

國立交通大學

資訊科學系

碩士論文

以 Gradient Vector Flow 與圖形演算法
為基礎尋找中軸

Centerline Generation based on Gradient Vector Flow and
Graph Algorithms

研究生：黃雅榆

指導教授：荊宇泰 博士

中華民國九十四年七月

以 Gradient Vector Flow 與圖形演算法為基礎尋找中軸

學生：黃雅榆

指導教授：荊宇泰 副教授

國立交通大學資訊科學研究所



中軸指的是在物體中央，可以用來描述物體的細長結構，因為中軸使用精簡的方法來表示物體，所以在影像分析和圖形辨識上中軸都有很大的幫助。本文提出了一個使用 Gradient Vector Flow 將欲尋找出中軸的物體影像轉換成具有方向性的圖形後，利用圖形的演算法以及中軸具有各種特性來尋找中軸的演算法。這個演算法兼具了處理二維影像及三維影像的能力，而且不同維度的影像，僅在轉換為圖形的部分過程中有所差異，而後在圖形的處理上則不需另外顧慮到影像維度的問題。最後尋找出來的中軸，即為圖形中的一部份，使用具有方向性的圖形來表示中軸，在之後的中軸的應用上有正面的幫助。

關鍵字：中軸、Gradient Vector Flow、圖形演算法

Centerline Generation based on Gradient Vector Flow and Graph Algorithms

Student: Ya-Yu Huang

Advisor: Dr. Yu-Tai Ching

Department of Computer and Information Science
National Chiao Tung University



Centerline is a thin structure which is located in the center of an object. It is useful in image analysis and shape recognition because centerline uses simple structure to describe object. We propose a method that constructs directed graph from image vector field generated by Gradient Vector Flow and then uses graph algorithms to build centerline by several properties of it. Since this method is based on graph, the only difference between two dimensional and three dimensional images is how to transfer from image to graph. After the transformation, the dimension of the image would not be very important to this method. The result of the centerline is presented as part of the directed graph which is useful for future application.

Keywords: Centerline, Gradient Vector Flow, Graph Algorithm

誌謝

非常感謝荆宇泰老師這兩年多來的指導，讓我在學業上和研究上都獲得了許多。

感謝醫學影像實驗室的學長同學和學弟妹，待在實驗室的兩年非常充實而且快樂，跟大家一起討論研究各式各樣問題的日子是非常珍貴的回憶。

感謝我的父母親，給我一個不管遇到什麼困難都可以回去的家。

感謝其他沒有提到的一直給我支持鼓勵的朋友們，謝謝你們。



目錄

中文摘要	ii
英文摘要	iii
誌謝	iv
目錄	v
圖目錄	vi
第一章	序論	1
1.1	簡介.....	1
1.2	研究目的.....	2
1.3	論文架構.....	2
第二章	相關研究	3
2.1	前言.....	3
2.2	Topological Thinning.....	3
2.3	Distance Map.....	5
2.4	Voronoi Diagram.....	7
第三章	以 Gradient Vector Flow 為基礎尋找中軸.....	10
3.1	Gradient Vector Flow Field.....	10
3.2	利用 GVF 來轉換影像為 Graph.....	11
3.3	使用 Graph Algorithm 來尋找中軸.....	13
3.3.1	尋找中軸的候選人.....	13
3.3.2	將中軸候選人相互連結成中軸.....	14
3.3.3	由中軸的候選人向外發展中軸.....	16
3.4	結論.....	19
第四章	實驗結果	21
4.1	前言.....	21
4.2	二維影像實驗結果與分析.....	21
4.3	三維影像實驗結果.....	28
第五章	結論與未來展望	29
參考文獻	30

圖目錄

【圖 1】[9]中字母 A 和平行處理的 Topological Thinning 演算法得出的中軸結果，圖中每個立方體表示的是影像中的一個 voxel。.....	4
【圖 2】[14]中的中軸結果，圖 A 為原圖，圖 B 及圖 C 為使用不同的數學型態法所得到的中軸結果。.....	5
【圖 3】內切圓與中軸的關係.....	6
【圖 4】[2]中的左圖顯示每個點跟邊界的歐式距離，右圖為計算每個點跟鄰近的點的中央差異梯度後得到的梯度向量空間。.....	7
【圖 5】黑色的點形成的 Voronoi Graph.....	7
【圖 6】圖 a 為原圖，圖 b 至 e 分別為樣本點為 10、50、100 和 200 個樣本點所得到的中軸結果，圖 f 顯示出雜訊對中軸的影響。.....	8
【圖 7】左圖為 Gradient Vector Flow Field，右圖為左圖中央部分的放大圖。.....	11
【圖 8】每個向量以八個標準向量中最接近的兩個標準向量來表示。.....	12
【圖 9】在 graph G 中，區域中向量都指向中間，中間的點的向量所指向的方向就會形成 Strongly Connected Component。.....	13
【圖 10】紅色的部分為尋找出來的 Strongly Connected Component。.....	14
【圖 11】紅色的部分為 SCC，綠色的部分表示兩個 SCC 中間的最短路徑，箭頭表示的是每個點向外的向量方向。.....	15
【圖 12】綠色的部分為將 SCC 互相連接起來的路徑。.....	16
【圖 13】在 graph G 裡，中軸兩側靠近邊界的點以及同在中軸上的點，都會存在向量指入中軸上的點。.....	17
【圖 14】紅色的點為 SCC，藍色的點為 InDegree 大於三的点。.....	17
【圖 15】藍色的部分為向外尋找得到的中軸。.....	19
【圖 16】尋找中軸的演算法流程.....	20
【圖 17】左圖為欲尋找中軸的物體，右圖為左圖經過 GVF 處理後的結果。.....	22
【圖 18】紅色部分為尋找出來的 Strongly Connected Component，藍色部分為 InDegree 大於三的点。.....	23
【圖 19】中軸的結果，綠色部分為將 SCC 彼此互相連接的中軸，藍色部分為由 SCC 向外發展的中軸。.....	24
【圖 20】左圖為真實的黑白楓葉影像，右圖為左圖經過 GVF 處理後的結果。.....	25
【圖 21】紅色部分為尋找出來的 Strongly Connected Component，藍色部分為 InDegree 大於三的点。.....	25
【圖 22】中軸的結果，綠色部分為將 SCC 彼此互相連接的中軸，藍色部分為由 SCC 向外發展的中軸。.....	26
【圖 23】左圖為用來尋找中軸的長方形立方體，右圖則是尋找出來的中軸結果.....	28

第一章

序論

1.1 簡介

中軸的概念最早是在 1967 年由 Blum[3]所提出，物體的中軸指的是位在物體中間的細線，可以用來描述物體和表示物體的結構。因為中軸使用精簡的結構來表示物體，減少了視覺上和計算上的複雜度，所以能夠運用在很多情形上。在醫學影像中，建立人類大腸的中軸可以在使用內視鏡時協助控制攝影機方向和移動；另外在影像分析和圖形辨識裡，中軸也有很大的幫助，像是可以幫助偵測影像中的血管以方便作影像切割。

尋找中軸一直以來都是熱門且具有挑戰性的工作，在過去很多文獻提出尋找中軸的演算法，但是都沒有標準的數學定義來決定何謂中軸，一致的部分只有中軸的三個基本條件，分別為中軸必須位於物體的中央 (centered)、中軸是細的 (thin) 和中軸要互相連結 (connected)。一般認為中軸是二維影像中物體每個區域裡最大內切圓的圓心的集合，三維影像則是物體每個區域裡最大內切球的球心的集合，但這樣的條件還不足以形成精簡的中軸。

除了上述的三個基本條件外，因應著不同的資料來源，以及尋找出中軸的後續應用，會在演算法中增加一些額外的條件來配合，以便找出適當的中軸，例如要能不受邊界的雜訊影響、要能夠從中軸重建 (reconstruct) 原本的物體、尋找中軸的演算法需要有好的效率等等。這些不同的尋找中軸的演算法，連帶的影響到中軸的結果，同樣的物體使用不同的演算法，可能會得到差距很大的結果。

因為中軸缺乏明確定義，許多演算法只針對某些特定形狀的物體來尋找中軸，醫學影像上最常見的是管狀的物體，對象為人類的腸和血管[5,6]。對於一些形狀較為複雜的物體，和三維影像中的物體，就缺乏高效率的中軸演算法。然而事實上隨著數

位醫學影像造影技術的快速發展，三維影像的影像分析愈來愈顯的重要，研究的領域也不僅是針對管狀的物體。本篇論文利用原本作為影像分割之用的 Gradient Vector Flow 為基礎，提出一個可以對二維影像和三維影像做處理的中軸演算法。

1.2 研究目的

過去文獻提出的尋找中軸演算法多半是分別對二維影像或三維影像來做處理，或是只尋找特定形狀物體的中軸。得到的中軸結果也沒有一個完整的資料結構來表示，於是無法針對類似物體的中軸結果來互相做比較。

本篇論文提出一個可以通用在二維影像和三維影像上各種不同物體的方法，並且嘗試使用樹狀的結構來表示中軸，希望可以提供一個有效率的演算法來尋找中軸，以及提供比較類似物體中軸結構的依據。

1.3 論文架構

本篇論文共分為五章，第一章為中軸的簡介、研究動機跟目的及論文架構，第二章將過去文獻中各種尋找中軸的演算法分為三類，分別加以整理，第三章介紹本篇論文提出的尋找中軸的演算法，以及原本用於影像分割但為本篇論文演算法的基礎—Gradient Vector Flow，第四章為使用本篇論文提出的演算法對不同物體尋找中軸的實驗結果，第五章為結論以及未來研究的方向。



第二章

相關研究

2.1. 前言

過去的文獻中提出了很多尋找中軸的演算法，中軸的條件也隨著不同的需要而有些不同。最基本的條件為中軸要在物體的中間、中軸要盡量的細但依然要保持互相連結，其他額外的條件如要能夠從中軸重建原本的物體等。即使針對基本的三個要求，如何去解釋「在物體中間」或是「怎麼樣算是互相連結」，也會隨著各種不同的尋找方法而有所差異。

本章將過去的尋找中軸的演算法分類成以下三種：Topological Thinning、Distance Map 和 Voronoi Diagram，其中 Topological Thinning 和 Distance Map 這兩種方法是離散型的方法，得到的中軸也是離散的，也就是說在二維情形下，中軸是一群以八鄰近（8-connected）為標準相連結的點；Voronoi Diagram 則是連續型的方法，不需要另外將資料來源轉換到離散空間，只要取邊界的樣本點來計算即可。以下將對這三種方法作更詳細的介紹。

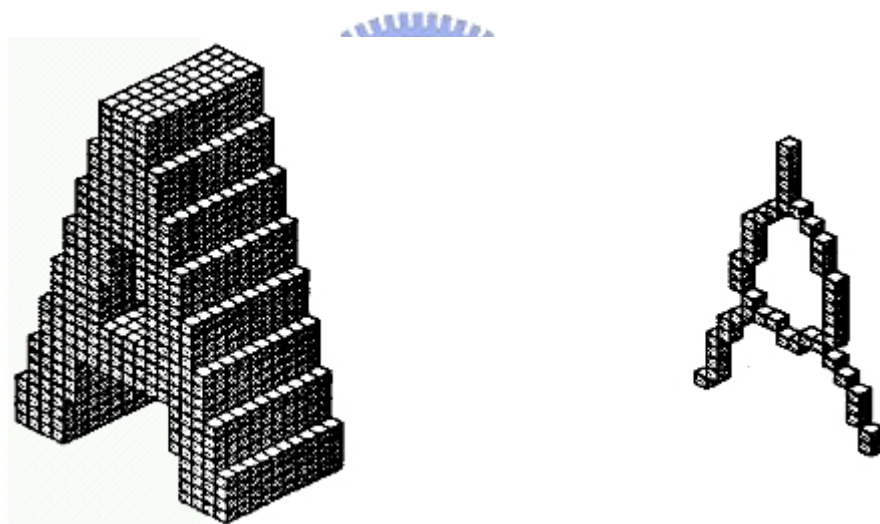
2.2. Topological Thinning

Topological Thinning 演算法又稱 Boundary Peeling 演算法或 Onion Peeling 演算法，這個演算法的基礎概念是由外而內，一層一層的將物體邊界的點去除掉，不斷的重複去除點的步驟直到留下來的點都是中軸才停止。

在去除點的過程中，必須要維持物體的連結性（connectivity）和拓撲結構（topology）不會改變，所以只有不會對這兩者產生影響的點，又稱 simple point，才

可以安全的被移除。除此以外，也需要避免不小心去除掉物體尾端的點，防止中軸過份的精簡而無法表示出原來物體的結構，例如一個資料來源為類似字母 b 的物體，尋找出來的中軸不應該像是字母 o，否則就是太過精簡，不是良好的中軸結果。

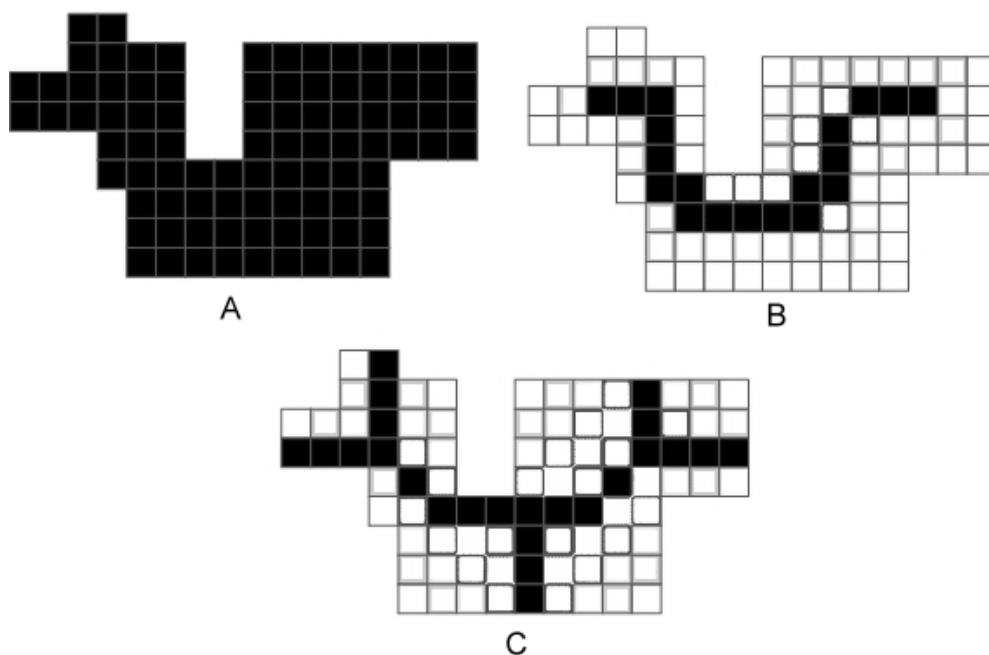
使用 Topological Thinning 演算法可以得到相當精確的中軸，加上條件的限制，可以保證得到的中軸結果一定是互相連結的。但是判斷每個點是否可以移除和將可移除的點移除這兩個步驟有很高的時間複雜度，所以在處理大量的點時，需要花費相當多的時間在這兩個步驟的運算上，因此有許多研究都著重在如何提高 Topological Thinning 演算法的效率。Sonka, et al. [9]提出可以平行處理的 Topological Thinning 演算法，設計出一次判斷出多個可移除的點的方法並將這些點移除掉，同時也保證移除掉的點不會影響連結性和拓撲結構，因此可以大幅加快計算的速度。



【圖 1】[9]中字母 A 和平行處理的 Topological Thinning 演算法得出的中軸結果，圖中每個立方體表示的是影像中的一個 voxel。

另外一種類似 Topological Thinning 的演算法為利用結構元素 (structure element) 來進行 erosion 及 dilation 的數學型態法 (morphological method) [11]，使用這個方法找到的中軸稱為數學型態中軸 (morphological skeleton)。數學型態法也可以平行處理，和 Topological Thinning 演算法最大的不同是數學型態中軸通常不會互相連結，不過數學型態法可以利用特殊的結構元素來得到互相連結的中軸，而且得到的數學型態中軸

結果會跟平行處理的 Topological Thinning 演算法所得到的中軸結果類似。



【圖 2】[14]中的中軸結果，圖 A 為原圖，圖 B 及圖 C 為使用不同的數學型態法所得到的中軸結果。

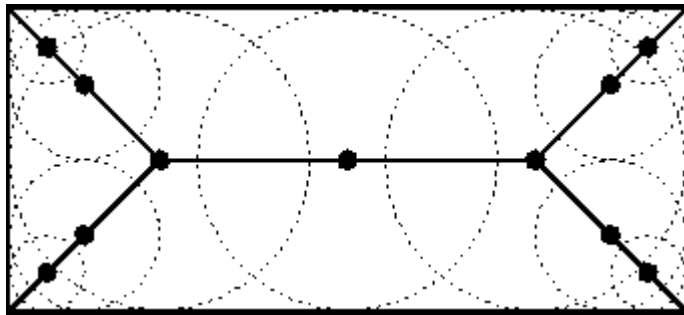


2.3. Distance Map

Distance Map 的演算法首先對輸入的物體進行 Distance Transform，計算出每個點的距離值（Distance value）後，利用區域性的特徵來尋找中軸上的點的集合：例如選擇距離邊界最遠，距離值為區域中最大的點，再將這些點相互連結起來成為中軸。各種不同的距離值演算法可以有很多變化，可以選擇不同的 Distance Transform 方法來計算距離值：像是給定起始點（source），計算物體中其他點與起始點的距離；或是以邊界為基準，計算物體中每個點和最近的邊界的距離。另外一個對中軸結果有很大影響的不同之處是計算距離的方式，標準的歐式距離（Euclidean distance）計算速度緩慢，為了增加演算法的效率，常會使用其他計算速度較快的距離計算方法來替代，算出近似歐式距離的值，這些方法有 2D 的 2-3 metric、3-4 metric，3D 的 3-4-5 metric 等等，不同的距離計算方法會直接的影響到中軸的點在物體中的位置，使得最後計算出來的

中軸結果有所不同。

以邊界為基準所計算出的歐式距離值在每個區域中的最大值，幾何上可視為區域中最大內切圓的圓心（三維影像為內切球的球心），最大的內切圓不會包括在其他的內切圓中，如果這個內切圓的圓心移動，直覺上該圓將會縮小以保持留在物體之中。理論上所有的最大內切圓的圓心都是中軸的候選人，只是需要再將它們互相連結才能形成中軸。

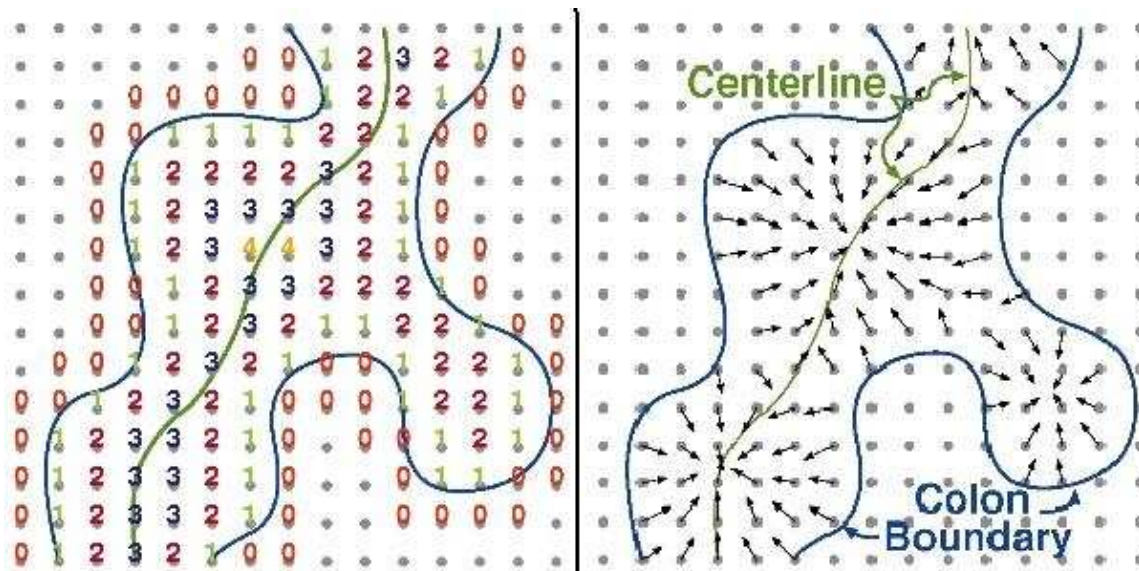


【圖 3】內切圓與中軸的關係

各種不同的 Distance Map 演算法還有一個很大的不同之處，就是如何將已經選出的各個中軸上的候選人（candidate）互相連結起來形成完整的中軸。Niblack, et al. [10] 定義了 Distance Map 中的 saddle point 作為連接候選人的點。若 Distance Map 中某個點的鄰近點會接連著出現距離值大於該點的距離值、距離小於該點的距離值、接著又是大於該點的距離值，這種點就是所謂的 saddle point，利用這個定義尋找出 saddle point 後，便可利用這類點的特性將區域中的最大值連結起來得到中軸。

一般的 Distance Map 演算法大都使用類似 Niblack, et al. [10] 的方法，找出區域中的最大值作為候選人後，再定義出符合特徵的 saddle point 作為連結候選人的依據。Bitter, et al. [2] 則混合使用了各種演算法來尋找中軸，首先使用速度較快的距離演算法計算每個 voxel 跟邊界的實際歐式距離後，再用每個 voxel 和鄰近六個 voxel 的距離值的差來計算中央差異梯度（central difference gradient），由每個點的梯度可以建立出一個梯度向量空間（gradient vector field），之後再利用這個梯度向量空間標出中軸的

候選人，此方法中，連結候選人的演算法則更為複雜。這個演算法使用了梯度的觀念，跟本篇論文提出的方法有類似之處，在第三章會有更詳細的比較。

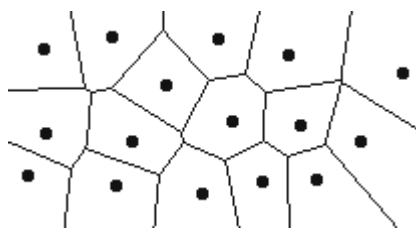


【圖 4】[2]中的左圖顯示每個點跟邊界的歐式距離，右圖為計算每個點跟鄰近的點的中央差異梯度後得到的梯度向量空間。

2.4. Voronoi Diagram

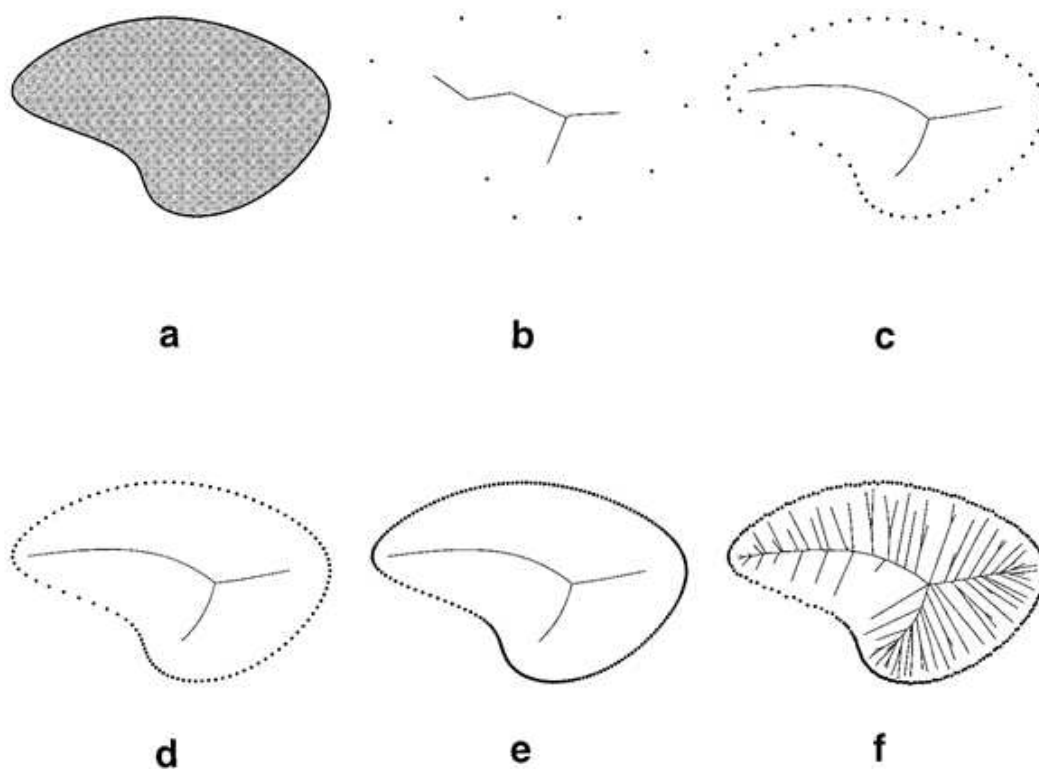
Voronoi Diagram 的定義為，令 E 為 R^N 中的有限集合 (finite set)， p 是 E 中的任意一點，則在 p 的 Voronoi Region 中的任意一點，與 p 的距離會比任何一個其他在 E 集合中的點的距離來的近，以數學式來表示即為：

$$V(p) = \{m \in R^N, d(m, p) \leq d(m, E)\} \quad (1)$$



【圖 5】黑色的點形成的 Voronoi Graph

從圖 5 可以發現，被 Voronoi Diagram 分割的區域中的每個點，與該區域中的樣本點的距離為最近，線上的點則與最近的兩個樣本點的距離是相同的，這個特徵跟中軸上的點一樣。D.T.Lee [8]證明了多邊型 (polygon) 的完整中軸是把多邊形的每個邊做 generalized Voronoi Diagram 的 subgraph。由於多邊形的邊是連續的 (continuous)，大部分的演算法都是在物體的邊界取樣本 (sample) 點，再使用樣本點做 Voronoi Diagram 來尋找中軸，因此樣本點取樣的數量便會直接的影響到結果。如果樣本點取樣的方法不佳，有可能會改變中軸表示出來的拓樸結構。圖 6 為 Attali, et al.[1]的演算法中，取不同數目的樣本點所得到的中軸結果。



【圖 6】圖 a 為原圖，圖 b 至 e 分別為樣本點為 10、50、100 和 200 個樣本點所得到的中軸結果，圖 f 顯示出雜訊對中軸的影響。

因為 Voronoi Diagram 演算法是連續型的尋找中軸方法，會使用 Voronoi Diagram 來尋找中軸的，主要是在幾何上為連續的物體，理論上使用 Voronoi Diagram 演算法得

到的中軸會是互相連結的。但實際上大部分的影像都會有雜訊，導致得到的中軸會類似上圖 5 的 f 小圖的狀況，產生出大量複雜的分支。需要再設計演算法適當的將不需要的分支砍掉，但砍分支的演算法容易把中軸尾端的部分也當作分支一併砍掉，這是設計去除分支的演算法困難的地方。另外使用 Voronoi Diagram 演算法尋找三維影像的中軸時，容易得到大量的平面並形成複雜的結構，提高後續處理的困難度。



第三章

以 Gradient Vector Flow 為基礎尋找中軸

3.1 Gradient Vector Flow Field

在第二章提到的 Bitter, et al. [2] 演算法，使用了 Distance Map 來建出一個梯度向量空間，並利用此梯度向量空間進行中軸候選人的尋找，並作為連結中軸的依據。在醫學影像的影像切割方法裡，Chenyang et al. [12] 提出了 Gradient Vector Flow Field (GVF)，可以將影像轉換成向量的資料，作為尋找 Active Contour [7]（又稱 snake）的外力依據。本篇論文就是將資料來源經過 GVF 的處理後，利用得到的 GVF Field 來進行之後的中軸尋找步驟。

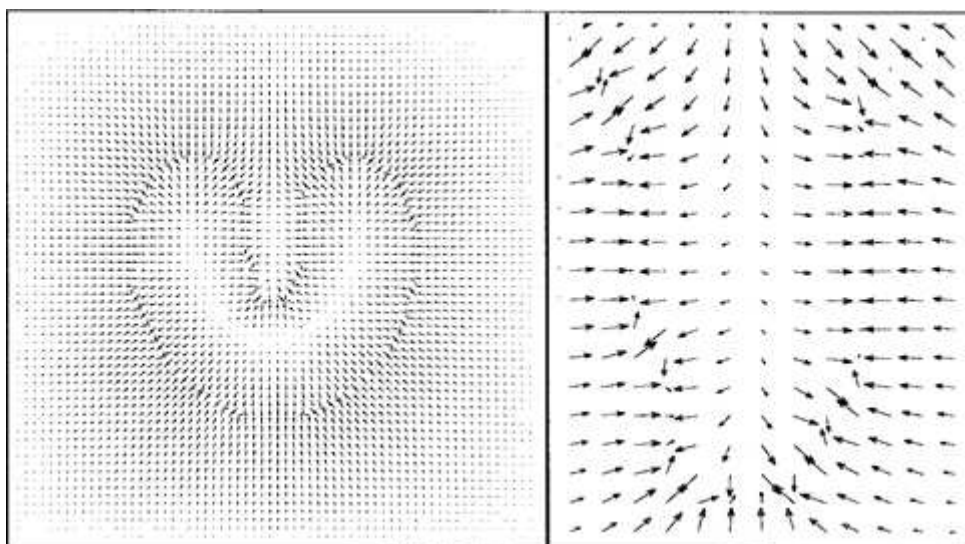
GVF 的定義為一個向量空間 \mathbf{v} ， $\mathbf{v}(x,y) = [u(x,y), v(x,y)]$ ，可以使得下面的能量函式（energy function）最小化。這個多變量（variational）的能量函式需要遵守一個標準的原則，當原始影像沒有資料的時候，得到的結果必須是平滑的（smooth）。

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dx dy \quad (2)$$

在詳細的介紹這個能量函式之前，要先介紹 GVF 的基礎—edge map。Chenyang et al. [12] 使用 $f(x,y)$ 來表示 edge map， $f(x,y)$ 是由影像 $I(x,y)$ 中計算出來的，它的性質是在影像中的邊界部分會有較大的數值。Edge map 有三個對 GVF 來說很重要的基本性質：首先，edge map 的梯度（gradient） ∇f 的向量指向邊界，在靠近邊界的地方向量會與邊界互相垂直，但在邊界裡面向量則是指向邊界的中央部分，這一點是本章之後的演算法主要會利用到的性質。其次，離邊界的距離愈近，這些向量的向量值愈大。最後，

在同質性高 (homogeneous) 的區域裡， ∇f 的向量值接近於零。

仔細的來分析這個能量函式，當 $|\nabla f|$ 的值很小時，能量是由對向量空間 \mathbf{v} 的偏微分的平方和結果所控制，結果會得到變化緩慢的區域；當 $|\nabla f|$ 的值很大時，函式的第二項控制了積分的結果，並且會在 $\mathbf{v} = \nabla f$ 時得到最小值。這樣的設計使得當 edge map 的梯度值很大時， \mathbf{v} 幾乎會跟梯度值相等，但在同質性高、梯度值很小的區域中， \mathbf{v} 的變化會很緩慢。參數 μ 是用來控制式子中第一項和第二項之間權重關係，需要視影像中的雜訊 (noise) 來調整大小，當雜訊愈多時， μ 值設的愈大。

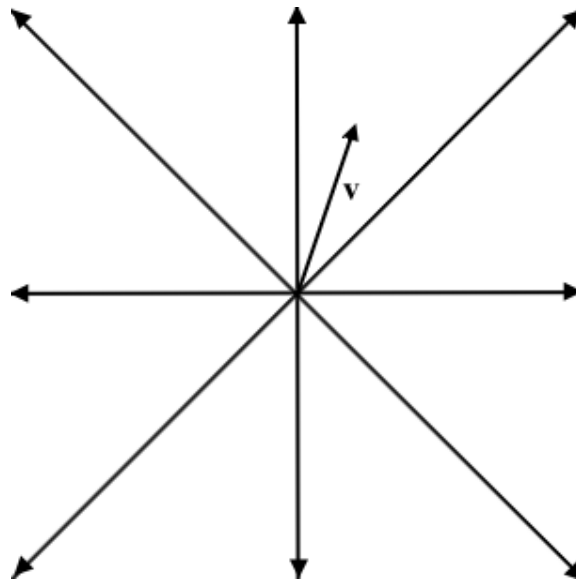


【圖 7】左圖為 Gradient Vector Flow Field，右圖為左圖中央部分的放大圖。

3.2 利用 GVF 來轉換影像為 Graph

GVF 的特性是邊界的內部靠近邊界的地方，會有跟邊界接近垂直而且方向是背離邊界的向量，離邊界愈近向量的值愈大，離邊界愈遠則向量值愈小。一般要做影像分割的影像中，物理的邊界是白色的，所以要尋找中軸的物體在影像中也就以白色來表示，再將此影像進行 GVF 的運算。利用這個特性，計算出來的 GVF 資料中每個向量的方向將指向中間，也就是形成一張類似圖 2-a 的梯度向量影像，這些向量最後會指到物體的中間，也就是中軸的位置。

計算出影像的 GVF 資料以後，採取的步驟是將影像轉換成 graph 來處理。將轉換過後的 graph 稱為 G ，一般 graph 的表示方法是 $G=(V,E)$ ， V 代表的是 graph 中的 vertex (點)， E 代表的則是 edge (邊)。在 graph G 裡面，點的數目是影像中位於要尋找中軸的物體內部的點的數目，邊則是利用 GVF 的資料來建立。以二維影像的情形為例，利用八個鄰近點的八個標準向量來表示 GVF 資料中每個點的向量方向—每一個向量都取距離該向量最近的兩個標準向量來代表它。也就是說，每個向量都是用兩個標準向量來表示，所以每個點最多有兩個向外連的邊 (OutEdge)。對物體內的每個點都進行這個步驟，就可以把 GVF 的資料轉換成一張由點和邊所組成的 graph G 。



【圖 8】每個向量以八個標準向量中最接近的兩個標準向量來表示。

三維影像在分割區域的部分較為複雜，若要使用 26 個鄰近點來計算，要分成等分的區域大小會增加計算上的複雜度。所以採取較為簡單的分區法：首先以 z 軸為基準，分為 z 值為正和 z 值為負的兩區，再將這兩區分為如二維影像的八區，八區再以 45 度的仰角分成上下兩區，總共分為 32 區。所以仰角大於 45 度的區域最接近的向量有三個，仰角小於 45 度的區域最接近的向量則有四個，完成分區以後，使用和二維影像相同的方法將三維影像轉換成 graph。

經過這個轉換過後，不管是二維影像或是三維影像，都會形成一張標準的 graph

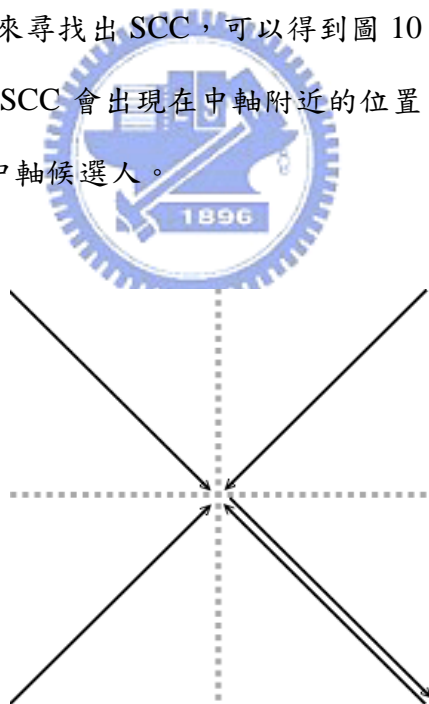
G ，因此之後的演算法就不需要另外再對二維或三維做不同的處理，這是把影像轉換成 graph 的優點。

3.3 使用 Graph Algorithm 來尋找中軸

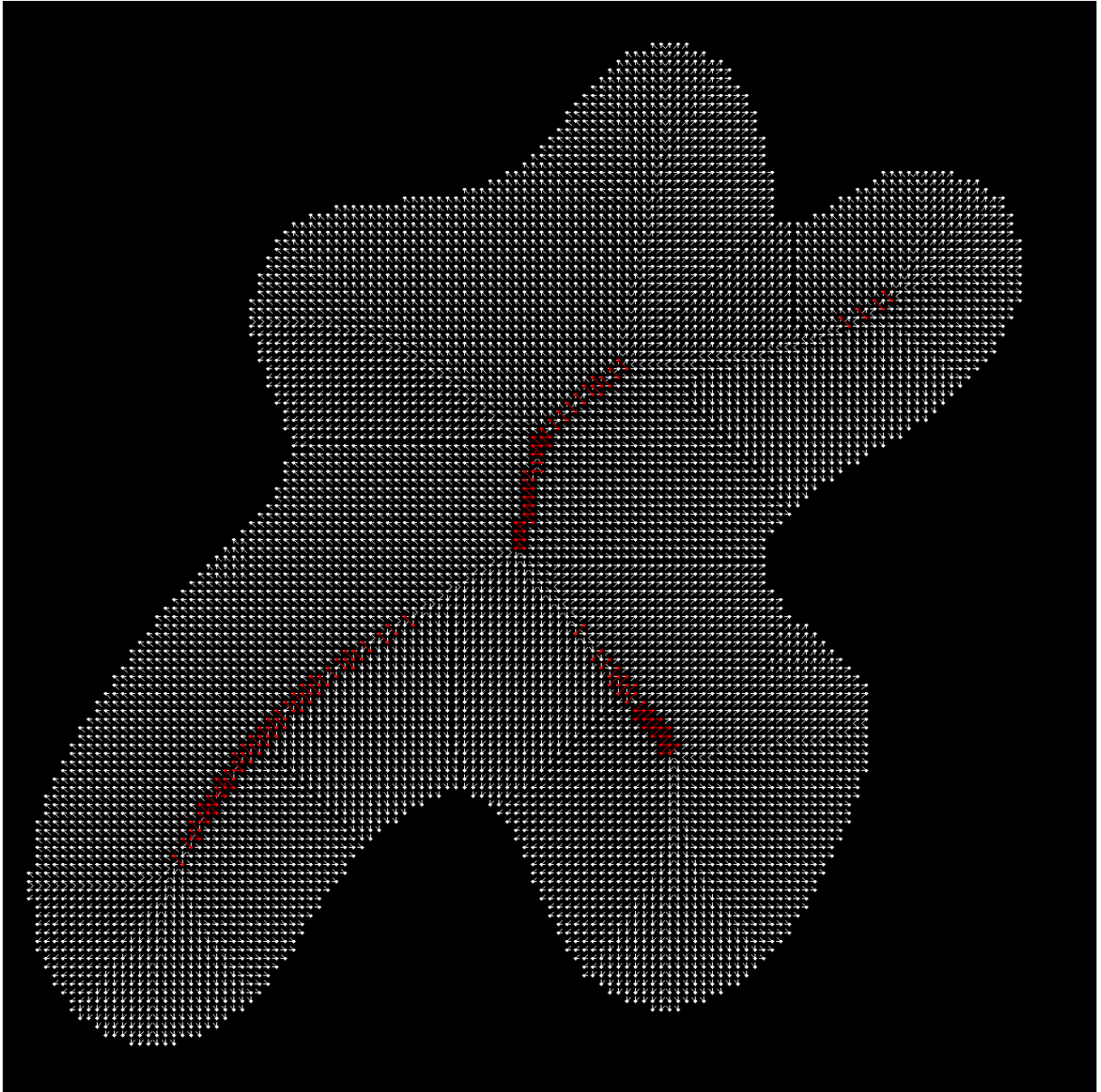
3.3.1 尋找中軸的候選人

因為在 graph G 中所有的向量都指向物體的中間，所以在物體中央的某些區域，會有區域內的向量都指向同一個點或是區域內的向量形成 loop，而 3.2 節中轉換影像為 graph 的演算法，會讓每個點都存在向外指的邊，所以在區域內的向量都指向同一個點的情形下也會形成 loop。存在 loop 的結構就表示 graph 裡面有 Strongly Connected Component (SCC) 的存在，而且 SCC 的位置應該是位於 graph 中間的區域。

利用 graph algorithm 來尋找出 SCC，可以得到圖 10 的結果，紅色的部分表示的是 SCC，從圖中可以看出 SCC 會出現在中軸附近的位置，因此把這些找出來的 SCC 作為是之後尋找中軸用的中軸候選人。



【圖 9】在 graph G 中，區域中向量都指向中間，中間的點的向量所指向的方向就會形成 Strongly Connected Component。

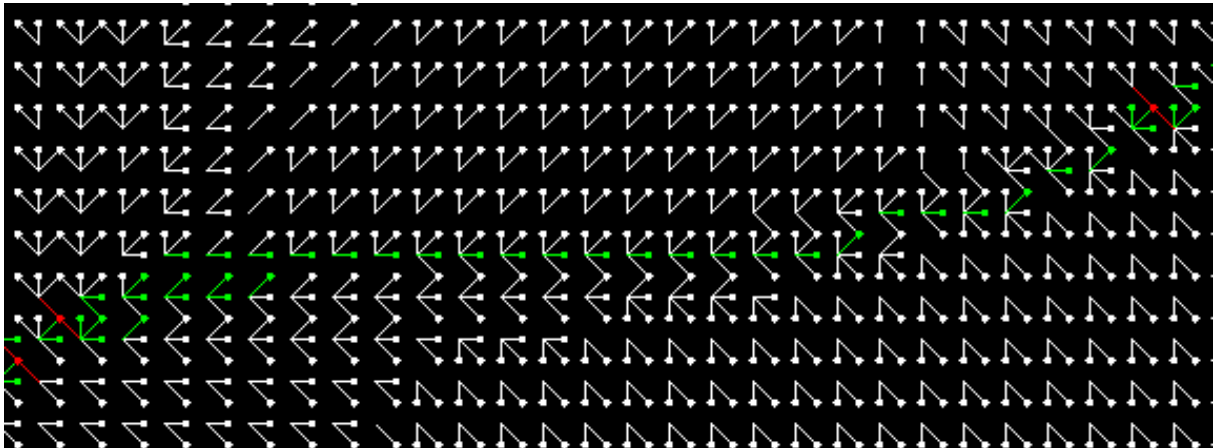


【圖 10】紅色的部分為尋找出來的 Strongly Connected Component。

3.3.2 將中軸候選人相互連結成中軸

尋找到 SCC 之後，利用 SCC 的點作為中軸的候選人，接著就要找出可以將這些候選人互相連結起來形成中軸的方法。由於 SCC 表示的是 graph 中的 loop，如果將連結 SCC 的邊都拿掉，就會從一個互相連結的 graph 變成多個不包含 loop 的的樹狀結構 (tree structure)。若一個樹狀結構中有數個 SCC 點的存在，可以假設在樹狀結構中，任兩個 SCC 點都存在一條路徑將他們互相連結起來。

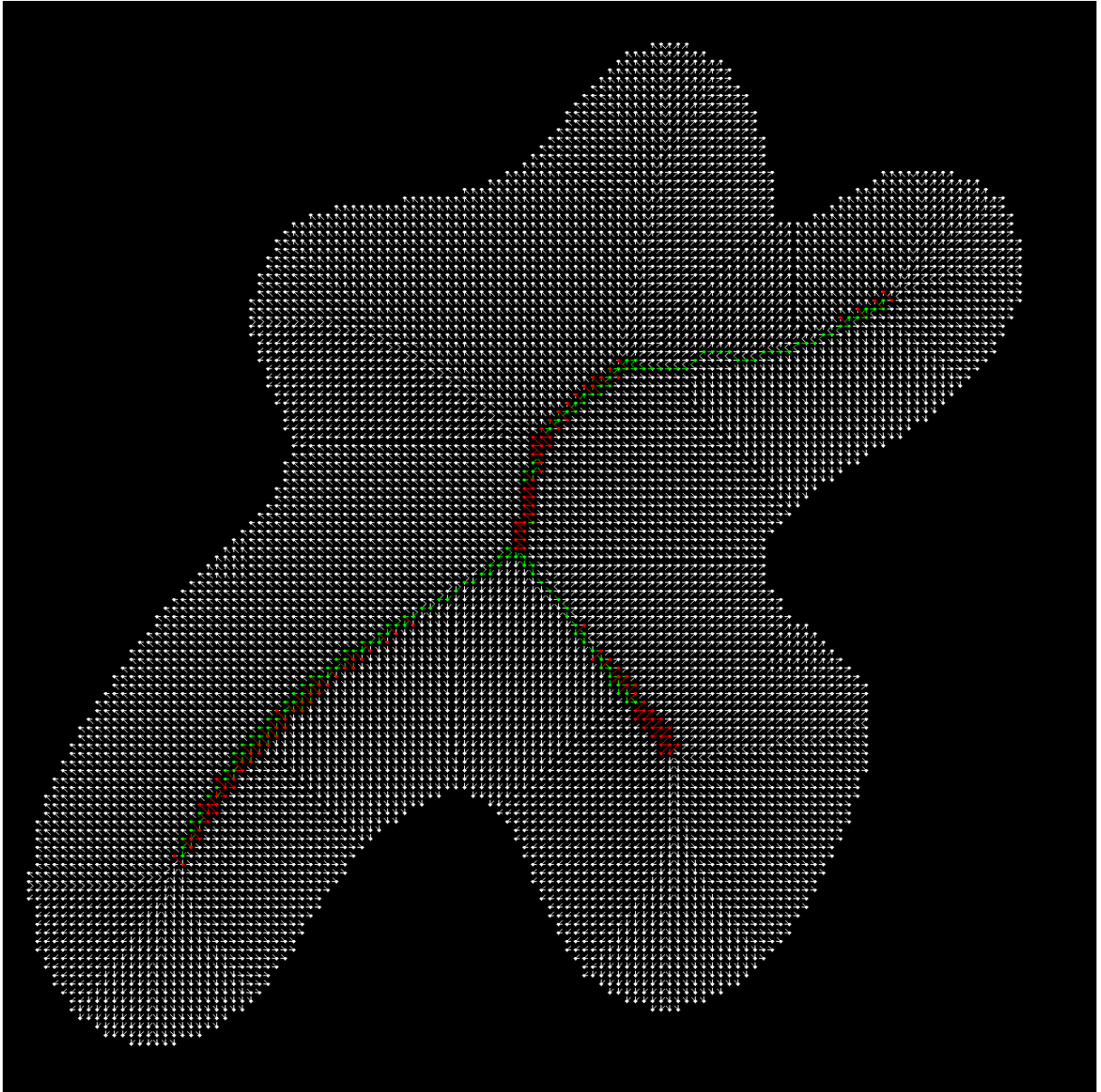
將對 graph G 進行 transpose 後的 graph 表示為 G^T ，則在 graph G^T 中，向量由指向物體的中央轉為指向物體的邊界，圖 11 可以看出向量都指向物體的邊界，所以連結兩個 SCC 間的最短路徑就是中軸。



【圖 11】紅色的部分為 SCC，綠色的部分表示兩個 SCC 間的最短路徑，箭頭表示的是每個點向外的向量方向。

在 graph G^T 中對每一個 SCC 做 Breadth-first search (BFS)，搜尋樹狀結構裡面所有的點，如果找到另外一個 SCC 後就將路徑 (path) 記錄起來，標示為中軸。因為是樹狀結構，所以找到的路徑是最短的路徑。圖 12 顯示由之前得到的所有 SCC 點進行 BFS 後尋找出來的路徑，綠色的部分為尋找出來的路徑，就是互相連結中軸候選人的中軸。

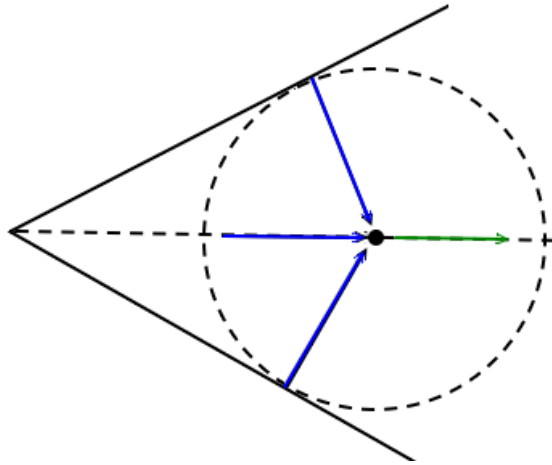
完成中軸候選人的互相連結後，下一個步驟是從由中軸的候選人向物體的邊界尋找連往邊界的中軸。



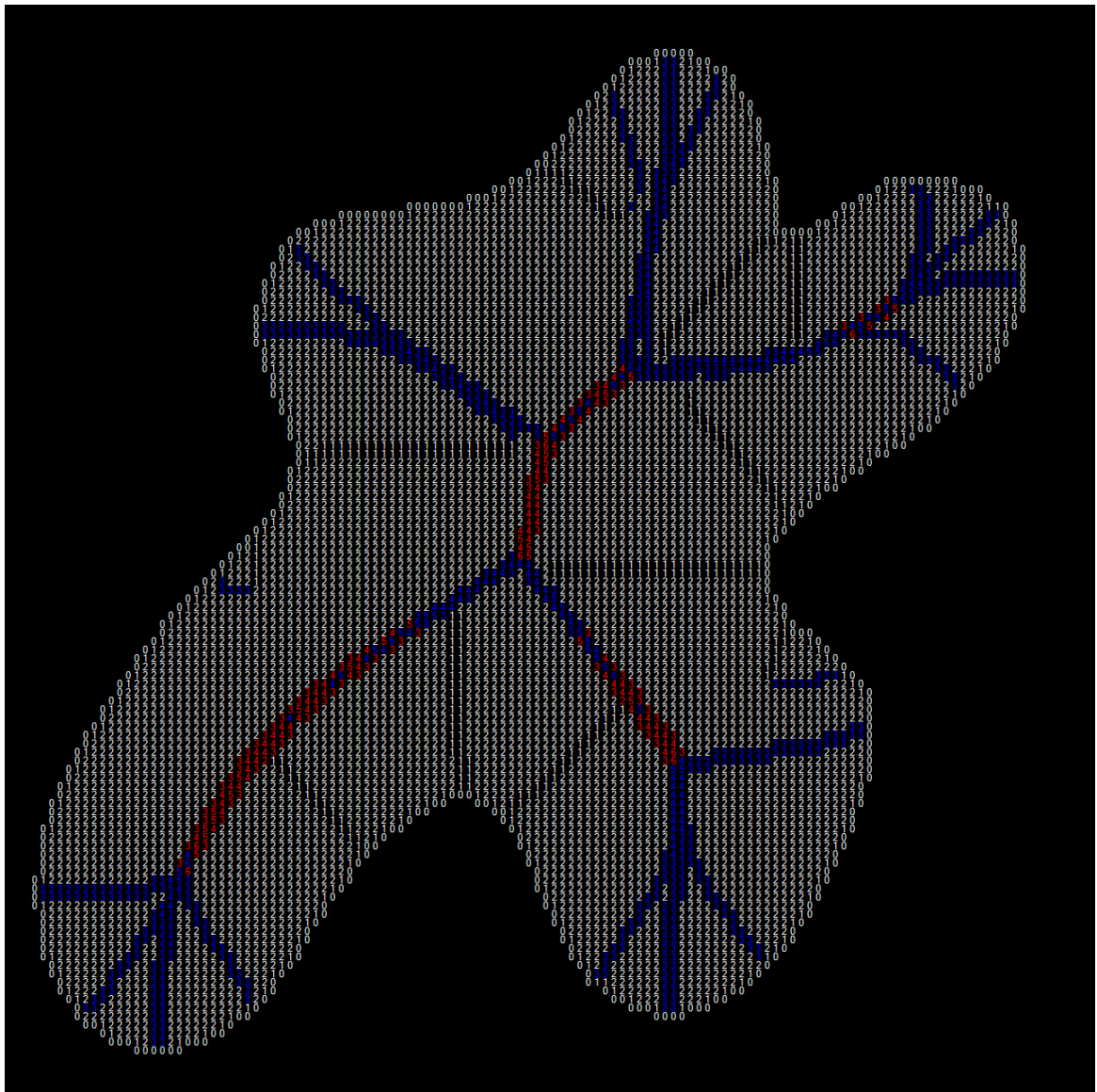
【圖 12】綠色的部分為將 SCC 互相連接起來的路徑。

3.3.3 由中軸的候選人向外發展中軸

仔細觀察中軸的特性可以發現到，在 graph G 中，中軸上的點應該至少要有三個指向該點的向量，即 InDegree 至少為三，三個向量分別為在中軸的兩側由邊界指向中軸的向量，以及同是中軸上的點的一個指向更靠近物體中間方向的向量。將所有 InDegree 大於三的点標出來之後，可以得到如同圖 14 的結果，紅色的點代表的是 SCC，藍色的點則是 InDegree 大於三的点，很明顯的跟所觀察到的中軸特性一致，藍色的部分很接近中軸的樣子，包括了連接 SCC 的部分及從 SCC 向外發展的部分。



【圖 13】在 graph G 裡，中軸兩側靠近邊界的點以及同在中軸上的點，都會存在向量指入中軸上的點。



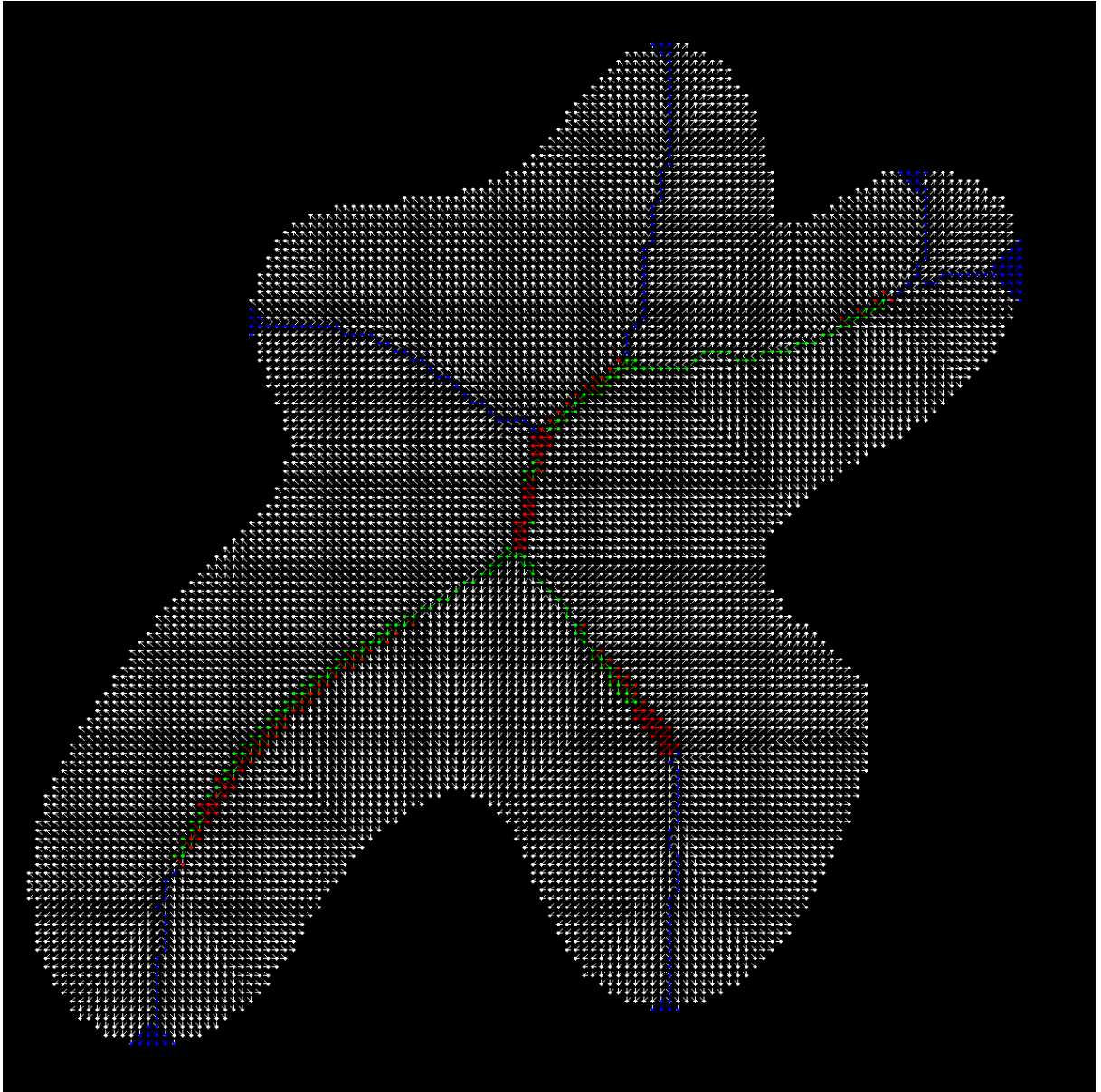
【圖 14】紅色的點為 SCC，藍色的點為 InDegree 大於三的点。

因為將 SCC 的邊去除掉後，graph 成為多個獨立的樹狀結構。在 graph G^T 中對每個 SCC 進行 BFS 來搜尋樹狀結構中的每一個點，在 BFS 樹的尾端會有數個稱為 Tree Leaf 的點，若這些點位於物體的邊界位置，便有可能是連往邊界的中軸。基於中軸的特性，這些點中距離 Tree Root 愈遠的點，愈有可能是中軸上的點。

因此將每個點的 InDegree 列入考慮之中，把連往物體邊界的路徑 InDegree 的平均值設計為參數，只選擇路徑的 InDegree 平均值大於該參數的路徑。選擇這些路徑中最長的路徑為中軸。

利用這兩個條件，在 graph G^T 中從每一個中軸候選人向外做 BFS 尋找向外連至邊界的中軸，可以得到圖 15 的結果，也就是所需要的中軸結構。





【圖 15】藍色的部分為向外尋找得到的中軸。

3.4 結論

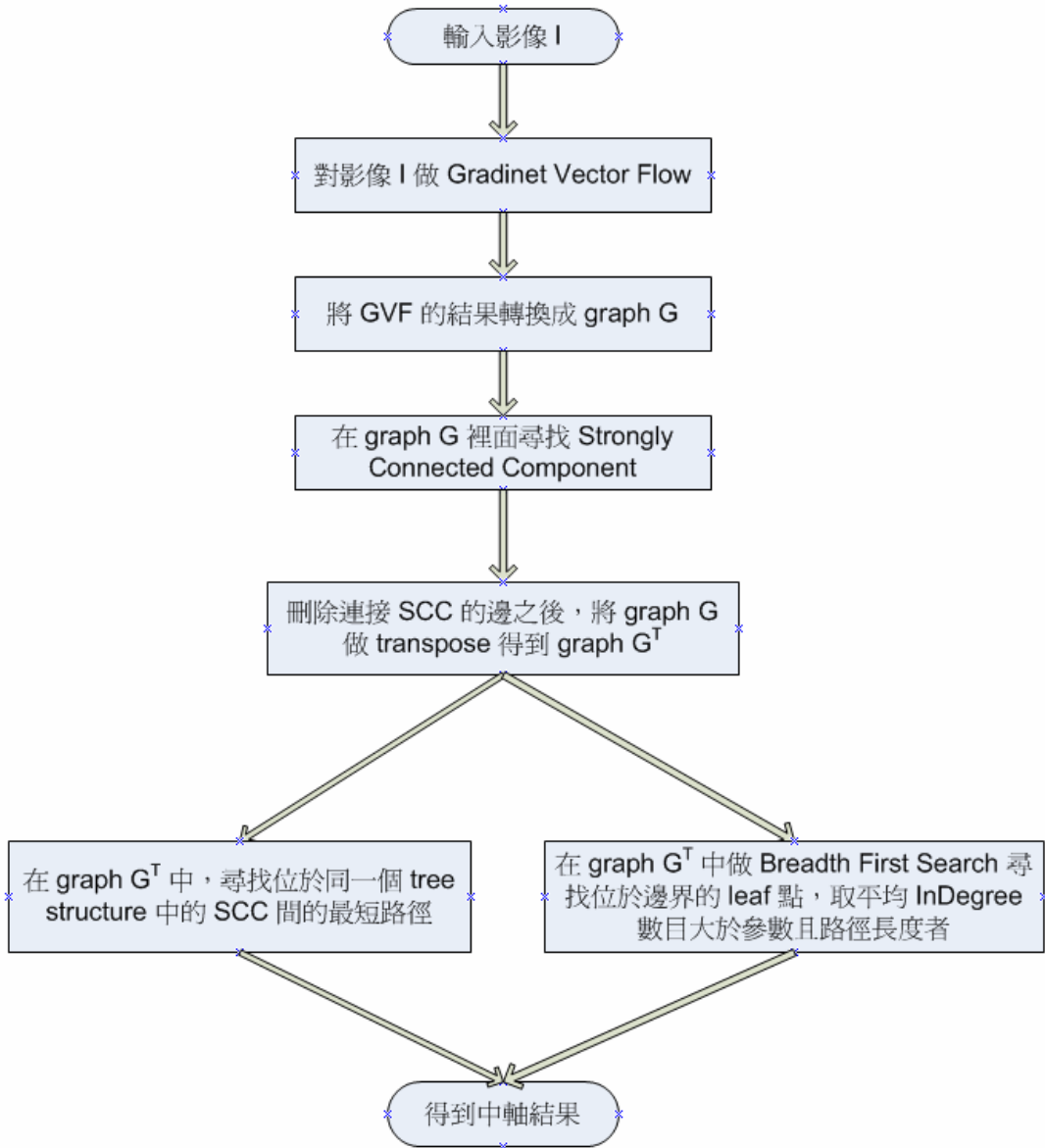
整理本章所提到的中軸演算法，可以歸納出將影像轉換為 graph 後，利用來尋找中軸的條件：

1. Strongly Connected Component 會位於中軸上。
2. 兩個 SCC 間若有中軸存在，必是兩點間的最短路徑。
3. 指入中軸上的點的向量有一定規則可循，由此可推論出向外發展的中軸的平

均指入向量。

4. 向邊界發展的中軸會是距離 SCC 最遠的路徑。

中軸的演算法可以用下面的流程圖來表示。



【圖 16】尋找中軸的演算法流程

第四章

實驗結果

4.1 前言

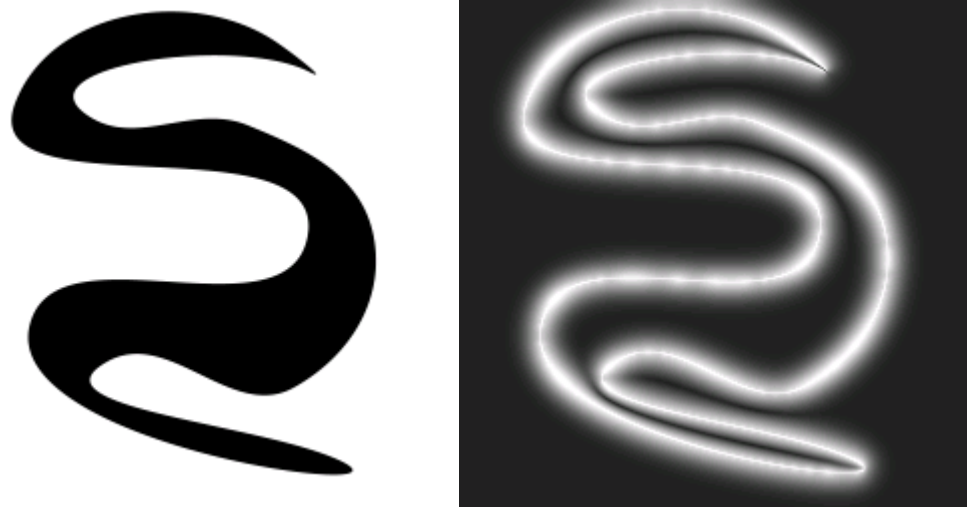
本章將以數種不同的影像來實驗第三章所提出的尋找中軸演算法，結果顯示在不同的物體上都有不錯的結果。演算法是以 Visual Studio .NET 來實做，使用的參數將針對不同的影像加以說明。

在二維影像的部分，使用了不規則的多邊型、管狀物體，以及較為複雜的楓葉形狀來實驗，至於三維影像的部分，則將用立方體和真實的大腸影像作為實驗資料。

4.2 二維影像實驗結果與分析

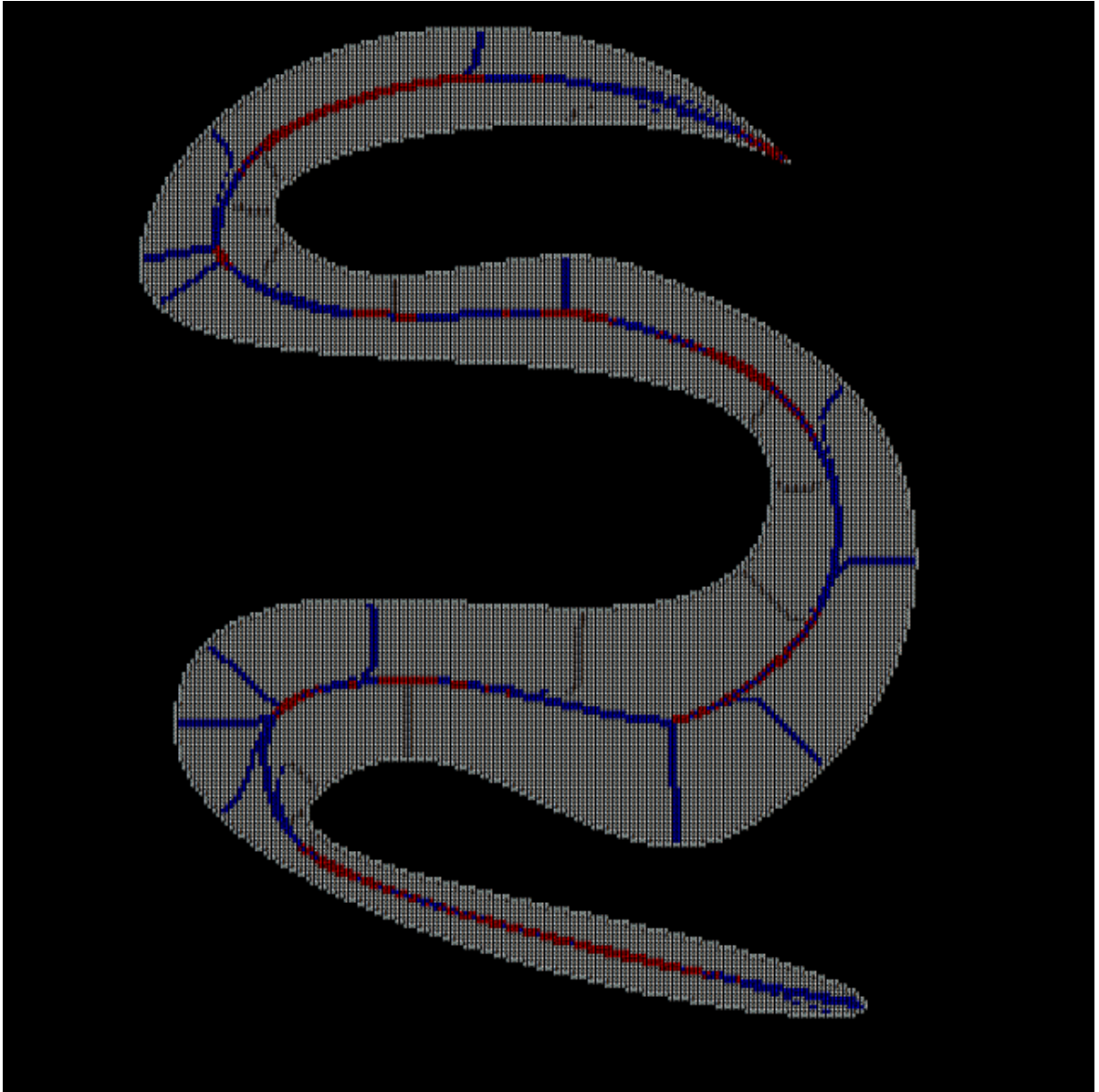
二維影像的結果有三種，但本章將只對其中兩種做說明，不規則的多邊型將只放入實驗結果。首先先對 S 型的管狀物體進行中軸的尋找，圖 17 的左圖為原本的物體，右圖為左圖做過 GVF 後以影像來表示 GVF 的結果。GVF 的影像中，亮的部分表示向量的值愈大，暗的部分則是向量值較小的部分。圖 18 為將 GVF 的影像轉換成 graph 後，尋找出來的 SCC 以及 InDegree 大於三的点，分別以紅色和藍色的點來表示。圖 19 則是最終的中軸結果，綠色的部分為將 SCC 互相連結的路徑，藍色的部分則為向外連至邊界的中軸。參數設定的部分，向外尋找中軸的平均 InDegree 需大於 3 的路徑才會被選取。

S 型物體尋找出來的中軸多為彼此間互相連結的結果，因其形狀較為細長，所以在中央的部分尋找出大量的 SCC，只有在最尾端的部分才存在向外尋找至邊界的中軸。

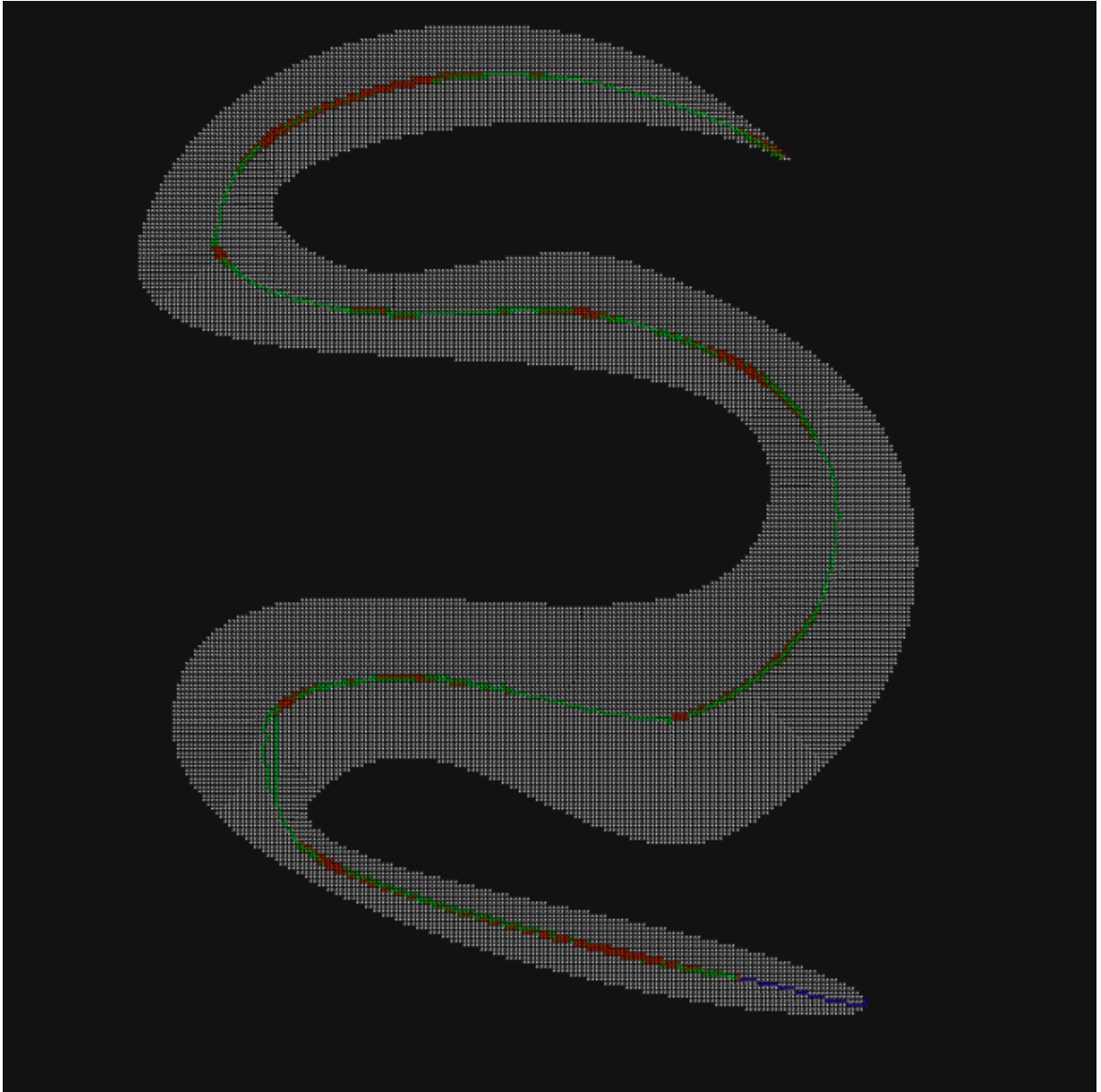


【圖 17】左圖為欲尋找中軸的物體，右圖為左圖經過 GVF 處理後的結果。





【圖 18】紅色部分為尋找出來的 Strongly Connected Component，藍色部分為 InDegree 大於三的点。



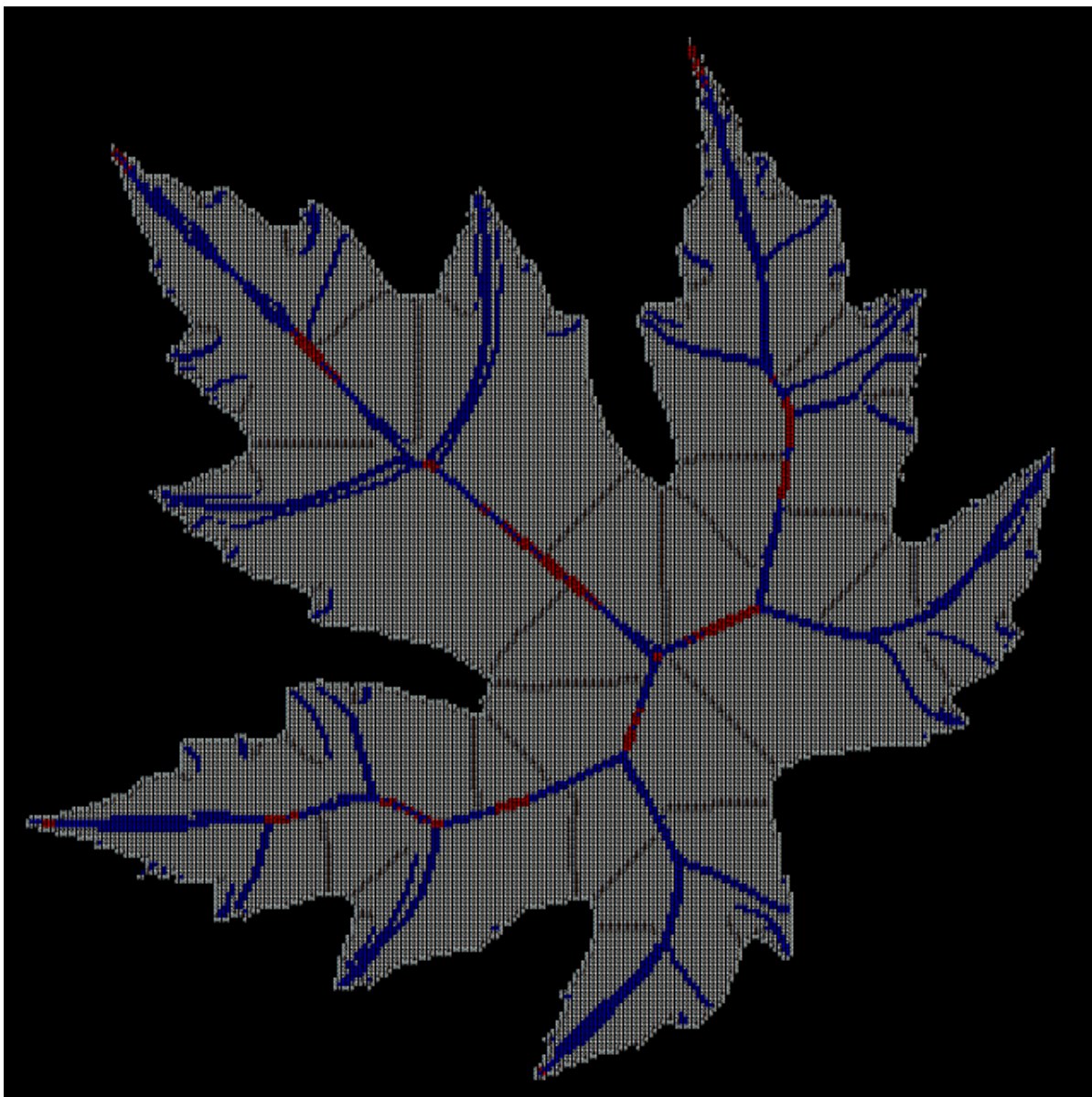
【圖 19】中軸的結果，綠色部分為將 SCC 彼此互相連接的中軸，藍色部分為由 SCC 向外發展的中軸。

接下來的楓葉影像是一張真實的楓葉影像，同樣經過 GVF 的處理以後將影像轉換成 graph，因為原本的物體形狀就較為複雜，可以看到圖 21 中 SCC 的位置和 InDegree 大於三的位置都較多而且複雜許多，同時也多了很多非中軸的點出現。但是在圖 22 的結果裡中軸的位置依然是正確的，不過在楓葉的左上方似乎有兩條小的分支並沒有被尋找出來，關於這個缺點，會在最後結論的地方做討論。

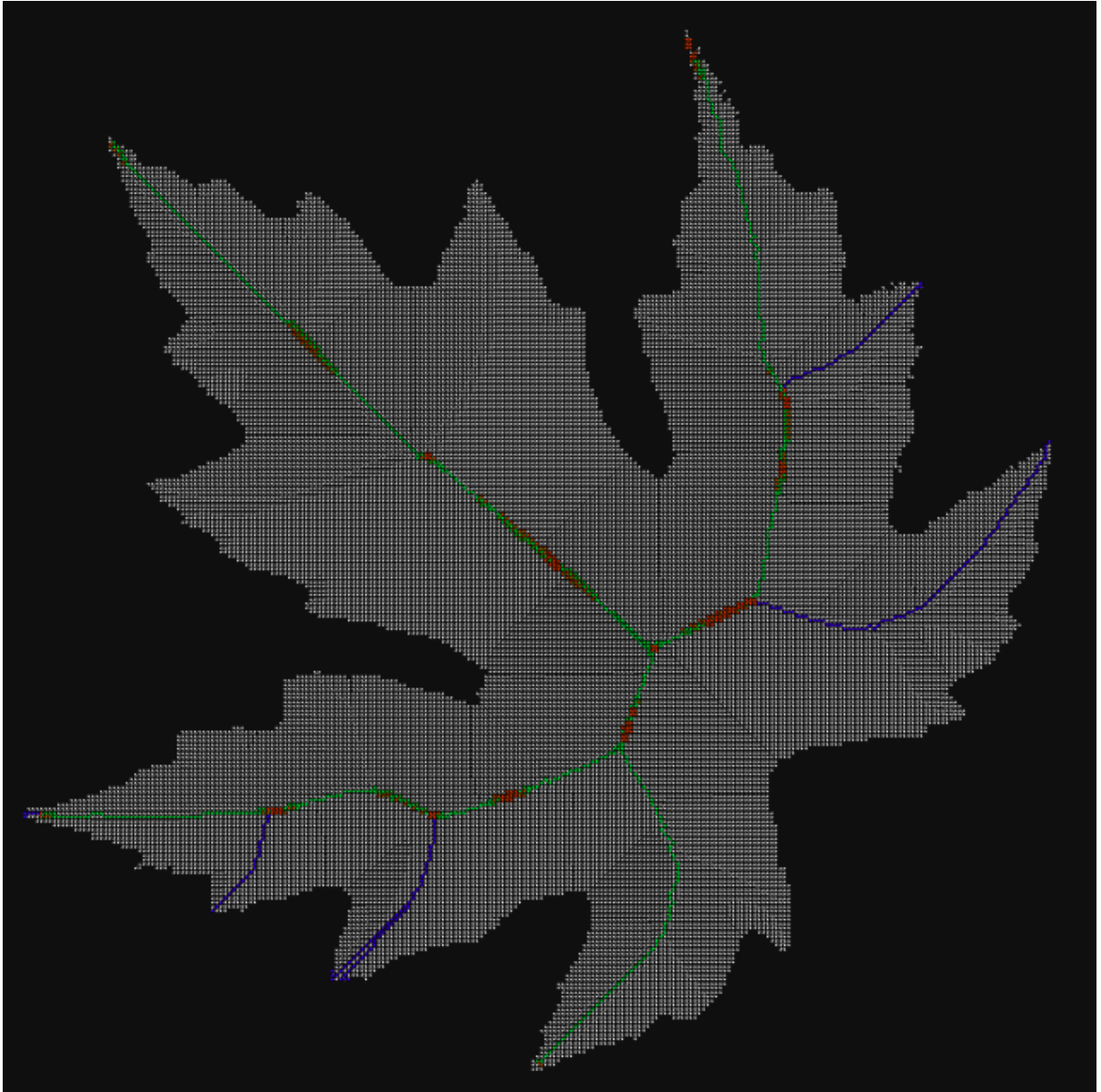
這張圖向外發展的中軸的平均 InDegree 參數設定為 2.99。



【圖 20】左圖為真實的黑白楓葉影像，右圖為左圖經過 GVF 處理後的結果。

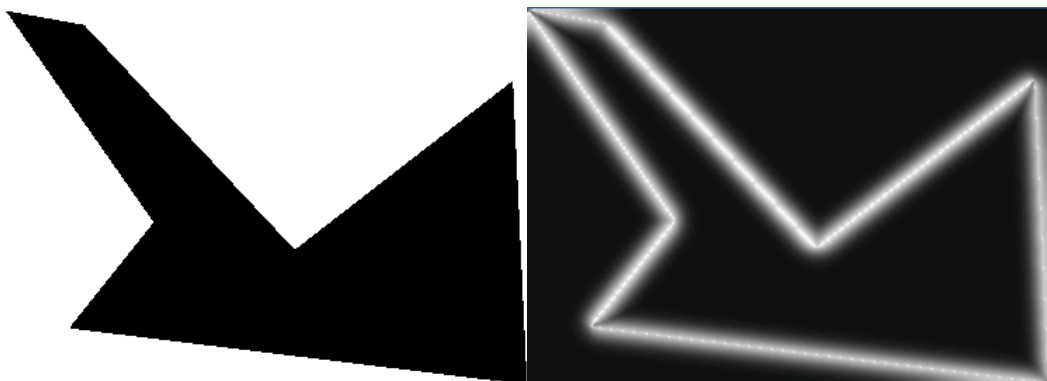


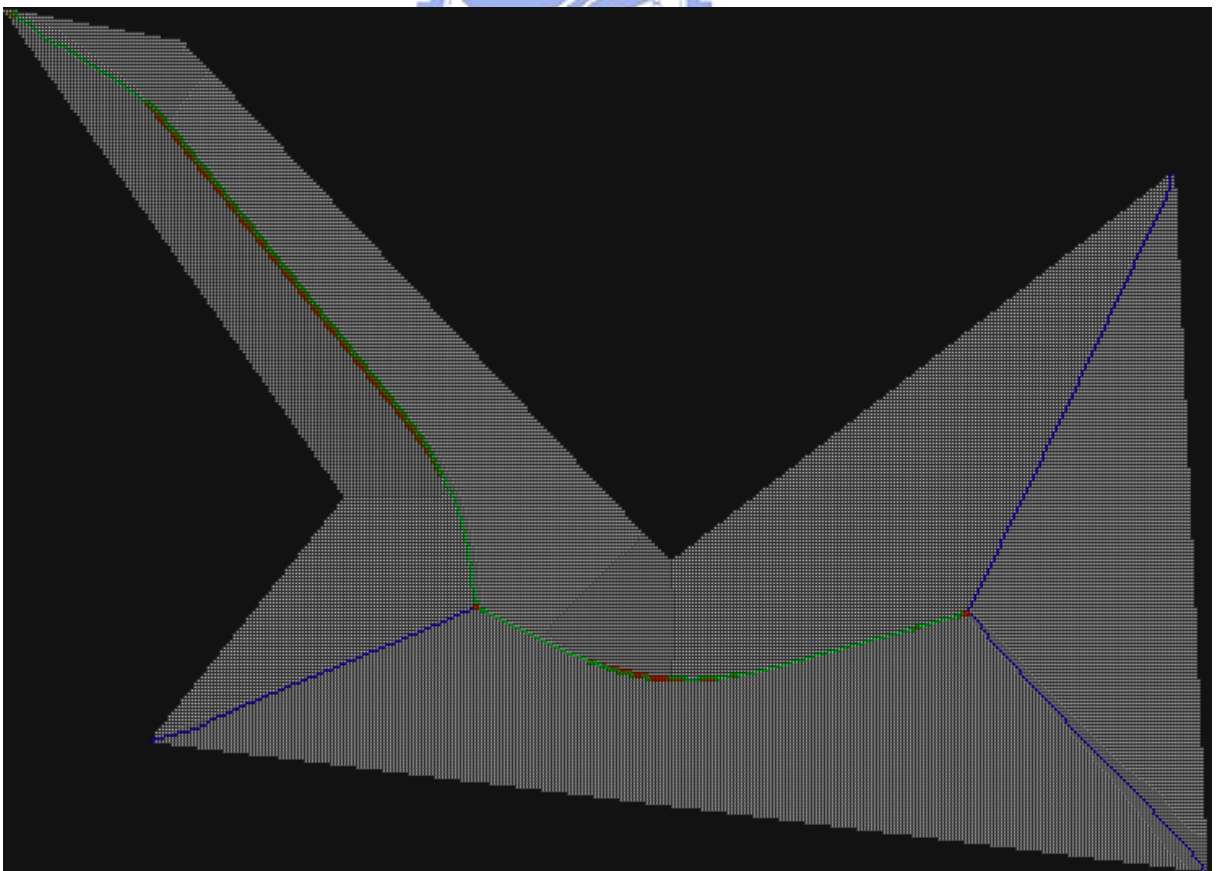
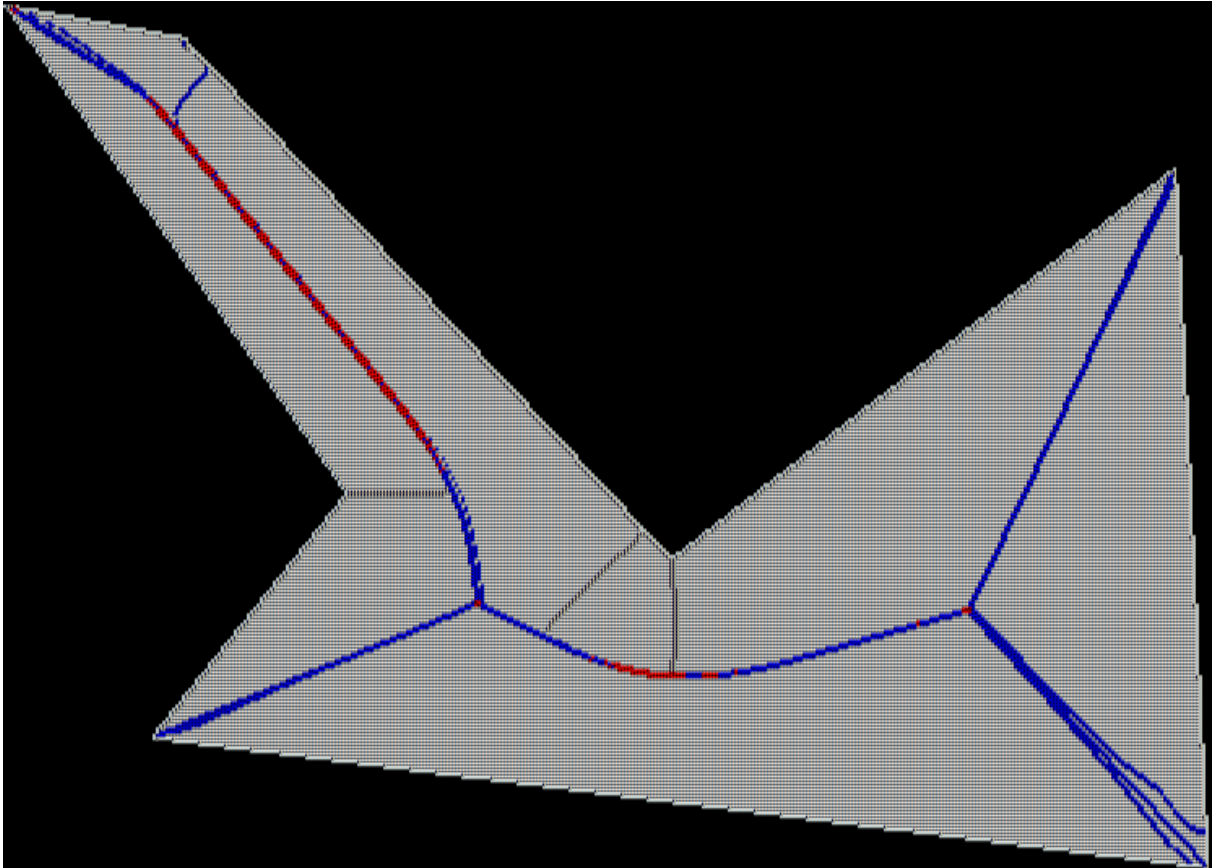
【圖 21】紅色部分為尋找出來的 Strongly Connected Component，藍色部分為 InDegree 大於三的点。



【圖 22】中軸的結果，綠色部分為將 SCC 彼此互相連接的中軸，藍色部分為由 SCC 向外發展的中軸。

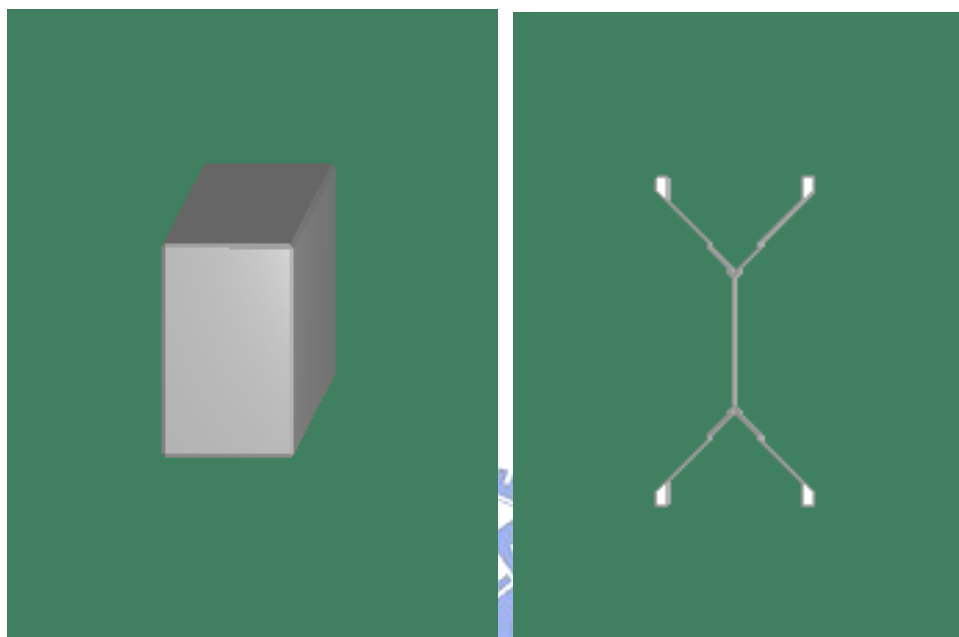
接下來則是不規則多邊型，因物體形狀較為簡單，便不另外多做說明。





4.3 三維影像實驗結果

由於三維影像的區域分割方法有所缺陷，所以得到的結果不如預期。本章只顯示一個三維影像尋找中軸的結果，用來尋找中軸的物體為一長方形的立方體，即圖 23 的左圖，圖 23 的右圖則是尋找出來的中軸結果。



【圖 23】左圖為用來尋找中軸的長方形立方體，右圖則是尋找出來的中軸結果

第五章

結論與未來展望

本篇論文提出了一個基於 Gradient Vector Flow，並轉換影像成 graph，主要針對 graph 來處理的中軸演算法。Gradient Vector Flow 的演算法可以幫助將影像上的資料轉換成有用的梯度向量資料，在利用梯度向量的資料將影像轉換成 graph 之後，便不需要特別針對二維影像和三維影像做不同的處理，可以使用一樣的方法來得到影像中的中軸，是這個演算法的方便之處。

但是在三維影像轉換成 graph 的過程中，分割三維空間為 32 個區域，導致有的區域被三個標準向量包圍，有的區域則是被四個標準向量包圍，而不是一致的向量數目，這在接下來的判斷上會造成困擾。如果可以設計更好的分割方法讓每個區域包圍的向量一致，在之後的判斷會更為準確。

其他的問題一是在向外連接的中軸尾端，因為具有相同距離的點有多個，導致最後會發生中軸散開的現象，並沒有找到好的方法來解決這個問題。另外在第四章的實驗結果中，可以看出沒有辦法控制要尋找多細緻的中軸結果，所以在楓葉的圖形裡似乎有中軸不夠完整的現象，如何加入這項控制的元素，也是之後可以繼續在努力的部分。

最後，由於演算法的某些地方需要利用參數來控制，所以有一個良好的使用者介面可以即時改變參數並顯示結果，會讓尋找中軸的過程更會有效率。

参考文献

1. D. Attali and A. Montanvert, "Computing and simplifying 2D and 3D continuous skeletons," *Computer Vision and Image Understanding*, vol. 67, pp. 261-273, 1997.
2. I. Bitter, A.E. Kaufman and M. Sato, "Penalized-Distance Volumetric Skeleton Algorithm," *IEEE Trans. Visualization and Computer Graphics*, vol. 7, pp. 195-206, 2001.
3. H. Blum, "A transformation for extracting new descriptors of shape," *Models for the Perception of Speech and Visual Form*, pp. 362-280, 1967
4. E.A. El-Kwae and M.R. Kabuka, "Binary object representation and recognition using the Hilbert morphological skeleton transform," *Pattern Recognition*, vol. 33, pp. 1621-1636, 2000.
5. Y. Fridman, S.M. Pizer, S. Alward, E. Bullitt, "Extracting branching tubular object geometry via cores," *Medical Image Analysis*, vol. 8, pp. 169-176, Sep. 2004.
6. H. Frimmel, J Nappi, H Yoshida, "Fast and robust method to compute colon centerline in CT colonography," *Proc. of the SPIE, Medical Imaging*, vol. 5031, pp. 381-387, 2003.
7. M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1987.
8. D.T. Lee, "Medial axis transformation of a planar shape," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, pp. 363-369, July. 1982.
9. C.M. Ma and M. Sonka, "A fully parallel 3D thinning algorithm and its applications," *Computer Vision and Image Understanding*, vol. 64, pp. 420-433, 1996.
10. C.W. Niblack, P.B. Gibbons and D.W. Capson, "Generating skeletons and centerlines from the distance transform," *CVGIP: Graph.Models Image Process.*, vol. 54, pp. 420-437, 1992.
11. Tun-Wen Pai and J.H.L. Hansen, "Boundary-constrained morphological skeleton minimization and skeleton reconstruction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 201-208, 1994.

12. Chenyang Xu and J.L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, pp. 359-369, 1998.
13. Y. Zhou, A.E. Kaufman and A.W. Toga, "Three-dimensional skeleton and centerline generation based on an approximate minimum distance field." *The Visual Computer*, vol. 14, pp. 303-314, 1998.
14. Y.C. Hsueh, *Mathematical Morphology Lecture Notes*, chap 5, pp. 109, 2004

