

## 解決三維細化曲線骨架擷取的多餘骨架節點問題

陳宏光

國立勤益科技大學

電子工程系

hankchentw@gmail.com

涂致遠

國立勤益科技大學

電子工程系

meditation0624@yahoo.com

### 摘要

三維曲線細化演算法(3D curve-thinning algorithm)是曲線骨架(curve skeleton)擷取方法中，簡單且計算效率最為快速的方法。但此類方法，大多無法保證所擷取之骨架為零容積(zero-volume)或單位寬度(unit-width)，且均未能有效解決骨架冗枝(redundant link)的問題。於本篇論文中，我們將提出一個新的方法。實驗結果顯示，此方法可有效解決多餘骨架節點(redundant region)的問題，並保證所擷取骨架能具有單位寬度、零容積、以及置中等特性，並適用於處理數種不同類型的骨架。

**關鍵詞：**三維曲線細化、最短路徑、曲線骨架。

### 1. 前言

現今 3D 模型已被廣泛運用在諸如電腦動畫、電腦輔助設計 CAD(computer-aided design)、醫學科學、電腦圖學、科學上的可視化、三維模型表面重建、形狀比對等許多不同的應用。其中，許多以節省記憶體成本為考量的應用，皆需針對某些原始模型的特徵作進一步處理，以從中取得較原模型更「精簡」的特徵化模型作為其後續處理的參考。曲線骨架(curved skeleton)即為此類特徵化模型中，運用的最廣泛的一種。

簡單來說，曲線骨架是一種以一組趨近於零容積的曲線表示模型特徵的方法。外觀上，它是由骨架節點相互連結所形成的多個線狀區段所構成。這些線狀區段所形成的連線是數組趨於平滑的曲線，故稱之為曲線骨架。醫學上，常用於分析二維影像之分析：如內視鏡檢查時，可利用曲線骨架分析病理結構以協助醫師診斷。此外，如角色動畫與模型比對等皆為以曲線骨架為基礎常見之應用：其中，曲線骨架可提供角色動畫所需的動作骨架以及提供模型比對時所需要的模型拓撲特徵。

傳統的曲線骨架擷取法，是找出模型的中軸轉換法(medial axis)做為其骨架。以二維模型的骨架擷取為例，其骨架擷取的方式，是在模型內部找出所有貼近模型邊緣的最大圓型平面，將這些所有的圓型平面的圓心以紅點標示出（如圖 1(a)所示），模型上所有的紅點，即為此模型相對應的骨架。

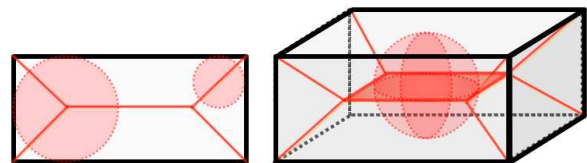


圖 1. (a) 二維中軸轉換法骨架擷取方法，以及(b) 三維中軸表面轉換法骨架擷取方法。

相對於二維模型的骨架擷取方式，三維模型骨架萃取方法則稱之為中軸表面轉換法(medial surface)，它是以模型內部的各個結點為球心，找出模型內部所能構成最大球體的所有節點。同樣的，將這些球體中心以紅點繪製出（如圖 1(b)所示），這些所有的紅點即為此模型的曲線骨架。但不論是中軸轉換法或是中軸表面轉換法骨架擷取方法，此類型的骨架擷取方法皆對模型表面雜訊極其敏感，所擷取出的骨架不盡符合需求。

對此，目前已有許多學者提出其他不同類型的曲線骨架擷取方法。Nicu D. Cornea [5]將這些曲線骨架擷取方法分析研究，並將現有的擷取方法給予歸類，將曲線骨架擷取演算法，概括的分為四大類型，分別為細化與邊界增值方法(Thinning and Boundary Propagation)、使用距離場方法(Using a Distance Field)、幾何方法(Geometric Methods)和一般場函數方法(General-Field Functions)等四類。

首先，細化與邊界增值方法主要是對模型邊緣的表面重覆侵蝕直至骨架萃取。使用距離場方法為利用模型內任意結點，以任意結點到模型表面的最短距離來做為骨架擷取的依據，擁有距離較長的模型內部結點即為其骨架。幾何方法採用模型本身基本的幾何關係來建構出骨架。例如，根據給定的等位函數(level-set function)建構模型的 Reeb graph 的方法。而一般場函數方法則是以勢場(potential field)函數的方式來擷取骨架。

本篇論文是以細化與邊界增值方法的類型來做為主要的曲線骨架擷取演算法。此類型的演算法，是重複的對模型作侵蝕邊緣的動作，直至骨架有效的萃取完成。侵蝕的條件則取決於各演算法對取樣節點的判斷。此類型的演算法是曲線骨架擷取演算法中計算效率極高的方法。不幸的是，萃取的骨架通常較為不平滑，並有可能產生過多的額外錯誤骨架分支問題，甚至有產生非單位寬度骨架的情況。

本篇將對非單位寬度的骨架節點，做單位寬

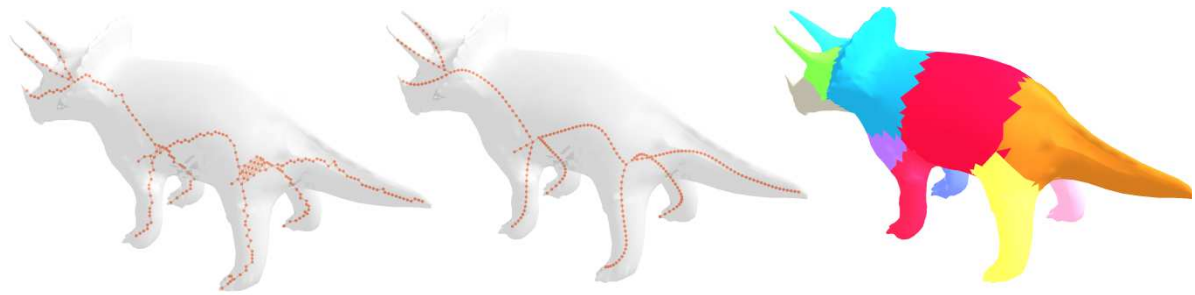


圖 2. 解決三維曲線細化演算法的骨架擁擠問題。(左圖) 骨架形成有多餘節點區域。(中間) 剔除多於節點後的單位寬度骨架。(右圖) 骨架所對應的模型表面網格分割。

度的曲線骨架擷取，以及減少額外錯誤骨架分支發生情況。並對擷取的骨架節點間，形成鋸齒的骨架做一平滑化處理 (如圖 2 所示)。

在本篇論文中，第二節說明本篇所使用三維細化演算法，以及利用立體像素化(voxelization)轉化，來取得三維細化演算法所需使用的三維二元物件資料型態。第三節為本篇所提出的曲線骨架擷取演算法整體流程。第四節為本篇演算法的演算流程，演算法中的各種定義以及步驟都會在此章節說明。另外，對於各種三維模型的曲線骨架實驗結果將展示於第四節，包括非單位寬度曲線骨架的曲線骨架比較結果。

除此之外，本實驗還利用所產生的單位寬度曲線骨架來對模型表面做的網格分割(mesh segmentation)骨架應用 (如圖 2(右)所示)。最後，第五章為本篇的結論以及未來研究方向。

## 2. 相關研究

三維模型的曲線骨架擷取方法中以三維曲線細化演算法最為快速。相較於其他三種不同類型的骨架擷取演算法，其本身的特點在於此方法具有計算效率高的特性。不幸的是，此方法是基於模型表面所做的骨架擷取方法對模型表面的雜訊非常敏感，容易萃取出不平滑骨架以及產生額外錯誤骨架分支的問題。目前，這些問題皆還存在於各種的三維細化演算法中。

三維細化演算法常用於醫學影像的 MRI、CT 影像分析，為了能夠幫助醫師診斷更細微的病情分析，將這些一張張的 MRI 或 CT 掃描影像資訊，以數據傳送至電腦來還原三維的病理結構組織。而這些掃描資訊在電腦是以二元物件的架構來表示，利用這些二元物件建構出三維模型，形成所需的三維二元物件，接著即可對此模型擷取出所需診療病理資訊來做病情診斷，而這種擷取方法，即為此類型演算法的研究目的。

一般來說，醫學影像的病理掃描資料不易取得。相較於此，三維網格模型有較易取得的特性，模型的製作也較於便利。基於三維細化演算法是對模型表面重覆侵蝕的方法來擷取骨架，故三維細化演算法所需要的三維二元物件必須是包含模型內部的實心物件。但是，三維網格模型為節省記憶體以及降低圖像繪製計算消耗，通常只存有模型表面的頂點座標和模型表面的連接關係。這些資訊所形成的網格模型，並非是一個實心的三維物件，無法對此網格模型以三維細化演算法的方式擷取骨架。

本篇論文是將網格模型以實心立體像素化(solid voxelization)[6]的方式，將三維網格模型填充 voxel 至模型表面以及模型內部，來達到三維二元物件的取得。網格模型所填充的立體像素數量和精細程度將影響三維細化演算法萃取曲線骨架的平滑程度，以及所擷取的骨架節點的數量。目前，三維曲線細化演算法[2][3][4]已有許多學者提出許多不同的取樣節點判斷演算法。但是這些演算法，皆無法完全保證所萃取出的骨架必定是零容積的骨架。不同的取樣節點判斷所萃取的曲線骨架結果不盡相同，可能萃取出非零容積的曲線骨架。

針對這個非零容積問題有相關研究提出了對於三維曲線細化演算法的單位寬度曲線骨架解決方法[7][8]。本篇也是為了解決三維細化演算法中，可能產生非單位寬度的骨架問題，並且減少可能形成多餘錯誤骨架分支的產生情況。

首先，找出骨架中形成非單位寬度的所有骨架節點，接著對這個錯誤的骨架區塊，利用最短路徑演算法[1]，找出此錯誤骨架區塊到最近的單位寬度骨架節點路徑。此方法能有效減少許多非單位寬度的骨架問題，所建立的最短路徑骨架結構，也能達到單位寬度的曲線骨架。除此之外，由於大部分三維細化演算法，皆是基於三維二元物件的架構來做為骨架擷取的前提。當考慮以每個立體像素連結的所有二十六種的關係來建立骨架關係時，可能形成的循環問題，本演算法也能有效消除。

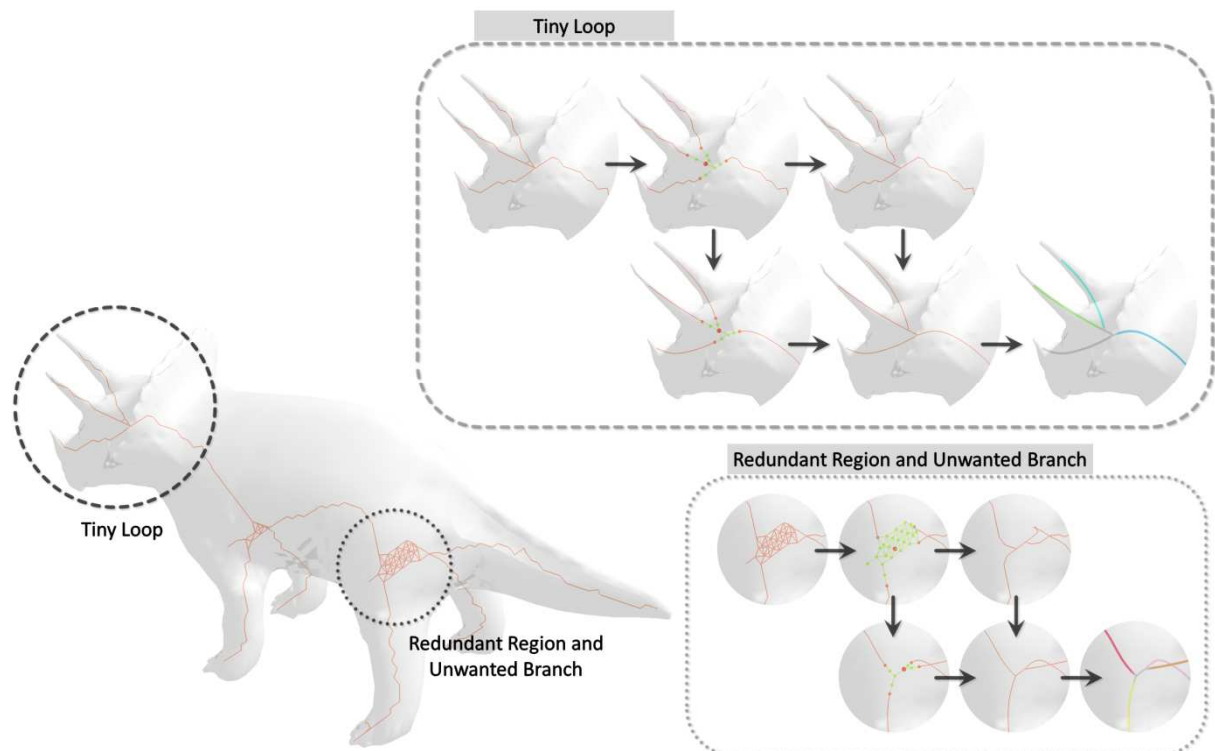


圖 3. 主要演算法流程。(a)建構各骨架節點間的連線並計算分支度(Degree)。(b)找出骨架中的多餘節點區域(Redundant Region)。(c)利用最短路徑(Shortest Path)演算法剔除不必要的骨架節點。

另外，三維細化演算法相較於其它的曲線骨架擷取演算法來說，有較容易產生多餘錯誤骨架分支的問題。在本篇方法中，在尋找非單位寬度的骨架區塊時，會將骨架分支長度過於短小的節點，考慮視為非單位寬度的骨架節點，以減少產生多餘錯誤骨架分支的情況發生。此方法可用於避免骨架上微小的凸出產生，並且能增加整體骨架的平滑性。

最後，對於三維細化演算法不平滑骨架的問題。當三維網格模型轉換為三維二元物件時，其立體像素化的精細程度會嚴重的影響到骨架擷取的品質。同時這也是三維細化演算法中，形成不平滑骨架的主要原因。本篇演算法將對最後所產生的單位寬度骨架節點，以一個平滑演算法，來平滑骨架節點間的鋸齒情況，改善擷取的骨架為平滑的曲線骨架。

### 3.單位寬度的曲線骨架演算法

本篇論文是對三維細化演算法後，可能產生多餘骨架節點(redundant-point)所提出的解決方法。首先，對於三維細化演算法後的所有骨架節點，要計算出各個骨架節點的分支度(degree)。計算各個骨架結點分支度的方法，是將三維曲線細化演算法所擷取出的離散骨架節點，以每個立體像素連結的所有二十六種關係來建構骨架連線，骨架連線數目即為此骨架節點分支度的程度。利用各個骨架節點的分支度，對所有的骨架節點，找出終端節點(end-point)、一般節點(middle-point)和關節點

(joint-point)等三種主要的骨架節點類型。最後剩餘的骨架節點皆視為多餘骨架節點。

接著於第二步驟找出多餘節點區域，將相互連接的多餘骨架節點所形成的區塊視為一個“多餘節點區域”。接著對各多餘節點區域的多餘骨架節點集合中，找尋為多餘骨架關節點(redundant joint-point)的骨架節點。當多餘骨架關節點搜尋完後，對可能形成循環的多餘骨架關節點，去做進一步的問題排除。將可能形成循環的多餘骨架關節點，與其相連骨架節點皆視為多餘骨架節點。

最後將所有形成多餘節點區域的骨架節點，利用最短路徑演算法的路徑搜尋，對每個多餘節點區域找尋出最佳的單位寬度曲線骨架。此最短路徑搜尋，是從每個多餘節點區域中，以多餘節點區域最接近物體中心的質心節點，做為最短路徑搜尋的起點。接著找尋此多餘節點區域中，從起點到所有多餘骨架關節點所能經過的最短路徑。

最後將非最短路徑經過的所有骨架節點剔除後，所剩餘的骨架節點即為此多餘節點區域中的單位寬度曲線骨架(如圖 3 所示)。詳細的演算法將於下一章節做說明。

#### 3.1 骨架節點分支度

三維曲線細化演算法所萃取出來的骨架，其所有的骨架節點皆為離散的三維頂點資訊。為了建構出骨架節點彼此之間的關聯性，本演算法將各個骨架節點與相鄰的骨架節點，以每個立體像素連結的



所有二十六種的關係，來做為骨架節點的連接性架構(如圖 4 所示)。

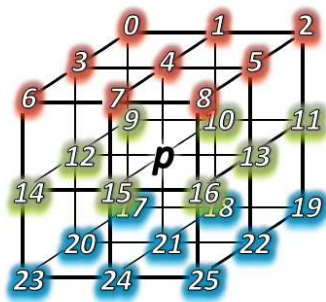


圖 4. 骨架節點的關連性是搜尋各骨架節點 P 的二十六種方向來建立骨架連線。

對所有的骨架節點做相鄰點的搜尋，當搜尋到此骨架結點的二十六種相鄰方向上擁有骨架節點時，將此搜尋到的相鄰骨架節點與其本身作為一個骨架連結關係。並持續對其餘的骨架節點作骨架連結關係的建立。當各骨架節點以二十六種相鄰方向，建構完所有骨架節點上的骨架連結關係後，即可利用各個骨架節點擁有的骨架連結關係數目，來做為各骨架節點的分支度。

參考[7]的方法，我們也按骨架節點的分支

度，做為一個初步的多餘節點區域建立(如圖 5 所示)。將分支度程度的不同定義為不同的骨架節點類型，這些骨架節點一共分為四種不同的主要類型，分別為終端節點、一般節點、關節點和多餘骨架節點。當分支度為一時，將骨架節點視為一個終端節點。

正常來說，終端節點多為此模型的特徵節點。以生物模型來說，分支度為一的終端節點代表著模型的肢體末端，如動物的四肢末端。而形成骨架的主要骨架節點元素“骨頭(bone)”，多有與其他骨頭頭尾相連的特性。故將骨架節點的分支度等於二時，將其視為一般節點，這些一般節點為構成骨頭的主要部分。

其餘的骨架節點在正常的狀況下，應當是此模型的關節部位。但是因為使用不同的三維曲線細化演算法，使用不同的取樣節點重覆侵蝕條件。最後所能擷取出的骨架節點，並不能確保完完全全皆為單位寬度的曲線骨架情況。有可能因為取樣節點侵蝕條件判斷不夠周詳，導致所擷取的骨架節點會有多餘節點區域的骨架節點集合。

本演算法在最糟情況下，將所有分支度大於二的節點，皆視為可能形成錯誤骨架的多餘節點區域骨架節點。這些分支度大於二的節點，將其視為多餘骨架節點。但是，假使多餘骨架節點集合中骨

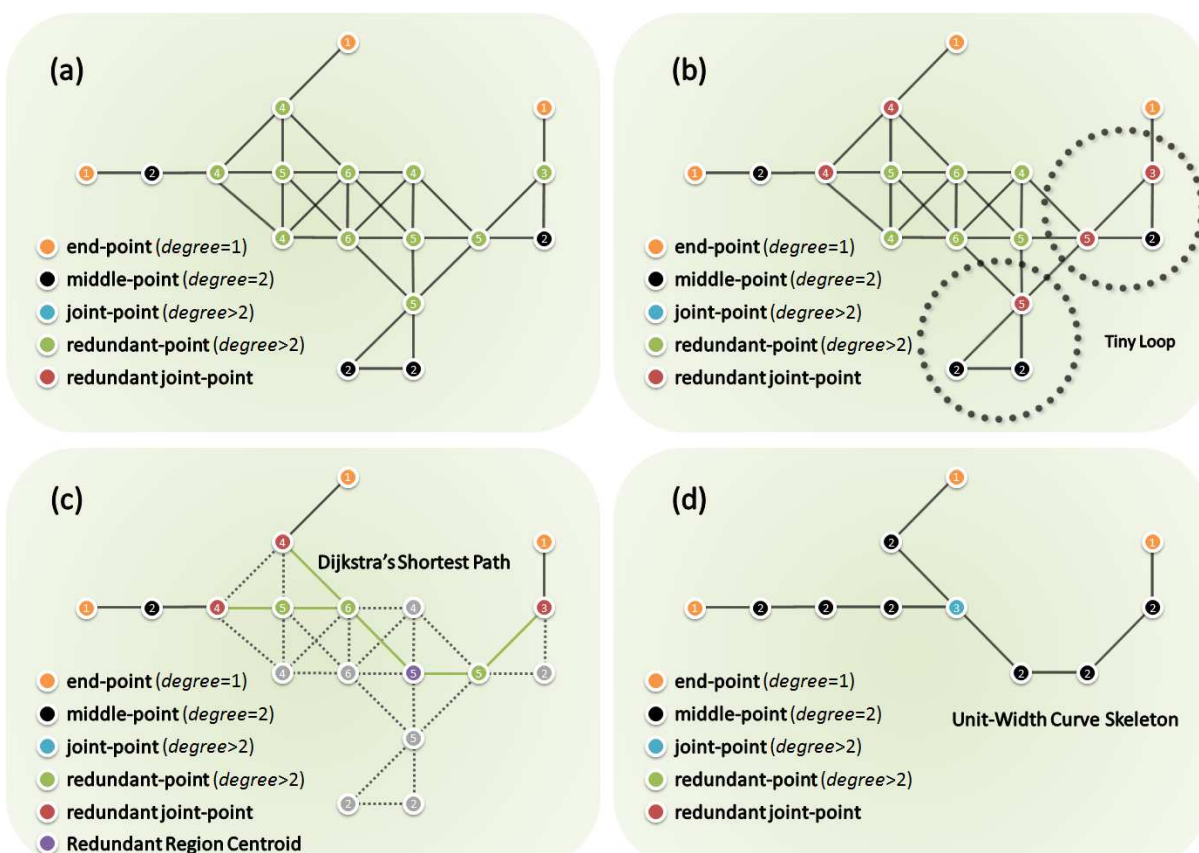


圖 5. (a)依骨架節點分支度，將結點分為：終端節點(degree=1)、一般節點(degree=2)、多餘骨架關節點(degree>2)。(b)、(c)可能形成的循環及解決方案。(d)經由最短路徑搜尋的結果。

架節點的二十六種連結關係  $Neighbor_{26}(v_i)$  的所有骨架節點皆為終端節點或一般節點時，則此骨架節點將還原成正常的關節點骨架節點。在之後演算法步驟中，將會剔除不必要的多餘骨架節點，來達到單位寬度的曲線骨架擷取。

### 3.2 多餘節點區域

多餘節點區域為利用三維曲線細化演算法骨架擷取時，所可能產生的非零容積骨架節點集合。在上一步驟中，本演算法以分支度做為骨架節點的分類方法。在本步驟中，將多餘骨架節點所形成的骨架節點集合，視為多餘節點區域。接著對每個多餘節點區域搜尋所有的多餘骨架關節點，找出多餘節點區域中與終端節點或一般節點相互連結的骨架節點。當此多餘節點區域骨架節點的  $Neighbor_{26}(v_i)$  關係中，擁有最少一個相互連結的骨架節點為終端節點或一般節點，則此骨架節點即為此多餘節點區域中的一個多餘骨架關節點 (如圖 5(b)所示)，將多餘骨架關節點以紅色標示。

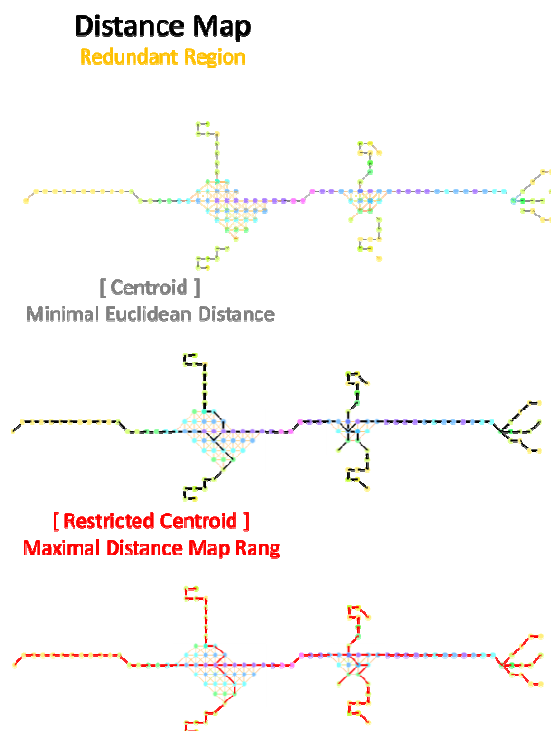


圖 6. (上圖)以距離圖(Distance Map)著色的多餘節點區域，不同顏色代表距離模型表面的距離長短，黃色距離最近，紫色則距離模型表面最遠。(中圖)以最短歐氏距離求得質心的單位寬度曲線骨架。(下圖)將質心限制於多餘節點區域中，最大的距離區間內的單位寬度曲線骨架。

由於本篇演算法是以每個立體像素連結的所有二十六種的關係來做為骨架節點的連接，可能有形成循環的問題。為避免可能產生此問題，本篇演算法找出可能構成循環的骨架節點及其

$Neighbor_{26}(v_i)$  皆改設為多餘骨架節點。此步驟能有效避免本演算法中循環的問題產生，能於最後的最短路徑搜尋去除錯誤的骨架循環 (如圖 5(c)所示)。

另外，由於形成多餘節點區域的骨架節點，通常是因各三維曲線細化演算法的取樣節點判斷方法的優劣所造成，而這些多餘節點區域的骨架節點可能會形成多餘的骨架分支。為改善此問題的發生，本演算法將對每個多餘節點區域中所有骨架節點的  $Neighbor_{26}(v_i)$ ，找出一個  $Neighbor_{26}(v_i)$  為終端節點或關節點情況。將此終端節點或關節點，及其所有  $Neighbor_{26}(v_i)$  接改設為多餘骨架節點。經由上述的多餘節點區域問題的排除，能於形成多餘節點區域的情況中，皆能有效的擷取出單位寬度曲線骨架 (如圖 4(d)所示)。

### 3.3 最短骨架節點路徑

要從多餘節點區域中擷取出單位寬度曲線骨架，在本篇演算法是使用最短路徑演算法。對每個多餘節點區域集合求出此多餘節點區域的質心，以此質心座標找其多餘節點區域集合中最接近質心的骨架節點。將此最接近質心的骨架節點為起點，利用 Dijkstra 最短路徑演算法[1]做為單位寬度骨架搜尋，以各多餘節點區域起點到多餘骨架關節點間的最短路徑，即為此單位寬度曲線骨架。接著只要剔除所有非最短路徑所經過的骨架節點，即為此多餘節點區域的單位寬度曲線骨架。

在以歐氏距離求得質心的方法中，對每個多餘節點區域的節點三維座標相加求得質量總和，接著再除以多餘節點區域的節點個數，即為此多餘節點區域的質心。而這個質心座標通常不一定會恰好坐落於多餘節點區域的任一節點上。我們將求得的質心座標，對多餘節點區域的每個節點求得兩點間的最短歐氏距離來找尋距離質心最短距離的節點，我們將此最短歐氏距離的節點設為質心。

但是這種質心搜尋法所找到的對應節點並無法確定位於物體中心，故我們將尚未細化處理的立體像素集合，計算出每個立體像素距離模型表面最短距離的距離圖(distance map) (如圖 6(上)所示)。接著，我們限制質心搜尋的區間，找出多餘節點區域中最接近物體中心的距離圖的數值區間，將收尋的質心節點限制在此區間內搜尋，即可確保質心能位於物體中心。

這個目的是為了避免區塊中最接近區塊質心的節點，所搜尋的最短路徑可能並非此多餘節點區域中最佳的骨架路徑。因為以最接近多餘節點區域的質心的節點為路徑搜尋起點的方法，是屬於折衷的路徑判斷 (如圖 6(中)所示)。我們並無法保證此多餘節點區域，以此最接近多餘節點區域的質心的節點為起點做搜尋，是一個最佳的骨架擷取方案，僅以此做為各種形成多餘節點區域的類型中一個

表 1.不同網格模型單位寬度曲線骨架擷取

Mesh Model	Voxelization	Curve-Thinning		Unit-Width Curve Skeleton			Mesh segmentation	
	voxel	voxel	Eage	voxel	Eage	Time (sec)	Bone	Time (sec)
Triceratops	31973	251	294	224	229	0.118	11	0.014
Dolphin	22640	241	434	170	173	0.111	7	0.017
Cow	64985	391	607	325	333	0.138	21	0.021
Octopus	24735	637	731	606	616	0.114	12	0.172
Hippo	135120	292	326	271	271	0.158	19	0.099
Horse	87620	348	351	344	343	0.148	15	0.312
Dinosaur	31692	294	313	285	284	0.129	12	0.271
Santa	84927	343	359	333	337	0.124	9	0.458
Dancing children	121855	627	666	623	650	0.163	47	0.917
Armadillo	140666	540	572	521	532	0.160	31	1.414
Fertility	58475	339	356	334	343	0.128	16	1.291

策略性的選擇。故以最接近物體中心的多餘節點區域質心節點，來做為單位寬度骨架的起點搜尋。如圖 6(下)所示以限制區間質心搜尋的單位寬度曲線骨架，能比圖 6(中)最短歐氏距離質心搜尋法的單位寬度曲線骨架更加符合骨架置中性。

### 3.4 骨架平滑化

三維曲線細化演算法受限於使用三維二元物件做為其資料結構，所擷取的曲線骨架會有鋸齒不平滑的情況。本篇演算法於單位寬度曲線骨架的擷取方法後，將擷取的曲線骨架節點求出平滑化後骨架節點。將曲線骨架節點中，不屬於終端節點或關節點的骨架節點做骨架的平滑化計算。平滑化後的骨架節點為此骨架節點的座標值及其  $Neighbor_{26}(v_i)$  的座標平均值。在本篇演算法使用此平滑化處理（如圖 7 所示），有效的緩和單位寬度曲線骨架鋸齒的情況，解決三維細化演算法常有的不平滑骨架問題。

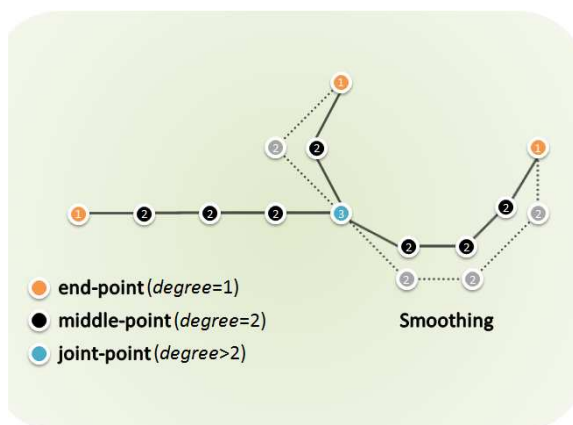


圖 7. 平滑骨架連線。

## 4. 實驗結果

本篇論文對各種不同模型所形成的多餘節點區域，所擷取的單位寬度曲線骨架結果（如圖 8 所示）。藉由本演算法實驗結果，能有效消除多餘節

點區域的骨架節點，其擷取的骨架也無任何循環並且符合骨架置中性的原則。

表 1 為本篇論文對各種不同模型的執行數據，本演算法於各種不同網格模型的骨架上，其計算時間平均只需要 0.1 秒。在本演算法消除骨架的多餘節點區域中，經由結果的數據顯示，本演算法能有效減少多餘的骨架連結，並且能於減少骨架連結後，還能保有骨架正確性（如圖 8 所示）。

以骨架對幾何模型做網格分割應用，是極其便利的。網格分割應用於很多不同的研究領域上，如電腦動畫、表面映射、網格變形、碰撞偵測、骨架擷取等。在本篇論文中，利用所擷取的單位寬度曲線骨架對網格模型所做的網格分割。首先對擷取的骨架模型，找出終端節點到關節點，或是關節點到關節點間的骨架節點，將成骨架關係的骨架節點集合生成相對應的骨頭。之後對網格模型，找尋與模型表面最恰和的骨頭，並將其互相做連結。

本篇論文是將模型表面與最接近的骨頭做為網格分割的依據（如圖 8 所示），以此方法對輸入模型做網格分割處理。將可利用本篇論文所提單位寬度曲線骨架對網格模型做電腦動畫，不同的骨頭來影響，輸入模型所對應的不同表面網格分割頂點。但是，由於三維細化演算法所擷取的骨架，經常會有擷取多餘的錯誤骨架分支問題。在實驗中，多餘骨架分支會造成網格分割數量過多或是分割不符合需求的情況。根據我們的觀察，這些問題主要是由兩種情況所造成的：其一為擷取的骨架中，擁有多餘的錯誤骨架分支，造成以骨架節點為網格分割的基礎的方法產生錯誤的分割；其二是由於本篇的網格分割方法是以模型表面與最接近的骨頭做為網格分割之依據。此方法並未參照網格模型本身的幾何關係。在圖 8 中，可以發現有些網格分割不盡符合解剖學。在解剖學中，一般於網格模型的骨頭與骨頭間的關節，同常會擁有幾何上角度較為變化劇烈的區域。在本篇網格分割應用上，並未詳加考慮網格模型的幾何關係，故會於網格分割結果影響結果的品質。另外，由於本篇的網格分割應用，是對網格模型原有的面上做分割。故模型的精



細度將會影響網格分割的品質。想要擁有較平滑的分割效果，需要對模型表面擁有更高的精細程度，同時也會造成演算法的效率降低。

## 5. 結論

本篇論文主要是提出解決三維曲線細化演算法可能形成多餘節點區域的問題。將多餘節點區域找其最接近物體中心的質心骨架節點，利用最短路徑演算法找出多餘節點區域中，擁有的相連骨架連結關係，來達到單位寬度的曲線骨架擷取，以及消除多餘節點區域目的和符合骨架置中性的原則。另外，在多餘節點區域分群步驟時，本篇演算法還考慮了，當以每個立體像素連結的所有二十六種的關係建構骨架時，可能會形成的循環問題。本篇演算法將可能形成循環的骨架節點，皆令其相互連結的骨架節點一同視為一個多餘節點區域問題來解決。本篇所提出的演算法確保了，即使以不同三維細化演算法所擷取的曲線骨架，皆能確保能萃取出單位寬度曲線骨架。

未來將針對三維曲線細化演算法所擷取的骨架，相較於其他類型的曲線骨架擷取演算法，可能擷取錯誤的額外骨架分支問題做改進。除此之外，本演算法的多餘節點區域骨架擷取，是以距離圖找尋最接近物體中心的多餘節點區域質心骨架節點，以此為多餘節點區域的關節點，並將此骨架節點來做單位寬度曲線骨架擷取方法。但是，以此骨架節點為起點所搜尋的骨架路徑，並無法保證必定是多餘節點區域中最佳骨架擷取路徑。未來將對此多餘節點區域的骨架搜尋起點做進一步改進，找出能更加符合網格模型解剖學的骨架擷取演算法。

## 致謝

本演算法實驗中所使用的三維網格模型，來至於 Aim@Shape 和 The Stanford 3D Scanning Repository 網站的網格模型檔，感謝這些網站提供網格模型做為本實驗測試。

## 參考文獻

- [1] Introduction to Algorithms. Cormen, T., Leiserson, C., Rivest, R., and Stein, C. 2001. McGraw-Hill.
- [2] A Parallel 3D 12-Subiteration Thinning Algorithm, K. Palágyi and A. Kuba, Graphical Models and Image Processing, vol. 61, no. 4, pp. 199-221, 1999.
- [3] A 3D 6-subiteration thinning algorithm for extracting medial lines, K. Palágyi, A. Kuba, Pattern Recognition Letters 19, 1998, 613-627.
- [4] A sequential 3D thinning algorithm and its medical applications, K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdőhelyi, K. Hausegger, in Proc. 17th Int. Conf. Information Processing in Medical Imaging, IPMI 2001, Davis, USA, Lecture Notes in Computer Science, Springer, 2001, to appear
- [5] Curve-Skeleton Properties, Applications, and Algorithms, Nicu D. Cornea, Deborah Silver, Member, IEEE, and Patrick Min, IEEE Transactions on Visualization and Computer Graphics Volume 13, Issue 3 (May 2007), Pages 530-548
- [6] Hardware Accelerated Voxelization, Shiao-fen Fang, Shiao-fen Fang, Hongsheng Chen, Hongsheng Chen, Computers and Graphics (2000)
- [7] Generation of Unit-Width Curve Skeletons Based on Valence Driven Spatial Median (VDSM), Tao Wang and Irene Cheng, International Symposium on Visual Computing (ISVC), LNCS 5358, pages 1061-1070, 2008
- [8] Mesh Segmentation Using the Object Skeleton Graph, D. Brunner, G. Brunnert., Proc. IASTED International Conf. on Computer Graphics and Imaging, 48-55, ACTA Press 2004.



圖 8. 本篇論文所提演算法的實驗結果。最左邊為骨架擁有多餘節點區域的情況。接著是單位寬度曲線骨架和為對骨架的骨頭以不同顏色著色。最右邊為表面分割的結果。