

章節目錄

第一章 緒論.....	1
1.1 研究動機.....	1
1.2 問題陳述.....	3
1.3 系統流程.....	6
1.4 相關研究.....	8
1.4.1 車牌定位、車牌字串分割方法及相關研究.....	8
1.4.2 車牌定位、車牌字串分割方法優缺點.....	13
1.5 論文大綱.....	15
第二章 車牌定位.....	17
2.1 基本原理.....	17
2.1.1 遮罩運算.....	17
2.1.2 車牌具有密集性高反差點特徵.....	21
2.1.3 以水平微分遮罩來偵測理想車牌位置.....	21
2.1.4 使用垂直微分遮罩對車牌定位並無助益.....	22
2.1.5 使用 $[-1 \ +1]$ 偵測實際車牌的困難.....	22
2.1.6 使用長遮罩來偵測車牌的優點.....	24
2.1.7 使用 $[-1 \ 0 \ 0 \ +1]$ 遮罩來偵測實際車牌.....	24
2.2 有效列的認定.....	26
2.2.1 群組化.....	26
2.2.2 固定微分臨界值的有效列認定.....	26
2.2.3 使用固定微分臨界值進行有效列認定的困難.....	28
2.2.4 動態微分臨界值的有效列認定.....	28
2.3 初步掃描.....	30
2.4 消除交錯現象.....	32

2.4.1 交錯現象形成的原因.....	32
2.4.2 消除交錯現象.....	32
2.5 細步掃描.....	34
2.6 微調整邊界.....	35
2.6.1 微調整左右界.....	35
2.6.2 微調整上下界.....	36
2.6.3 再微調整左界.....	37
2.7 車牌的確認.....	39
第三章 車牌字串分割 及 字元正規化.....	40
3.1 前處理.....	40
3.1.1 二值化.....	40
3.1.2 以交集運算消除孤立點及字元間連接單點.....	41
3.2 垂直投影量分割法及外圍輪廓分割法.....	43
3.2.1 垂直投影量分割法.....	43
3.2.2 使用垂直投影量分割法可能遇到的問題.....	44
3.2.3 外圍輪廓分割法.....	45
3.3 分割點的尋找.....	49
3.3.1 相對最小動態臨界值來尋找分割點.....	49
3.3.2 頭尾雜訊的去除.....	51
3.3.3 字元寬度的重新估計.....	51
3.3.4 破折線(dash)的尋找.....	51
3.3.5 雜訊的再去除.....	52
3.3.6 分割點確立.....	52
3.4 字元正規化.....	54
第四章 傾斜車牌矯正.....	58
4.1 傾斜角度計算.....	58

4.1.1 有關計算傾斜車牌影像的傾斜角相關研究.....	59
4.1.2 計算傾斜車牌影像的傾斜角基本原理.....	59
4.1.3 傾斜車牌影像的傾斜角計算.....	60
4.2 矯正傾斜車牌.....	62
第五章 研究結果.....	65
5.1 樣本來源.....	65
5.1.1 國內外研究車牌辨識其樣本取得來源.....	65
5.1.2 本車牌辨識系統硬體架構.....	65
5.1.3 本車牌辨識樣本取得來源.....	67
5.2 處理流程.....	69
5.2.1 輸入影像.....	69
5.2.2 初步掃描.....	69
5.2.3 除交錯現象.....	71
5.2.4 細步掃描.....	71
5.2.5 微調整邊界程式.....	73
5.2.6 量測傾斜角度.....	74
5.2.7 傾斜車牌矯正.....	75
5.2.8 外圍輪廓值分佈圖.....	75
5.2.9 分割點的尋找.....	75
5.2.10 字元正規化.....	76
5.3 實驗結果.....	77
5.4 實驗結果討論.....	80
第六章 結論.....	83
6.1 本論文和以往研究所不同處.....	83
6.1.1 測試樣本取得.....	83
6.1.2 本論文處理方法及效率和其他車牌辨識研究相比較.....	84
6.2 未來工作.....	85
參考文獻	

附錄

圖片目錄

圖 1.1	智慧型車牌辨識流程圖.....	7
圖 2.1	遮罩運算輸入輸出圖表示.....	18
圖 2.2	含有車牌的影像.....	20
圖 2.3	經 $\begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}$ 遮罩處理後影像.....	20
圖 2.4	經 $\begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}$ 遮罩處理後影像.....	20
圖 2.5	經 $\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$ 遮罩處理後影像.....	20
圖 2.6	經 $\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$ 遮罩處理後影像 ...	20
圖 2.7	經 $\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$ 遮罩處理後影像.....	20
圖 2.8	理想的車牌及其某水平列的灰階值.....	21
圖 2.9	車牌區域內某一列的灰階值.....	22
圖 2.10	經 $[-1 \ +1]$ 遮罩運算後的結果.....	22
圖 2.11	具有 5 個邊緣點的影像.....	23
圖 2.12	經 $[-1 \ +1]$ 遮罩運算後的結果.....	23
圖 2.13	短遮罩所能偵測到的反差變化.....	24
圖 2.14	較長遮罩所能偵測到的反差變化.....	24
圖 2.15	經 $[-1 \ 0 \ +1]$ 遮罩運算後的結果.....	24
圖 2.16	經 $[-1 \ 0 \ 0 \ +1]$ 遮罩運算後的結果.....	25
圖 2.17	經初步掃描後所得到的區塊.....	31
圖 2.18	交錯現象.....	32

圖 2.19 利用內插來消除交錯現象.....	33
圖 2.20 車牌的車牌字元區域.....	35
圖 2.21 細步掃描後的車牌影像.....	37
圖 2.22 未補償前的車牌影像.....	37
圖 2.23 車牌影像補償左界所計算範圍(實線方框).....	38
圖 2.24 補償後結果.....	38
圖 3.1 分五個互相重疊區塊做二值化.....	41
圖 3.2 未經過垂直交集運算.....	42
圖 3.3 經過垂直交集運算後的結果.....	42
圖 3.4 車牌字串區域.....	43
圖 3.5 垂直投影量分佈圖.....	43
圖 3.6 車牌字串區域.....	44
圖 3.7 垂直投影量分佈圖.....	45
圖 3.8 車牌影像.....	46
圖 3.9 上輪廓值分佈圖.....	46
圖 3.10 下輪廓值分佈圖.....	46
圖 3.11 外圍輪廓值分佈圖.....	47
圖 3.12 最終輪廓值分佈圖.....	48
圖 3.13 最終輪廓值分佈圖及相對最小動態臨界值.....	50
圖 3.14 第三個字元和破折號相連在一起.....	54
圖 3.15 消除破折號的車牌字元.....	54
圖 3.16 最近整數點法示意圖.....	56
圖 3.17 雙線性內插法取樣示意圖.....	56
圖 3.18 雙線性內插法座標示意圖.....	56
圖 4.1 傾斜的字串無法得到很好的分割點.....	58
圖 4.2 傾斜的車牌.....	59

圖 4.3 傾斜車牌的上輪廓值分佈圖.....	60
圖 4.4 以未進行傾斜矯正座標為計算基準的座標轉換.....	62
圖 4.5 以傾斜矯正座標為計算基準的座標轉換.....	63
圖 4.6 傾斜車牌矯正後的結果.....	63
圖 4.7 車牌定位、車牌字串分割及字元正規化流程圖.....	64
圖 5.1 車牌辨識系統架構圖.....	66
圖 5.2 輸入影像.....	69
圖 5.3 初步掃描後的結果.....	70
圖 5.4 預備細步掃描區域.....	71
圖 5.5 消除交錯現象後的預備細步掃描區域.....	71
圖 5.6 細步掃描後的結果.....	72
圖 5.7 細步掃描後決定的車牌區域.....	73
圖 5.8 微調整上下界後的結果.....	74
圖 5.9 用來量測傾斜角度的上輪廓值分佈圖.....	74
圖 5.10 傾斜車牌矯正後結果.....	75
圖 5.11 外圍輪廓值分佈圖.....	75
圖 5.12 6 個 30x20 的字元區塊.....	76

表格目錄

表 2.1	各類常用的遮罩運算.....	19
表 5.1	Meteor 的影像輸出規格.....	66
表 5.2	初步掃描後有效列列號及最大群組開頭、結尾行號.....	70
表 5.3	所有區塊的邊界.....	70
表 5.4	細步掃描後有效列列號及最大群組開頭、結尾行號.....	72
表 5.5	計算可能含有車牌的區塊.....	73
表 5.6	車牌左、右界的判定.....	73
表 5.7	計算出的字元群組.....	75
表 5.8	以車牌高為區分的成功率.....	78
表 5.9	以車種為區分的成功率.....	78
表 5.10	以加框與否為區分的成功率.....	78
表 5.11	以傾斜與否為區分的成功率.....	78
表 5.12	以拍攝地點為區分的成功率.....	79
表 5.13	找到破折號佔處理成功樣本比率.....	79
表 5.14	第幾次細步掃描找到車牌佔處理成功樣本比率.....	79
表 5.15	處理失敗樣本的原因.....	82

第一章 緒論

1.1 研究動機

近年來，由於國民所得的提高，購買及擁有汽車做為上下班、接送、休閒交通工具的風氣大增，使得臺灣近年來家庭或者公司擁用汽車的數量不斷增加，但受限於臺灣地稠人密，就衍生出停車位不足的問題。基於土地開發效益考量，住宅區多有新型公寓大廈建設地下停車場以解決停車位問題；同樣在都市商業活動密集地區，也採用建設地下公共的停車場，或者集中建設公眾停車樓塔為主以解決停車位問題。但還是常常發生停車位供不應求的問題，所以如何有效管理停車場，提高停車位利用率，降低管理成本，且達到安全管理目的，相信是停車場管理者關心之重要課題。

現行的收費停車場，採用的管理方式多為僱用專人在停車場的出入口，控制車輛的進出、停車的收費及預防車輛失竊。這樣管理停車場的方式對於管理人員來說，是相當枯燥而費時的工作，且車輛並非時時刻刻在進出，卻需要人員時時刻刻監督管理，相當浪費人力、時間及金錢。甚至在規模較小的停車場基於成本的考慮，則無任何管制的措施，任由車輛自由進出。另在一些本身就需要較開放性空間環境，如大學校園，常常各類活動頻繁，進出複雜，限於人力有限難以對所有進入車輛進行監控。如此則易產生安全上的疑慮及治安上的死角。

基於以上動機，設計此一自動車牌辨識系統，以電子自動化及管制機制來輔助管理需要控制車輛進出、收費的地方。而要達到：

- 自動化操作，以降低管理成本。
- 縮短流程，避免車輛在出入口造成回堵。
- 空位監控，使停車位的利用率達到最高。
- 安全管理，對進入停留過久車輛發出警訊，再追蹤管理。
- 防止人為弊端。

同時為了推廣大眾化，整個系統是以低成本為設計考量的重點，故在整個辨識運算核心上就不宜採用太過複雜的演算。如採用太複雜的演算法，整個處理流程的時間就會加長，另一方面也就要採用較昂貴的運算處理器以加強加計算能力，所以整個設計目標是以低運算量的車牌辨識計算核心為主，希望能達成低成本、高效率的要求。

1.2 問題陳述

自動車牌辨識系統之工作目標，是以攝影機取像，經影像擷取卡擷取後，再由數位影像處理方法，將影像中的車牌字元辨識出來。就本文所設計的系統，對象是現行自用轎車及計程車，主要面對之問題，是先從影像中(由動態影像擷取後)進行車牌位置之定位，再根據找到的車牌位置進行影像分割，把車牌字串分割成獨立且固定大小的車牌字元，再依序將每個獨立字元辨識出來。

本文中主要是討論如何從影像中找到車牌位置，再由找到的車牌位置中所得車牌字串分割成一個個獨立且固定大小的車牌字元，整個流程即包含車牌定位及車牌字串分割，至於車牌字元辨識部分主要是採用智慧型類神經圖樣比對法，詳細內容在溫福助所著「類神經網路圖樣比對法於車牌字元辨識之研究」[21]中有專門討論研究，本文則不再記述。

車牌定位方面，首先面對的問題便是車牌位置的尋找，由於車牌本身性質、取像設備、取像環境等方面條件不同，往往對車牌位置的尋找造成許多影響。以下就個別因素逐一討論：

(一)車牌本身性質

由於主管機關管理設計，自用轎車及計程車的車牌設計略有不同，自用轎車是前二個英文字母混合數字的字元，後四個則是純數字的字元，共計六個字元，計程車則是前二個英文字母混合數字的字元，後三個則是純數字的字元，共計五個字元。雖然車牌在汽車使用者剛領取時是制式的規格，可是隨著駕駛人的使用，往往會有逐漸污損、車牌字元模糊不清、車牌擺置歪斜的狀況，或者駕駛人常有自行在車牌加框架的情況，增加車牌定位的難度。

(二) 取像設備

就攝取車輛車牌的動態影像 CCD 攝影機而言，其本身規格依使用目的及價格不同，解析度、對比、信號雜訊比(Signal to Noise Ratio)等有所差異，如此便會影響到影像的品質。

(1)解析度不夠 CCD 攝影機，車牌影像易方塊化。如處理影像大小為

640x480 個像素，即約 30 萬個像素，若 CCD 攝影機的攝像晶片未達到 30 萬個像素以上，未能達到處理標準，則輸出的影像就易方塊化。

(2)信號雜訊比不夠高的 CCD 攝影機，車牌影像中易充滿干擾性高頻雜訊，所以 S/N Ratio 至少要大於 45dB，才能有效抑制雜訊。

(3)對比不高的 CCD 攝影機，車牌影像相較於周圍影像就模糊不清。

(4)性能、屏蔽效果較差的 CCD 攝影機則易因地磁或周圍磁場的影響使整張影像產生水波紋路。

(三) 取像環境

攝影環境則易受亮度、光度、取景視窗大小等因素影響著。

(1)攝影環境亮度不夠，得到車牌影像昏暗不明。

(2)攝影環境光度不均或車輛本身的材質，則易造成反光。

(3)攝影取景視窗太大，則易攝入非車牌物件，且本身車牌顯像過小。

(4)攝影取景視窗太小，則得到的影像範圍有限，車輛經過時，未必能把車牌照到。

在以上所述的各種因素影響下，更是額外增加了尋找車牌的困難度。

找到車牌的位置後，由於本國的車牌設計規格，車牌上除了包含車牌字元的區域，尚包含非車牌字元的區域，如車牌螺絲孔及標示車牌註冊機關的文字，所以還要去除非車牌字元的區域，僅留下車牌字元的區域。不過有時受限於攝影環境的影響，車輛所行經的路面並非水平，使得所攝得車牌影像有傾斜的情形發生，所以在車牌字串分割之前，則先必須做傾斜角計算，以決定是否進行車牌傾斜矯正。

再將車牌字串分割成一個個獨立的車牌字元，在這裡常見的問題，便是連字的問題。再將分割後單獨字元的做正規化(Normalization)，避免因為取像遠近的不同所造成車牌字元大小不一，不利後段類神經辨識使用，需將其正規成固定大小。而正規化後的字元由後半部的類神經網路來進行辨識，即完成整個車牌辨識的步驟。

1.3 系統流程

本系統的處理流程可分為幾個部分，流程圖如圖 1.1，包含以下幾個步驟：

- 車牌定位：尋找車牌位置，定位原理是採用類空間頻域分析，其包含了粗步掃描、細步掃描、微調整車牌邊界。
- 量測傾斜角度：由字串外圍輪廓分佈圖中字元高點來計算車牌傾斜角度。
- 傾斜車牌矯正：由已知的傾斜角度，進行座標轉換，將其矯正。
- 車牌字串分割：將找到的車牌字串分割成獨立的字元，主要是根據外圍輪廓分佈圖來進行分割。
- 車牌字元正規化：將大小不一的車牌字元正規化為 30x20 個像素大小字元區域。
- 車牌字元辨識：以智慧型類神經圖樣比對法進行字元辨識。

虛線所圍部分即本文所討論部分。同時為了日後系統變更方便，本系統對於每個步驟均採用模組化的設計。所以一張影像的輸入，經過本系統的處理，依車種的不同，可以得 5 個或 6 個大小為 30x20 的車塊字元區塊。再由溫福助所著「車牌字元辨識」的智慧型類神經圖樣比對法來進行車牌字元辨識。



影像擷取

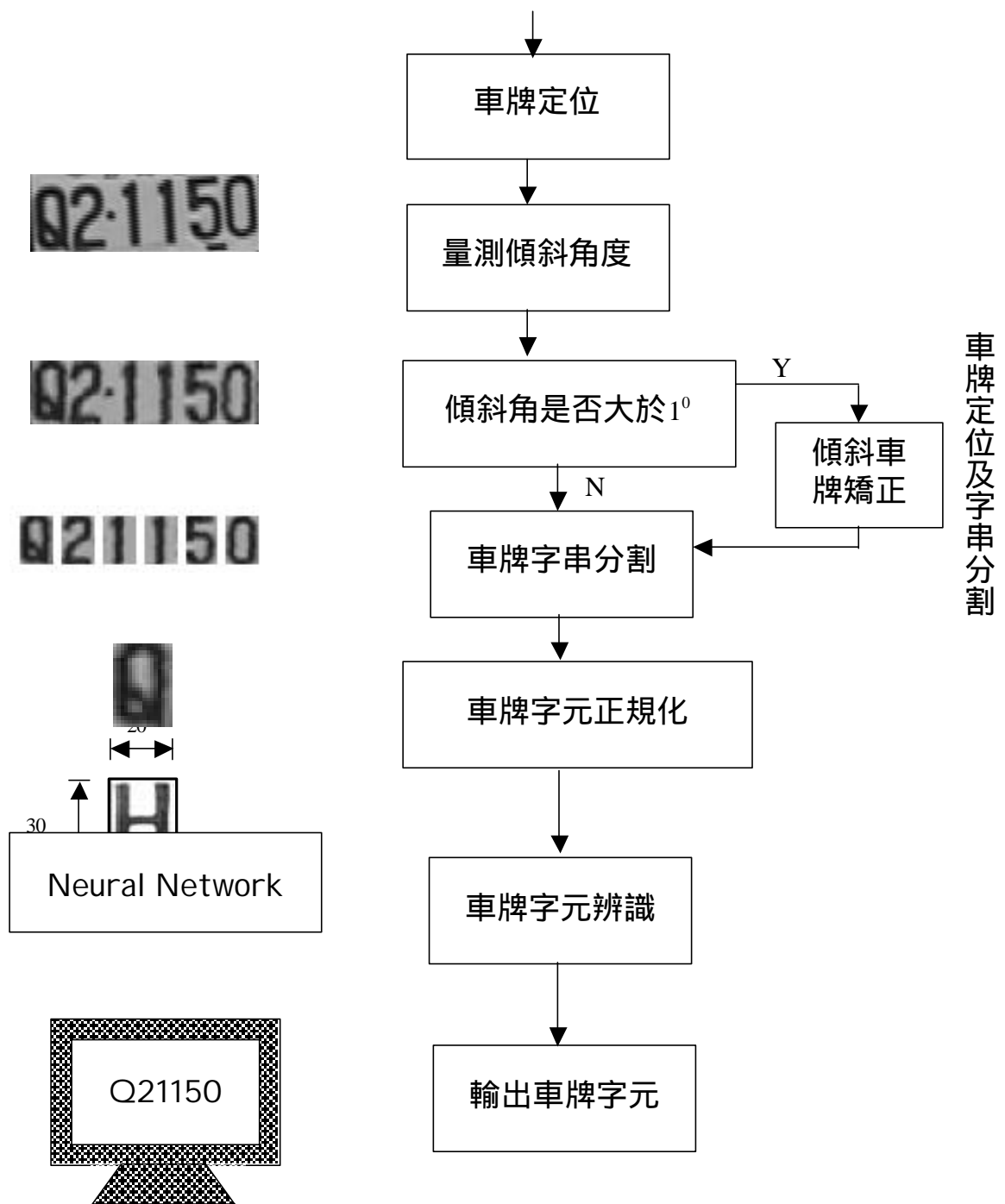


圖 1.1 智慧型車牌辨識流程圖

1.4 相關研究

國內外車牌定位、車牌字串分割研究，大致上可分為三大類的方法討論。

1.4.1 車牌定位、車牌字串分割方法及相關研究

(一) 連接物件法：

連接物件法是將車牌字元視為在影像中的物件，再利用車牌字元本身物件的特性來找到車牌物件。但個別研究方法及過程略有不同，如下記述：

林欣平所著的“車牌字元粹取”[14]中，提到現在一般使用交錯式攝影機擷取得到的畫面(frame)，是奇偶數列依序掃描，其間的時間差對運動稍快的物體，就可能產生影像互相交錯的問題。解決方法是只採用奇數列的資訊，偶數列則以內插的方式來計算，就可消除交錯現象的畫面。再利用事先照好的背景畫面和含有車輛畫面相互比較，受到攝影機本身性質及雜訊限制，不太可能做點對點的直接相減比較，所以採用數個像素點所形成區塊來進行前後比較，為了補償光線問題，採用影像處理中色階平衡的方法(Histogram Equalization)，再相互比較區塊的平均值如果差異夠大，則認為此一區塊可能含有車輛而加以保留。等全部區塊判定完時，再配合一些消去孤立區塊的方法，即可得到保留區域，爾後只要在此一區域計算即可，減少運算量。再將保留區域二值化，來進行區域內做物件的尋找。尋找物件的方法，主要是連通物件標記法，即影像處理中的8連通物件法(Connected Component)，就是把現行計算點視為中央點位置，再去判定八個方向點是否和中央點相連，如果有相連，則標示現行計算點和相連點為同一物件標記，如此依序在保留區域內做計算判斷，就可以得到許多的物件，同時記錄物件的屬性，包含物件的範圍(即左上角位置，範圍的寬和高)、寬高比、密度、面積。再從得到的物件屬性，去除掉一些不符合車牌字元物件特性的物件，如太寬、太高、太瘦、密度過大、密度過小、面積過大、面積過小。剩下的物件中依據車牌個別字元物件位置必然相互靠近的性質來做判定，如果有5、6個相連的物件，即可認定是車牌個別字元。

周俊男所著的“車輛牌照影像辨識系統”[15]，是先將得到的影像畫面以群組變異最小化法(Minimizing Within Group Variance)做二值化，類似灰階統計直方

圖(Grey Histogram)機率統計分佈。再逐列進行連接元件標示，即同列中的連續像素點為同一串列(list)，同時記錄串列的起始點及結束點。再進行垂直方向分析，如果是同一物件，則相鄰兩列的串列則標記為同一標號。處理完後，可以得到許多的物件，再進行物件的分析，利用物件的位置及寬、高、寬高比以累積加分的方式來進行連接元件串列分析。選出最高分的串列，即車牌位置所在。再以垂直投影法，以投影量為零的位置來分割字串。再以數學形態學的方式來進行字元辨識。

張銘豪所著的“利用分割辨識方法之英文數字辨識系統”[19]中，從二值化到累積加分的方式來進行連接元件串列分析，方法和上篇相同。後再利用個別車牌字元物件的相對中心位置並同時使用 Hough 轉換來分析其旋轉角度，來以座標轉換將其車牌轉正。字元分割則以垂直投影法來進行。先計算出垂直投影圖，再利用二階微分值較大者為波谷處，同時定義一目標函數來計算分割點。再以數學形態學的方式來進行字元辨識。

(二) 類空間頻域分析法：

類空間頻域分析法是利用車牌在影像中有急遽而密集的灰階變化來進行車牌字元粹取。同樣但個別研究方法及過程略有不同，如下記述：

楊文宗所著“車輛牌照號碼自動辨識系統”[20]，先對三種不同二值化方法做探討，根據灰階轉換的特性來選出車牌位置的候選區(車牌定位)，再進行車牌字串分割、字元辨識。先將得到的影像畫面做二值化，而分別探討以經驗固定臨界值、灰階矩量保持臨界值、形態法來做二值化。經驗固定臨界值法則取灰階中間值 128 來做二值化；灰階矩量保持臨界值法是利用輸入的灰階矩量(Gray moments)和二值化影像矩量不變來處理，類似統計機率分佈的方法；形態法是以影像處理中的邊緣偵測運算子(operator)先做運算，再檢查運算後值是否大於某一固定值來決定二值化後的影像。再從二值化的影像，利用車牌字元具有高度的灰階轉換特性來判定此列是否含有車牌，如果此列灰階轉換數小於某一數值或列

黑點總和大於某一數值，此列的灰階值全重設為白點(即無意義點)，再分別計算水平、垂直方向的連續黑點，如連續黑點量如大於某一數值，則同樣該區段的灰階值全重設為白點。後定義一個測試區塊(test block)去找尋殘留的黑點，來做為候選區的判定。再從候選區的色階統計直方圖找出最佳的二值化值進行第二次的二值化，利用車牌字元寬度大小一定，配合橫向、縱向連續點的判定把一些不屬於車牌字元的部分刪除，找出黑點的最大量分佈區域，即完成車牌定位。後以垂直投影法進行字串分割，將投影量為零處當做字元分割處。字元辨識則是以筆劃分析法進行。

義大利(Italy)的 E.D.Di Claudio, G. Lucarelli, G. Orlandi, and R. Paris 等人所著的“ Car Plate Recognition by Neural Networks and Image Processing ”[6]中，是以空間頻域分析的原理來找到車牌的位置，方法為離散傅利葉轉換(DFT,Discrete Fourier Transform)。先輸入影像先經過邊緣增強(Edge Enhancement)，且影像中的車牌區域最好變化不要超過 15%，如此才可以使用較快部分區耗時的尋找。基於車牌是影像中灰階變化最為激烈地方的假設，先做列的離散傅利葉轉換，再將列的離散傅利葉轉換後的數值取絕對值再相加以做為垂直諧波(harmonic)標準差估算能量。這裡的離散傅利葉轉換可以用現行的數位訊號處理器(DSP)來加快實現。得到垂直諧波的標準差估算能量，再從變化較明顯的突起處，可得到車牌的垂直位置，再從留下的區域去做行的離散傅利葉轉換，即可得到水平的諧波標準差估算能量，從變化較明顯的突起處，便可判斷出車牌的水平位置。抽取出車牌區域再做二值化，再以垂直投影來進行車牌字串分割，找二個車牌字元中的白色區域以做為分割點。車牌字元辨識則是以類神經來進行辨識。

韓國(Korea)的 B.T. Chun, Y.S. Soh, and H.S. Yoon 所著的” Design of Real Time Vehicle Identification System ”[5]，使用方法為一階水平差分空間頻域分析法。先將輸入影像經由線取樣(Line sampling)，為了減少運算的時間，建議最好每隔 3、4 列才取樣一列，再編碼(Coding)。使用 Sobel 運算子來強化影像，以其值的一階水平差分是否大於某一數值來當作確認是否有較為激烈灰階變化，整個突起

(peak)的大小需介於一定的長度之間，同時再忽略負的突起，及峰值不夠高的突起。再做群聚(Merging)，相互兩組突起之間如果長度小於可接受的車牌寬，則可群聚在一起。經過以上的步驟，就可以選出車牌的候選區。再由候選區中，進行上下兩方面的尋找，如果相當於車牌字元高的區域內其複雜度夠強，則認定是實際的車牌區域。

英國(UK)的 J. Barroso, J. Bulas-Cruz, E.L. Dagless, and A. Rafael 所著”Number Plate Reading Using Computer Vision “[1]中，使用方法為一階水平差分空間頻域分析及配合垂直投影來決定車牌的位置。根據事先已經統計好的車牌特徵資訊，如車牌區域內黑點的數量、黑點強度及黑點對黑點的相對距離，逐列在已定義好的一定長度區間用以來尋找一連串的連續極大值、極小值符合以上統計的車牌特徵資訊(事實上也就是在列上有激烈的連續灰階山峰、波谷)。找到大致的區域後，再做二值化，其二值化臨界值的選擇是基於區域內灰階最大值及最小值來選擇。再利用垂直及水平投影法找出車牌真正的實際位置。確認了車牌真正的位置，利用垂直投影法及配合 Lu.Y 所提出的峰值相對波谷(peak to valley)來尋找波谷的切割位置。再由所提出以尋找字元的轉折點來做為辨識字元的依據。

日本(Japan)的 I. Fujita, K. Miyamoto, K. Nagano, M. Tamagawa, and M. Yamamoto 所著的” Vehicle License-Plate Recognition by Image Analysis ”[8]中，尋找的方法是依據列中點的相互關係(correlation)。同上方法依據由車牌字元在影像中產生明顯的明暗交錯變化。先將輸入影像做個別局部的與點的特殊相互間的關係來偵測出由於車牌字元明暗變化產生的突波(Pluse)。在點與點相互關係分佈的平面上以兩個方向的樣本進行比較(template pattern matching)，找尋字元的圖樣或者是似字元的圖樣，以選出車牌的候選區，依據一些非車牌區域資訊來驗證車牌的候選區是否含有車牌，否則再找下一個區域。抽取出的車牌區域進行二值化及做標籤(labeling)的工作，再由建立好的標號樣本來確認標籤物是否為車牌字元。

德國(Germany)的 C. Busch, R. Domer, C. Freytag, and H. Ziegler 在” Feature

Based Recognition of Traffic Video Streams for Online Route Tracing”[2]中，主要是依邊緣點形成線的相互關係來找尋車牌的邊界。車牌的一些標準資訊需要事先知道，有些參數需跟隨著情況的不同而調變，如車牌的大小及光線的明亮度，則由感測器及攝影機的距離推算而得知。將輸入的影像經過水平 Sobel 運算子來加強邊緣點(白-黑-白)，而邊緣點形成線，確認這些線的長度是否介於一定的長度之間，由已知的車牌大小來找鄰近區域的線來確認車牌的上下界；以同樣的方法可以找到車牌的左右界。再經由找尋車牌中相連續的白色區域以做為車牌字元的上下左右界。

(三) 空間時域分析法：

空間時域分析法主要是利用車牌本身在影像中造成強烈特性來進行辨識。

荷蘭(Netherlands)的 H.A. Hegt, R.J. Haye, and N.A. Khan 在” A High Performance License Plate Recognition System”[10]中，先利用樣本比對來尋找車牌的四個明顯的角點，再以空間頻域的方法來確認四個角點所形成的區域，這空間頻域的產生即是來自車牌中字元相對背景的變化，同時由於取像角度的關係，車牌未必是完整的長方區塊，可能有傾斜彎曲，利用雙線性轉換(Bilinear Transformation) 將其矯正為完整的長方區塊。矯正後的車牌在利用統計出來的資訊，也就是車牌 60%以上的背景白點、10%以上的字元黑點，來做二值化，同時再利用垂直投影法，投影累積量值的高峰點來認定為分割點。字元的頂端及底部則是同樣利用水平投影法來找尋當列的黑點累積量大量增加時，即是字元的頂端及底部。

泰國(Thailand)的 K. Chamnongthai and T. Sirithinaphong 所著的”Extraction of Car License Plate Using Motor Vehicle Regulation and Character Pattern Recognition”[4]中，其主要尋找的依據為車牌的一些色彩及外觀特徵。由車牌制定的規格，車牌是含有二種顏色，一為車牌本身的背景顏色，另一為車牌的字元顏色，因此車牌的找尋便是基於色彩的資訊，即車牌背景點顏色及車牌的長方形

外觀來進行，再由車牌內的點連續的特性來確認車牌的位置。再利用垂直及水平投影法來進行字元的分割，分割好的字元區塊用利用類神經進行字元辨識，也驗證車牌的位置。

1.4.2 車牌定位、車牌字串分割方法優缺點

總結以上，國內的研究[14][15][19]較常使用連通物件法，連通物件法其優點就是不易受取像角度的影響，即使物件歪曲，但車牌的物件仍屬獨立的物件，仍可粹取出來，且確認是車牌字元物件，就不需要再經過車牌字串分割的步驟，[15]的研究方法即是如此。此外[15]、[19]則是同一系列先後的研究，其[19]相較[15]的研究不同，就是提出如何將傾斜的車牌做矯正，其矯正的依據則是根據尋找字元物件中心的分佈。但此連通物件法缺點是通常一定要對整張影像做尋找，且對整張影像二值化，而整張影像二值化未必即是對車牌區域做較佳的二值化，此外物件認定也較為費時。另外在[14]的研究中也是採用連通物件法，但尚需先照一張背景圖來比較，好處是可以減少尋找物件的時間，可是就必須增加額外的取像機制。

類空間頻率分析法基本的假設是車牌應是影像中變化最激烈的地方，利用此一特點來尋找車牌位置，優點是可以使用一些影像處理中的技巧，如遮罩等來輔助尋找車輛，缺點是一些參數往往無法明確的決定。在車牌字串分割常使用垂直投影法，遇到連字時常難以處理。[20]的研究中探討了不少二值化方法，可是問題同前所提，對整張影像做二值化，未必是對車牌做較佳的二值化，且對垂直投影量為零做為車牌字元的分割點，事實上大部分的车牌字元區域未必如此理想。[5]的研究報中對於一些如峰值大小認定為定為車牌則未有明確的交代。[1]對於尋找車牌位置的方法是找一連串的連續極大值、連續極小值，可是同樣未對尋找的演算法有明確的交代。[8]則需先建立

點和點的交互關係平面，且採用樣本比較來尋找車牌，對於樣本的建立則有一定的難度及認定。[2]中演算法其事先所需的資訊較多，同時也就是對於環境的限制較為嚴格，文中記述的找尋車牌中相連續的白色區域以做為車牌字元的上下左右界，對於我國的車牌未必如此理想。

空間時域分析法是利用一些車牌已知的特徵來選取，這種方法往往受限於各國車牌特性的不同而有所變動，算是彈性較差的方法，較少為使用，但是如果車牌具有較明顯的特性，其使用的效果應也不錯。[10]即是利用樣本比對來找尋車牌的明顯四個角落區域，來確認車牌的位置，可是對於我國的車牌則未必有如此良好的特性。[4]則是利用車牌的背影顏色及車牌的外觀來尋找車牌，同樣地，對於我國的車牌同樣未必有如此良好的特性。

1.5 論文大綱

本文章節依序為第二章的車牌定位，第三章的車牌字串分割及字元一般化，第四章的傾斜車牌矯正，第五章的研究結果，第六章的結論則做個總結。

第二章的車牌定位就是找出車牌的位置。2.1 節的基本原理即介紹使用車牌在

影像中具有較密集的高反差點特質來達成定位。2.2 節的有效列認定則談到如何認定掃描水平列中是否含有車牌區域。2.3 節的初步掃描則介紹如何嘗試找到車牌在影像中大概位置。2.4 節的消除交錯現象提到如何消除交錯式 CCD 攝影機因時間差引起奇偶列交錯開的現象。2.5 節的細步掃描則提到找到車牌大約的區域位置後，再從影像中某一範圍裡找出車牌的真正位置。2.6 節的微調整車牌邊界是去除車牌中非車牌字元區域，只保留車牌字串區域。2.7 節車牌確認則是根據車牌的基本特性來確認找到影像是否為真正車牌影像區域。

第三章的車牌字串分割及車牌字元正規化中是談到如何從第二章得到的車牌字元區域進行分割使其成一個個獨立的車牌字元，再把大小不一的車牌字元區塊正規化成固定大小的字元區塊。3.1 節前處理提到先做二值化，並配合交集運算消除高頻雜訊。3.2 節垂直投影量分割法及外圍輪廓分割法則是談到傳統垂直投影量分割法並不適用處理本系統取得車牌字串，另採外圍輪廓分割法來進行車牌字串分割。3.3 節分割點的尋找則到介紹如何根據外圍輪廓法的結果來得到分割點。3.4 節的字元正規化則是介紹如何將大小不一的車牌字元區塊正規化成固定大小的字元區塊。

第四章的傾斜車牌矯正是談到如何將傾斜的車牌區域矯正。4.1 節的傾斜角度計算則到談到如何利用一些資訊去量測車牌的傾斜角度。4.2 節的矯正傾斜車牌則是介紹如何根據我們所量測到車牌傾斜角去矯正還正車牌。

第五章主要是陳述一些實驗的結果及討論。5.1 節的樣本來源則是介紹本系統實驗樣本如何取得，同時比較和以往相關文章討論的樣本有何差異。5.2 節的處理過程則是舉一實例依序照第二章、第三章、第四章的步驟陳述每一個階段程式處理後的結果及數據。5.3 節則是程式實驗後的結果。5.4 節則是對程式實驗後的結果討論。最後的第六章的結論則是對以上的工作做總結。

第二章 車牌定位

車牌定位就是從含有車牌的影像中找到車牌的位置。本系統的定位原理是採用類空間頻域分析法，利用車牌應是影像中具有密集性灰階變化激烈此一特徵，來達成車牌定位。在整個車牌辨識的流程中，車牌定位往往是耗費計算資源較多的部分，雖本系統基本原理仍是採用類空間頻域分析法，但在低運算量、高效率的要求下，其實行演算方法和之前的研究則有不同。

2.1 節基本原理是介紹影像處理中常使用的遮罩運算，並介紹如何選取適合的遮罩運算來粹取高反差點(灰階激烈變化點)。2.2 節有效列認定介紹如何認定運算水平列是否含有部分車牌，其中一些參數是採用動態設定，以儘量消除光度的影響。2.3 節的初步掃描則是利用上節有效列認定來計算車牌的大概位置，再由這大概位置經 2.5 節細步掃描來找出車牌的真正位置，這兩段式掃描(初步掃描、細步掃描)正是本系統在車牌定位階段，強調可以達成低運算量訴求所在。還有其中 2.4 節消除交錯現象介紹如何消除取像設備因時間差造成交錯(Interlace)現象。2.6 節的微調整邊界即是去除經以上步驟處理後非車牌字串的部分，以避免造成後續車牌字串分割的困擾。2.7 節車牌確認則是判定目前經之前步驟計算出的影像區域是否真為車牌區域，若不符合車牌的基本假設，則依經驗定律再往上尋找區塊是否含有車牌。

2.1 基本原理

本節先介紹影像處理中常使用的遮罩運算，同時陳述車牌在影像中具有密集性高反差特徵。利用 $[-1 \ 0 \ 0 \ +1]$ 水平微分遮罩來粹取此一特徵，並解釋為何 $[-1 \ +1]$ 及垂直微分遮罩不適合用於偵測車牌密集性高反差此一特徵。

2.1.1 遮罩運算

電腦視覺(Computer Vision)主要研究目標即在使用影像處理的方法來獲影像相關資訊，其中最常使用是遮罩(mask)影像處理，在一定的範圍內將每一點的數值乘上事先設定好的權值(weighting)再加總，以強化特徵資訊或者進行特徵抽取。假設是 $f(x, y)$ 範圍內影像的數值(如灰階值)， w 是權值， $g(x, y)$ 是最後的輸出值，其輸入和輸出的關係如下示意圖 2.1。

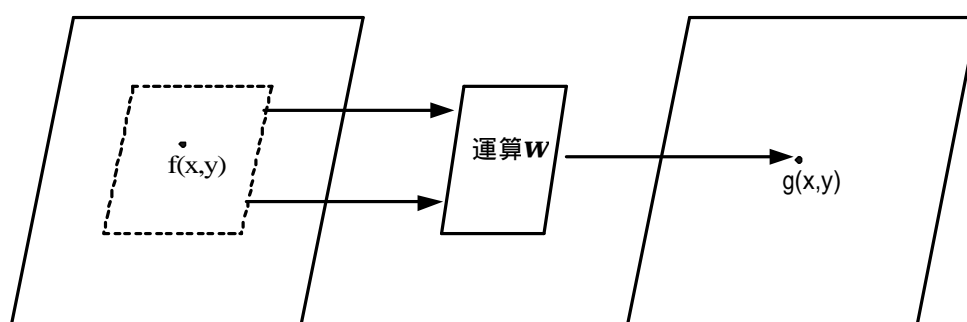


圖 2.1 遮罩運算輸入輸出圖表示

數學式表示為：

$$g(x, y) = f(x, y) * w(x, y) = \sum_{(k,l) \in w} w(k, l) f(x-k, y-l) \quad (2.1)$$

以常用的 3x3 遮罩運算為例：

$$\begin{aligned} g(x, y) = & w_1 \cdot f(x-1, y-1) + w_2 \cdot f(x, y-1) + w_3 \cdot f(x+1, y-1) \\ & + w_4 \cdot f(x-1, y) + w_5 \cdot f(x, y) + w_6 \cdot f(x+1, y) \\ & + w_7 \cdot f(x-1, y+1) + w_8 \cdot f(x, y+1) + w_9 \cdot f(x+1, y+1) \end{aligned}$$

例如影像中尋找含有由白變黑或者黑變白的特徵資訊，即確認在計算中的此點是否為灰階變化點(即邊緣點)時，常使用微分強化遮罩。微分強化遮罩除了是偵測邊緣點的工具，本身也是可做為影像邊緣強化工具。由於影像是二維離散信號，所以一次微分有二個分量，此一分量亦可稱為梯度，其定義的表示式如下：

$$G[f(x, y)] = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} \quad (2.2)$$

其中 $\partial f / \partial x$ 在 Δx 很小可以和 $[f(x + \Delta x, y) - f(x, y)] / \Delta x$ 相近似，在離散信號系統中，最小即是一個整數單位，所以取 $\Delta x = 1$ ， $\partial f / \partial x$ 則和 $[f(x + 1, y) - f(x, y)]$ 近

似；同理 $\partial f / \partial y$ 在 Δy 很小可以和 $[f(x, y + \Delta y) - f(x, y)] / \Delta y$ 相近，取 $\Delta y = 1$ ，則 $\partial f / \partial y$ 和 $[f(x, y + 1) - f(x, y)]$ 近似。如以 2x2 矩陣遮罩表示：

$$f(x+1, y) - f(x, y) = f(x, y) * \begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix} \quad (2.3)$$

$$f(x, y+1) - f(x, y) = f(x, y) * \begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix} \quad (2.4)$$

常用的一階微分運算則有如 Roberts、Prewitt、Sobel 遮罩[14]：

	w	
Roberts	$\begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$

表 2.1 各類常用的遮罩運算

Roberts 遮罩運算是基本的微分遮罩，依據使用目的不同，有水平(即 $\begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}$)、垂直(即 $\begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}$)及斜角方向(即 $\begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$ 、 $\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$)，圖 2.2 是含有車牌的影像，分別經過 Roberts 的水平、垂直、斜角方向遮罩處理後結果如圖 2.3、圖 2.4、圖 2.5，分別強化出垂直、水平、斜角方向的邊緣點。Prewitt 和 Sobel 遮罩運算則由微分運算和低通運算結合，具有降低雜訊的效果，同樣地有水平及垂直方向的遮罩運算，圖 2.6、圖 2.7 是分別以 Prewitt 和 Sobel 水平遮罩運算後結果，由於考慮加權範圍較 Robert 大，計算出數值較高，其垂直方向邊緣點均較 Roberts 強化更明顯，但 Sobel 遮罩又比 Prewitt 遮罩更加加強加權矩陣中央部分，所以在圖 2.7 垂直方向邊緣點又比圖 2.6 更加明顯。



圖 2.2 含有車牌的影像



圖 2.3 經 $\begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}$ 遮罩處理後影像

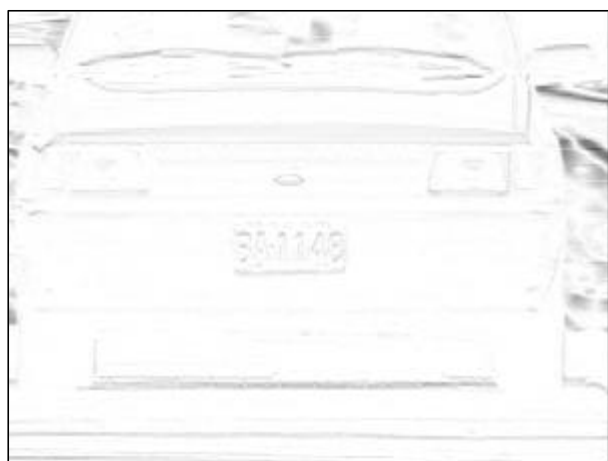


圖 2.4 經 $\begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}$ 遮罩處理後影像

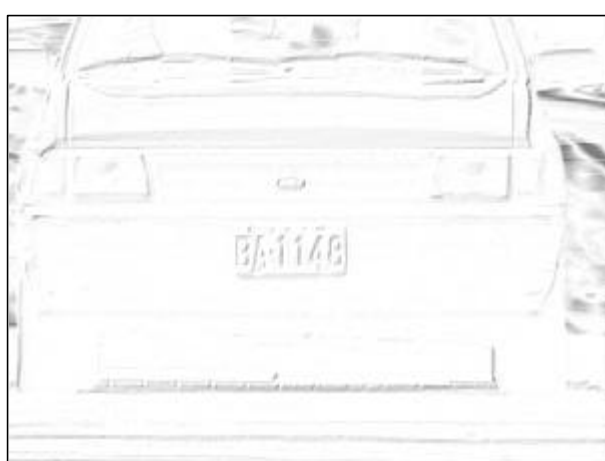


圖 2.5 經 $\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$ 遮罩處理後影像



圖 2.6 經 $\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$ 遮罩處理後影像



圖 2.7 經 $\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$ 遮罩處理後影像

2.1.2 車牌具有密集性高反差點特徵

將含有車牌的影像圖 2.2 以 2×2 水平微分遮罩(即 $\begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}$)處理後，如圖 2.3，可以看到物體的邊緣線都被強化，這邊緣線上的邊緣點即代表此點附近灰階色階變化具有白急遽變黑或者黑急遽變白的特徵，為灰階色階差高度的變化(因為白是灰階值中最小為零，黑是灰階值中最大為 255)。由於車牌處有一連串緊靠在一起的車牌字元及車牌本身的邊緣，所以在車牌此一區域灰階變化點(邊緣點)分佈較為密集。故車牌定位的基本原理就是利用處理後的影像邊緣點大量密集出現在車牌區域，具有較高度密集的高反差變化來找尋車牌位置。

2.1.3 以水平微分遮罩來偵測理想車牌位置

一個理想的車牌，車牌字元顏色是黑色(灰階值 255)，車牌本身背景顏色是白色(灰階值是 0)，且車牌本身不傾斜歪曲，如下圖 2.9。將車牌區域內虛線所經過一水平列做微分運算，使用 $\begin{bmatrix} -1 & 1 \end{bmatrix}$ 遮罩運算(即以上所討論的 $\begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}$ ，因為第二列的元素皆為零，可不予計算)。其計算結果如圖 2.10。可以發現在點附近由黑變白時，可以得到 9 點較高正微分值；同樣，在點附近由白變黑時，可以得到 9 點較高負微分值。如果將黑變白及白變黑的邊緣點變化視為同一類，也就是不論其微分值的正負，則可以發現共有 18 個。所以使用 $\begin{bmatrix} -1 & +1 \end{bmatrix}$ 水平微分遮罩來偵測邊緣點，考慮到車牌字元出現最差的情況(I1-1111)，則每一列至少仍有 12 個以上的邊緣點。此一情況相較其他物件在影像以遮罩處理完其邊緣點分佈，為其獨特，故可做逐列檢查水平列是否含有車牌的基本尋找條件。

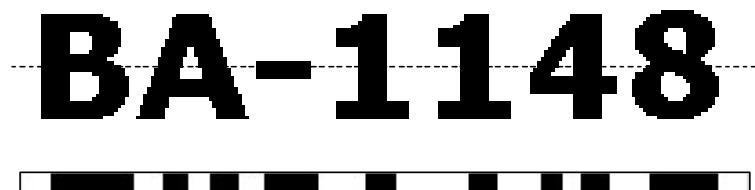


圖 2.8 理想的車牌及其某水平列的灰階值

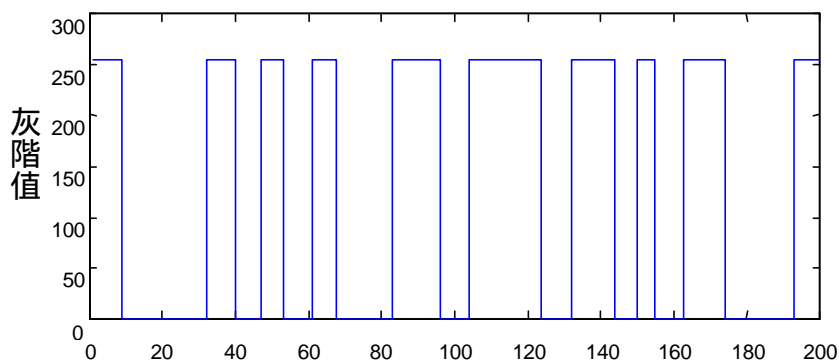


圖 2.9 車牌區域內某一列的灰階值

水平
微分
值

圖 2.10 經 $[-1 \ +1]$ 遮罩運算後的結果

2.1.4 使用垂直微分遮罩對車牌定位並無助益

如果以垂直微分遮罩 $\begin{bmatrix} -1 \\ +1 \end{bmatrix}$ ，即上所介紹的 $\begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}$ 來處理含有車牌的影像，可知道經過車牌每一行至少含有 2 個以上的邊緣點。相較使用水平微分遮罩在含有車牌的列則至少含有 12 個以上的邊緣點，此一現象用來當做找尋車牌定位的基本條件，顯得較為脆弱，因為影像中其他的物件也多呈現此一特徵，如拿此一條件來檢查垂直行是否含有車牌，則造成誤判的機會大增，所以較難做為定位判定的基本條件。

2.1.5 使用 $[-1 \ +1]$ 偵測實際車牌的困難

以上是以理想的車牌影像為例，使用 $[-1 \ 1]$ 來偵測水平列的邊緣點，則效果可說不錯，但受限攝影機器、攝影環境及車牌本身的使用狀況，情況常未必如此理想。邊緣點的變化很少能夠直接由現在計算點是白點，下一點則直接變成黑點；

或者現在計算點是黑點，下一點則直接變成白點，如此變化激烈。通常中間會有過渡灰色區域緩慢變化，且車牌字元顏色未必是絕對黑色，多只是較近似黑色；同樣地，車牌背景顏色未必是絕對白色，多只是較近似白色，如果採用 $[-1 \ 1]$ 的遮罩來運算於水平列時，得到的水平微分值未必能夠反應是否有邊緣點的存在。在圖 2.11，人類的視覺可以很明顯地判定出有 5 次邊緣點的變化，可是使用 $[-1 \ +1]$ 來偵測，如圖 2.12，只有 3 個明顯的邊緣點變化(其微分值的絕對值大於 45)，前 2 次的邊緣點變化則不明顯，在實際的尋找車牌中， $[-1 \ +1]$ 的遮罩將無法確認此前 2 次的邊緣點變化是車牌字元的變化引起或者是雜訊引起，損失了一些邊緣點變化的特徵資訊。



圖 2.11 具有 5 個邊緣點的影像

微
分
值

圖 2.12 經 $[-1 \ +1]$ 遮罩運算後的結果

2.1.6 使用長遮罩來偵測車牌的優點

由於過短的水平遮罩如 $[-1 \ 1]$ 將無法確認變化緩慢邊緣點變化，所以在此試著加長遮罩的偵測範圍，如下圖 2.13、圖 2.14 所示。箭頭往下的線所經過點灰階值則乘上權值-1，箭頭往上的線所經過點灰階值則乘上權值 1，如果兩線中間如果含有點存在，則乘上權值 0，圖 2.13 中其遮罩的大小並不是很大，只能測到整個反差變化的約 $1/3$ ；圖 2.14，取較大的遮罩，則可以測到整個反差變化的 $4/5$ ，可以說較能掌握到整個反差變化。

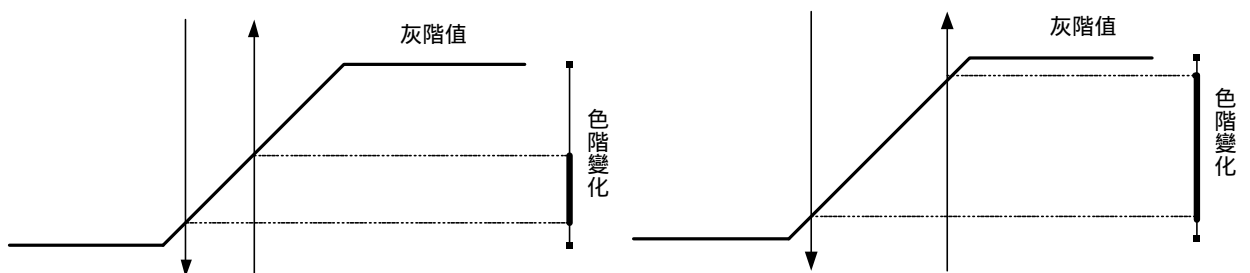


圖 2.13 短遮罩所能偵測到的反差變化

圖 2.14 較長遮罩所能偵測到的反差變化

2.1.7 使用 $[-1 \ 0 \ 0 \ +1]$ 遮罩來偵測實際車牌

再試著以 $[-1 \ 0 \ +1]$ 及 $[-1 \ 0 \ 0 \ +1]$ 來重新計算圖 2.11，如下圖 2.15、圖 2.16。以 $[-1 \ 0 \ +1]$ 來計算微分值，前二個邊緣點的特徵雖然還是不明顯，但是已比 $[-1 \ +1]$ 的效果好多了。以 $[-1 \ 0 \ 0 \ +1]$ 來計算微分值，則 5 個邊緣點則可謂相當明顯 (其微分值均大於 50)。

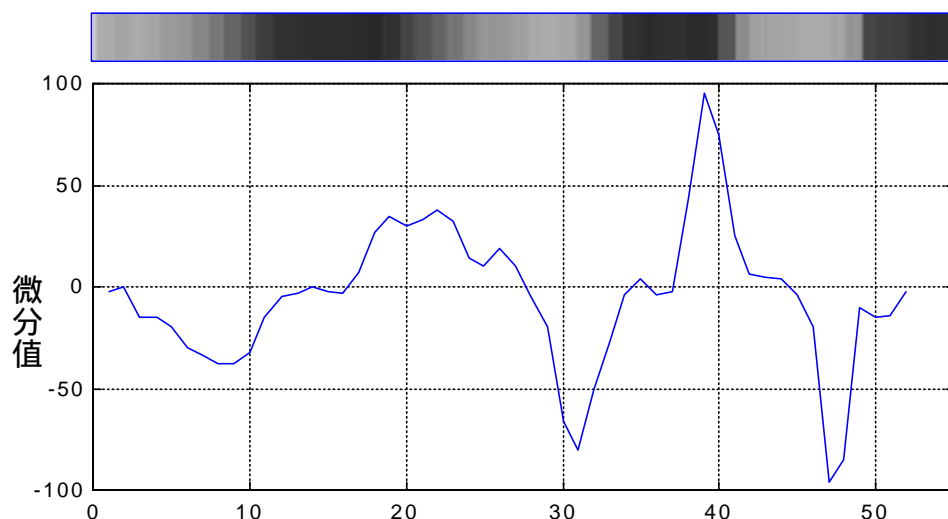


圖 2.15 經 $[-1 \ 0 \ +1]$ 遮罩處理後的結果

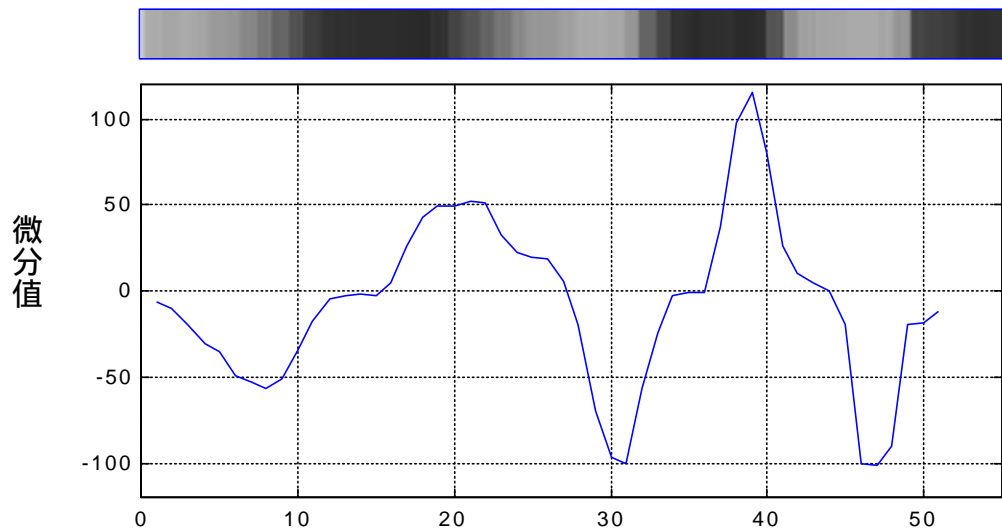


圖 2.16 經 $[-1 \ 0 \ 0 \ +1]$ 遮罩連算後的結果

故可以知道使用矩陣較大的遮罩來可較有效地偵測高反差，但是遮罩的大小不可能使用很大的矩陣。除了因為太大的遮罩矩陣計算出的微分值和現在計算點附近的變化已無太大的關係，且會漏失掉一些邊緣點的資訊。例如採用 $[-1 \ 0 \ 0 \ 0 \ +1]$ 的遮罩運算，即 $g(x, y) = f(x + 2, y) - f(x - 2, y)$ ，在面對寬度 2 格的快速變化是無法有效地的偵測出來，如下例：

此兩個邊緣點未偵測出

$$\begin{array}{cccccccccccc}
 & & & & & \downarrow & & \downarrow & & & & \\
 f(x, y) = & 255 & 255 & 255 & 0 & 0 & 255 & 255 & 0 & 0 & 255 & 255 & 255 \\
 g(x, y) = & & & & -255 & 0 & 0 & 0 & 0 & 0 & 0 & 255 &
 \end{array}$$

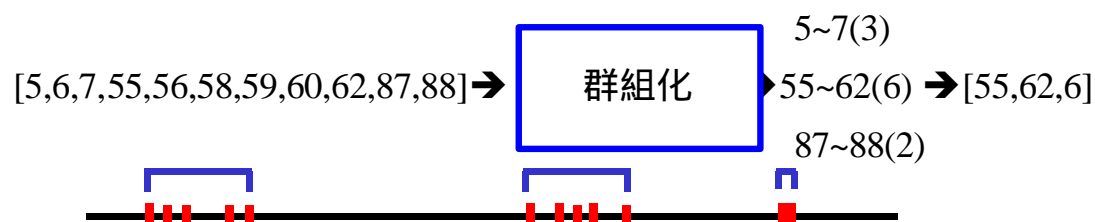
所以基於以上的考量，選用 $[-1 \ 0 \ 0 \ +1]$ 來偵測邊緣點。

2.2 有效列的認定

這節是主要介紹有效列的認定，所謂的有效列認定，是指在逐列運算中，可能含有車牌部分影像的水平列。其中為了儘量適應各種光度變化下所攝得的車牌影像，對於多少程度灰階變化才認定是邊緣點，是以動態值來來臨界值，這是和以往處理方法是有所不同。

2.2.1 群組化

群組化即對一排序過後的數列(由小到大)，當連續相連的兩個數不超過預先設定的 N 時即合成一群組，故如此集中在一起數字便會合成一群組，距離此一群組比較遠時，便會合成另新一群組，同時最後紀錄其群組的頭一數字及最後一個數字及本身群組的大小。如下例($N=3$)：



群組化結果即代表數列群聚的情形，通常是挑最大的群組來做處理，亦即為群聚數量最多的群組。如上例，即 $55 \sim 62$ 這個群組含有 6 個最多數量點。群組化在接下來的處理過程會大量使用。

2.2.2 固定微分臨界值的有效列認定

使用 $[-1 \ 0 \ 0 \ +1]$ 來偵測具有高反差特徵的邊緣點，在單一系列中的由左而右進行每一點遮罩的運算，因為所心關是每一列的邊緣點點數，不必去區分水平微分遮罩算出微分值的正負，只要根據微分值絕對值的是否大於預先設定 a_1 來判定是否為邊緣點(即高反差點)。如以下的表示：

判定運算點是否為高反差點：

$$g(x, y) = f(x + 2, y) - f(x - 1, y)$$

如果 $|g(x, y)| \geq a_1$ ，則 (x, y) 點附近具有高反差變化的特性 (2.5)

(其中 a_1 為「認定高反差點的臨界值」)

設定 a_1 為 50。考慮到逐點計算，會遇到單一灰階高反差變化會重複計算好幾次，所以在(2.5)式確認後必須再依據以下的規則來進行高反差點的認定：

高反差認定規則：

如果 現在高反差點和同列中的前一個高反差點的距離小於 3 點，

則 累記錄現在高反差點列號於 $\{a_1\}$ 及其高反差值 $(|g(x, y)|)$ 於 $\{a_2\}$ ；

或者如果 現在高反差點和同列中的前一個高反差點的距離小於 3 點，

且兩者水平微分值異號，

則 累記錄現在高反差點列號於 $\{a_1\}$ 及其高反差值 $(|g(x, y)|)$ 於 $\{a_2\}$ ；

或者 其他情況，

則 不做任何動作。 (2.6)

($\{a_1\}$ 數列為「高反差點行號數列」， $\{a_2\}$ 數列為「高反差數值數列」)

以上(2.6)規則就是如果現在高反差和同列中的前一個高反差點的距離小於 3 點，則認定彼此相關性不大，是為兩個獨立的高反差點，同時累記錄其高反差的列號及高反差值 $(|g(x, y)|)$ 於 $\{a_1\}$ 及 $\{a_2\}$ ；如果彼此行號距離小於 3，則兩點有可能是重複點，檢驗兩者高反差點是否同樣屬性(即遮罩計算出的微分值是正或亦負)，如果不同號，則兩個高反差點應是不相關的高反差點，再累記錄其行號 (y) 及高反差值 $(|g(x, y)|)$ 於數列 $\{a_1\}$ 及 $\{a_2\}$ 中；如果都不符合以上情況，即使現行

高反差點符合(2.5)的條件，但不符合(2.6)，也不計入高反差點的認定。

在經過(2.6)處理，再透過群組化，考慮到一些比較模糊不清的車牌及雜訊的影響，檢驗最大群組的個數是否大於 a_2 (a_2 為「認定為有效列的最少高反差點個數」，這裡預設為 8)點以上，如果成立，此列即為有效列。

2.2.3 使用固定微分臨界值進行有效列認定的困難

透過以上 2.2.2 程序，計算出效果不是很好，常常一些含有較模糊或對比不強車牌的效列無法檢索出來。試著調降 a_1 ，一些原本較模糊或對比不強車牌的有效列可以逐漸地檢索出來，可是同時也造成另一些非含車牌的水平列被檢索出來的機會也大增，增加其尋找車牌的困難度。故取較高的 a_1 來認定高反差的邊緣點，影像對比不夠強的影像，就可能錯失找到車牌的位置；如取較低的 a_1 ，則有可能易受非車牌的物件干擾。所以此一問題對於 a_1 值的大小和有效列的檢索是權衡(trade off)問題。同樣如果試著去調降 a_2 ，也會面臨到同樣的問題。綜合以上的考慮，目前是採用一種較為動態調整臨界值認定的作法，相較原來採用固定 a_1 作法要去認定影像品質變化極大的多張影像，能夠較有效確認正確有效列，進而找到車牌位置。

2.2.4 動態微分臨界值的有效列認定

動態微分臨界值的有效列認定，基本是先採用較低標準來認定高反差點，再依據所得高反差值來動態設定臨界值，再進行有效列的認定。

其步驟前面和固定微分臨界值的有效列認定相同，只是(2.5)中的 a_1 先取較低值為 25，再經過(2.6)的處理，可得 $\{a_1\}$ 及 $\{a_2\}$ 兩數列。再進行以下動態微分臨界值的有效列認定：

動態臨界值有效列認定規則：

如果 $\{a_2\}$ 的數量大於 a_2 ($a_2 = 6$)

則 $a_3 = \{a_2\}$ 平均值乘上小於 1 的數值(如 0.9)；

再找尋 $\{a_2\}$ 大於 a_3 的元素，將其所對應行號 $\{a_1\}$ 的元素記錄在 $\{a_3\}$ ；

如果 $\{a_3\}$ 的數量大於 a_2

則 將 $\{a_3\}$ 做群組化(群組化參數 60)，找出最大群組；

如果 最大群組的數量大於 a_2

則 此列是有效列。 (2.7)

(其中 $\{a_3\}$ 數列為「動態認定高反差點行號數列」， a_3 為「動態認定高反差點的臨界值」)

如果認定現在計算是有效列，則累記錄此列最大群組的開頭及結尾於「有效列中最大群組開頭數列」 $\{a_5\}$ 、「有效列中最大群組結尾數列」 $\{a_6\}$ ，累記錄有效列的列號於「有效列列號數列」 $\{a_7\}$ ；反之，則不予記錄。

從以上的運算，知道是先在現行計算列中取一個較低的臨界值來篩選較可能的邊緣點，再從可能的邊緣點的特性動態設定此一臨界作，找出屬於車牌區域的邊緣點。之前所找到數量大於 6 的數列 $\{a_2\}$ ，如果整體高反差值偏低(表示車牌模糊不清或對比不高)，則使用較低的臨界值來篩選；如果整體高反差值較高(表示車牌較清楚或對比較為明顯)，則使用較高的臨界值來篩選。再加上再次篩選數量點的臨界值，如此對影像清楚或是模糊不清的車牌影像，則能夠較為有效地偵測出有效列，不會發生在模糊不清的影像中找到非屬車牌的有效列，或者在對比較強的影像中易找到非屬車牌的物件的錯誤。

2.3 初步掃描

車牌定位的一般基本做法，在影像中逐列的掃描，先找出有可能含有車牌的有効列，再從這些有効列中找出真正車牌的位置。可是如果在影像中逐列掃描，可以發現大部分是在不含車牌的區域內尋找，相當地無意義且費時。於是試著先大約找到含有車牌的區域，再從這已縮小的局部範圍內，去找尋真正的車牌的位置。即先做初步掃描定出影像中部分含有車牌的區域，再做細步掃描來真正找出車牌的位置。這樣的處理步驟，可以減少不少時間在無意義的尋找，節省不少的運算時間。

初步掃描則是先透過 2.2 節處理方式，每隔 10 列進行有効列的判定，得到所有認定是有効列的列號數列 $\{a_7\}$ ，和有効列中最大群組的開頭數列 $\{a_5\}$ 、結尾數列 $\{a_6\}$ 。最大群組的開頭數列即車牌的左界的可能，最大群組的結尾數列即車牌的右界的可能，有効列的列數數列開頭及結尾即約可知道車牌的上下界。所以據此，利用 $\{a_5\}$ 、 $\{a_6\}$ 、 $\{a_7\}$ 來群組推算出車牌可能大約的範圍。

先檢驗單一有効列的對應的最大群組的有效距離是否大於「有効列最小長度」 b_1 ($b_1 = 40$)，即檢驗單一有効列的最大群組的結尾減去開頭是否大於 b_1 ，因為基本上車牌含有一定的長度，如此從 $\{a_5\}$ 、 $\{a_6\}$ 、 $\{a_7\}$ 中刪去不符合條件的有效列，再進行以下的區塊化：

對有効列進行區塊化：

第一個有効列屬於第一個區塊；

進行以下動作，計算有効列從第二列至最後一列

如果 目前計算有効列和前一個有効列的列號小於 b_2 ($b_2 = 30$)

且 彼此最大群組開頭差距小於 b_3 ($b_3 = 60$)

且 彼此最大群組結尾差距小於 b_3 ($b_3 = 60$)，

則 目前有效列和前一個有效列屬於同一個區塊；
否則 目前有效列是屬於一個新的區塊。 (2.8)

(其中 b_2 是「同一區塊內相鄰兩有效列最大列距」, b_3 是「同一區塊內相鄰兩有效列最大左、右界行距」)

經過(2.8)的步驗，便可以得到有限數量的區塊，區塊的上界即區塊含有有效列的最小列號，區塊的下界即區塊含有有效列的最大列號，區塊的左界即區塊含有有效列中最小的群組開頭，區塊的右界即區塊含有有效列中最大的群組結尾。如圖 2.17 所示，虛線代表為初步掃描所經過的列(每隔 10 列)，粗線即代表認定有效列中的最大群組，其最大群組的開頭即粗線的開頭；其結尾即粗線的結尾，經過以上的步驟處理後，可得到三個以斜線代表的區塊。其中最下方的區塊含有車牌。

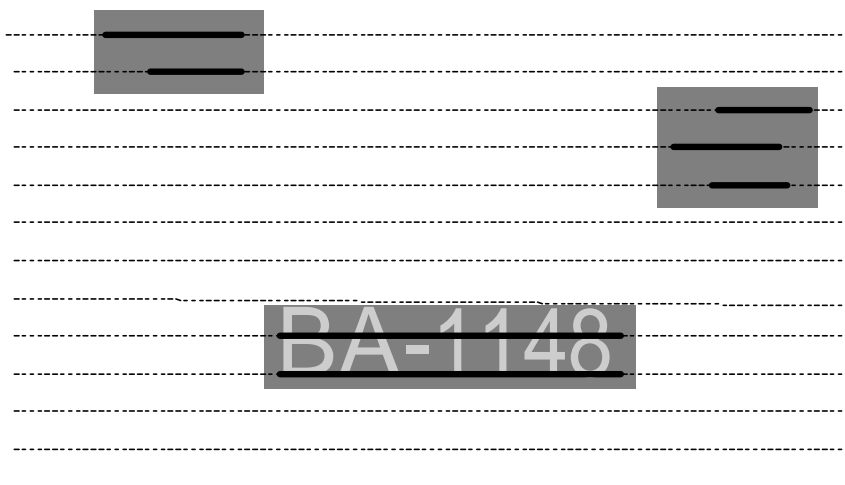


圖 2.17 經初步掃描後所得到的區塊

雖是有限量的區塊，理論上其中一個區塊應該含有車牌，可是究竟要從哪個區塊找起？由觀察可以得到經驗定律，在車牌下方干擾物件一般比在車牌上方干擾的物件來得少，所以可以由下而上，來確認區塊是否含有車牌(即後續的細步掃描)。另稱正在進行確認的區塊為「預備細步掃描區域」，同時為了避免漏失掉區

塊邊緣點的資訊，會再稍微向擴大一定的範圍來確認是否含有車牌。

2.4 消除奇偶列交錯(Interlace)現象

整個系統是採用 CCD 攝影機做拍攝，再從這動態拍攝的影片中擷取為影像。現行的 CCD 攝影機在設計上考慮到傳輸速度的問題，多數是採用交錯掃描，先由奇數列由左掃至右，再由上而下，當奇數列的訊號點全部都讀取完，才再讀取偶數列。偶數列的讀法如同奇數列，由左至右，再由上而下，當偶數列的資訊都讀取完，才算真正讀完一個畫面(frame)的訊號，下一個畫面同樣再由奇數列開始，如此周而復始。

2.4.1 交錯現象形成的原因

雖然現行 CCD 攝影機讀取完這樣一個畫面的速度可以說是蠻快，可是遇到速度較快的目標物，仍然會因有時間差，造成奇數列、偶數列再合成一個完整畫面時，因資訊的時間基點不同，使得目標物會有交錯開的現象。現象輕者，目標物尚隱約可見，現象重者，則無法知道原來目標物為何。為了仔細觀察此一現象，取出影像中含有車牌的部分區域，如圖 2.18，由於圖中的車牌(車輛)移動速度較快，造成交錯現象，其中車牌中的字元已模糊不清，車牌的邊界線更是不明顯，造成後續處理的困擾及問題。



圖 2.18 交錯現象

2.4.2 消除交錯現象

消除這樣的現象，可以參照 Photoshop 的處理方式，只取單一系列的資訊，即奇

數列或者偶數列，另一列則以內插的方式來處理，不用原來的資訊，這樣可以利用內插使得影像顯得較為連續，不會有交錯的現象。

例如保留原來偶數列資訊，則奇數列的資訊則以相鄰兩列資訊做線性內插而成，其數學表示式如下， f 的右下角記號即表示所在列：

$$\begin{aligned} \{ f_{even}(x, y) \} &= \{ f(x, y) \} \\ \{ f_{odd}(x, y) \} &= \left\{ \frac{f(x, y-1) + f(x, y+1)}{2} \right\} \\ \{ f_{new}(x, y) \} &= \{ f_{even}(x, y) \} + \{ f_{odd}(x, y) \} \end{aligned} \quad (2.9)$$

得到消除交錯畫面的圖如下圖 2.19：



圖 2.19 利用內插來消除交錯現象

至於要保留奇數列，偶數列內插完成；或者要保留偶數列，奇數列內插完成，哪一種方法比較好？其實都可以有效消除交錯現象，雖然得到新影像並不相同，彼此物件的灰階值會有稍微差異，物件的位置也會稍微偏移。不過因為接下來所關心是如何把車牌分離出來，不是車牌的真正位置是位於影像的哪個區域，而是直接拿處理完新的畫面做後續的細步掃描，所以採用哪一種方法其都可以。

另外得到的影像中如有交錯現象的產生，通常是全面性地，非局部區域。但是接下來只準備在計算中的預備細步掃描區域尋找真正消除交錯現象方法只在預備細步掃描區域，非整張影像，以減少不必要的內插運算而浪費時間。

2.5 細步掃描

2.3 節的初步掃描是在含有車牌的影像中，每隔 10 列進行一次有效列的認定，先決定出預備細步掃描區域。再由此目前計算中預備細步掃描區域中每隔 1 列便進行有效列的認定，即細步掃描，以得到真正車牌的位置。

細步掃描的過程，和 2.3 節的初步掃描的程序大致相同，只是現在細步掃描是在預備細步掃描區域每隔一列進行有效列的認定，如同初步掃描的參數設定，則可以得到所有認定是「有效列列號數列」 $\{d_1\}$ ，和所有「有效列最大群組的開頭數列」 $\{d_2\}$ 、「有效列最大群組的結尾數列」 $\{d_3\}$ 。

因為車牌所經過的有效列將會在 $\{d_1\}$ 呈現聚集，所以將 $\{d_1\}$ 做群組化(群組化參數取 6)，再取其中最大群組的開頭即當做車牌的上界；最大群組的結尾即當做車牌的下界。在這裡群組參數並不取 1，是為了讓定位步驟避免因雜訊而將整個車牌所在區域誤認為多個物件區域。車牌的左界是根據 $\{d_2\}$ 數列來決定，雖然車牌左界應是單一固定值，但是事實上同樣地會在 $\{d_2\}$ 數列呈現群聚現象，所以同樣做群組化(群組化參數取 12)，再取最大群組的開頭做為車牌的左界。車牌右界則是根據 $\{d_3\}$ 數列來決定，其推算原理同車牌左界，只是現在是取群組化後的最大的群組的結尾做為車牌的右界。

2.6 微調整邊界程式

經由(2.5)的細步掃描的步驟，就可以決定車牌的位置，可是就國內的車牌而言，車牌本身除了含車牌字元區域，尚包含一些非車牌區域，如螺絲孔所鎖的四個位置及標明發牌單位，這些區域在車牌辨識並不是所感興趣區域，會造成往後車牌字元分割的困擾。如圖 2.20 是細步掃描後結果，虛線所圍影像區域才是所感興趣車牌字元區域，其他以外的影像區域則是非車牌字元區域，包含了四個螺絲孔及標示車輛發牌的單位(如「臺灣省」)，此部分會干擾到後續的處理步驟。所以希望透過這節微調整邊界程序，去除車牌影像中非屬車牌字元區域，同時微調整細步掃描後因雜訊影響的車牌邊界，最後只留下車牌字元區域。

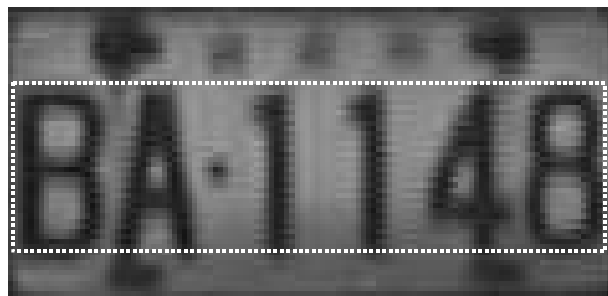


圖 2.20 車牌的車牌字元區域

2.6.1 微調整左右界

微調整車牌左右界是用來消除因雜訊引起的左右界的誤判，根據的原理是如果從非車牌區域到車牌區域，其車牌邊界區域的單行水平微分值的總和相較非車牌區域為大。

先求一「微調整左右界臨界值」 r_1 為：車牌高 $\times (12 + \text{車牌區域內水平微分值平均除以 } 200)$ ，其中車牌區域內水平微分值平均就是以 $[-1 \ 0 \ 0 \ +1]$ 計算後的結果。再依據以下的規則決定左右界是否進行微調整：

微調整左界：

進行以下動作從(車牌左界前 5 行)逐漸遞增一行至(車牌左界後(車高 $\times 1.5$))

行)

如果 目前計算單行的水平微分值大於 r_1

則 調整車牌左界為目前計算單行的行號 且 中斷跳出迴圈；

如果 調整後左界和調整前左界差異不大 或 以上迴圈未被中斷跳出

則 車牌左界則不進行調整，仍以未調整車牌左界為準； (2.10)

微調整右界：

進行以下動作從(車牌右界後 5 行)逐漸遞減一行至(車牌右界前(車高 \times 1.5)

行)

如果 目前計算單行的水平微分值大於 r_1

則 調整車牌右界為目前計算單行的行號 且 中斷跳出迴圈；

如果 調整後右界和調整前右界差異不大 或 以上迴圈未被中斷跳出

則 車牌右界則不進行調整，仍以未調整車牌右界為準； (2.11)

2.6.2 微調整上下界

微調整上下界是去除非車牌區域，就是螺絲孔及標示車輛發牌單位文字。不過由於此區域具有邊緣點高度密集，故不適合採用類空間頻域的方法來去除。所以在此是採用時域的方法來去除。

先求一臨界值為「微調整左右界臨界值」 r_3 ：

r_2 為車牌區域的灰階值平均；

如果 r_2 大於 107，

則 $r_3 = r_2 \cdot 0.9$

否則

$r_3 = r_2 \cdot 0.8$ 。 (2.12)

計算每列中大於 r_3 的點數量，如果大於車高的 0.5 倍且小於車寬的 0.8 倍，則

保留所對應的列號，反之，則去除。其理由是經以 r_3 為標準來做二值化後的螺絲孔所在列的黑點應不會大於目前計算車牌高的 0.5 倍，且車牌字元區域所在的列因為字和字之間或本身字有空白間隔，不可能全部都是黑點，所以單列黑點累積量不符合小於長的 0.8 倍，也去除此列。再將所保留下來的列號做群組化，群組化參數取 1，同樣地取最大的群組開頭是新上界，群組的結尾即是新下界。

2.6.3 再微調整左界

在 2.6.1 節的部分已做過一次微調整左右界，主要是消除車牌影像變化不大的地方，但是在 2.6.3 節微調整左界則主要是做補償。當車牌影像的成像品質太差，字的邊緣模糊，且車體本身為深色系，使得車牌的左界不明顯，使得車牌最左邊的字元會和車體連在一起，以致於在後續的字元分割上，分割出成的最左邊車牌字元將成為有缺陷的字元(特別字元左邊有一直豎最易出此一困擾，如 B、D、E、F、H、K、L、M、N、P)。如下圖 2.21 是細步掃描後的結果，經過 2.6.1 節的處理或者在之前的細步掃描中，即處理成圖 2.22，發現 B 之前的左邊一直豎因和車體相連被認為是屬於車體部分，以致 B 在處理中受到破壞，最後使得後續分割出字元無法進行辨識。

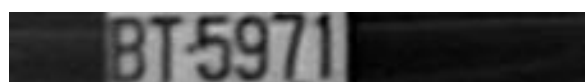


圖 2.21 細步掃描後的車牌影像

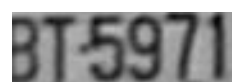


圖 2.22 未補償前的車牌影像

雖然在深色系車輛且車牌呈像模糊不清中易發生此一問題，但是可以利用車牌字元左上方仍有明顯的界限(如下圖 2.23 車牌左上方的實線方框處)，利用此一界限來加入車牌左界的判定，以補足字太胖將車牌左界擠掉的問題，特別是對外形屬方正的英文字母。在右界則較不會有這樣的問題，因為多其為數字，少有外形方正的情況發生擠掉邊界線的問題。

實現以上構想的方法，仍使用 $[-1\ 0\ 0\ +1]$ 這個遮罩內，在實線方框內，逐列由左而右確認灰階值是否有高度反差變化，在此只記錄每一列的第一點的高灰階反差變化點的位置，再將此一新增數列加入 2.5 節中車牌左界數列 $\{d_2\}$ ，再重新以群組化來計算車牌的左界，就可以得到如圖 2.24 車牌左邊字元未被刪除掉的結果。



圖 2.23 車牌影像補償左界所計算範圍(實線方框)

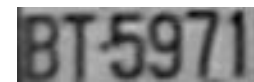


圖 2.24 補償後結果

2.7 車牌的確認

由前幾節的討論中，知道其整個車牌定位的流程是先經由 2.3 節初步掃描而決定預備細步掃描區域，再由 2.5 節細步掃描在預備細步掃描區域中決定真正車牌位置，再由 2.6 節微調整邊界程式來調整車牌的邊界。可是在 2.3 節初步掃描中，決定預備細步掃描區域，根據經驗定律，是先尋找最下方的區塊來當做預備細步掃描區域，可以有時候最下方的區塊不見得就是含有車牌的區塊，可能在底部倒數的第二個、第三個區塊。所以必須有一些確認車牌的條件來確定經過以上步驟處理的結果是否為車牌。

採取確認車牌的三個條件是：

- 車牌長度大小需小於 350 點
- 高度需大於 20 點
- 其長寬比介於 3.10 及 6.4 之間(長寬比即長除寬的比例值，規格上應是 4 左右)

如果全符合以上三個條件，則認定目前運算結果是車牌區域，則接續後面的車牌字元分割；如果不符合任何一個確認條件，則再往上找最接近的區塊當做預備細步掃描區域，再如(2.4)(2.5)(2.6)做細步掃描、消除交錯現象、微調整車牌邊界。如此最多可執行三次的程序，也就最多由下往上找三個區塊來確認是否含有車牌區域，如果三次以上不符合或者已經沒有區塊可供運算，則認定此張影像未含有車牌或者車牌定位失敗。不過當然是希望頭一次計算區塊便即是含有車牌的區塊，這樣可節省尋找的時間，前提是在車牌下方沒有太複雜的物件，如車輛本身裝飾物來干擾區塊的判定。

第三章 車牌字串分割及字元正規化

含有車牌影像經過第二章一連串的步驟處理，可以得到車牌中的車牌字串區域，接下來就將其進行車牌字串分割。即把原本 6 個或 5 個相連在同一區域的車牌字串分割成一個個獨立的字元方塊，再把一個個獨立的字元區正規化成 30x20 大小的方塊，以便後續智慧型類神經圖樣比對法來進行車牌字元辨識。

3.1 節前處理包含了二值化及交集運算兩個主要步驟，二值化是採用區塊二值化以求得較佳的效果，交集運算是用來消除兩個字元間連接單點及孤立雜訊點。3.2 節的垂直投影量分割法及外圍輪廓分割法是分別介紹兩種分割法，並解釋在此為何外圍輪廓分割法會比垂直投影量分割法更適合來分割車牌字串。3.3 節分割點的尋找則是根據外圍輪廓分佈圖來決定分割點，但受限影像品質不佳，不能夠採齊一的臨界值來認定分割點，必需採用動態臨界值來找出分割點。利用分割點找出字元群組後，尚需判定字元群組中是否含有非字元的字元群組，加以刪除。再試著找尋破折號，利用破折號的特性以增加車牌字串分割的成功率。3.4 節字元正規化則是將大小不一的車牌字元正規化 30x20 大小的車牌字元，正規化的方法包含最近整數法、雙線性內插法，其各有優劣點，在此，本系統是採用雙線性內插法以求得正規化後較佳的效果。

3.1 前處理

以車牌字串區域進行車牌字串分割之前，需做一些前處理。前處理包含了兩項的工作，一是先做二值化，以便以後的計算及分析；另一則後再做「交集」(AND) 運算，設法消除兩個字元間連接單點及孤立雜訊點。

3.1.1 二值化

為了避免車牌出現會有局部亮度變化，所以對車牌字串區域做分區塊二值化，

且每個區塊之間會互相重疊(overlap)。在這裡是分成五個區塊，如下圖 3.1，每個區塊的二值化臨界值則是取區塊內所有點灰階值平均再乘上一權值(如 0.88)，乘上權值的理由是根據觀察一般車牌屬於背景顏色白點的數量應該比字元前景顏色黑點數量還要多。在這裡是由上而下分成五個區塊，每個區塊互相重疊，再分別做二值化。

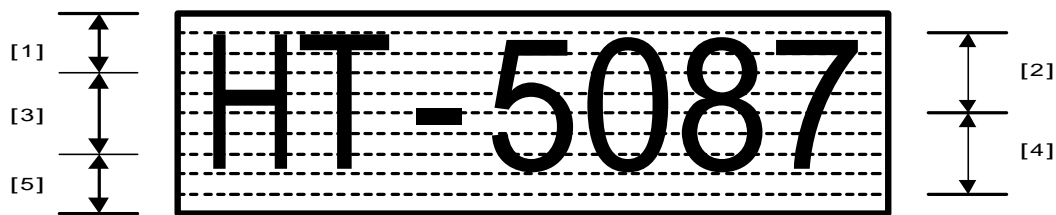


圖 3.1 分五個互相重疊區塊做二值化

3.1.2 以交集運算消除孤立點及字元間連接單點

「交集」運算是來消除兩個字元間連接單點及孤立雜訊點。把二值化後的黑點當做邏輯的 1，白點當做邏輯的 0，交集運算就是如果計算中的相鄰兩點都是黑點，則輸出的結果是黑點；其他相鄰兩點的組合，如黑點-白點、白點-黑點、白點-白點，則輸出的結果是白點。其垂直方向及水平方向的交集運算表示式如下， $A(x, y)$ 是二值化後的車牌， $A_1(x, y)$ 是經過交集運算後的結果。

水平方向的「交集」運算：

$$\begin{aligned} & \text{IF } A(x, y) \text{ is Black AND } A(x+1, y) \text{ is Black, then } A_1(x, y) \text{ is Black;} \\ & \text{else } A_1(x, y) \text{ is White.} \end{aligned} \quad (3.1)$$

垂直方向的「交集」運算：

$$\begin{aligned} & \text{IF } A(x, y) \text{ is Black AND } A(x, y+1) \text{ is Black, then } A_1(x, y) \text{ is Black;} \\ & \text{else } A_1(x, y) \text{ is White.} \end{aligned} \quad (3.2)$$

在以上的陳述，可能無法了解到交集運算的好處。可以觀察以下圖例，圖 3.2

是二值化後的結果，有兩個半圓形獨立的物件在兩側，中間彼此有一點相連，此外還有一點孤立雜訊點，這樣情形會造成以後車牌字串分割的困擾。如果經過垂直方向的交集運算，其結果如圖 3.3，發現除了去除孤立的雜訊點，原本兩個物件相連的單點也去除，達到了消除孤立點及字元間連接單點的效果。

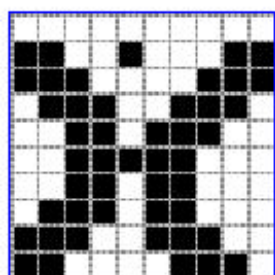


圖 3.2 未經過垂直交集運算

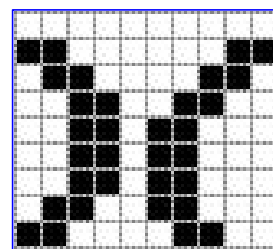


圖 3.3 經過垂直交集運算後的結果

不過發現經過垂直方向交集運算時，會發現物件的某些外圍會被縱向侵蝕掉一層，所幸攝得車牌字元寬度多大於 3 以上，不會造成太大的困擾。同時，字串分割是做縱向分割工作，所以只對二值化後的車牌，做一次垂直方向的交集運算。

事實上在整個前處理步驟，為了減少運算的時間，二值化和交集運算並不是分為兩個獨立步驟前後進行，是混合成一個步驟。如二值化之前的車牌字元影像是 $A(x, y)$ ，分區的二值化臨界值是 $Threshold$ ，經二值化及交集運算後的結果是 $A_1(x, y)$ ，其敘述式可表示為：

$$\begin{aligned} & \text{IF } A(x, y) \leq Threshold \text{ AND } A(x+1, y) \leq Threshold, \text{ then } A_1(x, y) \text{ is Black;} \\ & \text{else } A_1(x, y) \text{ is White.} \end{aligned} \quad (3.3)$$



3.2 垂直投影量分割法及外圍輪廓分割法

字串分割在一般字元辨識軟體或其他有關字元辨識研究中多是採用垂直投影量分割法。在此先介紹垂直投影量分割法，但是會發現垂直投影量分割法並不適用於本系統所取得的圖像，使用做為分割依據的效果並不好。所以採用外圍輪廓分割法來分割車牌字串，經測試後，發現效果都比垂直投影量分割法來得好。

3.2.1 垂直投影量分割法

垂直投影量分割法是在欲分割字串區域逐行統計二值化後近字元顏色的色階點數，統計完單行的點數量即為垂直投影量，其分佈圖即為垂直投影量分佈圖。對照原字串區域及垂直投影圖的數值變化，可以知道由於在字串分割處多有一小塊屬於背景顏色以利人類視覺區分的區隔區域，所以在字元分割處其垂直投影量應該是相對最小，在理想狀態下，字串區域影像對比夠明顯，且沒有雜訊的干擾，在分割處位置的垂直投影量應該為零，在垂直投影量分佈圖上即波谷位置，如圖 3.4 及圖 3.5。

圖 3.4 車牌字串區域

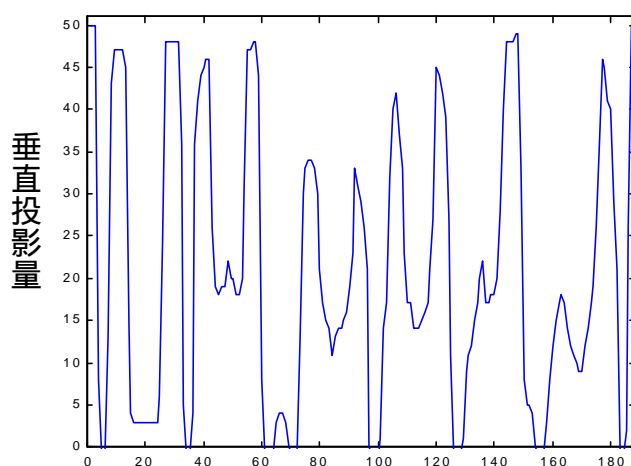


圖 3.5 垂直投影量分佈圖

3.2.2 使用垂直投影量分割法可能遇到的問題

圖 3.4 車牌字串其對比夠明顯及清晰，車牌字元區域的分割點即是圖 3.5 中垂

直投影量為零處，相當明顯而易於處理。可是把垂直投影量分割法應用在現行所取到的車牌，許多車牌字串處理的效果並不是很好，特別是遇到攝到尺寸較小的車牌或者對比不強的車牌，如圖 3.6 及圖 3.7。由圖 3.6 來看，知道 B、0 之間分割處對應圖 3.7 應為波谷位置(即橫向座標值約為 15 的位置)，可是相較於附近周圍的垂直投影量不見得是相對低點的位置，反而在 B 字母及 0 字母的中間位置(橫向座標值為 10、20 的位置)比較像是波谷。B、0 字母左右兩邊分割點的位置應是分別為橫向座標值 3 和 15、16 和 26，若單純從圖 3.7 垂直投影量分佈圖來看，座標值 10 到 20、20 到 27 反而比較像是有兩個字母，造成誤判。造成這樣現象是因為 B、0 字母中間部分的黑點量本來就比較少，再加上 B 和 0 字母又相連在一起，所以無法圖 3.7 來判定出正確的分割點。同樣地，9、6 字母中間部分的黑點量比較少，再加上 9 和 6 字母又相連在一起，使得 9 和 6 在圖 3.6 應為相對低點波谷訊號的垂直投影量反而沒有周圍兩個相對低點位置(約橫向座標值 35 及 45 處)的垂直投影量來得低，若單純從垂直投影圖來觀察，反而在橫向座標值 35 到 45 之間比較像是有一個字的範圍，造成了分割點的誤判。此一誤判則是因為 9、6 兩字母產生了連字現象，以致實際分割點訊號的垂直投影量值未必是相對最小。

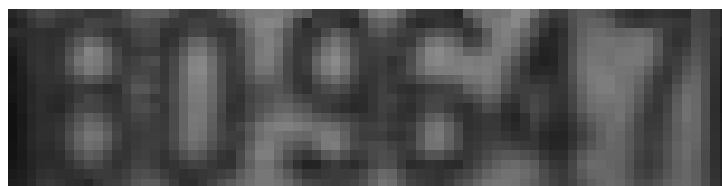


圖 3.6 車牌字串區域

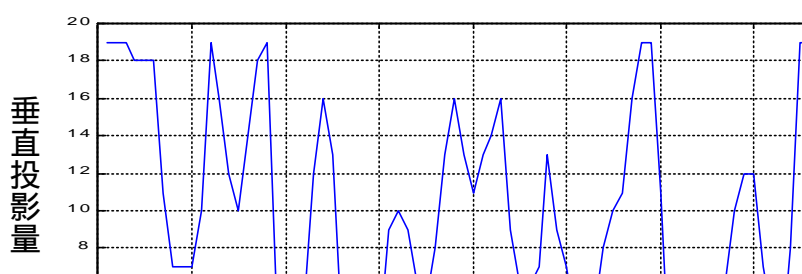


圖 3.7 垂直投影量分佈圖

故可知垂直投影量分割法對某些攝得尺寸較小的車牌及模糊不清車牌在進行車牌字串分割時，會因為波谷點錯誤訊號易造成分割點的誤判。不過如果車牌尺寸大小及清晰度在某一程度以上時，垂直投影分割法仍然可有不錯的效果，可是在收集的圖形樣本中，仍然有相當數量的尺寸較小或較模糊的車牌，所以在此，並不使用垂直投影法，改用外圍輪廓法來進行車牌字串分割。

另外在許多國外有關車牌字元辨識研究多採用垂直投影量分割法，這是由於國外車牌和國內車牌的規格不同，國外車牌如美國及歐洲國家的車牌其車牌字元間的空白間距明顯 相 較 於 國 車 牌 字 元 間 距 來 得 大 許 為緊密，如果遇到顯像較差的車牌，則可能會有兩連字，甚至三連字、四連字的問題，故在此國內車牌不適用垂直投影量分割法。

3.2.3 外圍輪廓分割法

由 3.2.2 可以知道，垂直投影量分割法無法對系統所取得的樣本有很好的分割效果，所以在此是採用外圍輪廓分割法來做為基本分割的依據。

外圍輪廓法分割法[12]是找字元的外圍輪廓來做分割的依據。因為準備是做字串的縱向分割，所以找尋單行輪廓的方法分別由上往下找及由下往上找，亦即為字串上輪廓及下輪廓。上輪廓的找法，即在單行中由最頂端點往下找，當遇到黑點時，即記錄從最頂端到遇到黑點之間的點距離，是為上輪廓值，如果都沒有遇到黑點時，則設車牌高為上輪廓值；同樣地，下輪廓亦由最底端由下往上找，當遇到黑點時，即記錄從最底端到遇到黑點之間的點距離，是為下輪廓值，如果都

沒有遇到黑點時，則設車牌高為下輪廓值。以上單行外圍輪廓找法，即由車牌左邊逐行往右計算，每一行都會有對應的上輪廓值及下輪廓值，當計算完時，則可以得到一連串的上輪廓值數列及下輪廓值數列。再配合對應的橫向座標，即為上輪廓值分佈圖及下輪廓值分佈圖。

圖 3.8 是原車牌字串區域。圖 3.9 是對應的上輪廓值分佈圖，為了符合一般直覺觀察，分割點即是分佈圖中波谷位置，所以圖 3.9 是車牌高減去上輪廓值。圖 3.10 則是對應的下輪廓值分佈圖，同樣是車牌高減去下輪廓值。圖 3.11 則是上下輪廓值合繪在同一張圖，車牌高減去上輪廓值以實線表示，下輪廓線則以虛線表示，可以發現上下輪廓線類似包絡線，每個包絡線所圍出近似封閉區域，事實上則是一個個獨立的車牌字元。

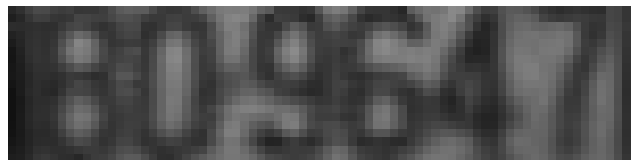


圖 3.8 車牌影像

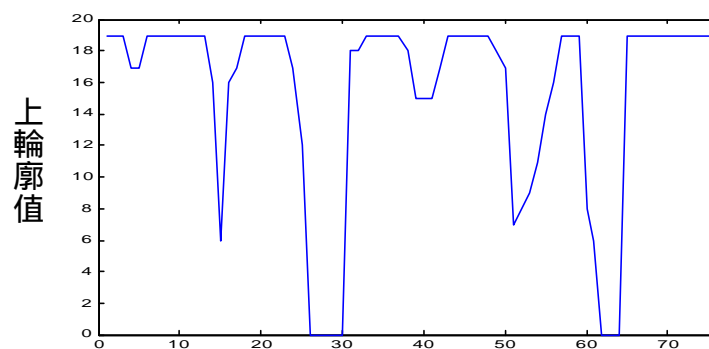


圖 3.9 上輪廓值分佈圖

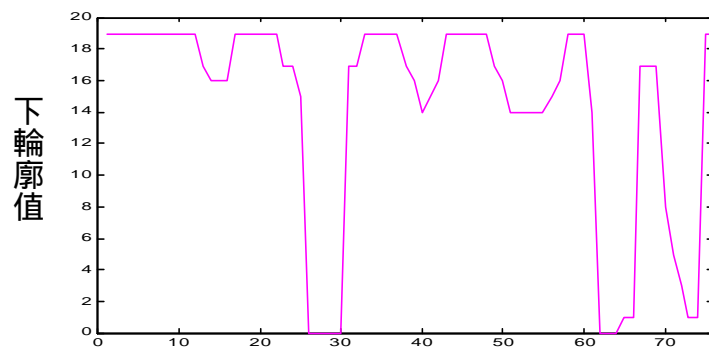


圖 3.10 下輪廓值分佈圖

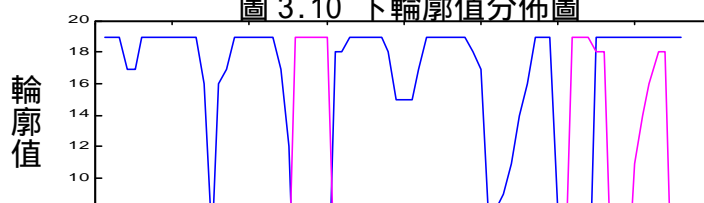


圖 3.11 外圍輪廓值分佈圖

圖 3.9 的上輪廓值分佈圖對照圖 3.8，可以看到在字串分割處，都有明顯的波谷(相對低點)的訊號，不會發生在垂直投影量分割法中非分割處的相對低點比應分割處的相對低點來得低的缺點，造成分割點的誤判。同樣地，圖 3.9 的下輪廓值分佈圖亦是如此。明顯地，由外圍輪廓來做分割依據，顯然比起以垂直投影量來做分割依據，其效果來得好。特別是對於上下輪廓是半圓形或近似半圓而相連在一起的字串效果特別好。

以上的討論，包含兩種輪廓值，那究竟要使用上輪廓值分佈圖做為分割點的判定，還是取下輪廓值？為了能取到兩種輪廓值的優點，在這裡是混合使用兩者資訊，再組合成單一的最終輪廓值。其單行最終輪廓值的決定先比較上輪廓值、下輪廓值孰大，將最大的輪廓值即記錄為最終輪廓值。再比較上輪廓值和下輪廓值相加是否大於目前計算的車牌高的 0.8 倍且上輪廓值和下輪廓值之差是否大於目前計算的車牌高的 0.5 倍；或者上輪廓值和下輪廓值均小於目前計算的車牌高的 0.3 倍，如果以上兩者條件成立任一個，則把上下輪廓值相加的結果來更新單行最終輪廓值，如果都不成立，則不做任何動作。還有如果最終輪廓值如大於車牌高，則設定最終輪廓值為車牌高。以上的動作，則從車牌最左邊開始逐行計算到最右邊，最後就可以得到最終輪廓值分佈圖。以圖 3.8 為例，其最終輪廓值分佈圖如下圖 3.12。同樣為了符合一般直覺觀察，圖 3.12 顯示是車牌高減去最終輪廓值。

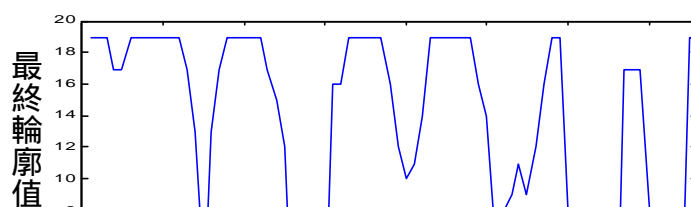


圖 3.12 最終輪廓值分佈圖

3.3 分割點的尋找

從以上幾節討論中，可以知道外圍輪廓分割法比較適合本系統做車牌字串分割。從上節得到上下輪廓值所組合出的最終輪廓值分佈圖，便可據此來做分割點的尋找。先利用相對最小動態臨界值找到分割點，再依分割點將整個車牌字元區塊區分成幾個字元群組，去除一些雜訊群組及消除不合理的假字元群組的動作，最後再去推敲最適個別車牌字元最適分割位置。其整個步驟依序如以下所介紹：

3.3.1 相對最小動態臨界值來尋找分割點

觀察圖 3.12 最終輪廓值分佈圖，再對照圖 3.8，人類視覺很容易看出圖 3.12 波谷位置就是分割點的位置。在最近的國外研究中，多來判定波谷訊號，但圖 3.12 的波谷分割點位置的最終輪廓值未必是零或者近似零，甚至每個波谷的最終輪廓值也不盡相同，這是因為受到光度不均、攝像模糊不清、字元連字的影響。所以採用固定臨界值來判定波谷訊號方法不見得適用本系統所攝得的影像，故在此是採用動態來調整相對最小臨界值的作法，以來判定輪廓值分佈中波谷的位置。

相對最小動態臨界值原理是認定單一車牌字元的左半邊及前一個車牌字元的右半邊，其間的最終輪廓值以分割處值最低，但是也許不知道其值有多低，至少可以此分割點為標準，將其兩邊字元拆開；同樣車牌字元約右半邊及後一個車牌字元的左半邊，其間的最終輪廓值以此分割處值最低，但是也許不知道其值有多低，至少可以以此為標準，將其兩邊字元拆開。如此不預設低點值(波谷值)應該為何，考量到單獨車牌字元和左右兩邊字元的變化，車牌字元左半部最終輪廓值應該都在其左處的分割處的最終輪廓值以上，車牌字元右半部最終輪廓值應該都在其右處的分割處的最終輪廓值以上，便可以彈性且動態來判定車牌字元的最初分割處。

基於以上原理，先求單行的相對最小臨界值，在目前計算行中往前及往後先各設定一個尋找範圍。在車牌高約是字高，字寬是字高的一半的假設下，其尋找範

圍取略大於字寬(也就是字高的一半),即往前取略大於字寬的一半,再往後取略大字寬的一半,在此之間找最小的最終輪廓值以做為相對最小臨界值。但是經過測試結果,現行範圍約取字高的三分之二較為合適,也就是往前後各取字寬的三分之二。如此重複以上計算,由車牌左而右逐行計算。

根據圖 3.12 最終輪廓值分佈圖,逐行由左往右計算,可得到如下圖 3.13 的相對最小臨界值線,實線即最終輪廓值,虛線即動態相對最小臨界值,可以看到最終輪廓值如在動態相對最小臨界值以上的區域,亦可以說藍線和紅線所包絡形成的封閉區域,其實就是一個個獨立的車牌字元。

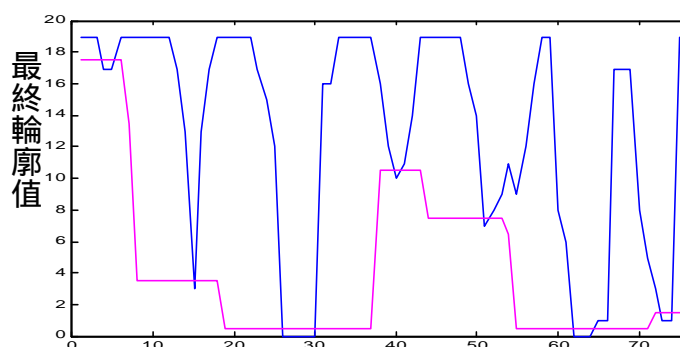


圖 3.13 最終輪廓值分佈圖及相對最小動態臨界值

再將每行最終輪廓值和相對最小動態臨界值相比,如果最終輪廓值大於相對最小臨界值,則認定是含有車牌字元的行,反之,則是可能分割點處。所以先統計含有車牌字元的行,再將其行號數列做群組化(群組化參數取 1),理想上圖中一個群組即是一個車牌字元,群組的開頭便是單獨車牌字元的左界,結尾便是其右界。

3.3.2 頭尾雜訊的去除

由圖 3.13,知道在一開始的群組及結尾的群組可能有非車牌字元的群組出現,

多是車體本身的延續，所以需以合理條件去除這些群組。求得的第一個群組大小如小於字寬的一半(即字高的四分之一)，且第一個群組的結尾距離第二個群組的開頭小於一定的數值，則去除第一個群組，否則則保留。同樣地，最後一個群組大小如小於字寬的一半，且最後一個群組的開頭距離最後第二個群組的結尾小於一定的數值，則去除最後一個群組，否則則保留。

以上作法的理由，限制群組的大小是基於如果第一個群組、最後一個群組是車牌字元，則其群組大小應大於一定的尺寸。但是考慮到如果是 1 或 l 的特殊情況，其本身群組小於字寬的一半，且又出現在第一個或最後一個群組的位置，所以利用 1 和 l 兩旁的都有相當一定程度幾乎為白點的區域來檢驗，如果第一個和最後一個群組相鄰群組的距離大於一定數值(即白點區域大於一定的大小)，則認為可能是 1 或 l，即使群組大小小於字寬一半，但仍其保留之。

3.3.3 字元寬度的重新估計

規格上，字寬應該為字高的一半，可是實際運算會有誤差的出現，因為分割程式後段一些參數有不少是基於字寬的數值來設定，所以有重新估計字寬的必要，非直接取字高的一半。重新估計的方法，即找群組中所有群組大小介於一定範圍的群組來重新估計，如大於字高的十分之四，且小於字高的十分之六，把假字元群組和一些連字及 1 和 l 的情況去除，以避免造成字寬的估計錯誤。後再平均這些符合條件的群組大小，便是重新估計後的字寬。

3.3.4 破折線(dash)的尋找

國內車牌字元中都有個破折號，本來破折號的存在造成程式判斷的困擾，因為破折號群組大小很接近 1 或 l 的群組大小，易誤判成是 1 或 l 的字元群組。但是在輔以判斷條件，反有助於分割點的尋找，增加分割成功的機率。因為在破折號之前都是有兩個車牌字元，在破折號之後則可能視車種而定，有三個或四個車牌

字元。如果二值化的影像能夠找到破折號，這樣就可以依據以上所提的破折號特性分前後兩部分尋找車牌字元，非直接找六個或者五個字母，可增加分割成功的機率。當然也有可能在二值化的影像能夠找不到破折號，找不到的原因可能是二值化後破折號根本就不見(如圖 3.12)或者和車牌字元連接在一起，所以找到破折號在此並非列為絕對必要條件，只是用來輔助分割條件。

其作法是先設定破折號的尋找範圍，破折號一般出現在車牌長度的某個範圍，約車牌左界五分之一的位置之後，二分之一的位置之前，且破折號群組大小多不會大於字寬的一半。另破折號是位於車牌縱向中間的位置，所以上、下輪廓值不會是零或者是字高，約介於字高的三分之一到三分之二。如果符合以上條件，便是認為此一群組所代表的物件是破折號。

3.3.5 雜訊的再去除

這裡的去除雜訊的方法，基本上和上面單元頭尾雜訊的去除方法相同，只是現在是檢查所有的群組，非只限於第一個或最後一個群組。先確認群組大小是否大於一定數值，如果大於則保留群組，如果小於則再根據 1 和 1 兩旁之間的一程度空白的區域，判定和前一個及後一個字群組的距離是否都大於一定數值，如果符合條件則保留下來，否則就是認為是雜訊去除之。

以上的方法和頭尾雜訊的去除的方法都相同，可是為什麼不在之前的步驟，便確認所有的群組？這是因為如果太早去確認雜訊群組去除步驟，可能會把破折號去除，造成分割率的降低。所以分成前後兩個步驟來進行。

3.3.6 分割點確立

分割點確立就是確認得到的群組是否為車牌字元，且單一車牌字元的左右界在哪裡，將其分割出來。依照是否有找到破折號，可分為兩大類的處理情形：

(一) 如果有找到破折號

把所有群組區分成兩個部分：

在破折號之前的群組：

- (1)只有一個群組：如果群組大小是車牌字寬的兩倍至兩點三倍，則將群組從中分開為兩個群組，這兩個群組即車牌字元。如果不符合以上的條件，則以破折號的左界為準，往前分割出兩個字寬區塊為兩個車牌字元。
- (2)有二個群組：如果兩個群組大小約符合字寬，則直接認定是破折號之前的兩個車牌字元。但是須考慮遇到是 1、l 的情況，所以如果群組大小小於字寬，再以則以前述同樣方法確認。如果以上條件都不符合，則將兩個群組合成一個群組，再將單一群組從中分割兩個群組為兩個車牌字元。
- (3)有三個群組以上：直接取最靠近破折號的兩個群組來當做兩個車牌字元。

在破折號之後的群組：

逐一確認群組是否為字元，如果群組大小略等於字寬，則認定是車牌字元，但是須考慮有可能遇到是 1、l 的情況，則同前述同樣方法確認，如果有約是車牌字寬的兩倍至兩點三倍，則約從中分開為兩個群組為兩個車牌字元，當找到四個車牌字元或者所有破折號之後的群組都確認完，即停止進行確認的動作。

(二)如果沒有找到破折號：

作法類似找到破折號的作法，程序上只是把前二個群組當做破折號之前的群組，其他群組則視做破折號之後的群組來處理。不過在這裡，要特別注意沒有找到破折號可能有二種情況，一種情況是破折號由於模糊不清在二值化後並不存在，此一情況尚未影響到後續的車牌字元辨識；第二情況則是和車牌字元相連在一起，以致和車牌字元形成同一物件，以致破折號未能夠找到，此一情形會影響到後續的車牌字元辨識。如圖 3.14，第三個車牌字元和破折號相連在一起，造成字元辨識困擾，不過依照當時車牌影像呈像，事實上此一情形也有可能發生在

第二個車牌字元，所以要設法消除和車牌字元相連的破折號，以避免造成車牌字元辨識的困擾。

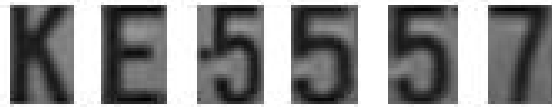


圖 3.14 第三個字元和破折號相連在一起

所以如果沒有找到破折號，就必須檢查車牌第二個字元右側、第三個字元左側是否有破折號相連在一起，將破折號去除。檢查的原理類似 3.3.4 節的尋找破折號的基本原理，利用破折號所在的位置其上輪廓值、下輪廓值約介於車牌高的 $1/3$ 到 $2/3$ 之間來進行逐行確認，第二個車牌字元則逐行由右向左檢驗，當檢驗行不符合以上條件，則停止檢驗，同時所在的行即此字元的新右界；第二個車牌字元則逐行由左向右檢驗，當檢驗行不符合以上條件，則停止檢驗，同時所在的行即此字元的新左界。經過以上處理，圖 3.14 則可以得到如圖 3.15 較好的結果。

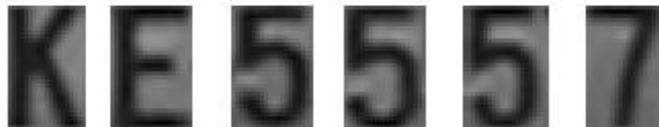


圖 3.15 消除破折號的車牌字元

3.4 字元的正規化

每個字元大小可能因車子本身距離照像機的遠近而有所不同，此時就需要做正規化的步驟，即影像縮放，把其縮放成固定的大小，才便於後續辨識處理。在這裡，考慮到類神經的計算量、辨識正確率，所以採取 30×20 來做為標準字元的大小。

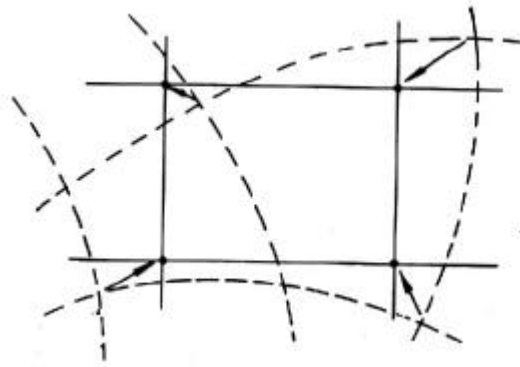
影像縮放是影

(1)座標轉換

(2)像素重新

單個或者

假設 (x, y) 為



幾何轉換

變公式及

階度， H 、 L 為原來

影像計算範圍的高、長；同樣地， (x', y') 為新影像的座標， $g(x', y')$ 為新的影像的灰階度， H' 、 L' 為新影像的高、長。

因為是採用線性比例來縮放原圖形，並未做任何的扭曲或者變形，故座標轉換式為：

$$(x', y') = (x \cdot \frac{L'}{L}, y \cdot \frac{H'}{H})$$

像素重新取樣，則一般使用有二種方法：

(一)最近整數法

最近整數點法是，其新影像 (x', y') 的色階是取相對位置最相近的舊影像 (x, y) 色階來得到。其數學式如下：

$$g(x', y') = f(\text{Int}(x \cdot \frac{L'}{L}), \text{Int}(y \cdot \frac{H'}{H})) \quad (3.4)$$

其幾何關係圖示如下，實線代表新座標，虛線代表舊座標：

圖 3.16 最近整數點法示意圖

(二) 雙線性內插法 (Bilinear Interpolation)

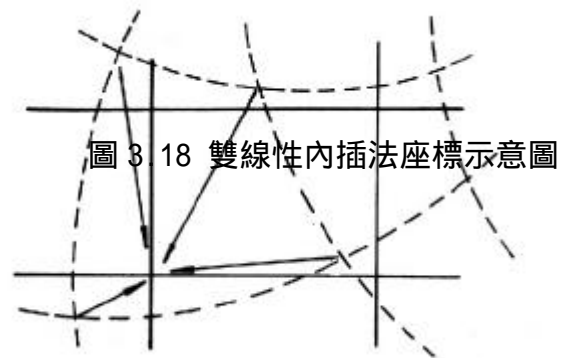
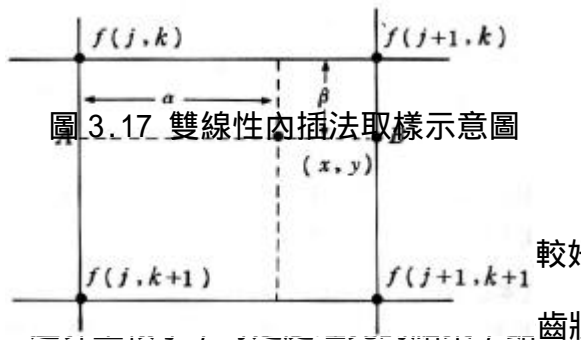
此法其新影像 (x', y') 色階是取相對位置最相近四個舊影像 (x, y) 點 色 階 總差而成。其數學式如下：

$$j = \text{Int}\left(x \cdot \frac{L'}{L}\right) \quad \mathbf{a} = x - j \quad (3.5)$$

$$k = \text{Int}\left(y \cdot \frac{H'}{H}\right) \quad \mathbf{b} = y - k \quad (3.6)$$

$$g(x', y') = (1 - \mathbf{a})(1 - \mathbf{b}) \cdot f(j, k) + \mathbf{a}(1 - \mathbf{b}) \cdot f(j + 1, k) \\ + \mathbf{b}(1 - \mathbf{a}) \cdot f(j, k + 1) + \mathbf{ab} \cdot f(j + 1, k + 1) \quad (3.7)$$

其幾何關係圖示如下，實線代表新座標，虛線代表舊座標：



到四個點來做線性插，運算量較大，單就數學式來看法的運算量近十倍之多，可是效果較好得多，正規化後物件的鉅齒化較不嚴重。在這裡，系統是採用雙線性內插法以提供較好品質的單一字元區塊以供類神經辨識，因為最後正規化的區域(至多 6 個 30x20 的字元區塊)並不算大，雖然是

運算量稍大，但實際佔系統的運算資源不會太多。

第四章 傾斜車牌矯正

一般正常情況下所攝得沒有傾斜的車牌影像，車牌的上下邊界線應該是平行於影像的上下邊界，可是常未必如此理想，常常發現攝得車牌影像會有傾斜的情形。發生車牌傾斜原因多來自車牌本身因螺絲鬆動或者曾受外力撞擊或者其他因素而傾斜，另外還有可能就是攝影環境不佳，其地面有凹陷或者凸起物，致使車輛經過，車體本身就已經呈現傾斜的情形，自然車牌影像也跟著傾斜。

當取得車牌影像發生傾斜時，對於車牌字串分割造成極大的困擾，如圖 4.1，BA 兩個字母約向右傾 15 度，在進行字串分割時，由於是採用縱向分割，分割線無論置於那一個位置，都無法得到很好的分割效果，造成分割後字元受到破壞，使得後續字元辨識率降低。

所以當車牌的傾斜角度太大，系統需做一調整程序，在進行車牌字串分割之前，將傾斜的車牌字串矯正回來，以利後續的車牌字串分割及字元辨識。計算傾斜角的基本依據則是從上章節所求得的外圍輪廓分佈圖中的多個字元群組的高點分佈來計算求得。



圖 4.1 傾斜的字串無法得到很好的分割點

4.1 傾斜角度計算

如要將傾斜車牌矯正，首先就必需要知道傾斜的角度多少，才能夠以座標轉換進行矯正，本節就是討論如果計算車牌傾斜的角度。

4.1.1 有關計算傾斜車牌影像的傾斜角相關研究

在張銘豪所著的“利用分割辨識方法之英文數字辨識系統”[3]中，提及利用獨立物件的中心點，計算出車牌傾斜角度。可是在本系統並非是以物件方式來尋找車牌，所以其提出以一個個車牌字元物件中心點位置來計算車牌的傾斜角度這種方法不是很適合本系統。若以車牌邊界線來做為傾斜角度的計算依據，困難度更高，其車牌的邊界線常常不明顯，不易尋找，且類似車牌的邊界線的物件在影像中則相當多，誤判的機會相當大。

4.1.2 計算傾斜車牌影像的傾斜角基本原理

在此，本系統是利用上輪廓值分佈圖來計算傾斜角度，除了正確率較高，另一方面上輪廓值本來就是系統在進行車牌字串分割所必須計算得到的資訊，如此可以節省不少不必要的運算。

圖 4.2 可以明顯看到此車牌影像有向左傾斜的情形，觀察其在第三章討論中決定車牌字串分割點之前所得到的上輪廓值分佈圖，可以發現字串的單獨字元群組中的最大上輪廓值單點(即圖 4.3 中所標示的星號)，在分佈圖中也會呈現向左傾斜的情形。再比較圖 3.7，車牌影像未有傾斜的情形，故在圖 3.8 的上輪廓值分佈圖中，字串的單獨字元群組中的最大上輪廓值單點數值均相同，在分佈圖中則呈現一水平直線的情形。所以基於以上的觀察，若車牌影像發生傾斜，在上輪廓值分佈圖中字串的字元群組最大上輪值單點，其分佈則會呈現斜直線的分布；反之，若車牌影像未發生傾斜，其分佈應該是一條沒有傾斜的水平直線。

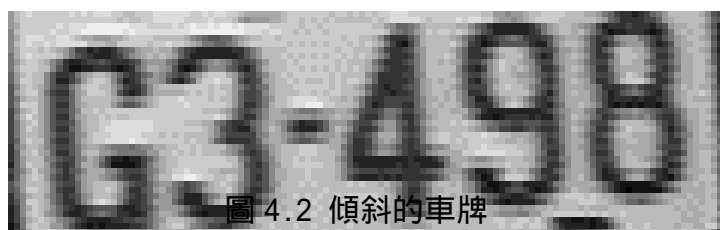


圖 4.2 傾斜的車牌

圖 4.3 傾斜車牌的上輪廓值分佈圖

4.1.3 傾斜車牌影像的傾斜角計算

基於以上的討論，若要計算傾斜車牌傾斜角度(即斜直線的斜率)，就必需先找出在上輪廓值分佈圖中每個字元群組所形成的最大上輪廓點位置。先設定一數值為 $range1$ 為車牌高的 $1/3$ (即車牌字元高的 $1/3$ ，亦即車牌字元寬的 $2/3$)，從車牌的最左邊開始逐行計算，以目前計算點往前的 $range1$ ，及其後的 $range1$ 為所計算的範圍，如果在此一範圍所計算出的最大上輪廓值的位置是目前計算點的位置(即目前計算範圍的最中間位置)，且其最大上輪廓值大於車牌高的 $1/2$ ，即累記錄其所對應的行號於 $\{d_1\}$ ，累記錄上輪廓數數值於 $\{d_2\}$ 。當從車牌左界逐行進行以上的計算到車牌右界，就可以得到其字元群組中最大上輪廓值點的分佈。

理想上，一個字元群組只要一個最大上輪廓值點就可以。可是觀察經過以上程序所得到的 $\{d_1\}$ 、 $\{d_2\}$ ，發現在單獨一個字元群組中常多有 2 到 5 點等值的最大上輪廓值點，但是只需要一點就可以，所以是取等值的最大上輪廓值諸點的中間點。其處理的程序，即將 $\{d_1\}$ 做群組化(群組化參數為 3)，再取群組的開頭及結尾值和的 $1/2$ 為 $\{d_3\}$ ，再取對應的上輪廓值於 $\{d_4\}$ 。 $\{d_3\}$ 、 $\{d_4\}$ 所對應的點即圖 4.3 中的紅點，但是可以發現圖 4.3 中星號點的分佈好像未必位於等值最大上輪廓值諸點的中間點，稍微偏向一方。這是因為單一字元群組具有偶數個最大上輪廓值點，且本身系統是離散資料系統，所以只好取稍微偏向一方。

再 $\{d_3\}$ 數列中取一次微分(即後一點的單位元素減去前一點的單位元素)則可產生一數列 $\{d_5\}$ (事實上即是 ΔX)；同樣 $\{d_4\}$ 中的值取一次微分即新產生一數列 $\{d_6\}$ (事實上即是 ΔY)。再從 $\{d_6\}$ 中取出不為零的部分為 $\{d_8\}$ ，再取出 $\{d_5\}$ 相對應的行號為 $\{d_7\}$ ，取 $\{d_6\}$ 中不為零的部分是因為如果相鄰兩點間的 ΔY 如果為零，則相鄰兩點間的連線只是平行 x 軸水平線，並沒有所謂的斜率可言，所以不予計算在內。

將 $\{d_8\}$ (即是 ΔY)的每個單位元素除於對應 $\{d_7\}$ 中的單位元素(事即是 ΔX)，可得到新的數列 $\{d_9\}$ (即是 $\Delta Y/\Delta X$)。再計算 $\{d_9\}$ 數列的平均值即為傾斜角 q 的 $\tan q$ 值，如果傾斜角度不大於 10° ，基於簡化運算量的考量，可以讓 $q \cong \tan q$ ，否則如要計算 $\tan q$ 的反函數將會相當的複雜。

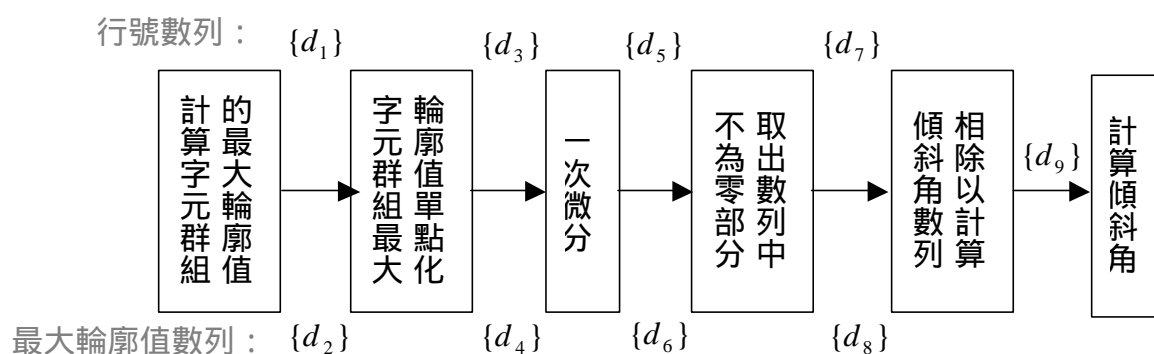


圖 4.2 的傾斜車牌經以上的計算後，可以算得傾斜角度為 3.5014 度。還有以上的討論是以上輪廓值分佈圖來做為計算的依據，如果採用下輪廓值分佈圖來做為計算的依據也可以達到計算出正確的車牌的傾斜角。

4.2 矯正傾斜車牌

根據 4.1 節的計算已知車牌的傾斜角度後，在本節中將以影像處理方式，完成

車牌傾斜影像矯正，以便於後續的車牌字串分割及字元辨識。系統所矯正的對象，是計算後車牌影像傾斜角度介於 1° 及 10° 之間(無論是左傾或是右傾)。如果計算後的車牌傾斜角度小於 1° 時，則沒有必要進行矯正；如果計算後的車牌傾斜角度大於 10° 時，除了 4.2 節的 $q \cong \tan q$ 無法使用來近似，一般正常情況下也不會有如此大的傾斜角，故此一情形並不進行矯正。

其矯正範圍不需對整張影像來進行矯正，只要對可能含有車牌的區域做矯正，同時為了能夠得到較好的效果，其矯正的範則”以預備細步掃描區”來進行傾斜矯正。以”以預備細步掃描區”的左上角為旋轉中心，令未進行傾斜矯正座標為 (X, Y) ，對應到傾斜矯正後座標 (X^1, Y^1) 的灰階值，座標轉換式：

$$\begin{bmatrix} X^1 \\ Y^1 \end{bmatrix} = \begin{bmatrix} \cos q & -\sin q \\ \sin q & \cos q \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

同樣因為 $0^\circ \leq q \leq 10^\circ$ ，所以 $\sin q \cong q$ ， $\cos q = \sqrt{1 - q^2}$ 。

但是計算完後的傾斜矯正後座標值 (X^1, Y^1) 多有小數點，所以需採用某種程度的取整數的方法(四捨五入或無條件進位)，又以未進行傾斜矯正座標所在的圖區域逐點計算，如圖 4.4。所以在傾斜矯正後座標所在圖區域，可能某些傾斜矯正後座標區域中的點會因為取整數點的緣故，沒有所謂未進行傾斜矯正座標區域的對應點，不知如何設定未對應到的傾斜矯正後座標區域點的灰階值。

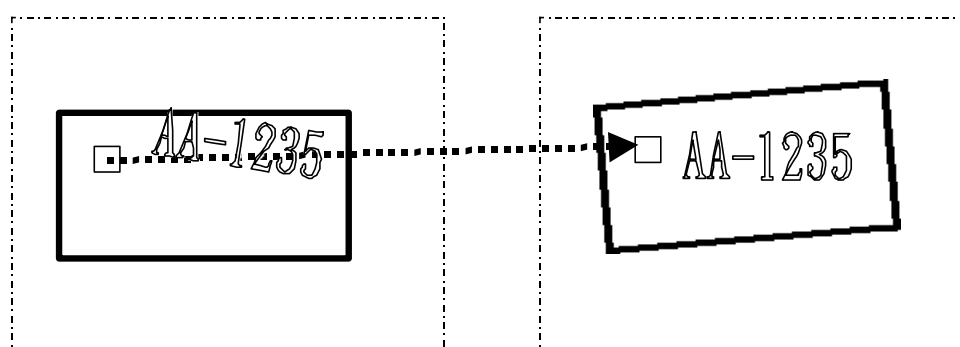


圖 4.4 以未進行傾斜矯正座標為計算基準的座標轉換

如果再去尋找這些未對應傾斜矯正後座標區域點，再以内插方式去計算該點的灰階值，將會較為費時。所以我們反過來計算，由傾斜矯正後座標區域逐點求出

所在的未進行傾斜矯正後座標區域所應該對應的位置，再把未進行傾斜矯正後座標區域所對應的灰階值傳給傾斜矯正後座標，如圖 4.5。如此，因為傾斜矯正後座標區域必然每一點都有其對應的灰階值，不會有以上的問題。

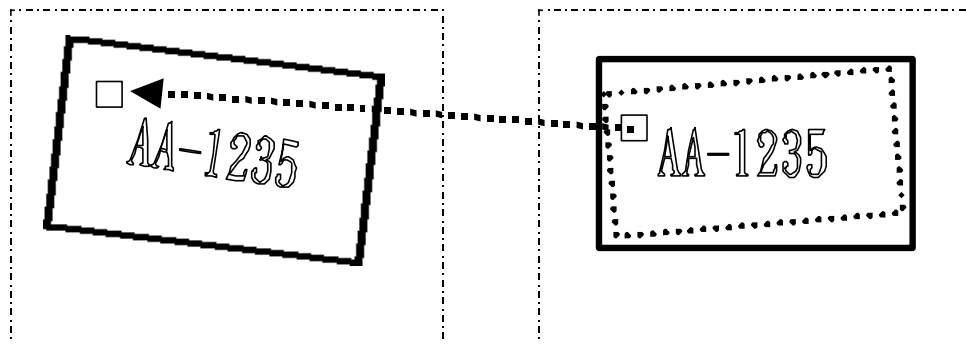


圖 4.5 以傾斜矯正座標為計算基準的座標轉換

其數學式可改為：

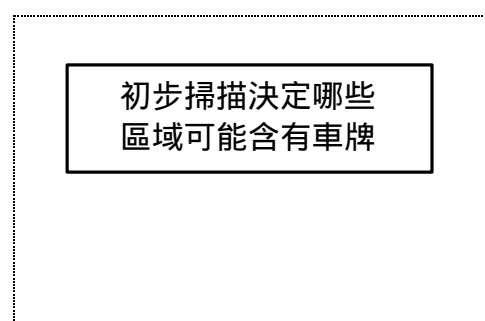
$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X^1 \\ Y^1 \end{bmatrix}$$

座標轉換後的圖形仍可能因為取整數的問題易產生不佳的鋸齒狀，如在原舊座標區域的一條斜直線，經過轉換後，可能變成鋸齒狀的平直線，非理想的平直線。

為了減輕此一問題，所以採用類似”消除交錯現象”的方法，即只對傾斜矯正後圖域的奇數列做以上的座標轉換，剩下傾斜矯正後圖域的偶數列點則以相鄰兩列做內插完成。

圖 4.6 傾斜車牌矯正後的結果

整個加入傾斜車牌矯正程式模組，其車牌字元粹取的整個系統流程圖如下：



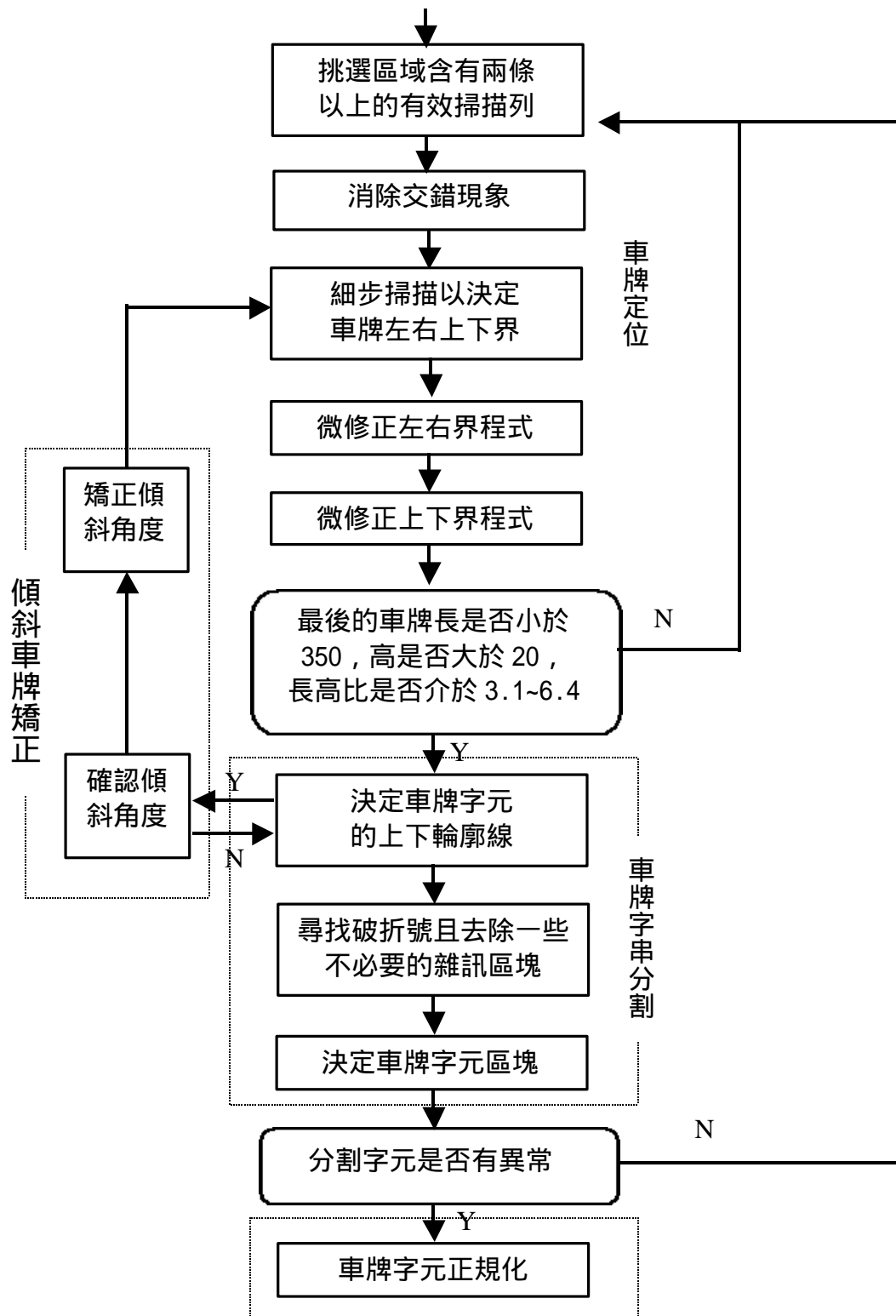


圖 4.7 車牌定位、車牌字串分割及字元正規化流程圖

第五章 研究結果

在這個章節中，將會介紹影像樣本如何取得，且以數張含有車牌影像為例，逐步介紹經本系統的一個個步驟處理完後的結果。最後再統計取得樣本中本系統能夠完成車牌定位、車牌字串分割步驟的成功率，並檢討其某些樣本失敗的成因。

5.1 樣本來源

5.1.1 國內外研究車牌辨識其樣本取得來源

國內外研究車牌辨識中，其系統架設多可分為兩種作法：一是純粹靜態取像，就是以數位相機取得靜止車輛狀態的照片，再以軟體撰寫進行車牌的辨識。另一作法，是以攝影機持續地拍攝車輛所經過的閘道，由於最後辨識處理是含有車牌的影像，所以設計上都有一感應式閘道機。當車牌通過閘道機的瞬間，即觸發呼叫影像擷取卡來擷取那時攝影機所拍攝到的車輛畫面，再將擷取到的畫面進行車牌辨識，故理論上，如果經過的車輛車牌制式規格都相同，其分別得到的車牌的大小應該是相同。

以上後者的處理難度通常比前者的處理難處較為高。還有某些系統可能為了避免攝影場地受到外在攝影光線的影響，而加裝光度感測器，藉由取得較多的資訊以利車牌辨識進行。

5.1.2 本車牌辨識系統硬體架構

本車牌辨識系統的架構主要分為三大部份(如圖 5.1)：(1)CCD 攝影機、(2)影像截取卡、(3)個人電腦。

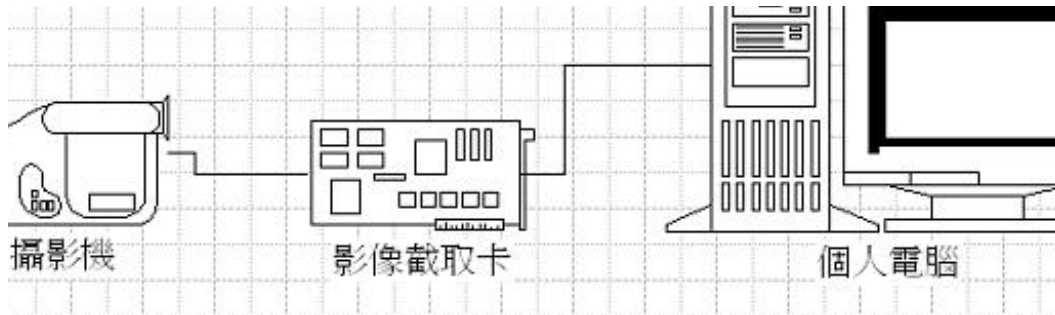


圖 5.1 車牌辨識系統架構圖

- (1) CCD 攝影機：主要用來拍攝動態車輛流，作為訓練樣本與實驗車牌的來源，而所用之 CCD 攝影機為相素 30 萬之 CCD 攝影機，具有自動對焦功能。
- (2) 影像截取卡：為 PCI 匯流排規格，插在個人電腦上，主要工作是將 CCD 攝影機的動態影像，以每秒三張的速度截取，並將截取到的影像存到個人電腦的硬碟。所用的卡為 Matrox 公司出品的 Meteor 影像截取卡，輸出規格有兩種，分別是 NTSC 與 PAL，整個車牌辨識系統所採用的影像規格為 NTSC。相關輸出規格詳列如表 5.1：

	NTSC	PAL
影像大小	640x480	768x576
顏色數	8,15,or 32-bit	8,15,or32-bit
取樣頻率	30FPS	25FPS
傳輸速度	> 35MB/s	> 42MB/s

表 5.1 Meteor 的影像輸出規格

- (3) 個人電腦：主要用來發展車牌辨識系統，包括車牌尋找、定位、辨識的演算法開發以及實驗結果的顯示與分析，電腦配備與發展環境如下：

- CPU 為 Intel Pentium 300。
- 記憶體為 128MB。

- 作業系統為 Windows NT 4.0 workstation。
- 程式的開發平台早期使用 Matlab 進行開發驗證，後有以 Borland C++ Builder 4.0 撰寫的版本。

5.1.3 本車牌辨識樣本取得來源

目前本系統取像機制所得到樣本是比較類似 5.1.1 節介紹中後者的作法，是以 CCD 攝影機持續地進行拍攝，再透過個人電腦上的影像擷取卡，將影像擷取下來，擷取轉換後影像的大小是 640x480 個像素，擷取的速率是每一秒三個畫面 (frame)。同樣地，由於是動態影像擷取為靜態影像，所以畫面中的清晰度沒有單純靜態影像拍攝來得好。

拍攝地點是選擇在校園入口處及市區道路交叉路口兩處。由於拍攝兩處皆是行人、車輛交通繁忙處，當擷取含有車牌的影像時，影像中可能也含有許多物件存在，如機車、行人、商店的招牌。和與一般不同系統取得樣本影像相比，通常單純只含有單一車輛的影像，沒有太多其他的物件存在，本系統處理影像的複雜度則較高。還有一般的系統多是架設在地下停車場，屬於較封閉環境，有些甚至會採用光度感測器來光度進行補償。本系統拍攝的地點是戶外，是屬於開放性空間，整個攝影的光度變化前後差異性較大，且未使用光度感測器來進行補償，所能得到的資訊較少。

另外本系統本預計不採用有觸發感測器機制來進行觸發擷取畫面，基本上得到畫面如果含有車牌的影像且車牌未被其他物件遮掩，如果是人眼能夠辨識車牌字元，希望透過本系統可以進行辨識。所以初期發展時辨識的樣本中，車牌的大小並不是固定的，車牌的大小變化較大(車牌高變化在 15 ~ 240 個像素)。不過這種無觸發感測器的車牌辨識系統難度較高，目前研究探討較少，大多車牌辨識系統是採用含有感測器，也就攝影機距離車牌是固定，所拍攝的車牌大小也是固定。為了使本系統能夠實用化，則選取含有車牌高度介於 30 ~ 60 個像素的影像(也就

是車牌大小介於 120x30 個像素 (240x60 像素)來進行車牌辨識,不過事實上有時甚至一些 80x20 大小的車牌也可辨識出來,可是成功率較低。故所辨識對象的難度是介於無觸發感測器車牌辨識系統及含有觸發感測器車牌辨識系統之間。

綜合以上段陳述,可以知道本系統辨識影像特徵相較於一般車牌辨識系統所取的影像所不同,其攝影環境較為複雜,光度較不穩定,車牌的大小變動則較大。

5.2 處理流程

為了便於了解第二章車牌定位、第三章車牌字串分割及字元一般化、第四章傾斜車牌矯正的每個流程處理,在本節中將以其中幾張實際進行辨識的影像來進行

解說，以便於了解以上數章節所陳述的處理過程及處理結果。

5.2.1 輸入影像

為含有車牌的 640x480 像素的圖像，其車號為 GC-0605。



圖 5.2 輸入影像

5.2.2 初步掃描

由第一列開始掃描，每隔 10 列掃描一次，即掃描列號為 1、11、21、31、41、51、61、71、81、91、101、111、121、131、141、151、161、171、181、191、201、211、221、231、241、251、261、271、281、291、301、311、321、331、341、351、361、371、381、391、401、411、421、431、441、451、461、471。經初步掃描後的有效列及其個別有效列中最大群組為：

有效列列號	1	21	71	81	101	111	141
最大群組開頭行號	44	2	2	9	4	20	9
最大群組結尾行號	158	158	46	26	38	52	59

有效列列號	181	211	221	231	241	251
-------	-----	-----	-----	-----	-----	-----

最大群組開頭行號	280	383	409	343	347	344
最大群組結尾行號	302	464	462	437	463	466

表 5.2 初步掃描後有效列列號及最大群組開頭、結尾行號

圖 5.3 中是以線條來代表有效列，其中有效列的最大群組開頭即白線的開頭，有效列的最大群組結尾即白線的結尾。



圖 5.3 初步掃描後的結果

再將這些有效列進行區塊群組化，可以得到四個區塊：

區塊編號	1	2	3	4
區塊上界	1	71	141	211
區塊所含的有效效個數	2	1	1	5
區塊下界	21	71	141	251
區塊左界	2	2	9	343
區塊右界	158	46	59	466

表 5.3 所有區塊的邊界

根據經驗定，車牌下方的干擾物較車牌一判定是否可為“預備細步掃描區域”的候選區，因為本系統辨識的對象是車牌高介於 30 60 之間，所以區塊如果含有二個有效列，則可以認定為“預備細步掃描

區域”，再進行後續的步驟，所以上表的區塊編號 2 是首先認定為可外擴為”預備細步掃描區域”，即下圖 5.3：

圖 5.4 預備細步掃描區域

5.2.3 消除交錯現象

對預備細步掃描區域做消除交錯現象處理，保留偶數列，奇數列則以相鄰兩列內插完成，其結果如下圖 5.5，比較上圖 5.4，可以明顯地看到交錯現象消除。

圖 5.5 消除交錯現象後的預備細步掃描區域

5.2.4 細步掃描

預備細步掃描區域中進行細步掃描可以得到有效列及有效列中的最大群組：

有效列列號	171	175	179	205	207	208	209
最大群組開頭行號	512	441	415	347	346	346	350
最大群組結尾行號	544	544	544	447	446	468	465
有效列列號	215	216	217	218	219	220	221

最大群組開頭行號	347	347	347	347	347	347	347
最大群組結尾行號	462	466	466	466	469	469	469
有效列列號	222	223	224	225	226	227	228
最大群組開頭行號	347	347	347	347	347	347	350
最大群組結尾行號	469	469	469	469	466	466	467
有效列列號	229	230	231	232	233	234	235
最大群組開頭行號	350	350	350	350	350	346	346
最大群組結尾行號	467	463	466	466	466	466	466
有效列列號	236	237	238	239	240	241	242
最大群組開頭行號	350	350	350	350	350	350	350
最大群組結尾行號	466	466	466	463	463	463	466
有效列列號	243	244	245	246	247	250	251
最大群組開頭行號	354	351	351	352	350	347	347
最大群組結尾行號	466	466	466	469	469	469	465

表 5.4 細步掃描後有效列列號及最大群組開頭、結尾行號

同樣地，圖 5.6 是以線條來代表有效列，其中有效列的最大群組開頭即線條的開頭，有效列的最大群組結尾即線條的結尾。



圖 5.6 細步掃描後的結果
再將這些有效列的列號進行群組化，可以得到二個列群組：

區塊編號	1	2
區塊上界	171	205
區塊所含的有效效個數	3	39

區塊下界	179	251
------	-----	-----

表 5.5 計算可能含有車牌的區塊

再選擇最大列群組認定為車牌所在列區域，故得到目前車牌上界即是 205，車牌下界即是 251。另外，左、右界的群組由群組大小排列如下：

左界群組編號	1	2	3	4
左界群組開頭	415	441	512	346
左界群組大小	1	1	1	39
左界群組結尾	415	441	512	354

右界群組編號	1	2	3
右界群組開頭	446	544	462
右界群組大小	2	3	37
右界群組結尾	447	544	469

表 5.6 車牌左、右界的判定

同樣地選擇左、右界群組中的最大群組來做為車牌左、右界的判定。車牌左界即為左界諸群組中的最大群組的開頭，為 346；車牌右界即為右界諸群組中的最大群組的結尾，為 469。所以經細步掃描後的車牌如下圖 5.7：

圖 5.7 細步掃描後決定的車牌區域

5.2.5 微調整邊界程式

圖 5.6 中除了車牌字元區域，尚包含非車牌字元區域，所以需將非車牌字元區域去除。在圖 5.6 中最主要去除即是車牌上方的螺絲孔及標示所屬管轄地”臺灣省”三個字，所以經過微調整上下界程式後，可以得到如圖 5.8：

圖 5.8 微調整上下界後的結果

5.2.6 量測傾斜角度

圖 5.8 中的車牌影像有傾斜的情形，會影響到後續的車牌字串分割及字元辨識，所以必須做傾斜的矯正。傾斜角度的量測便是根據上輪廓值分佈圖中個別字元群組的最大上輪廓值單點分佈來計算，如圖 5.9，曲線即代表上輪廓值，星號即是個別字元群組的最大上輪廓值，其星號分布約成為一條斜直線，斜直線的斜率便是代表車牌傾斜角度。

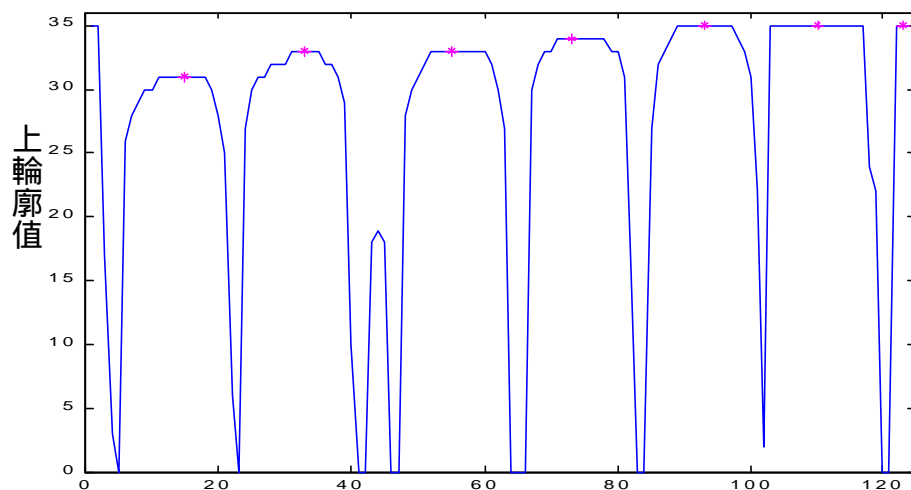


圖 5.9 用來量測傾斜角度的上輪廓值分佈圖

藉由星號點的分佈，可以計算出斜率約是 0.0375，換算成傾斜角度即是 2.1486 度。

5.2.7 傾斜車牌矯正

已經知道傾斜角為 2.1486 度，有必要進行傾斜車牌的矯正，依據 4.2 節所介紹的轉換處理步驟，可以得到如圖 5.10。

圖 5.10 傾斜車牌矯正後結果

5.2.8 外圍輪廓值分佈圖

接下來就對矯正後車牌進行車牌字串分割。其中字串分割的最主要依據便是外圍輪廓值分佈圖的決定，由圖 5.10 可求得外圍輪廓值分佈圖如圖 5.11。

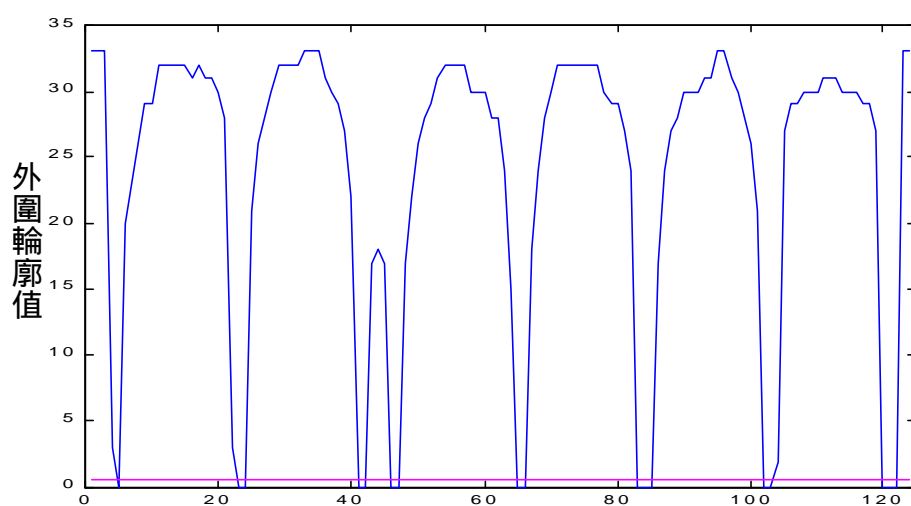


圖 5.11 外圍輪廓值分佈圖

5.2.9 分割點的尋找

由外圍輪廓值分佈圖再配合相對最小動態臨界值，可以找到字元群組為：

字元群組編號	1	2	3	4	5	6	7	8	9
字元群組開頭	1	6	25	43	48	67	86	104	123
字元群組大小	4	17	16	3	17	16	16	16	2
字元群組結尾	4	22	40	45	64	82	101	119	124

表 5.7 計算出的字元群組

接下來先找尋是否有破折號的存在，藉由上、下輪廓值及字元群組大小的確認，可以找到字元群組編號 4 即是破折號，再刪去一些較不可能為車牌字元的字元群組，如字元群組編號 1、9。再依據破折號的特性，破折號之前含有兩個車

牌字元，之後含有四個車牌字元及個別字元群組大小的合理性來判定剩下的群組是否為車牌字元。其中字元群組編號 2、3、5、6、7、8 事實上就是車牌字元 G、C、0、6、0、5。

5.2.10 字元正規化

得到分割好的車牌字元群組，便可以將車牌字元群組正規化成 30x20 個像素大小的字元區塊，如圖 5.12。

而整個系統最後輸出即 6 個 30x20 個像素大小的字元區塊。

圖 5.12 6 個 30x20 的字元區塊

5.3 實驗結果

這節將討論本程式處理樣本的結果，即處理完成成功率。

樣本取得來源正如 5.1 節所介紹，攝影地點是在臺大校園側門入口及市區交叉

路口，攝影方法採用是持續拍攝車輛所經之處，樣本是從攝得的動態影像再以影像擷取卡每一秒擷取三張影像。其中取含有車牌(車輛)高 30 60 之間的影像做為測試樣本，總共有 1450 張樣本，其中在校園側門口共取得 861 張，在市區交叉路口取得 589 張。

在這裡，處理成功的定義是在含有車牌的影像經過程式處理完後，能夠依據車種而區分成 5 個(計程車)或 6 個(自用車)30x20 大小的車牌字元區塊，諸字元區塊內便是即依序個別的車牌字元(如圖 5.11)。字元在字元區塊內位置大致上是置中，且水平傾斜角不致於太誇張(約小於 2 度以內)，字形大體未沒有破壞，人眼尚可以以較嚴格的標準辨認其字元。即就是車牌定位成功，車牌字串分割成功，且能夠對傾斜車牌影像矯正。

取得的 1450 張影像中，符合處理成功標準總共有 1290 張，未符合處理成功標準有 160 張，其處理成功率為 88.97%。

以取得樣本的车牌的高不同，其處理的成功率如表 5.8。

車牌高	失敗張數	成功張數	總張數	成功率
30	8	52	60	86.67%
31	5	57	62	91.94%
32	9	79	88	89.77%
33	8	64	72	88.89%

34	13	63	76	82.89%
35	9	66	75	88.00%
36	10	71	81	87.65%
37	10	73	83	87.95%
38	9	77	86	89.53%
39	8	58	66	87.88%
40	7	57	64	89.06%
41	5	66	71	92.96%
42	7	50	57	87.82%
43	5	62	67	92.53%
44	5	48	53	90.57%
45	4	45	49	91.84%
46	5	46	51	90.20%
47	3	40	43	93.02%
48	9	36	45	80.00%
49	4	29	33	87.88%
50	0	24	24	100.00%
51	2	25	27	92.59%
52	4	25	29	86.20%
53	0	17	17	100%
54	3	13	16	81.25%
55	2	10	12	83.33%
56	3	10	13	76.92%
57	2	13	15	86.67%
58	0	7	7	100.00%
59	1	3	4	75.00%
60	0	4	4	100%
總計	160	1290	1450	88.97%

表 5.8 以車牌高為區分的成功率

以取得樣本的车種觀點，其處理成功率：

車種	失敗張數	成功張數	總張數	成功率
計程車	30	220	250	88.00%
自用車	130	1070	1200	89.17%
總計	160	1290	1450	88.97%

表 5.9 以車種為區分的成功率

以取得樣本的车牌加框與否觀點，其觀察處理成功率：

車種	失敗張數	成功張數	總張數	成功率
加框	15	79	94	84.04%
未加框	145	1211	1356	89.30%
總計	160	1290	1450	88.97%

表 5.10 以加框與否為區分的成功率

以取得樣本的車牌的傾斜程度觀點，其觀察處理成功率：

傾斜程度	失敗張數	成功張數	總張數	成功率
傾斜	77	432	509	84.87%
未傾斜	83	858	941	91.18%
總計	160	1290	1450	88.97%

表 5.11 以傾斜與否為區分的成功率

以取得樣本的車牌的地點觀點，其觀察處理成功率：

地點	失敗張數	成功張數	總張數	成功率
校園側門口	77	784	861	91.06%
市區交叉口	83	506	589	85.90%
總計	160	1290	1450	88.97%

表 5.12 以拍攝地點為區分的成功率

以上是以 1450 張樣本來討論各種攝影環境及車輛特性來觀察成功率，接下來是以處理成功的樣本為樣本空間，來討論程式中一些處理的結果。

以取得處理成功樣本為主，統計是否有找到破折號：

處理結果	張數	佔處理成功比率
有找到破折號	1208	93.64%
未找到破折號	82	6.36%
總計	1290	100.00%

表 5.13 找到破折號佔處理成功樣本比率

以取得處理成功樣本為主，統計在第幾次細部掃描中找到車牌：

第幾次細步掃描找到車牌	張數	佔處理成功比率
第一次	1201	93.10%
第二次	81	6.28%
第三次	8	0.62%
總計	1290	100.00%

表 5.14 第幾次細步掃描找到車牌佔處理成功樣本比率

完成整個讀取影像、車牌定位、車牌字串分割、車牌字元正規化、傾斜車牌矯正所需的時間，在 5.1.2 節所介紹的硬體平台下，使用 Matlab 處理完以上的程序，約需要 15~20 秒之間。現改以 Borland C 撰寫，同樣在 5.1.2 節所介紹的硬體平台下，完成以上的程序，平均約需要 0.20 秒。不過事實上處理時間會因為車牌大小、環境的複雜度、車牌是否有傾斜有關，如需要做傾斜車牌矯正通常

會比無需做傾斜車牌矯正所需時間又約多出 0.04 秒(仍要視車牌的大小而定)。
所以如果條件愈單純，處理的時間便可以愈短。

5.4 實驗結果討論

由 5.3 節的實驗結果表 5.2，可以知道車牌高介於 30 和 50 之間，處理成功率差不多約 88%到 92%之間，另在車牌高介於 50 到 60 時，則因樣本比較少，所以成功率的變化比較大，介於 75%到 100%之間。整體看起來，車牌高的大小似乎和成功率沒有較明顯的趨勢關係，但是事實車牌高 30 ~ 60 是系統處理的最適範

圖。如果不考慮系統的實用性，明顯地，車牌高約 20 附近的處理成功率應比車牌高約 30 附近的處理成功率小得多。

表 5.9 顯示車種不同對於處理成功率似乎沒有太大的關係。因為就車種的車牌特徵而言，計程車的車牌是 5 個字元，其字元位置的分佈較自用車 6 個字元為分散(注意計程車和自用車的車牌大小是一樣)，所以就車牌字串分割而言，計程車這部分的成功率比較高；但是在灰階攝影的情況下，計程車車牌字元顏色明顯比自用車車牌字元顏色為淺，所以定位部分，自用車這部分的成功率比較高，由於兩者以上的因素相互影響，所以計程車和自用車的成功率差不多。

由表 5.10 明顯知道，車牌不加框明顯地比加框成功率較大，車牌加上框，車牌字元往往看起來和框相連在一起，難以分離，降低車牌字串的成功分割率。

另表 5.11 顯示傾斜程度也明顯影響處理成功率。如傾斜的車牌有經過矯正，其車牌字串分割的成功率的分割率也能提高；傾斜的車牌未經過矯正，會影響到車牌字串分割的成功率，進而影響到整個處理過程的成功率。不過事實上，傾斜的車牌未矯正，但是仍然車牌字串仍可以分割出(在傾斜角度未太大下，如 3 度)，對於後端的車牌字元辨識率也不會有太大的影響，這是因為在車牌字元辨識方法是採用智慧型類神經圖樣比對法，對於車牌字元位於字元區塊中有位移、有傾斜的情形則有一定程度的強健穩定性，所以對於傾斜車牌字元辨識率仍極佳。

還有從表 5.12 來看，攝影地點明顯影響處理成功率，在校園側門口攝得而擷取到含有車牌的影像，由於校園側門口有管制，車道有限，車輛進入校園時多會減速，也較少有其他物件的干擾，所以處理成功率明顯較高。另在市區叉路口所得的含有車牌的影像，雖然道路有分隔線，但車輛未按照車道行走，多有機車、行人、其他車輛在干擾目前出現在畫面的車輛車牌影像，光線也較不穩定，所以處理成功率明顯較低。

表 5.13 中，有找到破折號佔處理成功樣本的 93.64%。在處理成功的樣本未找到破折號，其影響的因素很多，影響較大的因素是攝得影像中，破折號和車牌字

元相連，形成同一個物件，和之前假設破折號為獨立物件來尋找的假設有違，對於此一類找到破折號的的比率則較低。但事實上破折號的尋找對於處理成功率有很大的影響，因為根據破折號的特性(在破折號之前含有 2 個車牌字元，在其後含有 3 或 4 個車牌字元)可以大大提昇車牌字元的分割率，以處理過程中含有尋找破折號程式和處理過程中未含有尋找破折號程式，其整體處理成功率約可以提昇 10%。

表 5.14 的意義是在如果愈能夠以少次的次數找到車牌，其處理的時間也就愈少，處理的速度也就愈快。由表 5.7 可以看到，約 99.3%的車牌在處理成功的樣本中，至多在第二次細步掃描就可以找到車牌的位置。

就整體處理失敗的樣本，區分其失敗的原因以處理流程中以最先影響處理成功的原因來討論，分類如下表 5.15。

由表 5.15 可以去試著去統計部分處理成功率，即車牌定位成功率與車牌字串分割成功率(而車牌字串分割成功率包含車牌字串割成功及傾斜車牌矯正成功，一起合計是因為這兩部分皆同樣以外圍輪廓法做為演算的依據)。在 1450 張的樣本中，共有 1335 張車牌定位成功，車牌定位成功率為 92.07%；在 1335 張車牌定位成功的樣本中，共有 1290 張車牌字串分割成功，車牌分割成功率為 96.63%。故整體完全處理成功率為 88.97%($92.07\% \times 96.63\% = 88.97\%$)。

其中以「細步掃描有誤差，即定位後左右上下界差太多」的原因影響處理失敗率似乎比較大，探究真正此項導致失敗的原因，則相當複雜，如車牌附近也有高反差的物件、車牌有加框、車牌字元顏色過淡等所影響著。

	處理失敗原因	張數	L	B	T	F	/
1	初步掃描對於車牌未掃到二列以上的有效列	5	1		3		
2	在預備細步掃描區中做細步掃描，其中雖含有車牌，但是定位到其他物件	2					

	細步掃描有誤差，即定位後左右上下界差太多(雖含有車牌,但是含有非車牌的區域仍太多)	47	2	3	4	13	4
3	微處理左右界程式過度修正	12	1				
	微處理左右界程式未修正	12	1		6		6
4	微處理上下界程式過度修正	14	4		4		1
	微處理上下界程式未修正	19	1		4	1	10
5	長高比臨界條件確認有誤	9				9	1
6	假車牌物件	6			2		1
7	決定上下輪廓線有誤(二值化未處理好)	8	1			1	
8	決定分割點有誤	26			3	3	11

表 5.15 處理失敗樣本的原因

(其中 L、B、T、F、/ 是表示處理失敗車牌樣本的特徵，L 是表示車牌字元顏色過淡，B 是表示車牌有污損，T 是表示計程車，F 是車牌有加框，/ 表示車牌明顯傾斜)

第六章 結論及未來工作

本論文在智慧型車牌辨識系統中討論為車牌定位、車牌字串分割、傾斜車牌矯正及車牌字元正規化。先藉由車牌定位找到車牌的位置，同時以內插消除交錯現象，透過微調整邊界程式來微調整車牌邊界位置，若車牌有傾斜的情形，則再進行傾斜車牌的矯正(扶正)，後再由外圍輪廓法分割法來決定車牌字串的分割點，再將分割好的車牌字元正規化成 30x20 個像素的字元方塊，以便後續的字元辨識。

樣本影像取得是採用 CCD 攝影機持續拍攝校園側門口及市區叉路口，將動態影像以每秒三個畫面進行擷取，再取含有車牌的影像為測試樣本，同時為了實用化起見，則取車牌高介於 30 60 之間，且車輛的車牌未被其他物件所遮掩，尚可為人眼所辨識的車牌的影像進行車牌字元粹取處理。總計共用 1450 張測試樣本，整個車牌字元粹取能夠完全處理完成，即車牌定位、車牌傾斜角、車牌字串分割、車牌字元正規化都能達到要求者，共有 1290 張，處理成功率為 88.97%。

6.1 本論文和以往研究所不同處

本篇論文和以往研究其他車牌辨識論文不同處，可以從測試樣本及處理方法及效率兩方面來看。

6.1.1 測試樣本取得

測試樣本和以往研究所不同處：

(1) 攝影機制

不是以數位相機在車體停止下直接靜態攝得，是將 CCD 攝影機所攝得的動態影像進行擷取。且非採取感應式觸發擷取，即得到的車牌大小都一樣，而是較偏向

無感應式擷取，其得到的車牌大小都不一樣，但為實用化，則限制車牌高介於 30 60 之間。

(2) 攝影環境

非像一般為單一車道，得到畫面多只有目前攝影單一車輛的物件存在，且光線多半可控制在一定範圍(甚到有的系統會以光度感測器進行補償)。而是校園側門口及市區叉路口，除了目前攝影單一車輛的物件存在，尚包含其他如車輛、行人、商店招牌等較為複雜物件，為戶外環境，光度變化較大而未以光度感測器輔助。

(3) 測試樣本特徵

本系統的測試樣本有 509 張有較大的傾斜度。另外在車牌字串分割部分，本系統多會遇到連字的問題，連字問題產生，除了我國車牌字元設計本來就較緊密外，另外原因就是影像在硬體擷取中其擷取效果並不好，導致本系統必須另外以外圍輪廓法來解決分割的問題，非以一般簡易的垂直投影量分割法便可以解決車牌字串分割的問題。

6.1.2 本論文處理方法及效率和其他車牌辨識研究相比較

本論文在處理方法及效率和其他車牌辨識研究相比較：

(1) 選取較合適的基本運算遮罩：

在車牌定位步驟是採用「類空間頻域分析法」，在使用偵測高反差變化遮罩，為了簡化計算步驟，降低計算量，採用較合適的 $[-1 \ 0 \ 0 \ +1]$ 遮罩，不同於以往使用 Roberts、Prewitt、Sobel 或者 $[-1 \ +1]$ 的遮罩。另外在單列中高反差分佈點確認是否為有效列，是採用動態微分臨界值來認定，相較一般採用固定微分臨界值來認定，對於環境光度變化可以有較高的穩健性。

(2) 兩段式定位，降低運算量

還在車牌定位過程較為特殊是採用初步掃描及細步掃描來達成，先利用初步掃

描(每隔 10 列)在畫面中找出車牌可能位於的區域，再根據經驗法則，由下往上找，逐一利用細步掃描(每隔 1 列)來確認區塊內是否為車牌及定出車牌的邊界，如此計算簡易且可信度仍高的處理，相較於以往耗時而複雜的全面式尋找車牌方法，搜尋範圍只要原圖像大小的近六分之一，所費時間自然就少，處理效率可以較快。

(3) 車牌字串分割

在車牌字串分割方面，本系統是採用外圍輪廓值分割法，可避免一般使用垂直投影量分割法在連字處發生問題，還有破折號的尋找，使得車牌字串分割的成功率大為提昇。

(4) 傾斜車牌矯正

另外透過上輪廓值分佈圖找出每個字元所形成的群組中最大上輪廓值點，進而計算出傾斜角度，則可達到矯正傾斜的車牌。

6.2 未來工作

由第五章的研究結果可以看到，處理成功率是 88.97%，仍有 160 張未處理成功，似乎以車牌定位的步驟處理成功率較低所致。探究其處理失敗的原因，主要是仍有部分傾斜車牌未進行矯正、在車牌附近加裝裝飾以致干擾的物件不少、車牌本身有污損(如螺絲孔髒污擴散)，太過暗的光線以致難以達成車牌定位等因素影響著。所以未來的工作及發展仍可以朝向完全解決此方面的問題發展，另外本系統為了實用化，一開始限制車牌高取 30 60 之間，事實上在車牌高大於 15 以上就可以為人眼所辨識，但是本系統在車牌高 20 30 的處理成功率仍較低，所以仍可以提高處理範圍努力。

參考文獻：

- [1] J. Barroso, J. Bulas-Cruz, E.L. Dagless, and A. Rafael, "Number Plate Reading Using Computer Vision ", Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE international Symposium on , 1997 , Page(s): 761 –766 vol.3
- [2] C. Busch, R. Dömer, C. Freytag, and H. Ziegler, " Feature Based Recognition of Traffic Video Streams for Online Route Tracing ", Vehicular Technology Conference, 1998. VTC 98. 48th IEEE Volume: 3 , 1998 , Page(s): 1790 -1794 vol.3
- [3] J. Carrabina, F. Lisa, C. Perez-Vicente, E. Valderrama, and N.A Vellana, " Two-bit Weights Are Enough To Solve Vehicle License Recognition Problem " Neural Networks, 1993., IEEE International Conference on , 1993 , Page(s): 1242 -1246 vol.3
- [4] K. Chamnongthai and T. Sirithinaphong, "Extraction of Car License Plate Using Motor Vehicle Regulation and Character Pattern Recognition" , Circuits and Systems, 1998. IEEE APCCAS 1998. The 1998 IEEE Asia-Pacific Conference on , 1998 , Page(s): 559 -562
- [5] B.T. Chun, Y.S. Soh, and H.S. Yoon, " Design of Real Time Vehicle Identification System ", IEEE International Conference on Volume: 3 , 1994 , Page(s): 2147 -2152 vol.3
- [6] E.D. Di Claudio, G. Lucarelli, G. Orlandi, and R. Paris, " Car Plate Recognition by Neural Networks and Image Processing ", Proceedings of the 1998 IEEE International Symposium on Volume: 3 , 1998 , Page(s): 195 -198 vol.3
- [7] Y. Fujikawa, K. Fujimoto, M. Horino, and K. Kanayama, " Development of Vehicle License Number Recognition System Using Real Time Image Processing and Its

- Application to Travel Time Measurement ”, Vehicular Technology Conference, 1991. Gateway to the Future Technology in Motion., 41st IEEE , 1991 , Page(s): 798 –804。
- [8] I. Fujita, K. Miyamoto, K. Nagano, M. Tamagawa, and M. Yamamoto, ” Vehicle License Plate Recognition by Image Analysis ”, Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on , 1991,Page(s): 1734 -1738 vol.3
- [9] R.C. Gonzalez and R.E. Woods, ” Digital Image Processing ”, Addison-Wesley,1992。
- [10] H.A. Hegt, R.J Haye, and N.A. Khan,” A High Performance License Plate Recognition System ”, Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on Volume: 5 , 1998 , Page(s): 4357 -4362 vol.5
- [11]L. Kreft and M.Raus “ Reading car license plates by the use of artificial neural networks “ , Circuits and Systems, 1995.,Proceedings., Proceedings of the 38th Midwest Symposium on Volume: 1 , 1996 , Page(s): 538 -541 vol.1。
- [12] Y. Lu ,” On the Segmentation of Touching Characters “ , Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on , 1993 , Page(s): 440 –443。
- [13]王敦厚, ” 中英文混合印刷文件之文字切割技術之研究 ” , 國立成功大學 , 資訊工程研究所碩士論文 , 1995。
- [14]林欣平 , “ 車牌字元粹取 ” , 國立交通大學 , 電機與控制工程研究所碩士論文 , 1999。
- [15]周俊男 , “ 車輛牌照影像辨識系統 “ , 國立中山大學 , 資訊工程研究所碩士論文 , 1995。
- [16]陳兩嘉, ” 使用三方向非線性切割演算法於字元分離 “ , 私立中原大學 , 電子工程研究所碩士論文 , 1996。

- [17]陳建志，”中英混合印刷文件之字元切割與分類“，國立臺灣大學，電機工程學研究所碩士論文，1997。
- [18]連國珍，”數位影像處理“，儒林圖書有限公司，1996年8月初版四刷。
- [19]張銘豪，”利用分割辨識方法之英文數字辨識系統“，國立中山大學，資訊工程研究所碩士論文，1996。
- [20]楊文宗，”車輛牌照號碼自動辨識系統“，私立中原大學，電子工程研究所碩士論文，1996。
- [21]溫福助，”類神經網路圖樣比對法於車牌字元辨識之研究“，國立臺灣大學，電機工程學研究所碩士論文，2000。
- [22]賴幼仙，”任意角度車牌之辨識“，國立交通大學，資訊工程研究所碩士論文，1992。

附錄

茲列出本系統處理 1450 張樣本中的 20 張，以供參考：



Image01



Image02



Image03



Image04



Image05



Image06



Image07



Image08



Image09



Image10



Image11



Image12



Image13



Image14



Image15



Image16



Image17



Image18



Image19



Image20