# 應用相連區塊為主的資訊於自然場景影像中的文字定位

范欽雄
國立台灣科技大學資訊工程系
csfahn@mail.csie.ntust.edu.tw

朱家輝
國立台灣科技大學資訊工程系
b6n@pchome.com.tw

## 摘要

隨著電腦科技的日益進步，使得以電腦輔助的自動化處理也越來越廣泛，尤其是結合日常生活方面的應用，其中光學文字辨識系統就是一個相當典型的例子。在光學文字辨識中，文字定位的前置處理扮演著相當重要的角色，特別是處於複雜背景或自然場景的影像。在本篇論文中，我們主要利用相連區塊的資訊在自然場景影像中進行文字區塊的定位。首先，我們將輸入的自然場景影像由 RGB 彩色空間轉為 YIQ 彩色空間，隨後使用一個有效的邊緣偵測運算子對 Y 分量所形成的影像做邊緣偵測以及二值化處理。接著，我們以一個經改良的標記演算法同時對二值化影像中的兩個二元值做連接處理而得到位於其中的相連區塊，再根據這些相連區塊在影像中的色彩、位置以及大小資訊分類出可能的文字區塊。最後，依照這些區塊在幾何上的特徵來辨別出真正的文字區塊。實驗結果顯示我們的方法能夠正確而且快速地定位出自然場景影像中的文字區塊。

**關鍵詞：**文字定位、相連區塊、自然場景影像、邊緣偵測、標記演算法。

## 1. 簡介

隨著電腦的發達與普及化，使得我們的日常生活越來越緊密地和它們結合在一起，而電腦的應用層面也越來越廣泛，其中又以電腦輔助的自動化處理為最。因為電腦輔助的自動化處理帶給了我們生活上莫大的方便，讓人們的生活比以往更有效率，也使得我們在工作上可以花費更少的時間。於多樣性的電腦輔助自動化處理中，文字辨識可以算是和我們的關係最為密切，目前它在生活上的應用可區分成兩大類型，其中一種類型是用在偵測及辨識文件影像上面的文字，而另外一種類型則是在動態影像(如電視節目、多媒體影像或是即時的影像)上面做文字的偵測及辨識處理。這兩種類型的應用都是希望能夠將影像當中的文字資訊，透過自動化的文字定位與辨識而將它們數位化，讓電腦可以知道影像中文字的數位化資訊，以利後續的應用。

近年來，對於自然場景影像當中的文字先進行偵測與定位的前置處理，再將結果輸入光學文字辨識系統做文字字元的自動化辨識已經有越來越多的成功例子，其中兩個非常普遍的應用範例就是車牌的自動化偵測與辨識以及郵件地址的自動化處理與分類。我們希望自然場景影像的文字偵測處理能夠再做更廣泛的應用，使它們能夠結合在我們的日常生活當中；例如推廣到一般的街道場景影像，因為這些影像通常含有許多的標誌以及招牌文字，它們主要提供給我們名稱、交通、商業或是注意等類型的資訊。透過文字偵測的前置處理，我們可以將上述所說的街道場景影像中包含文字的標誌以及招牌裡的文字區塊擷取出來，然後經由文字辨識系統進行辨識處理，使電腦可以得到關於這些標誌以及招牌文字的數位化資訊。

目前在文獻裡針對複雜背景或是自然場景影像以及動態的多媒體影像所做的文字偵測定位研究，主要分為兩大類型：基於相連區塊(connected-component-based)法[1-4]與基於區塊紋理 (texture-based) 法[5-8]。利用相連區塊法主要是從原始影像當中，依據相連區塊的資訊找出許多的子影像，再根據一些文字幾何以及影像的版面配置資訊來找出真正的文字區塊。而區塊紋理法則是利用一些紋理分析的技術如：嘉伯濾波器(Gabor filtering)[5]、空間變異數(spatial variance)[6]、離散餘弦轉換(discrete cosine transform)[7]或是小波轉換(wavelet transform)[8]等，藉由每一區塊的頻率係數來判定是否為文字區塊，然後利用數理形態學(mathematical morphology)上的運算對區塊做擴張及收縮的處理，以找尋出影像當中的文字區塊位置。利用區塊紋理法雖然能夠將影像中的文字區塊偵測出來，卻無法對文字區塊位置做定位；反之，利用相連區塊法不僅能將影像中的文字區塊偵測出來，還能夠同時對文字區塊進行定位。在本論文裡，我們

主要將適合於彩色複雜背景影像的文字定位方法[1]應用到自然場景影像中，並針對後者中的文字區塊特徵作修改，而可以正確地定出文字區塊的位置。

## 2. 文字區塊定位

為了降低影像在計算上的複雜度，我們先將輸入的彩色自然場景影像由 RGB 的彩色空間轉換到 YIQ 的彩色空間，然後將其中的 Y 分量作為後續處理的影像。

### 2.1 影像測邊及二值化

在先前的相關研究裡，大多數利用相連區塊法都是以 Sobel 運算子作為影像的測邊工具。但是對於低解析度且背景複雜的自然場景影像而言，經由 Sobel 運算子測邊以及二值化的結果得知：影像中的文字區塊會邊緣化得很厲害，而且有可能會連成一塊。因此我們不以 Sobel 運算子做影像的邊緣偵測，而採用一個效果較好的改良式 Laplacian 測邊運算子[9]來對影像 $f(x,y)$ 進行測邊以及二值化的運算，其對應的 5×5 遮罩如下所示：

$$\nabla^2 f(x,y) \approx \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \frac{1}{225}\begin{bmatrix} 9 & 9 & 9 & 9 & 9 \\ 9 & -16 & -16 & -16 & 9 \\ 9 & -16 & -16 & -16 & 9 \\ 9 & -16 & -16 & -16 & 9 \\ 9 & 9 & 9 & 9 & 9 \end{bmatrix} \quad (1)$$

圖 1 為我們對同一自然場景影像以不同的邊緣偵測運算子所做的測邊處理以及二值化的結果。從此圖可以明顯地發現：我們所使用的測邊運算子對於影像中的文字區塊有相當完整的結果。另外，為了加快整個文字定位的處理速度，我們在二值化臨界值的選擇上係利用一固定的臨界值 $\tau$ (在本論文中設為 0)。

(a)


(b)


(c)

圖 1 不同測邊運算子所得到的二值化影像：(a)原始影像；(b)Sobel 運算子的測邊結果；(c)改良式 Laplacian 運算子的測邊結果。

### 2.2 相連區塊連接

由圖 1(c)可以發現到文字區塊並非完全由二值化影像中的某一個二元值所連接而成，因此必須對二值化影像中的兩個二元值同時進行相連區塊的連接處理。我們修改於文獻[10]所提出的快速相連區塊標記演算法，使其能夠快速且同時對兩個二元值進行相連區塊的連接。此演算法敘述如下：

首先，我們設定一長度為 $2L$ 的一維陣列表格 $T[\ ]$，用來儲存標記過程中具有等價關係的標記值，$L$ 為此陣列表格的索引值。另外，我們亦給予兩個參數 *black_m* 以及 *white_m*，用來記錄現在分別標示到的黑色像素點和白色像素點的標記值，其中 *black_m* ＞0 且 *white_m*＜0，而 *black_m* 的初始值為 1，且 *white_m* 的初始值則為-1。

針對二值化影像 $b(x, y)$ 在第一次由左上角到右下角的掃描中，先給予每一個像素點一個暫時的標記值而另形成一張標記影像 $g(x,y)$，如式(2)：

$$
\begin{aligned}
&\textit{If } b(x,y) = \textit{BLACK, then}\\
&\quad \textit{If } b(x+i,y+j) = \textit{WHITE for } \forall (i,j) \in M_{\mathrm{FS}}\\
&\qquad g(x,y) = black\_m\\
&\qquad T[L + black\_m] = black\_m\\
&\qquad black\_m = black\_m + 1\\
&\quad \textit{Else}\\
&\qquad g(x,y) = \max_{(i,j)} \{T[L + g(x+i,y+j)] \mid (i,j) \in M_{\mathrm{FS}}\}\\
&\textit{Else}\\
&\quad \textit{If } b(x+i,y+j) = \textit{BLACK for } \forall (i,j) \in M_{\mathrm{FS}}\\
&\qquad g(x,y) = white\_m\\
&\qquad T[L + white\_m] = white\_m\\
&\qquad white\_m = white\_m - 1\\
&\quad \textit{Else}\\
&\qquad g(x,y) = \min_{(i,j)} \{T[L + g(x+i,y+j)] \mid (i,j) \in M_{\mathrm{FS}}\} \qquad (2)
\end{aligned}
$$

於式(2)中，$M_{\mathrm{FS}}$為像素點$(x,y)$與四個鄰居點$(x\text{-}1,y)$, $(x\text{-}1,y\text{-}1)$, $(x,y\text{-}1)$和$(x+1,y\text{-}1)$的相對位置所形成的集合，即{(-1,0), (-1,-1), (0,-1), (1,-1)}。在第一次掃描之後，再由右下角到左上角掃回去，根據標記等價關係表格的內容將具有等價關係的標記值做更新，如式(3)所示：

$$
\begin{aligned}
&\textit{If } g(x,y) > 0, \textit{ then}\\
&\quad g(x,y) = \max_{(i,j)} \{T[L + g(x+i,y+j)] \mid (i,j) \in M_{\mathrm{BS}}\}\\
&\quad \textit{If } g(x+i,y+j) > 0 \textit{ for } (i,j) \in M_{\mathrm{BS}}\\
&\qquad T[L + g(x+i,y+j)] = g(x,y)\\
&\textit{Else}\\
&\quad g(x,y) = \min_{(i,j)} \{T[L + g(x+i,y+j)] \mid (i,j) \in M_{\mathrm{BS}}\}\\
&\quad \textit{If } g(x+i,y+j) < 0 \textit{ for } (i,j) \in M_{\mathrm{BS}}\\
&\qquad T[L + g(x+i,y+j)] = g(x,y) \qquad (3)
\end{aligned}
$$

相似地，式(3)中的$M_{\mathrm{BS}}$為像素點$(x,y)$本身及其與四個鄰居點$(x+1,y)$, $(x+1,y+1)$, $(x,y+1)$和$(x\text{-}1,y+1)$的相對位置所形成的集合，即{(0,0), (1,0), (1,1), (0,1), (-1,1)}。

由於本篇論文主要利用相連區塊在自然場景影像中的資訊來定位出文字區塊，因此我們在做相連區塊的標記處理時，一併記錄每一個編號區塊在自然場景影像中的資訊，包括外接相連區塊的最小矩形的最左上角座標$(x_l,y_l)$、最右下角座標$(x_r,y_r)$，以及此一相連區塊的RGB色彩平均值$(m_R,m_G,m_B)$；有了這些資訊，我們便能夠對相連區塊進行分類。

經由相連區塊演算法的處理後，我們可以得到輸入影像內可能包含文字字元的相連區塊物件，但是並非全部的相連區塊物件都含有文字字元，因為一個文字字元在一張自然場景影像當中可能會有一定大小的範圍。據此，我們可以先利用文字字元區塊的大小範圍資訊來粗略地過濾出可能構成文字的相連區塊字元。下面幾個條件便是我們用來初步判斷可能沒有包含文字字元區塊的可能文字區塊：

1) 相連區塊的寬度或是高度和輸入影像的寬度或是高度都相等。
2) 相連區塊的寬度或是高度皆小於 5 個像素點。
3) 相連區塊的位置太靠近影像邊緣。

### 2.3 相連區塊分類

在相連區塊的分類上我們除了採用色彩以及位置資訊的分類外[1]，在位置分類之前，我們會先對同一色彩類別的相連區塊，依其大小做粗略的分類。

針對色彩資訊是以兩階段的分類方法來做分類，其中第一階段是先決定相連區塊的色彩分佈，再找出各個色彩類別的中心點；第二階段則是利用 K-means 的聚類演算法將各個相連區塊做色彩分類。接著，我們對於每一個色彩類別，根據它們各個相連區塊的寬度以及高度進行群聚分類。在利用相連區塊大小資訊做分類的方法中，我們對於每一個相連字元區塊計算它們與每一個中心點的寬高差距和，然後取最小的寬高差距和。若是此寬高差距和大於我們給定的閥值時，則新增一個類別，並將此相連區塊設為新類別的中心點；否則的話，將此相連字元區塊歸為寬高差距和最小的那一個類別，並且更新此類別中心的寬度與高度為當中所有相連文字區塊的寬度以及高度的平均值。

於文獻[1]中有提出一個對相連區塊做位置分類的遞迴切割方法(recursive XY-cut

procedure)，但由於此方法是先進行水平方向(Y-cut)的切割，再接著做垂直方向(X-cut)的切割，因此最後分類出的文字區塊限定為水平方向的文字區塊。自然場景影像的文字區塊並非皆為水平文字區塊，所以我們除了對每一大小類別做 recursive XY-cut 的處理外，同時亦做 recursive YX-cut 的處理。如此一來，我們便能夠同時切割出水平以及垂直的文字區塊。

### 2.4 辨別真正文字區塊

經由相連區塊的分類後，我們可以得到許多水平或是垂直的可能文字區塊。最後，我們利用[1]所提的文字區塊在幾何上的一些特徵範圍內辨別出真正的文字區塊。以下為這些文字區塊的特徵：

1)水平方向的文字區塊

- $W/H \geq 1.2$：$W$ 與 $H$ 分別為可能文字區塊的寬度與高度。
- $2 \leq N \leq 8(W/H)$：$N$ 為可能文字區塊中相連區塊的個數。
- $W_c/H \leq 1.5$：$W_c$為可能文字區塊中的相連區塊的寬度。
- $A/(W \cdot H) \geq 0.6$：$A$ 為可能文字區塊中相連區塊所佔的總面積。

2)垂直方向的文字區塊

- $H/W \geq 1.2$：$H$ 與 $W$ 分別為可能文字區塊的高度與寬度。
- $2 \leq N \leq 8(H/W)$：$N$ 為可能文字區塊中相連區塊的個數。
- $H_c/W \leq 1.5$：$H_c$為可能文字區塊中的相連區塊的高度。
- $A/(W \cdot H) \geq 0.6$：$A$ 為可能文字區塊中相連區塊所佔的總面積。

我們對水平或垂直方向的可能相連文字區塊，分別根據上述的條件辨認是否為真正的文字區塊。但是在自然場景影像之中，文字區塊裡面的字元區塊的排列情況並非為完全水平或是垂直排列，而有可能字元區塊的排列呈現歪斜的狀況。歪斜的文字區塊必定無法滿足上述所列的文字區塊的幾何特徵規則，因此我們再對這些可能文字區塊中的每一個可能字元區塊，各別去計算它們的寬度($\sigma_W$)和高度($\sigma_H$)的標準差；若是此寬度和高度的標準差總和($\sigma_S$)小於一標準差閥值的話，那麼我們同樣視此可能文字區塊為一真正的文字區塊。


圖 2 自然場景影像的文字定位圖例。

圖 2 為圖 1(a)利用前一節所提的文字定位方法處理過後，得到影像中每一個相連文字區塊的位置。在此圖中，我們可以發現影像中的'L'及'N'兩個字元，由於包含它們的字元相連區塊過小的關係而被過濾掉了，但是我們可能在最後的定位結果輸出中將它們表示出來。因為我們對於每一個相連區塊都可以得到色彩方面的資訊，所以在輸出結果的時候只要將文字區塊向外擴張一定的範圍，再比較每一像素點與文字區塊色彩資訊的距離。若是距離相近的話，那麼此一像素點便是自然場景影像中組成文字區塊的像素點；否則的話，此一像素點便為背景點。圖 3 就是經過文字定位方法處理所獲得的二值化文字區塊影像。


圖 3 已經定位的二值化文字區塊影像。

## 3. 實驗結果

本實驗係透過數位相機在一般的街道上拍攝到 40 張包含有文字區塊的自然場景影像，並固定為 320x240 個像素的大小。在圖 4 中，我們列出幾個不同拍攝情況的文字定位實驗結果，其中包括文字區塊偵測錯誤以及文字字元無法偵測出來的範例。另外，我們針對此實驗分別以式(4)及式(5)計算字元偵測率和區塊錯誤率，並將統計結果列於表 1。

$$\text{字元偵測率} = \frac{\text{偵測出來的字元數}}{\text{自然場景影像的總字元數}} \times 100\% \qquad (4)$$

$$\text{區塊錯誤率} = \frac{\text{錯誤的區塊數}}{\text{文字定位結果的總區塊數}} \times 100\% \qquad (5)$$

表 1 40 張自然場景影像的文字定位
實驗結果統計

| 字元偵測率 | 區塊錯誤率 | 平均處理時間 |
|---|---|---|
| 92.99%<br>(96.92%) | 21.68% | 0.421 秒 |

在表 1 的字元偵測率欄位中，括號內的字元偵測率為先前所提到的對文字區塊做向外擴張，再計算各像素點的色彩距離後可以得到的字元偵測率。觀察表 1 得知：利用相連區塊為主的資訊在自然場景影像中做文字定位的方法於文字字元偵測率上有著不錯的結果，並且在文字定位處理的效率上顯得相當的快速。上述實驗是以個人電腦配備 Pentium-4 1.6GHz 中央運算處理器，在微軟 Windows 2000 作業系統下執行，而發展程式係以J2SE 1.4.1 SDK電腦語言所撰寫。

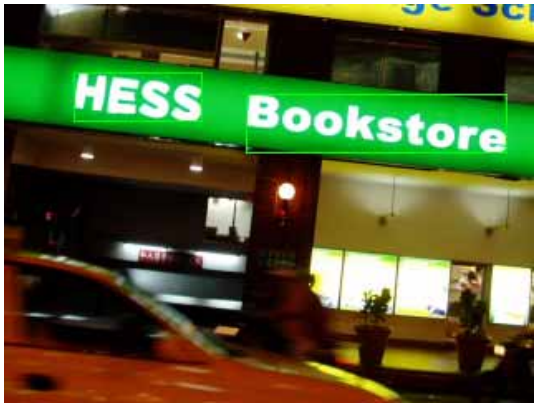
## 4. 結論

本論文主要是提出一個應用相連區塊為主的資訊在自然場景影像當中的文字區塊定位方法。在此方法中，我們利用一個有效的邊緣偵測運算子來對自然場景影像做測邊並將所得的影像二值化。接著，提出一個改良過的標記演算法，它可以快速地同時對二值化影像中的兩個二元值做相連區塊的連接處理。最後，針對相連區塊的色彩、位置與大小資訊，以及幾何特徵做分類，而驗證出真正的文字區塊。在實驗結果中，我們的方法不僅可以快速而且正確有效地定位出自然場景影像當中水平或是垂直文字區塊的位置，對於部份歪斜的文字區塊也能夠定位出來。在未來的研究中，可以透過結合相連區塊以及區塊紋理的方法來降低偵測錯誤率；另外，可以利用本文對自然場景影像所做的文字區塊定位結果當作文字辨識系統的前置處理，亦即將所獲得的二值化文字區塊影像，進一步做文字字元的辨識。

(a)

(b)

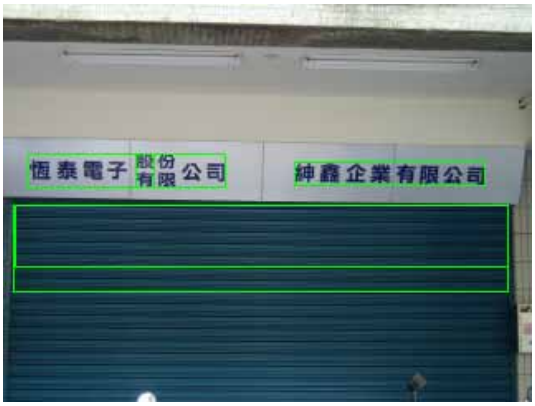
(c)



(d)


(e)


(f)




(g)


(h)

圖 4 部分自然場景影像的文字定位實驗結果：(a)白天場景；(b)夜晚場景；(c)左右歪斜；(d)上下歪斜；(e)顛倒文字；(f)垂直水平混合文字；(g)錯誤區塊；(h)遺漏字元。

## 5. 參考文獻

[1] H. M. Suen and J. F. Wang, "Segmentation of uniform-coloured text from colour graphics background," *IEE Proc. on Vision, Image and Signal Processing*, vol. 144, no. 6, pp. 317-322, 1997.

[2] J. Gao and J. Yang, "An adaptive algorithm for text detection from natural scenes," in *Proc. of the 2001 IEEE Comput. Soci. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 84-89, 2001.

[3] C. Li, X. Ding, and Y. Wu, "Automatic text location in natural scene images," in *Proc. of the 6th Int. Conf. on Document Analysis and Recognition*, pp. 1069-1073, 2001.

[4] J. Wu, S. L. Qu, Q. Zhuo, and W. Y. Wang, "Automatic text detection in complex color image," in *Proc. of the 2002 Int. Conf. on Machine Learning and Cybernetics*, vol. 3, pp. 1167-1171, 2002.

[5] A. K. Jain and S. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Applications*, vol. 5, no. 3, pp. 169-184, 1992.

[6] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," *Pattern Recognition*, vol. 28, no. 10, pp. 1523-1535, 1995.

[7] Y. Zhong, H. J. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, 2000.

[8] H. P. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digitial video," *IEEE Trans. on Image Processing*, vol. 9, no. 1, pp. 147-156, 2000.

[9] P. S. Yeh, S. Antoy, A. Litcher, and A. Rosenfeld, "Address location on envelopes," *Pattern Recognition*, vol. 20, no. 2, pp. 213-227, 1987.

[10] K. Suzuki, I. Horiba, and N. Sugie, "Fast connected-component labeling based on sequential local operations in the course of forward raster scan followed by backward raster scan," in *Proc. of the 15th Int. Conf. on Pattern Recognition*, vol. 2, pp. 434-437, 2000.