

# 國立交通大學

電機學院 IC 設計產業研發碩士班

## 碩士論文

影像同時放大與強化—

利用原始影像分析做外插或內插放大



Simultaneous Image Scaling and Enhancement –

A Pattern-Based Inter-/Extra- polation

研究生：吳思慧

指導教授：莊仁輝 教授

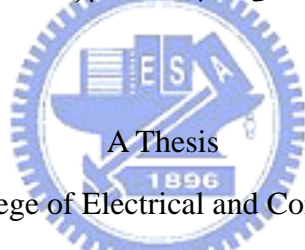
中華民國九十六年一月

影像同時放大與強化—  
利用原始影像分析做外插或內插放大  
Simultaneous Image Scaling and Enhancement –  
A Pattern-Based Inter-/Extra- polation

研究生：吳思慧  
指導教授：莊仁輝

Student : Szu-Hui Wu  
Advisor : Jen-Hui Chuang

國立交通大學  
電機學院 IC 設計產業研發碩士班  
碩士論文



Submitted to College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Industrial Technology R & D Master Program on  
IC Design

January 2006

Hsinchu, Taiwan, Republic of China

中華民國九十六年一月

# 影像同時放大與強化— 利用原始影像分析做外插或內插放大

學生：吳 思 慧

指導教授：莊 仁 輝 博士

國 立 交 通 大 學  
電機學院 IC 設計產業研發碩士班

## 摘要

本論文提出一個可以實作於硬體上的影像放大以及強化方法，一般常見的放大以及強化影像方法，都是先透過內插補點的方法將影像放大之後，再針對放大過後的影像做強化，而我們的方法則是在放大的同時強化影像。另外，不同於其他相關研究中，較沒有考慮到利用原始影像的資訊直接做不限比例放大或強化，我們則是透過分析原始影像的資訊，去放大影像。在分析原始影像之前，我們會先預先定義出邊緣或是角的種類，接著再針對一個小區塊作分析，判斷這個小區塊的原始影像是否符合我們預先定義的種類，再根據分析的結果選擇要使用內插補點或是外插補點的方式。因此本論文中也會探討小區塊的大小，以及預先定義種類的多寡，對影像放大強化造成的影響。

# Simultaneous Image Scaling and Enhancement – A Pattern-Based Inter-/Extra- polation

Student: Szu-Hui Wu

Advisor: Dr. Jen-Hui Chuang

College of Electrical and Computer Engineering  
National Chiao Tung University



## ABSTRACT

The thesis proposes an image scaling and enhancement method which is suitable for hardware implementation. Except for very limited image scaling ratios, most image scaling and enhancement methods first scale the image by interpolation, and then enhance the image without considering image content. With analysis of image content, the proposed method scales and enhances the image simultaneously without the above limitation. For such analysis, different pixel patterns of an object edge or corner are pre-defined for a particular window size. For each pattern, proper interpolation/extrapolation scheme is adopted by the proposed method for simultaneous scaling and enhancement. Effects of choosing different window sizes, and associated pixel patterns, on image scaling and enhancement are also examined in detail in this thesis.

# 誌謝

從進入研究所就讀的第一天開始，我就很期待寫誌謝這一天的來臨。因為在這兩年的求學過程中，對我伸出援手拉我一把的人實在太多了，希望能在最後的時刻好好感謝那些曾經幫助過我的人，沒有你們就沒有今天的思慧。

首先要感謝的是我的爸爸—吳啟周（吳重慶），除了在這一路求學的階段中供給物質上的支援，讓我毫無顧慮的唸書以外，還要感謝他在兩年前獨排眾議全力支持我、鼓勵我繼續攻讀碩士學位，沒有他的支持以及語重心長的一番話，我想我也不會有勇氣面對這一切的挑戰。感謝我的媽媽—呂美玉，從小到大的求學過程中，她始終扮演著避風港的角色，守護著我一路長大，即使我都已經念研究所了，她還是不辭辛勞常常往返交大接送我上下學，準備宵夜就怕我吃不飽，很感謝爸媽對我所付出的一切，讓我可以很安心的完成學業。

另外，還要感謝我的指導老師—莊仁輝教授，在這兩年當中，花了很多心思在我的身上，除了在課業上的指導以外，也時常關心我的身體狀況以及其他瑣碎的小事，讓我倍感溫馨，我真的覺得很幸運，可以跟隨在老師底下學習如何從事研究的工作以及待人處事的方法。也謝謝王才沛老師以及韓欽銓老師百忙中抽空參加我的口試，還有在口試中給我的寶貴建議，讓我受益無窮。

在研究所的生活中首先要特別感謝的就是實驗室的泓宏學長，謝謝你在我程式出問題的時候一行一行的幫我看程式；論文或是報告生不出來的時候，提供我意見；口試前一天熬夜幫我一起修改投影片，這些我都點滴在心頭。另外還要感謝實驗室其他成員—高肇宏學長、阿璞、庭瑋、開國、柏韻、志高、蝦姐、Acer、郁婷、光哥、杯口、怡伶，在這兩年當中的陪伴，尤其是在報告以及課業壓力都很沈重的時候，還好有你們陪我在實驗室玩耍，實驗室的出遊不管是五月的螢火蟲之旅或是八月的環島之旅還是其他活動，都讓我留下很美好的回憶。同時也謝謝 iic93g 的同學們—洪岳琪、蘇彥欽、張華鼎、陳宜寬、蔡宗翰，在我課業上的幫忙，跟你們一起討論讓我覺得唸書是一件快樂的事情。謝謝友達光電的王處

長、謝經理、黃經理以及其他曾經給予我意見的人，在計畫中提供許多寶貴意見，讓我的研究更加完善。謝謝在精神上給我鼓勵以及支持的大學同學以及高中同學—小佑、曼芸、Plan、虹君、小規、歐潔、永嘉、彥伶、阿冠、怡芬、燕芬，有你們不時傳來的溫馨簡訊以及電話鼓勵，讓我在沈重的課業壓力中，可以找到一個出口。

最後要感謝外婆、大舅舅、大舅媽、二舅舅、二舅媽、三舅舅、三舅媽、弟弟以及其他家人在這兩年當中給予的照顧、鼓勵、支持以及肯定，謝謝你們把我當女兒一樣的對待，當媽媽不在家的時候，多虧有你們的照顧，才讓我有家的溫暖。在這一條研究的路上，要感謝的人實在太多，無法在這邊一一列舉，但是，真的很謝謝所有曾經給予我幫助的人，讓我在這兩年當中學習到非常多的寶貴知識以及經驗，也讓我渡過了很愉快且充實的兩年，謝謝大家！



思慧 謹誌

二〇〇七年一月

# 目錄

表目錄.....	vi
第一章 簡介.....	1
1.1 文獻探討.....	2
1.1.1 適合硬體實作的方法.....	3
1.1.2 適合軟體實作的方法.....	4
1.2 影像同時放大及強化流程.....	6
第二章 基本影像放大方法.....	9
2.1 影像內插放大概述.....	9
2.2 基本內插演算法.....	10
2.2.1 Sinc演算法.....	11
2.2.2 Nearest Neighbor演算法.....	13
2.2.3 Linear/Bi-linear演算法.....	13
2.2.4 Cubic/Bi-cubic演算法.....	14
2.3 二維度Cubic內插補點演算法.....	16
第三章 影像放大與強化.....	17
3.1 基本觀念.....	17
3.1.1 影像放大後處理之Hi-Boost強化.....	17
3.1.2 影像放大與強化之內插與外插.....	22
3.2 內插與外插架構介紹.....	27
3.3 二維內插外插之影像放大與強化方法介紹.....	27
3.3.1 判斷 2x2 區域內邊以及角的處理.....	28
3.3.1 判斷 4x2 區域內邊的處理.....	30
3.3.2 判斷 4x4 區域內邊的處理.....	36
第四章 討論.....	51
4.1 分類的完整性.....	51
4.2 邊緣的連續性.....	53
第五章 實驗結果.....	54
5.1 判斷 2x2 區域內邊的實驗結果.....	54
5.2 判斷 4x2 區域內邊的實驗結果.....	59
5.3 判斷 4x4 區域內邊以及角的實驗結果.....	66
第六章 結論與未來的研究方向.....	86
參考文獻.....	87

# 表目錄

表 1 對圖 3.29 中的七個CASES再進一步判斷所要考慮的點及其順序。 .....37





# 圖目錄

圖 1.2	對於原始影像中，變化較劇烈的邊緣，我們以內插外插混合的補點方式來計算影像放大的點，以保持影像邊緣的銳利性。.....	8
圖 2.1	重新取樣。.....	9
圖 2.2	一維度線性內插放大。.....	10
圖 2.3	Sinc 函數。.....	11
圖 2.4	Sinc 函數作二維影像內插放大結果。.....	12
圖 2.5	波紋效應 (Ringing Effect)。.....	12
圖 2.6	Nearest Neighbor 作一維影像內插結果。.....	13
圖 2.7	一維線性放大影像內插結果。.....	14
圖 2.8	一維度 Cubic 內插核心函數。.....	15
圖 2.9	內插像素點與原始影像像素點之位置關係(s)示意圖。.....	15
圖 2.10	利用一維度 Cubic Convolution 作影像內插補點的結果。.....	16
圖 2.11	二維度內插補點示意圖，(a) 原始影像像素點位置，(b) 綠色三角形為垂直方向內插補點的位置，(c) 紅色叉叉為水平方向內插補點的位置。... ..	16
圖 3.1	(a) Bi-Cubic，(b) Bi-Cubic 加上 Hi-Boost 後處理。.....	18
圖 3.2	(a) Bi-Cubic，(b) Bi-Cubic 加上 Hi-Boost 後處理。.....	19
圖 3.3	(a) Bi-Cubic，(b) Bi-Cubic 加上 Hi-Boost 後處理。.....	20
圖 3.4	(a) Bi-Cubic，(b) Bi-Cubic 加上 Hi-Boost 後處理。.....	21
圖 3.5	方法 1：藍點是原本的點，紅點是做外插的點，綠點是做內插的點（放大十倍）。.....	23
圖 3.6	方法 2：藍點是原本的點，紅點是做外插的點，綠點是做內插的點（放大十倍）。.....	23
圖 3.7	原始影像：自然影像。.....	24
圖 3.8	經由第一種內外插交互運算補點方法，將圖 3.7 放大四倍。.....	24
圖 3.9	經由第二種內外插交互運算補點方法，將圖 3.7 放大四倍。.....	25
圖 3.10	原始影像：特殊的測試影像.....	25
圖 3.11	經由第一種內外插交互運算補點方法，將圖 3.10 放大四倍。.....	26
圖 3.12	經由第二種內外插交互運算補點方法，將圖 3.10 放大四倍。.....	26
圖 3.13	影像同時放大以及強化流程圖。.....	27
圖 3.14	小區塊大小為 2×2 時，共有四個參考點。.....	28
圖 3.15	(a)(b)分別為 Case A、B，藍線即為我們定義的邊緣線，(c)~(f)分別為 Cases C、D、E、F，這四種 Cases 都各自定義出三種邊緣，分別標示為藍線、紅線以及綠線，根據需求的不同，可以選擇不同的邊緣做計算。.....	29
圖 3.16	判斷中間四個參考點的流程圖，每個 Level 都是一個 if 判斷條件。..	30

圖 3.17	將圖 3.16 中之 else 部分刪除後之簡化圖。	30
圖 3.18	綠線為邊緣出現的地方 $f(x)$ 。	30
圖 3.19	判斷中間四個參考點的流程圖。	31
圖 3.20	Case A 的判斷樹狀圖。	32
圖 3.21	Case B 的判斷樹狀圖。	32
圖 3.22	Case C 的判斷樹狀圖。	33
圖 3.23	Case D 的判斷樹狀圖。	33
圖 3.24	Case E 的判斷樹狀圖。	34
圖 3.25	Case F 的判斷樹狀圖。	34
圖 3.26	分析原始影像小區塊大小為 $4 \times 2$ 的方法中，預先定義的各種邊緣，在每種邊緣中，綠色的線代表定義邊緣的地方。	35
圖 3.27	(a) 要補的點落在綠線左下方，用 3、4 兩個點的灰階值平均，當作要被補點的灰階值，反之在綠線右上方，用 1、2 兩個點的灰階值平均，當作要被補點的灰階值，(b) 要補的點落在綠線右下方，用 2、3、4 三個點的灰階值平均，當作要被補點的灰階值，反之在綠線左上方，用 1 的點的灰階值，當作要被補點的灰階值，(c) 要補的點落在綠線左上方，用 1、a、c 三個點的灰階值平均，當作要被補點的灰階值，反之在綠線右下方，用 2、3、4 三個點的灰階值平均，當作要被補點的灰階值。	35
圖 3.28	十六個參考點	36
圖 3.29	分析原始影像小區塊大小為 $4 \times 4$ 之七大種類。	37
圖 3.30	(a) 到 (g) 是 Case A 中的七種邊緣，(h) 到 (o) 是 Case A 中的八種角。	37
圖 3.31	(a) 到 (g) 是 Case B 中的七種邊緣，(h) 到 (o) 是 Case B 中的八種角。	38
圖 3.32	(a) 到 (g) 是 Case C 中的七種邊緣，(h) 到 (k) 是 Case C 中的四種角。	39
圖 3.33	(a) 到 (g) 是 Case D 中的七種邊緣，(h) 到 (k) 是 Case D 中的四種角。	39
圖 3.34	(a) 到 (g) 是 Case E 中的七種邊緣，(h) 到 (k) 是 Case E 中的四種角。	40
圖 3.35	(a) 到 (g) 是 Case F 中的七種邊緣，(h) 到 (k) 是 Case F 中的四種角。	40
圖 3.36	(a) 到 (g) 是 Case G 中的十二種角。	41
圖 3.37	定義中間補點區域的四個參考點。	41
圖 3.38	判斷中間四個原始影像參考點的流程圖。	41
圖 3.39	對 Case A* 再進一步判斷是那個種類的邊緣或是角 (共十五種)。	42
圖 3.40	對 Case A.1~Case A.7 再進一步判斷是那個種類的邊緣。	42
圖 3.41	對 Case A.8、Case A.9、Case A.10、Case A.11 再進一步判斷是那個種類	

的角。.....	42
圖 3.42 對 Case A.12、Case A.13、Case A.14、Case A.15 再進一步判斷是那個種類的角。.....	43
圖 3.43 對 Case B*再進一步判斷是那個種類的邊緣或是角（共十五種）。.....	43
圖 3.44 對 Case B.1~Case B.7 再進一步判斷是那個種類的邊緣。.....	43
圖 3.45 對 Case B.8、Case B.9、Case B.10、Case B.11 再進一步判斷是那個種類的角。.....	44
圖 3.46 對 Case B.12、Case B.13、Case B.14、Case B.15 再進一步判斷是那個種類的角。.....	44
圖 3.47 對 Case C*再進一步判斷是那個種類的邊緣或是角（共十一種）。.....	44
圖 3.48 對 Case C.1~Case C.7 再進一步判斷是那個種類的邊緣。.....	45
圖 3.49 對 Case C.8、Case C.9、Case C.10、Case C.11 再進一步判斷是那個種類的角。.....	45
圖 3.50 對 Case D*再進一步判斷是那個種類的邊緣或是角（共十一種）。.....	45
圖 3.51 對 Case D.1~Case D.7 再進一步判斷是那個種類的邊緣。.....	46
圖 3.52 對 Case D.8、Case D.9、Case D.10、Case D.11 再進一步判斷是那個種類的角。.....	46
圖 3.53 對 Case E*再進一步判斷是那個種類的邊緣或是角（共十一種）。.....	46
圖 3.54 對 Case E.1~Case E.7 再進一步判斷是那個種類的邊緣。.....	47
圖 3.55 對 Case E.8、Case E.9、Case E.10、Case E.11 再進一步判斷是那個種類的角。.....	47
圖 3.56 對 Case F*再進一步判斷是那個種類的邊緣或是角（共十一種）。.....	47
圖 3.57 對 Case F.1~Case F.7 再進一步判斷是那個種類的邊緣。.....	48
圖 3.58 對 Case F.8、Case F.9、Case F.10、Case F.11 再進一步判斷是那個種類的角。.....	48
圖 3.59 對 Case G*再進一步判斷是那個種類的角（共十二種）。.....	48
圖 3.60 對 Case G.1~Case G.6 再進一步判斷是那個種類的角。.....	49
圖 3.61 對 Case G.7~Case G.12 再進一步判斷是那個種類的角。.....	49
圖 3.62 綠線為邊緣出現的地方 $f(x)$ 。.....	50
圖 4.1 兩群各包含兩個參考點。.....	51
圖 4.2 一群包含有三個參考點，另一群只有一個參考點。.....	51
圖 5.1 (a) 是原始影像為像素 268×305 的三角形，(b) 是(a)經圖 3.15 中的藍線放大到像素為 800×800 的三角形，(c) 是(a)經圖 3.15 中的紅線放大到像素為 800×800 的三角形，(d) 是 (a) 經圖 3.15 中的綠線放大到像素為 800×800 的三角形。.....	55
圖 5.2 (a) 是原始影像為像素 155×116 的建築物，(b) 是(a)經由圖 3.15 中的藍線放大到像素為 800×800 的建築物，(c) 是經由圖 3.15 中的紅線放大到像素為 800×800 的建築物，(d) 是(a)經由圖 3.15 中的綠線放大到像素為 800×800	

的建築物。.....	57
圖 5.3 由 800×600 放大到 1280×1024，(a) 原始影像，(b) 以 4×2 影像區塊作分析，所得之部分影像。.....	60
圖 5.4 由 800×600 放大到 1280×1024，(a) 原始影像，(b) 以 4×2 影像區塊作分析，所得之影像，其中(b)在角的地方會比較模糊。.....	62
圖 5.5 由 800×600 放大到 1280×1024，(a) 原始影像，(b) 以 4×2 影像區塊作分析，所得之部分影像，(c) 將預先定義中水平方向定義的種類減少成與垂直方向種類數目一樣多。.....	62
圖 5.6 由 800×600 放大到 1280×1024，(a) 原始影像，(b) 以 4×2 影像區塊作分析，所得之部分影像。.....	64
圖 5.7 將中間四個點分為兩個群組。.....	66
圖 5.8 內部臨界值以及外部臨界值的示意圖。.....	66
圖 5.9 由 268×305 放大到 1280×1024，(a) 原始影像，(b) 經由 Bi-Cubic 放大影像，(c) 以 4×4 影像區塊作分析，所得之影像（內部臨界值=10，外部臨界值=250）。.....	68
圖 5.10 由 330×300 放大到 1280×1024，(a) 原始影像，(b) 經由 Bi-Cubic 放大影像，(c) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250），(d) 外插計算的部分，(e) 對圖(b)的 Edge Extraction，(f) 對圖(c)的 Edge Extraction。.....	69
圖 5.11 由 800×600 放大到 1280×1024，(a) 經由 Bi-Cubic 放大影像，(b) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250）。.....	73
圖 5.12 由 800×600 放大到 1280×1024，(a) 經由 Bi-Cubic 放大影像，(b) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250），(c) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=30，外部臨界值=120）。.....	75
圖 5.13 由 417×313 放大到 1280×1024，(a) 原始影像，(b) 以 4×4 影像區塊作分析，所得之影像做外插計算的部分，(c) 經由 Bi-Cubic 放大影像，(d) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250）。.....	78
圖 5.14 由 463×358 放大到 1280×1024，(a) 原始影像，(b) 以 4×4 影像區塊作分析，所得之影像做外插計算的部分，(c) 經由 Bi-cubic 放大影像，(d) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=30，外部臨界值=120）。.....	80
圖 5.15 由 320×214 放大到 1280×1024，(A) 原始影像，(B) 經由 BI-CUBIC 放大影像，(C) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=20，外部臨界值=100），(D) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=30，外部臨界值=40）。.....	82

# 第一章 簡介

由於近年來在顯示技術及製程上的進步，使得新型態的液晶或是電漿電視得以突破傳統真空管（CRT）顯示器的限制，能以更為合理的成本，獲得更大的顯示面積，與更高的畫面解析度。在此同時，影像的來源卻可能只含有較低解析度的資訊。最常被提出來的一個例子便是：當需要在 HDTV 顯示器上播放傳統 NTSC 標準的影像訊號時，由於 HDTV 的標準影像格式解析度達  $1920 \times 1080$  像素，而 NTSC 標準的影像格式解析度只有  $648 \times 486$  像素，其尺度大小相差約三倍，此時便得利用一些方法將影像放大。

在影像放大過程中，不同的類型的影像內容適合的放大方法也不盡相同，因此，即使是同一種放大方法，應用在不同類型的影像上，有可能會產生不同的效果。除此之外，放大後的影像品質也是我們需要考量的部分，當放大倍率越大時，若是只經由一般傳統的放大方法，例如：Nearest Neighbor、Bi-Linear、Bi-Cubic，去放大影像的話，會造成失真或是模糊的現象越嚴重。為了解決上述放大過程中產生的問題，本篇論文是以分析原始影像的局部特性，選擇做外插或是內插運算以進行影像放大，並同時完成強化影像作為我們研究的主題。

由於本篇論文所使用的影像放大強化方法，是透過分析原始影像的小區塊的內容，選擇做外插計算，或是內插計算，因此，在分析原始影像的過程中，我們所選擇的小區塊（Window）大小，會影響到我們分析後放大的結果，所以在本篇論文中，除了介紹內插以及外插混合使用的放大方法外，也會針對選擇不同大小影像區塊之效果做探討。除此之外，在分析的過程中，我們會利用到兩個臨界值，藉由調整臨界值，可以增減放大影像時外插計算的適用範圍以應用在不同類型的影像內容上，例如：一般自然影像需要比較多的內插計算，以得到人眼視覺上較佳的視覺效果；含有較多文字的影像則需要較多的外插計算，以得到人眼視覺上較銳利的效果，這個部分將於後面章節做介紹。由於本篇論文所使用的影像

放大強化方法，運算非常簡單，也只需要使用到有限的記憶體，因此適合於需要考量到記憶體以及運算量的硬體上實作。

本篇論文全文共分為六章，第一章，主要是文獻探討，我們將對於相關的文獻作大略的說明；第二章主要是討論影像放大的基本問題，包含不同放大方法的選擇；第三章則是影像同時放大以及強化的研究，著重在影像邊緣的判斷與內插外插作法的說明；第四章將討論演算法中，預先定義的分類之完整性以及放大後影像邊緣的連續性；第五章為各種不同大小視窗的實驗結果；第六章為結論以及本研究未來的研究方向之探討。

## 1.1 文獻探討

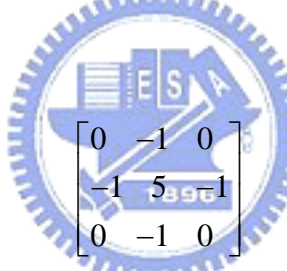
在影像放大銳利化的文獻探討中，我們將之分成「適合硬體實作」的方法與「適合軟體實作」的方法兩個部分。關於「適合硬體實作」部分，共參考了 11 篇相關的文獻，因為硬體實作需要考量記憶體的大小，以及運算的複雜度，因此此類的文獻中所探討的方法都是比較簡單放大的方法，在適合硬體實作部分的文獻探討又可以再細分為三種不同放大影像的方法：一般傳統利用核心函數的放大方法（無強化影像，單純放大）、先作基本放大再經過遮罩 (Mask) 處理（強化影像）、分析原始影像的局部特性再據以採用不同的放大方法（同時放大以及強化影像）。

而對於「適合軟體實作」的部分，則共有 11 篇參考文獻的概述，在此類的文獻中所探討的方法運算比較複雜，也不需要考量記憶體的方面的問題，有些方法甚至需要使用大量的 Training Set，因此經由此類方法放大過後的影像效果會比硬體實作方法放大過後的影像效果好很多。在適合軟體實作部分的文獻方面可以在細分為五種不同放大強化影像的方法：視覺上的幾何方法、機率混合模型 (Probabilistic Mixture Model) 作超解析度放大的方法、求解偏微分方程式的方法、解出轉換矩陣方式求解的方法以及以學習方法為主的超解析度方法。

### 1.1.1 適合硬體實作的方法

一般將影像放大的軟體或是硬體，最常使用Sinc、Bi-Linear [1]、Bi-Cubic [2]、Nearest Neighbor等作法將影像放大，其基本原理皆是利用不同的核心函數（Kernel Function）來對影像作放大，選擇不同的核心函數，即可得到不同的影像縮放效果。在Lanczos [3]的這篇論文中，是簡化Sinc核心函數的實作方法，而Mitchell-Netravali [4]、[5]則是採用Bi-Cubic廣義形式的核心函數；這些透過使用核心函數將影像放大的方法，由於較為基本且簡單，所以常被應用於軟、硬體影像放大的實作，但是這類的方法只有單純對影像做放大，而沒有考慮到影像的銳利化。

因此，為了增加放大後影像的銳利程度，傳統上影像處理的方法是先將影像做基本的放大過後，再透過一些濾波器的建構，如 [6]中所提到的其中一種強化高頻的的遮罩（Mask）：


$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

來強化放大後影像的銳利度，但此類的濾波處理，並沒有考慮到原始影像內容的資訊，因此銳利度處理的效果無法因原始影像內容的不同，作適切的調整，在強化影像的同時，也有可能同時強化了雜訊。因此，在論文 [7] 裡頭，提出了針對每個像素附近結構的不同，在影像的邊緣發生處，依邊緣的方向做內插。這樣做最明顯的好處，便是可以消滅鋸齒狀不自然現象發生的機會；Su與Willis [8]也利用了類似的想法，再加上他們所提出將影像三角化的方式，使得整體的運算時間大幅加速，達到近似於Bi-Linear方法的速度。在論文 [9] 中，根據附近較多的點，找出可能的邊緣方向，去內插出邊緣，讓決定內插的點有比較多的一致性，而且由於是根據附近資料的變異程度，來決定臨界值，可以在當下就馬上計算出來，因此會有更廣泛的應用，另一個優點就是在硬體複雜度最低的情況下，這個方法有最好的影像放大品質；而WinScale的方法 [10]，則是利用計算面積比例大小

的方式作影像的縮放，透過已知像素值的點被未知像素值的點覆蓋的面積大小，按比例去計算被放大影像中新的像素值，這個方法可以保有較高頻率的特徵，因而比Bi-Linear有更好的影像品質。由於此類的方法有考慮附近點的結構再做影像的放大，因此具有放大以及銳利化的效果。

除了影像放大之外，關於放大後影像品質的評比，則如 [11] 中描述，有透過客觀數據以及主觀人眼的兩種評比方式，但是人眼的評比與數據的評比結果可能會不一致，因此一般對於評比的結果來說，人眼評比的方法較具有公信力。另外，此篇論文也提到為了使放大之後的影像看起來比較銳利，增加高頻的資訊是其中的一個方法，在這篇論文裡面，一共比較了四個放大銳利化的演算法。其中由Tegenbosch *et al.*所提的第一個方法是經由一些簡單的濾波器，如Peaking或是Luminance Transient Improvement (LTI) 強化影像；由Kondo *et al.*所提的第二個方法則是將影像中的每個小區塊，經由濾波器放大以及銳利化，其中每個濾波器都可以利用訓練 (Training) 的方法找到係數，再將係數對應到不同的濾波器Peaking或是LTI上；由Atkins *et al.*所提的第三個方法是將第二種方法再延伸，這個方法是透過更多不同的濾波器，將影像放大以及銳利化；由Li and Orchard所提的第四個方法，是先利用解析度低的影像，經由LMSE-Fit內插出低解析度影像的Pattern，再利用這些係數去內插出解析度高的影像。在實驗結果上來看，第二個方法會有最低的MSE，而最後一個方法，在影像頻率比較高的情況下會有缺點，若以主觀的視覺效果來看，則第一個方法有較好的表現。因此，可以得知數據上的評比與人眼上的評比，並不一定會一致。在本篇論文中實驗結果就是採用人眼視覺為主要評比標準。

## 1.1.2 適合軟體實作的方法

除了上一小節所描述較基本的影像放大方法之外，還有比較複雜，只適合軟體實作的方法，此類方法都是可以同時放大以及強化的影像放大方法。如採用視



覺上的幾何來取代使用核心函數放大影像的方式 [12]，這個方法能在放大影像的過程中保持影像的精確度，且放大後的邊緣看起來較平滑。而在論文 [13] 中，是以機率混合模型 (Probabilistic Mixture Model) 作超解析度放大的方法。有別於以簡單濾波器作放大處理的作法，Storkey 所提出的混合機率法，則是由數個潛藏點 (Latent Nodes) 混合計算而得出高解析度的點，而每個潛藏點則可視為對應到一塊同質的影像區域，且這些潛藏點的機率模型，是以信息傳遞 (Belief Propagation) 的方法計算得出。透過這個方法所作出的放大影像，其效果優於以簡單濾波器處理的方法，且放大後的影像與真實大張的影像相比較，有較小的平方誤差，放大後的影像紋路也以較好的方式呈現。

在 [14]、[15] 的研究中，則是利用求解偏微分方程式的方法，來改善影像放大後邊緣出現鋸齒狀的情況。其方法大致是先在放大後的影像上，建立格子點，並依照影像內容 (如邊緣線分佈)，建構初始等高線與梯度函數，並驅動偏微方程求解的程序，以得出新的等高線分佈。經由求解偏微分方程式，修正過後的影像，會使得原本影像出現鋸齒的地方，藉由移動初始的等高線到新的等高線，而變得較為平滑，但由於此類方法需要求解偏微分方程式，故計算量頗大。

此外，對於影像放大比例很高的問題，又稱為超解析度問題。早期關於這方面的研究大多是從影像重建 (Image Reconstruction) 的觀點出發 [16]，此類做法，是先假設存在一張真實的高解析影像  $H$ ，經過  $L = TH + Z$  的運算得到我們所見到的低解析度影像  $L$ ，其中，Convolution 及 Down-Sampling 運算轉換稱之為  $T$ ， $Z$  為雜訊或誤差向量，且在 [16] 中假設  $T$  是較簡單的線性轉換，以解出轉換矩陣的方式求解。除了上述的方法外，還有利用最大化相似區間 (Maximum Likelihood) 求解，以得到一近似的高解析度影像  $H'$ ，Irani 等人在 [17] 中所提出的方法，即可快速求得這樣的  $H'$ 。但由於透過這樣的模型求得的解，本身可能包含無限多組的解，所以即使我們求得的  $H'$  在數值上正確，但對於人眼視覺來說卻可能會有相當不自然的感覺，如鋸齒狀及光暈的出現。對此，改進的模型是對於所要求得的  $H$  給予一先備知識 (Prior)，經由最大化事後機率 (Maximum

Aposteriori) 的計算，就可以重建出一張較為自然的影像。

為了想要得到更自然的影像，有些人則提出了以學習方法為主的超解析度方法 [18]、[19]、[20]、[21]、[22]，這類研究大都是先將低解析度影像區塊 (Image Patch) 及相對應高解析度影像區塊的資料庫收集起來，之後若是想將  $L$  的解析度放大時，就先將  $L$  切割成小塊的區塊，然後在從之前建構的資料庫中，找出最接近的低解析度區塊，再將其相對應的高解析度區塊取回，組成一高解析度的影像。這類方法的優點在於所得的高解析度影像大多較為自然，且可以在只有單張低解析度影像時放大到較大的倍率，其缺點則在於資料庫的內容與收集資料庫的時間成本以及應用範圍有關。

除了上述各種軟體上放大影像的做法以外，在利用軟體從事影像的銳利化的研究方面，可以利用非線性的方法，如論文 [12] 所提出，先利用 Gabor Filters 計算影像邊緣的方向 (Texture Orientation Map)，然後利用先前訓練所得到的資訊沿著邊緣方向做內插，這個方法可以得到比較銳利的邊緣，且可以減緩波紋效應，得到比較好的 PSNR。又如在 [20] 和 [22] 的方法中，皆提出在放大影像的過程中，將內插放大的影像視為低頻的部分，然後對於比較平滑的影像，依照影像的內容，加入高頻的資訊，來強化影像的可視效果。而這些高頻的資訊，則是依照低頻影像內容所推論出來；同時在高頻與低頻影像合成時，[22] 的方法也考慮了線條平順性的限制，以得出較好的強化結果。儘管上述的諸多方法，不適用於硬體的實作，而多屬於軟體影像處理的研究結果，但其研究所得到的影像處理效能，卻是未來發展趨勢的重要指標。

## 1.2 影像同時放大及強化流程

本論文的主要研究內容，在於探討將不同大小的輸入影像，同時放大以及強化成不同解析度的影像。影像放大的時候，可以利用兩種主要的方法，一種是內插補點法，一種是外插補點法。使用內插補點法放大影像的話，可以保持影像的

平滑度，而使用外插補點法放大影像的話，可以保持邊緣的銳利度，所以我們希望可以結合這兩種方法，讓放大後的影像能得到較好的品質。另外，由於不需要複雜的運算且只需要使用有限的記憶體，因此我們所提出的方法適合於硬體上的實作。

這個方法之基本觀念我們以下例來作說明：若是要將一維度的影像放大，會先分析一條 Scan Line 中六個點的灰階值的分佈，去推測中間兩點之間灰階值的變化，分析的結果可能會有兩種狀況，一種是變化較平緩的，另一種是變化較劇烈的，之後，再視不同的狀況選擇做內插補點或是外插補點。如圖 1.1，若是附近幾個已知點的灰階值呈線性變化，則我們判斷這兩點中間的變化是很平緩的，因此直接在中間的兩個點之間做內插補點（例如：Bi-Linear、Bi-Cubic）；如果左邊附近幾個已知點灰階值很相近，右邊附近幾個已知點的灰階值也很相近，但左右兩群已知點的灰階值，有很劇烈的變化（圖 1.2），則我們判斷這兩點中間是邊緣，所以先在兩點中間做部份的外插補點之後，再做內插補點，內外插補點的比例，可以依照原始影像內容的不同做適當的調整。因此，本篇論文所研究的影像放大方法，就是以上述一維度內插補點以及外插補點交互使用的基本觀念，推廣到二維度去做影像的放大。

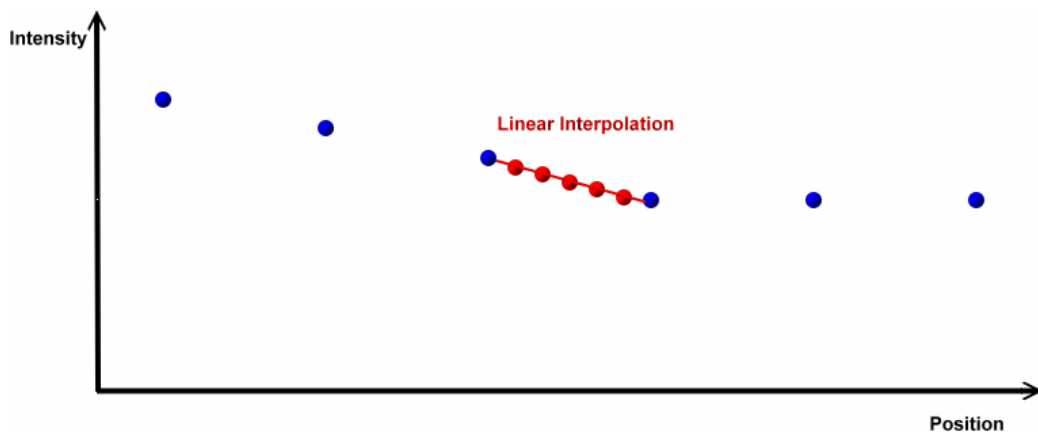


圖 1.1 變化較平滑的影像，我們以線性內插補點的方式。

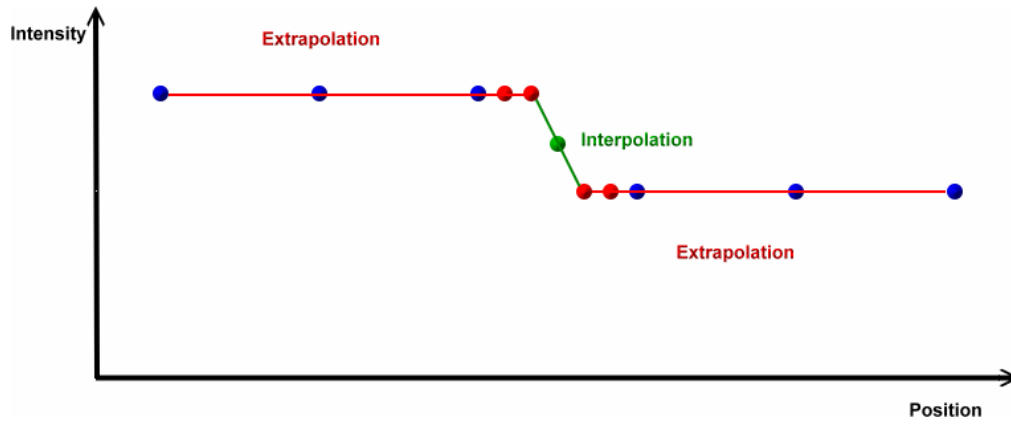


圖 1.2 對於原始影像中，變化較劇烈的邊緣，我們以內插外插混合的補點方式來計算影像放大的點，以保持影像邊緣的銳利性。



## 第二章 基本影像放大方法

本篇論文中，影像同時放大以及強化的演算法是使用內插與外插混合運算，得到新的放大強化影像，其中內插補點的運算，有許多不同的演算法，因此，本章節主要是介紹幾種在目前最廣為大家所熟知的內插補點演算法。在此，我們以內插補點原理與演算法為討論的主題，介紹幾種內插補點演算法以及其實驗結果，最後選擇其中一種內插補點方法，作為後面章節內插與外插混合運算方法中內插補點的運算。

### 2.1 影像內插放大概述

當一張影像放大至一個更大尺寸的影像時，我們必須利用原始點的資訊，計算出新增加出來點的像素值。我們以圖 2.1 作說明，左邊的函數  $f(t)$  代表的是原始小張的影像，右邊的函數  $g(t)$  代表的是密度放大三倍後的新影像，即  $g(t) = f\left(\frac{t}{3}\right)$ ，右圖實線的點為原始影像已知的像素值，右圖虛線的點為影像放大過後必須利用已知點像素值計算出新像素值的未知點。

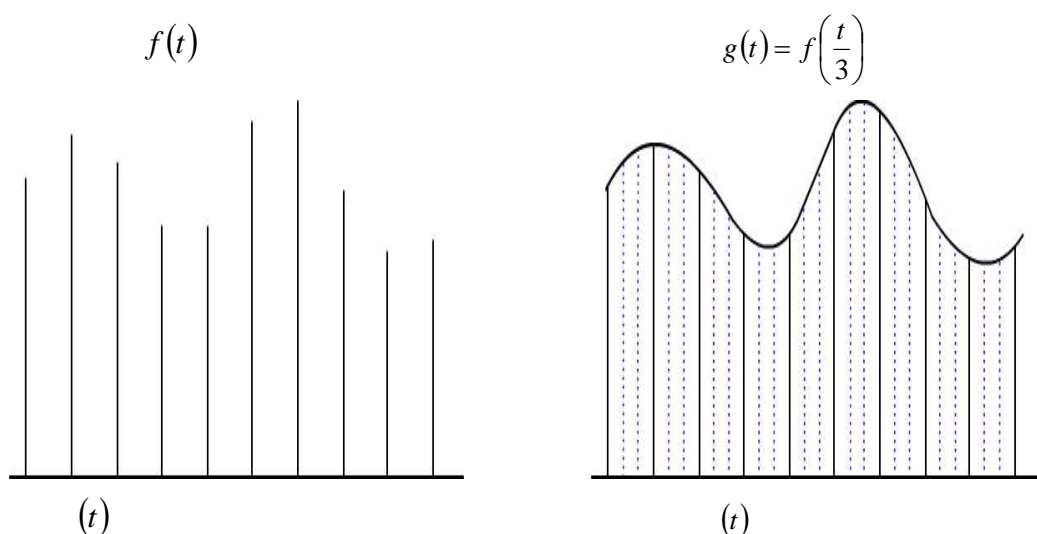


圖 2.1 重新取樣。

理論上，若是  $g(t)$  函數的頻寬是有限的，且取樣點的密度足夠，就可以完整的重建，但是需要利用到附近無限多的已知點去做理想的重建。實際上，在時域中的計算，我們做不到這樣的效果。因此，我們只能利用未知點附近左右的幾個已知點，去計算未知點的像素值。

在計算未知點的內插補點運算中，未知點與離未知點越近的已知點像素值應該是越相似的，因此，在計算未知點的像素值時，越接近的已知點將會乘上較大的權重值。如圖 2.2，若是要計算未知點  $X$  的像素值，首先要將已知點  $X_{k+1}$  的像素值乘上其權重值就會得到紅色點，已知點  $X_k$  的像素值乘上其權重值就會得到綠色點，再將紅色以及綠色兩個點相加起來，即為黑色點，也就是未知點  $X$  的像素值，在這個線性內插的例子中，求得權重值的三角形函數就是影像放大的核心函數。

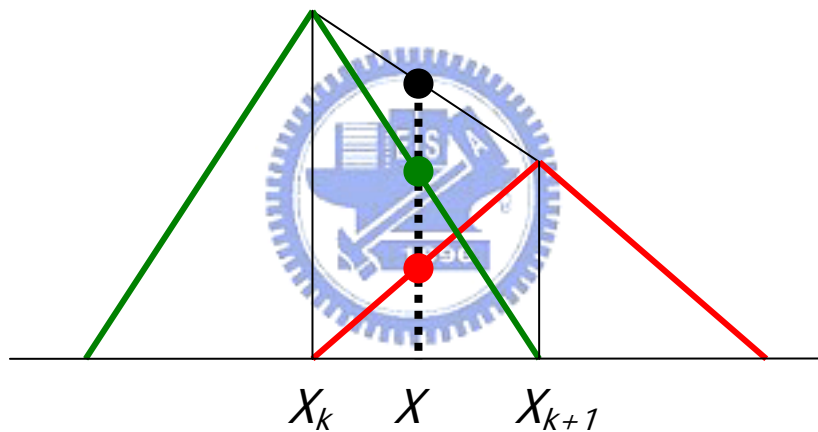


圖 2.2 一維度線性內插放大。

## 2.2 基本內插演算法

內插補點演算法中有幾種常見的核心函數，如 Sinc、Nearest Neighbor、Bi-Linear、Bi-Cubic。因此在本節中，會介紹以上這四種演算法的主要內容，並且討論其實驗結果。

## 2.2.1 Sinc 演算法

在信號分析的理論中，透過 Sinc 函數作內插是一個理想的內插補點方式。但運用到影像放大處理時，則是將影像看成二維度的信號，分別從水平方向及垂直方向進行內插補點。為了說明上的方便，以下我們將先利用一維度的信號來解釋 Sinc 內插補點方法。

Sinc 內插補點是利用 Sinc 函數直接在原本的空間上做信號的內插計算，Sinc 函數的定義如下：

$$\text{sinc}(s) = \begin{cases} 1, & s = 0 \\ \frac{\sin(\pi s)}{\pi s}, & s \neq 0 \end{cases}, \quad (1)$$

從上式我們可以看出 Sinc 函數在除了  $s=0$  以外的整數點上值均為 0，而如圖 2.3 其變化的強度離  $s=0$  愈遠有愈來愈小的趨勢。

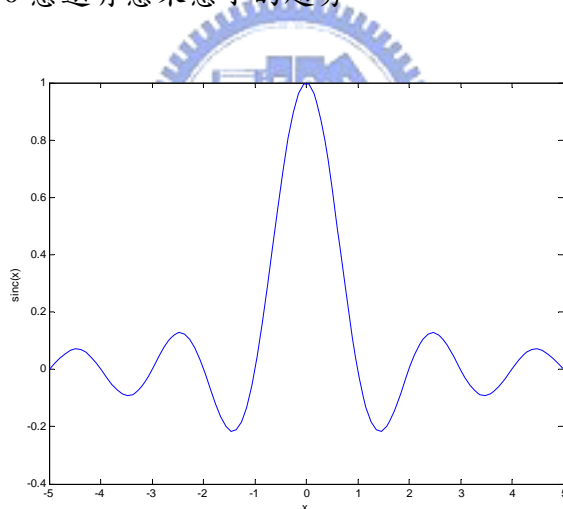


圖 2.3 Sinc 函數。

利用 Sinc 函數來做內插時，我們一次看一個新的取樣點  $x'$ ，以其為原點，將每一個原有取樣點  $x_n$  對於  $x'$  的相對位置  $(x_n - x')$ ，代入 Sinc 函數算出  $x_n$  對  $x'$  的影響再乘上  $f$  在  $x_n$  上的值後相加，所得的值  $\sum_{n=-\infty}^{\infty} f(x_n) \text{sinc}(x_n - x')$  即為  $x'$  上的內插補點值。透過 Sinc 核心函數作影像放大的處理，理論上是一個最佳的選擇，但由於 Sinc 函數具有波動的性質，因此放大的影像可能產生波浪狀的不自然影

像，稱之為波紋效應（Ringing Effect）。如圖 2.4 以及圖 2.5 所示，在二維度的放大影像中，我們可以清楚的看到利用 Sinc 函數做為內插補點之核心函數所造成的波紋效應。

由於 Sinc 函數在時域上涵蓋無限大的區域，因此在硬體上仍難以實現。許多其他的內插補點法的核心函數便採用不同的函數來取代 Sinc 函數，常見的有 Nearest Neighbor、Bi-Linear、Bi-Cubic 內插補點法等等，我們將在以下的小節中作討論。



圖 2.4 Sinc 函數作二維影像內插放大結果。



圖 2.5 波紋效應（Ringing Effect）。



## 2.2.2 Nearest Neighbor 演算法

最鄰近內插法 (Nearest Neighbor Interpolation)，是最簡單的內插補點方法，依據內插影像座標系的補點位置，找出原始影像中最接近點的像素值，作為內插補點之像素值。其作法為：(1) 找出放大後影像像素座標與原始影像座標之關係，(2) 以原始座標之最鄰近像素灰階值為放大後影像像素灰階值。這種內插補點方法並不是真的計算影像的像素值，而只是複製原始影像的像素值，且由於這個內插補點方法並不會改變影像的像素值，所以原始影像的影像特徵更容易被保留。

對於一維度的最近鄰內插演算法來說，需要利用到原始影像中兩個參考點的資訊，去計算出新影像的像素值。同理，對於二維最近鄰內插演算法來說，需要參考原始影像中四個參考點，去計算出新影像的像素值。以一維度的 Nearest Neighbor 內插放大方法而言，其核心演算法  $u(s)$  數學式可表示為：

$$u(s) = \begin{cases} 0 & |s| > 0.5 \\ 1 & |s| \leq 0.5 \end{cases} \quad (2)$$

其中  $s$  為需被內插出的像素點與原始影像格點的距離。其一維影像放大的結果如圖 2.6 所示。



圖 2.6 Nearest Neighbor 作一維影像內插結果。

## 2.2.3 Linear/Bi-linear 演算法

線性內插 (Linear Interpolation) 演算法是將原始影像的兩個點的像素值

(Intensity)，乘以其權重值來計算出內插補點之像素值。雙線性內插演算法 (Bi-Linear Interpolation) 即為兩個線性內插補點方式分別先後應用在垂直及水平兩個方向。以一維度的線性內插補點演算法 (垂直或水平方向) 而言，其核心演算法  $u(s)$  數學式可表示為：

$$u(s) = \begin{cases} 0 & |s| \geq 1 \\ 1 - |s| & |s| < 1 \end{cases} \quad (3)$$

其中  $s$  為需被內插出的像素點與原始影像格點距離。其一維影像內插補點的結果如圖 2.7 所示。



圖 2.7 一維線性放大影像內插結果。

## 2.2.4 Cubic/Bi-cubic 演算法

立方內插 (Cubic Convolution Interpolation) 演算法是將原始影像中四個參考點的像素值 (Intensity)，乘以其權重值來計算出內插放大點之像素值。以一維度內插補點演算法而言，其核心函數  $u(s)$  數學式可表示為：

$$u(s) = \begin{cases} \frac{3}{2}|s|^3 - \frac{5}{2}|s|^2 + 1 & 0 < |s| < 1 \\ -\frac{1}{2}|s|^3 + \frac{5}{2}|s|^2 - 4|s| + 2 & 1 < |s| < 2 \\ 0 & 2 < |s| \end{cases} \quad (4)$$

其中  $s$  為需被內插出的像素點與原始影像格點距離，其核心函數的曲線則如圖 2.8 所示。

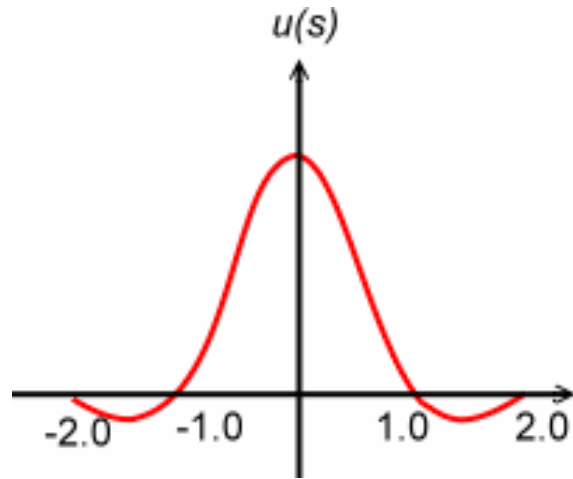


圖 2.8 一維度 Cubic 內插核心函數。

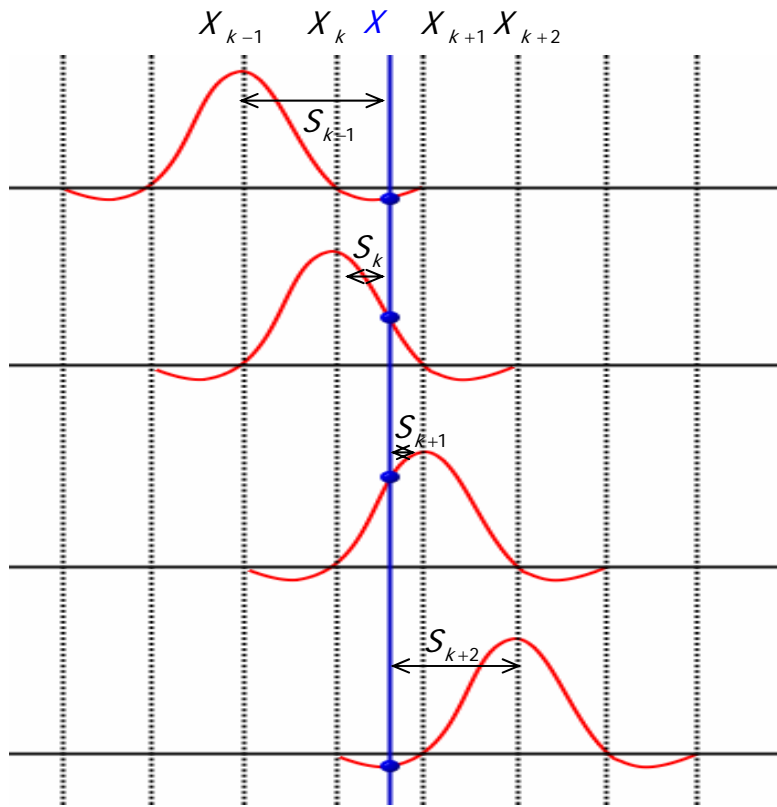


圖 2.9 內插像素點與原始影像像素點之位置關係(s)示意圖。

在圖 2.9 中，我們則是以圖示的方式，說明在一維度的 Cubic 內插補點運算中，相鄰的原始影像像素點  $x_{k-1}$ 、 $x_k$ 、 $x_{k+1}$ 、 $x_{k+2}$  的權重值。之後再透過式(5)的 Weighted Convolution 運算，我們便可以內插出新影像點  $x$  的像素值，其一維度影像放大的結果如圖 2.10 所示。

$$I(x) = I(x_{k-1})w(S_{k-1}) + I(x_k)w(S_k) + I(x_{k+1})w(S_{k+1}) + I(x_{k+2})w(S_{k+2}) \quad (5)$$



圖 2.10 利用一維度 Cubic Convolution 作影像內插補點的結果。

## 2.3 二維度 Cubic 內插補點演算法

一般而言，由於Cubic演算法應用到影像放大的人眼視覺效果較佳，因此，我們選擇了Cubic演算法作為本論文中內插補點所使用的演算法。應用到本論文中內插補點的部分時，我們必須將一維度的Cubic演算法推廣到二維度的Bi-Cubic內插補點演算法，原本在一維度的Cubic演算法只要利用四個原始影像參考點，但在二維度Bi-Cubic內插補點演算法中，必須參考到圖 2.11(a)的十六個原始影像參考點，根據 [24] 中所提到的方法，我們先如圖 2.11(b)內插補點出垂直方向的影像，再如圖 2.11(c)內插補點出水平方向的影像，就可以得到二維度的Bi-Cubic內插補點影像。

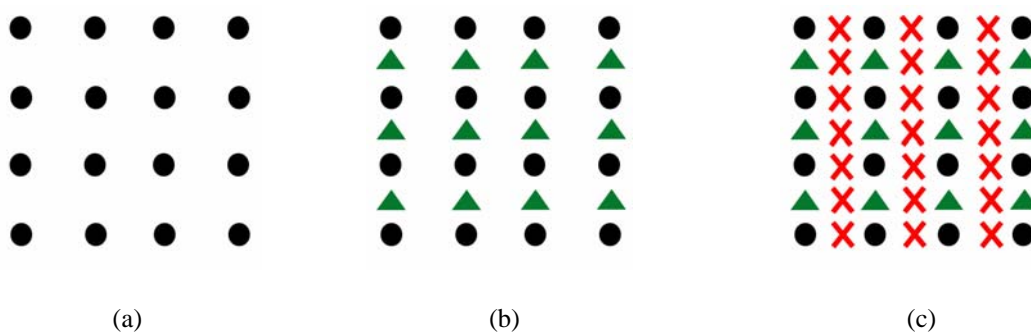


圖 2.11 二維度內插補點示意圖，(a) 原始影像像素點位置，(b) 綠色三角形為垂直方向內插補點的位置，(c) 紅色叉叉為水平方向內插補點的位置。

# 第三章 影像放大與強化

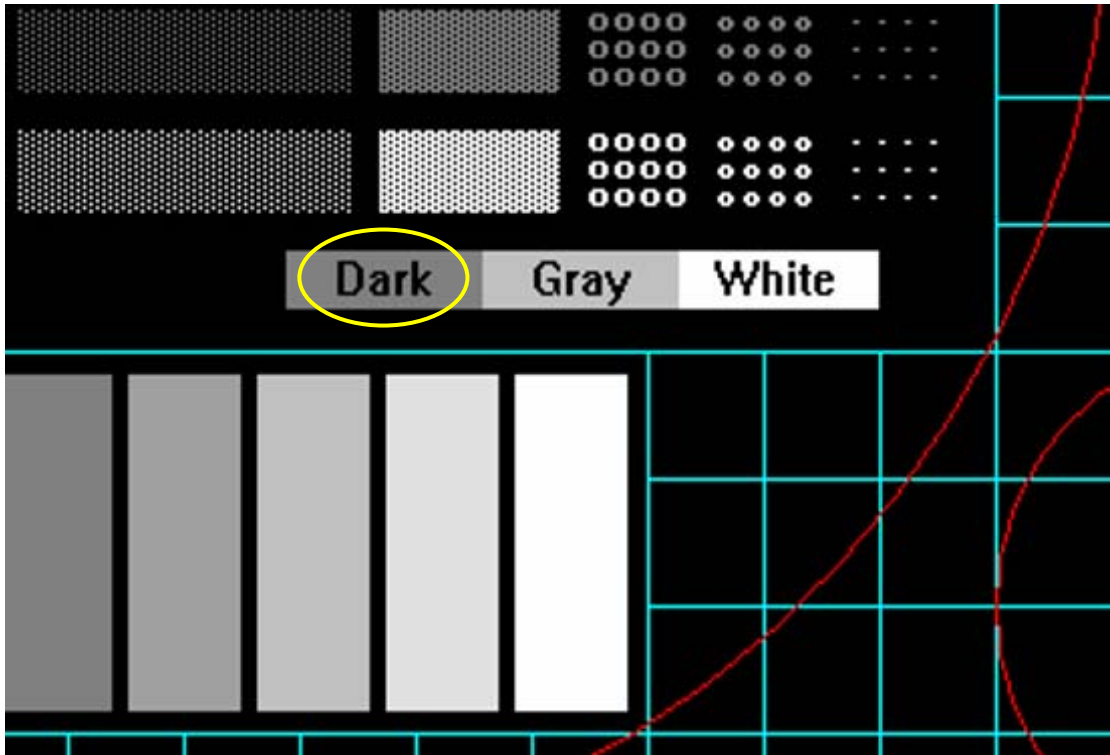
## 3.1 基本觀念

一般影像放大與強化的做法，通常是先利用一般常見的內插補點方法，例如：Nearest Neighbor、Bi-Linear、Bi-Cubic，將影像做基本的放大之後，再經由其他的演算法，或是經由一些可以強化高頻的濾波器，做影像的強化；而我們所提出的方法是經由分析原始影像，再決定作內插或是外插補點，同時放大以及強化影像。因此，在這個章節中一開始會先介紹目前常見到的影像放大強化的方法與我們所提出的方法之基本觀念，之後會詳細介紹本論文中影像放大以及強化方法，並且提出幾種不同的方法做為比較。

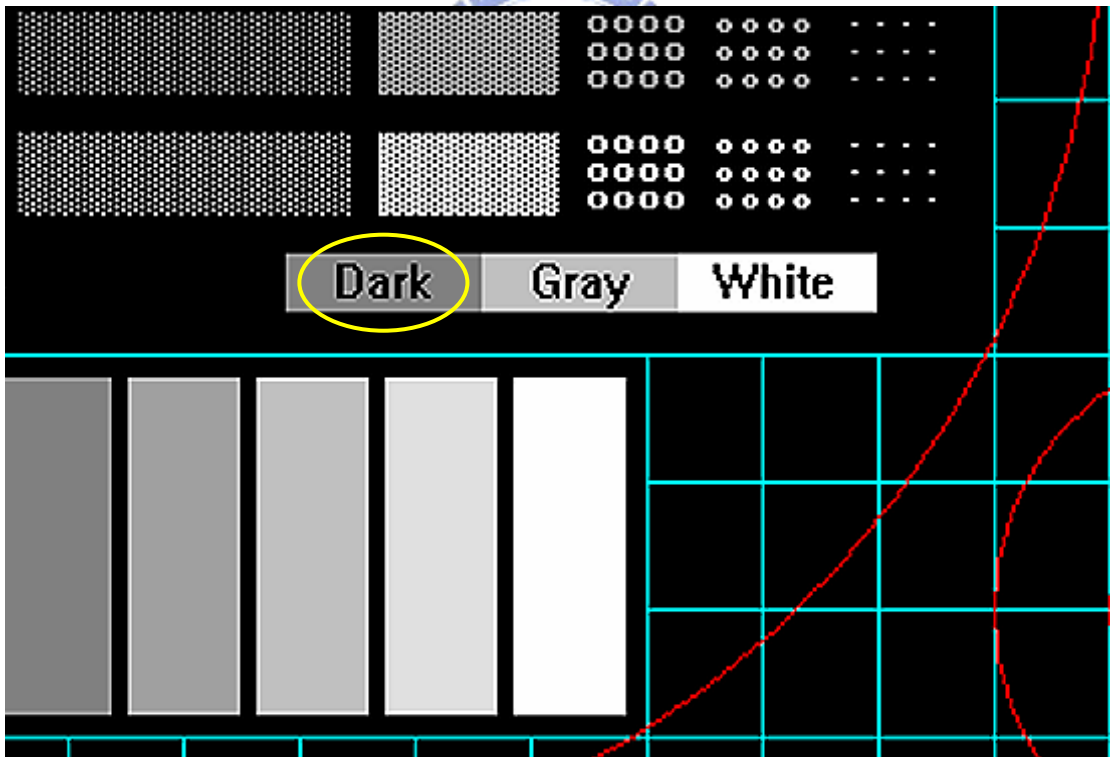
### 3.1.1 影像放大後處理之 Hi-Boost 強化

在 [11] 中提到，有一種後處理強化高頻的方法，是先經過一個基本的演算法的放大之後，再加上Hi-Boost的後處理，這類的方法可以得到一個在人眼視覺效果上比較良好的結果。因此，這個章節中是利用前面章節所提到的Bi-Cubic演算法做基本的放大之後，再加上一個比較簡單的後處理方法—Hi-Boost後處理。實驗結果如圖 3.1、圖 3.2、圖 3.3 以及圖 3.4，其中圖 3.1(a)、圖 3.2(a)、圖 3.3(a) 以及圖 3.4(a)都是只有經過Bi-Cubic放大的結果，而圖 3.1(b)、圖 3.2(b)、圖 3.3(b) 以及圖 3.4(b)是先經過Bi-Cubic放大之後再經過一個Filter的後處理，而這邊所使用的Filter Mask是根據 [6] 中所提到的Hi-Boost Filter Mask：

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



(a)

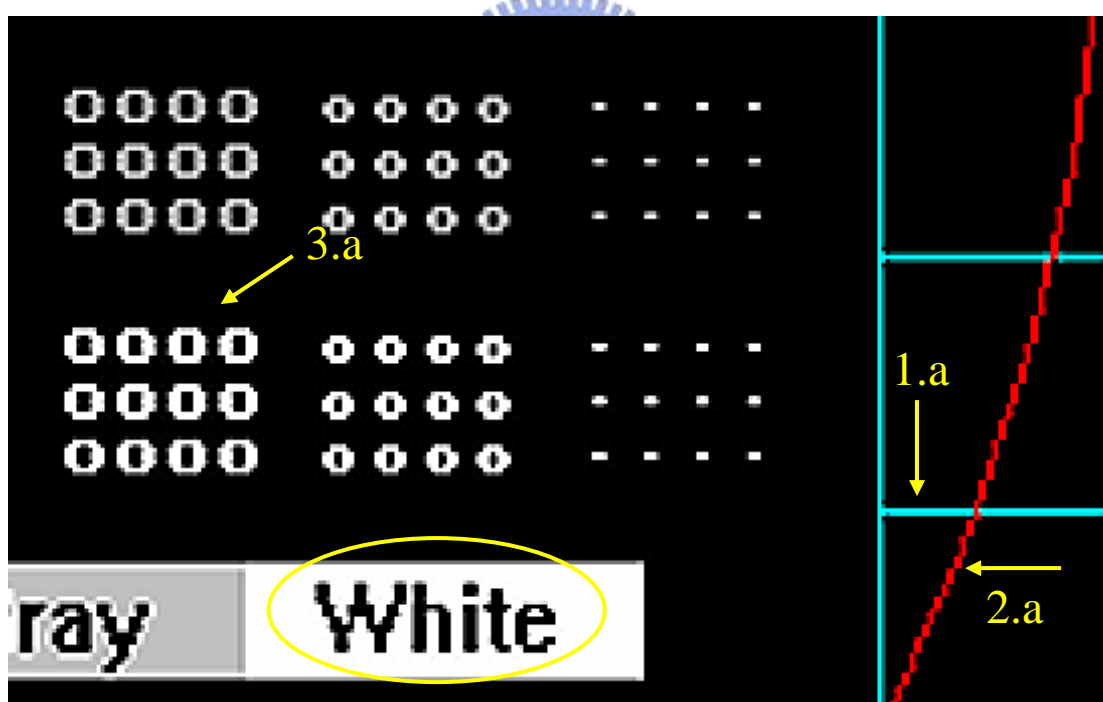


(b)

圖 3.1 (a) Bi-Cubic, (b) Bi-Cubic 加上 Hi-Boost 後處理。

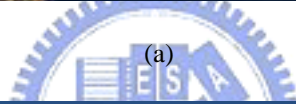


(a)



(b)

圖 3.2 (a) Bi-Cubic , (b) Bi-Cubic 加上 Hi-Boost 後處理。



(b)

圖 3.3 (a) Bi-Cubic, (b) Bi-Cubic 加上 Hi-Boost 後處理。



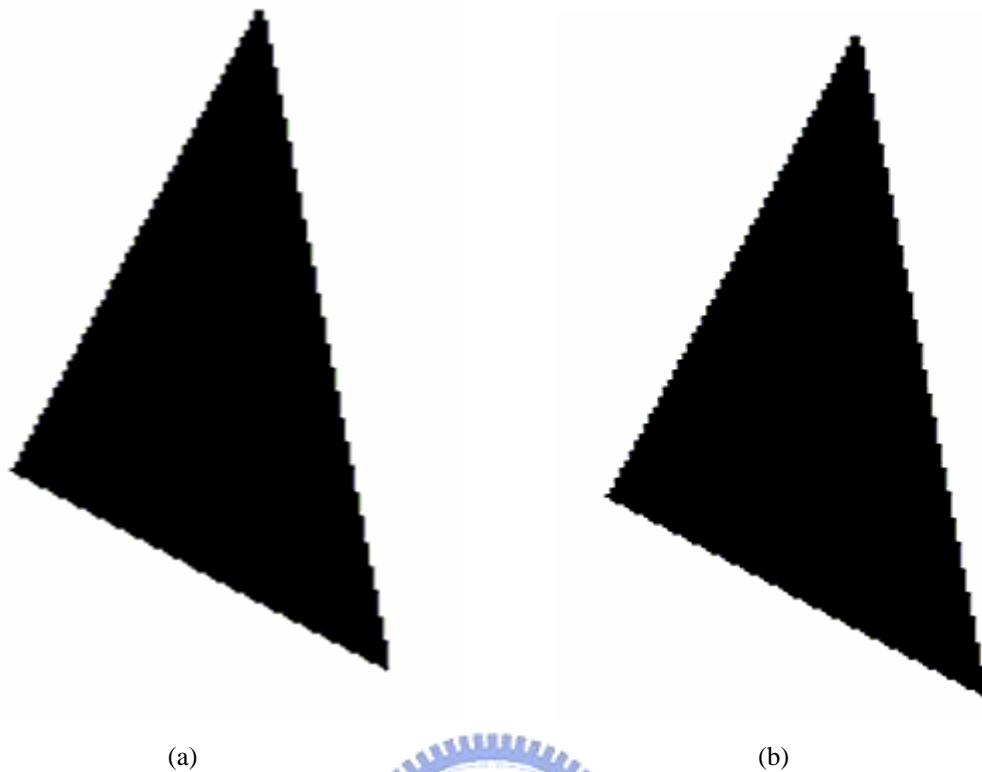


圖 3.4 (a) Bi-Cubic，(b) Bi-Cubic 加上 Hi-Boost 後處理。

實驗結果發現在圖 3.1(a)中被黃色圈起來的文字周圍看起來只有一點點變白的狀況，但是圖 3.2(b)中被黃色圈起來的文字周圍就有很明顯的變白，且亮度也有變亮的狀況。在圖 3.2 中，則有四個地方值得觀察，第一個地方是 1.a 所指的藍色線條，不管是直的或是橫的寬度都很一致，但是亮度比較暗，且邊緣也沒有那麼銳利，而 1.b 所指的藍色線條，亮度比較亮，邊緣也比較銳利，但是寬度看起來就不是那麼一致；第二個地方是 2.a 指的紅線，亮度也是比較暗，邊緣也比較不銳利，但是 2.b 所指的紅線亮度就比較亮，且邊緣較銳利；第三個地方是 3.a 所指的圓圈，亮度雖然比較暗，也看起來比較模糊，但是每個圓圈看起來大小一致，而 3.b 所指的圓圈，亮度雖然比較亮，看起來也比較銳利，但是每個圓圈大小看起來就比較不一致；最後一個地方是圖 3.2 (b)圖中黃色圈起來的文字比圖 3.2 (a)中黃色圈起來的文字，銳利很多。在圖 3.3 中可以看到 Hi-Boost 後處理之後產生的一些問題，如圖 3.3(b)天空紅色箭頭所指的地方經過 Hi-Boost 之後，將 Noises 強化了，相較之下圖 3.3(a)中，天空紅色箭頭所指的地方由於沒有經過

Hi-Boost Filter，因此 Noises 未被突顯出來；另外，在山的附近紅色箭頭所指的地方，圖 3.3(a)看起來是很自然的，但是圖 3.3(b)經過 Hi-Boost Filter 之後，會明顯看到一條黑色的線，且還有 Ringing Effect，不過在影像中其他的部分，確實可以發現，經過 Hi-Boost 後處理可以將影像銳利化。在圖 3.4 中，可以發現(b)的邊緣會比(a)銳利，但仍然沒有辦法讓放大過後的邊緣變的比較平順。

因此可以發現這個方法只需要在基本放大之後加上一個簡單的 Hi-Boost 後處理，就可以強化高頻。但是這個方法在強化高頻的同時，也會增加 Noises，且亮度也會有所改變，所以這種後處理的方法是一種效果中等且簡單的方法。

### 3.1.2 影像放大與強化之內插與外插

在一維度的外插內插混合運算方法中是根據1.2章節中提到的一維度內外插補點方法交互使用的觀念，我們嘗試了兩種的演算法。這兩種演算法中，若是分析出未知點附近的已知點變化很平緩，則我們做Bi-Cubic內插補點；若是分析出未知點附近的已知點變化很劇烈的話，則我們有兩種計算的方法，第一種如圖3.5所示，先利用第一個已知點以及第三個已知點連線延伸出去做外插，之後再用已經做好外插的點做線性內插；第二種如圖3.6所示，利用第一個已知點以及第二個已知點的中點，和第二個已知點以及第三個已知點的中點，連線延伸出去做外插計算，之後再利用已經做好外插計算的點，做線性內插計算。

經由實驗結果發現，若是輸入的原始影像為自然影像（圖3.7），則經由兩種方式放大過後的效果差不多（圖3.8以及圖3.9）。但若是輸入的原始影像是像圖3.10比較特殊類型的測試影像，則可以發現使用第一種方法放大的效果（圖3.11），會比使用第二種方法放大的效果來得好（圖3.12）。

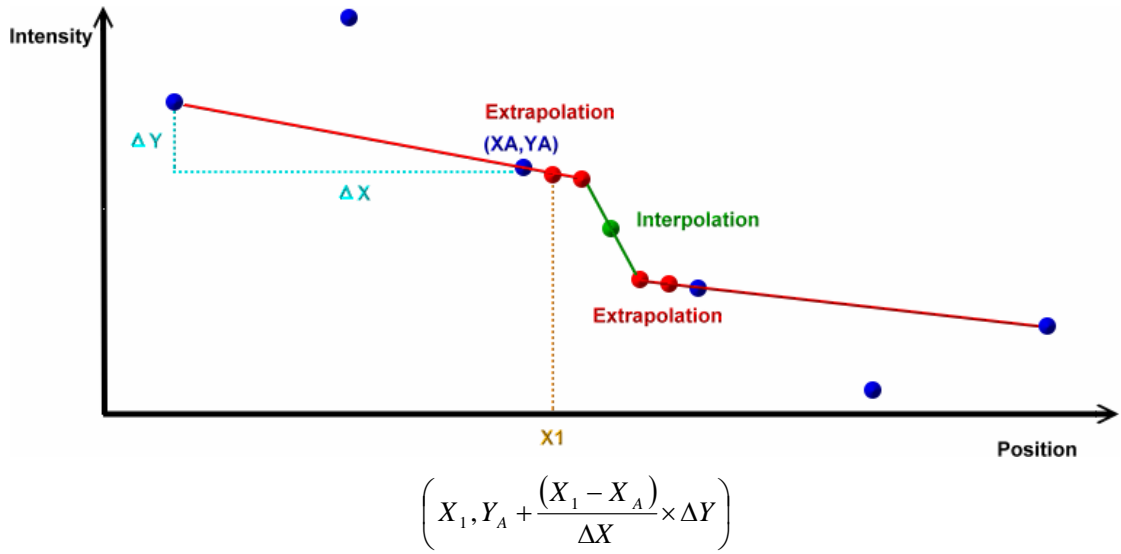


圖 3.5 方法 1：藍點是原本的點，紅點是做外插的點，綠點是做內插的點（放大十倍）。

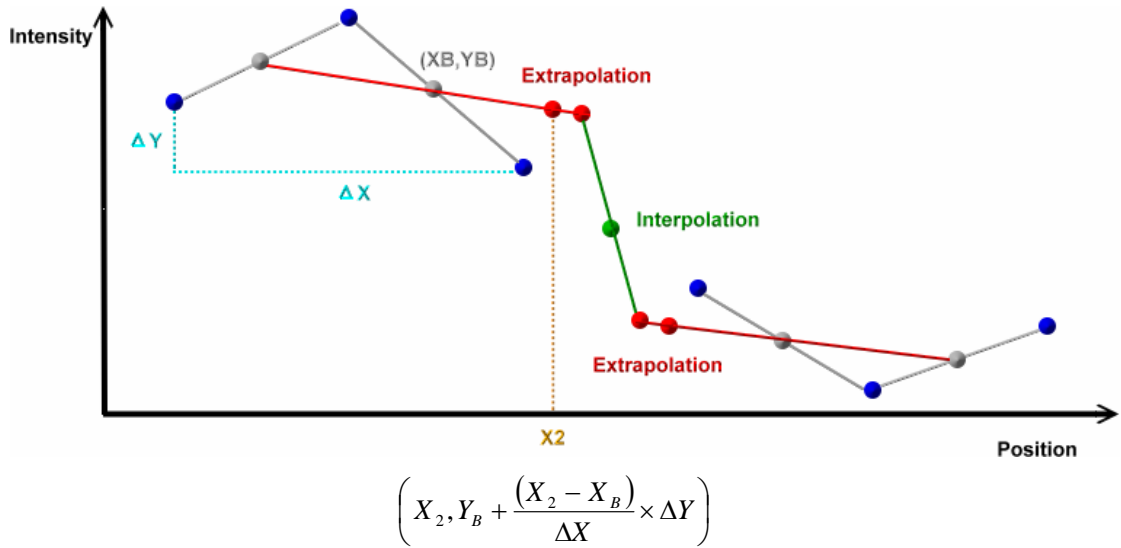


圖 3.6 方法 2：藍點是原本的點，紅點是做外插的點，綠點是做內插的點（放大十倍）。



圖 3.7 原始影像：自然影像。



圖 3.8 經由第一種內外插交互運算補點方法，將圖 3.7 放大四倍。



圖 3.9 經由第二種內外插交互運算補點方法，將圖 3.7 放大四倍。

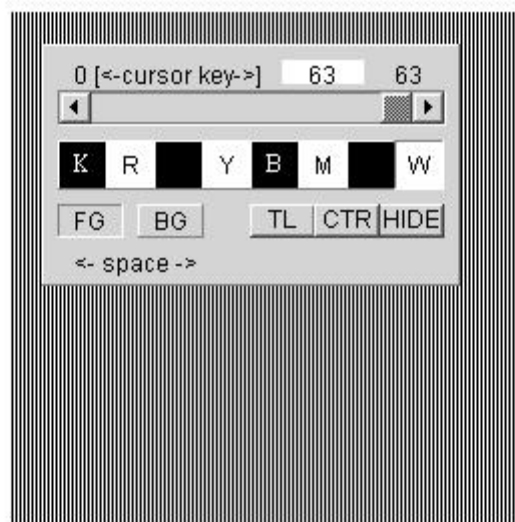


圖 3.10 原始影像：特殊的測試影像

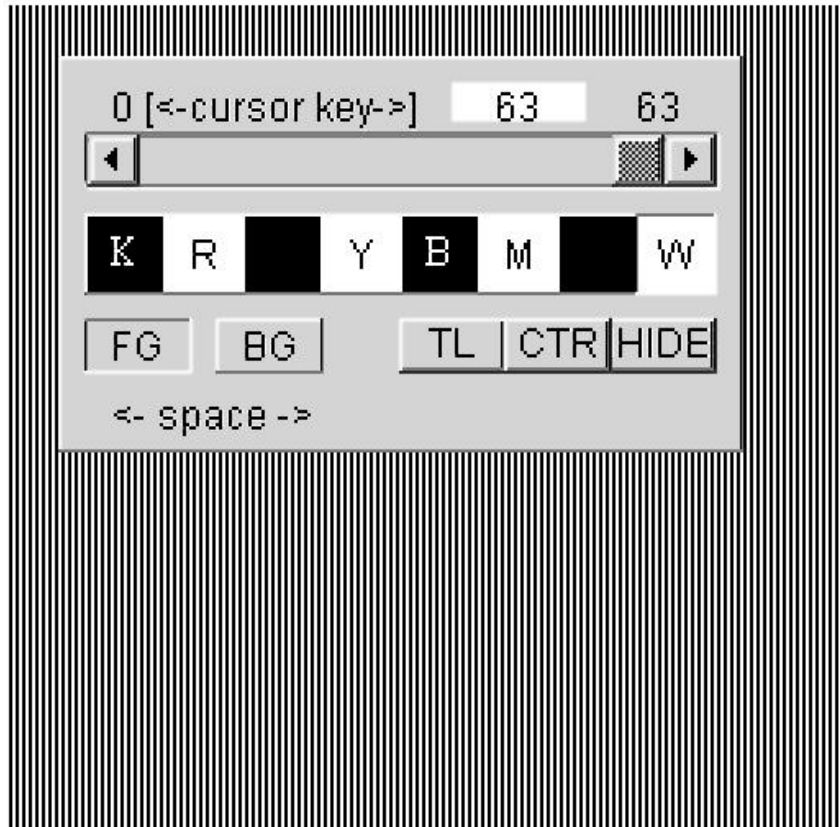


圖 3.11 經由第一種內外插交互運算補點方法，將圖 3.10 放大四倍。

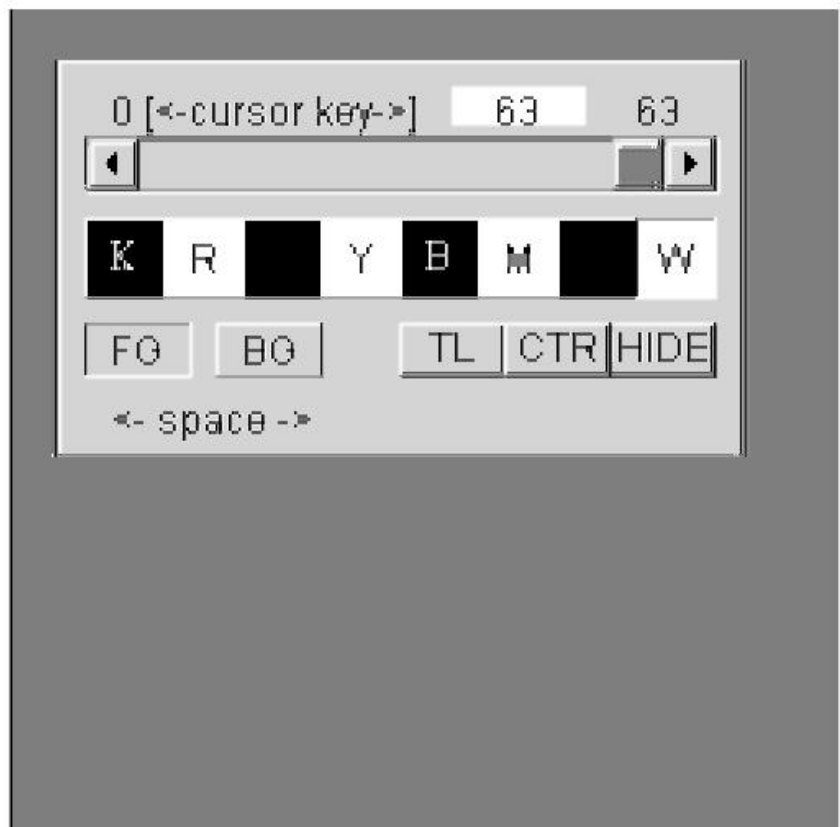


圖 3.12 經由第二種內外插交互運算補點方法，將圖 3.10 放大四倍。

## 3.2 內插與外插架構介紹

前面章節提到的一維度影像放大以及強化方法，我們只有分析原始影像中一條 Scan Line 的幾個點灰階值變化方式，決定要做內插補點或是部分的內插補點加上部分的外插運算。若是推廣到二維度影像放大以及強化的方法中時，我們由原本分析原始影像的一條 Scan Line 擴大成分析原始影像二維度的小區塊。

如圖 3.13，在分析原始影像二維度的小區塊時，主要作法是先定義出我們認為是邊緣或是角的種類，再將原始影像的小區塊（Window）中的資訊與預先定義的小區塊作比較，判斷原始影像的邊緣或是角的資訊符合預先定義的哪種邊緣或是角的種類，再依照各種不同的種類，做外插銳化放大。若是都不符合我們預先定義的種類，則做內插平滑放大，透過內插外插運算混合使用的方式，同時完成影像放大與影像銳利度強化。預先定義的分類以及判斷的流程，將於後面的章節做詳細的介紹。

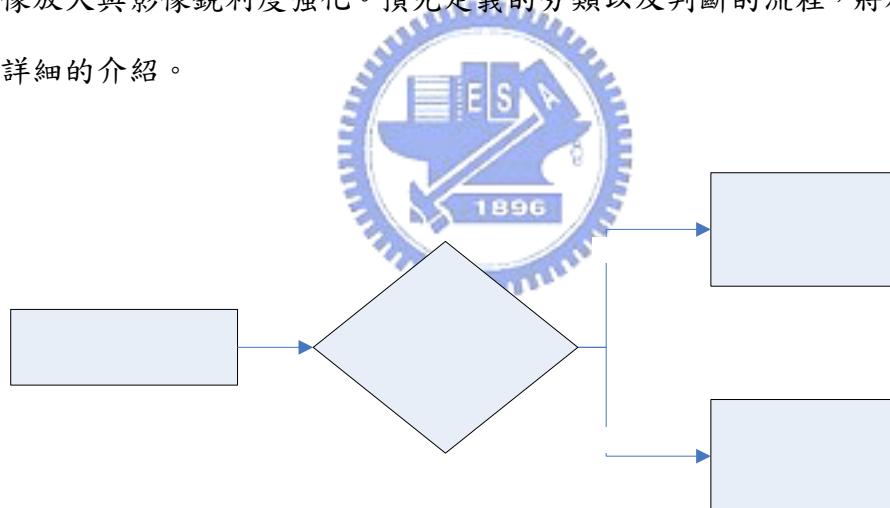


圖 3.13 影像同時放大以及強化流程圖。

## 3.3 二維內插外插之影像放大與強化方法介紹

在這一小節，我們會提出三種不同的二維度內插外插混合運算方法，三種方法在放大的過程以及觀念上，都是如前一小節所提到的基本觀念，唯一不同的就是分析原始影像的小區塊大小。選擇分析的小區塊大小時，因為最小的二維度小區塊是  $2 \times 2$ ，因此我們從最小的  $2 \times 2$  區塊開始探討，在  $2 \times 2$  的小區塊中，可以預

先定義出七種邊緣以及角的分類。

接著為了配合內插補點 (Bi-Cubic) 的運算，先擴大一個維度的分析，因此接下來選擇的是分析  $4 \times 2$  的小區塊。這樣大小的小區塊，可以預先定義出十六種邊緣以及角的分類，但是這樣的小區塊，會造成垂直與水平方向的原始資訊取得不平衡，因此在判斷上，垂直方向影像放大以及強化的效果會比水平的差。

最後為了平衡水平以及垂直方向原始影像的資訊取得，還有配合內插的運算架構，將分析原始影像的小區塊擴大成為  $4 \times 4$  的小區塊。這樣大小的區塊，可以預先定義出七十八種邊緣以及角的分類。

### 3.3.1 判斷 $2 \times 2$ 區域內邊以及角的處理

如圖3.14所示，當分析原始影像的小區塊大小為  $2 \times 2$  的時候，區域內只有4個參考點；因此，由這四個參考點我們可以定義出的邊緣種類一共有七種如圖3.15所示。接下來就是判斷原始影像小區塊中的4個參考點是屬於那一種類型的邊緣，圖3.16是  $2 \times 2$  的判斷樹狀圖 (圖3.17為其簡化圖，本論文之後的樹狀圖皆為簡化圖)，判斷的結果會有幾種可能：(1) 中間四個參考點，2對相鄰的參考點的灰階值很相近 (圖3.15(a)、(b))，(2) 中間四個參考點，其中3個參考點的灰階值很相近 (圖3.15(c)到(f))，(3) 對角的兩個點灰階值很相近 (圖3.15(g))。將未知點判斷完後，再針對每種邊緣做不同的計算，在這個方法中Cases A、B、G都只有預先定義出一種邊緣，而Cases C、D、E、F會有三種不同邊緣的方程式如圖3.15(c)到(f)，分別會畫出不同的區域大小，因此也會得出不同的結果，這個選項是可以調整的。

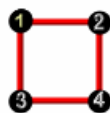
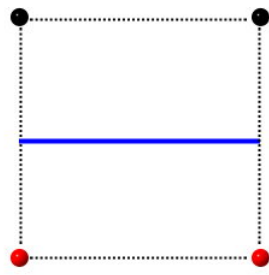
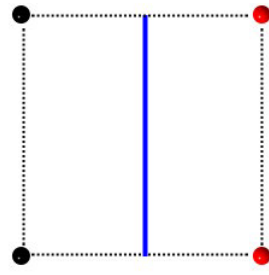


圖 3.14  $2 \times 2$  小區塊大小為  $2 \times 2$  時，共有四個參考點。

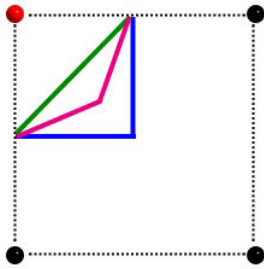




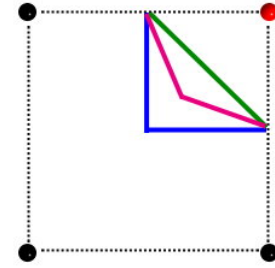
(a)



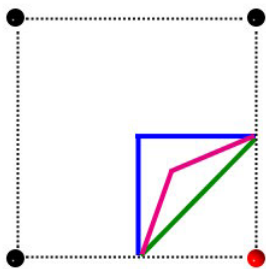
(b)



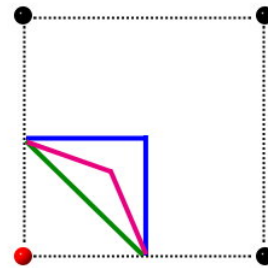
(c)



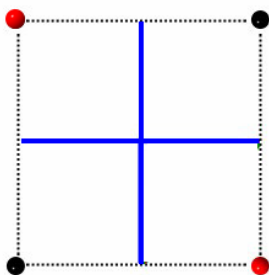
(d)



(e)



(f)



(g)

圖 3.15 (a)(b)分別為 Case A、B，藍線即為我們定義的邊緣線，(c)~(f)分別為 Cases C、D、E、F，這四種 Cases 都各自定義出三種邊緣，分別標示為藍線、紅線以及綠線，根據需求的不同，可以選擇不同的邊緣做計算。

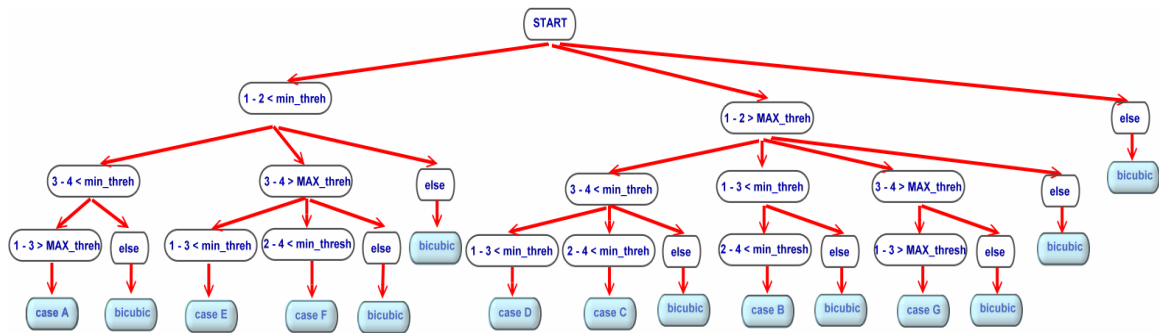


圖 3.16 判斷中間四個參考點的流程圖，每個 Level 都是一個 if 判斷條件。

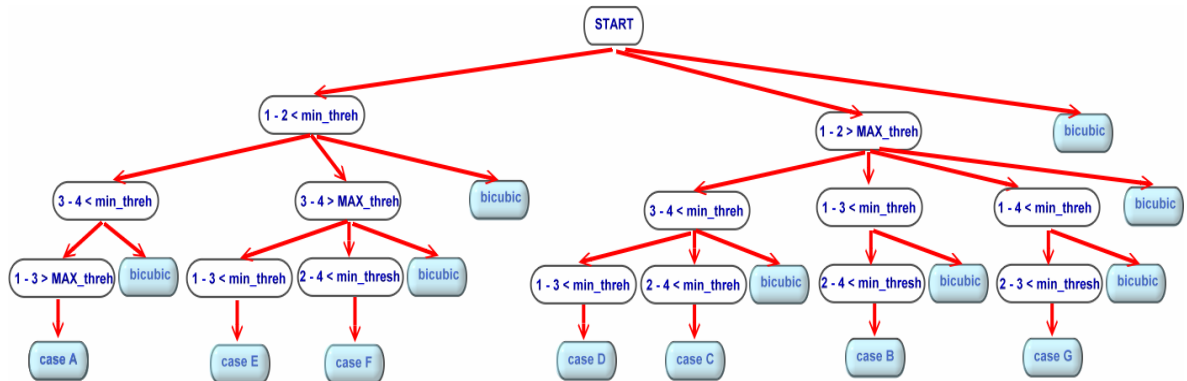


圖 3.17 將圖 3.16 中之 else 部分刪除後之簡化圖。

判斷出符合預先定義中的那一個種類的邊緣之後，再根據每種不同定義的邊緣，找出邊緣的方程式  $f(x)$  (圖 3.18 中綠色的線)。根據要被外插的點落的區域是屬於哪個區域，再做計算。以為圖 3.18 例：

$$\begin{cases} I_{new}(x) = [I_1(x) + I_2(x) + I_3(x)]/3, & \text{if } f(x) < 0 \\ I_{new}(x) = I_4(x) & , \text{if } f(x) \geq 0 \end{cases} \quad (7)$$

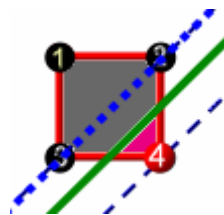


圖 3.18 綠線為邊緣出現的地方  $f(x)$ 。

### 3.3.1 判斷 4x2 區域內邊的处理

由於分析原始影像的小區塊大小為  $2 \times 2$  的時候，可以判斷邊緣的資訊只有四個參考點，因此只能定義出七種邊緣。影像做外插放大的地方雖然比較多，但是卻只有七個種類，所以為了要判斷出更多種類的邊緣，我們將分析原始影像的小

區塊大小增加為 $4 \times 2$ ，在這個小區塊中參考點一共會有8個點，因此能預先定義出來的邊緣會有十六種，但由於資訊仍然不足夠，因此在下面定義的邊緣種類中，會將角的種類看成邊的種類做處理。

一開始先判斷中間四個參考點，決定圖3.14的四個參考點（ $2 \times 2$ ）是屬於圖3.15(a)到(f)中那一種類型的邊緣（樹狀圖如圖3.19），之後，再判斷剩下外圍的四個點。在大小為 $4 \times 2$ 的小區塊中，Case A一共有七種邊緣的種類變化，判斷的樹狀圖如圖3.20，其中else的部分是作內插補點計算，而Case B由於受限於小區塊大小的關係，所以在垂直方向只能有一種邊緣的變化，因此Case B只有預先定義出一種垂直的邊緣，判斷樹狀圖如圖3.21。Cases C到F在這個方法中各預先定義出兩種邊緣，判斷樹狀圖如圖3.22到圖3.25，因此在這個方法中所有定義的邊緣種類如圖3.26共有十六種邊緣，這個方法是嚴謹的判斷每種Case，且完整的判斷Case A的邊界種類。

這個區塊大小的外插補點計算方式如圖3.27，若是如圖3.27(a)要補的點落在綠線左下方，則用3、4兩個點的灰階值平均，當作要被補點的灰階值；反之在綠線右上方，就用1、2兩個點的灰階值平均，當作要被補點的灰階值。若是如圖3.27(b)要補的點落在綠線右下方，則用2、3、4三個點的灰階值平均，當作要被補點的灰階值；反之在綠線左上方，就用1的點的灰階值，當作要被補點的灰階值。若是如圖3.27(c)要補的點落在綠線左上方，則用1、a、c三個點的灰階值平均，當作要被補點的灰階值；反之在綠線右下方，就用2、3、4三個點的灰階值平均，當作要被補點的灰階值。

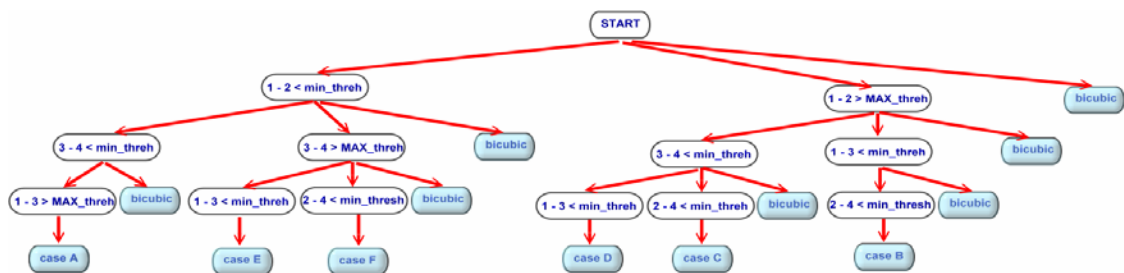


圖 3.19 判斷中間四個參考點的流程圖。

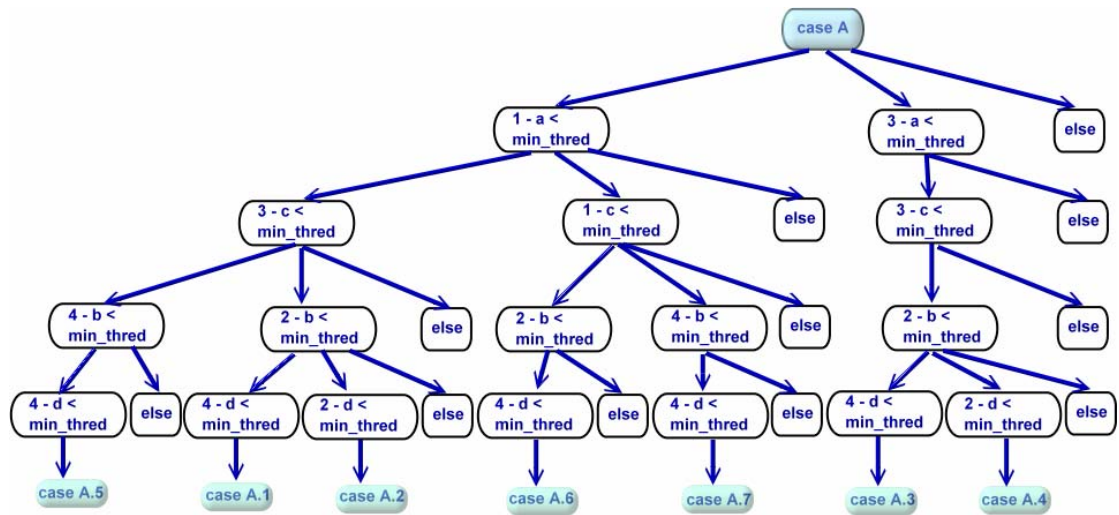


圖 3.20 Case A 的判斷樹狀圖。

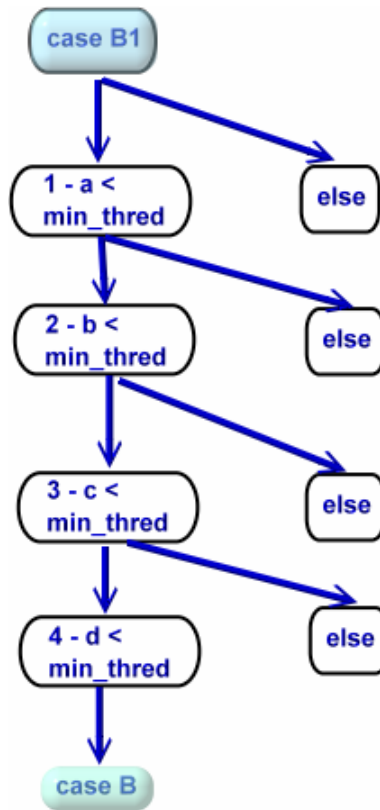


圖 3.21 Case B 的判斷樹狀圖。

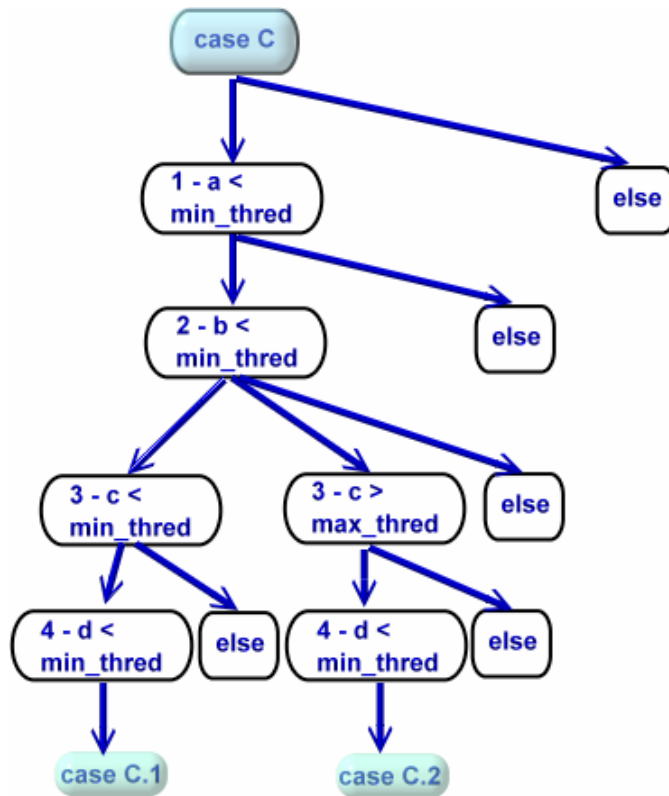


圖 3.22 Case C 的判斷樹狀圖。

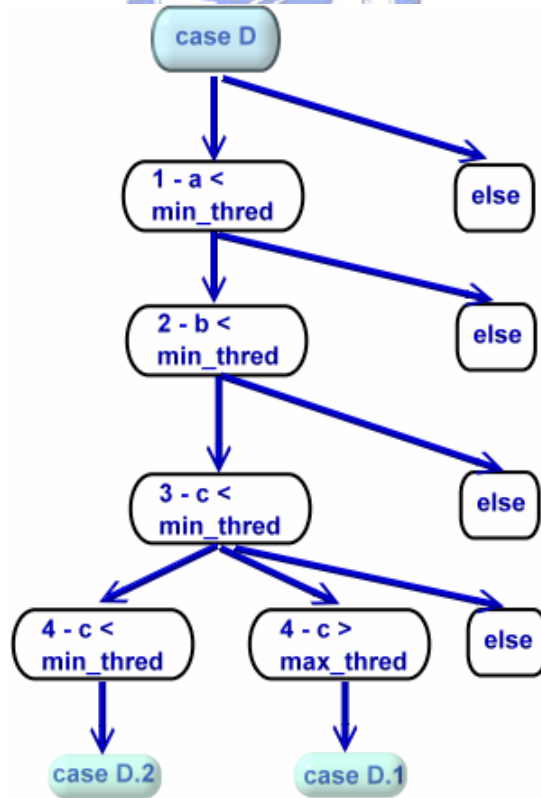


圖 3.23 Case D 的判斷樹狀圖。

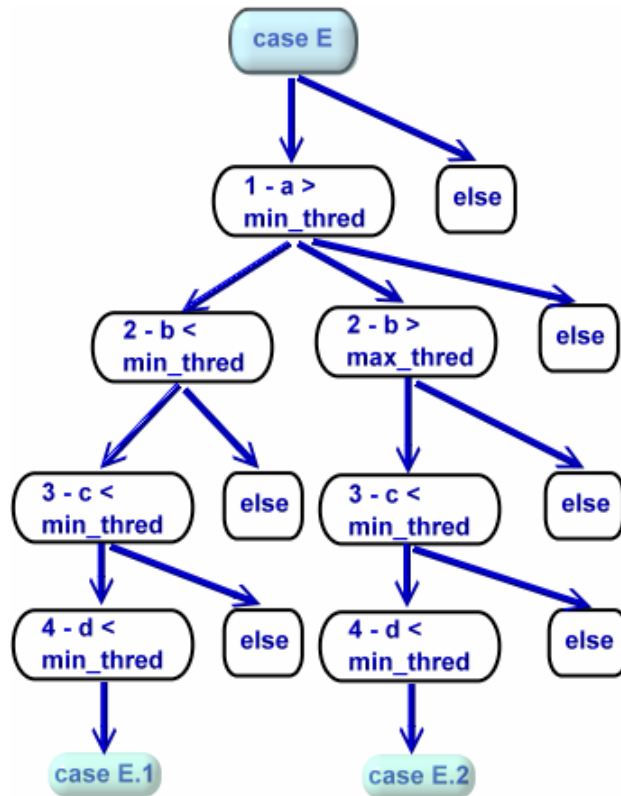


圖 3.24 Case E 的判斷樹狀圖。

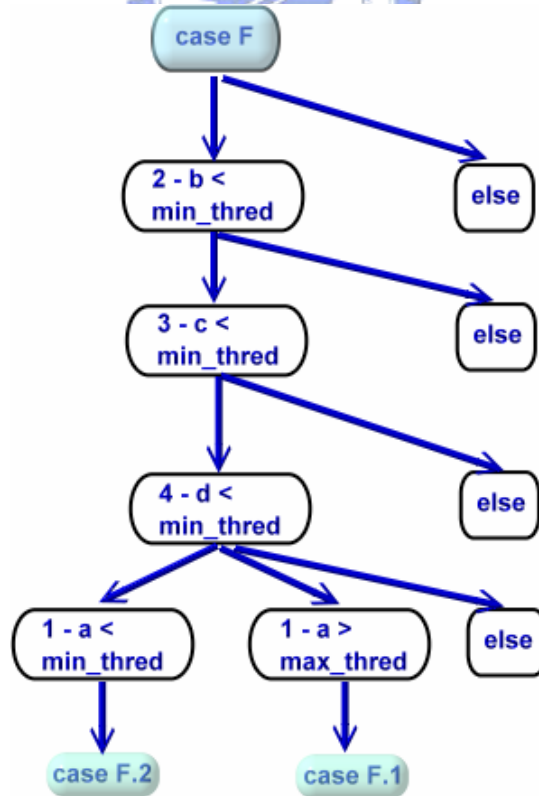


圖 3.25 Case F 的判斷樹狀圖。

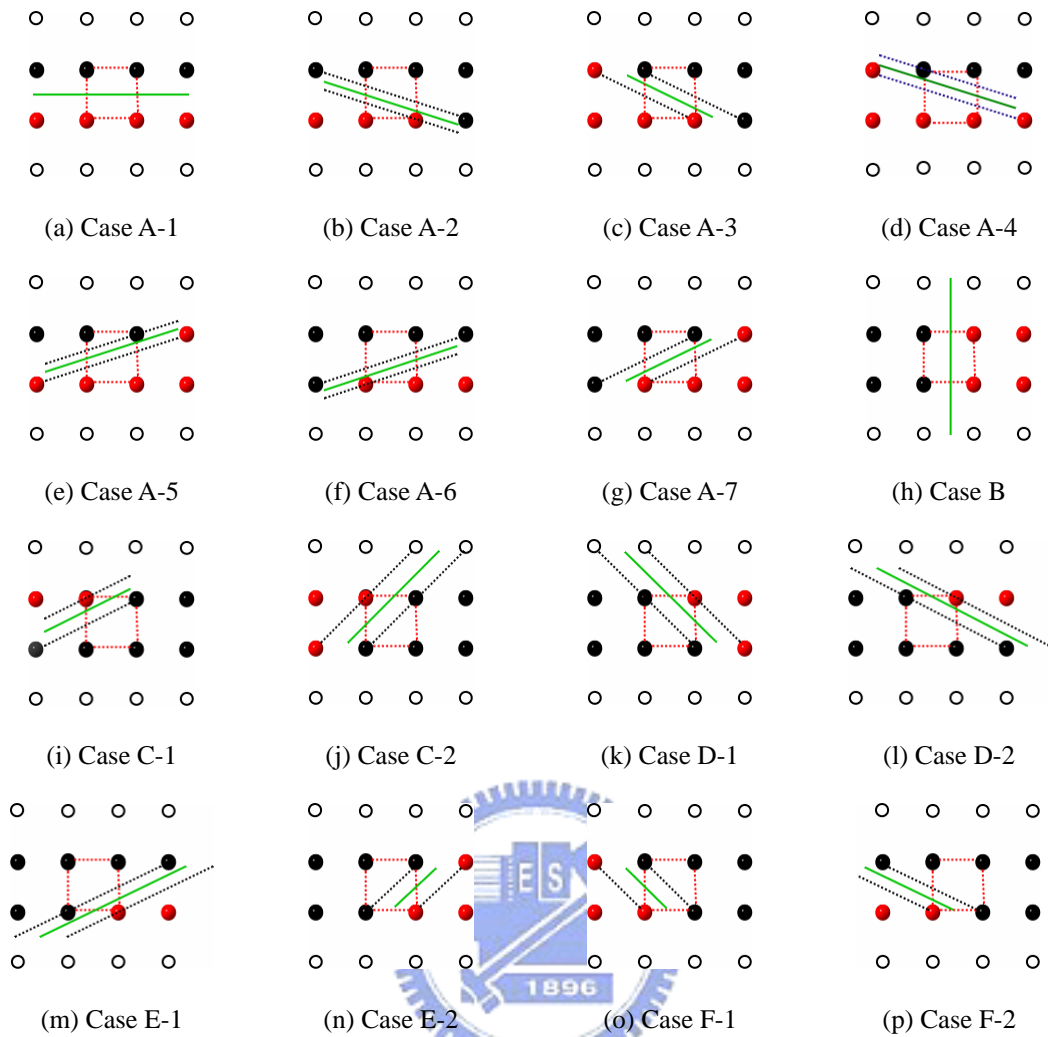


圖 3.26 分析原始影像小區塊大小為 $4 \times 2$ 的方法中，預先定義的各種邊緣，在每種邊緣中，綠色的線代表定義邊緣的地方。

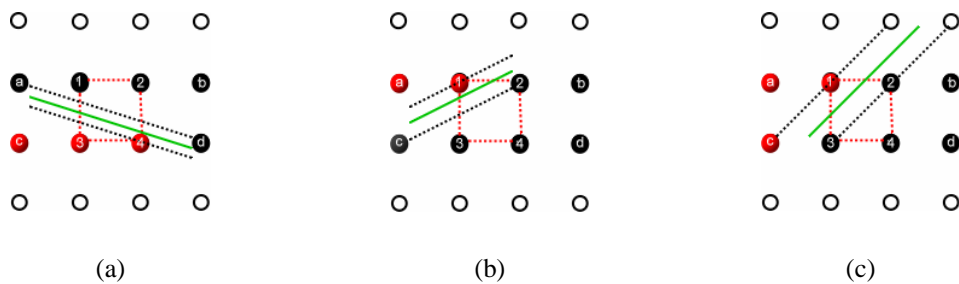


圖 3.27 (a) 要補的點落在綠線左下方，用 3、4 兩個點的灰階值平均，當作要被補點的灰階值，反之在綠線右上方，用 1、2 兩個點的灰階值平均，當作要被補點的灰階值，(b) 要補的點落在綠線右下方，用 2、3、4 三個點的灰階值平均，當作要被補點的灰階值，反之在綠線左上方，用 1 的點的灰階值，當作要被補點的灰階值，(c) 要補的點落在綠線左上方，用 1、a、c 三個點的灰階值平均，當作要被補點的灰階值，反之在綠線右下方，用 2、3、4 三個點的灰階值平均，當作要被補點的灰階值。

### 3.3.2 判斷 4x4 區域內邊的處理

為了配合Bi-Cubic演算法的架構，以及在放大強化後的影像中，其水平垂直方向被強化的平衡性，我們再將分析原始影像的小區塊擴張為4x4的大小。在這個小區塊內一共會有十六個原始影像參考點，透過圖3.28中的十六個原始影像參考點，我們可以預先定義出八十六種邊緣以及角，這八十六種邊緣以及角包含在圖3.29的七大種類中，而其中圖3.30的Case A、圖3.31的Case B這兩大種類各有七種邊緣以及八種角，圖3.32的Case C到圖3.35的Case F這四大種類也都各有七種邊緣以及四種角，剩下圖3.36的Case G這一大類，則包含有十二種角。之後做邊緣判斷時，我們會先判斷在圖3.37中十六個原始影像參考點中間的四個參考點是屬於圖3.29中的哪一大種類，判斷的樹狀圖如圖3.38，判斷的結果會有三種可能：(1) 中間四個參考點，兩組相鄰的參考點的像素值很相近(圖3.29(a)、(b))，(2) 中間四個參考點，其中三個參考點的像素值很相近(圖3.29(c)到(f))，(3) 對角的兩個點像素值很接近(圖3.29(g))。之後，再各自針對每個不同的大種類，更進一步的判斷是屬於八十六種中的哪一種邊緣或是角。

當再進一步判斷是屬於八十六種中的哪一種邊緣或是角時，我們利用判斷周圍幾個原始影像參考點與中間四個原始影像參考點的關係，便可以判斷出原始影像是屬於我們預先定義中的哪種邊或是角。如表1所示，對於每種Case會先評估周圍附近四個原始影像參考點，之後再評估其他的參考點。經由這樣的判斷之後，我們便可以得知，小區塊中的原始影像是否符合我們預先定義的邊緣或是角。

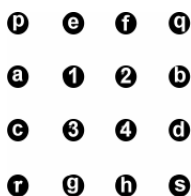


圖 3.28 十六個參考點



表 1 對圖 3.29 中的七個 Cases 再進一步判斷所要考慮的點及其順序。

	Case A	Case B	Case C	Case D	Case E	Case F	Case G
先評估之點	e, f, g, h (圖 3.39)	a, b, c, d (圖 3.43)	a, b, e, g (圖 3.47)	a, b, f, h (圖 3.50)	c, d, f, h (圖 3.53)	c, d, e, g (圖 3.56)	p, q, r, s (圖 3.59)
後評估之點	a, b, c, d (圖 3.40 到 圖 3.42)	e, f, g, h (圖 3.44 到 圖 3.46)	c, f, q, r (圖 3.48 及 圖 3.49)	d, e, p, s (圖 3.51 及 圖 3.52)	b, g, q, r (圖 3.54 及 圖 3.55)	a, h, p, s (圖 3.57 及 圖 3.58)	a, b, c, d e, f, g, h (圖 3.60 及 圖 3.61)

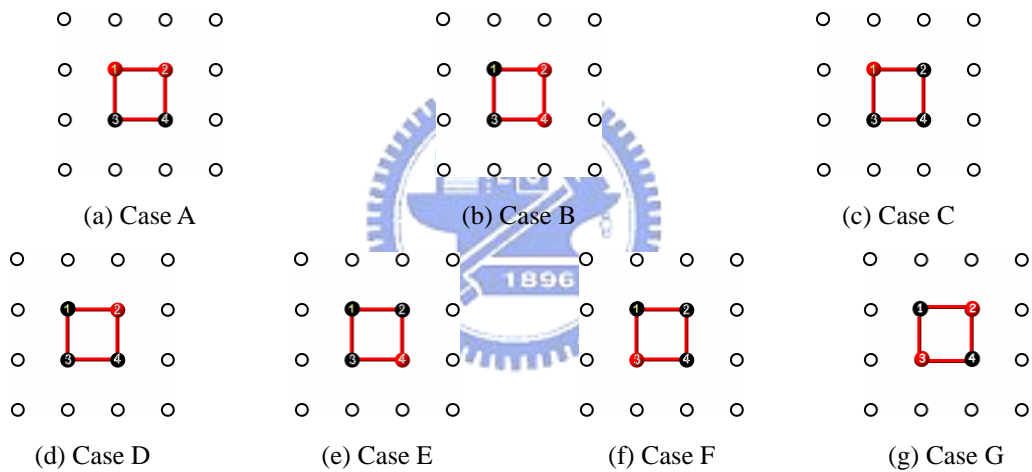


圖 3.29 分析原始影像小區塊大小為 4x4 之七大種類。

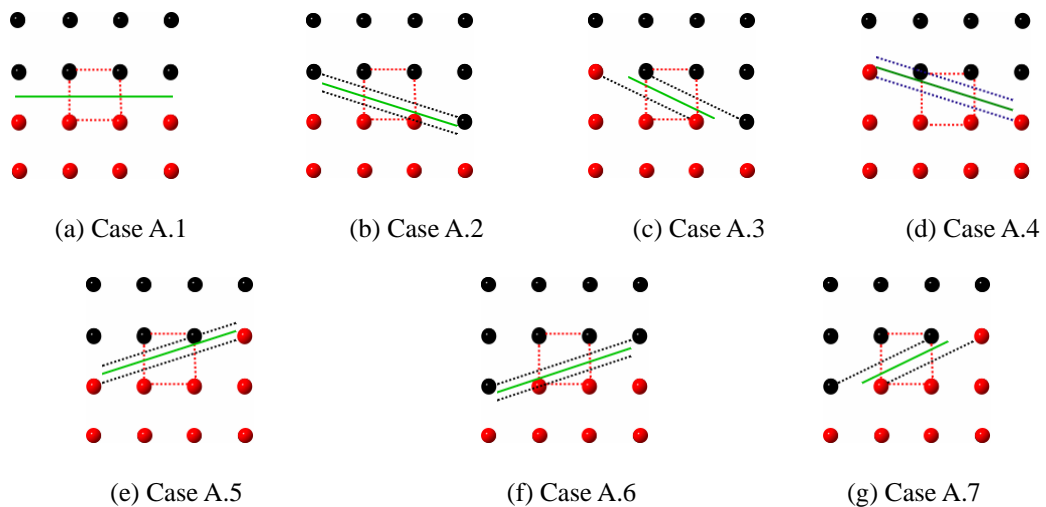


圖 3.30 (a) 到 (g) 是 Case A 中的七種邊緣，(h) 到 (o) 是 Case A 中的八種角。

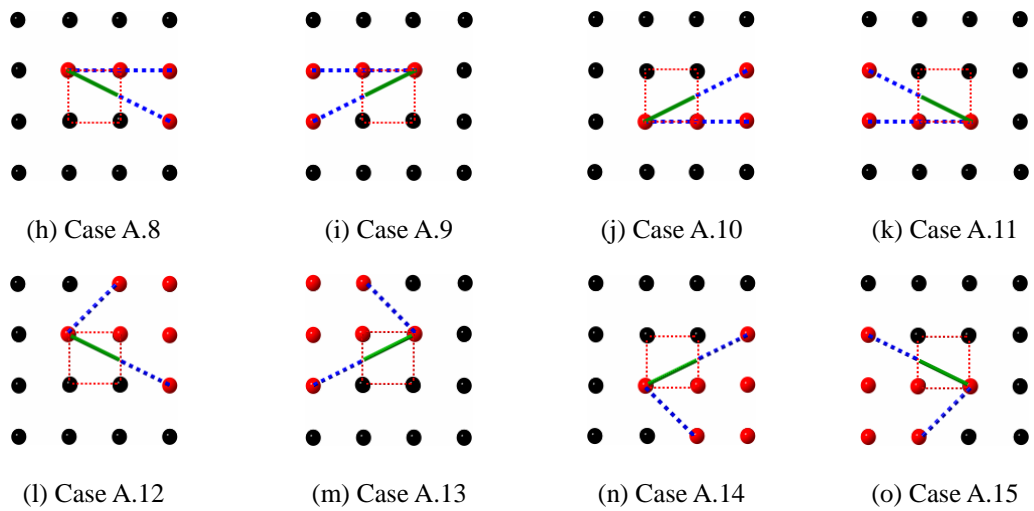


圖 3.30 (續)

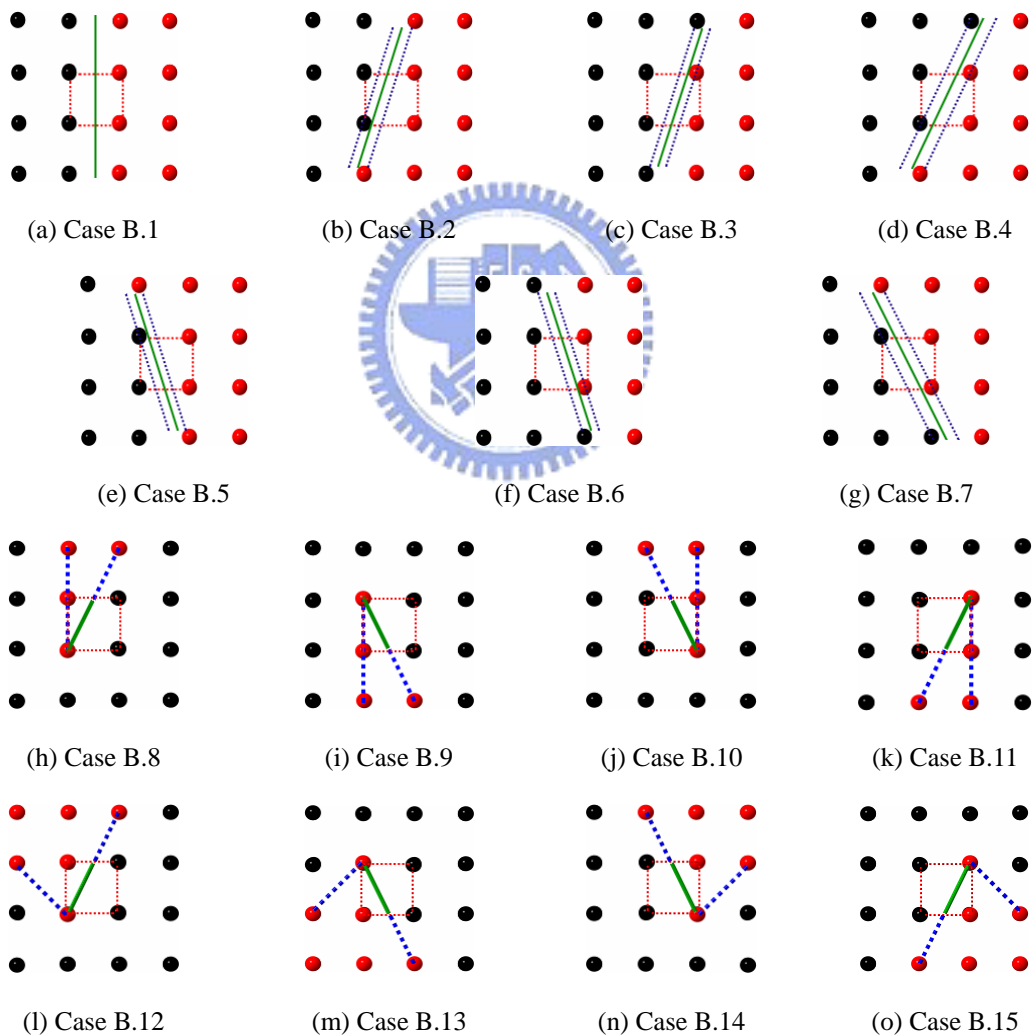


圖 3.31 (a) 到 (g) 是 Case B 中的七種邊緣，(h) 到 (o) 是 Case B 中的八種角。

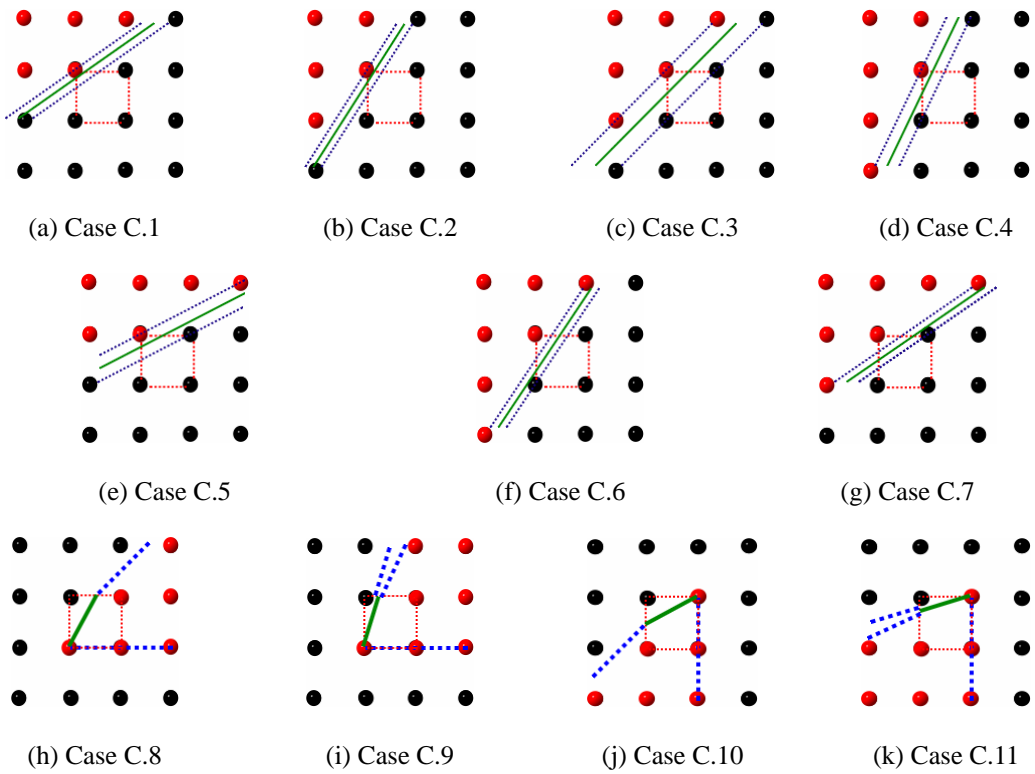


圖 3.32 (a) 到 (g) 是 Case C 中的七種邊緣，(h) 到 (k) 是 Case C 中的四種角。

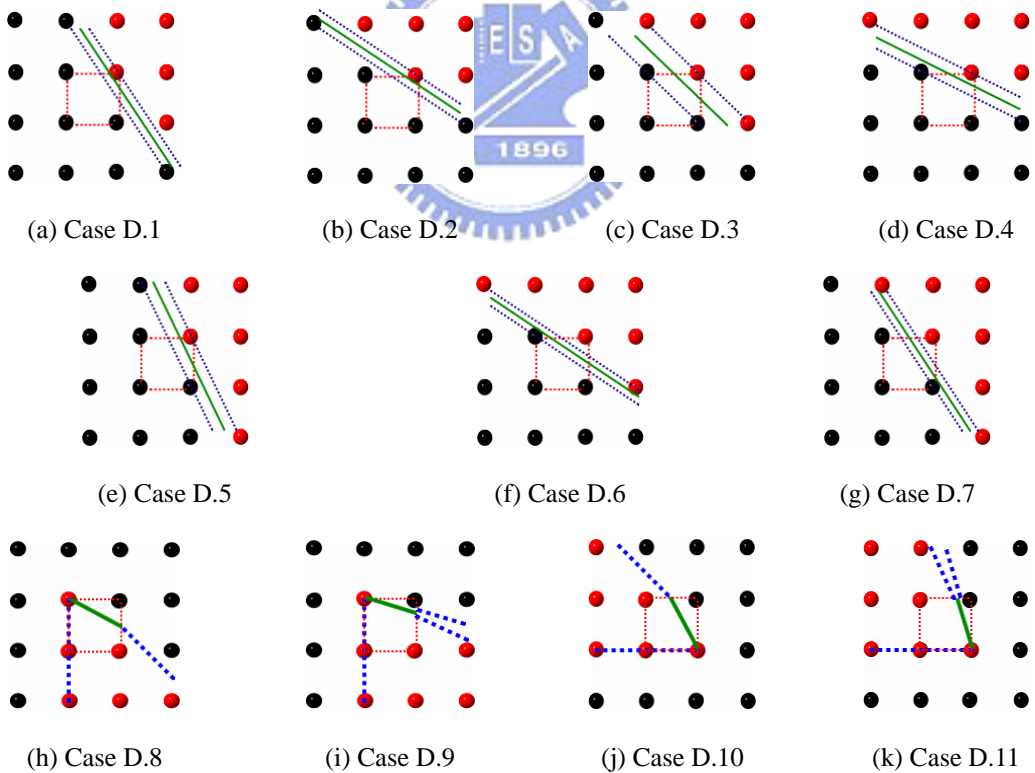


圖 3.33 (a) 到 (g) 是 Case D 中的七種邊緣，(h) 到 (k) 是 Case D 中的四種角。

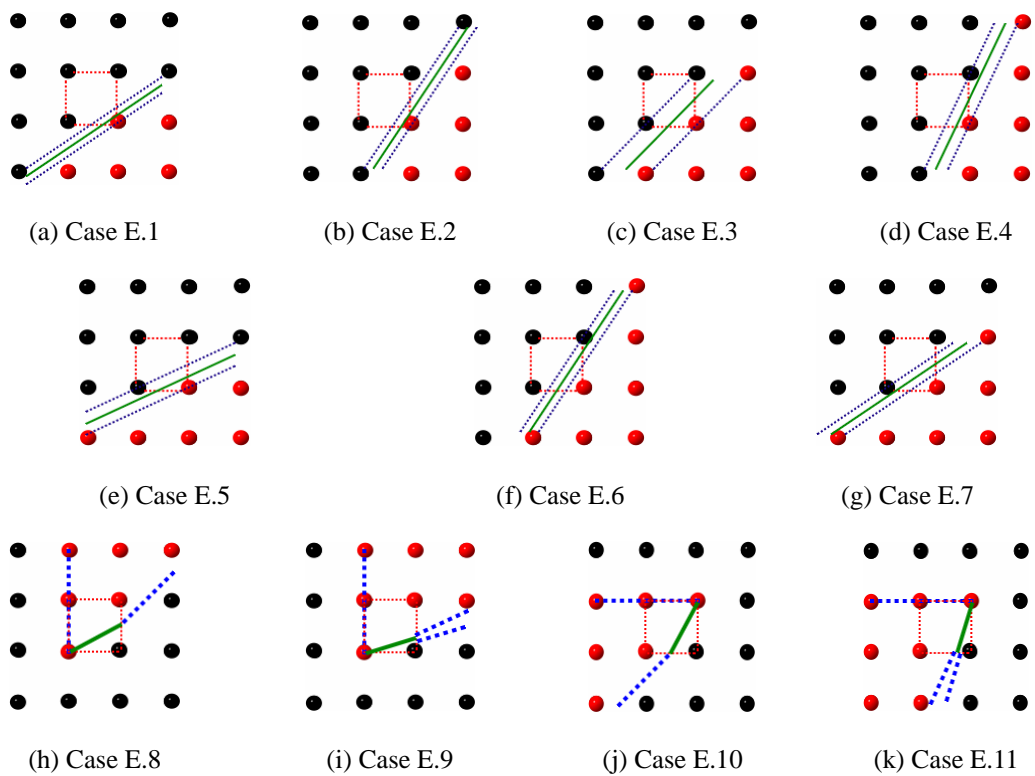


圖 3.34 (a) 到 (g) 是 Case E 中的七種邊緣，(h) 到 (k) 是 Case E 中的四種角。

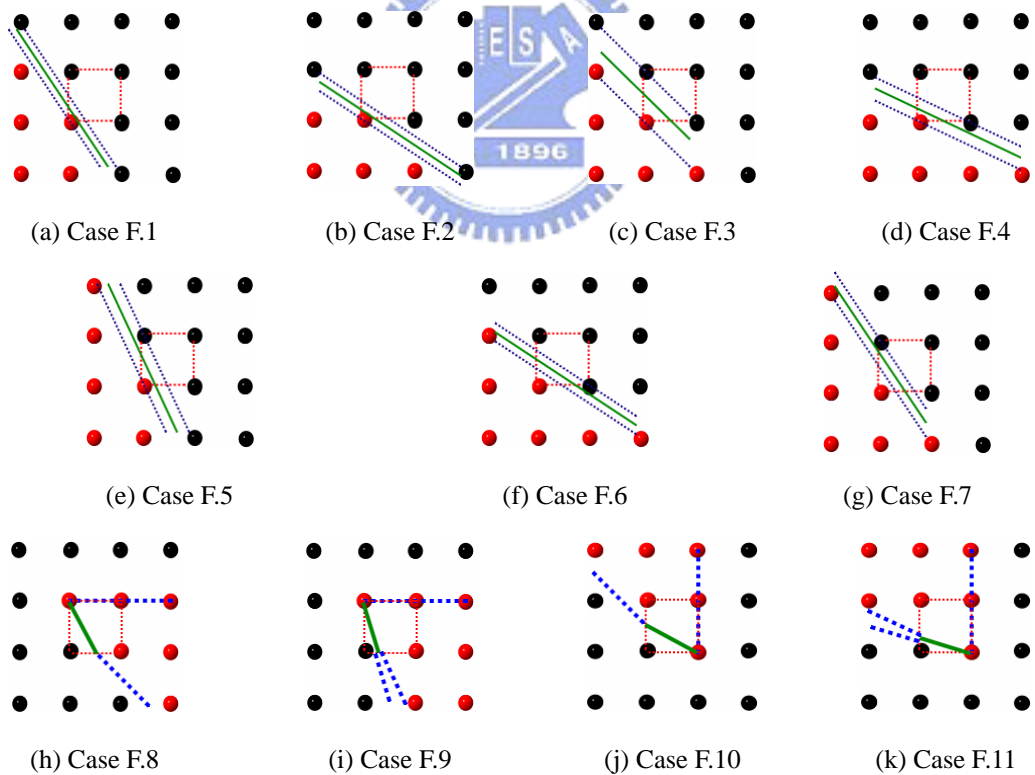


圖 3.35 (a) 到 (g) 是 Case F 中的七種邊緣，(h) 到 (k) 是 Case F 中的四種角。

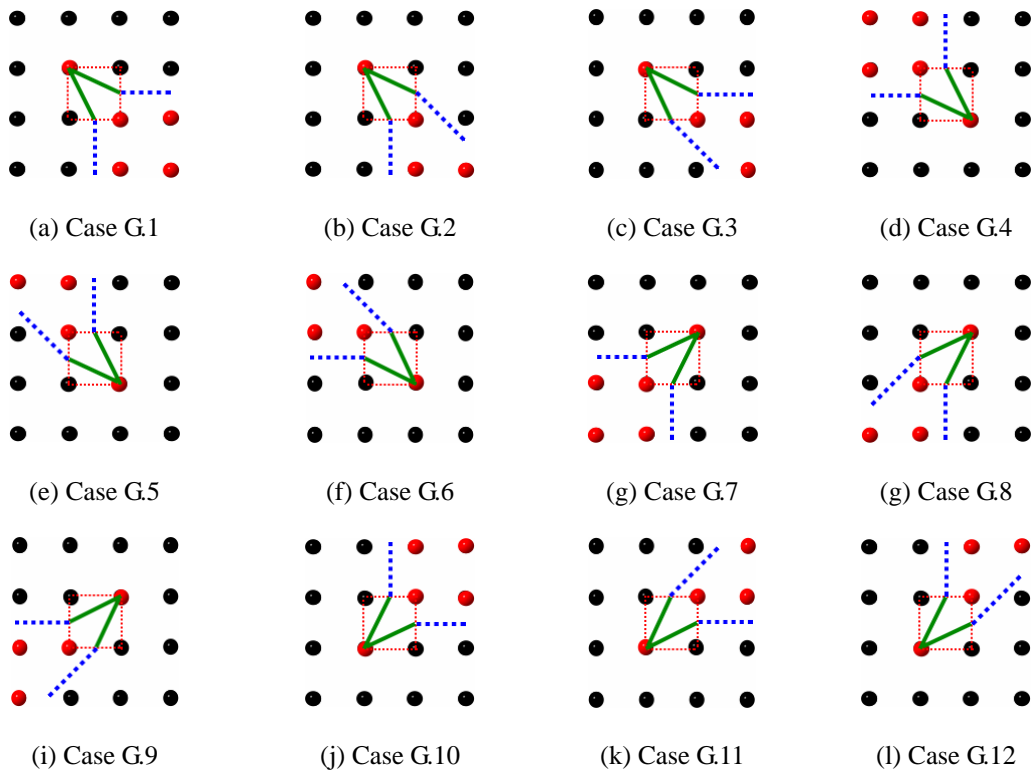


圖 3.36 (a) 到 (g) 是 Case G 中的十二種角。

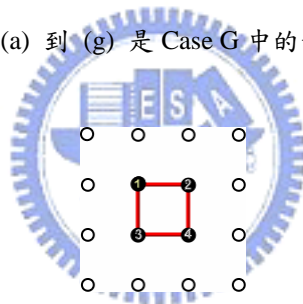


圖 3.37 定義中間補點區域的四個參考點。

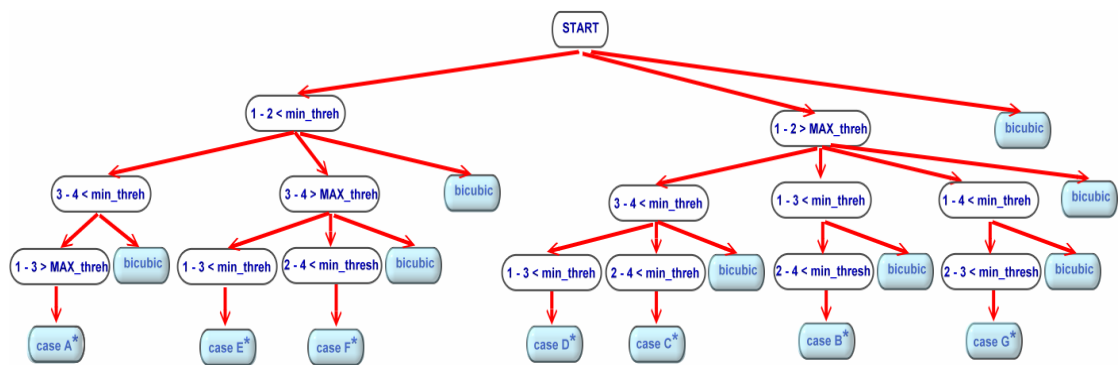


圖 3.38 判斷中間四個原始影像參考點的流程圖。

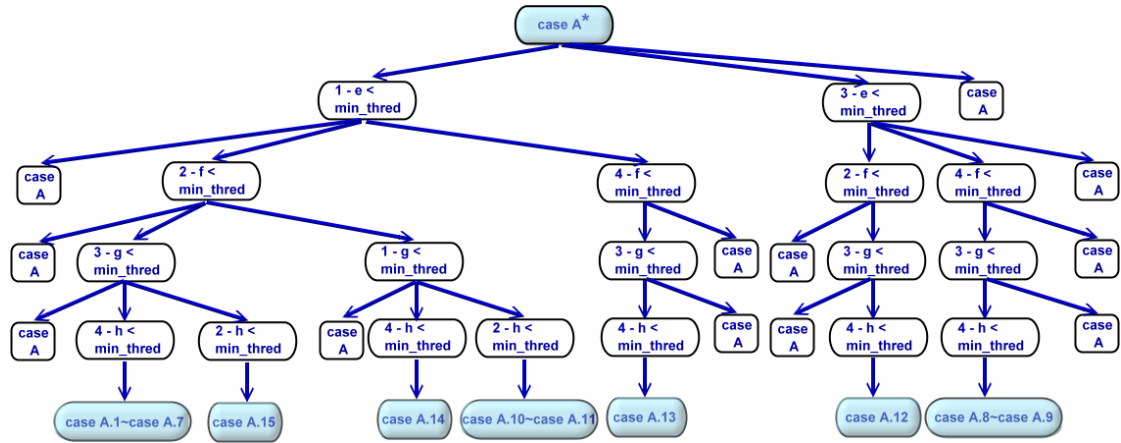


圖 3.39 對 Case A\*再進一步判斷是那個種類的邊緣或是角（共十五種）。

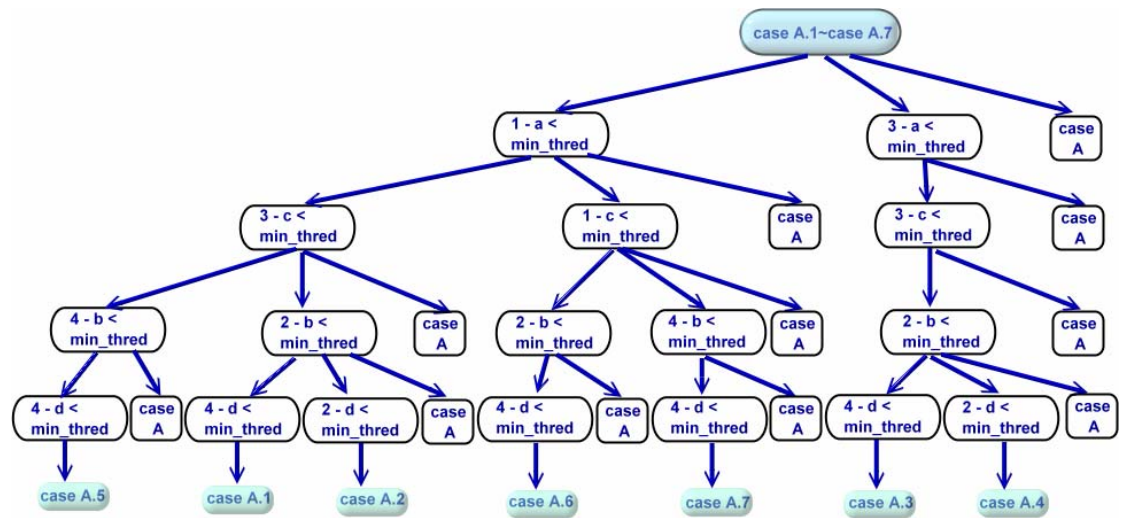


圖 3.40 對 Case A.1~Case A.7 再進一步判斷是那個種類的邊緣。

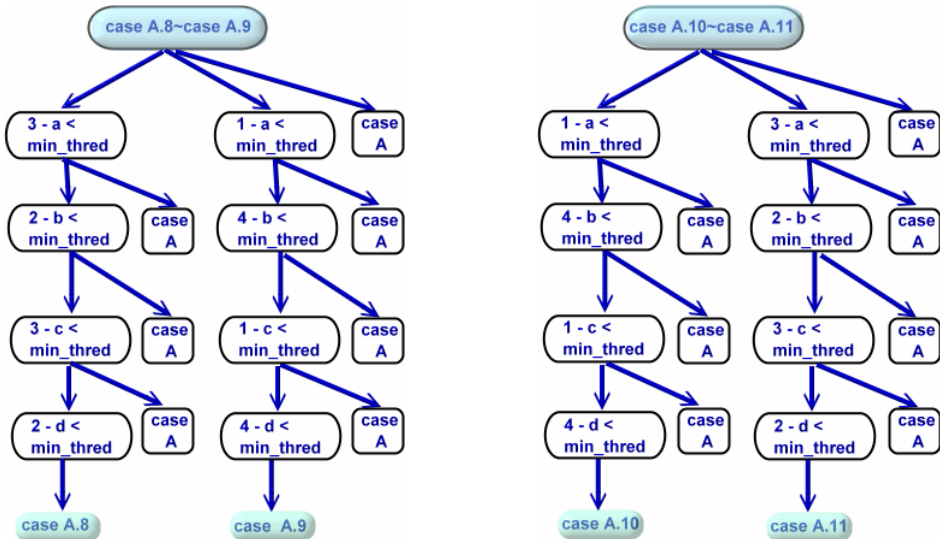


圖 3.41 對 Case A.8、Case A.9、Case A.10、Case A.11 再進一步判斷是那個種類的角。

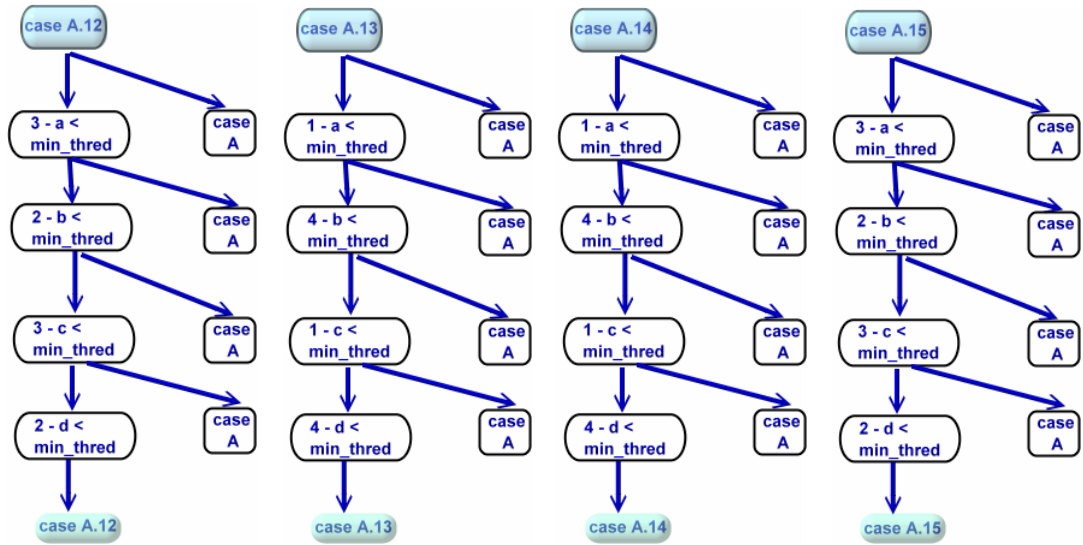


圖 3.42 對 Case A.12、Case A.13、Case A.14、Case A.15 再進一步判斷是那個種類的角。

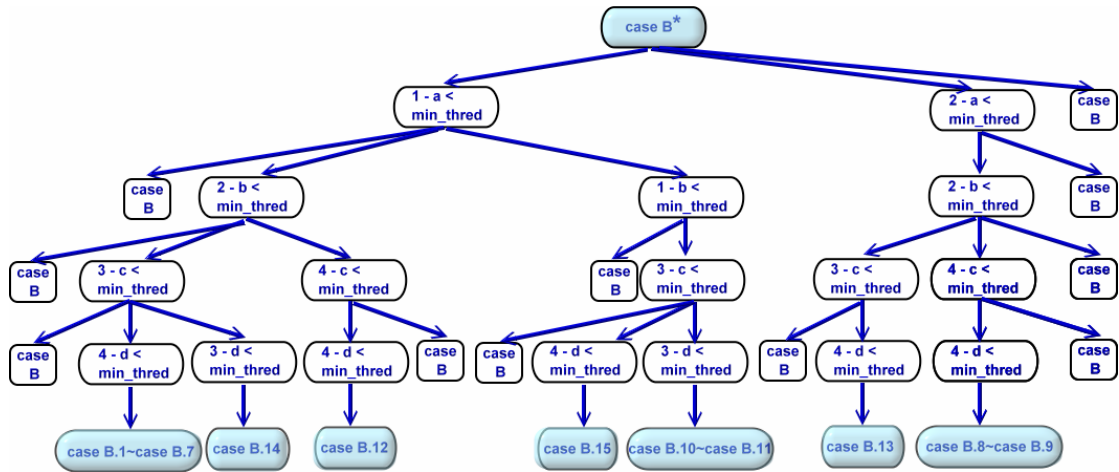


圖 3.43 對 Case B\*再進一步判斷是那個種類的邊緣或是角（共十五種）。

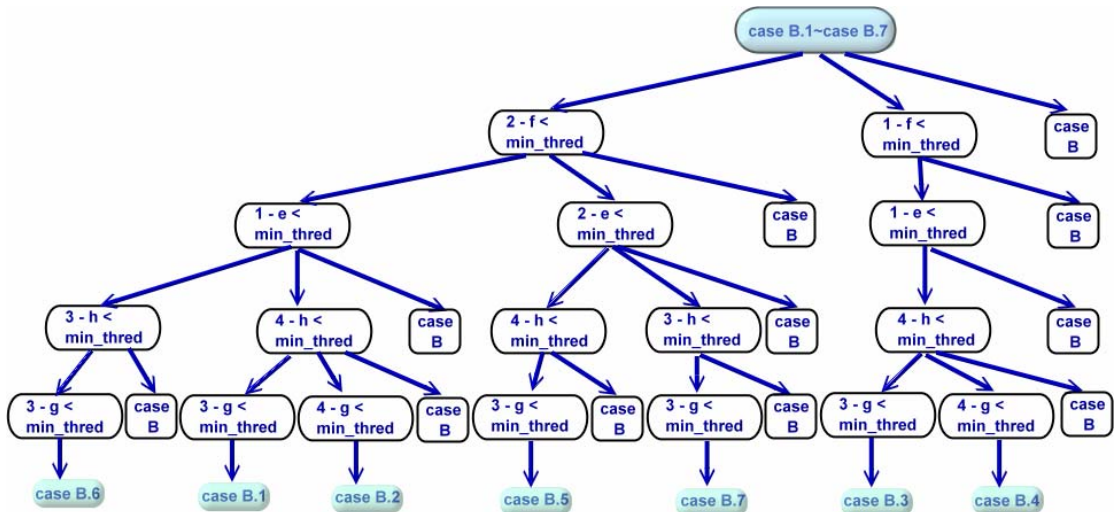


圖 3.44 對 Case B.1~Case B.7 再進一步判斷是那個種類的邊緣。

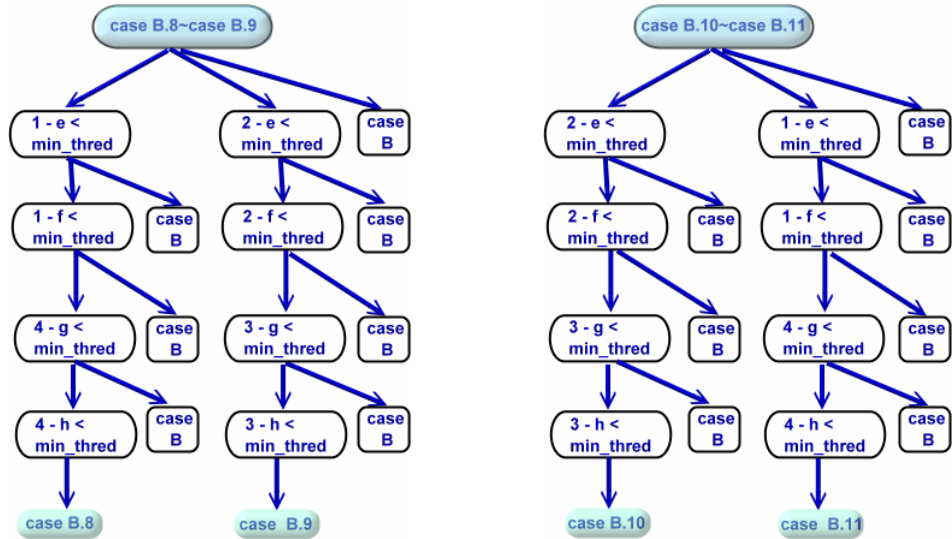


圖 3.45 對 Case B.8、Case B.9、Case B.10、Case B.11 再進一步判斷是那個種類的角。

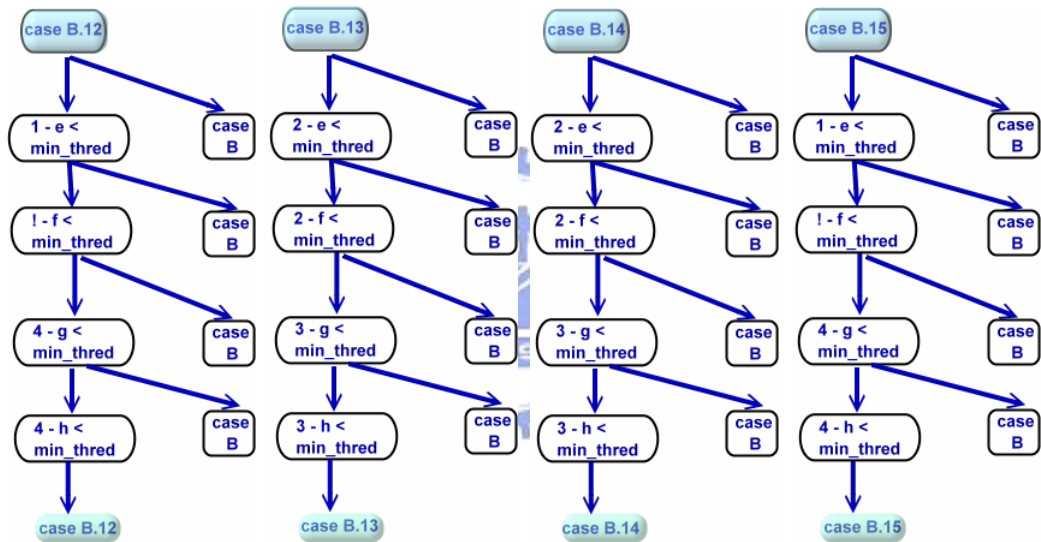


圖 3.46 對 Case B.12、Case B.13、Case B.14、Case B.15 再進一步判斷是那個種類的角。

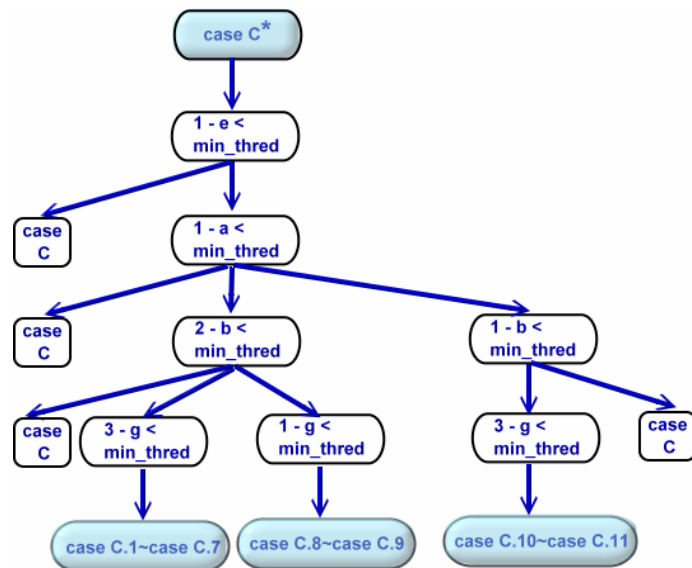


圖 3.47 對 Case C\*再進一步判斷是那個種類的邊緣或是角（共十一種）。



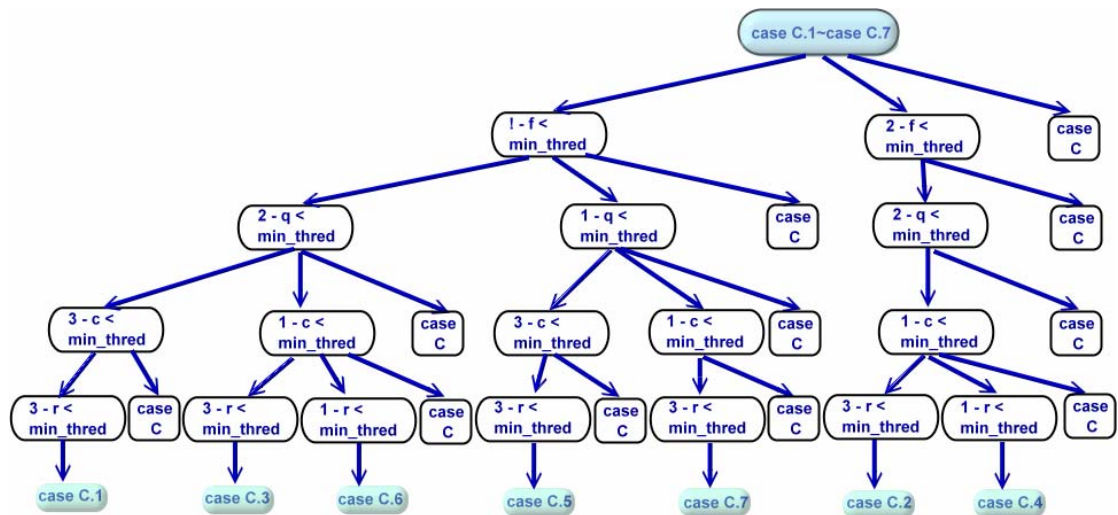


圖 3.48 對 Case C.1~Case C.7 再進一步判斷是那個種類的邊緣。

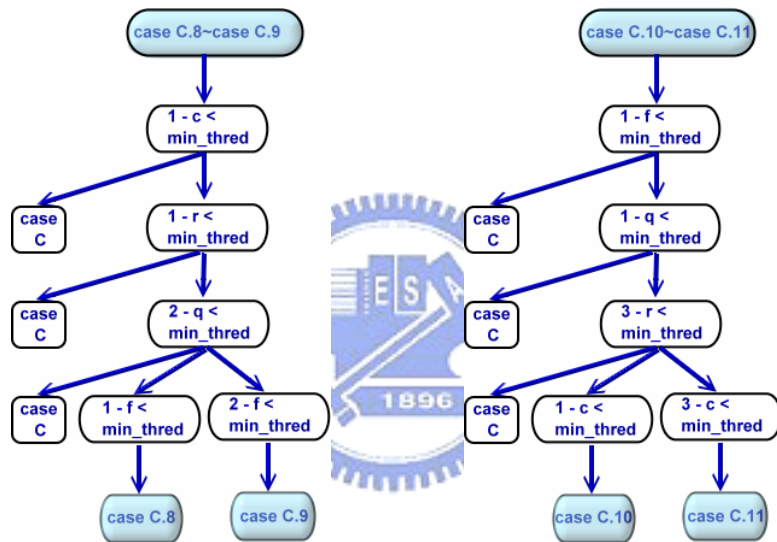


圖 3.49 對 Case C.8、Case C.9、Case C.10、Case C.11 再進一步判斷是那個種類的角。

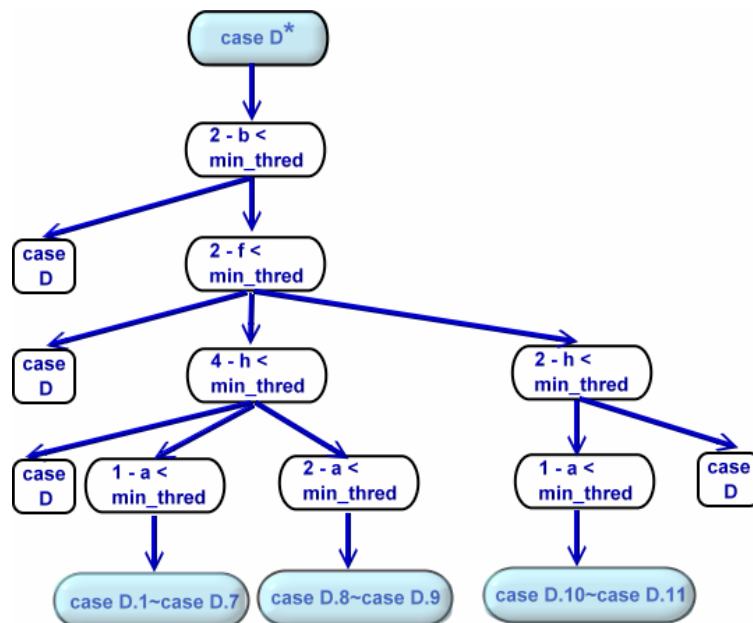


圖 3.50 對 Case D\*再進一步判斷是那個種類的邊緣或是角（共十一種）。

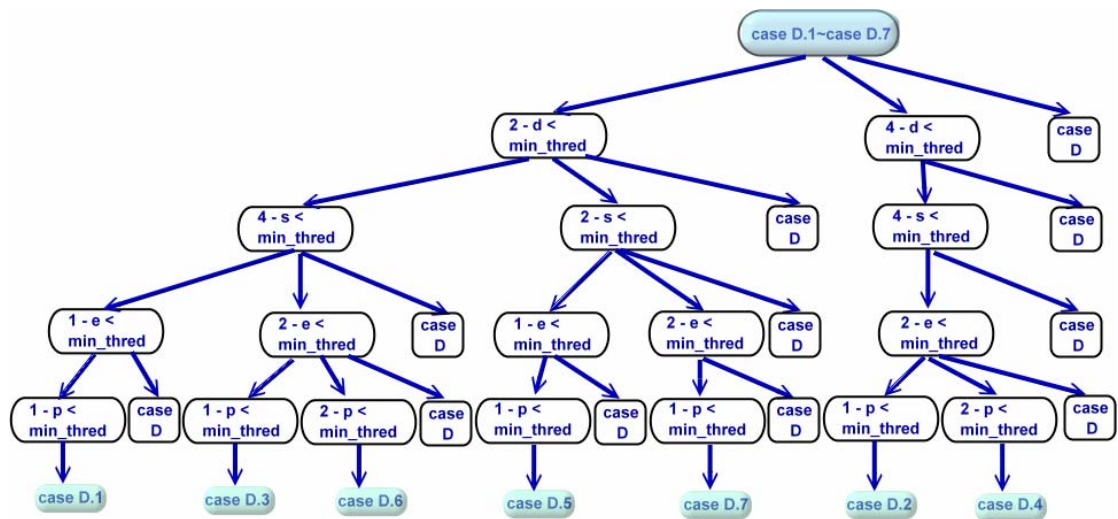


圖 3.51 對 Case D.1~Case D.7 再進一步判斷是那個種類的邊緣。

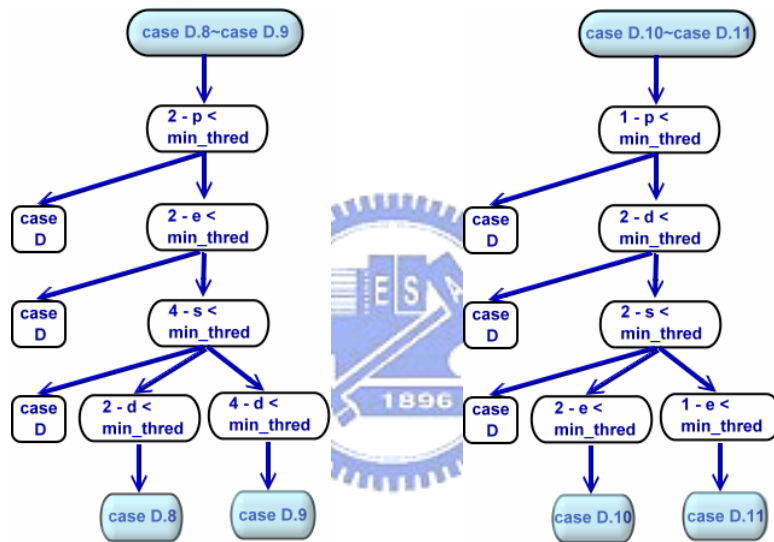


圖 3.52 對 Case D.8、Case D.9、Case D.10、Case D.11 再進一步判斷是那個種類的角。

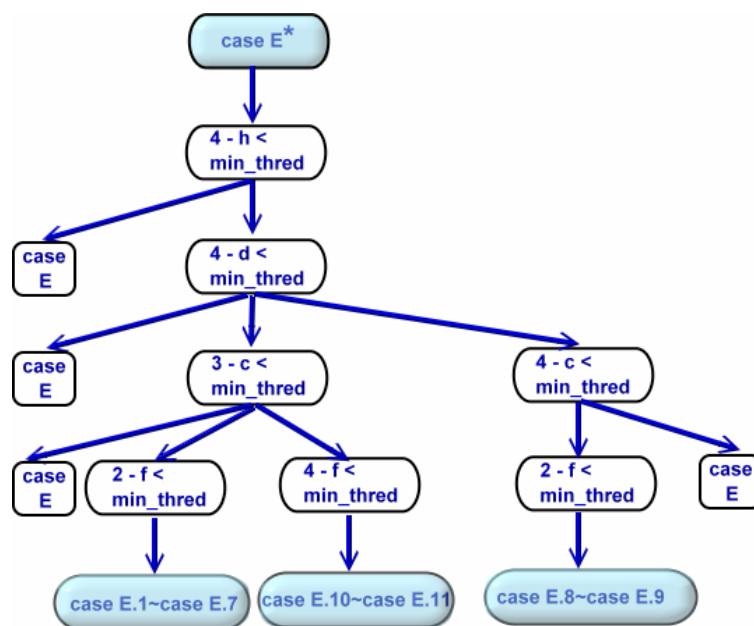


圖 3.53 對 Case E\*再進一步判斷是那個種類的邊緣或是角（共十一種）。

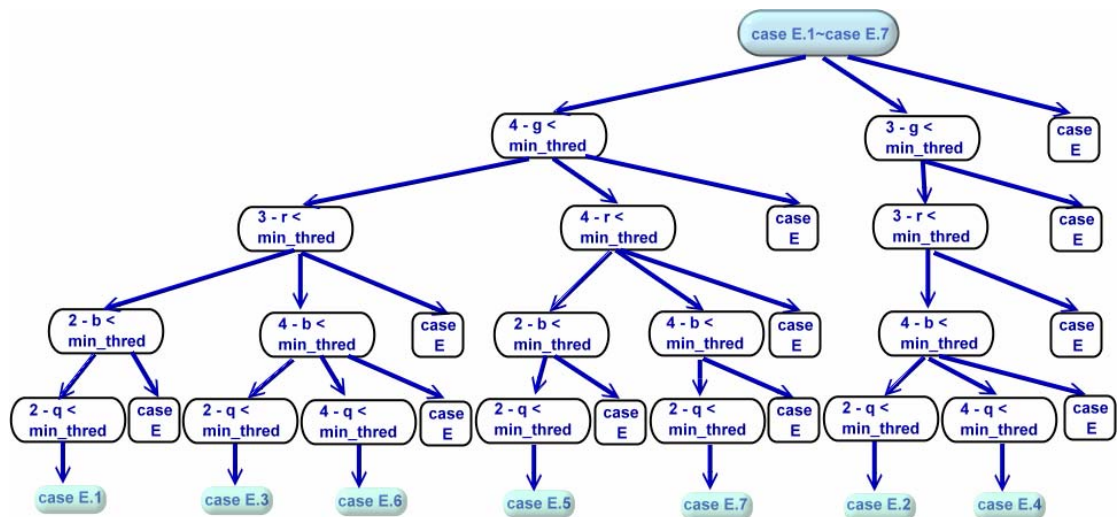


圖 3.54 對 Case E.1~Case E.7 再進一步判斷是那個種類的邊緣。

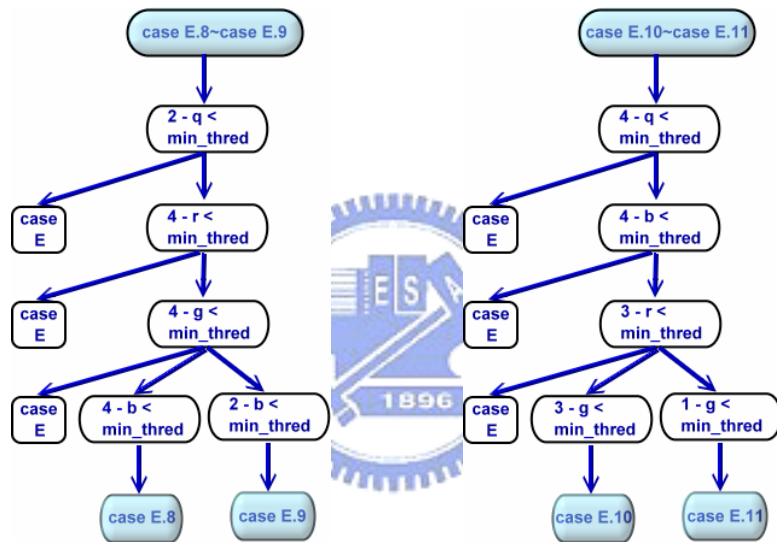


圖 3.55 對 Case E.8、Case E.9、Case E.10、Case E.11 再進一步判斷是那個種類的角。

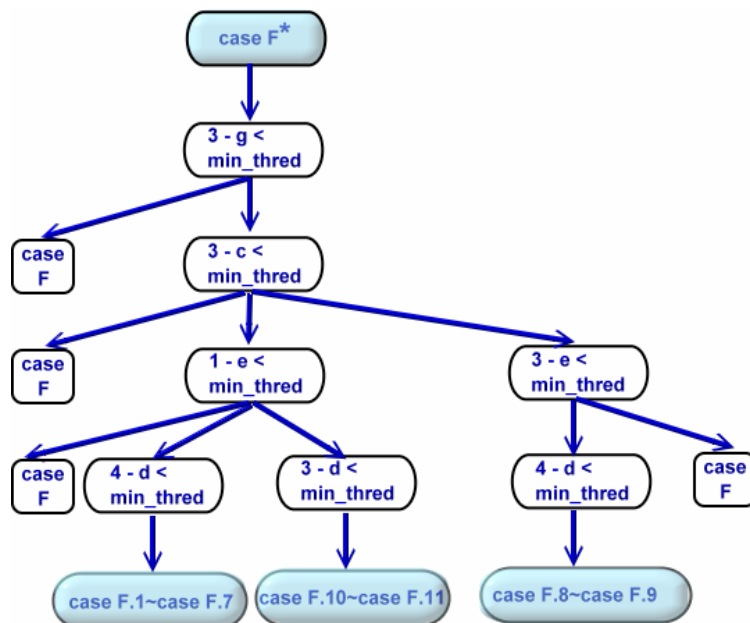


圖 3.56 對 Case F\*再進一步判斷是那個種類的邊緣或是角（共十一種）。

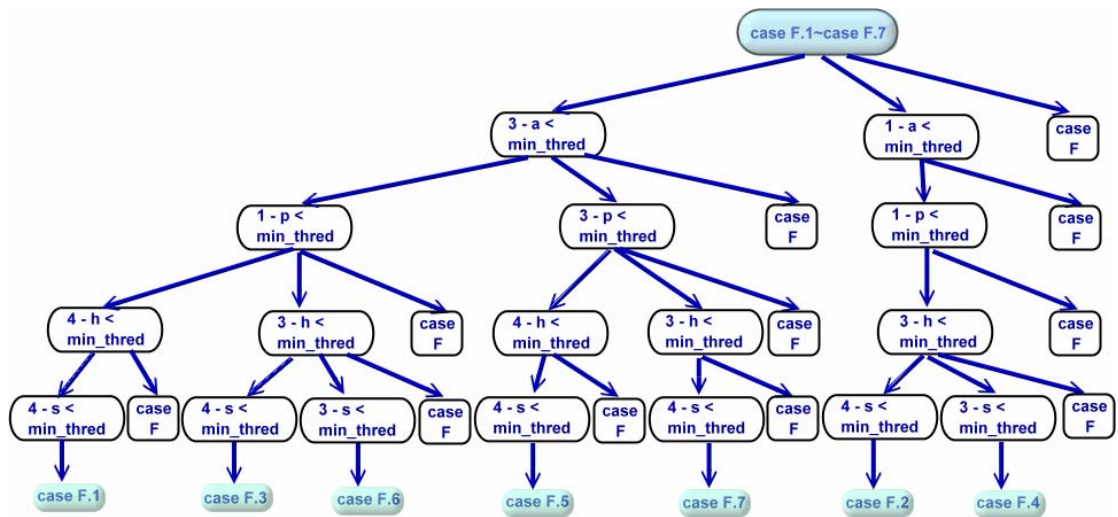


圖 3.57 對 Case F.1~Case F.7 再進一步判斷是那個種類的邊緣。

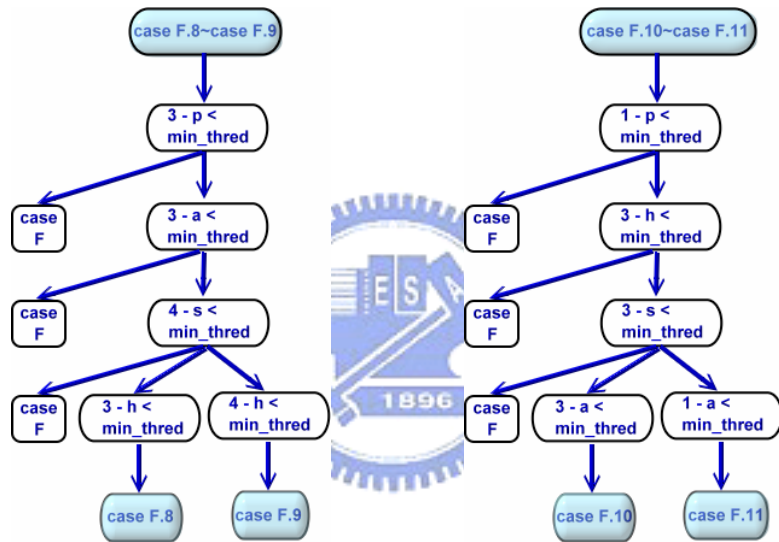


圖 3.58 對 Case F.8、Case F.9、Case F.10、Case F.11 再進一步判斷是那個種類的角。

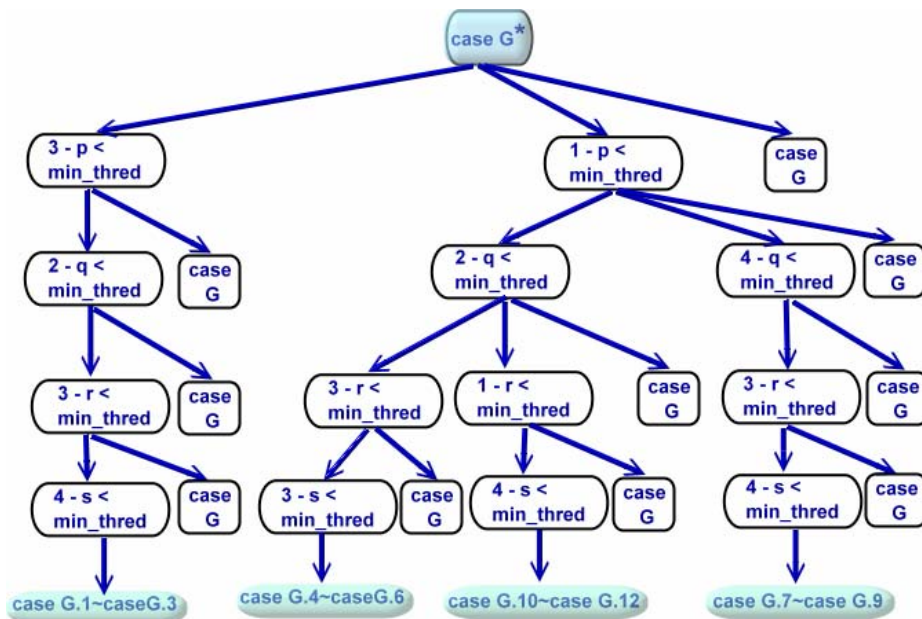


圖 3.59 對 Case G\*再進一步判斷是那個種類的角（共十二種）。

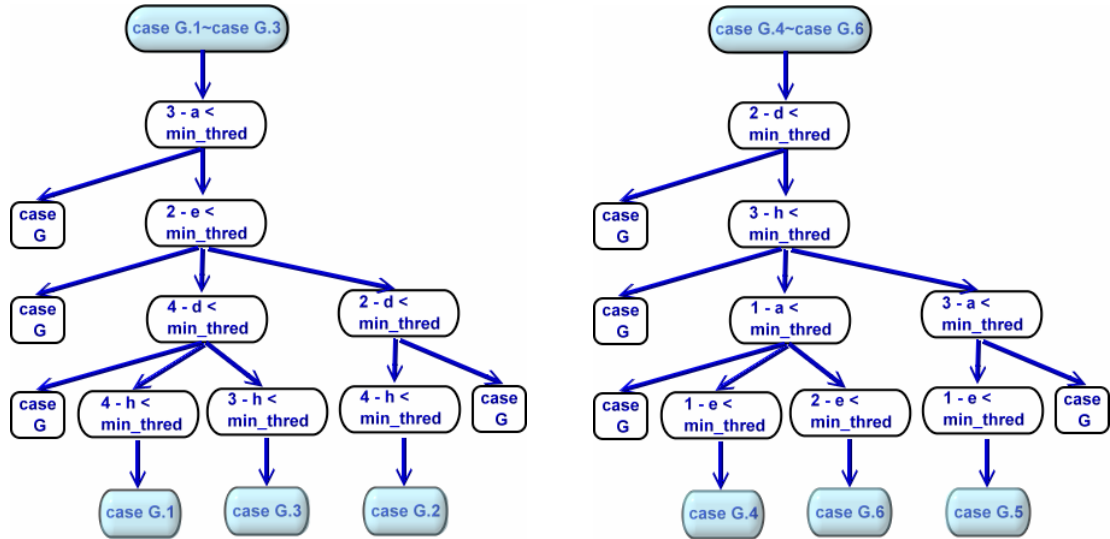


圖 3.60 對 Case G.1~Case G.6 再進一步判斷是那個種類的角。

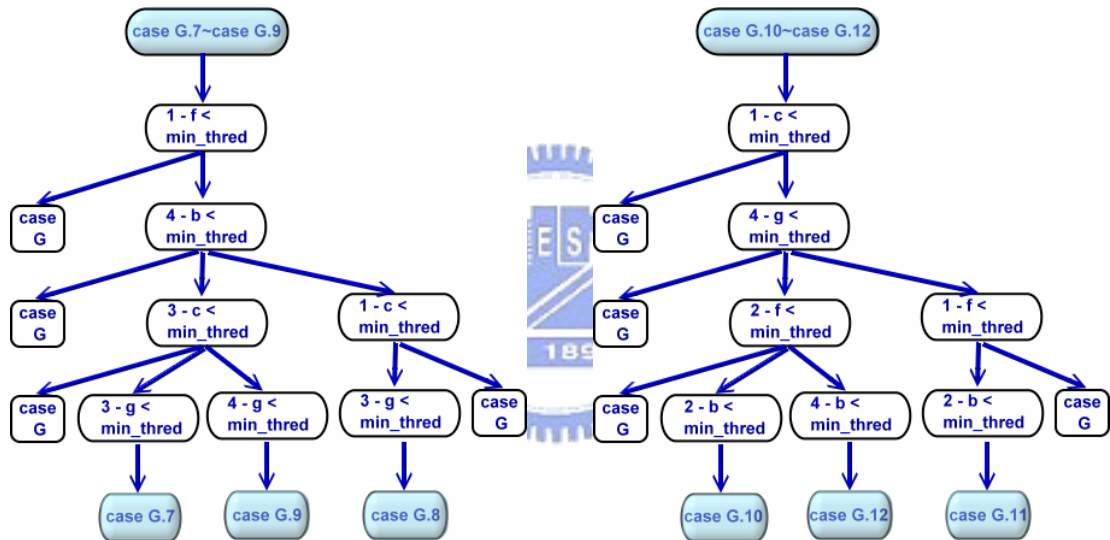


圖 3.61 對 Case G.7~Case G.12 再進一步判斷是那個種類的角。

要放大影像時，若是判斷不符合我們預先定義的邊緣或是角，則直接做 Bi-Cubic 內插補點。若是符合我們預先定義的邊緣或是角，則要先找出每種預先定義出的邊緣或是角的方程式  $f(x)$  (如圖 3.30 到圖 3.36 中綠色的線)，之後，再根據要被外插的點落的區域是屬於哪個區域做計算。以圖 3.62 為例，如式 (7) 所示，若是要被外插的點落在灰色區塊，則是用點 1、點 2 以及點 3 的平均，當作這個要被外插點的像素值；反之，若是要被外插的點落在粉紅色區塊，則用點 4 當作這個要被外插點的像素值。

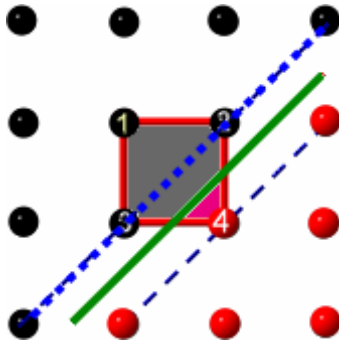


圖 3.62 綠線為邊緣出現的地方  $f(x)$ 。



# 第四章 討論

本論文中的每種方法的一開始都會預先定義邊緣以及角，因此，在這個章節我們主要討論兩個與預先定義邊緣以及角相關的議題，(1) 分類的完整性，(2) 邊緣的連續性。其中第一個議題是要討論上一個章節所預先定義的邊緣以及角是否涵蓋了所有邊或是角的種類，而第二個議題是要討論利用這樣預先定義出來的邊緣，做完影像外插放大之後，邊緣或是角是否都能連接的上。

## 4.1 分類的完整性

在分析原始影像的小區塊為 $2 \times 2$ 大小時，由於原始影像資訊只有四個參考點，因此在預先定義的種類中無法確定我們預先定義的是邊緣或是角，所以在這個方法中，我們只做最基本的分類，要決定外插補點出是邊或是角的樣子則留給判斷之後的邊緣方程式去決定。若是要將四個原始參考點分成兩群的話，只會有兩種可能性，第一種是這兩群各自包含兩個參考點；第二種是一群包含有三個參考點，另一群只有一個參考點。第一種中又可以再被分為三種分類，如圖4.1分別為水平、垂直、對角，而第二種又可以再被分為圖4.2中的四種分類。若是只有分成兩群，則這樣的分類就完全涵蓋了所有小區塊為 $2 \times 2$ 大小時的可能性。

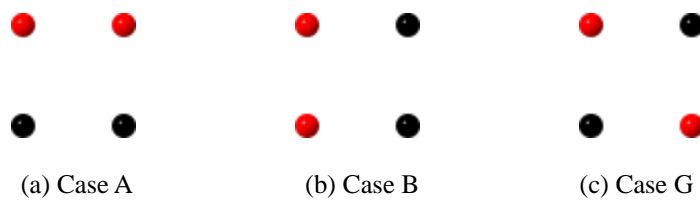


圖 4.1 兩群各包含兩個參考點。



圖 4.2 一群包含有三個參考點，另一群只有一個參考點。

在分析原始影像的小區塊為 $4 \times 2$ 大小，要預先定義出各種可能出現的邊緣、角或是線段的種類時，我們只考慮邊緣的各種狀況，而忽略角以及線段的狀況。這個方法一開始會先從中間四個點開始判斷，因此我們從中間四個點預先定義種類開始討論，做完初步的分類後，再分別仔細討論每個種類後向外擴張的部分。由於在小區塊大小為 $4 \times 2$ 時，我們預先定義的種類不考慮角以及線段的狀況，只考慮邊緣的狀況，因此，中間四個點在我們預先定義的種類中比起小區塊為 $2 \times 2$ 的大小時，少了圖4.1(c)的對角的種類Case G，因為對角線的狀況，只會產生角或是交線的狀況，因此，我們在這個方法中，中間四個點的預先定義種類只有Cases A到F。在初步分類之後，向外擴張分析時，因為只考慮邊緣的狀況，而不考慮角以及線段的狀況，且這個邊緣必須通過中間四個點圍起來的範圍，加上又受限於垂直方向少了許多資訊，因此在Case A中，可以完整的預先定義出七種類別的邊緣；而在Case B中，只能預先定義出一種垂直邊緣；在Cases C到F中也只能預先定義出兩種邊緣。這樣方法的定義中，沒辦法涵蓋的是角以及線段的狀況，但是邊緣的所有狀況都已經涵蓋在我們預先定義的種類裡面。

在分析原始影像的小區塊為 $4 \times 4$ 大小，要預先定義出各種可能出現的邊緣、角或是線段的種類時，我們會考慮邊緣以及角的各種狀況，而忽略線段的狀況。這個方法一開始會先從中間四個點開始判斷，因此我們從中間四個點預先定義種類開始討論，做完初步的分類之後，再分別仔細討論每個種類向外擴張的部分。而在小區塊大小為 $4 \times 4$ 時，中間四個點在我們預先定義的種類中與小區塊為 $2 \times 2$ 的大小時皆完全相同。因為考慮邊緣以及角的狀況，而不考慮線的狀況，且這個邊緣必須通過中間四個點圍起來的範圍，角的頂端也必須是出現在中間四個點上，因此只有在Case A到F中，可以完整的預先定義出七種類別的邊緣還有八種或是四種類別的角；而在Case G中只能定義出角的類別，而無法有邊緣的類別，因此在Case G中一共會有十二種角的類別。在邊緣必須通過中間四個點圍起來的範圍以及角的頂端必須出現在中間四個點上的兩個限制下，我們在大小為 $4 \times 4$ 小區塊的預先定義中，已經涵蓋了所有的可能性。



## 4.2 邊緣的連續性

在上個章節中，我們預先定義出許多種邊緣或是角的種類，而在每一種邊緣或是角的定義種類中，我們都有定義出邊緣或是角可能出現的地方，且之後會依據這些預先定義出來的資訊做外插計算。每一次外插或是內插補點的計算，只會針對落在中間四個原始影像參考點圍起來範圍中的未知點做計算，因此，對於一張經過放大強化的影像來說，便是將每個中間四個點圍起來大小的一張小影像直接拼湊起來，並沒有考慮附近小張影像的邊緣或是角結束的地方，因此在放大強化過後的影像，有可能會有一些較不平整的邊緣出現。

影響到放大強化後影像的邊緣或是角連續性的原因主要有兩個，第一是分析的小區塊中原始影像參考點數量的多寡；第二是預先定義出的邊緣或是角的種類多寡。在第一個原因中，本論文提出了三種不同大小分析原始影像的區塊大小，在這三種方法中，我們可以發現，區塊中的原始影像參考點越多，則放大強化後影像邊緣的連續性會越好。因為區塊中包含越多的原始影像參考點資訊，則我們可以預先定義出的邊緣以及角的種類越多，所以在放大強化影像時，可以表現出的邊緣以及角的種類也越多，因此放大強化後的影像在邊緣或是角的部分也會比較有連續性。而在第二個原因中，本論文提出的三個方法所包含的邊緣或是角的種類並不一樣多，但是在每個方法中，若是預先定義的邊緣或是角的種類越多，則放大強化過後的邊緣或是角的連續性也會越好。因此由上面兩個原因可以發現，只要是預先定義的邊緣或是角的數量越完整，也就是可以表現出的邊緣或是種類越多，則放大強化過後的影像在邊緣或是角的連續性就會越好。

## 第五章 實驗結果

在這個章節中，會比較第三章中所提到的三種同時放大以及強化影像方法的所有實驗結果。在實驗結果的比較，我們採用人眼視覺的評比取代數據上的評比。由這些實驗結果我們可以發現，若是分析原始影像的小區塊越小，對於同一張影像來說，越容易做外插的計算，因為預先定義的種類中，只有較少的參考點，因此較容易滿足做外插計算的條件，但是也由於參考點較少的關係，因此在預先定義的種類也會較少，所以在邊緣或是角的連續性會表現的較差；反之，若是分析原始影像的小區塊越大，則越不容易做外插的計算，因為預先定義的種類中，含有較多的參考點，因此較不容易滿足做外插計算的條件，但是也由於參考點較多的關係，因此在預先定義的種類會較多，所以在邊緣或是角的連續性會表現的較好。



### 5.1 判斷 2x2 區域內邊的實驗結果

如之前章節所討論過的，當分析原始影像小區塊為2x2大小時，由於小區塊中參考點只有四個參考點，因此預先能定義出的邊緣種類也較少，所以圖5.1中的三個放大強化方法作出來的結果都沒辦法有很平整的邊緣，但也是因為小區塊中參考點較少的原因，所以在放大強化影像的過程中較容易符合條件做外插放大，所以較不容易看到模糊的邊緣或是角。圖5.1中(a)是原始影像，圖5.1 (b)到(d)是放大後的影像，在這放大後的三張影像中，最主要不一樣的地方在紅色圈起來的地方：圖5.1 (b)是第一個方法作出來的結果，會讓紅色圈中的尖角角度比較垂直，圖5.1 (c)是第二個方法作出來的結果，會讓紅色圈中的尖角較平滑一點，圖5.1 (d)是第三種方法作出來的結果，會讓紅色圈中的尖角更加平滑一點。而在圖5.2的實驗結果中可以發現，屋簷的地方，有較明顯的不同，圖5.2(a)是原始影像，圖5.2 (b)到(d)為放大過後的影像，其中圖5.2 (b)是經由第一種方法放大過後的結

果，其效果類似於Nearest Neighbor影像放大方法，所以在屋簷的地方呈現鋸齒狀，而圖5.2 (c)是經由第二種方法放大過後的結果，會讓邊比較緩和，所以看到屋簷的邊緣就比較平順一點，圖5.2 (d)是經由第三種方法放大過後的結果，會讓邊更加緩和，所以看到屋簷的邊緣就更加平順。

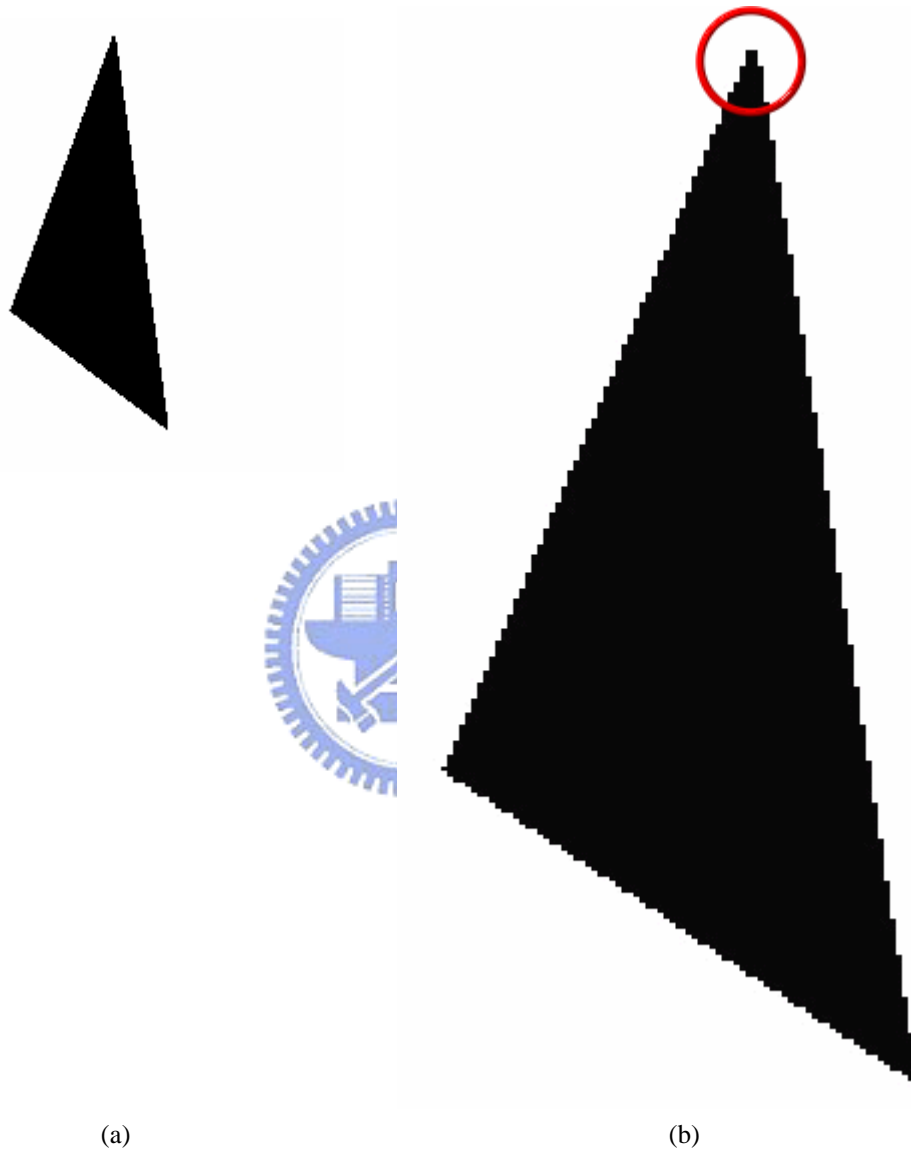
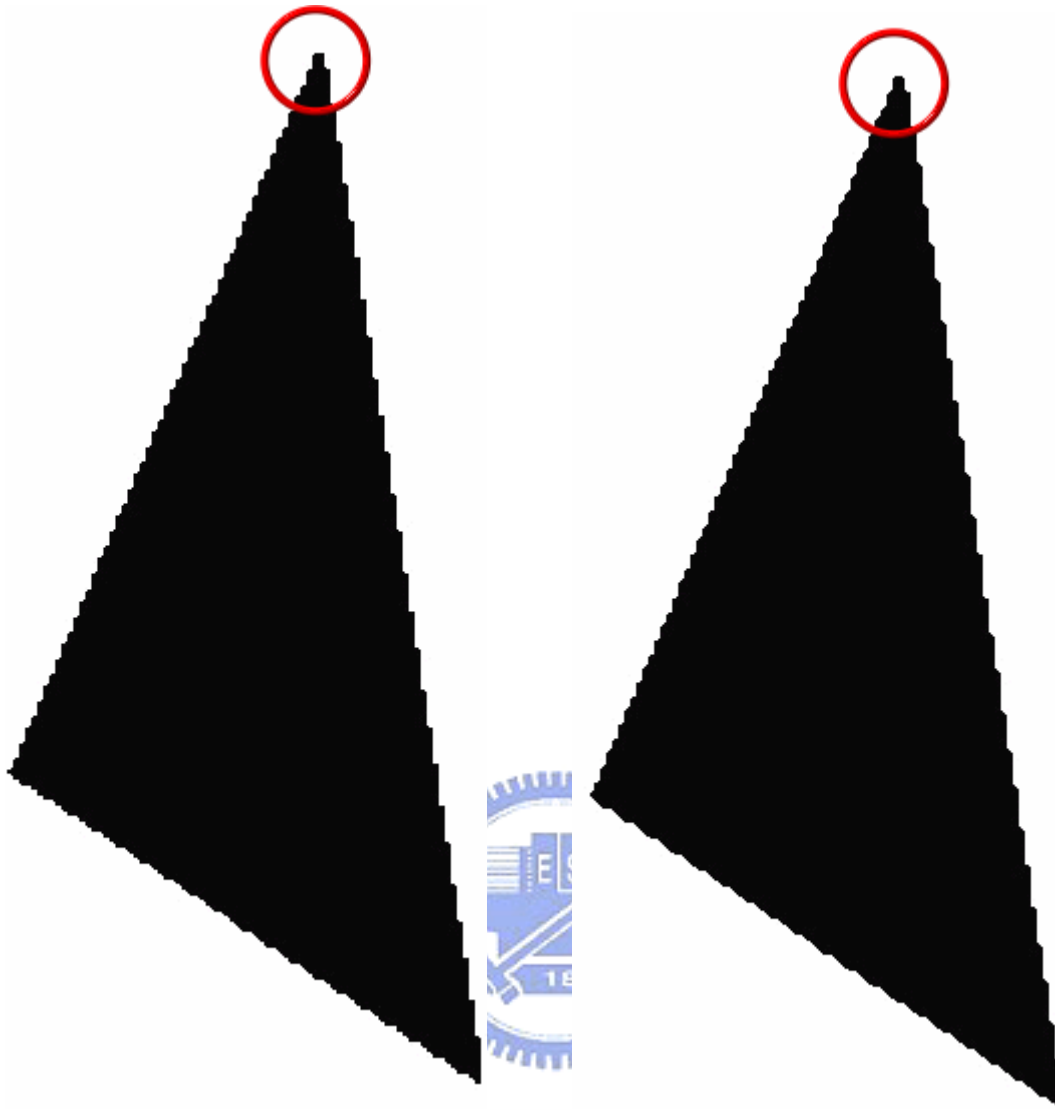


圖 5.1 (a) 是原始影像為像素  $268 \times 305$  的三角形，(b) 是(a)經圖3.15 中的藍線放大到像素為  $800 \times 800$  的三角形，(c) 是(a)經圖3.15 中的紅線放大到像素為  $800 \times 800$  的三角形，(d) 是 (a) 經圖3.15 中的綠線放大到像素為  $800 \times 800$  的三角形。



(c)

(d)

圖 5.1 (續)



(a)



(b)

圖 5.2 (a) 是原始影像為像素  $155 \times 116$  的建築物，(b) 是(a)經由圖3.15 中的藍線放大到像素為  $800 \times 800$  的建築物，(c) 是經由圖3.15 中的紅線放大到像素為  $800 \times 800$  的建築物，(d) 是 (a)經由圖3.15 中的綠線放大到像素為  $800 \times 800$  的建築物。



(c)

圖 5.2 (續)



(d)

圖 5.2 (續)

## 5.2 判斷 4x2 區域內邊的實驗結果

當分析原始影像小區塊為 4x2 的大小時，由於小區塊中參考點增加為八個點，預先定義的邊緣種類也增加為十六種，因此就如第四章所得到的結論—外插計算會減少，但是種類會增加。在圖 5.3(b)中我們可以發現，在這樣預先定義的條件下，一張影像，尤其是自然影像在放大過程中做外插計算的機會少很多，大部分都是做內插補點。另外，由於垂直方向預先定義的種類只有一種，所以在垂直方向的外插補點效果不如水平方向的外插補點效果，在實驗結果圖 5.4 中可以

發現角度接近水平方向的邊會比接近垂直方向的邊，要平整許多。另外，在圖 5.5 的實驗結果中，會發現白線的交叉點不會做出如原始影像垂直一般的結果，這是因為預先定義的邊緣種類不夠多所導致的錯誤計算，且由於我們在水平以及垂直方向預先定義邊的種類數目不一樣多，因此圖 5.5 (b)中紅色箭頭指的地方可以發現，有做外插的計算，但是在下一條線則沒有，也就是水平方向做外插計算比較多，而垂直方向做外插計算比較少，但是在圖 5.5 (c)中由於我們將水平方向預先定義邊的數目減少成與垂直方向預先定義邊的數目一樣多，因此在箭頭指的地方與圖 5.5 (b)有明顯的不同，亦即在水平以及垂直方向作外插計算的數量就會比較相同。在實驗結果圖 5.6 中，我們可以發現在每個直角的轉角處，與圖 5.5 一樣，並無辦法外插補點出與原始影像一樣垂直的角度。



(a)

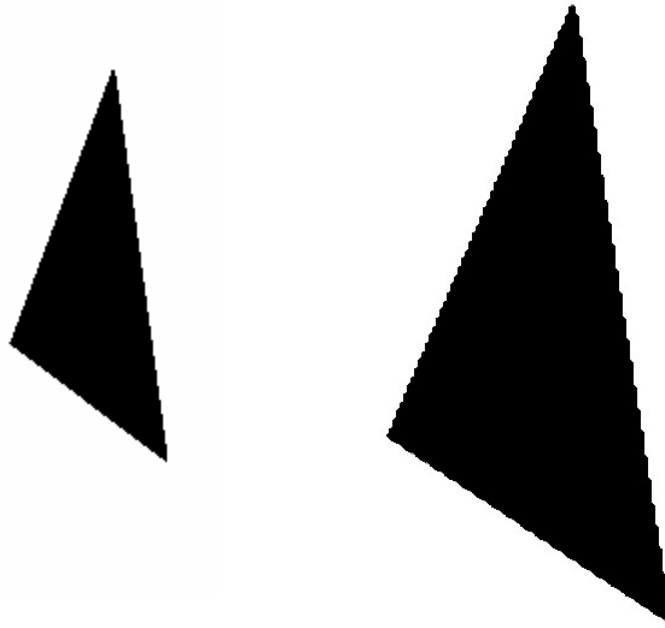
圖 5.3 由  $800 \times 600$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 以  $4 \times 2$  影像區塊作分析，所得之部分影像。





(b)

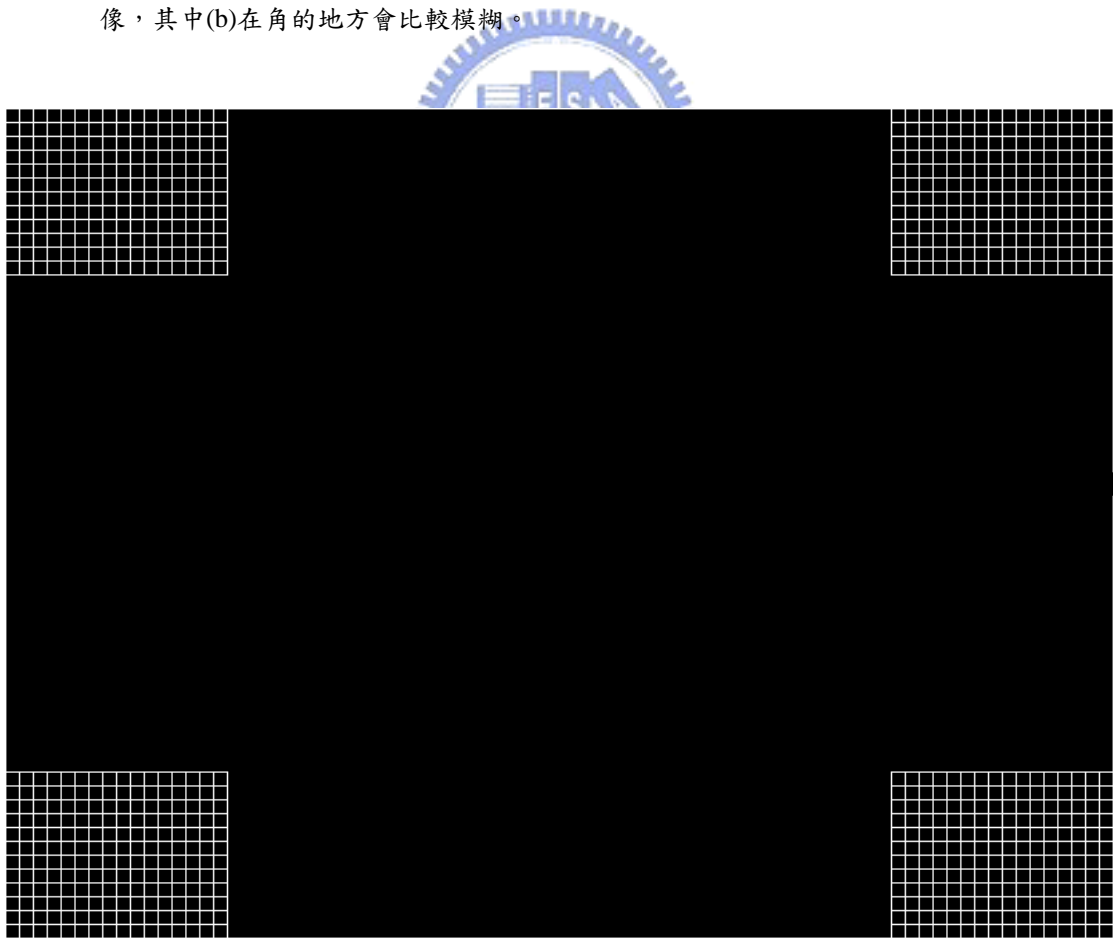
圖 5.3 (續)



(a)

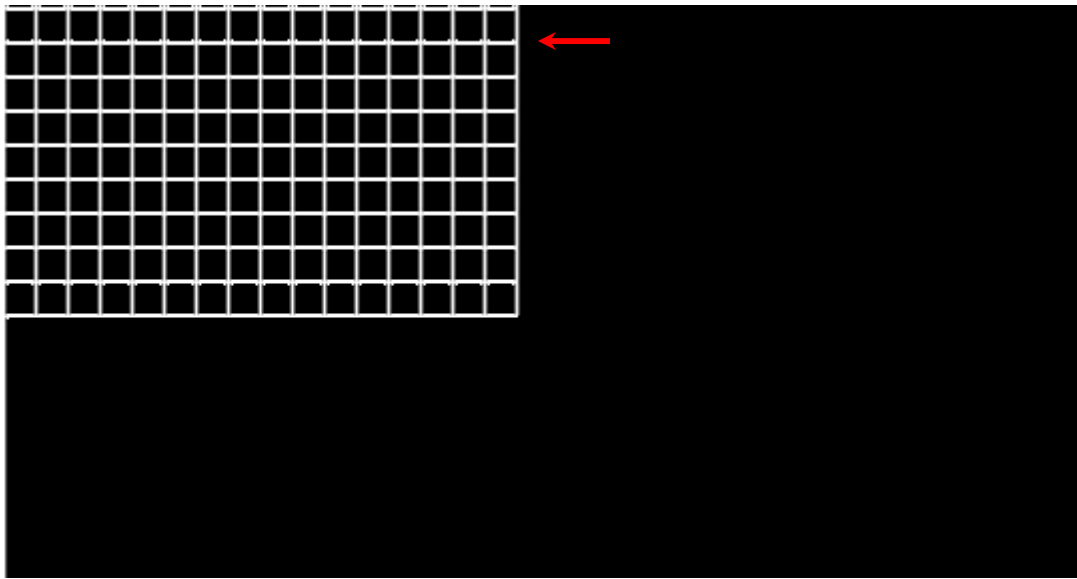
(b)

圖 5.4 由  $800 \times 600$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 以  $4 \times 2$  影像區塊作分析，所得之影像，其中(b)在角的地方會比較模糊。

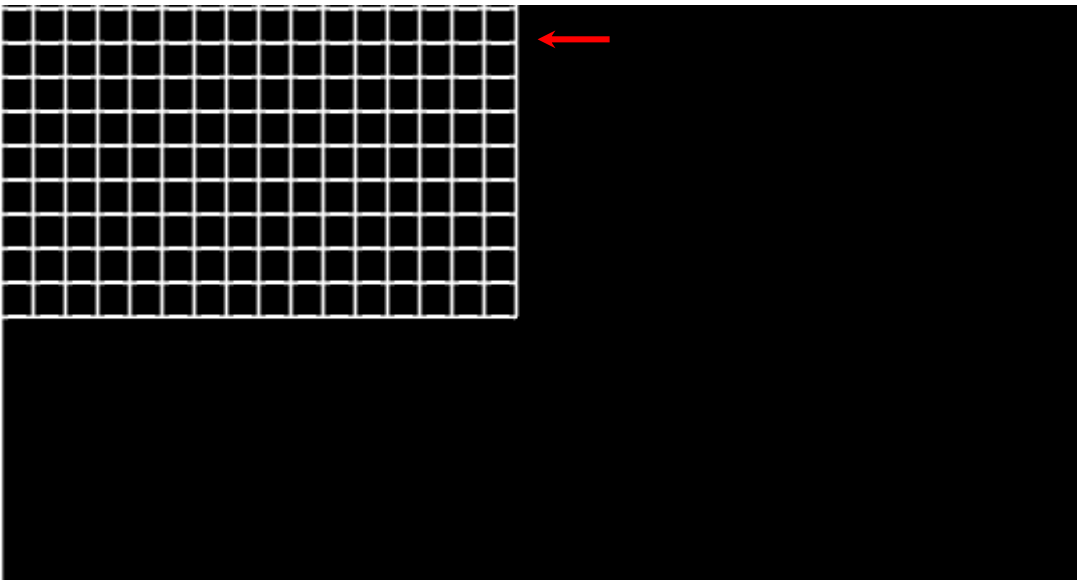


(a)

圖 5.5 由  $800 \times 600$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 以  $4 \times 2$  影像區塊作分析，所得之部分影像，(c) 將預先定義中水平方向定義的種類減少成與垂直方向種類數目一樣多。



(b)



(c)

圖 5.5 (續)

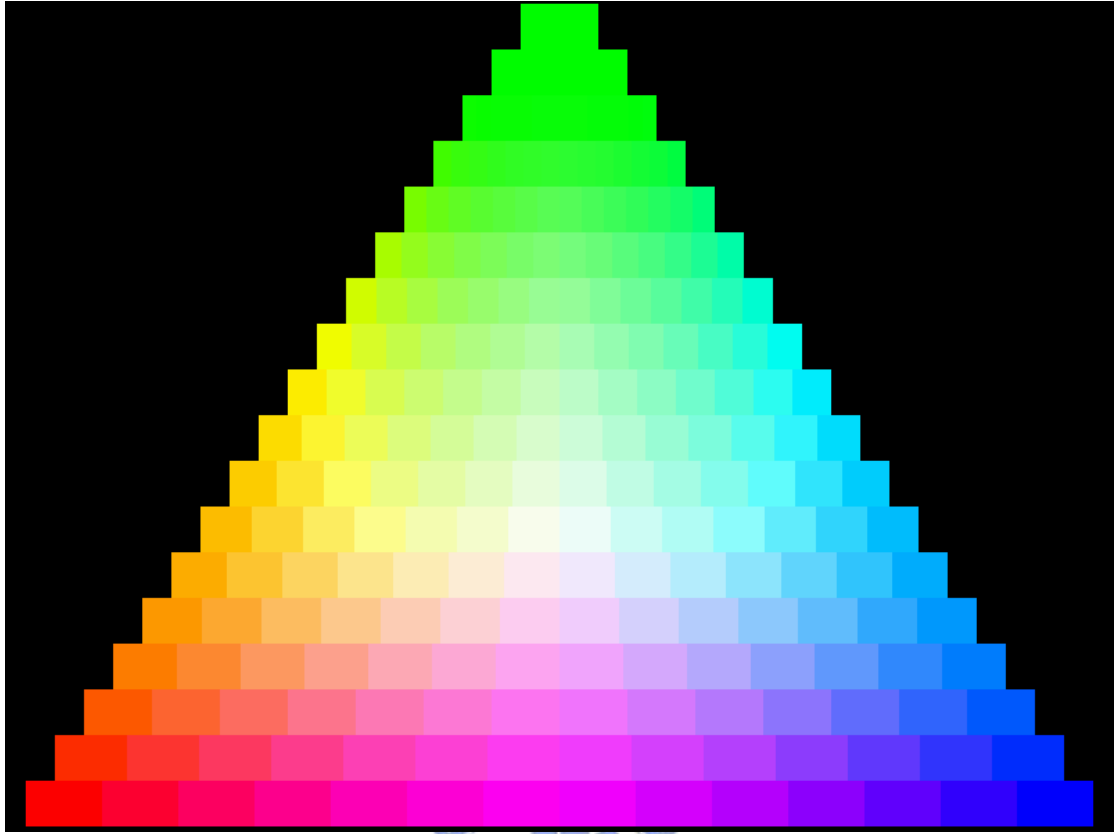
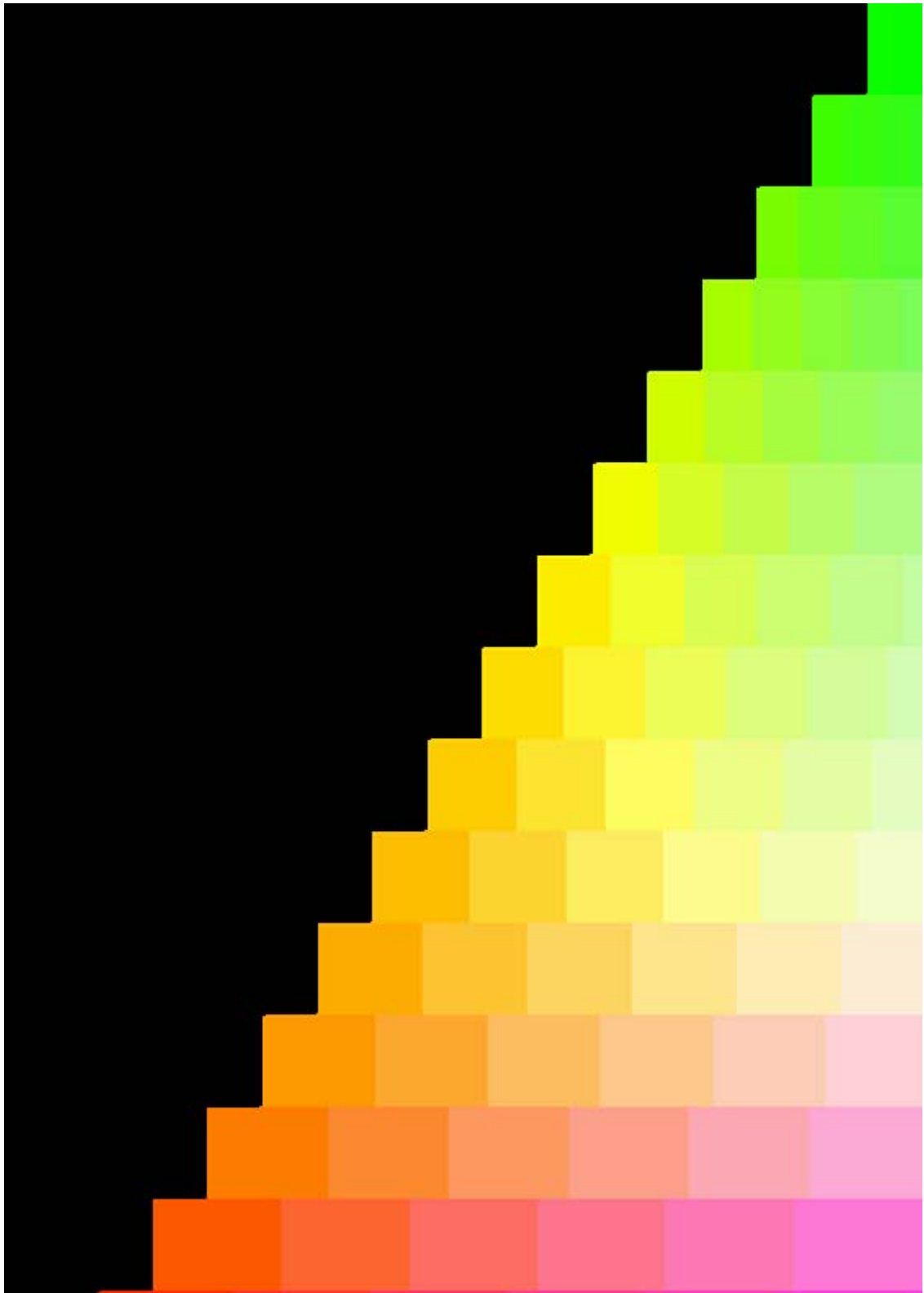


圖 5.6 由  $800 \times 600$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 以  $4 \times 2$  影像區塊作分析，所得之部分影像。



(b)

圖 5.6 (續)

### 5.3 判斷 4x4 區域內邊以及角的實驗結果

當分析原始影像小區塊為 4x4 的大小時，由於小區塊中參考點增加為十六個點，預先定義的邊緣以及角的種類也增加為八十六種，因此就如第四章所得到的結論—外插計算會比小區塊大小為 2x2 的少，但是種類會增加許多種。另一個影響到放大過程中，整張影像做外插計算量大小的因素就是在判斷過程中的臨界值，我們在判斷過程中會如圖 5.7 將中間四個點分為兩個群組，其中如圖 5.8 所示，要成為一個群組，兩點的像素值相減要小於內部臨界值 (Internal Threshold)，也就是樹狀圖中的 min\_thred; 而兩個群組間的差異值要大於外部臨界值 (External Threshold)，也就是樹狀圖中的 MAX\_thred。若是內部臨界值設的越小，外部臨界值設的越大，則條件越嚴格做外插運算的點也會越少，反之，若是內部臨界值設的越大，外部臨界值設的越小，則條件越寬鬆做外插運算的點就會越多。

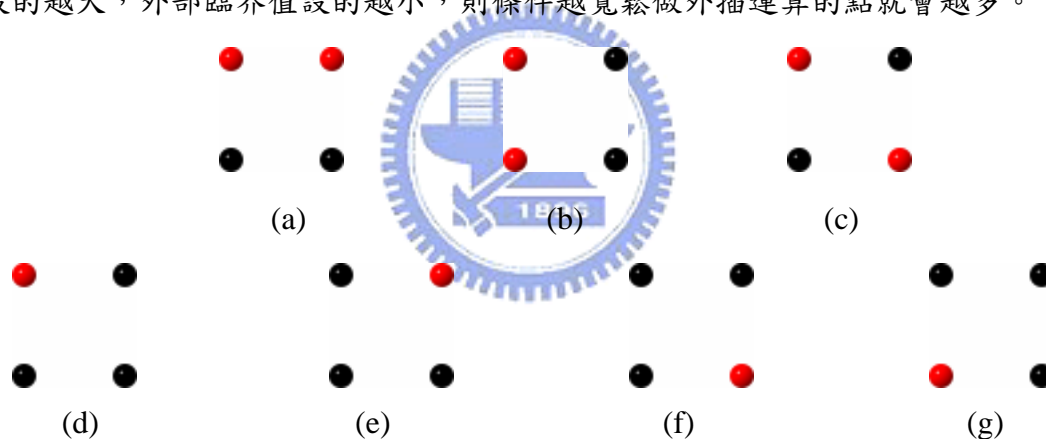


圖 5.7 將中間四個點分為兩個群組。

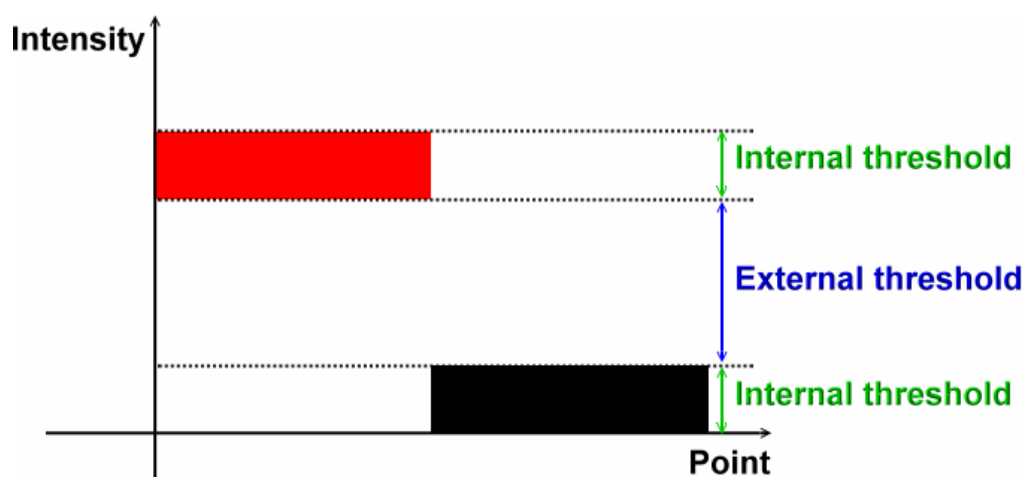


圖 5.8 內部臨界值以及外部臨界值的示意圖。

實驗結果中我們可以發現，在這樣的預先定義的條件下，可以將放大過後的邊緣變的比較平整，如圖 5.9 所示，圖 5.9 (a)是原始影像，圖 5.9 (b)是整張影像做內插補點 (Bi-Cubic)，圖 5.9 (c)是利用本論文中提出的方法所得之放大影像，內部臨界值設為 10，外部臨界值設為 250，我們可以發現在圖 5.9 (c)中的邊緣明顯的比圖 5.9 (b)中的邊緣要來的平整，看起來也比較銳利。圖 5.10(a)是原始影像，圖 5.10 (b)是整張影像做內插補點 (Bi-Cubic)，圖 5.10 (c)是利用本論文中提出的方法所得之放大影像，內部臨界值設為 10，外部臨界值設為 250，圖 5.10 (d)是標示出利用本論文中方法放大影像時，做外插計算的部分，圖 5.10 (e)是利用 Matlab 對圖 5.10 (b)做 Edge Extraction 的結果，圖 5.10 (f)利用 Matlab 對圖 5.10 (c)做 Edge Extraction 的結果。圖 5.11(a)是整張影像做內插補點 (Bi-Cubic)，圖 5.11 (b)是利用本論文中提出的方法所得之放大影像，內部臨界值設為 10，外部臨界值設為 250，其中，在白色線交叉的地方，圖 5.11 (b)效果看起來比較差。

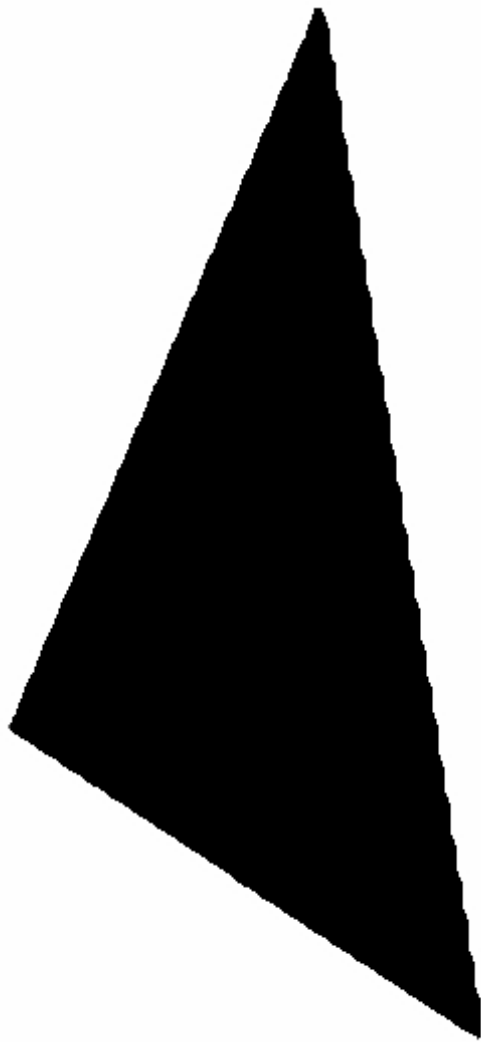
另外，從圖 5.12 這三張影像放大結果中，紅色與黃色圈起來的地方就可以發現，兩個臨界值也會影響到做外插運算的數量。在紅色圈起來的字中，圖 5.12 (a)是做內插補點計算，而圖 5.12 (b)、(c)是有做外插計算的，看起來就比較銳利一點，而在黃色圈起來的字中，圖 5.12 (a)、(b)是做內插補點，而只有圖 5.12 (c)中黃色圈起來的字，有做外插計算，因此只有圖 5.12 (c)看起來比較銳利。圖 5.13(a)是原始影像，圖 5.13 (b)是標示出利用本論文提出的方法放大影像時，做外插計算的部分，圖 5.13 (c)是做內插補點 (Bi-Cubic)，圖 5.13 (d)利用本論文中提出的方法所得之放大影像，內部臨界值設為 10，外部臨界值設為 250，其中，可以發現圖 5.13 (c)、(d)兩張圖中的角有明顯的不一樣，在圖 5.13 (d)中的邊緣不管是哪種角度都是比較平滑的，放大過後比較不會有鋸齒狀的邊緣出現。圖 5.14(a)是原始影像，圖 5.14 (b)是標示出利用本論文提出的方法放大影像時，做外插計算的部分，圖 5.14 (c)是做內插補點 (Bi-Cubic)，圖 5.14 (d)利用本論文中提出的方法所得之放大影像，內部臨界值設為 30，外部臨界值設為 120，其中，可以發現邊緣兩邊的點的像素值相減，超過外部臨界值的時候，才會做外插計算，因此不是



(a)



(b)



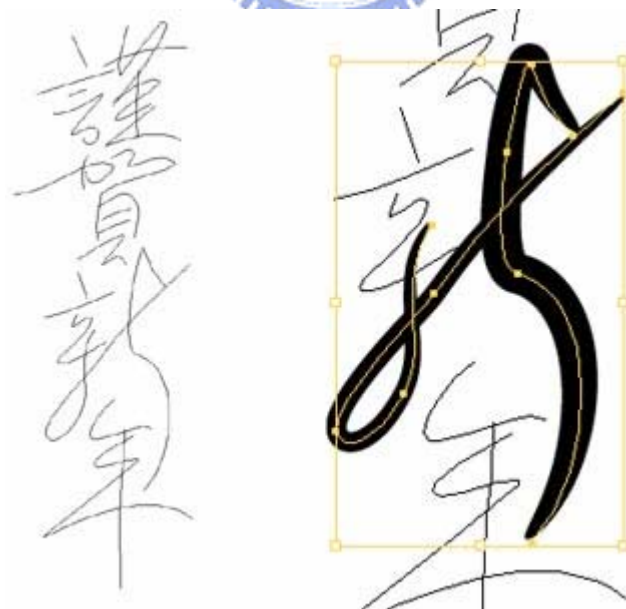
(c)

圖 5.9 由 268×305 放大到 1280×1024，(a) 原始影像，(b) 經由 Bi-Cubic 放大影像，(c) 以 4×4 影像區塊作分析，所得之影像（內部臨界值=10，外部臨界值=250）。



每個區塊的邊緣都會做外插計算，也因為這樣所以有些區塊的邊緣比較平整，而有些區塊的邊緣鋸齒狀比較明顯，且邊緣也比較模糊。圖 5.15(a)是原始影像，圖 5.15 (b)是做內插補點 (Bi-Cubic)，圖 5.15 (c)是利用本論文中提出的方法放大影像，內部臨界值設為 20，外部臨界值設為 100，(d)是利用本論文中提出的方法放大影像，內部臨界值設為 30，外部臨界值設為 40，在這個例子中，我們可以發現，臨界值設的越寬鬆，影像放大過程中做外插計算的部分也會越多，對於自然影像來講，外插計算越多的話，則會讓整張圖看起來比較不那麼真實，有點像是卡通的感覺。

因此從上面的實驗結果可以得知，分析原始影像的小區塊大小為  $4 \times 4$  時，在邊的部分比小區塊大小為  $4 \times 2$  或是  $2 \times 2$  時較平整且銳利另外在邊緣或是角上，外插運算的數量越多，就會看起來越銳利。因此藉由調整兩個臨界值，增加一張影像中外插運算的數量，也會增加放大影像的銳利度。而在自然影像的部分，若是將臨界值調整成比較極端的狀況，讓整張影像做較多的外插計算時，會讓整張圖看起來不真實。



(a)

圖 5.10 由  $330 \times 300$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 經由 Bi-Cubic 放大影像，(c) 以  $4 \times 4$  影像區塊作分析，所得之部分影像 (內部臨界值=10，外部臨界值=250)，(d) 外插計算的部分，(e) 對圖(b)的 Edge Extraction，(f) 對圖(c)的 Edge Extraction。



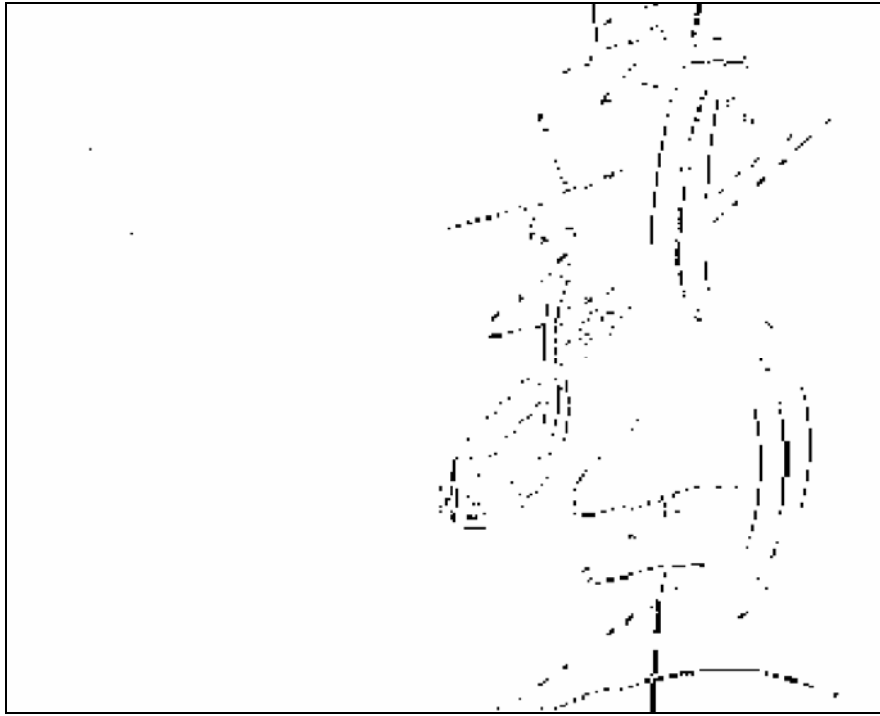
(b)

圖 5.10 (續)

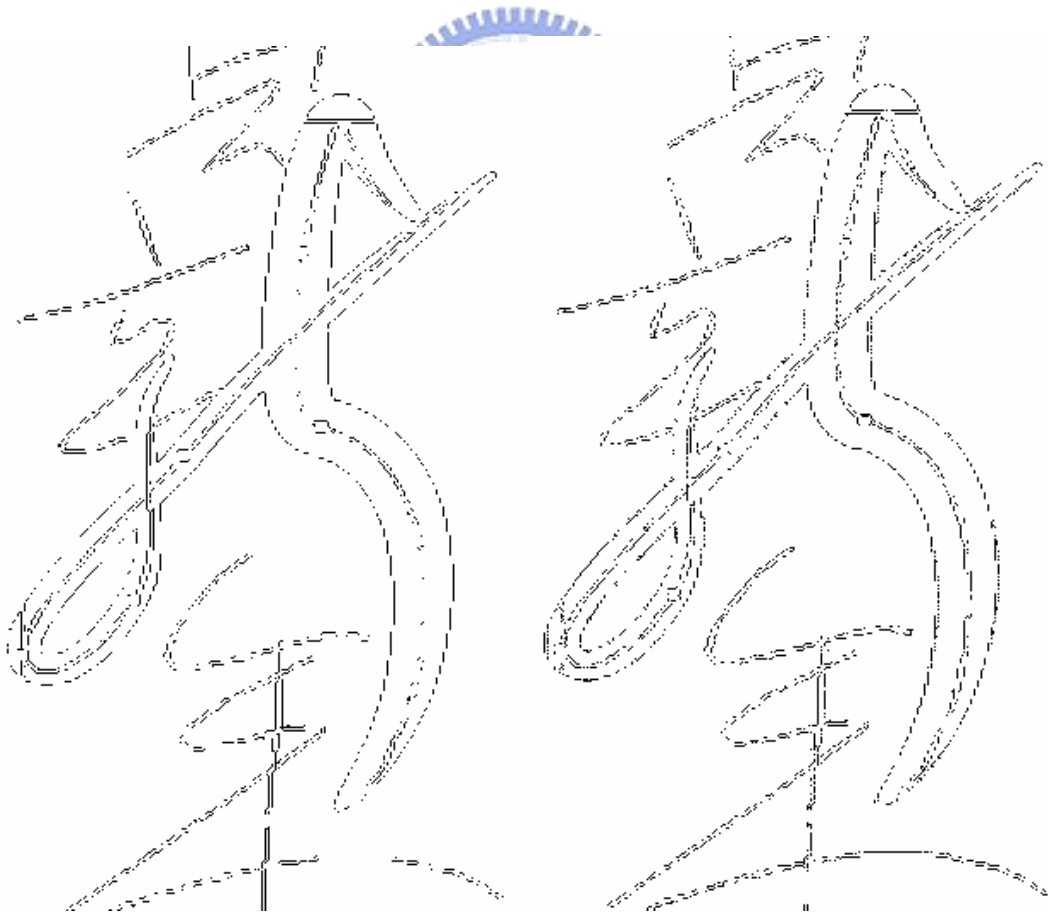


(c)

圖 5.10 (續)



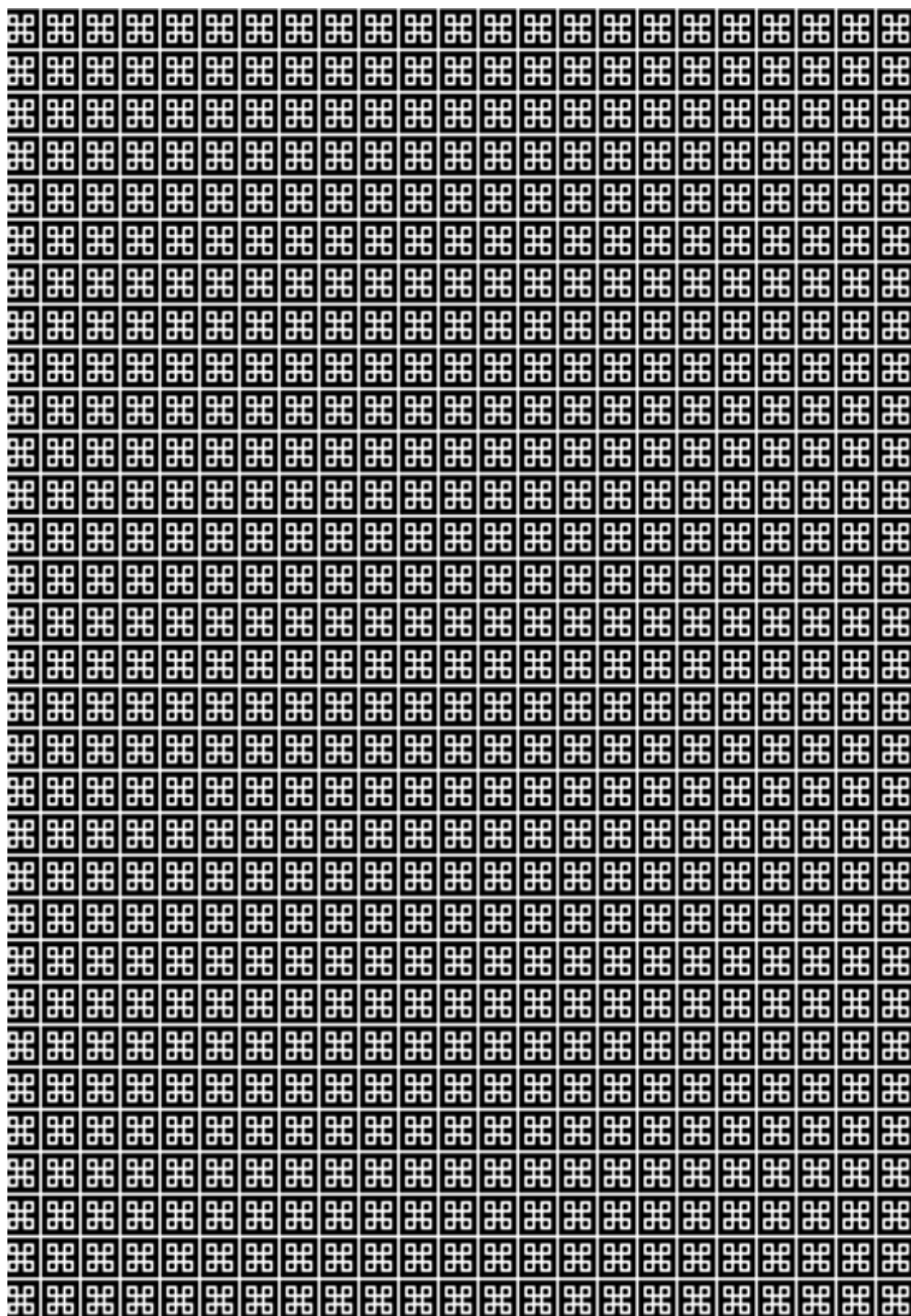
(d)



(e)

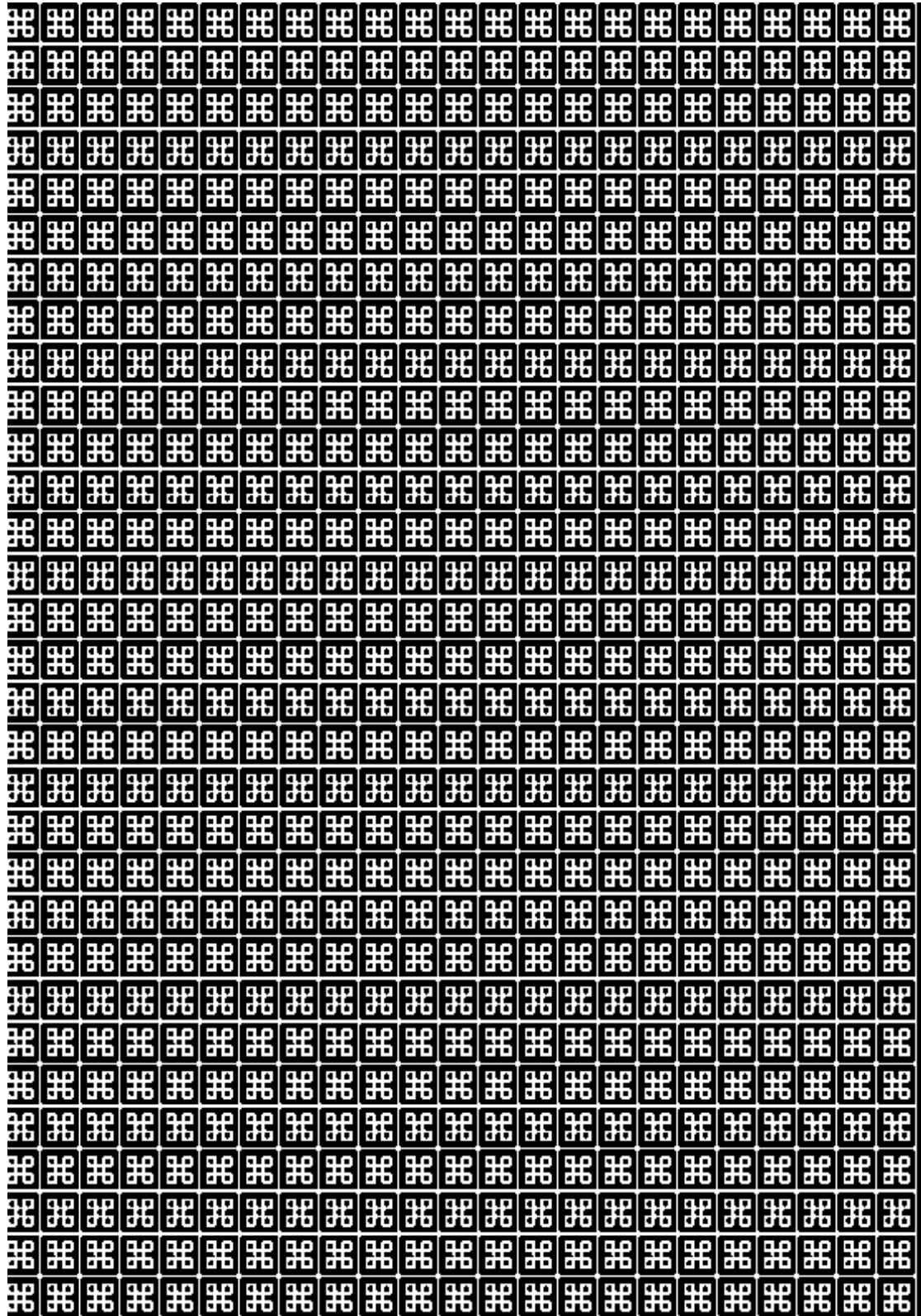
(f)

圖 5.10 (續)



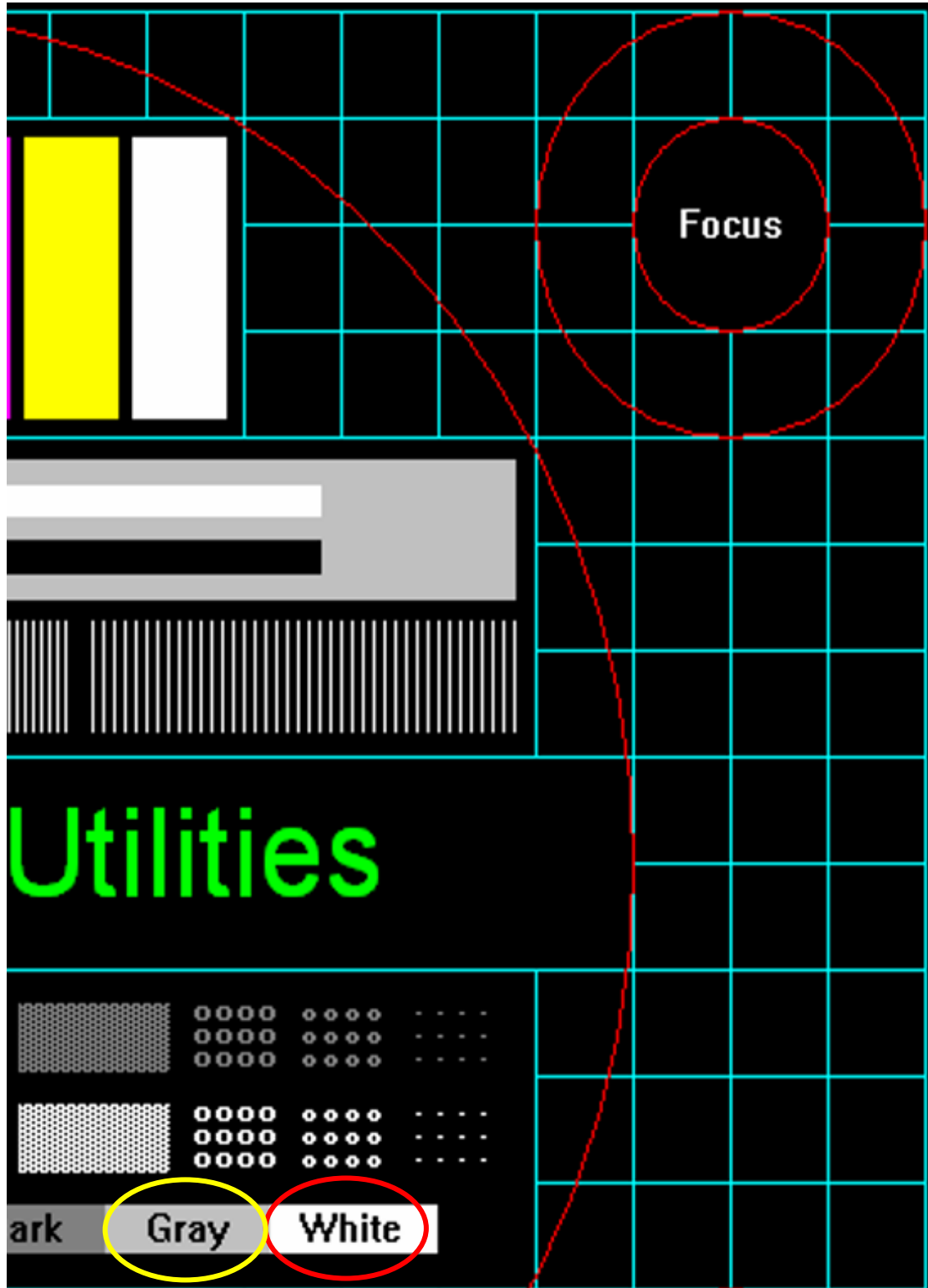
(a)

圖 5.11 由  $800 \times 600$  放大到  $1280 \times 1024$ ，(a) 經由 Bi-Cubic 放大影像，(b) 以  $4 \times 4$  影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250）。



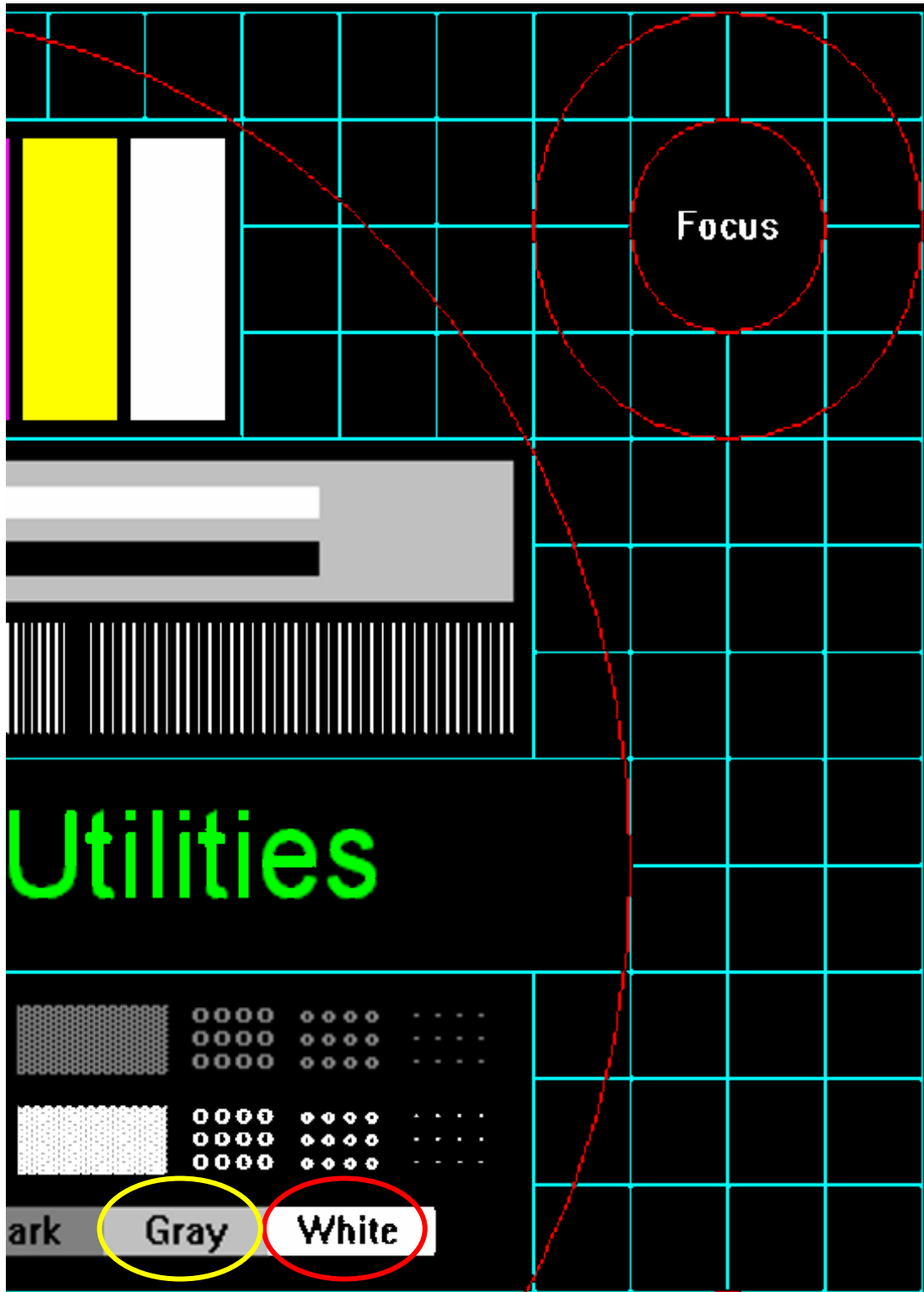
(b)

圖 5.11 (續)



(a)

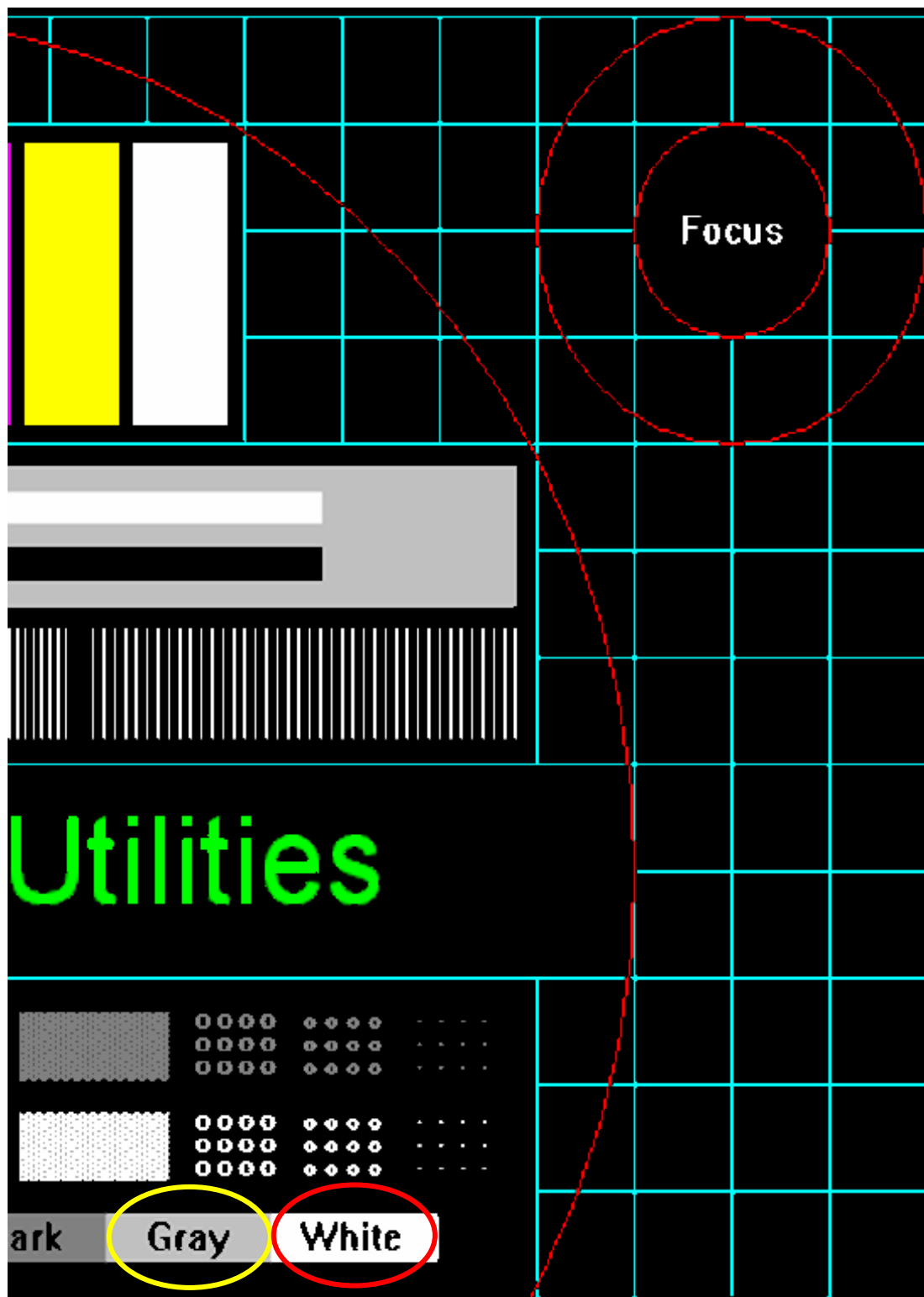
圖 5.12 由 800×600 放大到 1280×1024，(a) 經由 Bi-Cubic 放大影像，(b) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250），(c) 以 4×4 影像區塊作分析，所得之部分影像（內部臨界值=30，外部臨界值=120）。



(b)

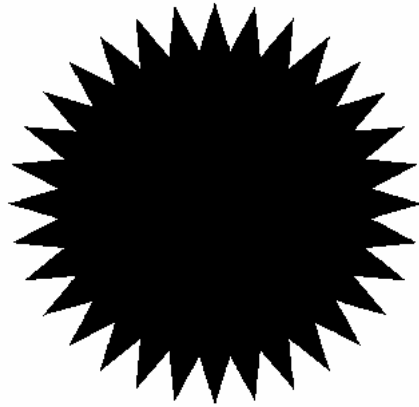
圖 5.12 (續)



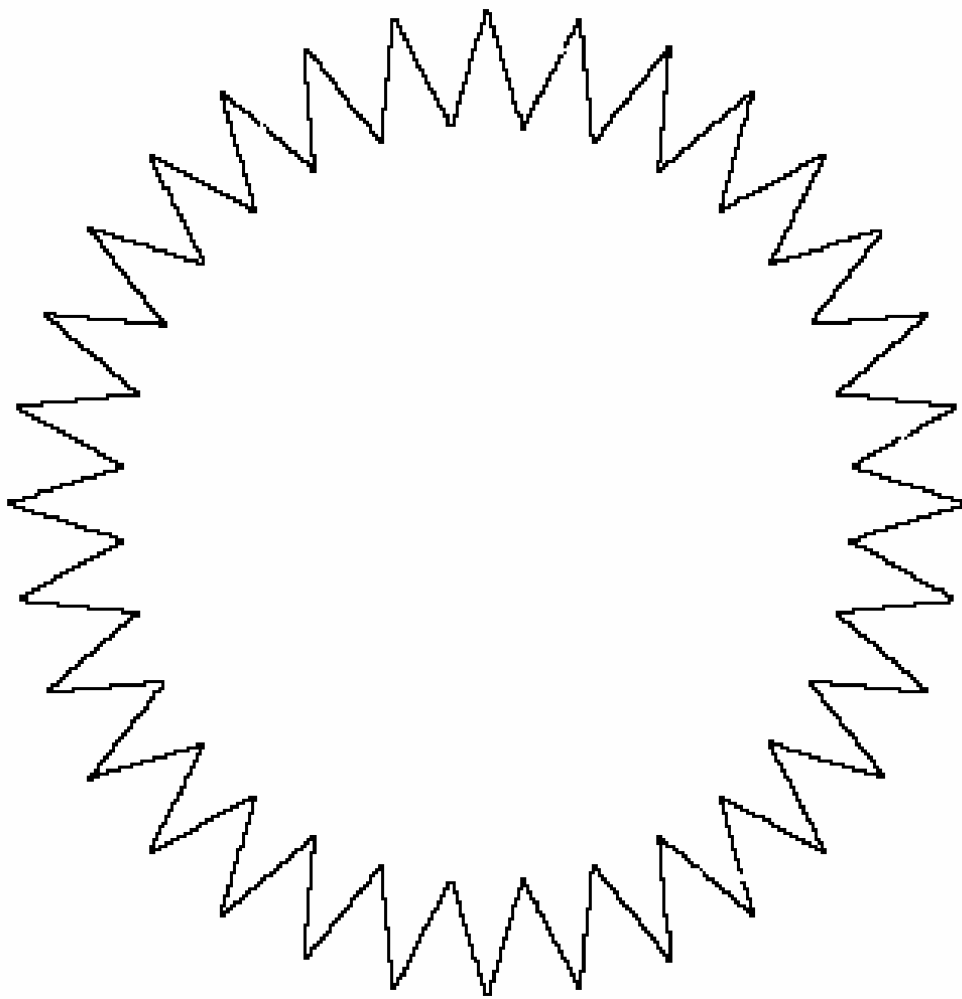


(c)

圖 5.12 (續)

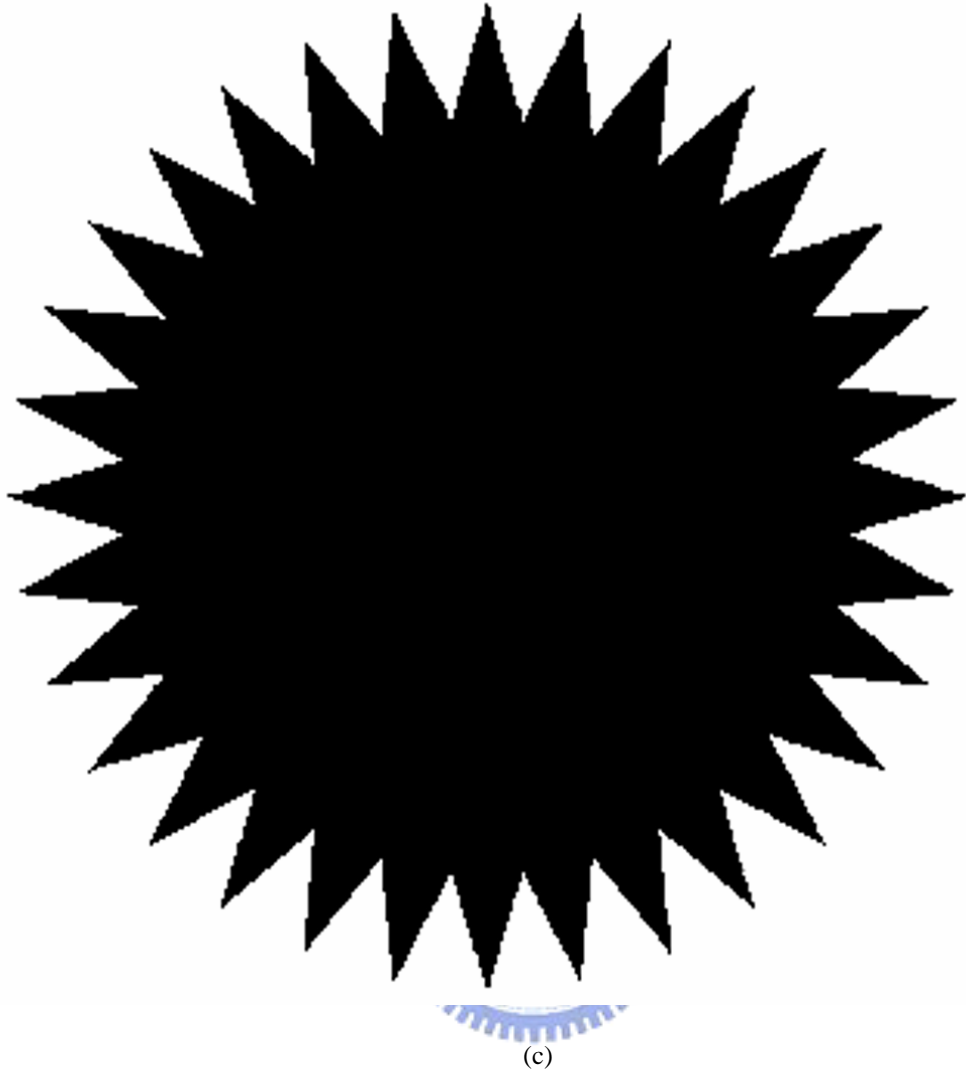


(a)



(b)

圖 5.13 由  $417 \times 313$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 以  $4 \times 4$  影像區塊作分析，所得之影像做外插計算的部分，(c) 經由 Bi-Cubic 放大影像，(d) 以  $4 \times 4$  影像區塊作分析，所得之部分影像（內部臨界值=10，外部臨界值=250）。



(c)

圖 5.13 (續)

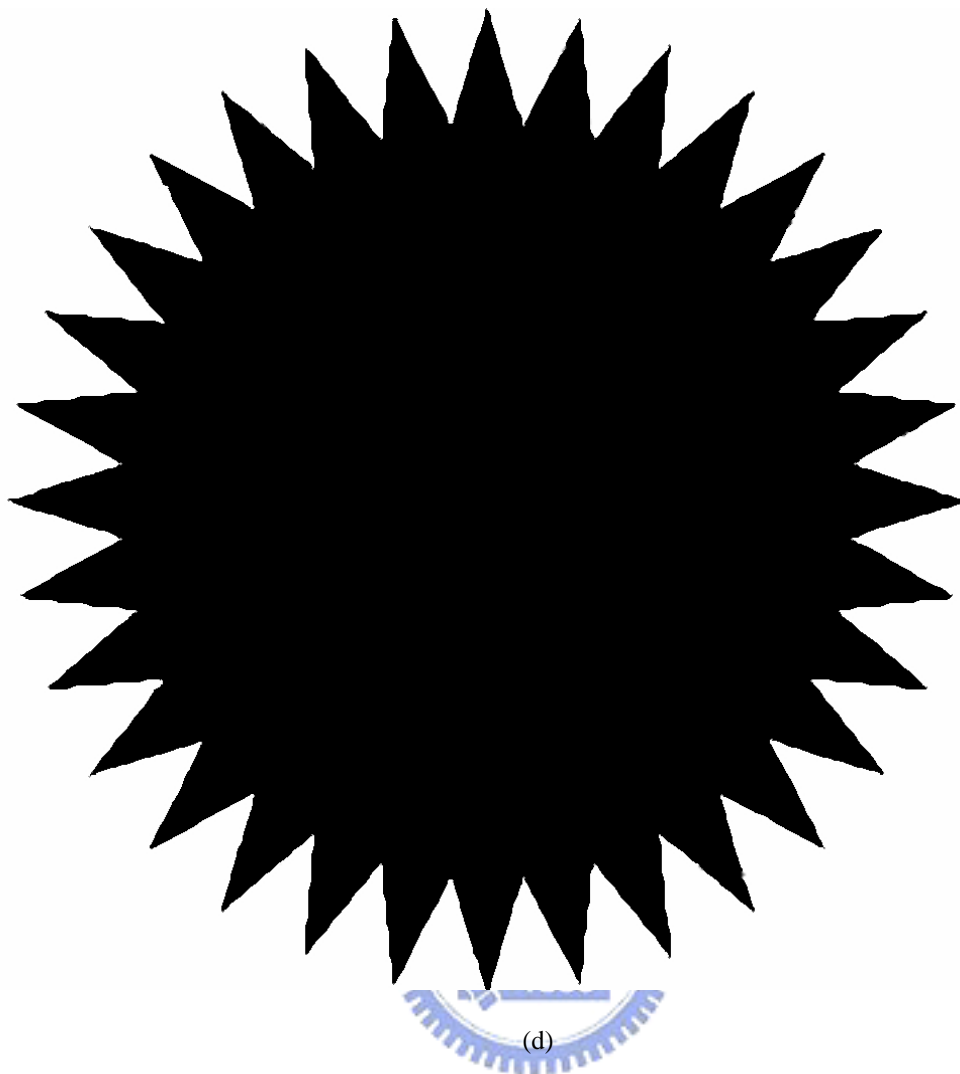


圖 5.13 (續)

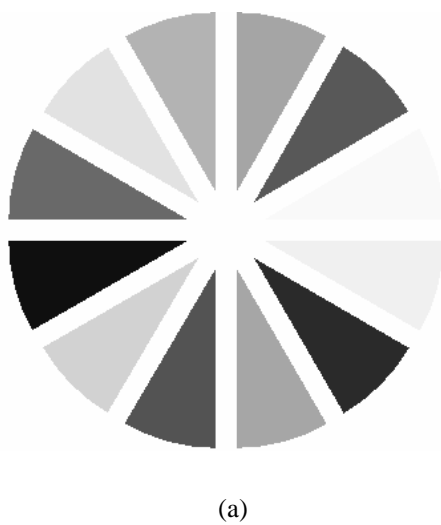
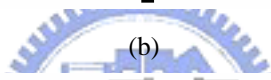
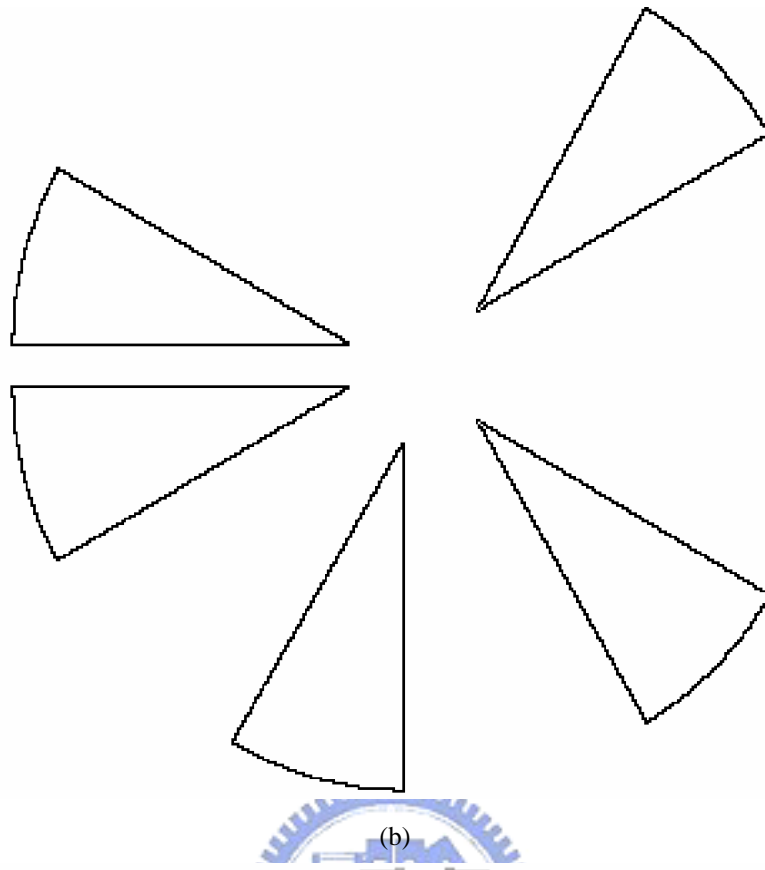
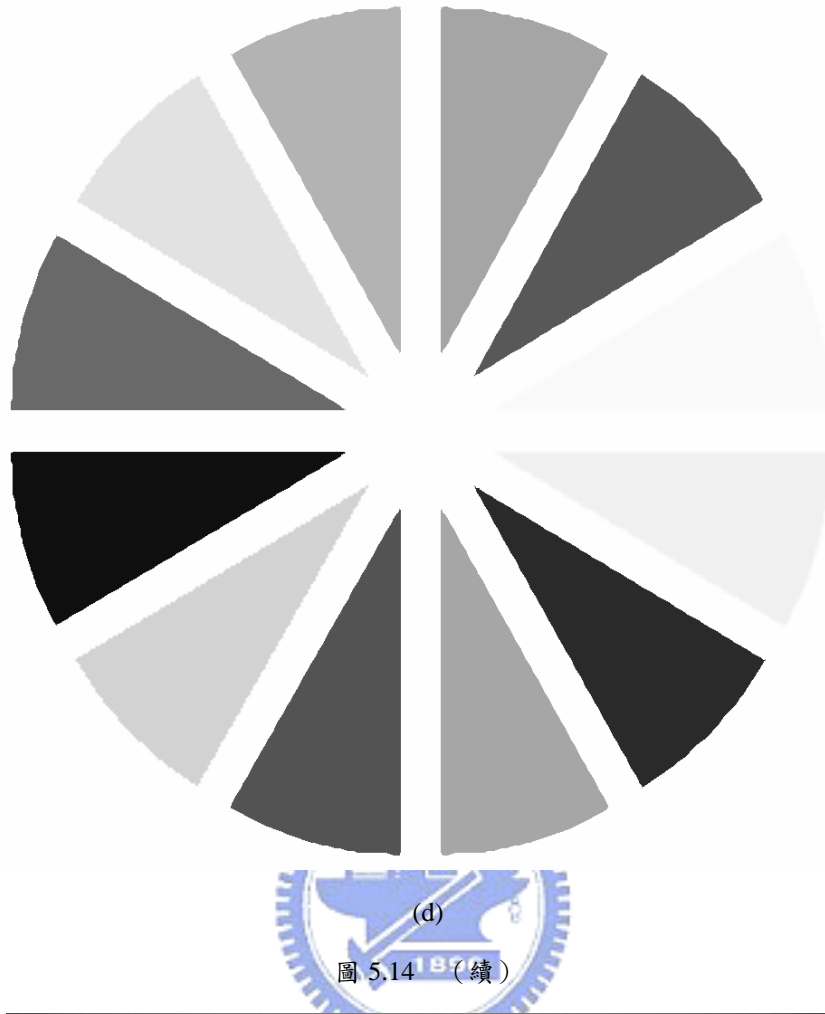


圖 5.14 由  $463 \times 358$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 以  $4 \times 4$  影像區塊作分析，所得之影像做外插計算的部分，(c) 經由 Bi-cubic 放大影像，(d) 以  $4 \times 4$  影像區塊作分析，所得之部分影像 (內部臨界值=30，外部臨界值=120)。



(c)

圖 5.14 (續)



(a)

圖 5.15 由  $320 \times 214$  放大到  $1280 \times 1024$ ，(a) 原始影像，(b) 經由 Bi-cubic 放大影像，(c) 以  $4 \times 4$  影像區塊作分析，所得之部分影像（內部臨界值=20，外部臨界值=100），(d) 以  $4 \times 4$  影像區塊作分析，所得之部分影像（內部臨界值=30，外部臨界值=40）。



(b)

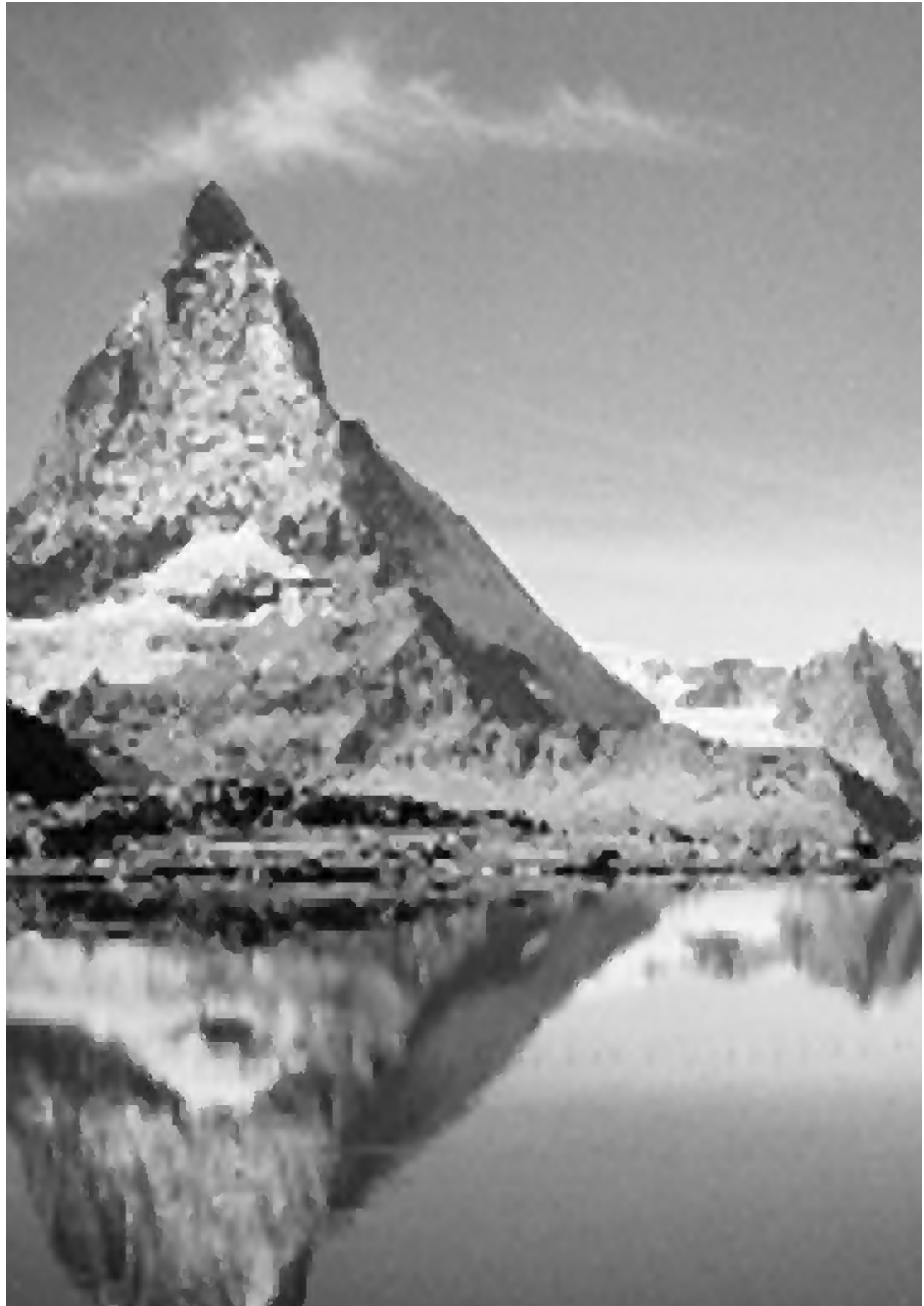
圖 5.15 (續)



(c)

圖 5.15 (續)





(d)

圖 5.15 (續)

## 第六章 結論與未來的研究方向

本論文提出了影像同時放大以及強化的方法，除了提出新的內外插補點方法之外，我們也在論文中探討這個放大強化影像方法中，分析原始影像時使用不同大小之局部區塊造成的影響。我們發現若是局部區塊內包含的參考點數量越多，可以預先定義的邊與角的種類越多，則放大之後的影像，在邊以及角的連續性上會有比較好的效果，但是分析原始影像的小區塊越大，則會讓運算複雜度增加，因此在兩者間需要取得一個平衡。而在本篇論文中，為了配合 Bi-Cubic 內插補點的方法，所以我們最大的小區塊只有探討到  $4 \times 4$  的大小。另外，在這個方法中，由於運算複雜度非常低，也不需要利用到額外的 Training Set，而且只要使用有限的記憶體，因此這個方法應該適合實作在硬體上。

目前這個方法在分析原始影像局部區塊為  $4 \times 2$  以及  $4 \times 4$  的大小時，仍有許多狀況我們沒有預先定義出來的，例如：在小區塊大小為  $4 \times 2$  時，我們並沒有考慮到角以及線段的狀況，而在小區塊大小為  $4 \times 4$  時，我們也沒有考慮到線段的狀況，所以在接下來的工作可以嘗試繼續把尚未預先定義的狀況都考慮進來。另外也可以嘗試將小區塊的大小再增加，這樣一來就可以包含更多種類的邊緣、角或是線段，對於放大後影像的邊緣、角或是線段的連續性會有較好的效果。再者，也可以嘗試簡化目前我們所預先定義出來的邊緣或是角的種類，以減少運算量，使得硬體實作上可以再簡化。另外，目前尚未將本論文中的方法實現到硬體上，在未來可以嘗試將此方法實現到硬體上；還有，現在的演算法中，符合我們預先定義的邊緣或是角的部分，是全部採用外插計算來補點，未來可以考慮先作一部份的外插計算後，再作內插補點，可能會讓放大過後的影像看起來效果更加自然。

## 參考文獻

- [1] T.W. Parks and C.S. Burrus, *Digital Filter Design*, John Wiley & Sons, 1987. Pages. 209-213.
- [2] R.G. Keys, “Cubic Convolution Interpolation for Digital Image Processing,” *IEEE Trans. Acoustic, Speech and Signal Processing*, vol. 29, pp. 1153-1160, 1981.
- [3] K. Turkowski, “Filters for Common Resampling Tasks,” *Graphics Gems I*, Academic Press, pp. 147-165, 1991.
- [4] D.P. Mitchell and A.N. Netravali, “Reconstruction Filters in Computer Graphics,” *Proc. SIGGRAPH*, volume 22, pages 221--228, August 1988.
- [5] <http://www.cg.tuwien.ac.at/~theussl/DA/node11.html>
- [6] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice Hall, 2<sup>nd</sup> Edition, 2002.
- [7] S.D. Bayrakeri and R.M. Mersereau, “A New Method for Directional Image Interpolation,” *Proc. Int. Conf. Acoustics, Speech, Sig. Process.*, Vol. 4, pp. 2383-2386, 1995.
- [8] D. Su and P. Willis, “Image Interpolation by Pixel Level Data-Dependent Triangulation,” *Computer Graphics Forum*, 2004.
- [9] L. Rodrigues, D.L. Borges, and L.M. Galves, “A Locally Adaptive Edge-Preserving Algorithm for Image Interpolation,” *Proc. Computer Graphics and Image Processing*, pp. 300-305, Oct. 2002.
- [10] C.H. Kim, S.M. Seong, J.A. Lee, and L.S. Kim, “Winscale: An Image-Scaling Algorithm Using an Area Pixel Model,” *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 6, pp. 549-553, June 2003.
- [11] G. de Haan and M. Zhao “Making the Best of Legacy Video on Modern Display,” *Society for Information Display DIGEST*, 2006.
- [12] X. Wu and X. Zhang, “Image Interpolation Using Texture Orientation Map and Kernel Discriminant,” *International Conference on Image Processing*, 2005.
- [13] A.J. Storkey, “Dynamic Structure Super-Resolution,” *Proc. NIPS*, 2002.
- [14] M. Bertalmio, A.L. Bertozzi, and G. Sapiro, “Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting,” *Proc. CVPR*, 2001.
- [15] B.S. Morse and D. Schwartzwald, “Image Magnification Using Level-Set Reconstruction,” *Proc. CVPR*, 2001.
- [16] M. Elad and A. Feuer, “Restoration of a Single Superresolution Image from Several Blurred, Noisy, and Undersampled Measured Images,” *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1646–1658, December 1997.
- [17] M. Irani and S. Peleg, “Improving Resolution by Image Registration,” *CVGIP*, 1991.

- [18] S. Baker and T. Kanade, “Limits on Super-Resolution and How to Break Them,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167-1183, September 2002.
- [19] H. Chang, D.Y. Yeung, and Y. Xiong, “Super-Resolution through Neighbor Embedding,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. I: 275–282, 2004.
- [20] W.T. Freeman, T.R. Jones, and E.C. Pasztor, “Example-Based Super-Resolution,” *IEEE Computer Graphics and Applications*, 22(2):56–65, March/April 2002.
- [21] A. Hertzmann, C.E. Jacobs, N. Oliver, B. Curless, and D.H. Salesin, “Image Analogies,” *Proc. of ACM SIGGRAPH’01*, pp. 327–340, 2001.
- [22] J. Sun, N.N. Zheng, H. Tao, and H.Y. Shum, “Image Hallucination with Primal Sketch Priors,” *Proc. CVPR*, 2003.
- [23] A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, Prentice Hall, 2<sup>nd</sup> Edition, 1999.
- [24] A.J. Eglit, “Method and Apparatus for Upscaling an Image in Both Horizontal and Vertical Directions,” US005739867, Feb. 1997.

