

建構個人化中文全真字型

洪文斌¹，陳志遠¹，彭建文²

1. 淡江大學資訊工程系, 新北市 25137

E-mail: horng@mail.tku.edu.tw

2. 致理技術學院商務科技管理系, 新北市 22050

E-mail: pchw8598@mail.chihlee.edu.tw

摘要: 目前的電腦字型多為統一固定的字型，缺乏個人書寫風格的特性。在本論文中，我們提出一個自動將個人書寫的中文字，透過影像處理技術，將之轉換成全真字型 (TrueType font) 的方法，使得電腦螢幕顯示或印表機列印時，仍可呈現個人書寫的風格。

關鍵字: 角點偵測, 貝茲曲線, 全真字型

Constructing Personalized Chinese TrueType Fonts

Wen-Bing Horng¹, Chih-Yuan Chen¹, and Jian-Wen Peng²

1. Department of Computer Science and Information Engineering, Tamkang University, Taipei, 25137

E-mail: horng@mail.tku.edu.tw

2. Department of Commerce Technology and Management, Chihlee Institute of Technology, Taipei, 22050

E-mail: pchw8598@mail.chihlee.edu.tw

Abstract: Most of the modern computer fonts are fixed, without personal style. In this paper, we propose a method which can automatically convert the handwriting Chinese characters, through the image processing techniques, to a personalized Chinese TrueType font. In this way, personalized handwriting style can be displayed in the monitor and printed by the printer.

Key Words: Corner Detection, Bézier Curve, TrueType Font

1 緒論

現行電腦中的字型 (font) 種類眾多，由多家字型公司製作，例如華康字型[1]、全真字庫[2]等。然其所製作出來的各種優美字型為一般普羅大眾所使用，早期多偏向於印刷字體與美工字體，例如：華康楷書體、華康明體、華康仿宋體、華康隸書體、華康行書體、華康POP1體、華康海報體等。近年來亦開始製作手寫字體，例如：華康鋼筆體、華康雅風體、華康秀風體等，雖然風格多變，然而使用者選取其字體時，所呈現的字形都一樣，缺乏個人書寫的風格與特質。

現今字體製作方式繁瑣，首先請書法專家由人工方式畫出初稿、加以描繪其外框 (outline)，再利用字型產生器 (例如：FontCreator [3]) 編輯、修正其外觀，最後自動產生向量描邊字，存成TrueType Font的全真字型。近來，也有研究者[4,5]提出一個自動將 (例如阿拉伯字母或拼音字母等) 簡單筆畫的數位字元影像 (digital character image) 自動取出其外框，並利用三次貝茲曲線 (cubic Bézier curve) 或是三次Hermite曲線來描述其外框，以便利電腦來顯示。然其所研究範疇也僅限於印刷字體與美工字

體，這些字體本身在設計時，就已有很平滑的外框，較易處理。

在本論文中，我們提出一個新的構想，那就是將書寫在紙上具有個人手寫風格的中文字體，透過影像處理的技術，自動去除數位化時所產生的雜訊 (noise)，尋找文字外框，並利用二次 (quadratic) 貝茲曲線來描述外框，以符合全真字型的規範[6]。最後，製作出可供電腦使用的個人化中文全真字型 (Personalized Chinese TrueType Font)，可以在螢幕及列印時，呈現使用者個人書寫字體的風格。

本論文的結構如下，第二節敘述建構個性化字體的方法，第三節為實驗的結果，第四節為最後的結論。

2 建構方法

本論文提出建構具有個人化風格之中文電腦字形。欲取得個人風格之字形，在本研究中先由使用者以紙筆寫下，透過掃描器 (scanner) 將之掃描為數位影像格式，再利用影像處理技術以取得字形資料，最後再將字形資料存成全真字型檔案以建構個人化字型。圖1為建構個人化字型資料的系統流程圖。以下小節將說明每個處理的步驟。

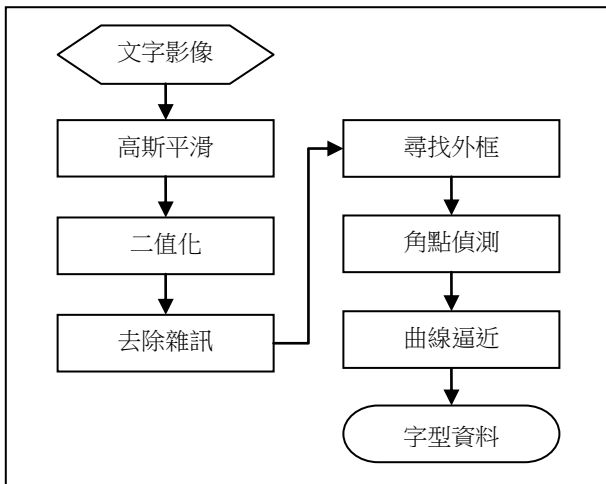


圖 1: 建構個人化字型資料流程圖

2.1 二值化

二值化 (binarization or thresholding) 是將所掃描進來的灰階文字影像轉成黑白影像。我們使用傳統的Otsu [7]演算法將手寫文字影像 (圖2(a)) 二值化 (圖2(b)) 以取得文字之輪廓外框。然因墨水會暈開及毛邊現象, 影響結果, 故先將原圖用高斯平滑化 (Gaussian smoothing) 方法減低暈開及毛邊, 然後再用Otsu方法二值化, 其結果如圖2(c)所示。

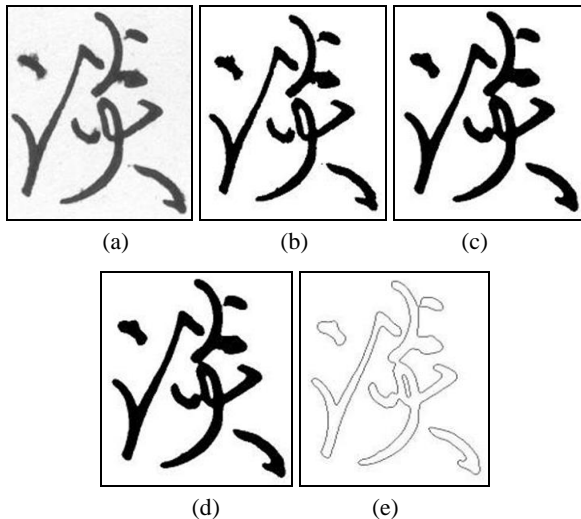


圖 2: (a) 原圖; (b) Otsu二值化; (c) 先經高斯平滑, 再二值化; (d) 將(c)去除雜訊; (e) 文字外框輪廓

2.2 去除雜訊

有時因紙張的關係, 在二值化後影像會存在著較小區塊的雜訊, 我們採用4-連通連接元件的方式 (4-connected component), 將低於某個門檻值的連接元件去除, 以得到較乾淨的文字影像, 如圖2(d)所示。在此例中, 恰巧沒有較小的連接元件, 所以圖2(c)與圖2(d)剛好相同。

2.3 尋找外框

到上一階段為止, 文字區域影像已經完整被保留在影像中。因全真字型為描邊字, 需取出文字的外框 (即外形輪廓 (contour))。透過一般尋邊演算法, 即可得到文字的邊界, 如圖2(e)所示。

2.4 角點偵測

建立全真字型資料最重要的一件事, 便是將字形外框輪廓以二次貝茲曲線[8]來描述一段一段的文字外框。因此, 在作曲線描述之前, 必須對文字外框作角點偵測 (corner detection)。角點一般定義為區域最大曲率 (curvature) 的位置。在數位影像上, 大都採用近似的方法來求得, 我們採用夾角餘弦的方式 (cosine of included angle) [9,10], 其方法如圖3所示。

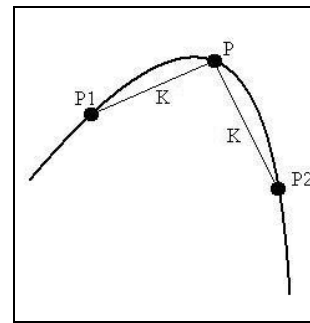


圖 3: 角點偵測示意圖

假設目前偵測一外框輪廓上點 $P(x, y)$ 的夾角, 外框點是以順時針方向來處理, 點 P 其後面第 k 個點為 $P1(x_1, y_1)$, 其前面第 k 個點為 $P2(x_2, y_2)$, 則點 P 的夾角餘弦為

$$C_k = a_k \cdot b_k / (|a_k| \cdot |b_k|),$$

其中, $a_k = (x - x_1, y - y_1)$, $b_k = (x - x_2, y - y_2)$ 。設定一門檻值 T , 點 $P(x, y)$ 若為角點必須符合下列兩個條件:

- (1) 其 C_k 是區域最大值,
- (2) $C_k > T$ 。

圖4為角點偵測結果, 每個空心圓代表一個角點。

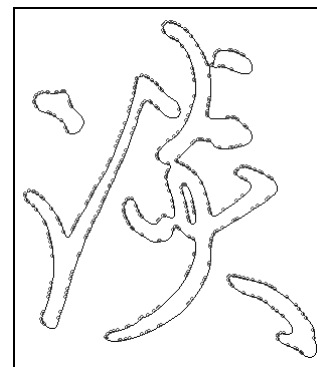


圖4: 角點偵測結果

2.5 曲線逼近

針對角點所切割出來的各段外框輪廓線作二次貝茲曲線逼近 (approximation)。一個二次貝茲曲線由三個控制點 (p_0, p_1, p_2) 所決定，其為一個參數式的 (parametric) 方程式[7]，定義如下：

$$P(u) = p_0B_0(u) + p_1B_1(u) + p_2B_2(u),$$

其中， u 為參數 ($0 \leq u \leq 1$)，決定繪製點的數量， p_0 與 p_2 為兩端控制點， p_1 為中間的控制點， $B_0(u) = (1-u)^2$ ， $B_1(u) = 2u(1-u)$ ， $B_2(u) = u^2$ 。

在求曲線逼近時，在外框輪廓上的兩個連續角點即決定一條二次貝茲曲線，兩個角點為兩個端點 (即 p_0 與 p_2)，剩下中間的控制點 p_1 待求。我們利用最小平方差 (least square error) 方式來求得。首先定義誤差公式如下：

$$E(u) = \sum_{i=0}^{D-1} d_i = \sum_{i=0}^{D-1} [P(u_i) - q_i]^2$$

其中， D 為外框輪廓在 p_0 與 p_2 之間點的個數， q_i 為外框線 p_0 與 p_2 上的點， $P(u_i)$ 為所求二次貝茲曲線上對應 q_i 的點， d_i 為 q_i 與 $P(u_i)$ 距離的平方。(注意： $p_0 = q_0$ ， $p_2 = q_{D-1}$ 。)

令 $E(u)$ 對控制點 p_1 作偏微分等於 0，亦即

$$\frac{\partial E(u)}{\partial p_1} = 0$$

可得 p_1 如下：

$$p_1 = \frac{\sum_{i=0}^{D-1} [q_i - B_0(u_i)p_0 - B_2(u_i)p_2]}{\sum_{i=0}^{D-1} B_1(u_i)}$$

圖5為曲線逼近結果，其中，方點為外框輪廓線上的角點 (即 p_0 與 p_2 控制點)，而圓點為所求得的 p_1 控制點。

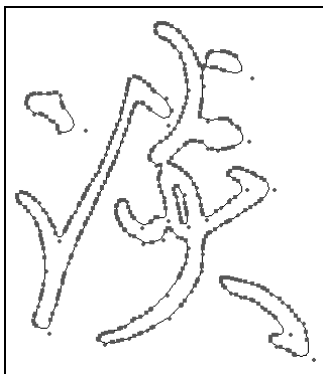


圖5: 曲線逼近結果

2.6 字型資料

將這些文字外框貝茲曲線控制點資料儲存成全真字型檔案 (.TTF) 中的描述資料，轉換的內容必

須遵照全真字型[6]檔中相關表格所定義之資料格式作轉換，最後即得到全真字型檔案。圖6(a)為原始手寫字形，圖6(b)為由全真字型檔所產生出來的向量字形，兩者非常神似。

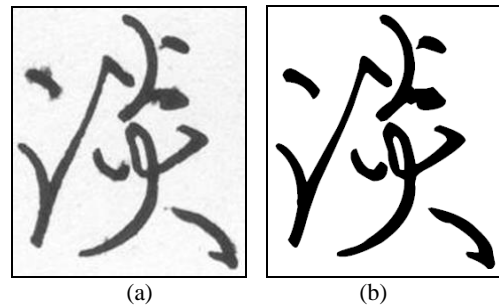


圖6: (a) 原圖；(b) 個人化全真字型

3 實驗結果

在此研究中，我們邀請當今書法名家張炳煌老師 (也是淡江大學中文系教授、文緝藝術中心主任、書法研究室主任) 書寫唐詩詩句，然後依據上述步驟轉換成具有張炳煌老師個人書寫風格的全真字型。圖7為張炳煌老師所寫原始手稿的部份影像，圖8為張炳煌老師所手寫的唐詩，經轉換成全真字型，9 point size所呈現的結果。利用微軟視窗作業系統內建的秀字功能，可以自動將字體做各種變化，其中第一列為標準字，第二列為粗體字 (bold)，第三列為斜體字 (italic)，第四列為粗體加斜體 (bold + italic)。圖9為「淡江」兩字各種不種大小的字體。

一	平	口	江	晴	石	八	秋
二	下	白	湘	台	在	天	舍
三	坐	百	山	吾	非	未	會
上	半	自	洵	黃	片	朱	更
五	六	中	山	征	州	春	使
江	華	里	畔	梧	君	火	淡
玉	年	重	出	岸	若	秦	暮
十	任	曲	笛	香	群	金	莫
生	日	但	車	似	入	東	便
干	住	直	唱	不	人	來	草

圖7: 張炳煌老師所寫原始手稿的部份影像

李白 下江陵 (標準)	張繼 楓橋夜泊	王維 渭城曲
朝辭白帝彩雲間 千里江陵一日還 兩岸猿聲啼不住 輕舟已過萬重山	月落烏啼霜滿天 江楓漁火對愁眠 姑蘇城外寒山寺 夜半鐘聲到客船	渭城朝雨浥輕塵 客舍青青柳色新 勸君更進一杯酒 西出陽關無故人
李白 下江陵 (粗體)	張繼 楓橋夜泊	王維 渭城曲
朝辭白帝彩雲間 千里江陵一日還 兩岸猿聲啼不住 輕舟已過萬重山	月落烏啼霜滿天 江楓漁火對愁眠 姑蘇城外寒山寺 夜半鐘聲到客船	渭城朝雨浥輕塵 客舍青青柳色新 勸君更進一杯酒 西出陽關無故人
李白 下江陵 (斜體)	張繼 楓橋夜泊	王維 渭城曲
朝辭白帝彩雲間 千里江陵一日還 兩岸猿聲啼不住 輕舟已過萬重山	月落烏啼霜滿天 江楓漁火對愁眠 姑蘇城外寒山寺 夜半鐘聲到客船	渭城朝雨浥輕塵 客舍青青柳色新 勸君更進一杯酒 西出陽關無故人
李白 下江陵 (粗+斜)	張繼 楓橋夜泊	王維 渭城曲
朝辭白帝彩雲間 千里江陵一日還 兩岸猿聲啼不住 輕舟已過萬重山	月落烏啼霜滿天 江楓漁火對愁眠 姑蘇城外寒山寺 夜半鐘聲到客船	渭城朝雨浥輕塵 客舍青青柳色新 勸君更進一杯酒 西出陽關無故人

圖8：張炳煌老師所手寫的唐詩，轉換成全真字型 9 point size所呈現的結果

8	9	10	11	12	14
淡江	淡江	淡江	淡江	淡江	淡江
16	18	20	24		
淡江	淡江	淡江	淡江		

圖9：各種不同大小的「淡江」兩字手寫字體

4 結論

當今流行的電腦字體雖然樣式非常多，但選擇好字體後，大家所列印的樣式都相同，缺乏個人化風格的字體。本論文嘗試提出一個全自動將個人書寫的中文字，經過影像處理技術，將之轉換成電腦所儲存的全真字型，具有個人化的風格，使得在螢幕顯示或是列印時，可以呈現獨特的個人書寫風格，讓電腦處理字型更加多采多姿。

致謝

感謝淡江大學中文系教授張炳煌老師對於此研究所提供的手寫文字稿。

參考文獻

- [1] 華康字型，<http://www.dynacw.com.tw/>。
- [2] 全真字庫，<http://tops.com.hk/av-font.htm>。
- [3] FontCreator，<http://www.high-logic.com/>。
- [4] M. Sarfraz and M.A. Khan, Automatic Outline Capture of Arabic Fonts, Vol. 140, No. 3-4, 269-281, 2002.
- [5] M. Sarfraz and M.F.A. Razzak, An Algorithm for Automatic Capturing the Font Outlines, Computers & Graphics, Vol. 26, No. 5, 795-804, 2002.
- [6] Microsoft, TrueType 1.0 Font Files, Technical Specification, Revision 1.66, 1995.
- [7] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 1, 62-66, 1979.
- [8] H. Baker, Computer Graphics with OpenGL, third edition, Pearson, 2004.
- [9] A. Rosenfeld and E. Johnston, Angle Detection on Digital Curves, IEEE Transactions on Computers, Vol. C-22, No. 9, 875-878, 1973.
- [10] A. Rosenfeld and J.S. Weszka, An Improved Method of Angle Detection on Digital Curves, IEEE Transactions on Computers, Vol.C-24, No.9, 940-941, 1975.