中華民國八十四年全國計算機會議論文集
Proceedings of National Computer Symposium 1995

基於連續點寬度的中文字筆劃抓取

Run-Length Based Chinese Character Stroke Extraction

林志文　　李錫智
J. W. Lin and S. J. Lee

國立中山大學電機研究所
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424, R.O.C.

摘要

本篇的目的在於解決因筆劃間複雜的相交情況所產生的錯誤特徵(feature)，以及因中文字的藝術外形所產生的多餘筆劃。我們的方法是利用直向與橫向掃描求取連線點寬度(run-length)來分隔寬度相似的區塊(block)，再以此求得筆劃的交叉區(fork section)和筆劃線段(stroke segment)。此方法除了大大地改善了筆劃間相交的特徵，並且還加入了過渡區塊(transitional block)和邊緣區塊(boundary block)來消除雜訊和多餘的筆劃.

關鍵詞：連續點寬度，橫向掃描，直向掃描，區塊，交叉區，筆劃線段，筆劃區塊，交叉區塊，過渡區塊，邊緣區塊

Abstract

Due to artistic shapes and complex intersections among lines, false features and spurious strokes may be extracted from handwritten Chinese characters. We propose a run-length based mehod in which run-length, obtained by horizontal and vertical scanning, are used to identify different blocks. Subsequently, fork sections and stroke segments can then be obtained. The method greatly improves the characteristic of intersections and the quality of features. Moreover, noise and spurious strokes can be eliminated with the introduction of transitional and boundary blocks.

Keywords : run-length,horizontal scan, vertical scan, fork section, stroke segment,stroke block, fork block,transitional block,boundary block

1. 介紹

特徵抓取(Feature extraction)在文字識別(character pattern recognition) 中佔有相當重要的角色，尤其是中文字的識別上更是重要，因爲特徵就代表著一個字所給予辨識系統的所有資訊。如果代表這個字的資訊無法適切的表達原來的字，那麼系統也就無從辨認起了。一般而言，中文字辨認所抓取的特徵是每個字的筆劃(stroke)與從筆劃間相互關係所得的資訊。目前咸認爲比較困難的問題在於筆劃交叉時，會產生許多複雜的情況，而可能導致原先一個筆劃被分割成相連的兩筆劃，或是多出一個連於原先筆劃中間的跨接筆劃(Necking)[1]。每個人千奇百怪的書寫習慣，更是加深了中文字的複雜度(complexity)。習慣上，我們將特徵抓取分成兩大類：一是經由消點程序(erosion)的細線化[2,3]。另一則不是經由消點程序的特徵抓取[4,5,6,7,8]。消點法的過程是將字的筆劃由外向內一層層的消去，只剩下筆劃的骨幹，它的發展原先是用於影像的辨別上，雖然發展上已臻成熟，可是在文字識別上卻是問題重重，諸如中空點(noise hole)，錯誤的捉取線(spurious projection)，兩線相交產生多餘的筆劃(tail generation at acute limb angle)[1]。另外有以計算每個 pixel 向周圍延伸的長度來定義交叉點及筆劃的方法[4]，在交叉點的抓取上有不錯的結果，但其複雜度卻是異常的高(每一點向四周延伸時，都幾乎要掃描整個字)。而另一種 run-length 的方法[5,6,7,8]，雖然比前者有著較快的處理速度，但因是比較相鄰 run-length 的差異度而做後續之處理，在抗雜訊度上就有點低了，而且在特徵的抓取上，如交叉點和筆劃抓取上，存在著很大的缺失。

本篇所提出的方法是將整個筆劃交叉的

地方設爲交叉區(fork section)，其餘的則爲筆劃線段(stroke segment)，以直向和橫向掃描分別取得直劃和橫劃的中心線後，以其共線(collinear)的程度將之連接，如此即可免除了筆劃因交叉所帶來的複雜性。另外我們定義了邊緣區塊(boundary block)來減少因中文字的藝術外形，如頓筆，會在筆劃的尾端產生多餘的線段[6,7]，而一些因雜訊而導致連續點寬度的判斷錯誤， 則以過渡區塊先行表示，再與其相鄰的區塊結合。如此我們不但在筆劃的交叉處有很好的特徵，甚至於每筆劃也有不錯的細線化結果.

2. 區塊

如圖 1 ，我們直覺上可以將一個字分成兩個部份，即筆劃線段(stroke segment)和交叉區(fork section)，而從交叉點和筆劃線段的連接情況，再將因交叉而分割的筆劃線段連接起來。

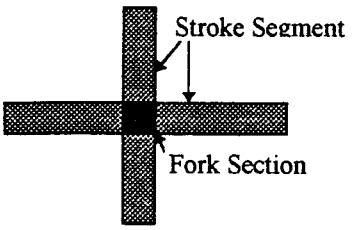

圖 1. Stroke Segment 和fork Section的關係。

2.1 區塊

一般來說，一個字筆劃的寬度變化是緩慢的，如此方能表現此一筆劃的連續性。如圖 2 所示，在橫向的筆劃上，我們以垂直的掃描方式，比較其相鄰左右的變化。首先我們定義在掃描過程中，每一個連續的點集合爲"連續點"(run)，如果左右的連續點寬度(run-length)變化小於或等於 2 點(pixel)時，其中心點便可以相互連接，所以我們便可將之歸爲同一個區塊(block)。設相鄰的兩連續點的寬度各爲 up-run-length 和 down-run-length ，則下列式子爲形成區塊的條件：

(up-run-length - down-run-length) < = 2

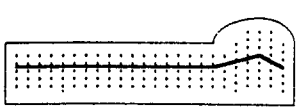

圖 2. 以垂直掃描取得橫劃的中心點。

2.2 雜訊的免除

在特徵抓取時，經常會遇到雜訊(noise)的干擾，如果雜訊是單獨存在的話，那是毫無疑問的可將之濾除。但若是在筆劃中受到雜訊的干擾，那麼就比較難以處理，例如無法將同一個筆劃線段歸於同一個區塊中，此時我們定義一個過渡區塊來解決此一問題。首先我們先定義一個臨界點： Stroke-min-length ，即是筆劃的最小點數，小於此點數的筆劃線段即可以視爲過渡區塊而與相鄰的區塊結合，在一般的應用上可設爲 1 或 2 即可得到良好的結果。而邊緣區塊則是筆劃邊緣連續點所形成的，因爲在中文筆劃的週圍經常會有一些修飾外形的圖點，或是由雜訊所附著在區塊上，而使得在作筆劃中心點抓取時會有些微偏移，所以必需先行將之濾除. 但如果是頓筆在結尾所形成的較大的多餘區塊，則需要另外再加以處理了。圖 3 顯示橫向掃描中表現邊緣區塊的情形。

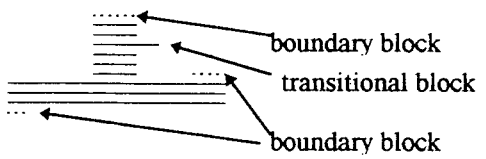

圖 3. 橫向掃描中出現邊緣區塊和過渡區塊的情形。

2.3 區塊的類別

我們可將抓取區塊的步驟粗分爲兩個過程，即是橫向掃描和直向掃描。在掃描過程中就每個區塊和其相鄰區塊的關係，定義如下幾個區塊的類別：

1)UP-FORK ：即表示上面有兩個非雜訊形成的區塊交叉在此一區塊中，其中非雜訊區塊的判斷是以區塊的連續點數(run-count)需大於或等於筆劃最小點數。其形成的條件如下：

up-fork-no >= 2
and every
up-block-run-count > Stroke-min-length
此種區塊如圖 4(a)所示。

2)DOWN-FORK: 即表示下面有兩個非雜訊形成的區塊交叉在此一區塊中，其中非雜訊區塊的判斷是以區塊的連續點數(run-count)需大於或等於筆劃最小點數。其形成的條件如下：

down-fork-no >= 2
and every
down-block-run-count > Stroke-min-length
此種區塊如圖 4(b)所示。

3)BI-FORK ：表示此一區塊爲筆劃的交叉點。其形成的條件如下：
up-fork-no >= 2 and down-fork-no >= 2
and every
up-block-run-count > Stroke-min-length
and every
down-block-run-count > Stroke-min-length
此種區塊如圖 4(c)所示。
4)LONG：代表交叉區塊(fork block)，表示此一區塊與相鄰上下兩個區塊的寬度相比，是較長的，其代表意義即是此一區塊有其它的筆劃穿越。此種區塊如圖 4(d)所示。
5)SHORT：表示此一區塊與相鄰上下兩個區塊的寬度相比，是較短的，其代表的意義即是此一區塊爲一個筆劃區塊(stroke block)，可由其中心點相連得出筆劃線段。此種區塊如圖 4(e)所示。
6)TRANS ：區塊的連續點個數(run-count)小於定義的最小筆劃點數時，或是區塊間的中心點相距太遠(大於 2 點以上)，即設此區塊爲過渡區塊(Transitional Block)，小於最小筆劃點數的過渡區塊與相鄰的區塊結合，而另外一種過渡區塊則視爲交叉區塊，即是無法構成一個筆劃區塊，因爲其中心點代表的筆劃線段不盡理想，需改由另一種掃描的筆劃區塊來求得筆劃線段。

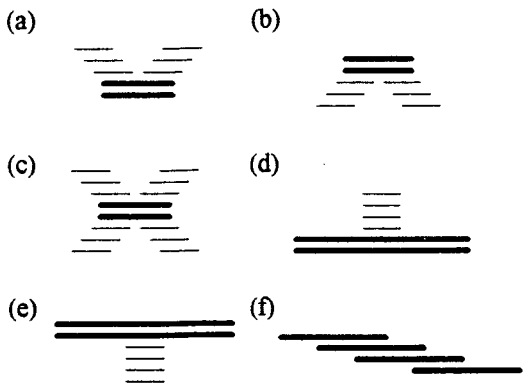

圖 4. 相鄰區塊間的區塊定義:(a)UP-FORK; (b)DOWN-FORK; (c)BI-FORK; (d)SHORT; (e) LONG; (f)TRANS 。

2.4 交叉區塊和筆劃區塊

在使用連續點寬度的比較後，我們可將區塊先粗分爲交叉區塊(fork block)和線段區塊(stroke block)。顧名思義，交叉區塊是用來表示此一區塊存在於交叉區，而線段區塊則表示此一區塊的中心點爲筆劃線段。因爲筆劃的寬度是緩慢變化的，如果發生了寬度急劇變化的時候，即代表著此筆劃與其它的筆劃有相交的情形，所以相鄰的兩區塊寬度較長的即是交叉區塊，較短的則是線段區塊. 我們不難由此窺見只要點所在位置的直向和橫向區塊都爲交叉區塊時，此點即是位於交叉區之內。圖 5 顯示了直向和橫向交叉區塊的形成情形。

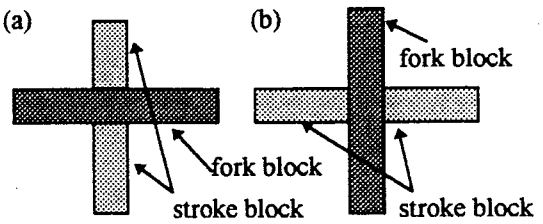

圖 5. (a)橫向掃描中，將整個字分成 3 個橫向的區塊，其中兩個爲 stroke block，一個爲 fork block.(b)直向掃描所分割區塊的情形。

2.5 筆劃線段

每個筆劃皆會有橫向掃描(H-Scan)的區塊、中心點(H-MID)和橫向掃描長度(H-run-length)，以及直向掃描(V-Scan)的區塊、中心點(V-MID)和直向掃描長度(V-run-length)。假如這個筆劃是橫劃則以 V-MID 來代表此一橫劃，而直劃則以 H-MID 來代表。而判定區塊是屬於直向掃描的區塊或是橫向掃描區塊可以用下列的方法:
1)假如橫向掃描的區塊是筆劃區塊，而直向掃描的區塊是交叉區塊，如圖 6(a)示，則取出以橫向掃描區塊的中心點爲筆劃線段，反之亦然。
2)假如直向與橫向的掃描區塊均是筆劃區塊(斜線的情形)，那麼則檢測橫向 H-MID 是否與交叉區相連，若是則判定此橫向的 H-MID 代表這個區塊的筆劃線段，直劃則相反。
3)假如直向與橫向的筆劃區塊均沒有與任何的交叉區相鄰，如圖 6(b)所示，則比較 H-run-length 和 V-run-length ，由兩者中，取寬度較小的來取出筆劃線段，但如果是非固定寬度的字，則以其首尾的 first-run-length 及 last_run-length 的和來做以上的比較：

H-first-run-length + H-last-run-length
> V-first-run-length + V-last-run-length

如上式成立時，我們便可將這個區塊的筆劃線段當成是以直向掃描 V-MID 表示的一橫線.反之則是橫向掃描 H-MID 表示的一直線。

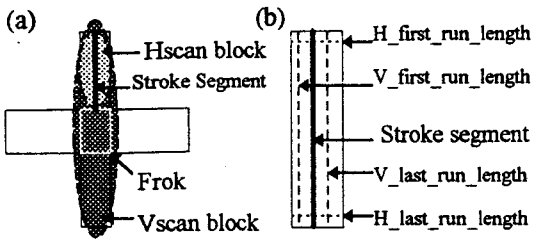

圖 6. (a)H-Scan block 因爲與交叉區相交，所以其所涵蓋的區域則以 H-Scan H-MID 來產生筆劃線段：(b)因爲 H-run-length < V-run-length 所以以 H-scan H-MID 來產生 Stroke Segment 。

3. 交叉區的處理

如圖 5 所示，在經過橫向掃描之後，我們可以明顯看出，整個字圖被橫向掃描切割成三個區塊，其中有 2 個是 stroke block ， 1 個是 fork block 。而在直向掃描時，我們亦可得到 2 個是 stroke block， 1 個是 fork block ，而所有直向與橫向 fork block 都有通過的地方就是交叉區了。

4. 筆劃擷取

我們可將各筆劃區塊所代表的筆劃線段及其和交叉區的互相的關係以圖 7 表示之：

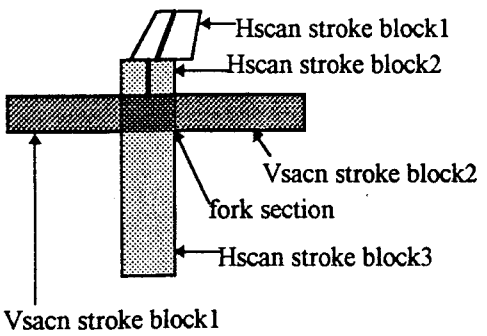

圖 7. 顯示筆劃區塊相互間的關係。

4.1 結合相鄰的筆劃

在受到部份雜訊干擾時，可能將原先的同一個筆劃線段分成兩個區塊，但仍能保有上下或左右相鄰的關係，如圖 7 所示，所以如果兩個相鄰的區塊的斜率互近的話，即能將之連接起來，而其判斷斜率是否相近的臨界值則視辨識時特徵的定義如何而決定。舉例來說：如果辨識的特徵需要將一個弧型的筆劃視爲兩個直劃，那它所要求的臨界值就必需低一點，我們將這個臨界值定義爲 min-collinear-gap 。

4.2 連接與交叉區相鄰的筆劃線段

因爲筆劃在交叉時，會被交叉點分成兩個筆劃線段，所以我們在做特徵抓取時，必需將之再連接起來，我們用以下的推論來做爲我們連接的依據，請參考圖 8 ：

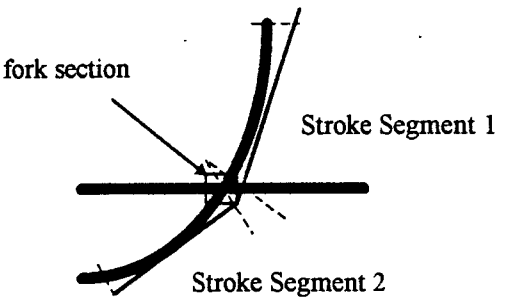

圖 8. 交叉點相鄰筆劃線段的相連

推論：如果 Stroke Segment 1 和 Stroke Segment 2 在交叉點的斜率相近的話， Stroke Segment 1 和 Stroke Segment 2 便會形成共線，如此我們便能有效的將與交叉區相連的筆劃線段有效地重新連接起來。

5. 演算法

我們的方法大致上可分爲 4 個主要的步驟： 1)取得橫向掃描的區塊並求得各區塊的種類； 2)取得直向掃描的區塊並求得各區塊的種類； 3)將橫向掃描與直向掃描的交叉區塊所交集的區域設爲交叉區，並求得各直向的筆劃區塊和橫向的筆劃區塊爲筆劃線段的地方，擇一代表這個筆劃線段； 4)將各筆劃區塊的中心點相連形成筆劃線段，並與相鄰的交叉區做連接。圖 9 、 10 展示上述的 4 個步驟。以下說明這 4 個步驟所用到的一些程序。

H-SCAN ：

1)將相鄰相同連續點寬度的區塊結合成一個區塊(但在 DOWN-FORK 下則否)，並濾除邊緣的連續點；

2)將上下皆有連接的過渡區塊與其上下的較長的區塊結合，若非上下皆區塊有與之相鄰，則視爲邊緣區塊濾除；

3)由區塊與上下的關係，求其區塊的類別.

V-SCAN ：

同上 H-SCAN 。

Get fork section:(H-Scan)

假如橫向掃描的區塊是交叉區塊且其連續點通過的直向掃描的交叉區塊，則將這些連續點設爲交叉區的點。

Modify fork section:(V-Scan)

假如直向掃描的區塊是筆劃區塊且其連續

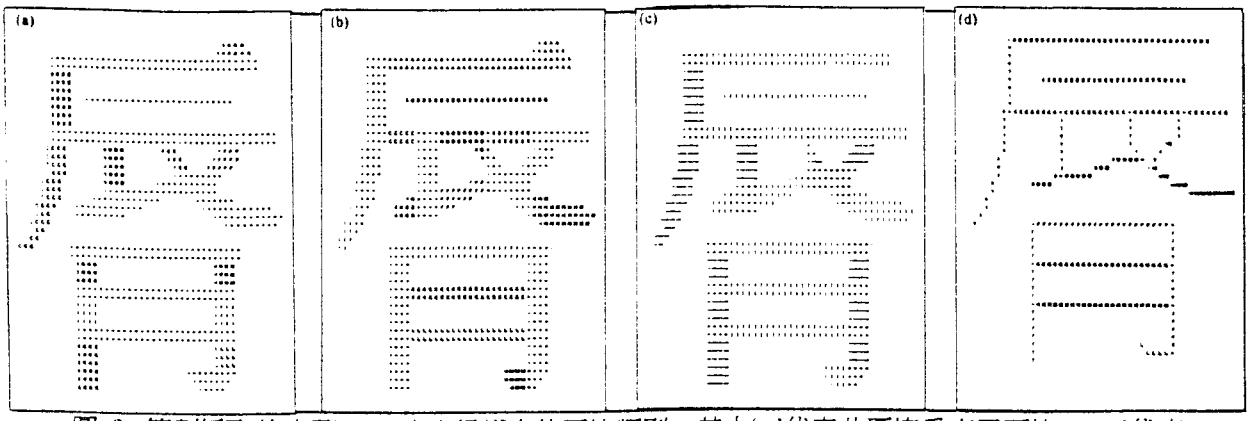

圖 9. 筆劃抓取的步驟：(a).先取得橫向的區塊類別，其中'+'代表此區塊爲交叉區塊，'·'代表已濾除的點，其餘的則爲筆劃區塊；(b).直向掃描的區塊類別；(c).以'|'代表此筆劃區塊需以直向掃描的中心點代表，'-'代表橫向掃描；(d).捉取結果。

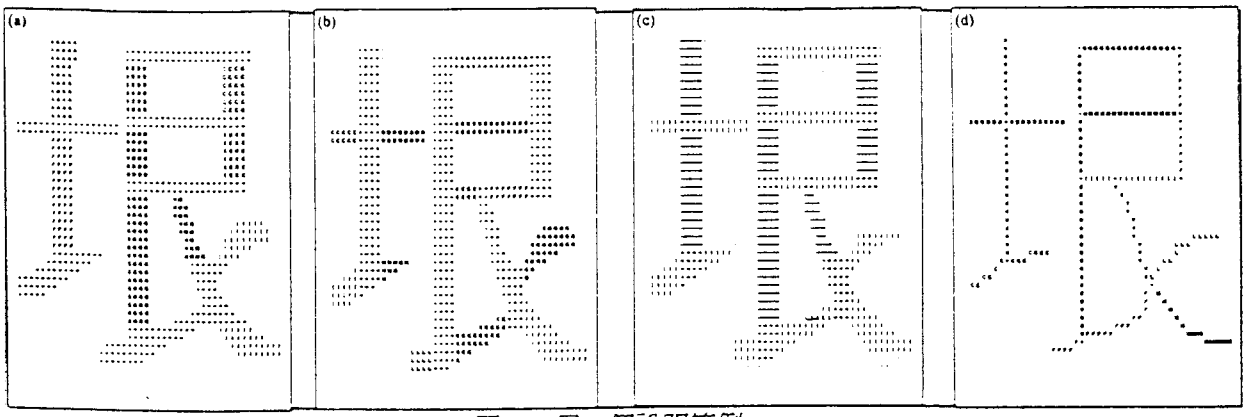

圖 10. 另一個說明範例。

點都通過交叉區和單一個橫向掃描區塊，則將通過的交叉區改爲直向掃描的筆劃區塊。

Remove artistic shape:

假如筆劃同時是橫向的筆劃區塊和直向的筆劃區塊，則可能有下列三種情形：

1)爲單一的區塊，沒有相鄰的區塊，此時要比較直向和橫向的連續點寬度，來決定是橫向的筆劃區塊或是直向的筆劃區塊來代表這個筆劃。

2)直向區塊完全包含橫向區塊，則橫向區塊即是中文的藝術外形(art shape)，如圖 11 所示，則可將之濾除；若橫向的區塊完全包含住直向區塊，則將直向區塊濾除。

3)直向與橫向區塊有相交但沒有完全包含，則此筆劃爲一斜劃，取有與交叉區相鄰的筆劃區塊代表這一筆劃。

Get H-scan stroke segment:(H-Scan)

假如橫向掃描的區塊類別是筆劃區塊，則取出其中心點的連線爲筆劃線段。

Get V-Scan stroke segment:(V-Scan)

假如直向掃描的區塊類別是筆劃區塊，則取出其中心點的連線爲筆劃線段。

Connect fork stroke:

先取得代表每個交叉區的交叉點(fork point)再將之與相鄰的筆劃線段相連，交叉點的求法有下列三種情形：

1)相鄰的筆劃線段沒有共線的情形產生，則取交叉區的幾何中點爲其交叉點；

2)有一組共線產生，則將共線的兩筆劃線段相連，其餘的筆劃線段則以其與共線相交的交點爲其交叉點；

3)有兩組以上的共線產生，將兩共線的筆劃線段相連，取兩共線的交點爲其交叉點。

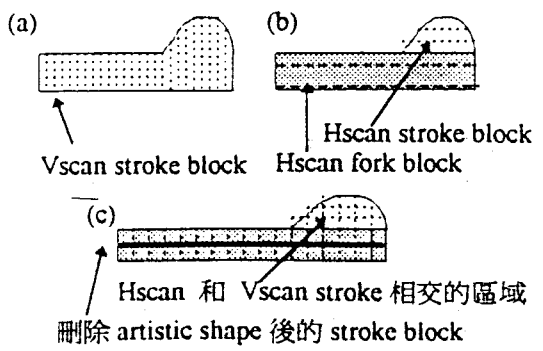

圖 11. 刪除中文字的 art shape 過程。

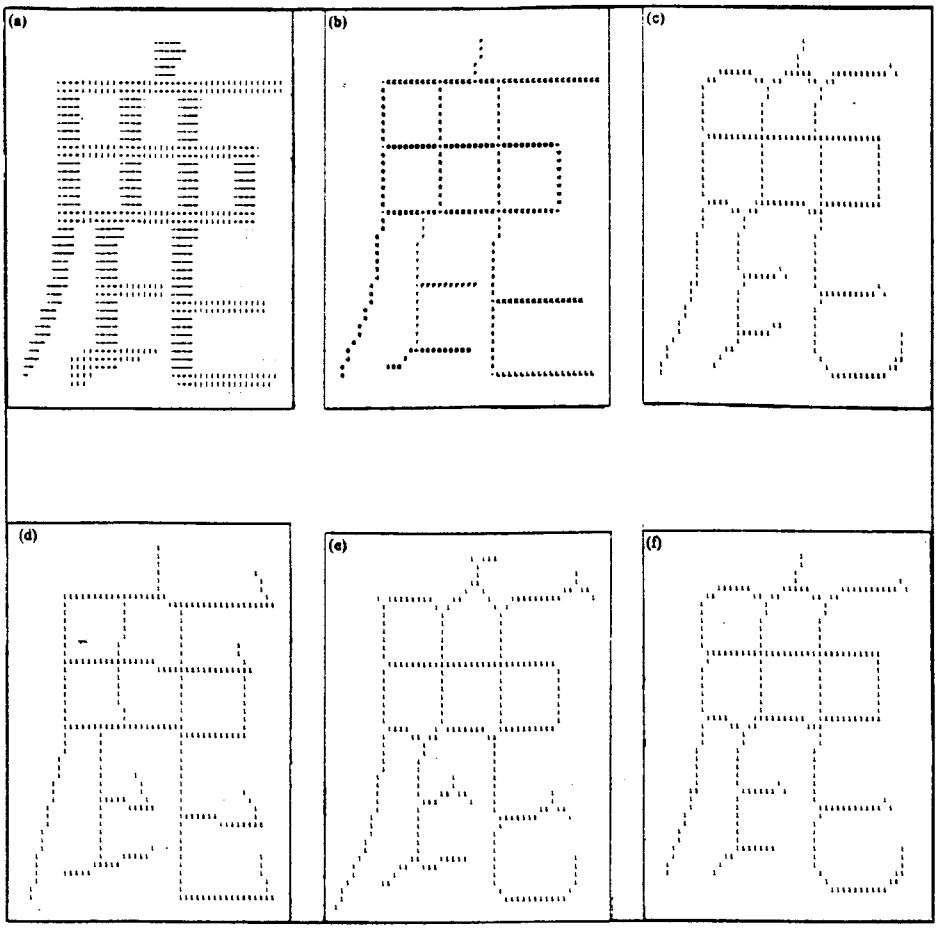

圖 12. 與其它演算法的比較：(a). 直向和橫向的筆劃區塊；(b). 我們的方法；(c). Zhang and Suen 方法；(d). Lin and Chen 方法；(e). Naccache 方法；(f). Wu and Tsai 方法。

6. 結果比較

在這一節裏我們做了兩種比較： 1)我們的方法和Zhang and Suen[3]、 Lin and Chan[6] 、 Naccache[9] 、 Wu and Tsai[10]相比較，如圖 12 、13 所示； 2)我們的方法和 Chen[4]作比較，如圖 14 所示。因爲在文獻上找不到共同的字，所以我們分開來做比較。我們的方法所顯示的是每個筆劃(stroke)的軌跡，其餘的方法顯則是顯示細線化的結果。在第一種比較中，顯然地(d)、(e)、(f)各圖的細線化結果跟原先的圖(a)的特徵(feature)相差甚多，而圖(c)卻因中文字的藝術外形出現了多餘的線段，另外在橫劃的捉取也會被分成兩個 4 方向(four connected)相交的線段，而我們的方法卻是完整的一橫線。在交叉點的選擇上也是較其它方法爲佳。在第二種的比較中，我們的方法在處理交叉點上，更能以有效的交叉點來代表筆劃交叉的情形。

7. 結論

在傳統的細線化過程，通常的辨識系統都需再將細線化後的結果，再抓取筆劃，在本篇中所提出的中文字的特徵抓取方法，除了可提供細線化的結果，亦可直接將筆劃直接取出，只需將筆劃區塊的首尾兩中心點相連即可，更省去了相當多的時間，除了複雜度控制在固定的掃描次數以外，所抓取的特徵也有下列幾項優點： 1)不會有因消去點方法所出現的不佳的細線化情形； 2)另外在交叉點的改善情況來講，也大大的除去辨識系統因交叉時的不良的筆劃抓取的麻煩； 3)除去中文字的藝術外形所多出的多餘筆劃； 4)加入了對抗雜訊的功能。

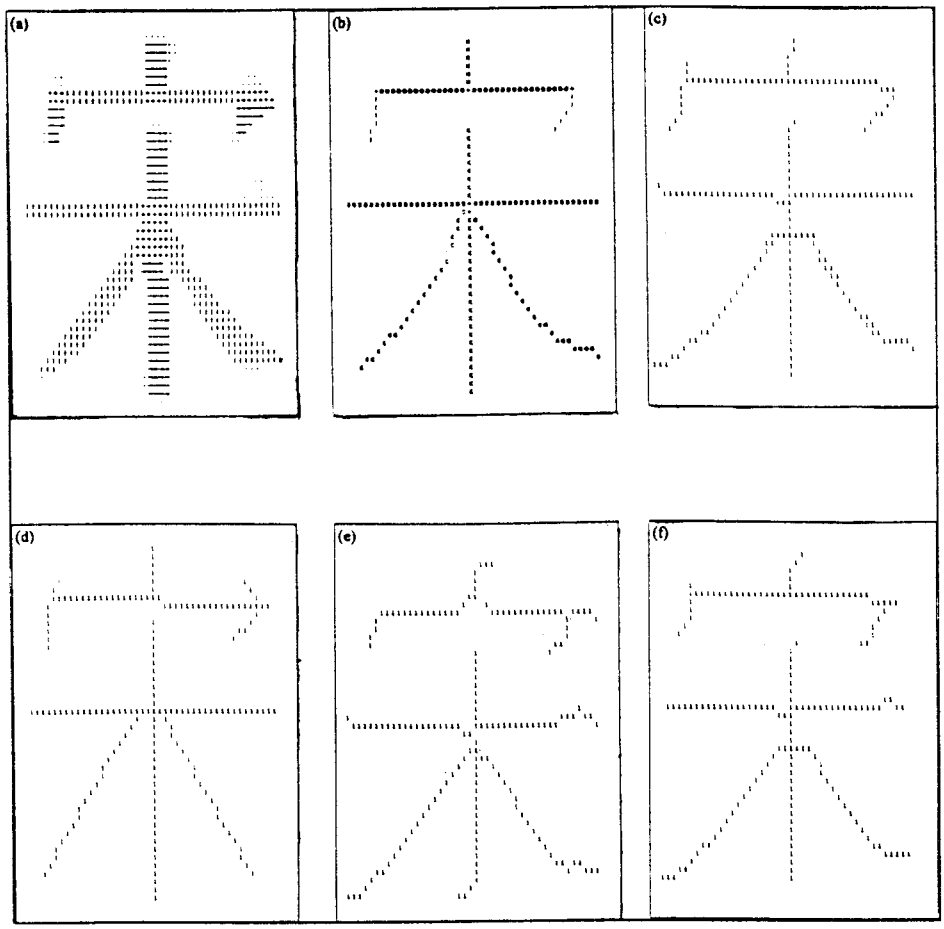

圖 13. 另一個比較：(a). 直向和橫向的筆劃區塊；(b). 我們的方法；(c). Zhang and Suen 方法；(d). Lin and Chen 方法；(e). Naccache 方法；(f). Wu and Tsai 方法。

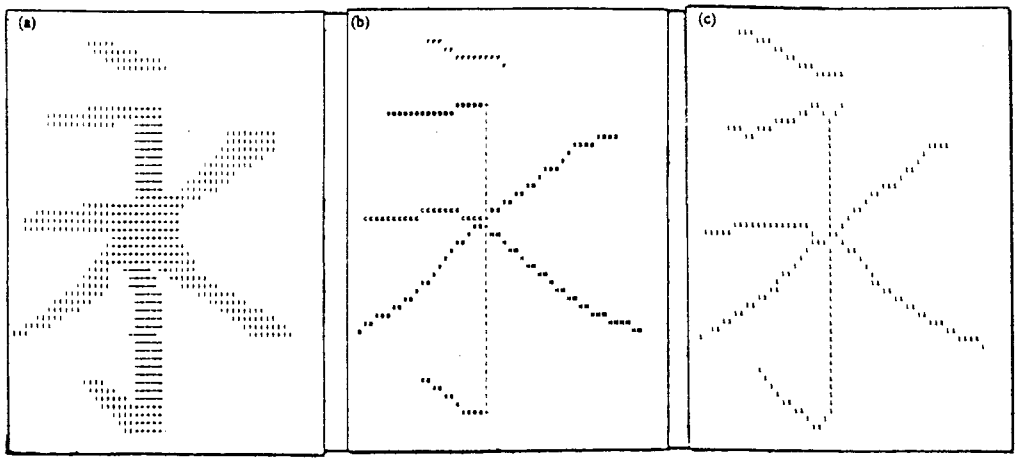

圖 14. 中文字"永"的比較：(a).直向和橫向的筆劃區塊；(b).我們的方法；(c).Chen[4] 方法。

8. 參考資料

1. F. W. M. Stentiford and R. G. Mortimer, Some new heuristics for thinning binary handprinted characters for OCR, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 13, No. 1, pp. 81-83,1983.
2. A. Data and S. K. Parrui, A robust parallel thinning algorithm for binary images, *Pattern Recognition*, Vol. 27, No. 9, pp.1181-1192, 1994.
3. T. Y. Zhang and C. Y. Suen, A fast parallel algorithm for thinning digital pattern, *Commun. ACM*, Vol. 27, pp. 236-239, 1984.
4. Y. S. Cheng, Segmentation and association among lines and junctions for line image, *Pattern Recognition*, Vol. 27, No. 9, pp. 1135-1157, 1994.
5. C. T. Chuang and L. Y. Tseng, A heuristic algorithm for the recognition of printed Chinese characters, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 25, No. 4, pp. 710-718, April. 1995.
6. J. Y. Lin and Z. Chen, A Chinese-character thinning algorithm based on global features and contour information, *Pattern Recognition*, Vol. 28, No. 4, pp. 493-512, 1995.
7. L.Y.Tseng and C.T.Chuang, An efficient knowledge-based stroke extraction method for multi-font Chinese character, *Pattern Recognition*, Vol. 25, No. 12, pp. 1445-1458, 1992.
8. L. Y.Tseng, A stroke extraction method for multi-font Chinese characters based on the reduced special interval graph, Technical Report, Dept. Applied Mathematics, National Chung Hsing Univ., Taiwan, 1992.
9. N. J. Naccache and R. Shinghal, SPTA:a proposed algorithm for thinning binary patterns, *IEEE Trans. on Systems, Man, and Cybernetics* , Vol. 13, No. 3, pp. 409-418, 1984.
10. R. Y. Wu and W. H. Tsai, A new one pass parallel thinning algorithm for binary images, *Pattern Recognition Lett.* ,Vol. 13, pp.715-723,1992.