第三章 車牌定位方法

車牌定位是車牌辨識系統裡面最關鍵的步驟，也因此一套車牌辨識系統的好壞，可以拿車牌定位作為評斷的依據。既然車牌定位如此的重要，本文首重於研究車牌定位的方法，為了達到動態且即時的車牌偵測，所能容忍之運算時間極少，所以提出之系統特色為運算複雜度低，處理節省時間且仍有保有一定效果。

本文自動車牌偵測的處理步驟主要由「移動物偵測模組」與「車牌定位模組」兩大部分組成，此外，在系統的前端加入車輛進入與否的判斷，整體系統處理流程圖如圖 3.1 所示。

首先，輸入影像序列資料，我們在影像上下適當的位置設定觸發線(Trigger Line)，判斷是否有移動物件進入畫面，如果有，則啟動系統開始處理運算，將影像序列資料送入後續「移動物偵測模組」的步驟。

在移動物偵測模組裡，本文使用跳耀式背景相減法，擷取出移動物件，考慮車牌出現之幾何位置，取出更小之運算區塊，並將此區塊丟入車牌定位模組運算。

最後，在車牌定位模組裡，我們考慮車牌裡的字元部份會有灰階反覆的強烈變異，並利用此特徵尋找出精確之車牌區塊後，再透過 Sobel 垂直邊緣偵測取得梯度圖，並將此梯度圖做佈線找出車牌候選區，完成車牌定位。3.1 節說明系統啟動機制，3.2 節為移動物偵測模組，3.3 節為車牌定位模組，以下分別詳細介紹之。

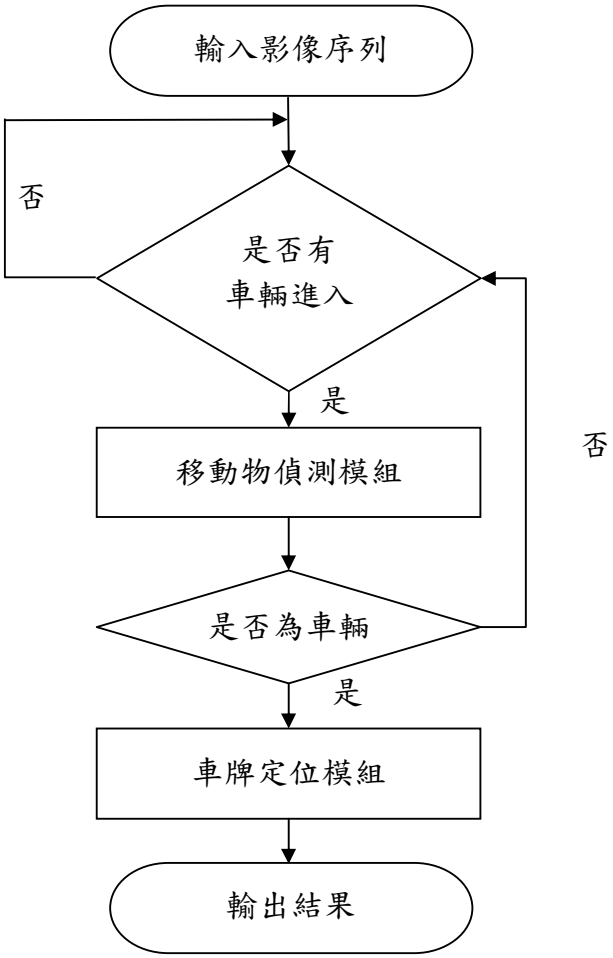


圖3.1　本文系統處理架構圖

3.1 系統啟動機制

為了不讓系統一直呈現在工作滿載的狀況下，即謂不必時時刻刻處理一大堆不相干的資料，本文設計了系統啟動機制。在影像的上下邊界距離四分之一影像高的位置，各定義了一塊區域當作觸發線，這塊區域的寬度與影像大小同寬，高度有 3 個像素點，如圖 3.2 所示。其計算概念很簡單，當連續畫面出現了背景以外的移動物體時，在畫面裡面會出現一定數量的差異像素點。因此，考慮影像序列做觸發線相減的動作，並採用二值化的方式來比較差異程度，如果大於門檻值T_1，則設為前景(白色)；最後，統

計位於觸發線裡面出現的白色像素點的次數*Count*；倘若*Count*的數目大於門檻值T_2，則啟動系統運作以下流程。

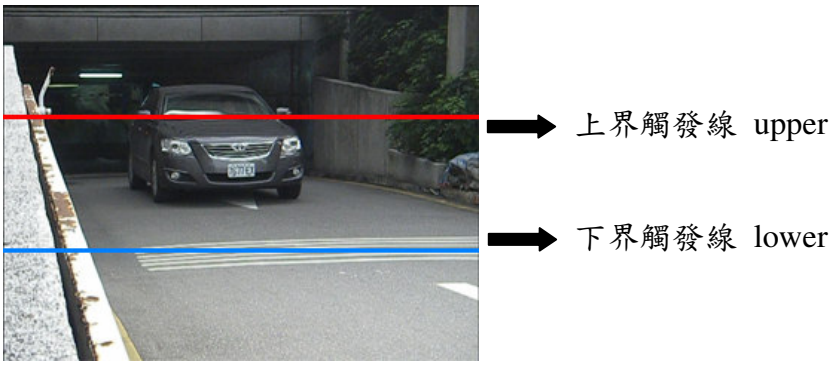


圖3.2　觸發線高度示意圖

```
int  Count1 = 0 ;
Boolean  Judge1 = = False;
for   y = upper - 1   to   upper + 1
{
   for   x = 0   to   Width
   {
      if (| F1(x, y) − F2(x, y) | > T1 )
      Count1 + +;
   }
}
if ( Count1  >  T2  )
Judge1  = = True;
```

(*a*) 上界觸發線演算法

```
int  Count2 = 0 ;
Boolean  Judge2 = = False;
for   y = lower – 1   to   lower + 1
{
    for   x = 0   to   Width
    {
       if ( | F1(x, y) − F2(x, y) | > T1 )
       Count2 + +;
     }
 }
  if ( Count2  >  T2  )
Judge2  = = True;
```

(*b*) 下界觸發線演算法

圖3.3　觸發線區域的演算法則

圖 3.3(a) 圖 3.3(b)與為上下界觸發線的演算法，upper 為四分之一畫面高，lower 為四分之三畫面高，$F_1(x,y)$與$F_2(x,y)$分別為兩張連續影像之座標(x,y)的灰階值，T_1與T_2分別為事先設定之門檻值，而上述之情況只要任一成立就會啟動系統開始進入移動物偵測模組，即$Judge_1 \cup Judge_2$的結果為「真」時，車牌偵測的流程就開始動作。

3.2 移動物偵測模組

移動物偵測模組又可以細分為數個步驟，依序為「背景相減」、「簡單類連通成分分析」、「考慮車牌幾何位置」三項；其處理的流程加上系統啟動機制如圖 3.4 所示。

這個部分的重點為利用影像序列之編號奇數的圖框(*Frame*)，兩兩圖框做所謂「跳躍式」的相減，並找出疑似移動的物件；至於選用奇數圖框相減的原因，是怕鄰近兩張圖框差異程度太小，不足以擷取出移動物件；此外，編號偶數的移動物位置，一定會出現於兩側編號奇數圖框的差異中間。接著，考慮車牌出現的幾何位置，再把矩形對角座標算出，並且於編號偶數的圖框中，做標記處理的後續運算。這麼做的好處在於花些許時間、並用很簡單的運算法擷取出移動物件，換得更小更具有信賴度的運算面積做後續處理，而不必像靜態的影像系統，反覆且盲目地對整個輸入的大圖框處理並運算，消磨極大量的時間。

相減畫面間隔數其實也可以依照不一樣的使用狀況做調整，在某些特殊情況即使差一間隔畫面差異還太小的話，也可改為跳躍間隔數差二或三做相減，所以，編號奇數相減並非唯一。

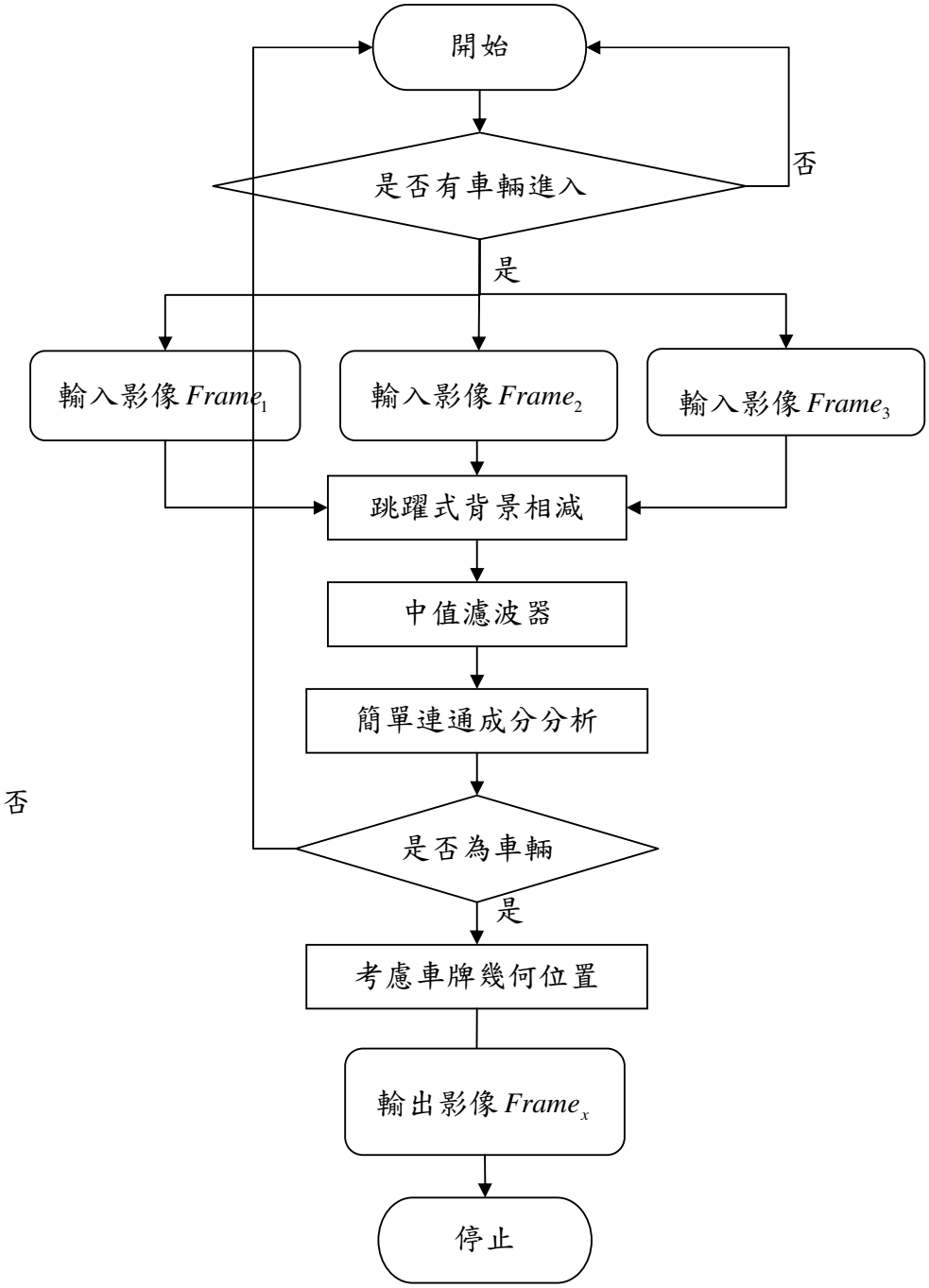
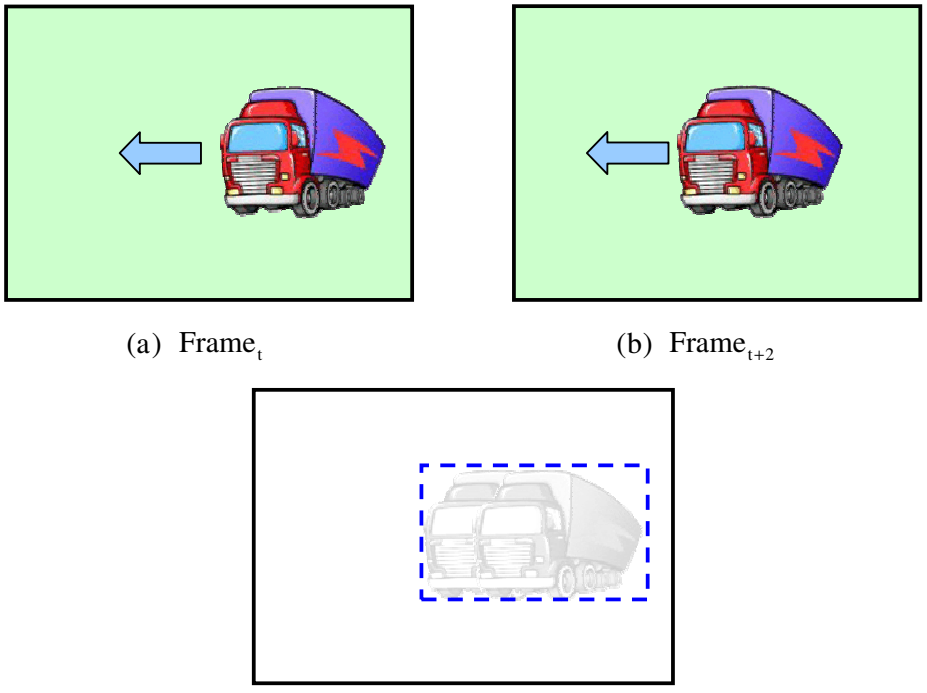


圖3.4　移動物偵測流程

$Frame_1$、$Frame_2$、$Frame_3$為連續輸入三張影像序列，編號分別如下標1、2、3，而$Frame_x$為經過一連串的處理步驟後，最後輸出的圖形。

3.2.1 跳躍式背景相減

如圖 3.5 所示，這是一個跳躍式的背景相減小模組。系統一經觸發運作的時候，便會開始將一張張進來的序列影像編號。假設在第 t 張圖框時(如圖 3.5(a))，這輛車子初始位置在畫面的最右邊且欲往畫面之左邊移動；而在第 t+2 張圖框裡(如圖 3.5(b))，車子移動快到畫面中間，則跳躍式背景相減小模組的運算方式是拿編號第 t+2 的圖框與編號第 t 的圖框，其對應同樣位置座標 (x, y) 內的灰階值做相減，藉此判斷出差異性像素點並表示成二元圖；在此，t 值為 1,3,5,7,9,…。接著，以改良式的簡單連通成分分析法，在編號第 t+1 的圖框上(如圖 3.5(c))框出移動目標，我們可以很明顯的看到虛線框框，即是移動目標物的位置。



(a) Frame_{t}　　(b) Frame_{t+2}

(c) Frame_{t+1}

圖3.5　跳躍式背景相減示意圖

背景相減是一種最簡單，且能夠快速反應兩張圖片差異性的方式。由

於計算複雜度簡單，也不會造成即時系統運作的負擔，這是採用跳躍式背景相減法的主要原因。在跳躍式背景相減的小模組裡，只有編號為 t+2 與 t 的圖框兩兩相減，依次為 F_3-F_1, F_5-F_3, F_7-F_5,…(於此 F 代表 Frame,即圖框之意)，而編號 t+1 的圖框則是拿來統計並計算平均亮度。至於為何在此計算平均亮度，用途稍後做說明。

擷取移動物件，首先要利用圖框差異產生二元圖，不過，由於影像品質容易受亮度的影響，即天候的不同或其他外在光源(即亮度不一)造成品質參差不齊的圖像,若使用 Otsu 所提出之統計式二值化閥值或許可以改善二元圖的品質，不過由於需經過全域之灰階值統計，才能計算出二值化閥值相當耗費時間，可能不適合用於即時動態偵測；因此，在這邊提出了一個能表示明顯差異的動態二值化閥值。

文獻[25]提過，動態門檻值T的設定法，如公式 3-1 所示。

$$T = \beta\mu \tag{3-1}$$

β為常數，其值可針對不同的實驗環境，不一樣的 CCD 攝影器材參數，然後依據研究需求，來做不一樣的比例變動；μ值則為一張圖框的亮度平均值，所以透過實驗我們修改此動態門檻值成T'為適用於此實驗環境的值，如公式 3-2 所示。

$$T' = 0.5\mu + 20 \tag{3-2}$$

圖 3.6 為跳躍式背景相減法的實際情形，圖 3.6(a) 為室外動態資料庫其一中的第 9 張圖框，圖 3.6(b) 為第 11 張圖框，觀察這兩張圖框，有些微的差異。因此在跳躍式背景相減法的小模組裡，先計算第 10 張圖框的

平均亮度，取得動態二值化閥值，再將第 11 張與第 9 張圖框，分別轉灰階、相減並取絕對值，再與T'比較取得二元圖，如圖 3.6(c) 所示。

(a) 第 9 張圖框

(b) 第 11 張圖框

(c) 經動態二值化之二元圖

圖3.6　跳躍式背景相減法實際情形

3.2.2 中值濾波器(Median Filter)

觀看圖 3.6(c) 的二元圖，可以看到雖然車輛的區域，經過跳躍式背景相減法已經出現，不過位於非車子區域的旁邊仍有散落的白色小雜點，這些小白點與我們所關心的車輛是無直接關係的，因此，我們必須修改成像品質。雜訊的形成因素可能有 CCD 的抖動、被照物的本身灰階分布太特別，或者光線的問題所產生...等，常用的雜訊濾除有平滑濾波器與中值濾

波器。本文在這邊只介紹所用到的中值濾波器[20]，其考慮遮罩大小為 3x3，如圖 3.7 所示。

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

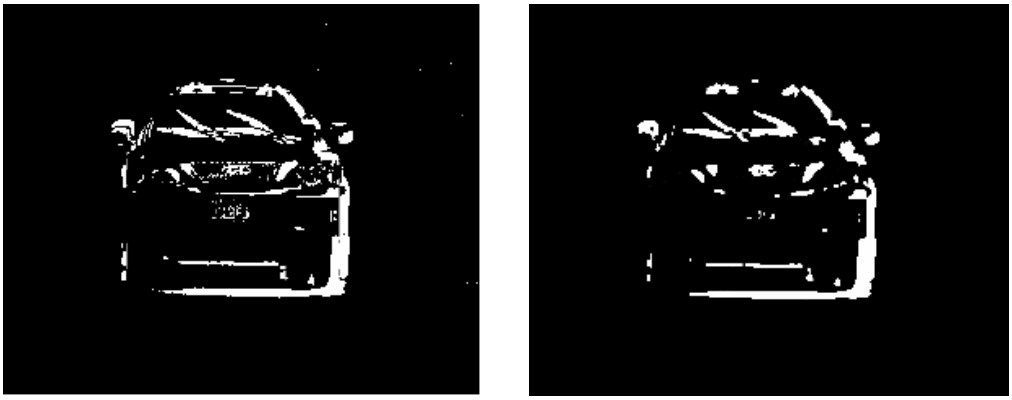
圖3.7　3x3 遮罩

100	24	80
200	54	75
164	35	20

圖3.8　中值濾波器實例

中值濾波器在灰階值的圖裡，其考慮方式則是將 P_1~P_9 的像素點灰階值由小至大做排列，然後 P_5 的值則代換為排列後之中間值。實例如圖 3.8 所示，考慮此 3x3 遮罩大小裡面的 9 個數值，由小至大排列順序如下，{20, 24, 35, 54, 75, 80, 100, 114, 200}，因此，經過中值濾波後，中間的數值會從原先的 54 改成 75 來做替代。而在二元圖的圖像裡，考慮的數值只有 0 或 1，一樣可以用同樣的方式來做中值濾波，功能就是消除散落的小白點。

圖 3.9 為中值濾波器的實例，可以看到原先的圖 3.9(a) 的右上角有一些散落的白點，這些並不是我們所要的資訊；不過，透過中值濾波之後，很明顯的消除非預期中的雜訊外，圖片也經過改善變的平滑許多，結果如圖 3.9(b) 所示。



(a) 中值濾波前　　(b) 中值濾波後

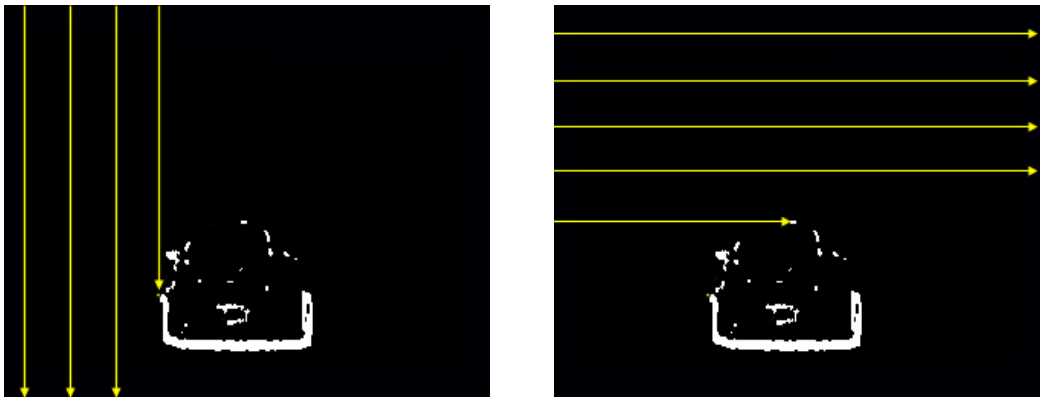
圖3.9　中值濾波實例

3.2.3 簡單連通成分分析

經過跳躍式的背景相減模組二值化及中值濾波器後，可以很明顯的看到移動物體所在的位置，接下來就是要標示出移動物的所在座標。對於取得物體的詳細資訊，多數的文獻方法皆是採用較嚴謹的「連通成分分析法」(Connected Component Analysis, CCA)來標示出物件座標或者是 Pixel 數量等等，對於一台車子(大圖框)而言，這個方法是比較花時間的，不利於即時系統運用；此外，經過動態二值化所得的二元圖形，並非「真實」的連通成分，只是極為類似連通的破碎小區塊之集合，所以，在此提出一個「簡單連通成分分析法」，不但可以快速標出移動目標，又不必像 CCA 需要耗費很大的處理運算時間，其方法則如圖 3.10 所示。

兩側 x 座標的找法如圖 3.10(a) 所示，我們只需要做掃瞄動作判斷，由上而下、且由左而右，進行掃描，當找到白色像素點時便跳出，並紀錄此點座標為 x_1。反之，由上而下、且由右而左，掃瞄做同樣的動作得到座標為 x_2。同理，y 座標的找法則如圖 3.10(b) 所示，搜尋到 y_1(上)及 y_2(下)座標。最後，我們便可以利用(x_1,y_1)及(x_2,y_2)於編號偶數的圖框上，標示出

移動物件的位置。



(a) x 座標判斷法則　　(b) y 座標判斷法則

圖3.10　簡單連通成分之 x 與 y 座標的找法

圖3.11　擷取移動物件

圖 3.11 為經過簡單連通成分分析之後所擷取的移動物件示意圖，框框即為移動物(車體)。在程式的撰寫上，為了達到加快速度，可以把 x、y 座標的搜尋法則寫在一起，首先，由上而下、且由左而右的搜尋白色像素點，當掃描到每一個白色像素點時換行繼續掃描並記錄此時的 x 座標與 y 座標，當畫面掃完一次時，最左側白色像素點的 x 座標即為 x_1，最右側白色像素點的 x 座標即為 x_2，而每當在掃描到白色像素點時也不斷的去比較其 y 座標值，較小的取出即為 y_1，如此一來便有了 3 個座標值。剩下最後一個座標的找法為，由左而右、但改為由下往上掃描，當紀錄過後此列是完

全穿越(亦即此列完全沒有白色像素點)時，便跳過不掃描此列，只需掃描沒有完全穿越列，並比較每次掃到白點的 y 座標值，最大取出即為 y_2。

3.2.4 考慮車牌幾何位置

經過簡單的類連通成分分析法，我們可以框出的移動物件幾乎是一台車子，但如果直接對這台車子的影像來做後續處理，仍然還未降低運算量，因為有太多面積並不會出現所感興趣的車牌候選區；所以，根據 CCD 與車子的拍攝角度，以及車輛行駛的歪斜情形如圖 3.12 所示，我們考慮了

(a) 由上往下拍攝車牌右偏情形　　(b) 由下往上拍攝車牌左偏情形

圖3.12　CCD 與車輛拍攝角度情形

在一台車子的影像裡，車牌大概會出現的幾何位置，然後擷取我們欲處理的圖像部份做後續的運算即可，如圖 3.13 所示。

在圖 3.13 裡，X 表示寬度大小，數值為 x2-x1；Y 表示高度大小，數值為 y2-y1。在寬度的部份，我們捨棄掉左右各百分之二十的運算面積；而高度部分，則捨棄掉百分之四十的運算面積；這樣的結果使得最後運算的面積，縮小為移動物偵測面積的百分之三十六，如斜線部分所示，可以大幅的縮小圖像及降低運算量，如此一來，便會加快系統運算時間。接下

來的步驟，即是把這百分之三十六的圖像丟入至車牌定位模組，做更精確的車牌偵測與判斷定位。

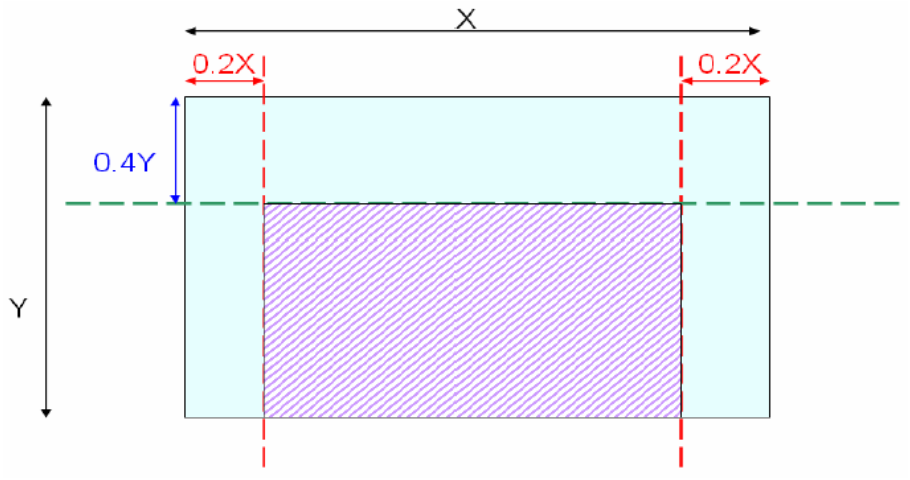


圖3.13　考慮車牌幾何位置

圖3.14　車牌幾何位置應用情形

圖 3.14 畫面中之大矩形所框出物體為移動物件，而小矩形的範圍為考慮車牌出現的幾何位置，從示意圖上可以知道 不論 CCD 與車子的相對位置是左偏或者右偏以及稍為上下的傾斜，車牌的位置大多會出現在小矩形的範圍內。

3.2 車牌定位模組

本文經過了移動物偵測模組，已經幾乎能夠看到突顯的車牌，不過在這粗略車牌圖像裡尚含有一些我們不想要且多餘的資訊。因此，緊接著就是要想辦法濾除非車牌字元區塊，進而框出更精準的車牌位置，能夠讓車牌辨識系統做後續的「字元分割」、「字元辨識」。整個系統的成效關鍵即在此步驟。車牌定位模組亦可分為數個部份，依次為「影像前處理」、「掃描灰階變異」、「垂直邊緣偵測」、「佈線演算」、「車牌定位」共五個步驟，在以下的小節將會分別詳細介紹之。

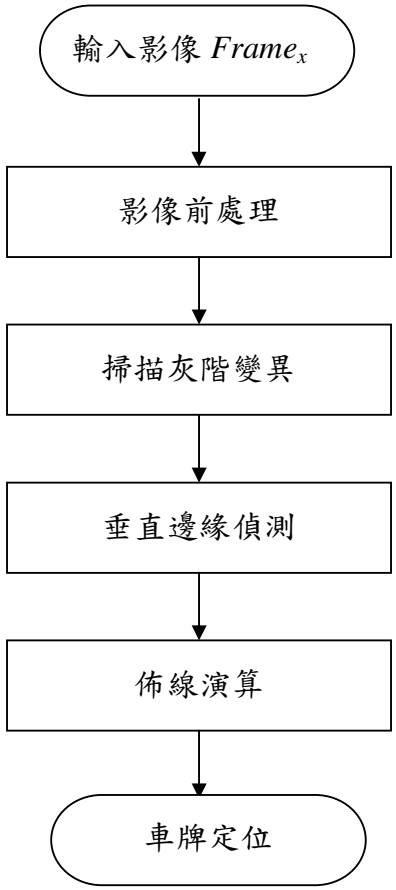


圖3.15　車牌定位模組流程圖

圖 3.15 為車牌定位模組的處理流程圖，首先，透過影像前處理降低資料維度及減少亮度依賴，再透過掃描灰階變異尋找車牌候選區，在這步驟產生的車牌候選區，至多只有兩個，至於原因在以下小節將會做說明。產生車牌候選區之後，再經過垂直邊緣偵測濾除強烈的水平雜訊，最後透過佈線演算與車牌認證定位來得到最終的車牌候選區。

3.3.1 影像前處理

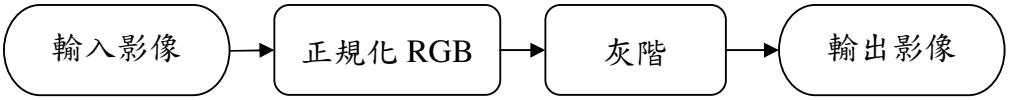


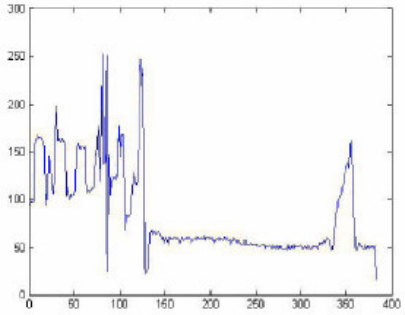
圖3.16　影像前處理步驟

為了降低資料維度，首選當然是將彩色 RGB 三通道圖像轉為灰階單通道圖像，灰階轉換公式如第二章裡之公式 2-4 所示。此外，由於同一個物體的拍攝影像容易受到光線的影響造成顏色不同，因此，本文採用第二章曾經提過之正規化 RGB 來做影像前處理，其先後順序如圖 3.16 所示，先將影像轉灰階之後，再做正規化 RGB 的動作。

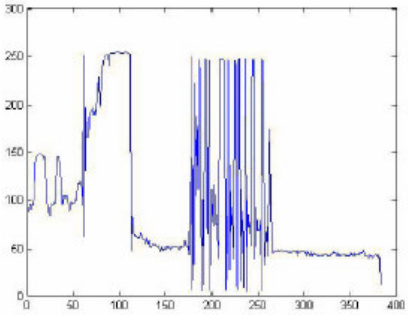
3.3.2 掃描灰階變異

由於 Xiaobo Lu 等人[26]提出的方法裡提到，如果在含有車輛的影像裡面做水平掃描，觀察發現，在非車牌的區域並無顯著特徵，但在車牌的區域會有強烈的灰階變異的特徵，如圖 3.17 所示，可以用這顯著的特徵來判斷車牌裡字元的相對高度位置。

(a) 灰階含有車牌之車輛圖像



(b) 掃描 1 號線的灰階變化情形



(c) 掃描 2 號線的灰階變化情形

圖3.17　觀察掃描線的灰階變化[26]

如圖 3.17(a)為一轉灰階後含有車牌的車輛圖像，並在畫面中間下方附近水平掃描兩條線，如圖上標示號碼 1 與 2，圖 3.17(b)為掃描第一條線的灰階變化的情形，圖 3.17(c)為掃描第二條線的灰階變化情形，透過兩個圖的比較，很明顯的經過車牌的第二條線在字元部份有強烈的反覆變化的灰階值特徵出現，本文欲考慮並使用此特徵來縮小搜尋區塊。

既然車牌在字元附近有如此強烈的特徵，Xiaobo Lu 等人提出一個方法來描述這些鄰近的灰階情形，假設掃瞄單一條線上的座標 x 的灰階值為 $f(x)$，則 $f(x)$ 的一階導數 $\frac{df(x)}{dx}$ 可以用差分的方式來表示，如公式 3-3 所示[26]。而 x 的範圍從 2~Width，Width 為圖片寬度。

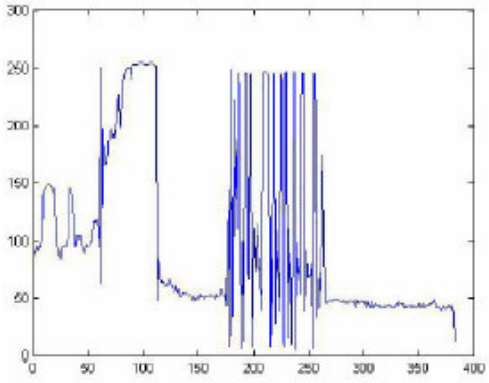
$$\frac{df(x)}{dx} = f(x) - f(x-1) \tag{3-3}$$

圖 3.18 為考慮灰階變化的一階導數情形，圖 3.18(a)為掃描含有字元的車牌裡的其中一條線，觀察看到灰階變化情形在車牌附近有凸向上與凹向下的反覆出現特徵，且值介於 0~255 之間。不過，雖然有這個特徵，仍然不好直接使用，需經過轉換。所以，經過一階導數後，觀察其圖形如圖 3.18(b)所示，利用閥值T_d濾除掉變化不夠強烈的部份，結果如圖 3.18(c)所示。

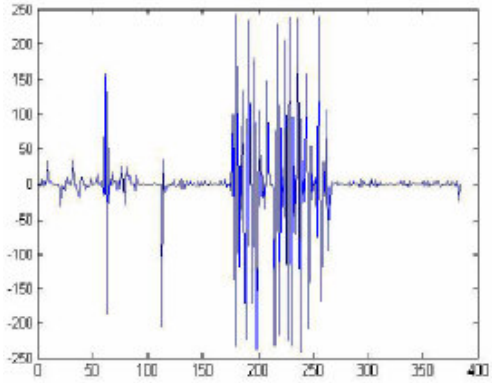
觀察圖 3.18(c)發現，既然在車牌附近的字元的掃描線，具有這麼強大的特徵，本文提出了一套計算方式來尋找合乎車牌字元位置的掃描線，其方法為統計線密度，如公式 3-4 表示。

$$\xi = \frac{\sum_{i=left}^{right} (\frac{df(x)}{dx})^2}{L} \tag{3-4}$$

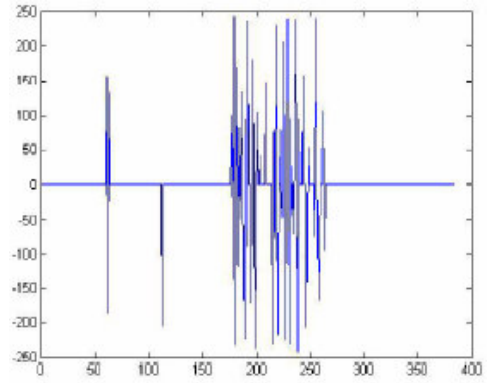
ξ為線密度的算法，left 與 right 的數值如圖 3.19 上所標示。從左往右尋找到第一個非零點就定義為 left；而從右往左尋找到第一個非零的點就定義為 right。分子為出現在 left 與 right 間隔裡面的所有差分平方和，L值大小為 right 減去 left，在這邊一樣設定一個門檻值T_ξ，如果大於T_ξ則判定此掃描線位於車牌區塊內。



(a) 掃描線含有字元的的車牌



(b) 考慮灰階變異之一階導數

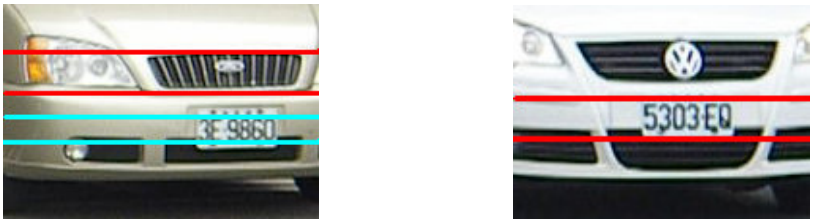


(c) 濾除較小的變異點

圖3.18　利用一階導數描述灰階變化情形[26]

圖3.19　車牌附近掃描圖

利用掃描灰階變異可以得到最多一至兩個次候選區，其原因乃因為進入這一個步驟的圖形大概可分為兩種，如圖 3.20 的情形，圖 3.20(a)由於車牌上方具有強烈的垂直紋理特性，利用本文所提出之方式，仍會統計出上方具有垂直紋理之非車牌區塊，所以此情形的次候選區會出現兩個；圖 3.20(b)由於車牌上方為水平邊緣，在做線密度的判定時，並不會出現此區塊，所以次候選區是唯一的。



(a) 車牌上方具有垂直紋理特徵　　(b) 車牌上方是水平紋理特徵

圖3.20　候選區出現的情形

3.3.3 垂直邊緣偵測

產生了次候選區之後，緊接著就是要能夠擷取出更精確的車牌文字部分，所以下一個步驟主要是針對文字的邊緣特徵來處理，即所謂的邊緣偵測。

一般做邊緣偵測的目的是考慮在字元附近的邊緣，有很強烈的黑白交叉出現的對比層次，如圖 3.21 所示。因此，如果將此圖片做邊緣偵測的話可以很明顯的觀察到，密集的邊緣出現在字元附近，我們便可以利用這樣的特性找出正精確的車牌位置。

圖3.21　灰階車牌圖

本文所採用的是 Sobel 邊緣偵測 3x3 的遮罩，其遮罩先前於第二章已討論過。若考慮一整張車輛的圖像時，由於在車牌附近具有很強烈的水平邊緣，例如防撞檔桿或引擎排氣散熱孔...等，如圖 3.22 列出情形所示；如果考慮 Sobel 的水平加垂直邊緣偵測的運算是不智的，也因此，幾乎所有的文獻在做邊緣偵測時通常都只有考慮了垂直邊緣偵測，因為垂直邊緣偵測不但能夠保留文字垂直方向邊緣的特徵之外，更能夠濾除掉相當多的水平方向的邊緣雜訊。

(a) 引擎蓋的散熱孔

(b) 防撞檔桿

圖3.22　具有強烈水平特徵的車輛圖像

如圖 3.23 為 Sobel 的邊緣偵測結果比較圖；圖 3.23(a)為車輛原始圖形；圖 3.23(b)為經過掃瞄灰階變異後所得到的車牌次候選區；圖 3.23(c)為考慮垂直與水平方向的所有邊緣梯度的結果圖；圖 3.23(d)為只考慮水平方向的邊緣梯度圖；圖 3.23(e)為只考慮垂直方向的邊緣梯度圖。很明顯的看出利用 Sobel 垂直方向遮罩所考慮出的邊緣是最佳的，因為它可以濾除掉相當多的強烈水平邊緣(雜訊)。

(a) 車輛原始圖形

(b) 利用灰階變異所得次候選區

(c) Sobel 邊緣偵測結果圖

(d) 考慮 Sobel 水平邊緣偵測結果圖

(e) 考慮 Sobel 垂直邊緣偵測結果圖

圖3.23　Sobel 邊緣偵測結果比較圖

本文所採用的方式也是 Sobel 的垂直邊緣偵測，不過雖然包含了濾除強烈的水平邊緣之外，其實最主要的原因是如果只考慮單一一個垂直邊緣偵測的遮罩，其又能減少處理的運算時間。此外，在 Sobel 邊緣偵測的步驟裡面，將先前依據亮度所算出的動態二值化閥值T'拿來運用，只要邊緣偵測出的結果梯度值大於T'，則保留原值；如果小於T'，則設為黑色，產生一張邊緣偵測的梯度圖。

(a) Sobel 水平邊緣偵測結果圖　　(b) 經過動態閥值保留邊緣圖

圖3.24　Sobel 邊緣偵測加入動態閥值保留圖

圖 3.24(a)為直接做 Sobel 水平邊緣偵測結果，圖 3.24(b)為 Sobel 水平邊緣偵測加入動態閥值T'並且進一步判斷，篩選保留邊緣像素點之結果。

3.3.4 佈線演算

經過垂直邊緣偵測後，已濾除強烈的水平邊緣，從梯度圖觀察，保留的像素點大多數為字元的邊緣特徵，利用其叢聚的特性拿佈線演算將之連結起來。所以在考慮了車牌字元相距的距離 w 範圍後，設計一個佈線濾波器，其想法源自於[4]的 AIM 技巧，經過修正後，如圖 3.25 所示。

```
產生一張全黑的圖為 OutputImage[x][y];
 for   y = 0   to   Height
 {
   for   x = 0   to   Width
   {
     Pixel_start1 = InputImage[x][y];
       Pixel_start2 = OutputImage[x][y];
     if ( Pixel_start1 > threshold   ||   Pixel_start2 ! = 0 )
       found = false;
         for   i = x + w   to   x
         {
           Pixel_end = InputImage[x][j];
           if ( Pixel_end > threshold)
             found = true;
           if ( found = true)
             OutputImage[i][y]==1;
       }
   }
 }
```

圖3.25　佈線演算法

圖 3.25 中，InputImage 為原始輸入梯度影像；OutputImage 為一張全黑影像，與輸入影像同大小，經過運算會不斷的修改其內容值；*threshold* 即先前所提過的動態閥值T'，w 即為考慮在畫面裡，預定出現的字元間距寬度像素值。

演算法的運作法則如圖 3.26 表示。由上而下，由左而右做一個水平方向的掃描，當掃描時持續判斷，邊緣偵測梯度圖的每點像素值，是否大於我們所預定的 *threshold*，如果有的話定義為起始點 $Pixel_{start}$，然後往右跳 w 的距離，並依序往回判斷是否有結束點 $Pixel_{end}$ 存在，而結束點的成立條件，即其梯度值也必須大於 *threshold*，如果條件皆成立，依次往回填入白點(值 255)，即將起始點與第一個結束點連成直線。

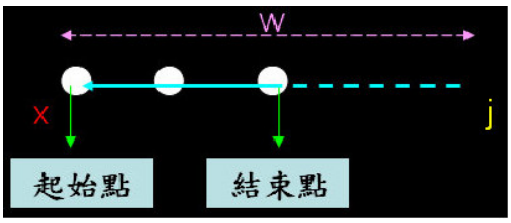


圖3.26　佈線方式

(a) 經過動態閥值保留邊緣圖　　(b)　經過佈線結果圖形

圖3.27　佈線演算結果

圖 3.27(a)為經過動態閥值所保留的邊緣圖，圖 3.27(b)為經過佈線演算所得到的結果圖，可以看到車牌的位置顯而易見。

3.3.5 車牌定位

由於圖片經過上述各步驟後，其大小已經大幅縮小，所以最後步驟就是把這些白點作一次連通成分分析，亦不會浪費太多時間，此部份考慮條件相當簡單，設 W 為車牌寬度，H 為車牌高度，Wp 為位於候選車牌區塊內的白點數量。本文考慮條件為寬高比(W/H)以及面積大小(W×H)在一定的合理範圍內，還有密度值 D = Wp/(W×H)大於一定的範圍，我們便能夠把車牌定位出來。其中，本實驗之寬高比參數範圍介於 2.2 與 4.3 之間，而面積大小(單位:像素點)範圍為大於 1500 且小於 8000，密度範圍為大於 0.65 且小於 0.93，也就是說在本實驗系統只要合乎以上所有值，便可以定位出車牌，否則判斷失敗。

圖3.28　車牌定位結果

在佈線結果圖上，考慮了寬高比及面積大小、密度大小後，於原圖找出合乎標準的最終候選區塊，如圖 3.28 所示，框框即是定位之車牌位置。