

二值化影像之線萃取

梁世昌 陳文儉

大葉大學資訊工程學系

{e9406001, cwj}@mail.dyu.edu.tw

摘要

經過二值化的工程圖影像，再進行細化與線萃取，不僅大幅降低資料儲存量，自動建立點與線的鄰接位置和相對關係，更提供自動檢驗系統、圖像識別系統以及向量化(Vectorization)等應用。

本研究針對工程圖影像的特性，提出新的細化處理、線萃取的演算法，實驗結果證明本研究採用的方法較既有的各種方法，具有較高的可靠性，以及較佳的成果品質。

關鍵詞：細化、影像分割、向量化、弧點資料模式。

Abstract

Thinning and linear extraction of binary images not only reduces data storage amount, automatically creates the adjacency and relativity between line and points but also provides applications for automatic inspection systems, pattern recognition systems, and vectorization.

Based on the features of construction drawings, new thinning and linear extraction algorithms were proposed in this study. The empirical results showed that the proposed method has a higher reliability and produces better quality than the various existing methods.

1. 導論

工程圖影像的線特徵萃取，一直是影像自動檢測系統、圖像識別系統，以及影像資料向量化系統的重要步驟。但是，隨著精確度與處理速度的要求日益提高，如何從工程圖影像中萃取出線特徵，構建點與線的幾何資料，再以高效率的結構來儲存，是一項極為重要的課題。

二值化之後的影像中，物件以 1 或黑色表示，背景則以 0 或空白表示。通常，工程圖包含了數量龐大的線條，細線所佔的比例較高，

但是線寬分佈並不平均，想要正確地找出線的中軸(Medial axis)加以萃取，在人類視覺系統中是一件輕而易舉的事，但在數位影像處理則是相當複雜的演算。

本研究使用骨架化(Skeletonization)方法，將二值化影像的線細化。骨架化是以逐層剝除的方式，將影像中的物件，細化成為一個像點寬度的線條，而且必須保留二值化影像的所有圖形結構性資訊，例如線條的位置、方向和長度。

經過骨架化將影像內的線都細化之後，物件像點會被大量的減化。接下來必須將影像中各自獨立的物件像點，與相鄰物件像點串接，組成一群一群的線特徵。最後，適度地修正線特徵的連續性與平滑性，以較精簡及高效率的方式儲存所有的線特徵。

因此，線細化處理以及線特徵萃取，是本研究的兩個主要的程序。在線細化處理方面，本研究加強了特殊線型辨識，使得直線的每個像點，減少多餘的鄰接像點。在線特徵萃取方面，本研究採用向量資料的弧點資料模式(Arc-Node Data Model)，以節點陣列和線串陣列來表現線物件的位置和相對關係，以及不同維度的成果。

本文所提出的二值化影像線萃取方法，經由實際開發之系統實驗結果證明，具有良好的成果品質。在第二節將說明線細化處理之相關研究及改良的方法；在第三節中將說明線特徵萃取之相關研究及本研究提出的新方法；實驗結果與結論將分別在第四節與第五節。

2. 線細化處理

2.1 線細化相關研究

Zhang & Suen[1]所提出的二值化影像中軸擷取方法，是利用多次迭代將線條兩邊逐次縮減，直到剩下一個像點寬度。每一次的迭代再

分為兩個子迭代，第一個子迭代判斷處理中的物件像點的八鄰接像點的連結關係，符合條件時將處理中的物件像點先標記刪除；在第二個子迭代時，判斷已被標記刪除的像點，當符合條件將該像點清除為背景，否則清除標記，整個子迭代若都未清除任何物件像點，則結束細化作業。這個方法速度快，又可以用於多處理器的平行運算。

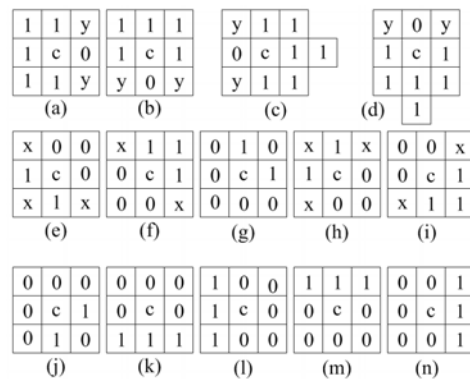
Holt 等人[2]提出了改良 Zhang & Suen[1]的方法，除了取消了兩個子迭代之外，並且完全利用邏輯表示式的運算，取代相鄰物件像點數的計算，以加快處理速度。完成邊緣物件像點清除為背景之後，接著還要再利用鋸齒狀樣式進行判斷，以清除骨架化後多餘保存的像點。Wu & Tsai[3]進一步改良 Zhang & Suen[1]與 Holt 等人[2]的演算法。提出了下列四項細化的目標：1. 必須保留物件的連接性，2. 不可過度侵蝕，3. 具抗雜訊能力，4. 成果須接近原影像形狀。利用 12 個 3×3 及一個 3×4 及 4×3 的細化模型如“圖一”所示，逐一比對物件像點的鄰接分佈，若處理中的物件像點符合 14 個模型的任何一個，則清除為背景。迭代進行細化，直到未發生物件像點異動為止。

Deng 等人[4]提出了改良 Wu & Tsai[3]的方法，在平行細化演算清除物件像點時，滿足下列三項基本要求：1. 不可清除端點，2. 不可違反連接性，3. 不可過度侵蝕。除了沿用 Wu & Tsai[3]的 14 個細化模型，以清除邊緣(edge)物件像點。此外，當滿足 14 個其中的 8 個模型時，必須再依新提出的條件判斷某些鄰接像點是否須要加入凹型像點(Concave pixel)的標記陣列。被標記為凹型像點的，再利用八鄰接距離 $d_8(p, q) = \max(|x_p - x_q|, |y_p - y_q|)$ 進行確認後予以清除。

Huang 等人[5]提出了依據八個鄰接像點中的物件數量，從 2 個到 7 個的鄰接數中，總共列出了 50 個 3×3 的細化模型，符合任一個模型時，除非滿足另外 6 個 4×4 及 3×4 及 4×3 的保存模型，否則清除為背景像點，並迭代進行直到沒有異動為止。這個方法可以應用在經常有封閉輪廓的字元辨識。

Ahmed & Ward[6] 將旋轉不變(Rotation invariant)的特性，進行影像細化處理時，利用物件特徵予以分類，在各分類下再檢查多個不同的 3×3 細化規則，進行字元及中文的細化。

Rockett[7]是利用圖結構的折點連接定義，以及折點的鄰接陣列等方式，提出了改良 Ahmed & Ward[6]的方法。Patil 等人[8]也在 2005 年針對 Ahmed & Ward[6]的方法，提出適用在指紋辨識及符號的演算法，利用 21 個 3×3 的細化規則，以及 4 個斜線規則，使得只有一個像點寬的指紋紋路位於線中央，而且沒有因細化而斷裂；另外的 24 個偵測紋路分叉點(Ridge bifurcation)的規則，更可去除錯誤的分叉點，強化指紋辨識的應用領域。



1=物件，0=背景，x=可為 0 或 1，y=至少 1
個為 0，c=處理中的物件像點

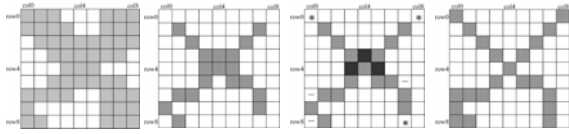
圖一 Wu & Tsai[3]的 14 個細化模型。

2.2 本研究的線細化處理

本研究的線細化處理可分為兩個步驟，首先利用 Wu & Tsai[3]的方法，由上而下由左而右，逐一掃描物件像點，找到物件像點時將其相鄰像點進行“圖一”的 14 個模型的比對，若是符合任一模型時，即將物件像點標註刪除。掃描完所有的像點後，將被標註為刪除的物件像點清除為背景，並進行下一次的迭代作業，直到沒有像點被標註刪除為止。

“圖二”是一張 9×9 的二值化影像，塗色的格點表示物件，空白的格點表示背景像點。最左側的圖(a)是細化前，圖(b)則是 Wu & Tsai[3]細化後的結果。如同 Deng 等人[4]所提出的問題，Wu & Tsai[3]的方法，在凹型像點時會有多餘的物件像點被保存，例如“圖二(c)”深色塗色標示出的 (col4,row3)、(col3,row4)、(col5,row4)等三個像點，在“圖二(b)”的相同位置上仍有網線，表示 Wu & Tsai[3]無法清除這類的凹型像點。“圖二(d)”表示本研究所期望達

成的最後結果，除了上述凹型像點之外，還有”圖二(c)”標示出-及*記號的像點須處理。



(a) 細化前 (b) 細化後 (c) 處理前 (d) 期望結果
圖二 9×9 影像細化實例。

Deng 等人[4]的方法在每次的迭代中，掃描的像點只要符合(a)-(n) 模型之一時，將像點清除為背景時，就須判斷模型類別是否屬於(a)-(f)及(h),(i)等八種模型，接著再依各類別的條件式檢查鄰接像點，將符合條件的候選凹型像點加入標示陣列之中，完成每次的迭代後再比對”候選凹型像點標示陣列”，清除其中的凹型像點。

本研究提出的解決方法，是在完成 Wu & Tsai[3]方法的細化之後，再另外執行一次迭代，由上而下由左而右掃描細化後的物件像點，將該像點的 8 鄰接特徵，比對”圖三(a)”的凹型像點模型，若是符合任一個可旋轉方向的凹型像點模型，則判定該處理中的物件像點必須立即予以清除。本研究的方法，由於只要在完成 Wu & Tsai[3] 方法的細化之後進行唯一的一次掃描，又不須額外的陣列空間，條件判斷式的運算量又低，因此，處理速度較快。

0	1	x
1	c	0
0	1	z

x	1	0
1	c	0
x	1	0

1=物件, 0=背景
x=可為0或1
c=處理中像點
z=與x的值相反

(a) 新模型

P ₈	P ₁	P ₂
P ₇	P	P ₃
P ₆	P ₅	P ₄

$$N_4 = P_1 + P_3 + P_5 + P_7$$

$$N_8 = P_2 + P_4 + P_6 + P_8$$

$$x=1,3,5,6 \text{ if } P(x)=0$$

$$P_0=P_8, P_9=P_1$$


```

if N4 = 3 {
    if N8 = 1 and (Px-1=1 or Px+1=1)
        P = 0 /* P is a concave pixel */
    if N8 <= 2 and (Px-1=1 and Px+1=1)
        P = 0 /* P is a concave pixel */
}

```

(b) 條件式

圖三 凹型像點判斷模型及條件式。

”圖三(a)”的兩個凹型像點模型，都有四種旋轉方向，總計必須比對八種 3×3 的模型，以辨識出凹型像點。本研究為了減化比對計算的複雜度，將各個旋轉方向的凹型像點模型轉換

為”圖三(b)”的條件運算式，利用極少數的加法運算及邏輯判斷，可大幅提昇處理速度。

本研究雖然解決了凹型像點的問題，但是在工程圖向量化時，上述的細化處理後會產生兩種常見的問題，必須在線萃取的階段(Phase)才能加以修正：第一類問題是線轉彎處的折點，會由短線段連接而造成缺口，例如”圖二(c)”的(col0,row6)、(col0,row8)、(col7,row5)等位置，圖上標示出-的三個像點。第二類問題在線端點會被清除，造成細化後的線長度縮短，例如”圖二(c)”的(col0,row0)、(col8,row0)、(col7,row8)等位置，標示出*的三個像點。

3. 線特徵萃取

3.1 線特徵萃取之相關研究

Bachnak & Celenk[9]將細化後的二值化影像進行彎角偵測，逐一比對所有物件像點為中心的 3×3 模型(Templates)，若 8 相鄰中至少有 3 個物件像點時為候選彎角，須再進一步判斷確認後，記錄彎角的幾何資訊。接下來，再建立彎角間的物件像點，取得直線的位相性質，以完整的表示 2-D 影像。

Venkateswar & Chellappa[10]將精簡邊緣偵測 (Canny edge detector)處理後的邊緣影像 (Edge image)，由左而右上而下(LRTB)依序掃描，將位於同一多邊形線上的邊緣像點，標註相同的線編號，利用空間索引串連出多邊形。

Chiang[11]利用迭代程序找尋線段的最大內切圓，再用端點最大內切圓(MIC)的直徑計算線段的寬度，線邊沿點與直徑垂直方向則可計算出線段的斜率，進而取得線段的中心像點，完成線影像向量化作業。

Haron 等人[12]從已經完成影像二值化、細化及數位化處理，而且已被編碼為八鄰接的多邊形鏈碼影像中，取得每段編碼的斜率，分析出彎角的像點坐標位置。

3.2 本研究的線特徵萃取

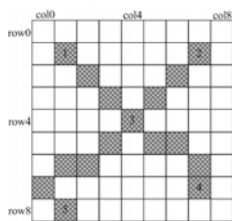
傳統的線萃取，必須利用 5×5 或 7×7 的鄰接模型，辨識出線交叉點及線端點，再使用鏈碼(chain code)描述影像。不僅處理速度不理

想，也無法有效的處理線轉彎處空洞、線端點縮短等細化產生的問題。

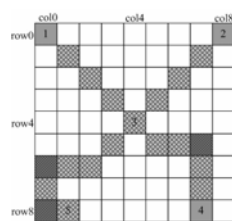
本研究提出的線特徵萃取方法，可分為兩個步驟，第一個步驟是建立弧點資料模式；第二個步驟是修正弧點資料內容。弧點資料模式是將上一個階段的線細化成果，利用鄰接關係將物件像點，串接出向量式資料模式的節點和線串。節點包含了端點及交叉點，如”表一”所示；線串的兩端為節點中間為折點串列，線串之間不可以相跨越，如”表二”所示。採用弧點資料模式，不僅可以快速地組成線特徵，儲存點及線的位相紀錄，更可提供彈性的資料維護能力，提供強固穩定的向量化加值應用。

步驟一：建立弧點資料時，須由左而右由上而下(LRTB)掃描整個細化後二值化影像，當找到物件像點時，則向其鄰接的物件像點串接，直到線串的兩端為端點或是線交叉點為止。最後以節點陣列”表一(a)”與線串陣列”表一(b)”的方式，儲存全部的線特徵，完成整個線萃取的作業。演算法說明如下：

- (1) 尋找線串 L_{arcs} 的起始物件像點 P_s ，計算 P_s 的八相鄰數
 若 $N_8=2$ ，將 P_s 加入 L_{arcs} 的折點串列 P_{vertex}
 若 $N_8=1$ 或 $N_8 \geq 3$ ，將 P_s 加入節點陣列 P_{nodes}
- (2) 由 P_s 向兩端找相鄰像點，直到遇到節點
- (3) 記錄 L_{arcs} 的起點與終點節點陣列號及折點串列
- (4) 若還有未處理的物件像點，執行(1)
- (5) 輸出 P_{nodes} 及 L_{arcs}



(a) 細化後結果



(b) 線特徵萃取修正後

圖四 9×9 影像細化結果。

”圖四(a)”是 9×9 影像細化結果，其中標示出 1-5 數字的像點是節點，兩節點間的塗色像

點連接成線串。以”圖四”內容為例，弧點資料的輸出結果列示於”表一”及”表二”。

表一 節點陣列(P_{nodes})

Node ID	col	row	type
1	1	1	End point
2	7	1	End point
3	4	4	Junction
4	7	7	End point
5	1	8	End point

表二 線串陣列(L_{arcs})

Node ID(start)	Vertices (col., row)	Node ID(end)
1	(2,2) (3,3)	3
2	(6,2) (5,3)	3
3	(3,5) (2,6) (1,6) (0,7)	5
3	(5,5) (6,5) (7,6)	4

步驟二：修正點資料時，首先逐一處理線串陣列的每筆紀錄，如果線串內折點方向改變，而且前後兩線段的延伸交點在細化前為物件像點，例如”表二”第四筆資料的折點(6,5)到(7,6)，則須將延伸交點插入折點內，以修補線折點的空洞。所有必須修正的像點，標示於”圖四(b)”的三個黑色位置。修正後的線串陣列於”表三”。

表三 修正後的線串陣列

Node ID(start)	Vertices (col., row)	Node ID(end)
1	(2,2) (3,3)	3
2	(6,2) (5,3)	3
3	(3,5) (2,6) (1,6) (0,6) (0,7) (0,8)	5
3	(5,5) (6,5) (7,5) (7,6)	4

接著修改節點位置時，只處理節點形態屬於端點的資料。因此，線串陣列內節點為端點者，判斷端點與兩個鄰接折點的行與列的差值，當三點是一直線時，則往端點方向延伸一個像點，比對該像點在細化前是否為物件像點，成立時則修改節點陣列內該節點的位置。上述直線延伸增長線端點，必須重複進行，一直到細化前為背景像點為止。所有必須修正的像點，標示於”圖四(b)”的節點編號(Node ID.)為 1 及 2 及 4，修正後的節點陣列如”表四”所示。

表四 修正後的節點陣列

Node ID	col	row	type
1	0	0	End point
2	8	0	End point
3	4	4	Junction
4	7	8	End point
5	1	8	End point

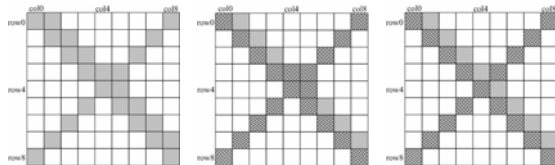
為了提昇資料儲存以及後續應用的效率，線串陣列必須再進行資料減量的處理，將每一筆線串內的直線上折點予以移除。例如”表二(b)”第一筆資料，起始的節點編號為 1，終止的節點編號為 3，兩節點間的折點(2,2)及(3,3)，線串的四個點為一直線，須清空折點例如”表三(b)”的第一筆資料的 nil。經過資料減量後，所有的弧點資料修正如”表三”。

表五 本研究作業結果的線串陣列

Node ID(start)	Vertices (col., row)	Node ID(end)
1	nil	3
2	nil	3
3	(2,6) (0,6) (0,8)	5
3	(3,5) (7,5)	4

4. 實驗結果

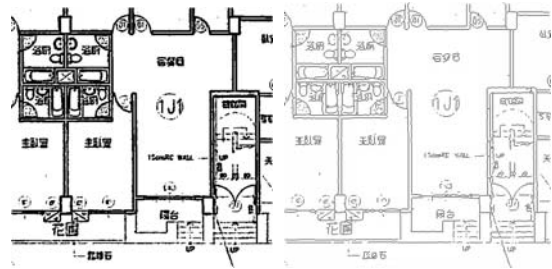
為了測試一個像點寬度以及兩個像點寬度的線細化品質，利用 9×9 的二值化的原始影像“圖五(a)”，使用 Wu & Tsai[3]的方法細化後為(b)，(c)則是使用本研究的方法細化的結果。兩個方法都可以雖然都可以骨架化，但是只有本研究的方法，比較接近中軸，而且可以在線萃取處理時，正確地的分辨出一個交叉點，將該像點位置加入節點陣列內。



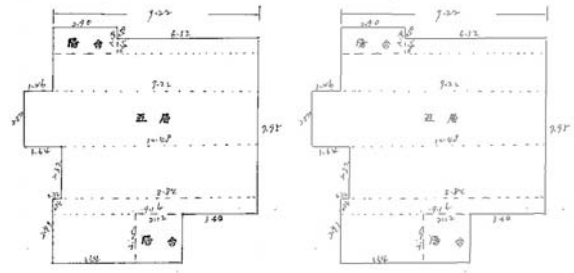
(a)原始影像 (b)方法[3]細化 (c)本研究細化

圖五 本研究細化的另一範例。

“圖六”是常見的工程圖範例，左圖是二值化後的原始影像，右圖是利用本研究的方法細化處理後結果。



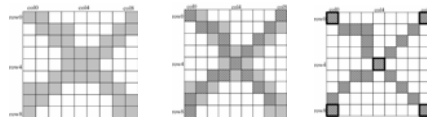
(a) 藍晒圖(Blueprints)



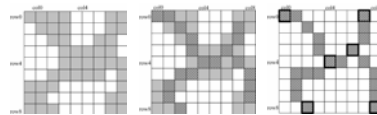
(b) 手繪圖(Hand drawings)

圖六 工程圖 512×512 影像。

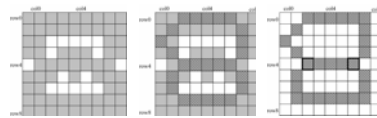
“圖七”的(a)到(d)，由左而右分為 9×9 的二值化的原始影像、本研究的細化結果、本研究的線萃取結果。有格網的像點是線串，其中方框加上粗線的是節點。



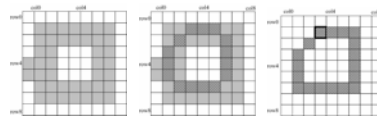
(a) 交叉點較寬



(b) 不同方向的線



(c) 複雜的封閉多邊形



(d) 封閉多邊形寬度變化

圖七 本研究提出的細化與線萃取範例。

5. 結論

本研究針對工程圖具有的大尺寸以及大量線條資訊的特性，提出的細化以及線萃取的作業方法與流程。經過實際開發應用程式的驗證結果，本研究提出的二值化影像線萃取方法，可以獲得良好的成果品質以及精確度，所建立的點與線的位相資訊，對後續向量化應用提供了極佳的助益。

參考文獻

- [1] T.Y. ZHANG,C.Y. SUEN,"A Fast Parallel Algorithm for Thinning Digital Patterns", Image Processing and Computer Vision, 1984, Vol.27, No.3,pp. 236-239.
- [2] CHRISTOPHER M. HOLT, ALAN STEWART, MAURICE CLIENT, RONALD H. PERROTT,"An Improved Parallel Thinning Algorithm", Image Processing and Computer Vision,1987,Vol. 27, No.3,pp.156-160.
- [3] Rei-Yao Wu,Wen-Hsiang Tsai, "A new one-pass parallel thinning algorithm for binary images", Pattern Recognition Letters 13(1992) , pp. 715-723.
- [4] Weian Deng, S. Sitharama Iyengar, Nathan E. Brener, "A Fast Parallel Thinning Algorithm For The Binary Image Skeletonization",The International Journal of High Performance Computing Applications. Vol.14, No.1, Spring 2000, pp. 65-81.
- [5] Lei Huang,Genxun Wan,Changping Liu, "An Improved Parallel Thinning Algorithm", Proceedings of the Seventh International Conference on Document Analysis and Recognition(ICDAR'03).
- [6] Maher Ahmed,Rabab Ward, "A Rotation Invaient Rule-Based Thinning Algorithm for Character Recognition", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.24, NO.12,pp. 1672-1678.
- [7] Peter I. Rockett," An Improved Rotation-Invaient Thinning Algorithm", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.27,NO.10, pp.1671-1674.
- [8] Pradeep M. Patil,Shwkar R. Suralkar,Faiyaz B. Sheikh," Rotation Invaient Thinning Algorithm to Detect Ridge Bifurcation", Proceedings of the 17th IEEE International Conference on Tools with Artificial (ICTAI'05).
- [9] Rafic Bachnak, Mehmet Celenk, "A CORNER DETECTION-BASED OBJECT REPRESENTATION TECHNIQUE FOR 2-D IMAGES", Intelligent Control, 1988. Proceedings., IEEE International Symposium on ,pp.186-190.
- [10]V. Venkateswar, Rama Chellappa, "Extraction of Straight Lines in Aerial Images", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.14, NO.11, NOVEMBER 1992,pp. 1111-1114.
- [11] John Y. Chiang, "A New Approach For Binary Line Image Vectorization", IEEE International Conference on Vol.2, Oct.1995,pp.1489-1494.
- [12] HABIBOLLAH HARON,SITI MARIYAM SHAMSUDDIN, DZULKIFLI MOHAMED, "A new corner detection algorithm for chain code representation", International Journal of Computer Mathematics, Vol.82, No.8, August 2005, 941-950.