，直到叢聚樹階層最小的葉節點為止。除了上述的 R113 被合併到 R13 外。R331 也合併到 R31 而形成一群。如圖 3-13(b)為 R331 被合併到 R31 後，相鄰節點表格調整後的結果。如圖 3-14 所示，其為圖 3-8 叢聚樹的合併結果。因此葉節點由六個減少為四個。圖 3-15 為這四群的分割結果，而雜訊點未處理。

葉節點	相鄰節點	相鄰節點
R111	<u>R13</u>	
R331	R333	R31
R333	R331	
R13	<u>R111</u>	R31
R31	R331	R13

(a)

葉節點	相鄰節點	相鄰節點
R111	R13	
R333	<u>R31</u>	
R13	R111	R31
R31	<u>R333</u>	R13

(b)

圖 3-13 (a) R113 合併到 R13 後之相鄰節點表格
(b) R331 合併到 R31 後之相鄰節點表格

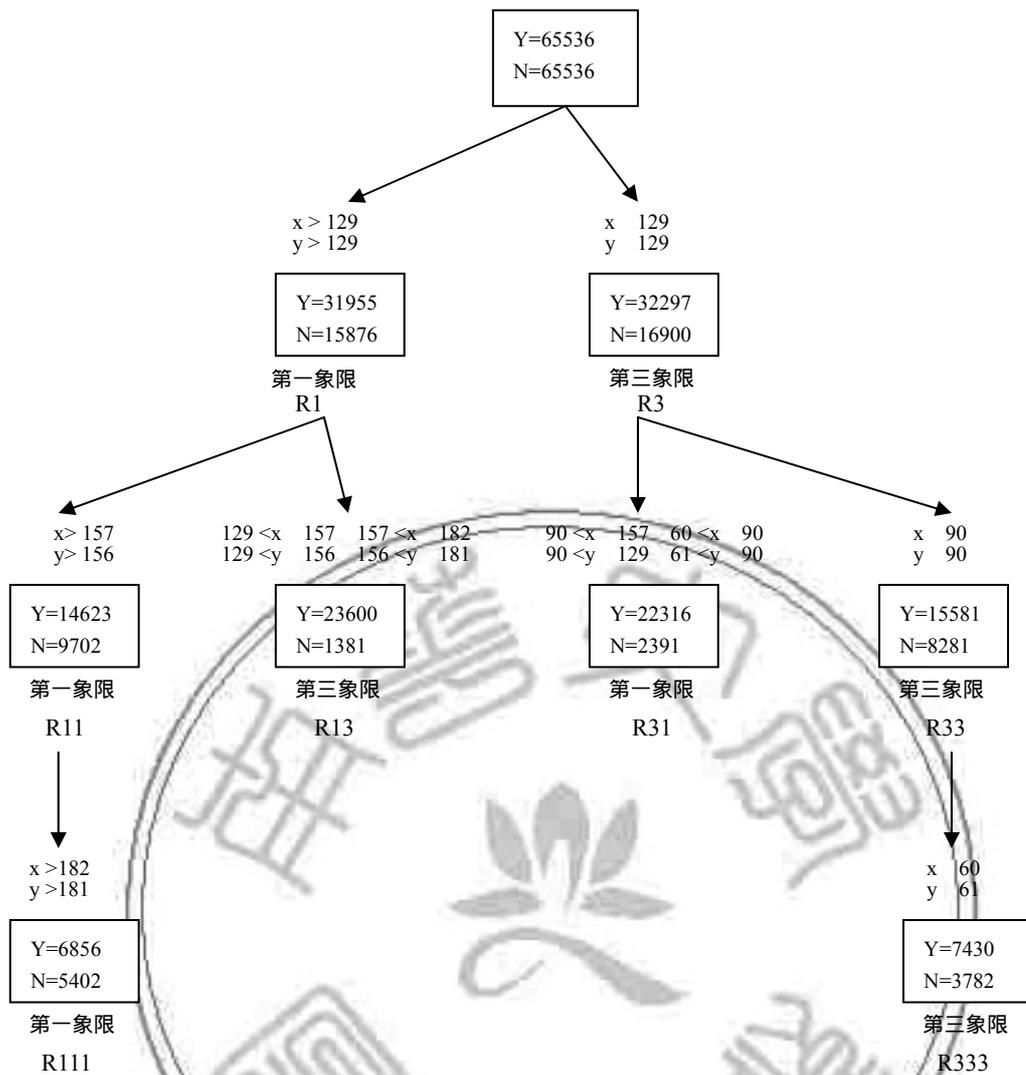


圖 3-14 修剪合併後的叢聚樹



圖 3-15 分割為 4 群之結果，雜訊未處理

步驟三：初步修剪的處理

最後一個步驟是處理在初步修剪過程中被視為雜訊值而暫時不處理的區域。我們採用歐基里德距離（Euclidean distance）來量測這些區域內的資料與合併後各群的相似程度性。如公式(3-4)所示，我們把這些區域中的點數皆設為雜訊，以 x_l 來表示雜訊值的集合。並且計算合併後各叢聚的群心，以 $c_i, i=1\sim j$ ，來表示第 i 個叢聚的群心。若雜訊點與第 i 個叢聚的群心之距離小於其與其他群心的距離，則歸屬於第 i 群。

$$\|x_l - c_i\| = \min_{i=1}^j \|x_l - c_i\| \quad (3-4)$$

$\| \cdot \|$ 表示歐基里德距離的計算

如圖 3-14 所示，為初步修剪已處理過的最終結果，最後分為 4 群。



圖 3-14 最終結果-共 4 群

第四章 實驗結果與討論

第一節 實驗設計

在本章中，我們實作我們所提出之演算法，並與 L. Cao, et al. [16]於 2002 年所提出之自動多重分閥值法做比較。我們使用 Matlab 6.1 的開發工具來實作程式。執行環境為 Pentium -866 MHz, 384MB 之記憶體，作業系統為 Microsoft Windows 2000。我們實驗影像是以灰階值大小 256×256 的灰階影像來測試。而影像分為人造影像 (synthetic image) 與五組真實影像 (real-world images)。這五組影像包含了麗娜、蝴蝶、心血管、青椒、和鸚鵡。見圖 4-2 至圖 4-6。

第二節 人造影像之實驗結果與討論

首先在人造影像的實驗結果中，人造影像為像素值大小 256×256 的人造影像，並加入高斯雜訊 (Gaussian Noise), SNR 為 10dB, 如圖 4-1(a) 所示。在人造影像中，主要分為五個物件，為圓形、矩形、正方形、三角形、和不規則狀之區域。而我們從肉眼能得知這張人造影像雖然主要分為五個物件，然而這五個物件的灰階度只分為二群，即左邊灰階度較黑的圓形與三角形外圍的正方形，其餘的物件其灰階度較偏向於白色。

由圖 4-1(b) 是利用表 4-1 中人造影像之停止建樹及修剪樹的條件來進行分割，分割後之二維灰階統計直方圖主要分為二群。即第一象限與第三象限。第一和第三象限的資料點這二群分別代表為灰階度較小與較大的物件；而第二及第四象限的資料點，我們視為雜訊來處理。因此由圖

4-1(c)所知，我們的方法有把主要的群數區分出來。而與圖 4-1(d)相較之下，二者分割的結果皆能有效地區分出各個物件。

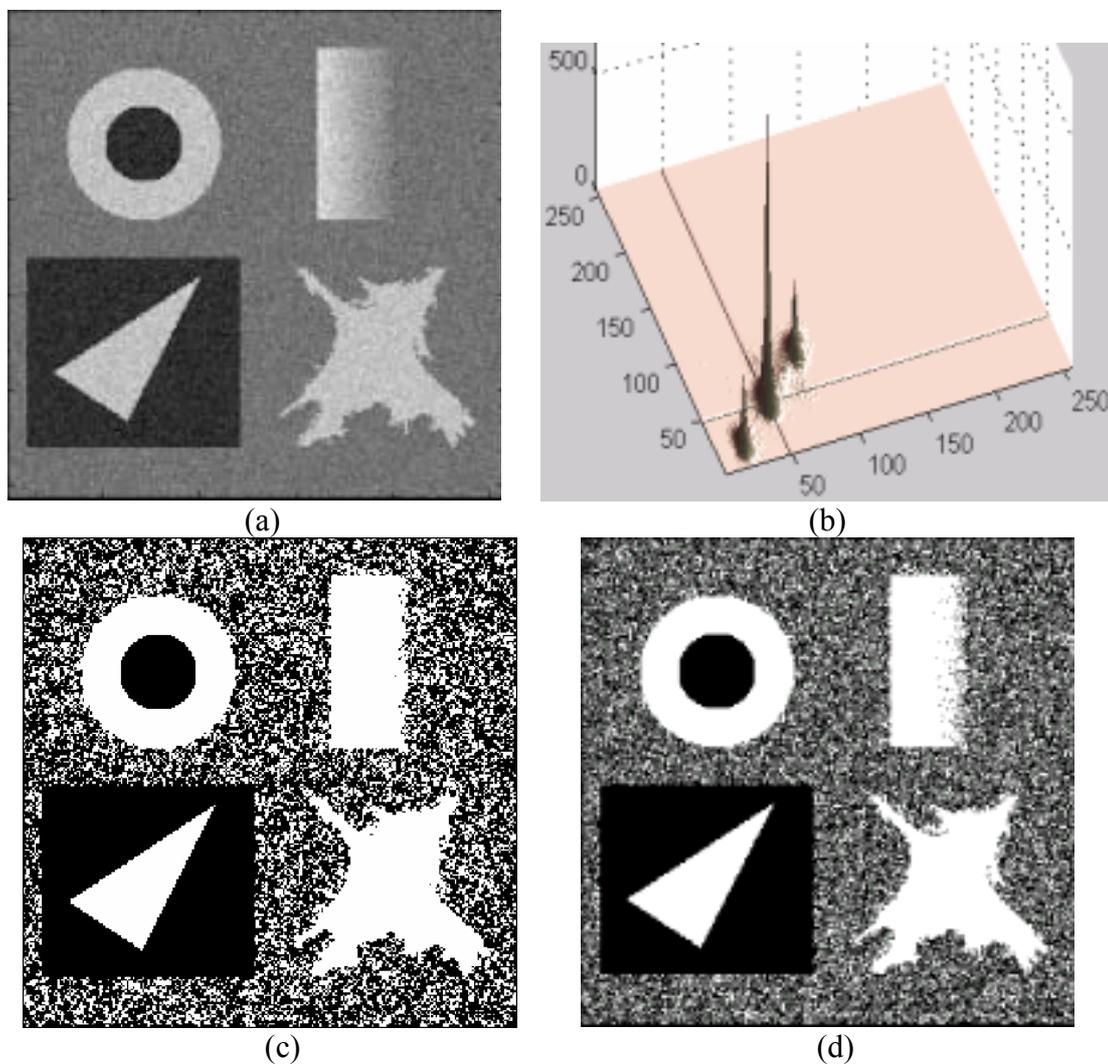


圖 4-1 (a) 為人造影像 (b) 依據我們的方法分割二維灰階統計直方圖後的情況。切割點為(50, 50) , 主要分為二群：第一象限區域和第三象限區域 (c) 我們方法之結果：二群之影像 (d) L. Cao 提出的快速自動多重分閥值法之結果：三群之影像，分閥值為 45、53

第三節 真實影像之實驗結果與討論

在五組真實影像的實驗中,表 4-1 列出了以我們的方法分別為這五組影像,所設定的停止建樹條件與修剪條件。這些條件皆是依據經驗判斷所給定的。而其影像分割的結果分別顯示在圖 4-2 至圖 4-6 中。

	停止建樹條件		修剪條件	
	<i>min_y</i>	<i>min_rd</i>	<i>min_y</i>	<i>P</i>
人造影像	15%	3	15%	75%
麗娜	5%	3	5%	75%
蝴蝶	5%	3	5%	75%
心血管	5%	3	5%	70%
青椒	5%	3	5%	85%
鸚鵡	5%	3	5%	97%

表 4-1 停止建樹條件和修剪條件

由這五組實驗結果的二維灰階統計直方圖中,圖 4-2 至圖 4-6 中的 (b)所示。可以發現我們分割後的群數皆置於對角線的位置上,這說明了我們正確地將整張影像前景和背景的地方,切割出多個主要的物件出來。而離對角線較遠的皆是點數極少的區域,意味著這些點數可能是雜訊或是邊界值。在我們的方法中,皆是將這些區域視為雜訊,並採用歐基里德距離來量測這些區域內的資料與合併後各群的相似程度性。因此由圖 4-2 至圖 4-6 中的(c)所示,可以看出我們的方法能有效地處理雜訊。

從這五組實驗的分割影像結果,圖 4-2 圖 4-6 中的(c)所示。可以

看出我們的方法在沒有預先指定群數下，能自動地將影像中主要物件分割出來。在與 L. Cao 所提的快速自動多重分閥值法之結果的比較中，因它是必須預先指定分割群數，所以其指定的群數數目，以我們所提出的方法之分割結果，來當作比較方法之群數數目。

一、麗娜灰階影像之討論

圖 4-2(a)為麗娜灰階影像。其二維灰階統計直方圖，經分割後如 4-2(b)所示，共分為六群，區域 R1 至 R6。從圖中，可以發現我們的分割點，皆落於山谷的位置，因此能有效地分割出物件。然而在修剪後，R2 和 R3 合併成一群，R4 和 R5 合併成一群。因此其結果為四群。圖 4-2(c)為四群之影像分割的結果。在與 L. Cao 所提出的快速自動多重分閥值法之結果比較之下，圖 4-2(d)所示。我們的結果的影像較差，背景及臉部的輪廓較不明顯。

二、蝴蝶灰階影像之討論

圖 4-3(a)為蝴蝶灰階影像。其二維灰階統計直方圖，經分割後如圖 4-3(b)所示，共分為五群，區域 R1 至 R5。圖 4-3(c)為五群之影像分割的結果。在與圖 4-3(d)相較之下。在蝴蝶右翅上方和右邊之樹葉，我們的結果有把樹葉之紋路區別出來，而 L. Cao 提出方法之結果則呈現模糊的情況。而影像中右翅的右下方之背景，我們的結果則與右方葉子上方之影子連成一群，使得右下方之背景較不清楚。然而圖 4-3(d)之右下方背景則有區別出來。

三、 心血管灰階影像之討論

圖 4-4(a)為心血管灰階影像。其二維灰階統計直方圖，經分割後如圖 4-4(b)所示。共分為三群，區域 R1、R2、R3 與 R4 合併成一群。圖 4-3(c)為三群之影像分割的結果。在與圖 4-4(d)相較之下，整體來講，我們的方法效果較為而 L. Cao 提出方法之結果來得好，血管皆有區分出來。而圖 4-4(d)的結果其血管與背景皆連成了一片。

四、 青椒灰階影像之討論

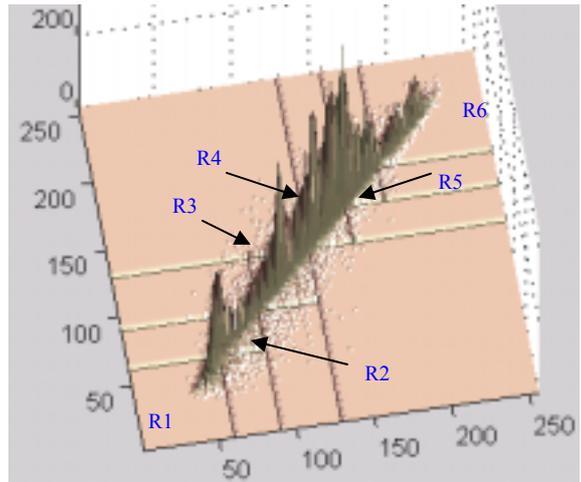
圖 4-5(a)為心血管灰階影像。其二維灰階統計直方圖，經分割後如圖 4-5(b)所示。共分為六群，區域 R1 至 R6。圖 4-5(c)為三群之影像分割的結果。在與圖 4-5(d)相較之下，我們的方法在影像最左邊的青椒較 L. Cao 提出方法之結果來得清楚。而就以上方較大的青椒分割結果來說，圖 4-5(d)較有立體感，而我們的方法在青椒的左方上則有黑點出現。

五、 鸚鵡灰階影像之討論

圖 4-6(a)為鸚鵡灰階影像。其二維灰階統計直方圖，經分割後如圖 4-6(b)所示。沿著對角線共分為六群。圖 4-6(c)為六群之影像分割的結果。在與圖 4-6(d)相較之下，在我們的方法中，鸚鵡頭頂上的羽毛，我們的結果較 L. Cao 提出方法之結果，羽毛的灰階度較為一致。L. Cao 的結果則雜訊較多。而在影像中右下方的羽毛紋路，我們的結果則沒有 L. Cao 的結果來得明顯。



(a)



(b)



(c)

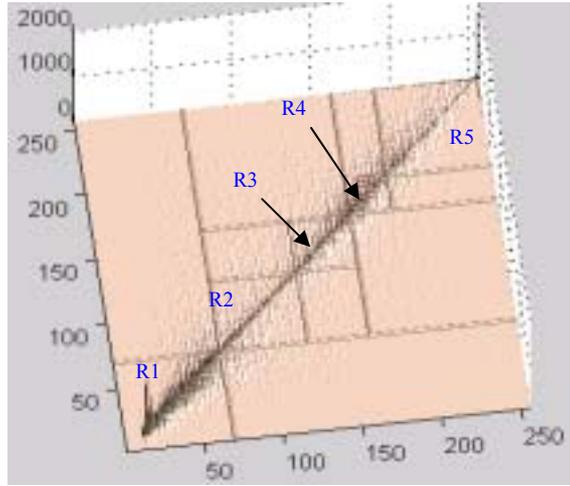


(d)

圖 4-2 (a) 為原影像 (b) 依據我們的方法分割二維灰階統計直方圖後的情況。經修剪後分為四群，即 R1、R2 與 R3 合併、R4 與 R5 合併、R6 (c) 我們方法之結果：四群之影像 (d) L. Cao 提出的快速自動多重分閾值法之結果：四群之影像，分閾值為 91、129、155



(a)



(b)

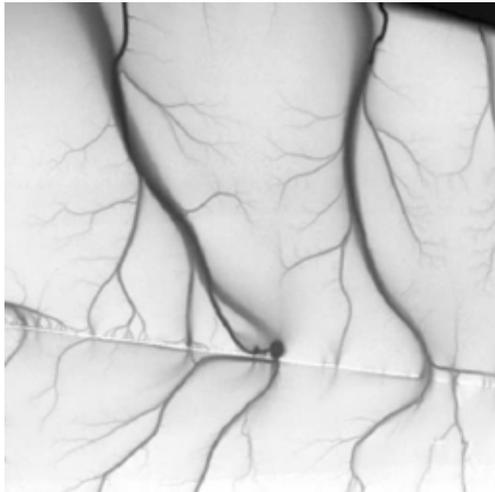


(c)

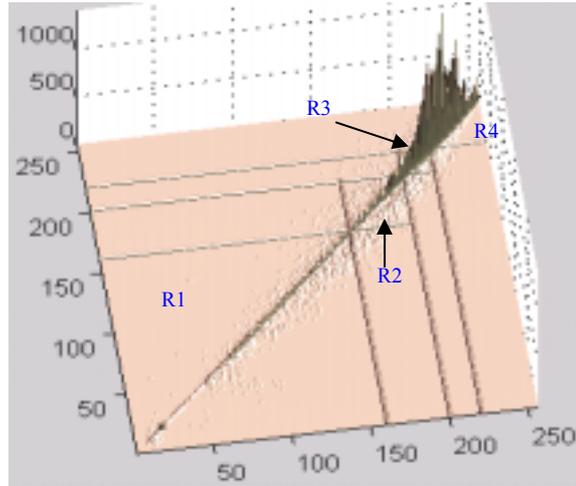


(d)

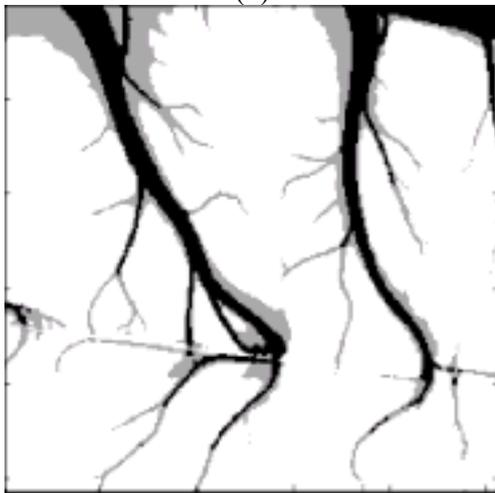
圖 4-3 (a) 為原影像 (b) 依據我們的方法分割二維灰階統計直方圖後的情況，共分為五群，R1 至 R5 (c) 我們方法之結果：五群之影像 (d) L. Cao 提出的快速自動多重分閥值法之結果：五群之影像，分閥值為 30、50、93、168



(a)



(b)



(c)



(d)

圖 4-4 (a) 為原影像 (b) 依據我們的方法分割二維灰階統計直方圖後的情況，共分為三群，R1、R2、R3 與 R4 合併成一群 (c) 我們方法之結果：三群之影像 (d) L. Cao 提出的快速自動多重分閥值法之結果：五群之影像，分閥值為 214、227

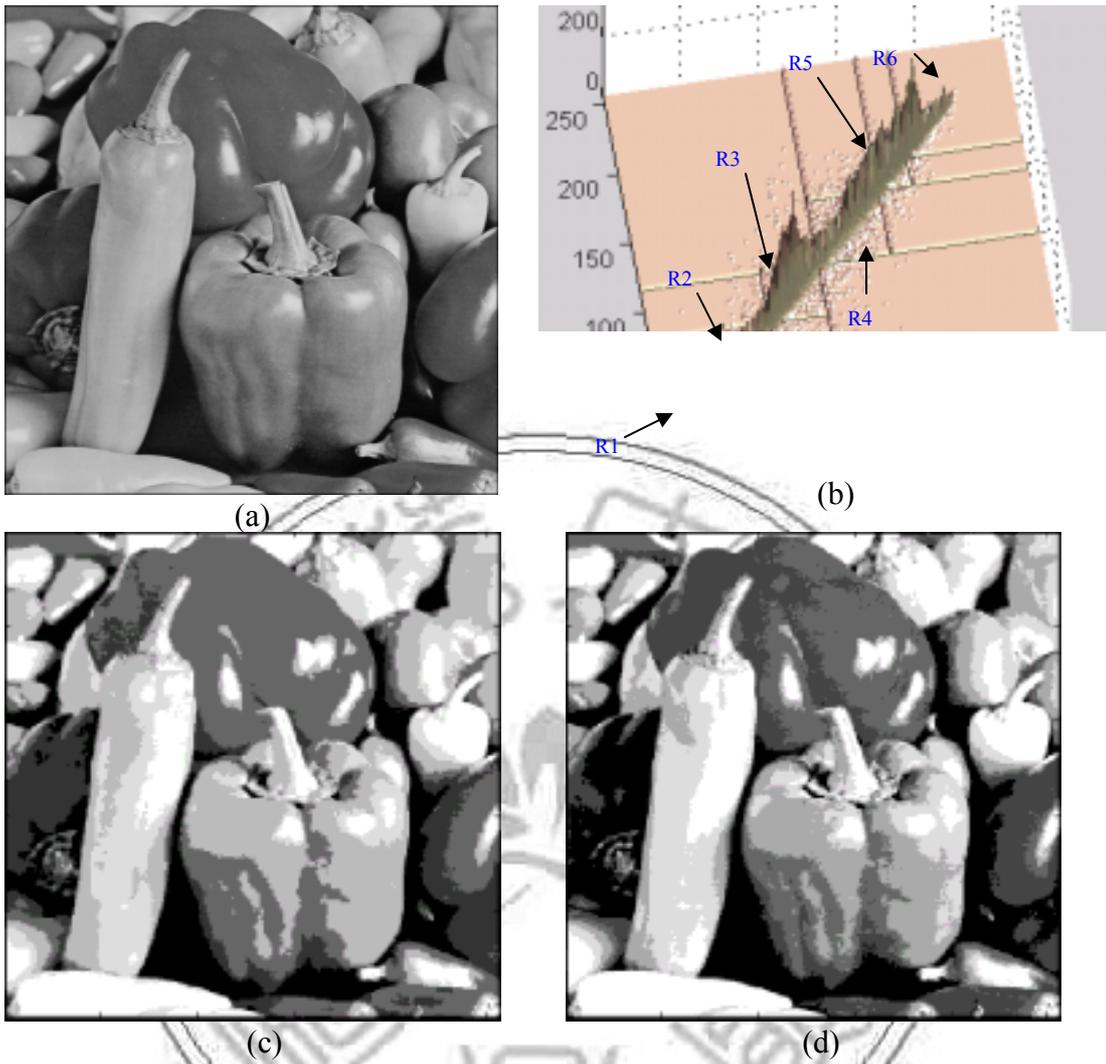
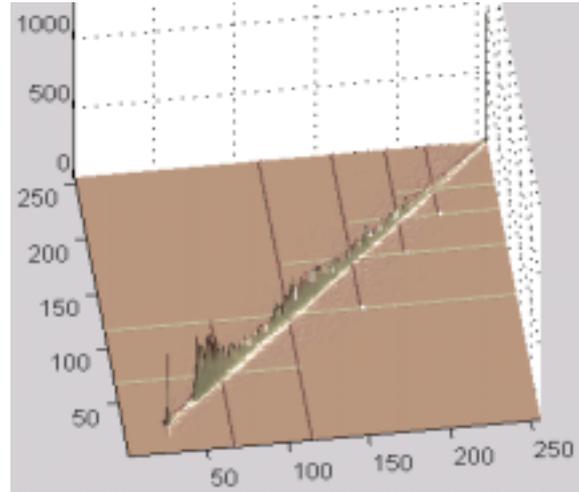


圖 4-5 (a) 為原影像 (b) 依據我們的方法分割二維灰階統計直方圖後的情況，分割為六群，R1 至 R6 (c) 我們方法之結果：六群之影像 (d) L. Cao 提出的快速自動多重分閥值法之結果：六群之影像，分閥值為 63、90、116、150、175



(a)



(b)



(c)



(d)

圖 4-6 (a) 為原影像 (b) 依據我們的方法分割二維灰階統計直方圖後的情況，共分為六群 (c) 我們方法之結果：六群之影像 (d) L. Cao 提出的快速自動多重分閾值法之結果：六群之影像，分閾值為 58、81、114、138、179

第五章 結論與未來展望

第一節 結論

在論文中，我們提出一個植基於決策樹建構法之二維多重分閥值法。其方割的過程是以資訊增益 (Information Gain) 為準則，經由建構叢聚樹將二維灰階統計直方圖做切割。其具有二項優點：

1. 改善了以往以熵值為基礎之多重分閥值法 (multilevel thresholding) 大都需預先指定群數的缺點。它利用叢聚的觀念來分析資料空間本身是否具有叢聚的現象。因而能找到最適當的群數。
2. 對於雜訊能夠更有效地處理，因為這些資料點數一定是置於點數非常少的區域。我們採取歐基里德距離來衡量雜訊值與合併後各叢聚的相似程度。讓這些資訊不致於被忽略，減少分割錯誤的產生。

由人造影像和真實影像的實驗結果可發現，我們的多重分閥值法能有效地分割出多個物件，並與 L. Cao et al. 提出的快速自動多重分閥值法的比較下，分割結果並不會差很多，甚至在有些影像中有更好的表現。

第二節 未來展望

在未來的發展上，因為在修剪叢聚樹的合併上，我們的準則是比較鄰近區域的相對密度，若這二群密度是相差不大，則合併。然而對於每張影像來說，各自有其灰階統計直方圖，因此在判斷合併條件時，每張影像所適合的相對密度相似程度的範圍皆不一樣。到底範圍需設多少，才會有較好的合併呢？這個動作則需要以經驗法則慢慢去嚐試，也許所

分割出來的區域皆代表各自的物件，若合併則可能發生錯誤。此時的合併參數 P 就必須取大一點，意味著群與群之間相異程度大，在有較嚴苛的條件下，則不會產生任何的合併動作；相反地，取小一點，則可能發生合併。因此未來希望能有更方便的設計，不用視不同情況去改變。

最後，我們的多重分閾值法目前所處理的是灰階影像，希望未來可以擴展到彩色影像，甚至是動態影像上的影像分割。

中文參考文獻：

- [1] 連國珍，數位影像處理，第二版，台北，儒林圖書，1999。
- [2] 繆紹綱，數位影像處理 活用 Matlab，全華科技圖書，2000。
- [3] 黃小娟，以決策樹為基礎的影像分群法，南華大學資訊管理學系碩士論文，2002。
- [4] 陳宏銘，設計最佳化之演化式模糊決策樹，逢甲大學資訊工程學系研究所碩士論文，2001。

英文參考文獻：

- [5] A. S. Abutaleb, “Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Entropies”, *Pattern Recognition* 47, pp. 22-32, 1989.
- [6] Bulent Sankur, Mehmet Sezgin, “Image Thresholding Techniques: A Survey Over Categories”, *Pattern Recognition*, 2001.
- [7] B. Liu, Y. Xia, P. S. Yu. “CLTree-Clustering through decision tree construction” Forthcoming, an earlier version appeared as an IBM Research Report RC 21695, 20/3/2000.
- [8] B. Liu, Y. Xia, P. S. Yu. “Clustering through decision tree construction”, *ACM CIKM-2000*, pp. 20-29.
- [9] C. K. Leung, F. K. Lam, “Maximum Segmented Image Information Thresholding”, *Graphical Models and Image Processing*, Vol. 60, No. 1, January, pp. 57-76, 1998.
- [10] E. David Jansing, T. A. Albert, D. L. Chenoweth, “Two-Dimensional Entropic Segmentation”, *Pattern Recognition Letters* 20, pp. 329-336, 1999.

- [11] J. N. Kapur, P. K. Sahoo, A. K. C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram", *CVGIP* 29, pp. 273-285, 1985.
- [12] J. C. Yen, F. J. Chang, S. Chang, "A New Criterion for Automatic Multilevel Thresholding", *IEEE Transactions on Image Processing*, Vol. 4, No. 3, March 1995.
- [13] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.
- [14] J. R. Quinlan, "C4.5: Programs for Machine Learning." San Mateo, CA: Morgan Kaufmann, 1993.
- [15] L. Cao, Z. K. Shi, E. K. W. Cheng, "Fast Automatic Multilevel Thresholding Method", *Electronics Letters*, Vol. 38, No. 16, 1st August 2002.
- [16] P. Sahoo, C. Wilkins, and J. Yeager, "Threshold Selection Using Renyi's Entropy", *Pattern Recognition*, Vol. 30, No. 1, pp. 71-84, 1997.
- [17] P. K. Sahoo, D. W. Slaaf, T. A. Albert, "Threshold selection using a minimal histogram entropy difference", *Optical Engineering*, Vol. 36, No. 7, pp. 1976-1981, July, 1997.
- [18] P. K. Sahoo, S. Soltani, K. C. Wong, "A Survey of Thresholding Techniques", *Computer Vision, Graphics, and Image Processing* 41, pp. 233-260, 1988.
- [19] Quinlan, J. R., "Learning Decision Tree Classifiers", *ACM Computing Surveys*, Vol. 28, No. 1, 1996.
- [20] Robert L. Kirby, A. Rosenfeld, "A Note on the Use of(Gray Level, Local

Average Gray Level) Space as an Aid in Threshold Selection”, IEEE Transactions on System, Man, and Cybernetics, Vol. SMC-9, No. 12, pp. 860-864, December 1979.

[21] S. E. Umbaugh, Computer Vision and Image Process, Prentice Hall, 1998.

[22] T. Pun, “Entropic Thresholding, A New Approach”, CVGIP 16, pp. 210-239, 1981.