

以輪廓辨識為基礎的車牌辨識處理程序

張逸中

致遠管理學院電機工程學系

本研究以包含車牌的影像輪廓辨識資訊為分析主體，完成一個適用於一般停車場環境的完整辨識流程，並已製作為單一處理軟體，以一般電腦執行約 1.5 秒內可完成一張約 50 萬畫素大小影像之車牌辨識。主要程序包括：最佳化灰階處理→二值化影像→目標物輪廓描繪與分析→車牌字元擷取→字元影像縮放與旋轉→字元比對。在程式效能持續提升與進行更多實況測試調整之後，預期可迅速成為實用之商用系統。

一、 簡介

車牌辨識技術有相當高的實用價值，舉凡停車場管理、犯罪行為監測、交通行為監視等等都需要快速、正確且成本合理的車牌辨識系統。此類系統軟體技術開發之門檻較高，建置實用系統時還必須使用到高品質的影像擷取系統，與高速的電腦處理能力，目前仍屬於高價位的產品。但在近年數位影像品質與電腦硬體效能提升與價格低廉化之後，系統普遍化的可能性大幅攀升，核心成本逐漸集中於軟體設計的技術與執行效能。

綜觀此議題之軟體設計研究，主要處理程序包括：

1.在影像中鎖定車牌位置並擷取之；2.車牌影像之形狀正規化與顏色對比強化；3.字元形態之辨識。[1]

這些步驟中，通常技術的瓶頸是影像中車牌位置的鎖定。最常見的是以空間掃描的方式，偵測車牌一般具有的高顏色反特徵(Edge detection)；或者以車牌預期形狀作移動相關性運算(Match filtering)。前者缺點是難以預期周遭是否同樣有高反差的其他影像元素；後者則難以預期車牌因為距離遠近不同造成的大小差異，以及因為拍攝角度造成的影像傾斜。

本文嘗試跳脫上述常用的技術，在常見的影像二值化之後，擷取出相當複雜且大量的邊界目標物，在眾多目標中以車牌字元應有的概略尺度，包括長、寬、長寬比，以及字元連續緊密排列等特性，將車牌字元抽離；接著以數個字元中心的連線計算車牌的傾角，並取得字元的平均高度作為影像正規化時縮放比例與旋轉角度的參考，影像經正規化校正後再與標準字元模板比對。

此外，對於極易誤判的字元如(0, D)、(8, 9, 6)、(P, F)等等組合，在辨識的相似度接近時，會以影像缺口位置或像素分布的差異進行進一步的辨識，如中心線兩邊黑色像素分布不對稱者為 D，否則為 0 等等。對於粘連雜訊的字元，如某字元與車牌中的小橫線相連成為單一目標時，以任何字元比對相似度都很低，此時會以橫向掃描方式做較複雜的比對，仍可順利完成正確的辨識。

二、 本文

2.1 影像前處理

與一般車牌辨識程序相似，全彩數位影像在本系統中會先經過轉換灰階、增強與二值化的處理。主要是將 RGB 色彩簡化為灰階模式，並進行增強對比的像素等值化處理(Histogram Equalization)[2][3]，最後以一閾值將灰階影像轉為二值化的黑白影像，因為車牌是白底黑字，預期會有相當明顯的對比。如下圖一至圖三所示：



圖一：原始影像

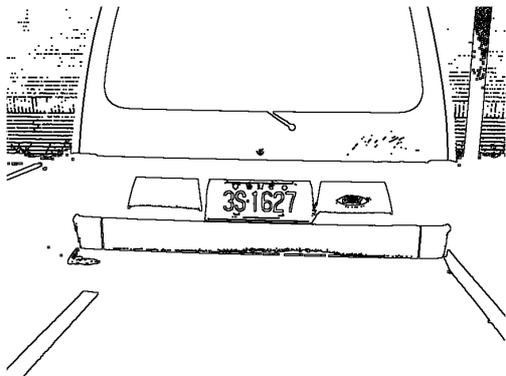


圖二：像素等值化處理後的灰階影像



圖三：二值化之後的黑白影像

由於車牌字元通常排列緊密，在進一步描繪輪廓線之前通常需要進行一個像素的侵蝕處理[1]。這會使得黑色目標縮減一個像素的寬度，對於數個像素點寬度的字元筆畫辨識並無影響，但是可以降低字元目標互相粘黏無法正確切割的機率。經此處理後描繪之邊界輪廓如下圖四。



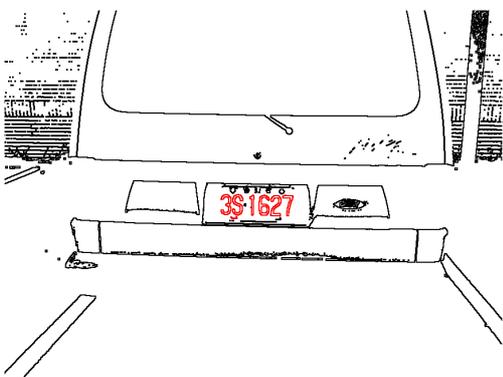
圖四：邊界點辨識後的輪廓影像

2.2 輪廓目標的篩選

在完成圖四的輪廓圖之後，下一個步驟是將相連的輪廓點組成目標物[3][4]。凡是相鄰的黑色像素點連接後被視為單一目標，可以看出在影像解析度良好時各個車牌字元將自成一目標。接下來的工作是排除非字元的目標，讓畫面上僅存六個車牌字元的目標，以人類的思考模式來看，就是將不像『字』的目標除去，而『像不像字』的準則是我們程式設計的關鍵。

在此，本系統設定的篩選原則是字的概略長寬與目標中心點的排列方式。太長、太寬、太細或太小的目標首先排除，剩下的少數目標則檢視其中心點的排列方式，取出『間距規律』且『橫向』排列的六個目標即為最可能的車牌字元。

一個本系統目前尚難完全自動化的唯一步驟就是決定字元或車牌的概略尺度。令程式自動搜尋各種可能大小的字元，理論上可行，但實務上將耗去相當長的運算時間，目前本系統未找出快速合理的方法。基於本系統開發的目的為停車場管理之用，車牌應在相當近似的距離下拍攝，車牌影像尺度差距不大，可以預設近似的尺度，且目前程式可以容忍約 30%的誤差，亦即設定字元大小為 40 像素時，實際尺度為 30 像素的字元亦能正確處理。圖五為圖四經過目標篩選的結果。



圖五：車牌字元目標擷取結果

在正確取得目標物邊界之後，程式會依此邊界資訊回頭切割黑白影像(圖三)內的像素點資訊作為個別的字元影像，每個字元隨後將以個別的方式進行辨識。

2.3 字元影像的正規化

字元辨識的主要步驟是將前面步驟擷取的字元黑白影像與標準字模進行黑白像素重疊與否的比對。但擷取之字元影像可能因拍攝距離遠近而有大小的不同，或因拍攝角度而略有傾斜。如何正確的旋轉以及縮放字元影像到與標準字模大小及相同且角度正確，稱為字元影像的正規化，此舉可以大幅提高辨識的準確率。

在本系統前面的處理程序中可以取得各字元的中心位置，如某目標的像素 X 座標分布範圍為 $x1-x2$ ，Y 座標範圍 $y1-y2$ ，則其中心點座標可定為：

$$(x1+(x2-x1)/2, y1+(y2-y1)/2)$$

計算各字元目標中心點的 Y 值變化，取其近似直線與水平線的角度即為應旋轉修正的角度，如圖六之 θ ；此外取得各字元的高差 $(y2-y1)$ ，也可以知道字元大小應修正的比例，如標準字模的高度為 H 像素，字元高為 h 像素，縮放比應為 H/h 。



圖六：字元正規化運算示意圖

因為白色背景通常佔了字元影像的大部分，任何比對結果都會有約半數以上的相同點，為了加強數值顯示的對比，本程式以相同者加一，不同者減一的倒扣方式累加數值。此舉並不影響辨識之結果，與相似度的排序。

2.4 字元辨識

本系統目前以 50X50 像素的字元模板為比對基礎，程式起始時將英數字模板影像載入，並轉換為矩陣資料；被擷取完成正規化的字元影像亦轉為陣列資料，在資料處理過程中如非資料展示所需，將不輸出影像，以加快資料比對的速度。為了掌握辨識的效能，系統預設會顯示出最接近的六個字元(共 36 字元)，並列出其相似度。圖七為圖五車牌的字元辨識結果：

3	S	1	6	2	7
3	S	1	6	2	7
[J]: 72.4 [3]: 70.24 [S]: 69.28 [9]: 66.56 [8]: 66.08 [B]: 63.52	[B]: 63.92 [S]: 62.32 [8]: 57.04 [6]: 56.64 [W]: 56 [D]: 54.16	[1]: 87.28 [7]: 68.56 [4]: 68.4 [3]: 49.28 [2]: 45.36 [5]: 40.56	[6]: 75.44 [0]: 65.68 [8]: 61.76 [9]: 60.96 [3]: 56.32 [4]: 55.92	[2]: 74.64 [8]: 65.92 [6]: 63.6 [8]: 63.2 [S]: 62.56 [P]: 62.32	[7]: 81.44 [1]: 72.8 [Y]: 70.56 [V]: 67.84 [I]: 65.76 [3]: 62.64

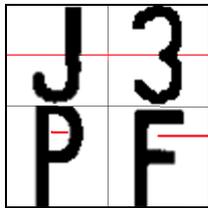
圖七：字元正規化運算示意圖

其中最上方是辨識結果以標準字體顯示，第二行是正規化之後的個別字元影像，最下方是最接近字元的相似度顯示，完全符合為 100%。但請注意到辨識結果在

此並不完全以模板相似度排行為依據，事實上左方第一字元模板比對後顯示與『J』字元的相似度略高於字元『3』！對於解晰度不高且可能有雜訊的字元影像來說，這是常有的情形，即使影像完全正確，其辨識結果的數值差異也很小。

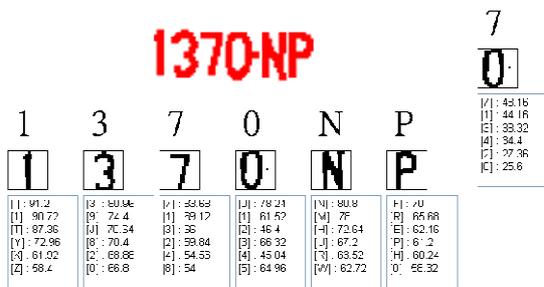
本系統的輔助辨識措施是以『字型特徵』為進一步確認的方式。譬如上例的『J與3』錯置，解決方式是凡碰到辨識結果為J或3時都進一步分析其影像上下半部的黑色像素個數，若為3，其數目應非常接近；若為J則會有明顯的下多於上的現象，以此準則即可輕易斷定出正確的字元。同樣常見的『I與1』以及『0與D』錯置的解決方式也很類似，它們的左右或上下半邊目標點像素比例會有相當的差別。

至於另一種常見的『P與F』錯置則須使用偵測字型缺口的技巧，在P字元中，上半部自水平向中心的空白點向右移動將無法直通邊界，F則有部份位置可以！以此特徵可以輕易辨識P與F。此外『8, 6, 9』與『S, B』的常見誤判也可以用類似的辨識方式輕易辨識。示意圖如下圖八：



圖八：字型特徵辨識示意圖

另一種狀況如圖九所示，正常情況下車牌字元中包含的短橫線會被視為過小目標而排除，但也很可能因為光影或解析度問題與相鄰字元粘連成為單一目標，此時以任何字元模板比對符合率都很低，結果當然也常常出乎預期，如圖九右上角顯示偏離中心的目標字元0按正常程序會被解析為字元7。



圖九：字元影像粘連雜訊時的狀況

針對此情況，系統預設如果全體字元相似度都低於60%，會認定字元影像有明顯的中心點偏離。自動進行移動掃描的處理，字模會逐步左右移動多達半個圖幅的距離以尋找最高符合度的位置，如此當然會多花些許時間，但最終可以得到合理的辨識結果，如圖十。



圖十：橫向字模掃描比對

2.5 目標尺度設定

本系統所以能排除大半雜訊的一個關鍵參數，事實上是概略的字元尺度。譬如設定尋找高度約40像素的字元，但是如果實際字元尺度差距太大，如僅20像素高的字元則會被排除無法進行辨識。理論上可以進行全尺度全圖的掃描，以決定字元尺度，但如此處理時間將遠超出合理範圍。目前處置方式是設計一個簡單的設定介面，使用者可用滑鼠點選原始影像的車牌兩側(如圖十一)，系統即可獲得概略的車牌以及字元尺度的資訊，據以進行目標辨識，並以檔案儲存，下次啟動時自動載入，直到再次被重設為止。



圖十一：車牌尺度設定

這個尺度設定並非十分嚴苛，程式內部其實可以放寬容錯率，但是代價是有些雜訊可能因此無法排除，經過測試調整，目前容錯率設定約為30%，亦即設定字元為40像素高時仍可偵測約30像素高的字元。對於影像尺度差距不大的環境，如停車場入口的監視器，應該已足敷使用。

結論

對於目前多數車牌辨識系統而言，搜尋車牌位置的機制多以小波轉換的掃描方式為主[5]。直接以二值化影像邊界群為處理核心的方法較為少見。其困難在於不定形狀像素點群組處理的程式設計較為複雜[4]，但此種技術對於其他目標物的辨識也會是一個良好的基礎。在字元辨識方面，除了傳統的模板比對辨識，本文也對易於誤判的字元組合進行針對性的診斷辨識，可在電腦時間消耗較低的前提下有效提升辨識正確率。本研究以成為實用系統為目標，其效能未硬體化的階段已可達到在1.5秒之內完成50萬畫素影像的辨識。

參考文獻

- [1] S. L. Chang, L. S. Chen, Y. C. Chung and S. W. Chen, "Automatic license plate recognition," IEEE Transactions on Intelligent Transportation Systems, vol. 5, pp. 42-53, Mar. 2004
- [2] 林宸生，數位信號—影像與語音處理，全華科技圖書，1997，第四、五章
- [3] N. Otsu, "A Thresholding Selection Method from Gray Level Histograms," IEEE Transactions on Systems, Man and Cybernetics, vol. 9, pp. 62-66, 1979
- [4] Yet-Chung Chang, Gwo-Shyh Song and Shu-Kun Hsu, Automatic Extraction of Ridge and Valley Axes Using the Profile Recognition and Polygon Breaking Algorithm. Computers & Geosciences, vol.24, no.1, p.83-89., 1998
- [5] 陳翔傑，自動化車牌辨識系統設計，中央大學電機工程研究所碩士論文，2005