可攜式車牌辨識系統之演算法研究

A Study of Algorithms for Handheld License Plate Recognition System

楊仁華
國立臺灣海洋大學
電機工程研究所
M96530059@mail.ntou.edu.tw

呂紹偉
國立臺灣海洋大學
電機工程研究所
B0119@mail.ntou.edu.tw

詹景裕
國立台北大學
電機工程研究所
gejan@mail.ntpu.edu.tw

李政宏
國立臺灣海洋大學
電機工程研究所
D93530006@mail.ntou.edu.tw

摘要—車牌辨識系統經過多年發展，目前已有不少商業應用，但主要是以定置型電腦為計算平台，如停車場出入管理等相關應用。本文提出一種適於可攜式影像擷取裝置之車牌辨識演算法，使得車牌辨識的實現更有彈性。

本文所提出之車牌辨識演算法實作於定點式數位訊號處理器上，包含車牌定位、字元切割及字元辨識等三個階段。車牌定位步驟利用影像輪廓及亮度特徵來進行定位，並利用影像縮圖提升計算速度；字元切割步驟則先以複合式車牌水平度校正方法進行處理，提升切割及後續步驟的正確率；字元辨識步驟使用階層式單層類神經網路來進行辨識，藉此解決相似字型辨識問題。

本研究在德州儀器 TMDXVDP6437 DSP 開發板上實作本文所提出之車牌辨識(License Plate Recognition, LPR)演算法。本文基於 539 張影像所組成的樣本空間，其中每張影像均包含一個或一個以上的車牌，採用不同的拍攝角度，並且在不同的天氣及亮度條件之下，平均辨識率為 87.2%且單一車牌執行時間約兩秒。

關鍵詞—可攜式裝置、車牌辨識、影像處理

Abstract—Automatic license plate recognition has attracted many researchers' attention in recent years, and their efforts have resulted in many installed business applications. However, most existing installations are non-moving, such as automatic parking control and payment systems. In order to provide flexibility for implementation, this article develops a set of algorithms suitable for realizing license plate recognition (LPR) capability on handheld mobile devices.

The proposed LPR algorithm consists of three major steps, namely, license plate location, character segmentation, and character recognition. Plate location is achieved mainly by extracting image outlines of probable areas. For pictures of acceptable quality, two-dimensionally reduced images are used to save running time. To separate the characters with higher accuracy, the character segmentation step applies the techniques of hybrid rotation correction. Finally, a one-layer artificial neural network is deployed for character recognition.

We have implemented the proposed LPR algorithm on a DSP-based development board. The experimental results show that, the total average rate of successful recognition is 87.2% within two seconds. This is based on a sample space of 539 pictures including the ones containing more than a single license plate, taken with various shooting angles and under a range of weather and light conditions.

Keywords: Image processing, license plate recognition, handheld device.

一、簡介

車牌影像辨識的研究最近十年來相當受到矚目，因為車牌影像辨識對於停車管理、交通執法、以及贓車追蹤等方面，具有很高的實用價值。一般車牌辨識系統的應用以停車場所車輛進出管理與收費為大宗，此類場合大都將攝影機置於固定位置，以固定的角度拍攝進出車輛之車牌影像，搭配的計算平台也大多屬於定置型的電腦系統。本文考量員警值勤時之機動性、操作方便性、以及使用場所多樣化的需求，試圖在嵌入式系統上建置一執行迅速、辨識正確率高之車牌影像辨識系統。一般在需要取得或紀錄車輛資訊時，值勤員警大多使用類似 PDA 的車牌資料查核裝置，以手動方式輸入車牌字串。若能以自動影像辨識取代手動輸入，將可大幅提升值勤效率。但由於可攜式裝置的硬體資源較為有限，為了符合執行速度的要求，必須使用運算過程較為精簡的影像辨識演算法，並且在執行速度與辨識正確率之間要做一些取捨。因此，本文的目的即在設計及整合一套適用於可攜式裝置的車牌影

像辨識演算法，並且呈現我們的實驗結果。

車牌辨識系統主要可分為三個步驟，依序為車牌定位步驟、字元切割步驟、以及字元辨識步驟。車牌辨識研究方面，近年國內外均有相關論文針對演算法做出各種改進。從車牌辨識演算法的層面來看，自動化車牌定位、切割及辨識等動作的目標為找出車牌字元的特徵，再利用該特徵來達成自動化的目的。利用人工建置方式選取車牌字元特徵的研究中，Anagnostopoulos 等學者使用車牌固定範圍內的劇烈的灰階值變化，或是利用車牌影像的顏色數值統計分配做為車牌字元的特徵[1、4、5、6、9、13]。在量化上述特徵的方式上，亦有各種不同的方式被選用，例如，王中山使用離散型小波轉換來取得灰階數值轉換的高頻區段[1]，Anagnostopoulos 利用區域影像灰階值統計分配設計了一 Sliding Concentric Windows (SCWs)方法來突顯出車牌影像區域的特徵[9]。用機器學習的方式選取車牌字元特徵的研究中，車牌字元特徵的選取亦可利用非線性的計算方式自動產生車牌特徵，經由樣本訓練後，直接使用訓練後的結果。許多論文使用各種不同類型的類神經網路來進行辨識[5、9、11、16]，或是利用其他影像特徵值抽取演算法來訓練出車牌定位用的特徵[7、8]。

而在系統實作方式上，亦可選用不同的硬體平台。例如在嵌入式系統的應用上，Arth 等學者利用 DSP 處理器或其他類型嵌入式系統來實作車牌辨識系統[2、5、7、8、10、15]。為了加速計算效能考量，部分論文亦利用 DSP 處理並配合 FPGA 將部分計算功能硬體化[11、12]。綜合上述，隨著硬體技術的進步，各種複雜的運算功能將更適於各種可攜式裝置應用上，例如智慧型手機、DV、或數位相機等等，可攜式裝置相關應用的蓬勃發展更促成本文的研究動機。

以下為本文內容提要：第一節對車牌辨識研究進行簡介；第二節描述車牌影像定位步驟所使用的理論及方法；第三節闡述車牌字元切割以及傾斜度校正的處理流程；字元辨識步驟與相似字處理於第四節說明；第五節介紹實驗方法、所使用的軟硬體資源、以及實驗結果；第六節為結論及未來展望。

二、車牌影像定位

2.1 車牌特徵擷取

一般而言，在一幅灰階影像中，車牌所在區域的灰階值變化比其他區域劇烈。藉由輪廓擷取運算所計算出的輪廓強度值，即可量化影像中每個座標所對應的灰階值變化。找出影像的輪廓線後，便可框選出輪廓線相對密集的區域，以做為車牌影像的候選區域。

2.1.1 影像輪廓擷取

本文選用 Prewitt 法來進行輪廓擷取[3]。Prewitt 法是一種常見的輪廓擷取運算，針對二維影像的每一個點座標，利用代表八個不同方向的水平微分遮罩，分別計算出八個方向的灰階值差值，再以八個差值中的最大值代表該點座標的輪廓強度值。但對車牌字元而言，我們無須對八個方向灰階變化做比較，而是只取由左到右，或由右到左兩個方向的灰階值變化。輪廓擷取運算完成後，再進行二值化運算。

2.1.2 輪廓密集度判定所用之擴張運算

此步驟的目的是將經由前述步驟處理之影像中，白色像素相對密集處加以連結，而相對稀疏處則予以消除並更改為黑色，使得白色處連結成一可能的車牌所在區域，在此稱為圈選遮罩。此步驟類似型態學中的擴張運算，用以決定輪廓線的疏密程度。以下是本文作為擴張運算用之水平密集度判定函式流程，其中 w 為使用者用以設定寬度之參數：

Function 1: 水平密集度判定擴張運算函式
Begin
- Step 1: 待輸出影像全部設為黑色像素。
- Step 2: 由左上到右下掃描整張輸入影像，碰到白色像素則進入 Step 3，整張影像掃描完則函式執行結束。
- Step 3: 若白色像素座標為 (x, y)，則由左到右，從 (x, y) 到 $(x+w-1, y)$ 座標之間的範圍內，找尋下一個為白色像素的點座標 $(x+t, y)$，即 $t < w-1$。若在上述

範圍內找到另一白色像素，則進入 step 4；否則回到 step 2。

Step 4: 將輸出影像的(x, y)至$(x+t, y)$範圍內的點全部標記為白色，並回到 Step 2。

End

上述輪廓密集度判定的寬度參數 w，其值的選定與畫面中車牌的遠近有關聯。過遠的車牌原本即可能因為車牌字元太小而影響辨識率，而在寬度參數被選定的狀況下，過近的車牌卻也可能因為輪廓線較為稀疏，使得整個車牌區域無法連結起來。

2.1.3 輪廓密集度判定用侵蝕運算

本文所設計之侵蝕運算目的是清除雜訊連結以分離出車牌可能所在區域，根據影像掃描方向可分為水平及垂直兩種，需設定之參數包含寬度與高度兩種。圖 1 為實際執行結果，其中(b)為經過水平擴張及侵蝕運算後的結果。

水平侵蝕運算的目標是將未達所設定寬度參數值的白色區塊更改為黑色，以減少圈選遮罩與其上下方背景雜訊連結之機率，例如機車車牌與下方擋泥板區域。寬度限制侵蝕運算流程如下，其中 w 為使用者所設定的寬度參數：

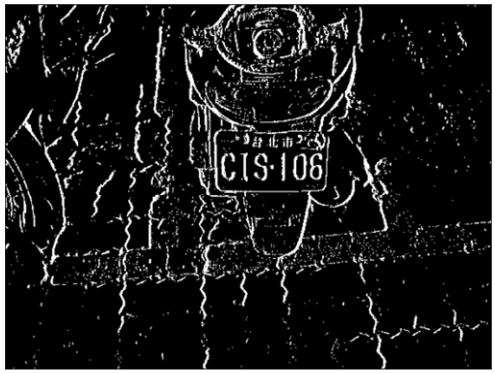
Function 2: 水平密集度判定侵蝕運算函式

Begin

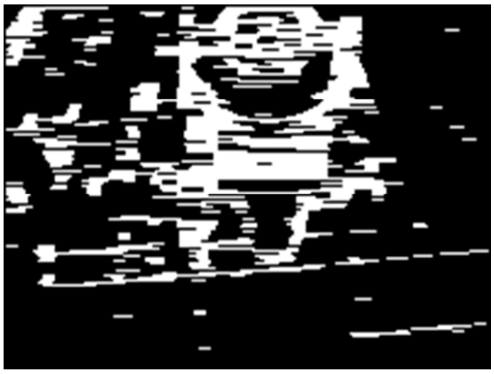
Step 1: 將輸入之影像複製到待輸出影像中。

Step 2: 由左上到右下掃描整張輸入影像，碰到白色像素則進入 Step 3，整張影像掃描完則函式結束。

Step 3: 若白色像素座標為(x, y)，則由左到右，從(x, y)到$(x+w-1, y)$座標之間的範圍內，確認範圍內是否全為白色像素。若點座標$(x+t, y)$在上述範圍



(a)Prewitt輪廓擷取及二值化運算



(b)水平擴張及侵蝕運算

(c)區隔亮度特徵

(d)重複執行水平擴張及侵蝕運算

圖 1 車牌影像特徵擷取結果。

內且為黑色像素，則進入 Step 4；若在範圍內全為白色像素，則回到 Step 2。

Step 4: 將輸出影像的 (x, y) 至 $(x+t, y)$ 範圍內的點全部標記為黑色，並回到 Step 2。

End

垂直侵蝕運算流程需設定高度參數，用以判斷圈選遮罩中垂直走向的白色直線是否超過所設定的高度，沒有超過則將該線段加以侵蝕變為黑色。所使用的高度參數，亦以可能的車牌字元高度的做為設定依據，本文所設定的高度參數為 10 個像素長。該高度限制侵蝕運算流程如下，其中 h 為使用者所設定的高度參數：

Function 3: 垂直密集度判定侵蝕運算函式
Begin

Step 1: 將輸出影像全設為黑色像素。

Step 2: 由左上到右下掃描整張輸入影像，碰到白色像素則進入 Step 3，整張影像掃描完則函式結束。

Step 3: 若白色像素座標為 (x, y)，由上到下，從 (x, y) 到 $(x, y+h-1)$ 座標之間的範圍內，確認範圍內是否全為白色像素。若點座標 $(x, y+t)$ 在上述範圍內且為黑色像素，則進入 Step 4；若在範圍內全為白色像素，則回到 Step 2。

Step 4: 將輸出影像的 (x, y) 至 $(x, y+t)$ 範圍內的點全部標記為黑色，並回到 Step 2。

End

2.1.4 灰階影像二值化

灰階影像可利用影像的二值化運算轉換為黑白兩色影像，作為亮度特徵的分割動作。選取一適當臨界值，當影像灰階值小於該臨界值者則設為黑色，反之設為白色，而臨界值的選取方式，本文使用 Sauvola 法最為二值化函式之自動臨界值選取方法[9]，並根據用途進行調整。

Sauvola 法的臨界值選取，是以所有像素的灰階值計算出平均數及變異數，再利用這兩筆統計資料的線性組合來得到臨界值。Sauvola 法會先對目標圖片以正方形區塊進行切割，每個區塊具有區域性的統計資料，利用各區塊的統計資料可以求得對應於像素 (x, y) 的臨界值 $T(x, y)$，如式 (1) 所示：

$$T(x, y) = m(x, y) + \left[1 + k\left(\frac{\sigma(x, y)}{R} - 1\right)\right] \quad (1)$$

其中 $m(x, y)$ 表示對應座標 (x, y) 的局部平均數；$\sigma(x, y)$ 表示對應座標 (x, y) 的局部變異數；k 與 R 為常數，其值各為 0.5 與 128；區塊大小為 10×10 pixels。

2.2 階段式車牌影像定位

本文車牌定位步驟分兩階段進行定位，第一階段快速地刪除非車牌的區域，第二階段再進行細部掃描，以濾除更多非車牌區域的背景雜訊。

2.2.1 縮圖定位簡介

為車牌定位或其他步驟的需要，至少需掃描整張影像一次。假設輸入影像大小為 600×450 pixels，再假設影像中車牌區域大小為 100×20 pixels，雖然車牌區域相對甚小於整張影像，但車牌定位步驟還是必須對整張影像進行處理，因而耗費大部分的執行時間，因此車牌定位步驟是車牌辨識整體計算效能的瓶頸所在。

經由實測得知，針對本論文所設定之距離範圍內的車牌進行定位，只需要車牌的輪廓線密集度或亮度特徵即可，原影像的縮圖便足以進行車牌定位動作，達成縮短執行時間的目的。由縮圖所計算出之車牌位置座標，再縮圖的縮放倍率映射回原圖來抓取車牌影像，藉此達到加速車牌定位步驟的目的。

2.2.2 車牌影像定位第一階段

本文第一階段是以原影像的縮圖來進行定位。首先我們對縮圖灰階影像進行 Prewitt 輪廓擷取運算，輪廓密集度之水平擴張及侵蝕運算來產生車牌候選區域。圈選遮罩中的白色區域，可利用灰階影像之亮度特徵再進行一次濾除動

作。由於此時的圈選遮罩圈包含車牌與背景雜訊，而車牌區域與背景雜訊的亮度組成分配將有所差異，因此可利用影像二值化運算將其再加以區分。以下略述利用亮度特徵區隔的流程：首先取得待處理的圈選遮罩與原縮圖灰階影像，圈選遮罩白色區域像素值設為 1，黑色區域設為 0。將待輸出影像像素設為對應點座標的原縮圖影像與圈選遮罩像素值乘積。接著對待輸出影像進行 Sauvola 法二值化動作，即產生處理完畢之圈選遮罩，經過處理後之影像可參考圖 1(c)。

經過上述亮度區隔處理，再利用前面所述水平擴張及侵蝕運算修補圈選遮罩後，即可利用圈遮罩來進行影像切割的動作經過處理後之影像可參考圖 1(d)。於本實驗中，使用的圖片大小為 600×450 pixels，而用以定位的縮圖大小為 200×150 pixels，所以由縮圖所得之圈選遮罩矩形區域座標可直接乘上縮放倍率後，即可用調整後的新座標從原圖中切割出車牌候選影像，並利用影像儲存串列加以儲存。

2.2.3　車牌影像定位第二階段

第二階段的目標是針對由影像儲存串列所存放之車牌候選影像進行再切割動作，目標是將仍與車牌區域相連的背景雜訊加以排除。第一階段所產生的影像儲存串列的各節點，亦儲存了各子影像的長寬等訊息，此時可利用該訊息將過小的選取區域先行濾除。第二階段車牌影像定位與第一階段差別在於作用對象更改為每一張車牌候選影像。先對縮圖灰階影像進行 Prewitt 輪廓擷取運算，再利用水平擴張運算產生車牌候選區域。由於此時非縮圖，被定位的車牌影像由人眼觀察已可看出明顯的車牌字元，所擷取出之輪廓在一般狀況下應可呈現出字元的外觀與緊密的排列方式，所以輪廓密集度判定便可利用字元與字元之間距離的統計資料做為設定寬度參數的依據，本文所設定的寬度參數為 20 個像素長。經過水平擴張運算後，便可依實際字元可能高度設定高度參數，進行垂直侵蝕運算，產生第二階段定位用遮罩。

若經過上述檢測後，仍包含一個以上的白色區塊，由於車牌候選影像中幾乎不會出現同時包含兩個車牌的狀況，所以無條件保留面積最大的白色區塊當作最後的圈選遮罩。圖 2 為本步驟實際定位過程結果示意圖。

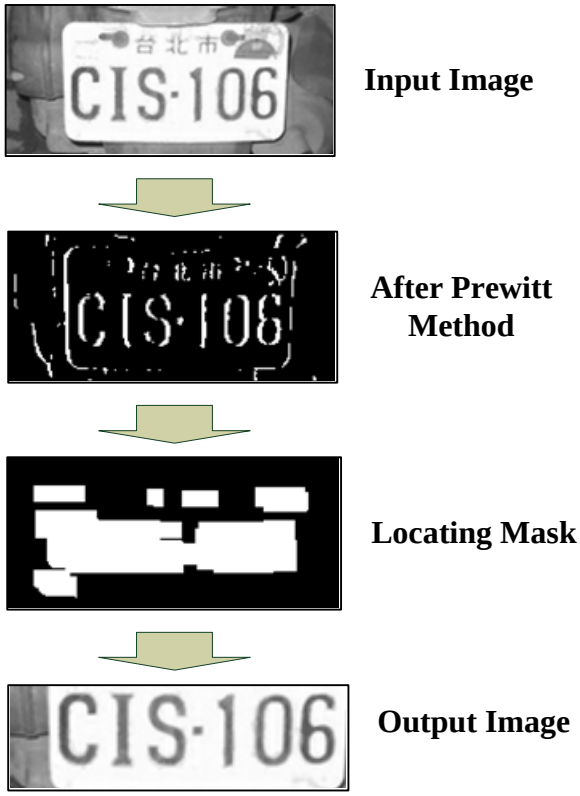


圖 2 車牌定位步驟第二階段結果示意圖。

三、車牌字元切割

3.1　複合式傾斜度校正

3.1.1 字元連接物件及傾斜角度取得方式

在取得車牌候選影像後，首要工作便是取得字元連接物件。取得字元連接物件的資訊可用於各種操作之中。字元連接物件步驟流程如下：先將影像利用 Sauvola 法進行二值化，再對二值化影像以八連結的方式進行連接物件編碼。上述流程完畢後，利用座標記錄串列結構來記錄各別連接物件相關資訊。

字元連接物件擷取與車牌定位類似，均為針對輸入影像中找出目標物件，但除了目標物件外還包含許多雜訊，所以需要附加篩選動作。取得字元連接物件之座標記錄串列。字元連接物件之座標記錄串列記錄了字元的相關資訊，每個節點所儲存的字元資訊如表 1 所示，利用節點所存放

表 1 座標記錄串列節點儲存變數說明

串列儲存變數	說明
id_value	連接物件編號
max_y	連接物件下端點
min_y	連接物件上端點
max_x	連接物件右端點
min_x	連件物件左端點

的資訊，便可配合統計資料進行連接物件篩選，本文所使用的連接物件篩選檢測步驟有高度及寬度檢測、長寬比檢測、密度檢測。高度及寬度檢測可將過小的雜訊加以刪除，變數請參閱表 1。長寬比或是寬長比可用來檢測車牌影像或字元影像，連接物件的高度 *height*、寬度 *width*、以及長寬比 *aspect_ratio* 以下列方式計算：

$$height = max\_y - min\_y + 1 \quad (2)$$

$$width = max\_x - min\_x + 1 \quad (3)$$

$$aspect\_ratio = \frac{height \times 100}{width} \quad (4)$$

上式在計算時乘上固定大小的常數，可使檢測比例以整數來進行運算而不造成捨位後的過度誤差。密度是表示二值化影像中，單一連接物件上下左右端點的矩形範圍內，連接物件面積占整個矩形面積的比例。連接物件的密度 *density* 以下列方式計算：

$$density = \frac{weight \times 100}{height \times width} \quad (5)$$

表 2 為針對字元連接物件所設定的參數，針對字元的高度、寬度以及密度（在此指端點所成之矩形內所占面積比）來加以區分。特別需要注意的是，「I」及「1」字元所使用的密度篩選判定參數與其他類型字元不同，須先將其利用寬度與高度比例來區隔。篩選過程中，符合下列參數之連接物件加以保留，反之則予以刪除。

符合表 2 條件的字元連接物件，本文使用一「高度中位數篩選」之篩選方式對其再進一步篩

表 2　字元連接物件篩選用參數列表

最大高度	(無限制)	
最小高度	10 pixels	
最大寬度	影像寬度/4	
最小寬度	2 pixels	
最大寬高比(寬/高)	75%	
最小寬高比(寬/高)	15%	
I 及 1 字元及其他字元寬高比分界	40%	
最大密度	I 或 1 字元	非 I 且非 1 字元
	65%	85%
最小密度	(無限制)	

選，其步驟如圖 3 所示。非字元連接物件可能因為恰好符合高度、寬度及密度篩選條件，而連同其他字元連接物件一起被保留。然而，根據實際測試，經過上述刪除方式，所保留下來之連接物件絕大部分為實際的字元區域，這些字元連接物件將具有十分相近的高度參數。由於字元具有相近的高度範圍，非字元者便可能不具相同的高度範圍。此時對剩餘連接物件以高度來進行排序，取其排序後之中位數加上適當的誤差值，即可產生新的篩選範圍，本文將該誤差值依實驗測試結果定為 3 pixels。以上為字元連接物件的擷取與篩選，取得字元座標記錄串列後便可繼續進行後續相關操作。

本文取得傾斜角度的方式，在於取得字元連接物件的同時，根據座標記錄串列的資料，亦取得每個字元的端點座標，因此可算出代表每個字元的中心座標。如圖 4 所示，找出靠近兩端的字元中心座標，便可依此算出傾斜角度。

3.1.2　傾斜度校正運算

為了使字元切割步驟成功率提高，必需對車牌候選影像進行水平度校正運算。水平度校正使用兩種影像處理運算，分別是影像旋轉及影像平移，如圖 4 所示，兩種操作均是以車牌候選影像的中點座標當作旋轉中心，其中 *w_rotation* 表示左右端點兩側字元連接物件中心的水平距離，*h_rotation* 則表示垂直距離。由於本程式數值變

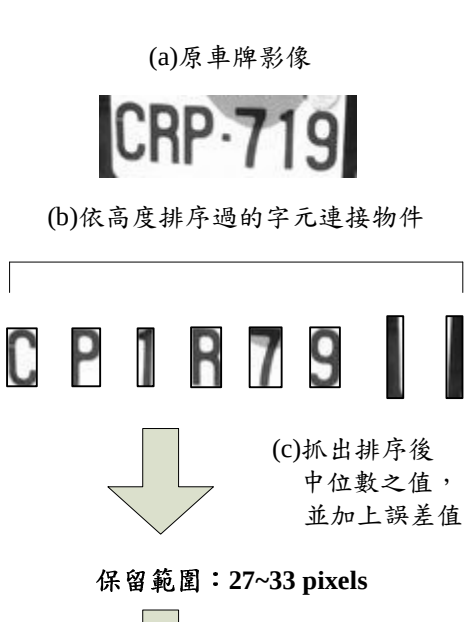


圖 3 高度中位數篩選流程示意圖。

數均整數型態，角度相關三角函數運算亦以 *w_rotation* 及 *h_rotation* 兩變數來表示。在影像旋轉運算中，影像所進行的旋轉運算會以輸入影像中心作為軸心進行旋轉動作，若 (x,y) 為輸入影像中任一點點座標，(x',y') 為輸出影像中對應 (x,y) 的座標，表示輸入影像中 (x,y) 處之灰階值等於輸出影像中 (x',y') 處之灰階值。事實上，由於以整數進行運算的緣故，將造成部分 (x',y') 座標無法取得對應座標灰階值，所以必須將上述算式進行反運算，以 (x',y') 座標來找出對應的 (x,y) 座標及其灰階值，再利用 *w_rotation* 及 *h_rotation* 兩變數代換後得到座標轉換後的整數逼近值，座標轉換算式如下所示：

$$y = \frac{x' \times h\_rotation + y' \times w\_rotation}{\sqrt{w\_rotation^2 + h\_rotation^2}} \quad (6)$$

$$x = \frac{x' \times w\_rotation - y' \times h\_rotation}{\sqrt{w\_rotation^2 + h\_rotation^2}} \quad (7)$$

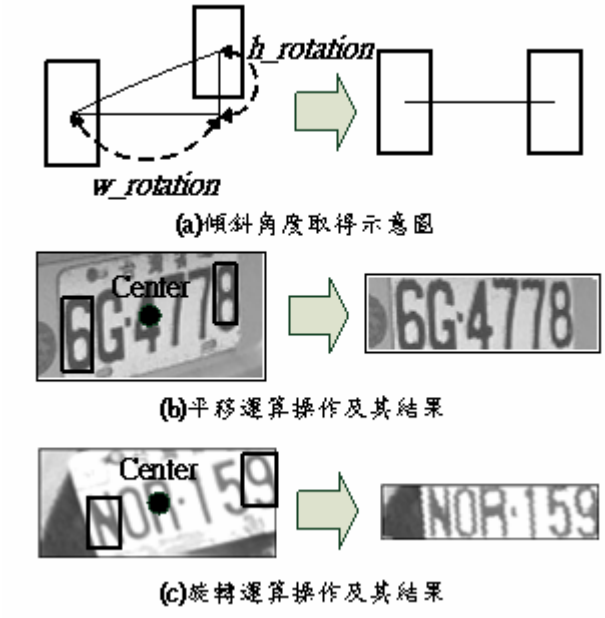


圖 4　傾斜度校正運算示意圖。

在影像平移運算中，影像所進行的上下平移運算亦如同影像旋轉運作一般，以輸入影像中心 (x_c,y_c) 作為軸心，進行平移運算，且輸入影像中 (x,y) 處之灰階值等於輸出影像中 (x',y') 處之灰階值。如同旋轉運算，部分 (x',y') 座標無法取得對應座標灰階值，算式將以 (x',y') 座標來找出對應的 (x,y) 座標及其灰階值，其中的座標轉換算式如下所示：

$$y = y' + \frac{(x' - x_c) \times h\_rotation}{w\_rotation} \quad (8)$$

$$x = x' \quad (9)$$

3.1.3　複合式傾斜度校正流程

由於本文系統欲建置於可攜式裝置上，可能會擷取到不同角度下所拍攝到的車牌影像，由圖 6 可知依照機車的不同的停放方式亦可能拍攝到的傾斜車牌，所以本文使用了影像旋轉與影像平移兩種不同的調整函式來處理不同角度的車牌影像，影像旋轉與影像平移各有其適用狀況，所以必須依影像狀況選用適合的調整函式。

本文提出一複合式水平度校正機制，對車牌影像 I 分別進行影像旋轉與影像平移函式後，產生兩個不同結果的輸出影像 I_1 及 I_2，I_1 及 I_2 分別

進行字元切割步驟及字元辨識步驟。字元切割步驟完成後分別會產生對應的鍵值 key_1 及 key_2，該鍵值表示字元切割步驟成功與否，成功鍵值為 1，失敗鍵值為 0。字元辨識步驟完成後分別會產生對應的權 $value_1$ 及 $value_2$，以及辨識後的結果 $string_1$ 及 $string_2$。若 $key_1 \times value_1 \geq key_2 \times value_2$，則採用 I_1 的辨識結果 $string_1$；反之，則採用 I_2 的辨識結果 $string_2$。當傾斜角度大的車牌使用了不適宜的水平度校正函式，往往會造成切割失敗，得到的鍵值為 0，因此可用此法自動選取出適宜的校正方式。而在產生鍵值均為 1 的狀況下，亦可由權值來比較不同的校正結果間，與字元比對陣列的相似程度為何，以取得相似程度最高的辨識結果。

3.2 字元切割步驟

在字元切割步驟流程中，車牌影像經過 Sauvola 法二值化後，再利用此二值化影像產生字元分割遮罩，最後利用字元分割遮罩來定位出原車牌影像中每個字元的位置，切割出各別車牌字元，並利用影像記錄串列加以儲存。

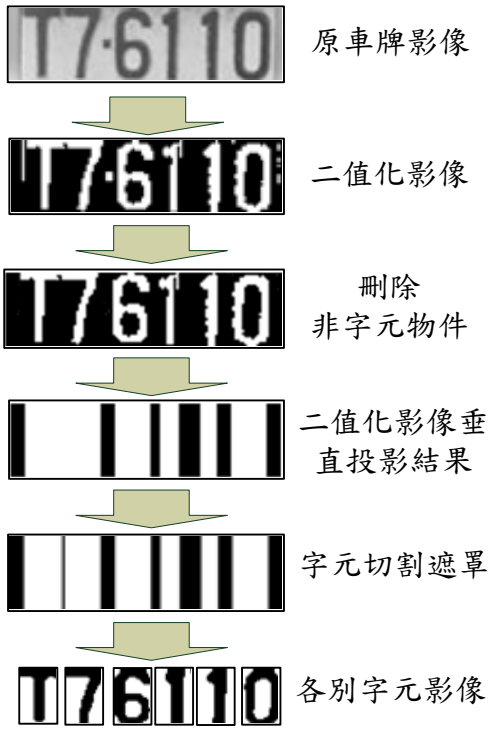


圖 5 垂直投影字元切割遮罩作用示意圖。

字元分割遮罩產生的方式有兩種，第一種方式是直接利用字元連接物件的座標產生字元分割遮罩，來進行字元分割。缺點如前述，對於環境狀況不佳的車牌樣本，其切割失敗率高，但在一般環境良好，車牌影像清晰無陰影的狀況，或是複合式傾斜度校正無法得涵蓋的狀況下，如此切割方式仍有其必要性。第二種方式是二值化影像垂直投影，由圖 5 可知，影像二值化後，影像陣列中垂直方向上的白色像素多寡來決定切割的位置。當車牌影像經過傾斜度校正處理後，取得車牌字元高度，因此可界定出車牌影像中的新的上下邊界。

令垂直投影後各行的白色像素數量為一維陣列 $A[i]$，一維陣列 $B[i]$用以記錄分割界線，若為 0 則表示該行 i 座標處為界線，若為 2 則表示該行 i 座標處為字元區域。下列步驟可裁定出字元界線，其中 $xsize$ 為車牌影像寬度，$ysize$ 為車牌影像高度：

Function 4: 字元分割遮罩產生函式
Begin
Step 1: i 由 0 到 $xsize-1$，將 $B[i]$全設為 1。
Step 2: 找出 $A[i]$的最小值 $global\_min$，並且當 $A[i]=global\_min$ 時，令 $B[i]$為 0。
Step 3: i 由 0 到 $xsize-1$，量測 $B[i]$為 1 處的連續區間區域 (l,r)，其寬度為 $r-l+1$，若 $r-l+1 \geq \frac{3ysize}{4}$，則跳到 Step 4；反之，則將區間 (l,r) 處之 $B[i]$ 均設為 2，繼續往下掃描。當 $B[i]$ 上沒有任何值為 1 時則函式結束。
Step 4: 於區間 (l,r) 中，擷取位於中心的 $\left(l+\frac{(r-l+1)}{3}, r-\frac{(r-l+1)}{3}\right)$ 區間，找出 $A[i]$於此區間內的區域最小值 $local\_min$，並且當 $A[i]=local\_min$ 時，令 $B[i]$為 0。並回到 Step 3。
End

上述方式依實驗測試結果定為字元高度的三分之四，作為寬度區間是否再進行細分的依據，此比例亦是經測試後，車牌字元適宜的長寬

比範圍。當寬度區間大於字元高度的三分之四時，則表示此區間包含兩個以上的車牌字元，車牌字元相互連接處靠近區間中心，所以細分區間時會盡可能從中點進行切分，故從原區間內中心三分之一寬的範圍內找尋新切割點。經過上述步驟後即可取得字元切割用遮罩，即使車牌影像因環境因素造成部分車牌字元相連，此法仍可依字元相對位置關係來加以區分。

本文車牌以六字車牌座為主要目標，程式實作上會自動判斷字元切割遮罩所選取的字數。若未滿六字，則代表被定位影像非車牌區域，或是因環境因素造成字元切割失敗。若超過六字，則代表被定位的影像選取範圍過大，背景區域被包含進來，此時可斷定左右兩側的字元應該為背景雜訊，應予以刪除。

刪除背景雜訊的規則，亦類似水平度校正，字串最左邊及最右邊兩側單一字元辨識步驟完成後，分別會產生對應的權值，比較兩權值大小後，刪去權值較小之字元，並重複上述動作直到車牌字元剩餘六字為止。然而，上述刪字動作仍有誤刪的可能性，造成字元切割失敗，字元切割是否正確還是必須依靠車牌定位圈選適當的範圍。

四、車牌字元辨識

4.1 字元辨識步驟流程簡介

本文車牌字元辨識方法使用單層類神經網路來進行辨識[5]。每個字元用一個 20×30 大小的整數陣列作為辨識計算依據，字元「0」與「O」暫時當做同一字元，共需 35 個整數比對陣列。經類神經網路訓練期後，整數陣列上每個位置經訓練後產生不同的數值權重。字元所在處為正整數，非字元處為負整數。在類神經網路回憶期，字元影像進行二值化以及正規化為 20×30 pixels 之二值化影像，該影像對應一計算用字元陣列 $letter_{ij}$，其中字元處數值為 1，非字元處為-1。將此字元陣列各別與 35 個比對陣列進行運算，比對陣列以 $f_{ij}(k)$ 表示，其中 $0 \le i \le 30$，$0 \le j \le 20$，k 用以區分比對陣列順序，所以 $0 \le k \le 34$。每一個比對陣列經計算會產生權值 $weight(k)$，其計算方式如下：

$$weight(k) = \sum_{i=1}^{30} \sum_{j=1}^{20} letter_{ij} \cdot f_{ij}(k) \qquad (10)$$

字元辨識結果 *result* 可以下列式子表示：

$$result = \{k \mid \max weight(k), 0 \le k \le 34\} \qquad (11)$$

result 之值亦落在 0 到 34 之間，表示不同的辨識結果。由上述運算便可完成字元辨識動作，在配合對應車牌字串規則的相似字元校正後，將使辨識流程更趨於完備。

4.2 相似字元調整

字元辨識中的相似字元調整一直是亟待克服的難題之一，辨識失敗的主因也在此處。字元組合「0」與「D」、「1」與「I」、「8」與「B」是最常發生相似字辨識問題的組合。本文於辨識字元完成後，將會附加上一到檢查機制。當辨識完成產生車牌字元時，會對結果進行確認，若其辨識結果屬於字元組合「0」與「D」、「1」與「I」、「8」與「B」其中一種，則重新進行相似字辨識並修改辨識結果；反之，若非上述相似字組合，則直接送出辨識結果。相似字辨識則針對上述三種相似字組合建立各別的單層類神經網路架構。

利用台灣車牌字串命名規則，亦可進行相似字校正。參見圖 6，可利用字元連接物件或字元

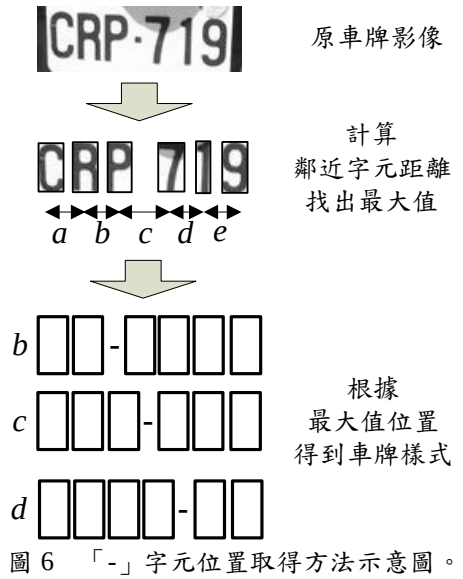


圖 6 「-」字元位置取得方法示意圖。

切割遮罩來取得「-」位置後，便可依此來畫分車牌樣式。舉例來說，六字汽車車牌「-」可區分出二字與四字區段，其中四字區段應該全為數字，所以四字區段處比對陣列應可利用純數字組成的比對陣列進行辨識。

五、實驗結果

本文程式以 C 語言實作，於德州儀器 TMDXVDP6437 DSP 實驗板上運作，該實驗板所使用處理器為德州儀器 DM6437 處理器，頻率可達 600 Mhz [15]。所使用樣本為 600×450 pixels 圖片，縮圖定位架構所使用的縮放倍率為圖片長與寬的三分之一，即 200×150 pixels 圖片。車輛與鏡頭距離在 1 到 8 公尺之間，包含了白天與夜間各種環境下，共 539 個車牌影像。距離範圍內之車牌影像於畫面中出現，且經辨識得到正確的車牌字串即視為成功，字串中有任一字元錯誤視為辨識失敗。

本實驗車牌定位步驟成功之定義為：在符合前一節所述條件之樣本集合範圍內，單一車牌經定位處理後，能夠產生出一車牌候選影像，並可圈選住六個車牌字元。換句話說，是以車牌為基本單位，而非樣本影像。舉例來說，以一張樣本影像出現三個車牌的狀況下，若圈選出兩個車牌，則代表有兩個成功案例，一個失敗案例。本實驗車牌定位步驟成功率為 99.07%。本文之車牌定位步驟，為符合可攜式裝置所拍攝的樣本條件，可於單張樣本影像中，抓取一個以上的車牌，如圖 7 所示。而失敗的案例中，部分案例原因在於車牌邊框裁剪步驟切割到邊緣字元，導致車牌定位區域不完整，事實上，本文所採取之邊框裁剪函式切割條件較為嚴格，以避免圈選過多背景區域，因此對整體的辨識效能有所助益。另外，其他定位失敗的案例多為環境因素所致，例如車牌反光、車牌表面光線不均勻等等。另外，從圖 8 可得知，本文程式有可能因為背景過於複雜，而抓取到背景部分區塊。由於本程式可以抓取一個以上的車牌候選影像，若於其中能抓取到樣本影像中的，以本文定義來說即表示成功。

本實驗字元切割步驟成功率為 98.88%。樣本影像於車牌定位成功的狀況下，才會進行字元切割步驟。本文所使用的車牌傾斜度校正函式，可針對不同拍攝角度進行調整。如圖 8 所示，機車車牌呈現不同的傾斜角度，傾斜度校正函式便可分別對其調整。

(a)車牌影像樣本

(b)定位步驟產生之車牌影像

(c)經傾斜度校正調整過後之車牌影像

圖 7　本實驗字元切割步驟傾斜度校正案例。

然而上述動作仍有無法作用的情形。在失敗的案例中，有些車牌處於無法調整的傾斜角度，而造成切割失敗，而有些定位範圍過大的狀況下，仍無法順利的挑選出正確的車牌字元。另外，在部分惡劣的車牌環境下，比方說車牌字元汙損不清晰，也會造成字元切割步驟失敗。

字元辨識時，會先將「O」與「0」視為同一字元，等辨識結果出來再依車牌編列規則來區隔。樣本影像於字元切割步驟成功的狀況下，進行字元辨識步驟才有意義，字元切割步驟失敗的樣本，不計入字元辨識步驟的樣本內。

本實驗字元辨識步驟成功率為 89.02%。絕大部分仍以相似字辨識錯誤為主，如圖 9 所示，特別是在拍攝角度較為傾斜的狀況。上述失敗案例的解決方式，建立於訓練樣本的挑選，因此仍有改善的空間。由表 3 得知平均狀況下，整體辨識率達 87.20%。雖未達九成辨識率，但若於比對樣本訓練階段能夠訓練得更完備，則正確率可望大幅提升。另外，本文於 TMDXVDP6437 DSP 實驗板上，車牌辨識函式之執行時間對於定位出

(a)車牌影像樣本

(b)定位步驟產生之車牌影像

圖 8　本實驗車牌定位步驟定位到背景雜訊案例。

(a)車牌影像樣本

(b)定位步驟產生之車牌影像

(c)經傾斜度校正之車牌影像

(d)辨識結果

圖 9　本實驗字元辨識步驟失敗案例。

單一車牌之影像需要大約 2 秒的時間，其中不含其他影像傳輸額外處理時間。在執行時間上的差異，推測為樣本影像背景的複雜度所致。

六、結論

本文提出一適於可攜式裝置之車牌辨識系統，在 DSP 處理器上達到 87%的辨識成功率與約 2 秒的平均執行時間。在半自動的手持式影像擷取工具上，附加上本文的車牌辨識功能， 應可達到快速辨識取得車牌資訊並進行比對處理的目的，進而取代一般 PDA 型態裝置。然而，本文所提出之系統，仍有許多可以改進的地方。本文使用了 539 張車牌影像，其中九成為白天所拍攝的影像。不可諱言，本研究所使用的車牌影像樣本空間周延性尚有待加強。未來若要針對更多特殊環境進行演算法修正，必須建立車牌影像資料庫並加以分類，使得實驗結果能更接近實際的情況。

表 3　本文效能統計列表

項目 / 步驟	成功數	各步驟成功機率
定位步驟	534	99.07%
切割步驟	528	98.88%
辨識步驟	470	89.02%
合計	470	87.20%
備註	樣本總數共 539 個。	

另外，根據硬體平台或軟體開發環境的改變，本文所提出之系統仍有許多可以改善之處。再使用 DSP 處理器開發板的狀況下，本文所有程式為了快速開發，皆以 C 語言完成，但若將效能瓶頸之處以 DSP 處理器所使用的線性組合語言指令集來進行開發，應能有效提升執行效能。本系統所使用的影像處理函式，如區域二值化或連接物件編碼，亦可利用 FPGA 進行硬體實作，若能將具有高度規律性且計算量龐大的影像處理運算交由 FPGA 處理，再配合適當的排程規劃，應可大幅加速整體的運算速度。

參考文獻

[1] 王中山，使用小波轉換於車牌偵測，國立中山大學機械與機電工程學系碩士論文，2003。

[2] 李志文，嵌入式即時多標的汽機車牌照辨識系統，國立台灣科技大學電子工程系碩士論文，2008。

[3] 八木伸行、林正樹、三谷公二、奥井誠人、吳上立、林宏墩，C 語言數位影像處理，全華出版，2005。

[4] 呂炎坤，不限環境之車牌定位系統，靜宜大學資訊管理學系碩士論文，2007。

[5] 官宗緜，利用數位訊號處理器實現車牌字元辨識系統，國立台灣大學電機工程學研究所碩士學位論文，1999。

[6] 林家緯，以統計數據為基礎的車牌影像處理與辨識方法，國立中山大學機械與機電工程學系碩士論文，2004。

[7] C. Arth, F. Limberger, and H. Bischof, "Real-Time License Plate Recognition on an Embedded DSP-Platform," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[8] C. Arth, "Visual Surveillance on DSP-Based Embedded Platforms," Ph.D. dissertation, Graz University of Technology, Institute for Computer Graphics and Vision, 2008.

[9] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 3, September 2006.

[10] G. Yang, Q. Zhu, J. Chi, and X. Zhuang, "A License Plate Recognition System Based on GFNN and DSP," *IEEE International Conference on Networking, Sensing and Control*, pp. 160-164, April 2008.

[11] H. Fu and M. Xie, "Design and Realization of License Plate Recognition System Based on DSP and FPGA," *IEEE International Symposium on IT in Medicine and Education*, pp. 39-43, December 2008.

[12] J.-M. Gao and Y.-F. Liu, "License Plate Localization and Character Segmentation with Feedback Self-Learning and Hybrid Binarization Techniques," *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 3, May 2008.

[13] M. Xie, H. Fu and Z. Liu, "One Design Method of License Plate Recognition System with High Recognition Rate," *IEEE International Symposium on IT in Medicine and Education*, pp. 44-49, December 2008.

[14] Texas Instruments Inc, TMS320DM6437 Evaluation Module Technical Reference, 2007.

[15] Y. P. Huang, and T. Tsai, "A Practical License Plate Recognition System on PDA," *Int. Computer Symposium*, pp. 731-736, December 2004.

[16] Z. W. Liu, H. D. Fu, and M. Xie, "Multiple Processors License Plate Recognition System for Intelligent Transportation Management," *Second International Symposium on Intelligent Information Technology Application*, Vol. 1, pp. 333-336, December 2008.