

第二章 文獻探討與回顧

過去已經有相當多的研究者致力於車牌辨識系統的研究，根據國內外之車牌辨識系統的相關文獻，依照處理的資料類型，大致上可以分兩類，即靜態影像系統與動態影像系統。所謂靜態影像系統是指利用數位相機來拍攝，存成點矩陣(BMP)的圖像後再來做運算的動作；而動態影像是指利用 CCD 或攝影機來攝影取像，將所得的動態影像(AVI)檔案來運算。這兩個的最大區別為靜態影像運算不受時間限制，所以相當強調精確度、成功率；動態影像為序列流資料，系統順暢與否跟運算時間有直接關係，因此通常作法是將低運算複雜度，結果導致成功率相較於靜態系統低了許多。而本文所採用的資料為動態影像系統，希望在降低運算複雜度的同時，也能夠提升系統的運算速度和提高成功定位率。

目前車牌偵測定位的技術很多，包括了「邊緣偵測」(Edge Detection)[2][10]，「形態學運算」(Morphological Operations)[5][11]，「人工類神經網路」(Artificial Neural Network, ANN)[12]，「模糊集合理論」(Fuzzy Sets Theory)[13]，「向量量化」(Vector Quantization, VQ)[14]，色彩分析(Color Analysis)[15][16]，「霍夫轉換法」(Hough Transform)[17]等等，在眾多方法中，每種方法的發展皆分別針對不同的問題加以克服處理，但是，仍然有些問題，它們無法獨自解決。這是因為使用單一的方法，無法得到最有效的偵測結果。所以，最近這幾年，越來越多的研究，都採用上述各自方法的優點，互相結合來達到最佳偵測結果。而本文主要是利用「掃描灰階變異」取出車牌之次候選後，再搭配「垂直邊緣偵測」來完成車牌定位的動作。以下 2.1 節為車牌偵測基本理論相關介紹；2.2 節為靜態系統相關之研究；2.3 節為動態系統相關之研究；2.4 節為綜合比較分析。

2.1 基本理論相關介紹

2.1.1 色彩空間

一般常見的色彩空間是由國際照明委員會(Commission International de l'Eclairage, C.I.E.)所制定的，而在車牌辨識系統這領域常見的色彩空間有如 RGB, 正規化 RGB, YIQ,...等，依序介紹如下：

(1) RGB 色彩空間

在 RGB (俗稱三原色) 的色彩空間中，每種色彩皆是以紅(R)、綠(G)、藍(B)為主要成分表示，而每一個成分的亮度範圍皆從 0~255，因此所有的顏色都可以看作是三原色任意混合搭配，即顏色數目在 RGB 的色彩空間下共有 $2^8 \times 2^8 \times 2^8 = 16,777,216$ 種。為了方便表示，假定所有色彩值已經被標準化，所有的 RGB 值都標準化至 [0,1] 的範圍內，利用直角座標來表示 RGB 的系統如圖 2.1 所示。

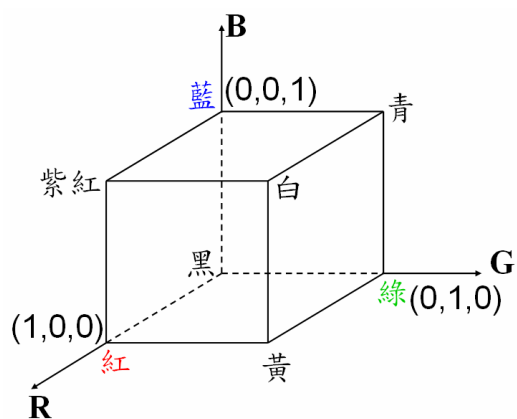


圖2.1 RGB 色彩立方體的圖示[18]

圖 2.2 為彩色影像之 RGB 色彩空間三分量之分解與混合圖，圖 2.2(a)

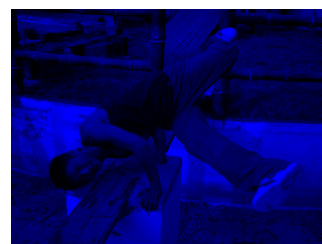
為 R 平面，圖 2.2(b)為 G 平面，圖 2.2(c)為 B 平面，圖 2.2(d)為三個分量的加總圖形。



(a) R 平面



(b) G 平面



(c) B 平面



(d) RGB 混合圖形

圖2.2 彩色影像的 RGB 分解與混合圖

(2) 亮度正規化 RGB 色彩空間

一般所使用的 RGB 色彩空間系統容易受到光線的影像，因此同一種比例成分的颜色，在光源亮度不同的條件之下，所呈現的颜色給予我們的感受是不同的，所以亮度正規化 RGB 色彩空間，主要是希望能夠減少颜色對亮度的依賴性。公式如 2-1、2-2、2-3 所示[19]。

$$R' = \frac{R}{R+G+B} \quad (2-1)$$

$$G' = \frac{G}{R+G+B} \quad (2-2)$$

$$B' = B \quad (2-3)$$

利用 R' ， G' ， B' 來取代原先的 RGB 色彩空間，便能使顏色對光的靈敏度減少。至於 B 對於光線的敏感度較小，故在這邊可以忽略，不經轉換直接使用 B 值就可以。圖 2.3 為經過亮度正規化的前後比較圖。



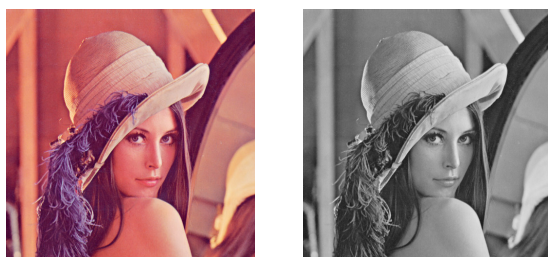
(a) 原圖 (b) 經過 RGB 正規化後

圖2.3 亮度正規化比較圖

(3) YIQ 色彩空間

YIQ 其為 NTSC(National Television Standards Committee)傳送影像時所用的規格，而 YIQ 與 RGB 色彩空間存在著線性轉換關係，其公式如 2-4 所示[19]。

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.327 \\ 0.212 & -0.323 & 0.313 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-4)$$



(a) 彩色 Lena 影像 (b) 灰階 Lena 影像

圖2.4 Lena 的 Y 成分圖

其中 Y 為亮度(Luminance)成分，I 與 Q 則為色差；主要應用於電視訊號，以配合黑白電視訊號，此色彩空間能夠把明亮度與彩色成分分離。另外，Y 值在這邊也是俗稱的灰階值，其值範圍為 0(黑色)~255(白色)，如圖 2.4 所示為 Lena 的彩色影像轉換為灰階影像。

2.1.2 二值化

一般的灰階圖像每一個像素點的灰階度有 0~255 共 256 階，但是在某些應用中，有時只需要若干個灰階度就夠了。例如，黑白的圖像裡只需要 0 與 1 兩個灰階度表示。因此如何決定一個適當的門檻值使得灰階影像轉換為黑白影像的確值得討論[20]。以下介紹固定式二值化法與統計式的 Otsu 二值化法。

(1) 固定式二值化法

固定式二值化判斷法則就是設定一個門檻值 T ，依序掃描灰階圖之原來像素值 $f(x, y)$ ，如果大於 T 設為白色(255)，小於 T 則設為黑色(0)，如公式 2-5 表示， $f'(x, y)$ 為經改變後之像素值[21]。二值化的優點在於能夠將灰階圖轉換為黑白圖，使得後續的影像處理步驟，運算更加簡單。如圖 2.5 的(a)與(b)分別為固定閾值 $T=70$ 與 $T=150$ 的二值化圖形。

$$f'(x, y) = \begin{cases} 255 & f(x, y) > T \\ 0 & f(x, y) < T \end{cases} \quad (2-5)$$

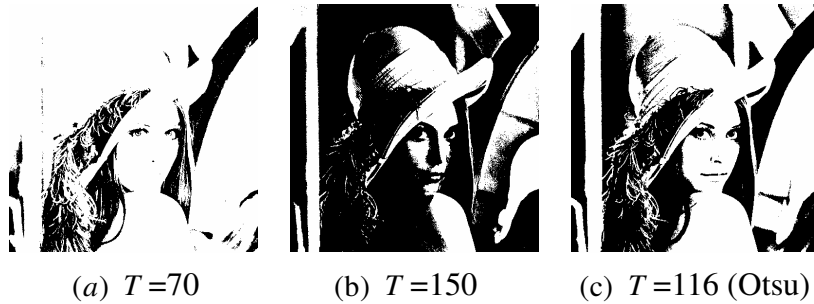


圖2.5 Lena 的二值化圖

(2) 統計式 Otsu 二值化法

Otsu 二值化法[20]目的是利用統計方法找到一個最佳閾值，並將灰階圖轉為二元圖，其方法如下說明。倘若假設 T^* 是最佳閾值，且 T^* 將影像中的像素值分為 C_1 和 C_2 二區如圖 2.6。位於 C_1 內的像素點，其灰階值 $f(x, y)$ 必滿足 $0 \leq f(x, y) \leq T^*$ ；位於 C_2 內像素點，其灰階值 $f(x, y)$ 必滿足 $T^*+1 \leq f(x, y) \leq 255$ 。Otsu 提出兩個條件，只要擇一成立即可決定 T^* 的值。

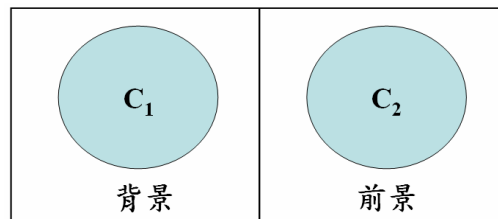


圖2.6 二區的例子[20]

第一條件為 T^* 的決定會滿足 C_1 和 C_2 之間的變異數(Between-Variance)為最大，即兩群分的越開；第二條件為 T^* 的決定會滿足 C_1 內的變異數(Within-Variance)加上 C_2 內的變異數之和為最小，即群體之間的灰階值較近似，差異性越小。令影像大小為 $N = I_w \times I_h$ ， I_w 與 I_h 分別為影像的寬與高，則灰階值 i 的機率表示為

$$P(i) = \frac{n_i}{N}$$

n_i 表示灰階值 i 在影像中出現的次數，且 $0 \leq i < 255$ ，根據機率原理

$$\sum_{i=0}^{255} P(i) = 1$$

假設 C_1 和 C_2 內的像素點個數所佔的比例分別為 R_1 和 R_2 ，如公式 2-6，並且滿足 $R_1 + R_2 = 1$ 。而 C_1 和 C_2 的期望值分別為 E_1 和 E_2 ，計算方式如公式 2-7 所示[20]。

$$R_1 = \Pr(C_1) = \sum_{i=0}^{T^*} P(i) \quad R_2 = \Pr(C_2) = \sum_{i=T^*+1}^{255} P(i) \quad (2-6)$$

$$E_1 = \sum_{i=0}^{T^*} \frac{P(i)}{R_1} \times i \quad E_2 = \sum_{i=T^*+1}^{255} \frac{P(i)}{R_2} \times i \quad (2-7)$$

則利用 E_1 和 E_2 ，可以算出 C_1 和 C_2 的變異數為 σ_1^2 和 σ_2^2 ，如公式 2-8[20]。

$$\sigma_1^2 = \sum_{i=0}^{T^*} (i - E_1)^2 \times \frac{P(i)}{R_1} \quad \sigma_2^2 = \sum_{i=T^*+1}^{255} (i - E_2)^2 \times \frac{P(i)}{R_2} \quad (2-8)$$

所以， C_1 和 C_2 內變異數和為

$$\sigma_w^2 = R_1 \sigma_1^2 + R_2 \sigma_2^2$$

在 C_1 和 C_2 之間的變異數可表示為

$$\sigma_B^2 = R_1 (E_1 - E_{T^*})^2 + R_2 (E_2 - E_{T^*})^2$$

E_{T^*} 表示整張原始影像的平均值。 E_{T^*} 計算如下

$$\sum_{i=0}^{255} \frac{n_i \times i}{N} = \frac{1}{N} \sum_{i=0}^{255} n_i \times i$$

因此透過證明推導，可以知道 σ_B^2 ， σ_W^2 和 $\sigma_{T^*}^2$ 存在這樣的關係：

$$\sigma_B^2 + \sigma_W^2 = \sigma_{T^*}^2$$

而 $\sigma_{T^*}^2$ 為原始影像的變異數。由於 $\sigma_{T^*}^2$ 為定值， C_1 和 C_2 之間的變異數最大化的解也就等於 C_1 和 C_2 內的變異數和的最小化解。因此只需考慮找一個 T^* 使 C_1 和 C_2 內的變異數和最小就夠。方法為在 0~255 之間，一個個將灰階值代入 σ_W^2 式子內，將所有的值全部求出後，再經比較最小的 σ_W^2 值所對應的閾值即為 T^* [20]。如圖 2.5(c) 所示為 Lena 的 Otsu 二值化圖形，最佳二值化閾值經求得後為 $T^*=116$ 。

2.1.3 邊緣偵測

在車牌辨識系統的文獻中，邊緣偵測[18][20]幾乎是不能少的步驟，原因在於車牌裡的字元與背景的黑白對比顏色，經過邊緣偵測之後，會有很密集群聚的邊緣特徵，如圖 2.7 所示。



(a) 車牌灰階圖

(b) 車牌經過邊緣偵測

圖2.7 車牌邊緣特徵

邊緣偵測的概念如圖 2.8，很明顯的從觀察知道，黑白色的交界處 (Boundary) 會顯現出大的灰階值變化，如果我們把交界處再拿放大鏡看，可以看到剖面上的灰階值呈一個斜坡往上的特性；將灰階剖面進行一次微分，顯示一階導數在進入斜坡和離開斜坡上的是正的，而在斜坡上的點是定值，在同質區塊中的點為零；且二階導數在交界邊緣的暗邊之變遷處是正的，在交界邊緣的亮邊之變遷處是負的，沿著斜坡或者灰階值不變的區域中為零。

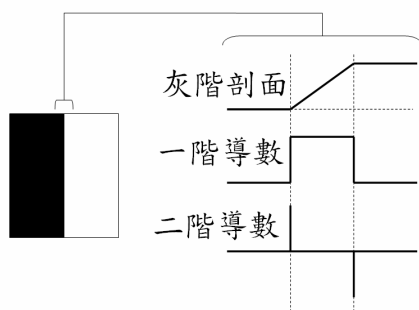


圖2.8 不同色之邊緣概念說明圖[18]

因此，二階導數在邊緣附近有兩個特性：(1) 其會對影像中的每一個邊緣產生兩個值(一正一負)；(2) 一條連接二階導數之正極值和負極值的假想直線將會穿過接近邊緣中央的零點。此二階導數之零穿越(zero-crossing)性質對於找出邊緣的中心極有用。目前較常使用的邊緣偵測有如拉普拉斯運算子 (Laplacian Operator)、Prewitt 運算子及 Sobel 運算子...等，以下分別詳細介紹之。

(1) 拉普拉斯測邊

設 $f(x, y)$ 為座標 (x, y) 的灰階值表示函式，先考慮一維的梯度算法，我們對 $f(x, y)$ 沿 x 軸微分得差分

$$\nabla_x f = f(x+1, y) - f(x, y)$$

再進而對 x 軸微分得

$$\begin{aligned}\nabla_x^2 f &= f(x+2, y) - f(x+1, y) - [f(x+1, y) - f(x, y)] \\ &= f(x+2, y) - 2f(x+1, y) + f(x, y)\end{aligned}$$

令 $x = x+1$ 作參數變換，可得

$$\nabla_x^2 f = f(x+1, y) - 2f(x, y) + f(x-1, y)$$

同理，沿著 y 軸微分二次得

$$\nabla_y^2 f = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

合併二個軸的二次微分效應，可得拉普拉斯運算子如下

$$\begin{aligned}\nabla^2 f &= \nabla_x^2 f + \nabla_y^2 f \\ &= f(x, y+1) + f(x+1, y) + f(x, y-1) + f(x-1, y) - 4f(x, y)\end{aligned}$$

在實作上時，拉普拉斯運算子可以透過圖 2.9 的遮罩(Mask)以迴旋積(Convolution)的方式，作用在影像上以完成測邊的工作。

0	1	0
1	-4	1
0	1	0

圖2.9 拉普拉斯運算子對應的面罩[20]

圖 2.10 為拉普拉斯測邊的應用實例結果。



(a) 彩色原圖

(b) 經過拉普拉斯邊緣偵測

圖2.10 拉普拉斯測邊結果

(2) Prewitt 測邊

圖 2.11 分別為 Prewitt 邊緣偵測之測 x 方向以及 y 方向變化的遮罩，在 3×3 的遮罩裡第一列和第三列的差值，類似在 x 方向上的導數；在 3×3 的遮罩裡第一行和第三行的差值類似在 y 方向上的導數。

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

(a) 測 x 方向的變化

(b) 測 y 方向的變化

圖2.11 Prewitt 測邊遮罩[20]



(a) 彩色原圖

(b) 經過 Prewitt 邊緣偵測

圖2.12 Prewitt 測邊結果

圖 2.12 為 Prewitt 測邊的應用實例結果。

(3) Sobel 測邊

Sobel 邊緣偵測的遮罩跟 Prewitt 的極為類似，只有一個微小變動，如圖 2.13 所示。在中心係數使用 2 的加權值，藉由給中心點更大的重要性，2 的加權值用來達成一些平滑的效果。

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

(a) 測 x 方向的變化 (b) 測 y 方向的變化

圖 2.13 Sobel 測邊遮罩[20]



(a) 彩色原圖

(b) 經過 Sobel 邊緣偵測

圖 2.14 Sobel 測邊結果

圖 2.14 為 Sobel 測邊的應用實例結果。

而本文經過實驗比較得知，同樣遮罩大小下的各種邊緣偵測運算子中，Sobel 的運算子能夠偵測出的邊緣最多，效果最佳，也因此觀看近年來的國內外文獻，幾乎都採用 Sobel 邊緣偵測的理由大概於此。圖 2.15 為各種邊緣偵測運算子的比較圖形。



(a) 彩色原圖



(b) 拉普拉斯邊緣偵測



(c) Prewitt 邊緣偵測



(d) Sobel 邊緣偵測

圖2.15 各種邊緣偵測運算子比較

2.1.4 形態學運算

形態學運算，主要是用來處理與描述圖片的結構形狀，例如區域形狀、邊界、骨架...等。在車牌辨識系統上，常常在分離前景與背景的步驟後，需要進一步處理或描述前景之特徵；因此，形態學運算子在車牌辨識系統裡，仍然佔有一席之地。在這裡，可利用數學上的形態學為背景當做抽取影像成份的工具。這邊主要介紹常用的四大形態學運算子如下，侵蝕(Erosion)、擴張(Dilation)、斷開(Opening)、閉合(Closing)[18]。

(1) 侵蝕(Erosion)

侵蝕運算是利用一個結構化元素(Structure Element, SE)來對二值化圖(以下簡稱二元圖)作用，功用為除去影像裡散落(距離較遠)且不相干的雜點。考慮一個 3×3 的結構化元素如圖 2.16，其運算法則如公式 2-9 所示。

也就是說要填入 Z_5 的值，就是考慮周圍八個像素點的 AND 邏輯運算之後的值。即周圍八個點要全部都是 1， Z_5 值才為 1。

$$Z_5 = \{Z_1 \cap Z_2 \cap Z_3 \cap Z_4 \cap Z_6 \cap Z_7 \cap Z_8 \cap Z_9\} \quad (2-9)$$

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

圖2.16 3x3 的結構化元素

圖 2.17 為侵蝕運算前後比較圖，很明顯的看到影像裡包含了蠻多散落的白色雜點或狹長的細線，利用侵蝕運算可以將這些不想要的雜訊做濾除的動作，使畫面變的更加清楚。



(a) 尚未侵蝕運算

(b) 經過侵蝕運算

圖2.17 侵蝕運算前後比較

(2) 擴張(Dilation)

擴張運算同樣是利用一個結構化元素來對二元圖的影像作用，其功能為填補破碎區塊，例如同質的東西因為雜訊干擾使的二元圖產生破碎或縫

隙，則經過擴張運算能夠將除了能夠使斷開部分或缺口補起來之外，也能夠使輪廓變的部份平滑。同樣考慮一個 3×3 的結構化元素如圖 2.16，其運算定義如公式 2-10 所示。 Z_5 的值，為考慮周圍八像素點的 OR 邏輯運算後的值。即周圍八像素點只要有一個為 1，則 Z_5 值為 1。

$$Z_5 = \{Z_1 \cup Z_2 \cup Z_3 \cup Z_4 \cup Z_6 \cup Z_7 \cup Z_8 \cup Z_9\} \quad (2-10)$$

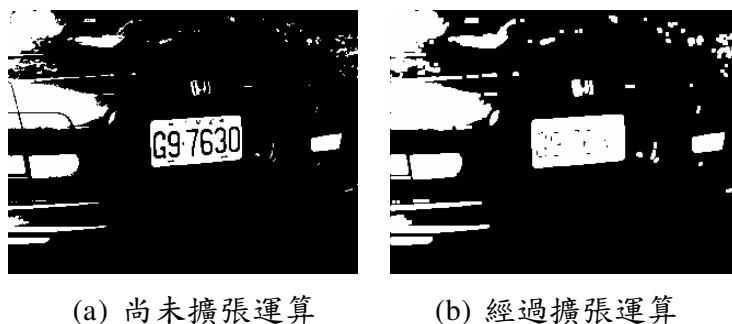


圖2.18 擴張運算前後比較

圖 2.18 為擴張前後比較圖，看圖中央的車牌部份，在二元圖裡面字元中間是皆是空心的區塊，則經過擴張運算後，幾乎都把車牌區塊補滿了。

(3) 斷開(Opening)

斷開運算為二元圖先經過一次侵蝕，再經過一次膨脹的組合運算；其通常是使物體輪廓平滑，消除細的峽部以及截斷窄的細頸。圖 2.19 為斷開運算的前後比較圖。很明顯細小的雜訊先經過濾除後，再使畫面膨脹至原大小並平滑之。



(a) 尚未斷開運算

(b) 經過斷開運算

圖2.19 斷開運算前後比較

(4) 閉合(Closing)

斷開運算為二元圖先經過一次膨脹，再經過一次侵蝕的組合運算；其功用也傾向於使輪廓部分平滑，不過與斷開運算相反的地方是，把窄的中斷部分和長細缺口連接起來，消除破洞，填補輪廓上缺口。圖 2.20 為閉合運算的前後比較圖形。觀看車牌字元部份，可以清楚知道閉和運算會先填補缺口，再使畫面趨於平滑。



(a) 尚未閉合運算

(b) 經過閉合運算

圖2.20 閉合運算前後比較

2.2 靜態系統相關之研究

靜態車牌定位系統的研究，可以從形態學開始談起，利用形態學達到車牌定位的方式，最簡單的作法則是利用 3×3 的遮罩，不斷的對原始影像

做侵蝕或膨脹的動作，最後再利用連通成分分析即可達到定位效果。不過，P.V. Suryanarayana 等人[5]提出修改型態學運算子的方式，設計獨特的結構化元素(SE)，利用一連串的形態學運算，如侵蝕、膨脹、開放、閉合等來完成車牌擷取的動作。此方法利用了字元的邊緣特徵，優點為不論車牌傾斜或是車牌破碎，都可以偵測到車牌；不過由於每次的型態學都是對整張圖像來做運算，所以致命缺點為運算時間過於冗長，而且當背景紋路 (Background texture) 太過於複雜時，則會產生過多的車牌候選區，進而導致偵測失敗。這個實驗的資料庫包含印度、中國、香港...等 7 個國家的車牌總共計 342 張，其正確偵測率為 95%。在其結論中提到，如果對不同國家車牌字元特徵，設計不一樣的結構化元素，則其車牌正確偵測率，可以達到 100 %，不過結論還是此方法的處理時間繁瑣，無法用於即時動態系統處理。

由於利用型態學來做車牌定位，容易產生過多的候選區而導致判斷失敗，因此，Premnath[4]提出了一個改良傳統型態學的方式，概念為考慮車牌附近的字元特徵，在經過二值化之後，並作水平投影，會有強烈的強度累積特性如圖 2.21。所以，其方法是先利用 Sobel 垂直邊緣偵測，藉此來濾除掉車牌周圍的強烈水平特徵後，再以獨特發表的強度累積式型態學 (Accumulative Intensity Morphing, AIM) 方法來保留車牌區塊，其優點為可以容忍 ± 30 度的角度偏斜問題。而實驗資料庫為 150 張泰國車牌測試影像，像素大小為 640x480 Pixels，包含了不同的拍攝視角與多種光線條件的影像，其正確偵測率高達 97%；缺點是當影像對比不夠強烈時，判斷會失敗，另一個缺點是 AIM 參數的設定限制，其 CCD 與車牌的拍攝距離只能容忍 $\pm 1\text{m}$ ，如果想要偵測大於或小於此距離的車牌，皆得再重新訓練新的一組對應參數，並重新設定之。

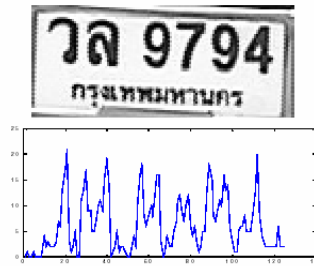
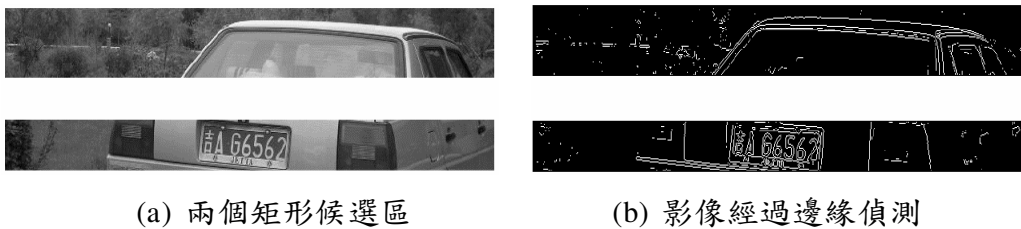


圖2.21 二值化車牌的水平投影[4]

上述之方法主要為考慮文字特徵，有一相似的方法為分析紋理的特徵，利用文字的紋理來達到車牌定位，其方法如 Jun Kong 等人[3]使用 Otsu 所提出的統計式全域二值化法對整張灰階圖像做二值化後，再使用「近似區塊搜尋分類法」(Approximate Region Searching and Classification, ARSC) 先將圖像進一步分割成矩形區塊。接著，利用 Canny 運算子做邊緣偵測，並考慮紋理特徵，將矩形區塊分類，找出符合條件之候選區。最後，使用 Radon 轉換來完成車牌的傾斜校正。此實驗資料庫為 640x480 Pixels 共 380 張彩色影像，背景包含人、車、建築物、樹木等複雜因子。成功偵測率為 96.1%。



(a) 兩個矩形候選區

(b) 影像經過邊緣偵測

圖2.22 使用紋理特徵尋找車牌[3]

圖 2.22(a)為近似區塊搜尋分類法所找出之矩形候選區，圖 2.22(b)為矩形候選區經過邊緣偵測的圖形。在此，考慮文字的密集邊緣紋理出現的特性，便能進一步的找出車牌候選區。

此外，如果考慮字元附近的邊緣會有密度較大的聚集特性的話，有如 Hongliang Bai 等人[1]提出之方法，為先利用 Sobel 的垂直邊緣偵測，產生了這張邊緣梯度圖(Edge Gradient Map)之後，考慮車牌字元附近的邊緣密度極大且聚集的特性，利用一個 15x3 的遮罩(Mask)來做平均梯度的迴旋積(Convolution)如公式 2-11，產生邊緣密度圖(Edge Density Map)。接著作 Otsu 二值化後，再使用連通成分分析法(Connected Component Analysis)來找出車牌後選區，這個資料庫的取材地點是高速公路收費站的近拍攝影圖像，圖片大小為 768x534 Pixels，在光線充足的條件下(夜晚仍有燈光照射)，478 張有 96%的正確率。

$$d(i, j) = \frac{1}{45} \sum_{x=-1}^{x=7} \sum_{y=-7}^{y=7} g_v(i+x, j+y) \quad (2-11)$$

$d(i, j)$ 為座標 (i, j) 本身及周遭共 45 個點之邊緣梯度平均值， $g_v(i, j)$ 為座標在 (i, j) 上之邊緣梯度值。

隔一年後，Hongliang Bai 等人[2]在的既有方法之下，加入了邊緣統計分析，把原先方法變為階級式(Hierarchical)的混合處理方法。依序統計線密度，保留線後，再計算矩形密度保留矩形，然後把這些矩形再搭配公式來做車牌候選區判斷。當既定的 4 個二值化門檻值(64、32、16、8)取交集還無法找出車牌後選區塊時，再跳入[1]所提出的型態學方法來找出車牌。此方法架構較大且複雜，實驗資料庫影像擴充為 9825 張，不過正確率高達 99.6%。

綜觀上述方法，大多有一致的特質，就是其作者皆犧牲運算時間來做高複雜度或多步驟的運算處理，等價交換極高的正確率。

2.3 動態系統相關研究

Sunghoon Kim 等人[22]提出之方法，其方法分為全域搜尋與局部搜尋兩大部分。而全域搜尋之方法為統計特徵，依序統計「梯度變異」、「邊緣密度」以及「密度變異數」，找出多個候選區後，並根據合併條件，將鄰近區塊結合，再接著進入局部搜尋的步驟。局部搜尋主要是拿一塊面板 (plate template) 反覆做邊緣修正，慢慢修正候選區之邊緣，再分別判斷區塊內是否含有車牌。此資料庫為 CCD 錄製動態影像，雖然正確率達到 90%，卻無提及系統處理時間。圖 2.23 為利用公式 2-12 來得到局部車牌影像內的梯度變化情形。

$$F_1\left(i + \frac{N-1}{2}, j\right) = \frac{\sum_{i=1}^N G_{i,j}^2}{N} - \left\{ \frac{\sum_{i=1}^N G_{i,j}}{N} \right\}^2 \quad (2-12)$$

$F_1(i, j)$ 為座標 (i, j) 之梯度差異值， $G_{i,j}$ 代表座標 (i, j) 的梯度值， N 為考慮變異的 Pixel 數量，遮罩大小於此為 1×9 ，所以 N 代入 9。

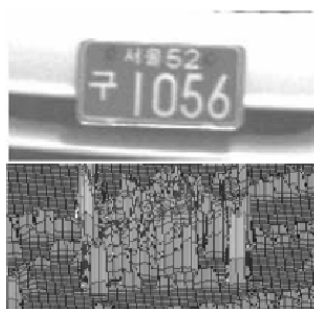


圖2.23 車牌影像內的梯度變異[22]

Seiki Yoshimori 等人[23]提出之方法，是以 $YCbCr$ 的色彩空間為系統為基礎，將兩張連續影像相減，可利用 Y 的閾值判斷出差異，找出位於高速

公路上的高速移動物體。再以實數編碼遺傳演算法(Real Coded Genetic Algorithm, RGA)辨識色彩的特徵，進而尋找車牌位置。此資料庫包含晴天及雨天的動態影像，在晴天的效果可達 91% 的正確率而雨天則降到 70.2%，系統平均正確率為 80.6%。雖然正確率不高，不過其系統處理每張圖片的時間為 180ms，合乎即時系統需求。

此外，考慮車輛在高速公路上高速行駛時，利用 CCD 取像會有模糊失焦的虞慮，如果要更換較佳的儀器設備無疑又是一項硬體成本，在不改變硬體配套措施下，R Bremananth 等人[24]利用分析二維的傅立葉頻譜(Fourier Spectrum)加上支持向量機(Support Vector Machine, SVM)來擷取移動物件，分離車輛與背景。接著，利用顏色相減會出現邊緣的特徵，經過 Sobel 的邊緣偵測後，再產生加權邊緣密度圖(Weight Based Density Map, WBDM)來達到偵測。其資料庫影像為不同背景、不同光線下之高速公路圖像，圖片解析度為 640×480 Pixels 共 1750 張。其正確率雖然高達 98.65%，但是每張圖片處理的時間為 3 秒，不夠符合動態「即時」系統之需求。

2.4 綜合分析比較

綜上每篇文獻的方法與結果來比較，很明顯的知道，想要系統提高正確率，最直接的作法，通常都是增加系統複雜度，做更多步驟的處理，讓運算結果越趨近於期望結果，只是如此一來，便會加長運算時間。為了加快系統速度，那就只有盡可能的降低系統之演算法複雜度，但是，降低運算複雜度或者減少處理步驟，往往都會導致美中不足的小缺憾，意即降低了系統正確偵測率。

因此，本文研究期望在動態影像系統上，建構一套演算法複雜度不高，但是又能夠提升正確率的即時動態車牌偵測。在下個章節，將會詳細的介紹本文所採用之車牌定位方法。