

國立交通大學

多媒體工程研究所

碩士論文

利用 Feature-Based Metamorphosis 對 Brain Data
進行變形



Deform Brain Data using Feature-Based Metamorphosis

研究生：楊博鈞

指導教授：荊宇泰 博士

中華民國九十六年八月

利用 Feature-Based Metamorphosis 對 Brain Data 進行變形

學生：楊博鈞

指導教授：荊宇泰 博士

國立交通大學多媒體工程研究所



由於 Feature-Based Metamorphosis 有著隨著特徵點的對應，而改變外型，這樣一來可以看出一些因為變形而改變的地方，這裡我們提出一種利用 Triangulation 和 Feature-Based Metamorphosis 來對 3D data 進行變形的動作，必且不去破壞整體的結構。

Deform Brain Data using Feature-Based Metamorphosis

Student : Po-Chung Yang

Advisor : Dr. Yu-Tai Ching

Institute of Multimedia Engineering
National Chiao Tung University



Abstract

Because the way of Feature-Based Metamorphosis can deform with feature, we can observe the change of the part which we interest. Here, we provide the way of deformation of 3D data using 3D triangulation and Feature-Based Metamorphosis. It can maintain the construction of 3D model.

致謝

能完成這篇論文，我想感謝荆宇泰老師、我的父母、實驗室的學長和同學們，跟支持我的朋友們。最後，引用陳之藩先生的話，那就謝天吧。



目錄

摘要.....	i
英文摘要.....	ii
致謝.....	iii
目錄.....	iv
圖表目錄.....	vi
第一章 緒論.....	1
1.1 簡介.....	1
1.2 研究動機.....	1
1.3 論文架構.....	2
第二章 文獻回顧.....	3
2.1 前言.....	3
2.2 Feature-Based Image Metamorphosis.....	3
2.3 Feature-Based Volume Metamorphosis.....	7
第三章 程式流程.....	9
3.1 簡介.....	9
3.1 初始.....	9
3.2 三角化.....	11
3.2.1 概論.....	11
3.2.2 Delaunay Triangulation.....	14
3.3 Wrapping.....	17
3.4 特殊處理（不同維度下 Morrphing）.....	20
3.5 流程圖.....	22
第四章 實驗結果.....	24
第五章 未來展望.....	38



圖表目錄

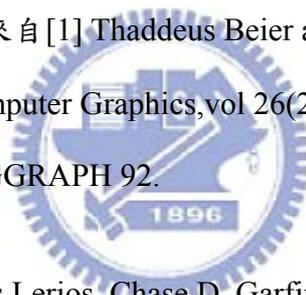
圖【2-1】 Feature based image metamorphosis 單一 control line 的狀況.....	4
圖【2-2】 Feature based image metamorphosis 多條 control line 的狀況.....	6
圖【2-3】 Feature based volume metamorphosis wrap data 時的問題.....	8
圖【3-1】 特徵點選取位置(1).....	10
圖【3-2】 特徵點選取位置(2).....	10
圖【3-3】 特徵點選取位置(3).....	10
圖【3-4】 特徵點選取位置(4).....	10
圖【3-5】 特徵點選取位置(5).....	11
圖【3-6】 特徵點選取位置(6).....	11
圖【3-7】 特徵點選取位置(7).....	11
圖【3-8】 特徵點選取位置(8).....	11
圖【3-9】 三角形取樣問題.....	12
圖【3-10】 一般 3D Triangulation 的結果.....	13
圖【3-11】 使用一般 3D Triangulation Morphing 後的結果(1).....	13
圖【3-12】 使用一般 3D Triangulation Morphing 後的結果(2).....	13
圖【3-13】 Delaunay Triangulation 外接圓檢查.....	14
圖【3-14】 Delaunay Triangulation 流程 1.....	15

圖【3-15】Delaunay Triangulation 流程 2.....	15
圖【3-16】Delaunay Triangulation 流程 3.....	16
圖【3-17】3D Delaunay Triangulation 結果.....	16
圖【3-18】Delaunay Triangulation Morphing 後結果(1).....	17
圖【3-19】Delaunay Triangulation Morphing 後結果(2).....	17
圖【3-20】對 Cell 的 Wrapping 過程.....	18
圖【3-21】ICP 後部分腦跟全腦.....	21
圖【3-22】Morphing 的流程圖.....	22
圖【3-23】部分腦跟全腦的 Morphing 流程.....	23
圖【4-1】Model 001.....	25
圖【4-2】Model 002.....	25
圖【4-3】Model 003.....	26
圖【4-4】Model 004.....	26
圖【4-5】Model 001 morph 成 002 的結果.....	27
圖【4-6】Model 001 morph 成 003 的結果.....	28
圖【4-7】Model 001 morph 成 004 的結果.....	28
圖【4-8】Model 004 morph 成 001 的結果.....	29
圖【4-9】Model 001 morph 成 001 的結果(背景不同).....	30
圖【4-10】歪斜腦 Data(1).....	32



圖【4-11】歪斜腦 Data(2).....	32
圖【4-12】歪斜腦 Data(1) Morph 成歪斜腦 Data(2).....	33
圖【4-13】全腦.....	34
圖【4-14】部分腦.....	34
圖【4-15】部分腦 morph 成全腦的結果.....	35
圖【4-16】全腦(不連續).....	36
圖【4-17】部分腦(不連續).....	36
圖【4-18】部分腦 morph 成全腦的結果(不連續).....	37

註：圖【2-1】跟圖【2-2】來自[1] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In Computer Graphics, vol 26(2), pp 35–42, New York, NY, July 1992. Proceedings of SIGGRAPH 92.



圖【2-2】來自[2] Apostolos Leros, Chase D. Garfinkle, and Marc Levoy. Feature-Based Volume Metamorphosis. In Computer Graphics, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 449-456, Proceedings of SIGGRAPH '95

第一章

緒論

1.1 簡介

Feature-based Metamorphosis 這個方法已經運用在各個領域上很久，像是早期的電影特效處理，或是 3D 的臉型變形，都可以得到不錯的結果。因此，這裡我們提出以同樣的觀念來應用到生醫影像處理上。

在腦部研究上，要瞭解腦神經的運作方式，是利用不同的外在刺激來觀察神經反應；進而瞭解觸發了哪些神經，藉此將大腦劃分成不同的功能區，不過這些研究常常會包含了數以百萬的神經元。人類出生時約有一千億個神經元，就和銀河裡的星星一樣多，即使有好的技術也難以觀察，所以需要有一個好的模式生物來研究。

常見的生物模式，從高等到低等有老鼠、果蠅、線蟲、以及大腸桿菌。在生物研究領域裡，果蠅一直是一種重要的模式生物。與其它模式生物相比較，老鼠雖然與人類相近，但是複雜度過高，並且以往對老鼠的神經科學瞭解不如對果蠅透徹；大腸桿菌屬於單細胞生物，並沒有神經系統；線蟲更複雜一些，但與人類相比，仍然過於簡單。從世代交替的角度來看（世代交替是指生物從出生到生育下一代的循環週期），果蠅的世代交替時間約為十日，而人類的世代交替約為一萬日。相較之下，在神經科學領域利用果蠅來當作模式生物培養和觀察有其優勢。

在觀察果蠅神經時，可以透過各種染色方式顯示想研究的區域。假設我們要觀察兩組神經之間的關係，兩組神經取自同一隻果蠅是最理想的情況；

但是每次實驗犧牲的是不同隻果蠅。受限於每組神經影像會取自不同的果蠅，而且每隻果蠅頭的形狀大小多少會有差異，再加上取像時顯微鏡的倍率不同，所以將兩組神經放進同一組頭之中是一個高難度的 warping 的問題。

另一個更具野心的應用是建立果蠅腦資料庫的問題。果蠅有十萬個神經元，如果每一個都可以被成功的解碼並知道其功能，則我們必須有一個果蠅全腦地圖。因為每個神經都是由不同的個體得到，所以我們必須將其變形之後並全部存入(deposit)一個標準腦。更進一步，我們會希望訂出一個敘述果蠅腦地圖中的座標的方法，以描述一個神經細胞核或是一個神經的地理位置。

目前全世界居領導地位的果蠅實驗室都在做類似的工作。本碩士論文提出一個半自動的果蠅腦 warping 的方法。果蠅腦 warping 的概念與 Feature based Metamorphosis 的觀念很像，都必須以固定住對應的特徵來改變 image 或是 volume。我們所提出的方法是使用者必須在幾個重要 neuralpils 上訂出重要但是數目不多的點集。根據這些點集三度空間三角化的結果，我們將一個果蠅腦 warp 到另一個果蠅腦。

1.2 論文架構

在接下來在第二章會介紹一些 Feature-Based Metamorphosis 在不同方面的應用，第三章介紹我們所使用的方法和演算法流程，在第四章為 deform 後的結果，最後一章為未來的研究方向。

第二章

背景及文獻回顧

2.1 前言

Morphing 是常用來處理兩張圖片互相變形的技術，可以使用在很多方面，例如應用在影片的特效之中。而在實作時最簡單的方法就是把第一張圖片漸漸的淡化，同時把第二張圖片所佔的比重漸漸增加，這種方法在兩張圖片輪廓很接近時是很好用的。在 Morphing 的步驟裡，會分成 Warp 跟 Blending，不過因為在我們目的裡，是希望把 Source 改變成 Target，不需要有 Target 的資訊存在，所以 Warp 的方法會顯得比較重要。

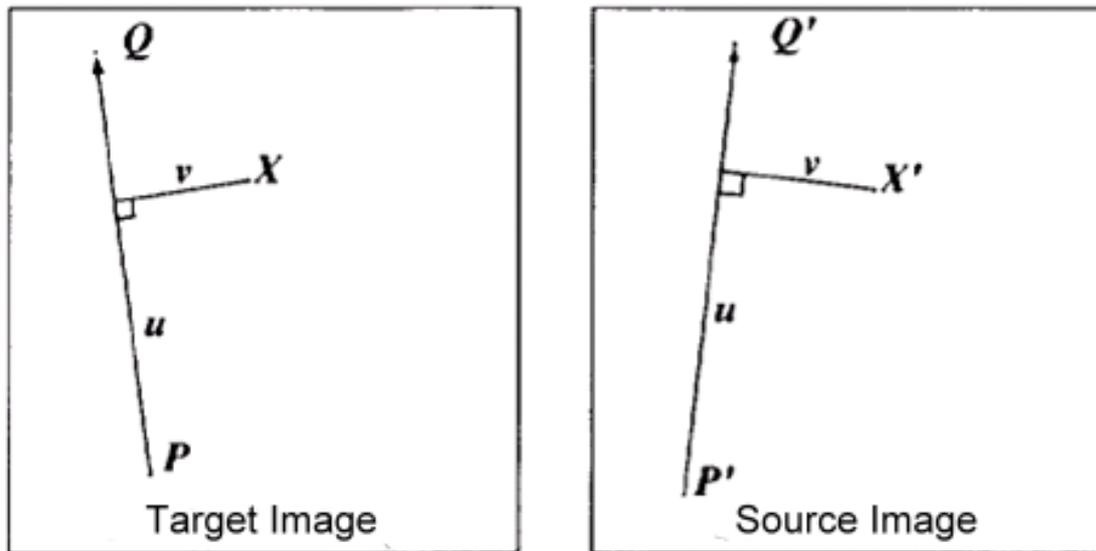


而 Warp 會需要注意 mapping 的方法跟選取的原則。在 mapping 的方法有兩種。第一種是“forward mapping”，首先我們會 pixel by pixel 地掃描 source image，然後把每個點複製到 target image 上相對的位置上。使用這個方法的優點是轉換的過程很直觀，缺點是有些 target image 上的位置並不會被填上顏色。所以 mapping 方法大都是使用“reverse mapping”，也就是先掃描 target image 上的每一個點，再找出在 source image 上的相對應位置的像素。用這個方法做 mapping 就不會有畫素沒有被填上的問題。接下來就是如何去利用幾何關係來選擇適當的點，完成整個 Warp 的動作

2.2 Feature-Based Image Metamorphosis

在這篇 paper 裡，是用“reverse mapping”來做 mapping 的動作，選取點的方法是利用 control line 跟點的關係來做轉換，這裡使用的方法像是把點轉換成另一個座標系統，在從其中找出幾何的關係。下面介紹一些這個方法的流程。

一開始有兩張影像，分別叫作 Source 跟 Target，分別給這兩張一些 control line，做好對應關係，先對其中一條 control line 來看，假設在 Target 這邊的 control line 為點 P 到點 Q，而 Source 這邊的 control line 為點 P' 到點 Q'，如圖【2-1】所示：



圖【2-1】

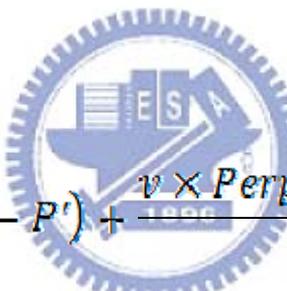
對於 \overline{PQ} 來說點 X 會是距離它垂直距離是 v ，而水平距離是 u 的地方，理論上在 Source 要轉換成 Target 時，在 Target 上的 control line 變成 Source 上 control line 的樣子時，原本在 Target 上的 X 也要能維持著在變化前的幾何關係（因為是”reverse mapping”，所以要從 Target 反回去找 Source），因此接著計算出 Target 中的每一個點 X 跟這條 control line 的幾何關係，而這些動作像是 control line 能形成一個座標系統，則點 X 可以被轉換成座標 (u, v) ，而 u 表示到 control line 的平行距離， v 表示到 control line 的垂直距離。

計算 (u, v) 公式如下：

$$u = \frac{(X-P) \times (Q-P)}{\|Q-P\|^2}$$

$$v = \frac{(X-P) \times \text{Perpendicular}(Q-P)}{\|Q-P\|^2}$$

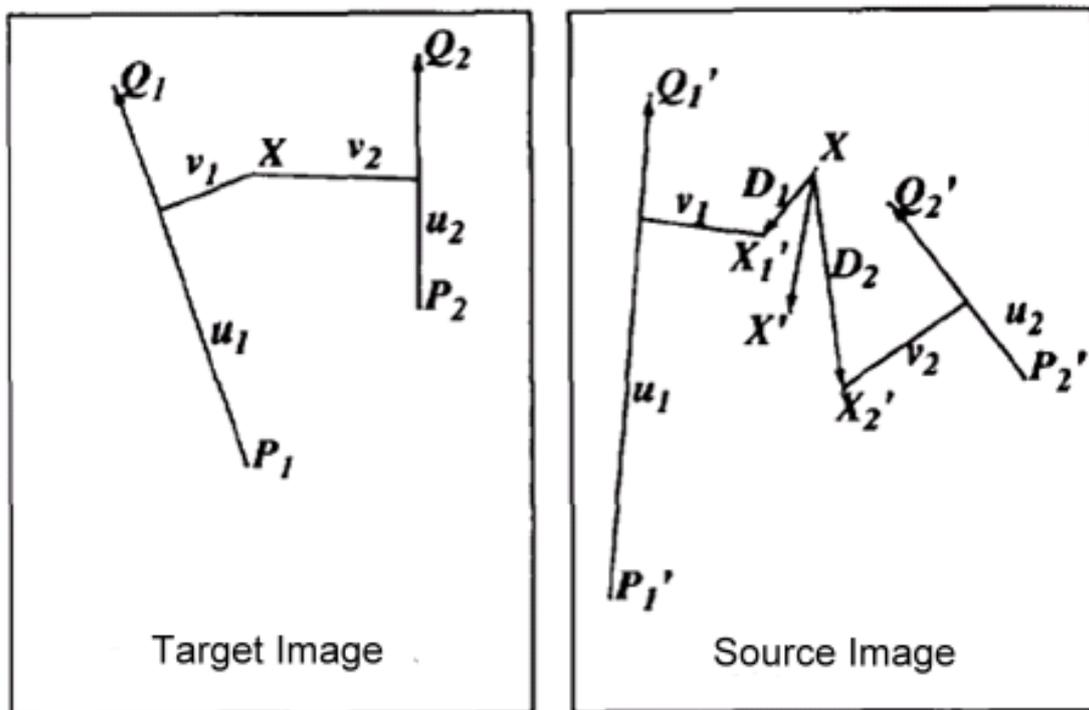
接下來為了把這個幾何關係能維持在轉換過程中，在 Source 相對應的 control line 也需要看作是一個座標系統，算出在這個座標系統的 (u, v) 是在一般的座標系統中的點位，就會是我們所需要的點，所以在 Target 點 X 相對應的在 Source 點 X' 座標為：



$$x' = p' + u \times (Q' - P') + \frac{v \times \text{Perpendicular}(Q' - P')}{\|Q' - P'\|}$$

之後只要把計算出的點 X' 的顏色，填入 X 座標上，就可以看出 control line 附近的幾何關係，和其影響的結果。

不過一般來說特徵不會是只有一個，在這樣的狀況中，為了呈現出每個 control line 的交互影響，就需要有一個機制，來去找出最後適合的結果，因此在 multi-pair control lines 下，需要知道每一條 control line 的適當影響範圍，假設有兩條 control line 為 $\overline{P_1Q_1}$ 、 $\overline{P_2Q_2}$ ，如圖【2-2】所示：



圖【2-2】

為了去了解每一條 control line 對這個點會影響多少，需要就利用 *weight* 來決定每一條 control line 來決定影響的範圍，計算 *weight* 的公式為：

$$weight = \left(\frac{length^p}{(a + dist)} \right)^b$$

dist 為點 *X* 對 Target 上的 control line 的最短距離，*length* 為 control line 的長度，*a*、*b*、*p* 為調整的參數，可以看出這個 *weight* 的特性：

1. 離 control line 越近的点，影響越大。
2. Control line 愈長，影響的範圍愈廣。

當計算出所有 control line 的 $weight_i$ 後，先算出所有 *weight* 的總和 *total*，把 $weight_i$ 除以 *total*，算出百分比 p_i ，之後把每條 control line 所得到的 X_i' 乘上 p_i ，

如同決定出這條 control line 對點 X 能影響，最後把這些值加起來，也就是

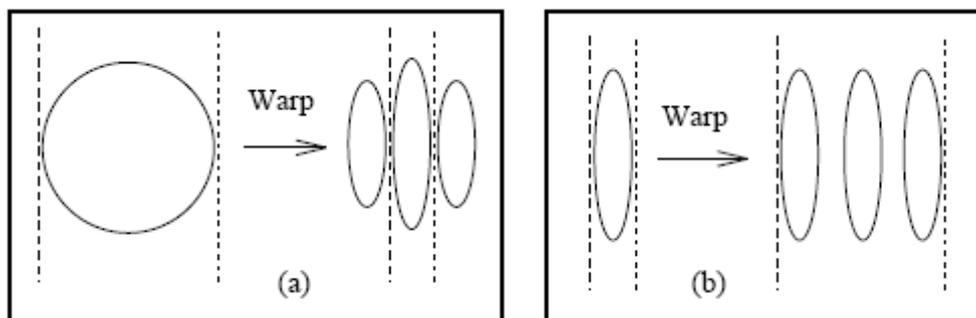
$X' = \sum x_i$ ，為所有 control line 影響點 X 的最後位置。

2.3 Feature-Based Volume Metamorphosis

把 Feature-Based Morphing 的方法用在 3D 中時，特徵的表示方法就不限於是線或是點的關係，就可以是 box 等 3D 物件，來框住特徵。這個方法大致跟之前的 2D 相同，一樣是分成兩個步驟：

1. Warping

這裡大致跟 2D 的方法差不多，利用 Target 上的點跟 control element 關係來定義出座標關係，不過為了消除鬼影。多加了 scaling factor S_x 、 S_y 、 S_z ，來將 Target 跟 Source 對應 control element 有好的對應關係，如圖【2-3】(a)中，假設在 Target 中圓的兩旁都有 control line 存在，希望能在 Source 中能被壓成橢圓，若是沒考慮到 scaling factor，會讓 control line 兩側也產生出兩個小的橢圓；同理，在(b)中希望能把橢圓拉成圓形，一樣是沒考慮 scaling factor，會在 Source 生成出 3 個橢圓，而不是一個圓的出現。這種情形會在 3D 的 control element 中出現，因為在定義的 element 中也許會是 Rectangle 或是 Box，就會有 control line 包住的狀況，所以在 Warp 之前先把 Source 的 control element 除上 scaling factor，可以避免產生出錯誤的幾何圖形。



圖【2-3】

2. Blending

在這篇 paper 中提出一個不同的 Blending 的方法，稍微介紹一下。在作 Blending 時，大都是用一般的 linear cross-dissolving 方法來做，也就是依照比率來去 interpolate 出中間的顏色，但是 linear cross-dissolving 的方法會在某些地方出現問題，像是原本是透明的區塊會產生出錯誤的物體出現，而影響判斷，所以在這裡提出 non-linear cross-dissolving 的方法，來去決定最後的結果會在 Target 跟 Source 中占多少的比率，而這個計算比率的 function 是憑著經驗跟實驗慢慢去得來，雖然不一定可以滿足所有的 data，但是可以藉由調整來減少錯誤的訊息產生。

在大略介紹之前使用過的 Feature-Based Metamorphosis 後，在下一章節會提出我們所使用的方法，是如何去 deform brain model，而不去破壞原有的特徵。

第三章

程式流程

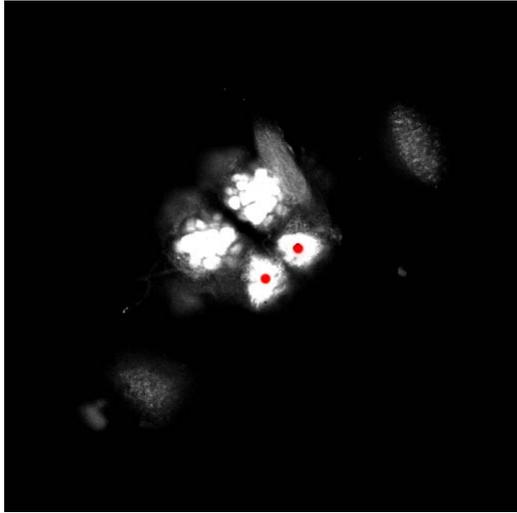
3.1 簡介

如果要把兩組 Data 作變形，最簡單的方法就是依照比例去做壓縮，但是這樣的結果，只有單純的改變外型，不會依照特徵來變化；因此為了方便觀察，須要給定特徵點，來指出感興趣的部分；之後利用這些特徵點，定義出周遭點的幾何關係，之後就可以用來作 Warp 的動作。

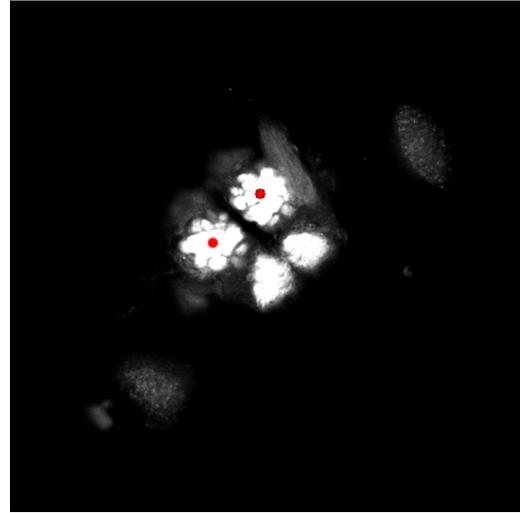
所以整個程式流程會分成三個部分：1. 初始 2. 三角化 3. Warp。並且在最後一節另外說明對於不同維度的 Data，要如何做 Morphing。

3.2 初始

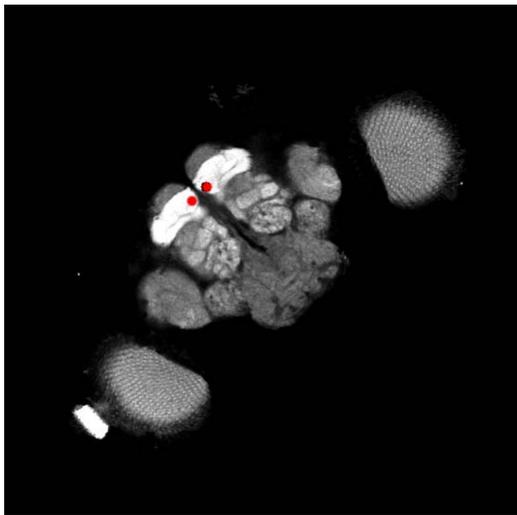
在一開始，先必須對兩個 model（分別稱做 Source 和 Target）決定好特徵點的對應，才可以進行三角化的動作，來切割這兩組 model 的空間。每一組特徵點可以是嗅小球的中心、口器兩旁的中心、眼睛的中心、大約有 15 個特徵點，在加上 8 個頂點的位置（也就是最外圍的立方體 8 個頂點），這兩組點集分別稱做 source points 和 target points，下面幾張圖為我們選取特徵點的位置。



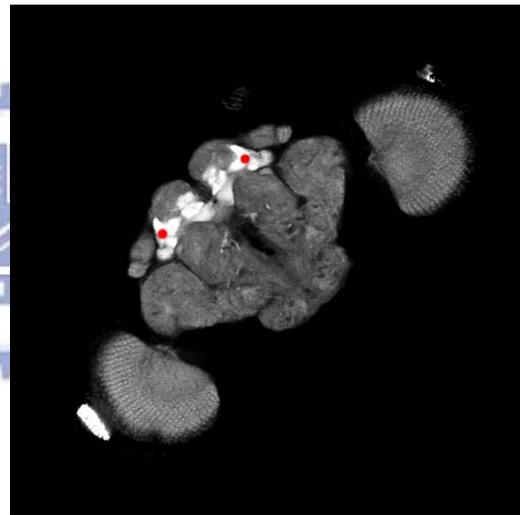
圖【3-1】



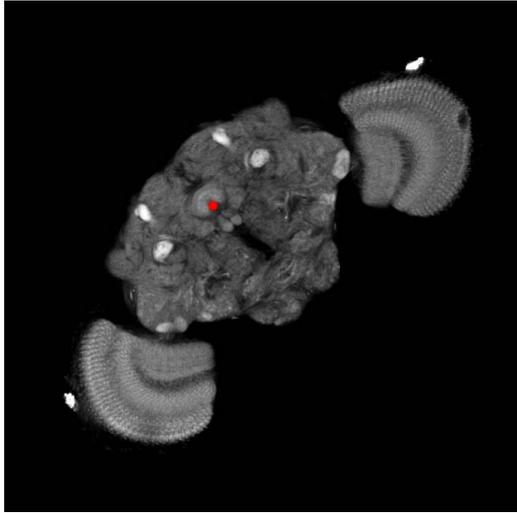
圖【3-2】



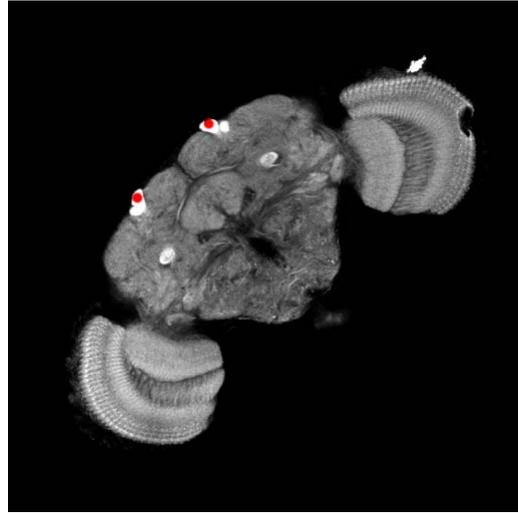
圖【3-3】



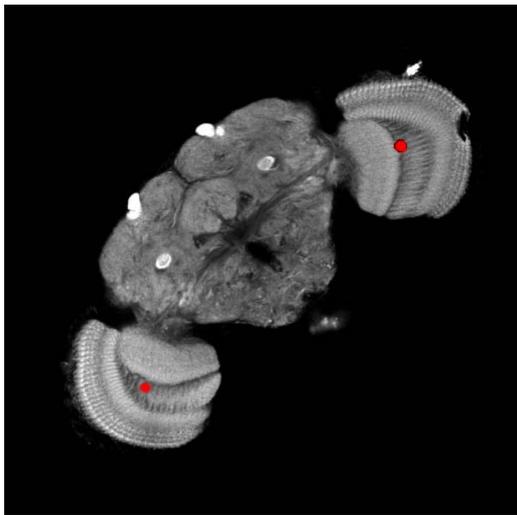
圖【3-4】



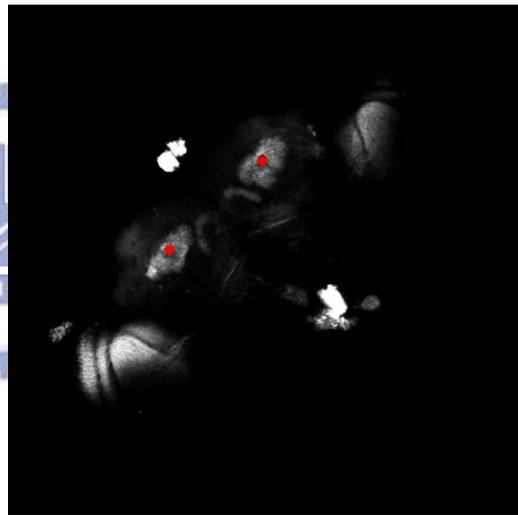
圖【3-5】



圖【3-6】



圖【3-7】



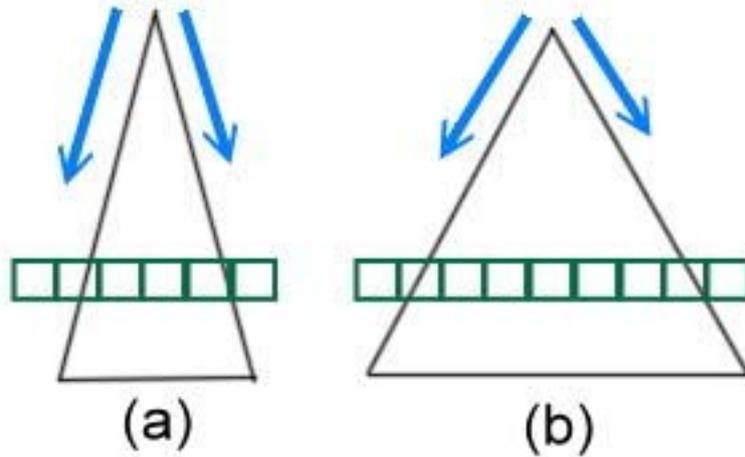
圖【3-8】

3.3 三角化

3.3.1 概論

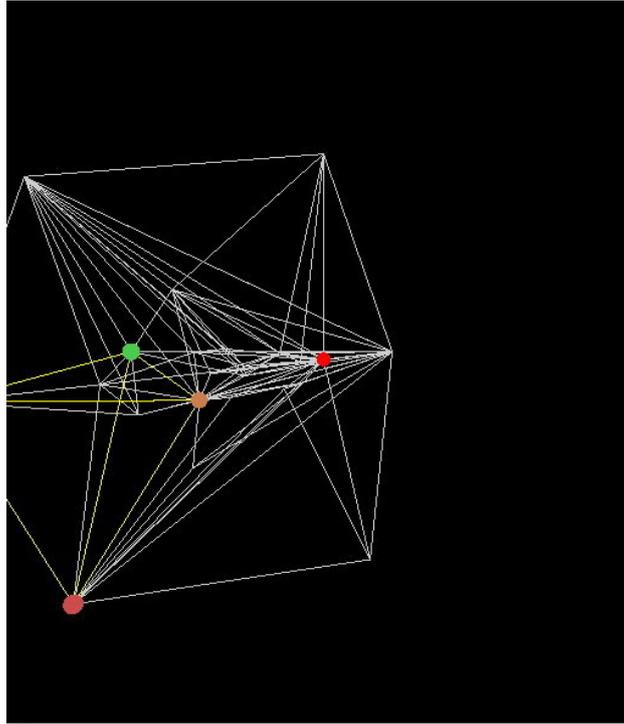
決定好特徵點後，就要利用這些點產生出多組的 tetrahedron（下面通稱為 Cell），來劃分空間，這麼一來就像是把 Target cell 的其中一組壓縮成對應的 Source Cell，而在這裡三角形的好壞就決定之後 Morphing 得結果，判定三角形

的好壞是在於最小角的值是多少，也就是希望最小角的角度不能太小，如果角度太小容易產生誤差。



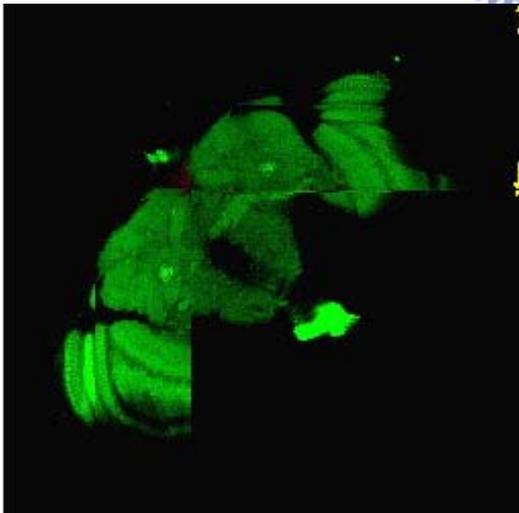
【圖 3-9】

像是在【圖 3-9】的 (a)，若是三角所得到的三角形是這樣，在下一個步驟 Morph 可以參考的點只有 3 個，如此一來所得到的效果就有被壓縮過的感覺，而且因為參考點太少，反而讓兩個相鄰的三角形，所選取的點彼此間會有一定的差距，Morph 後的結果看起來不連續。【圖 3-10】為用一般 Triangulation 的方法所得到的結果，可以看出存在許多 bad triangle（也就是最小角過小的三角形）。

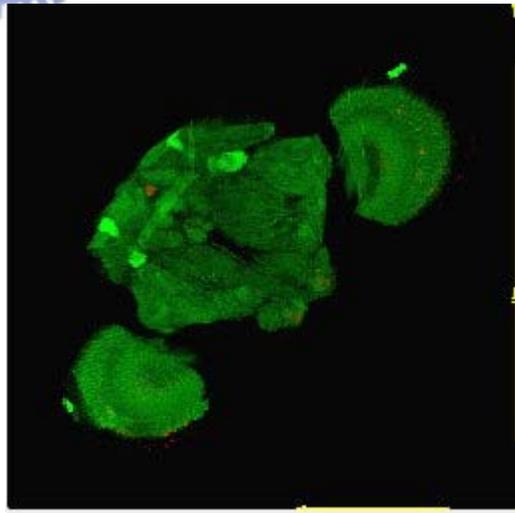


【圖 3-10】

如此一來所得到的結果可能就會是破碎、不連續，如【圖 3-11】跟【圖 3-12】



【圖 3-11】



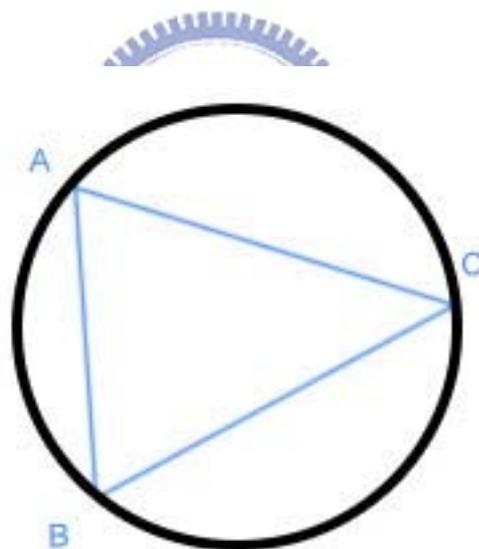
【圖 3-12】

3.3.2 Delaunay Triangulation

為了避免產生 bad triangle，在這裡所使用的方法改成 Delaunay Triangulation，用這個方法的好處在於

1. 最大化最小角，使結果最接近於 Regular 的 Triangulation。
2. 唯一性（若任意四點不能共圓）。

Delaunay Triangulation 的方法是檢查每一個三角形的外接圓裡面是否有其他的點，這樣的三角形稱作 Delaunay Triangle。如【圖 3-13】，三角形 abc 的外接圓裡不存在任何點，就停止檢查。



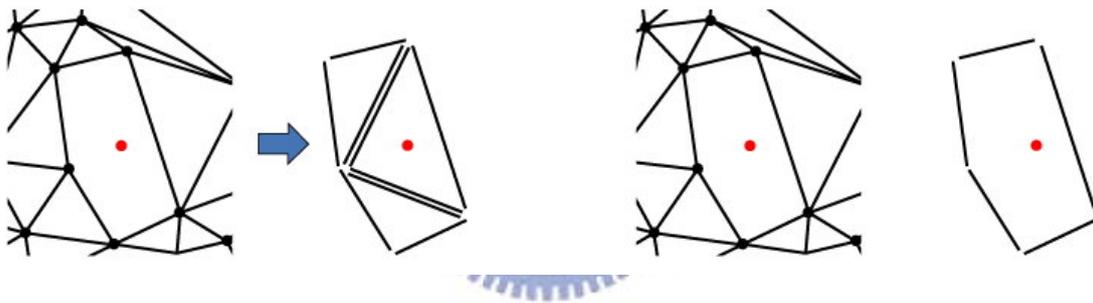
【圖 3-13】

所以一開始，我們依序把點插入已經完成 Delaunay Triangulation 裡面，開始進行外接圓的檢測，若是三角形的外接圓內包含著其他點，則必須破壞三角形，如圖【圖 3-14】，在檢查的過程中有紅點剛好在三個三角形的外接圓上。



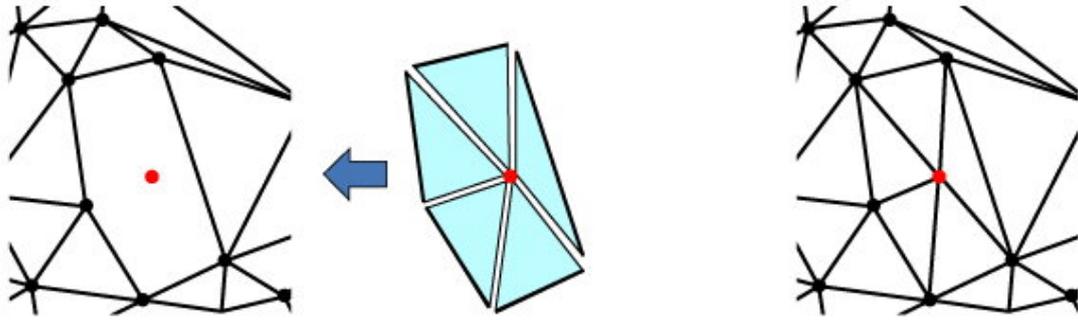
【圖 3-14】

把包含紅點的外接圓三角形移除，但是要記錄 edge 跟點，再來移除掉有 doubled 的 edge。如此一來就可以得到哪些點跟邊需要重新三角化，而不至於破壞已經完成的 Delaunay Triangle。如【圖 3-15】



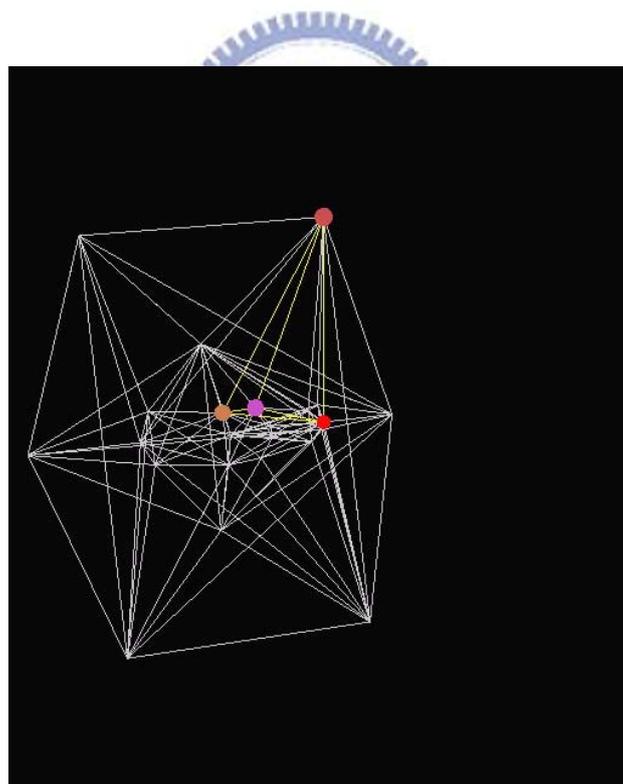
【圖 3-15】

接著把所得到的點跟紅點作 Delaunay Triangulation，產生出新的三角形，來維持 Delaunay Triangulation 的特性。



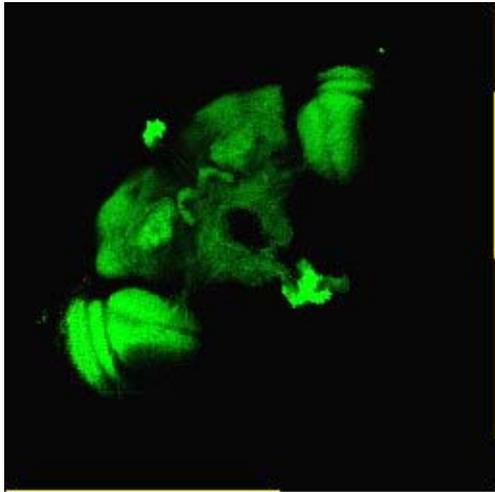
【圖 3-16】

【圖 3-17】為我們演算法中，所產生出的 3D Delaunay Triangulation，可以看出比之前的 Triangulation 去掉許多銳角三角形，產生出更多的三角形分割，把空間劃分的更詳細。

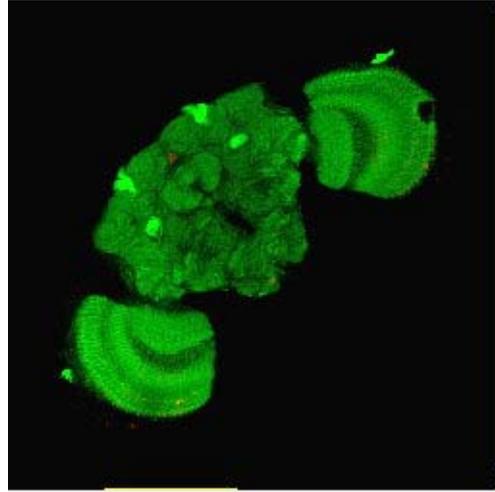


【圖 3-17】

這樣的分割，在 morphing 的最後結果，看起來也比較完整。



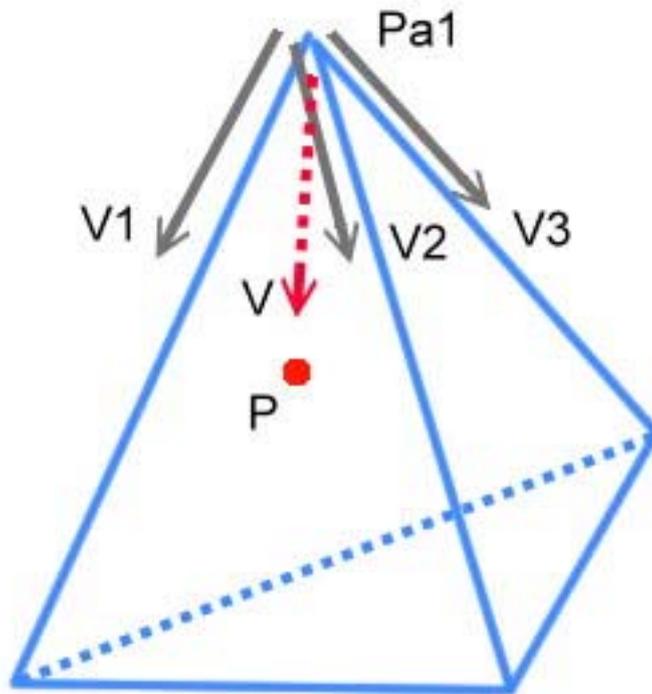
【圖 3-18】



【圖 3-19】

3.4 Warp

在上一個步驟中，可以把 data 分成 Target Cell 跟 Source Cell 兩組三角化的結果，接著要把 Source 的 data 變形成 Target 的樣子，因為事先把兩組 data 的特徵點對應好，所以在 Target Cell 裡的每一個 Cell（也就是 tetrahedron）可以找出在 Source Cell 裡對好的 Cell；接著開是描述點 p 跟 Cell 的幾何關係，假設知道點 p 在 Target Cell 裡的某一個 cell 稱為 Tc，為了描述 p 跟 Tc 之間的關係，我們利用 Cell 的頂點跟由這個點連成的三個邊來描述，也就是先將某一頂點 Pa1 作為原點，以及跟這個頂點相連的三個邊所形成的三個單位向量（V1、V2、V3）新的座標系統，以及因為 Pa1 為原點後，點 p 跟 Pa1 的向量 V。



【圖 3-20】

接下來類似 Feature based image metamorphosis 裡所提到的方法，去算出 Target 新的座標系的座標，再轉換到 Source 相對應的座標。也就是假設 P 跟 Pa1 的向量 $V=(x,y,z)$ ，而單位向量 $V_{i1} = (x_{i1}, y_{i1}, z_{i1})$ ， $V_{i2} = (x_{i2}, y_{i2}, z_{i2})$ ， $V_{i3} = (x_{i3}, y_{i3}, z_{i3})$ ，若 $V = aV_{i1} + bV_{i2} + cV_{i3}$ ，則要解出新的座標 (a, b, c)。也就是解連立方程式，如下：

$$x = ax_{i1} + bx_{i2} + cx_{i3}$$

$$y = ay_{i1} + by_{i2} + cy_{i3}$$

$$z = az_{i1} + bz_{i2} + cz_{i3}$$

解出 a、b、c 後，在 Source Cell 裡找出對應的 Cell，同樣的頂點上，找出

對應的三個單位向量 $V_{s1} = (x_{s1}, y_{s1}, z_{s1})$ ， $V_{s2} = (x_{s2}, y_{s2}, z_{s2})$ ， $V_{s3} = (x_{s3}, y_{s3}, z_{s3})$ ，再把之前計算出的線性組合 a、b、c 帶入。於是可得到新的點 $Result_1 = (x_{r1}, y_{r1}, z_{r1})$ ，

$$x_{r1} = ax_{s1} + bx_{s2} + cx_{s3}$$

$$y_{r1} = ay_{s1} + by_{s2} + cy_{s3}$$

$$z_{r1} = az_{s1} + bz_{s2} + cz_{s3}$$

同理可以對另外三個頂點利用這樣的幾何關係，找出另外三個點位，假設每個頂點所求出的結果分別稱做 $Result_1$ 、 $Result_2$ 、 $Result_3$ 、 $Result_4$ ，理論上因為每個頂點都是特徵點，因此這些頂點的周遭要是變動最少的，也就是越靠近頂點的點位，應該是要影響越大，根據這樣的原則，設計出下面的權重來找出適合的點，公式為：



$$weight_i = \frac{1}{(1 + d_i^2)}$$

其中 d_i 表示點 P 到 Target Cell 的第 i 個頂點的距離。

所以最後的點會是：

$$Result = \sum_{i=1}^4 Result_i \times weight_i / T$$

這裡 $T = weight_1 + weight_2 + weight_3 + weight_4$ 。

之後對 Target 每個點都去做計算，在 Source 中找出合適的點位，之後填入其顏色，就可以得到最後的結果。

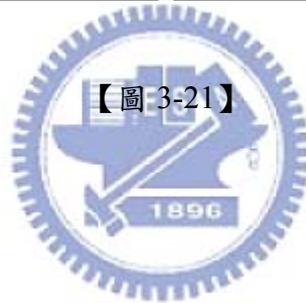
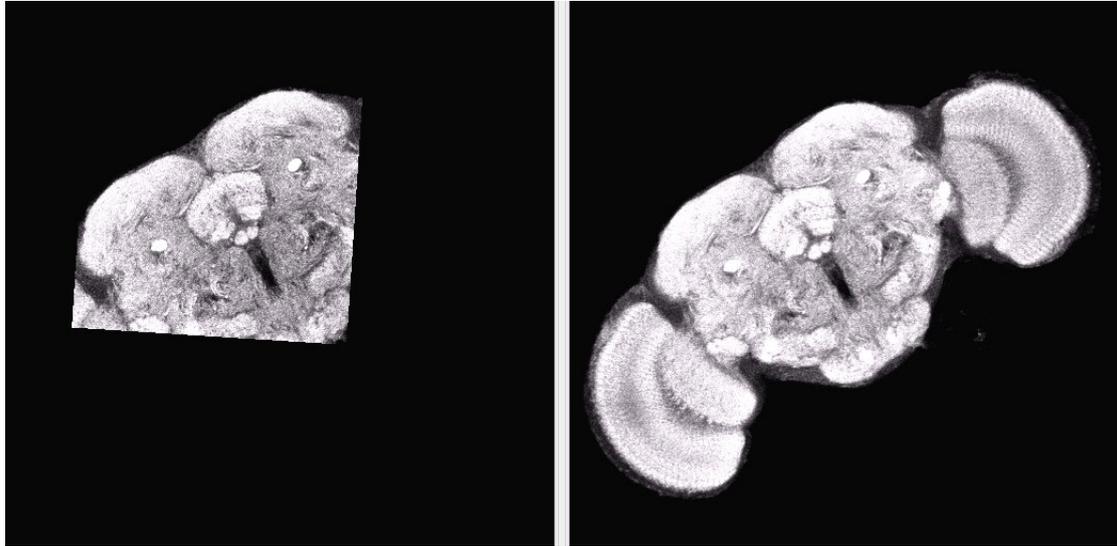
3.5 特殊處理（不同維度下 Morphing）

在處理部分腦跟全腦的 deform 時，部分腦是掃描比較精細的 data，但是缺少某些部份的資料，而全腦是完整但是比較粗糙的掃描，因為兩者間的 scale 不同（也就是維度上不同），會先將部分腦依照兩者的物理距離比作適當的 down sample，將兩者的 scale 調成一樣，這裡的物理距離可由 LSM 檔案中的資訊求得。

由於部分腦的圖裡沒有眼睛的部分，而我們所取的特徵點裡有眼睛的點位，所以要先將全腦的眼睛特徵點，利用兩者 Zsize（也就是掃描的張數）的比率，在部分腦裡找出適合的眼睛位置，來當作部分腦的虛擬眼睛特徵點，接下來利用 ICP 找出 rotation matrix 跟 translation matrix，將 down sample 後的圖調整到適當位置（包括所選取的特徵點），如此一來可以有比較好的結果，不會有太大到誤差產生。之後就可以直接進行我們的方法。

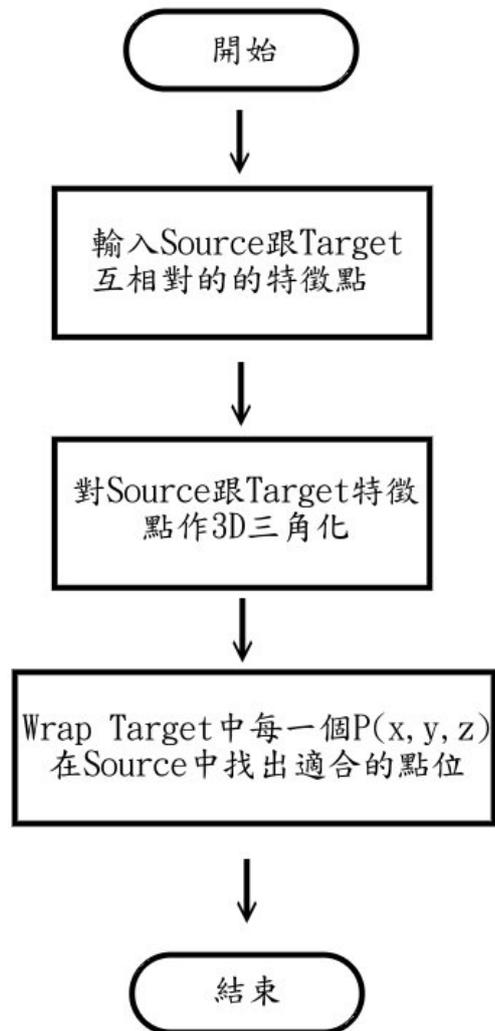
下面介紹一下 ICP 的流程，ICP (Iterative closet point) 主要是為了校準兩組資料模型的幾何結構，這裡希望能盡量縮小 Source 跟 Target 間距離下，找出適當的對應關係。ICP 演算法是用於兩個不知道對應關係的點集上作校準，計算出一組最小平方的轉換矩陣，迭代縮小兩物體間的 MSD (mean square distance)，最後算出最好的轉換矩陣。雖然 ICP 會計算出一個收斂到最佳值的轉換矩陣，但這個轉換矩陣是一個區域最佳值 (local minimum)，並不保證是整體最佳值 (global minimum)。所以在 ICP 原始論文中建議，可以對所要校準的物體作不同角度的旋轉與轉移後再執行校準，把這些經過不同幾何轉換後的物體拿來作 ICP 計算後，所可以得出最小 MSD 值的解，就是所要逼近整體最佳值的解。

【圖 3-21】為 ICP 後，部分腦跟全腦的比較，Morphing 之後的放在第四章中。



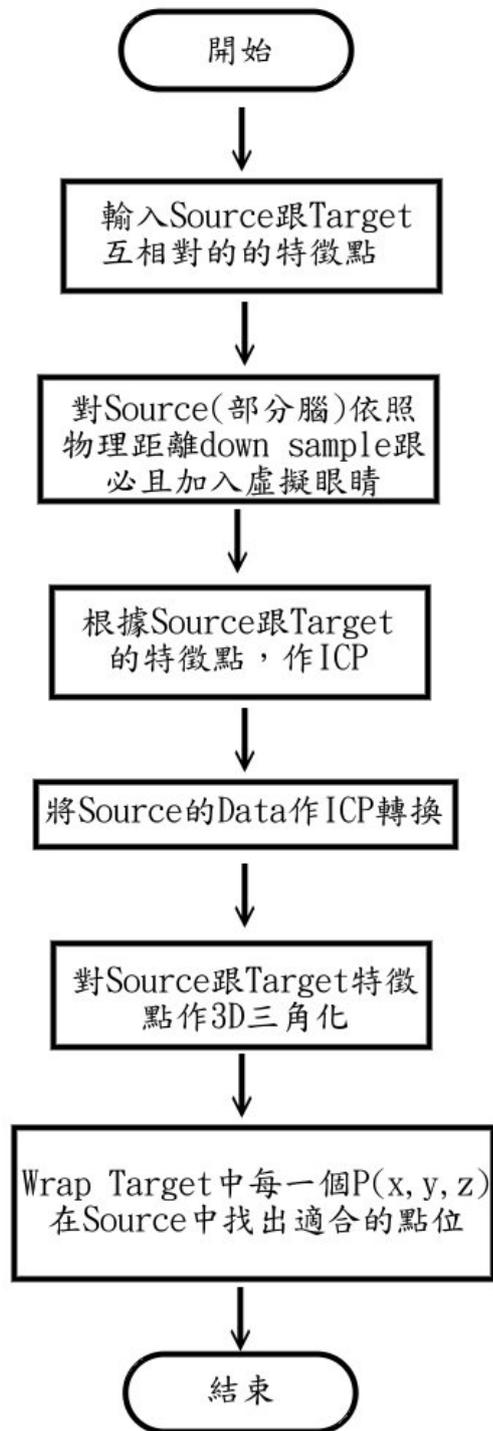
3.6 流程圖

Morphing 的流程圖：



【圖 3-22】

部分腦跟全腦的 Morphing 流程：



【圖 3-23】

第四章

實驗結果

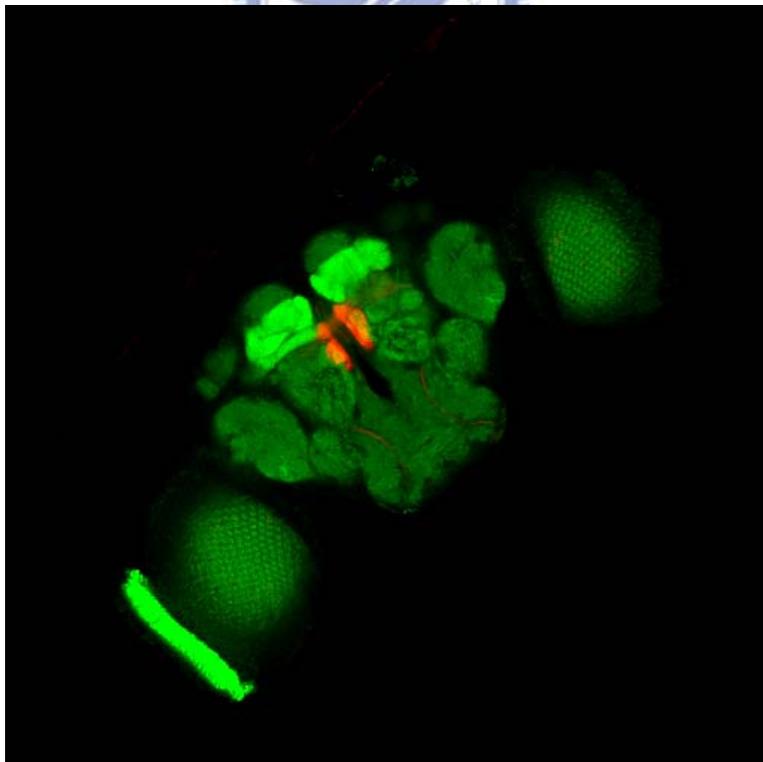
第一個實驗是以四個全腦來做 Morphing，每一組的大小都不相同，如下圖表

編號	維度
001	$512 \times 512 \times 84$
002	$512 \times 512 \times 67$
003	$512 \times 512 \times 87$
004	$512 \times 512 \times 81$

下面為每一個 Data 的樣子，圖【4-1】為 001，圖【4-2】為 002，圖【4-3】為 003，圖【4-4】為 004。



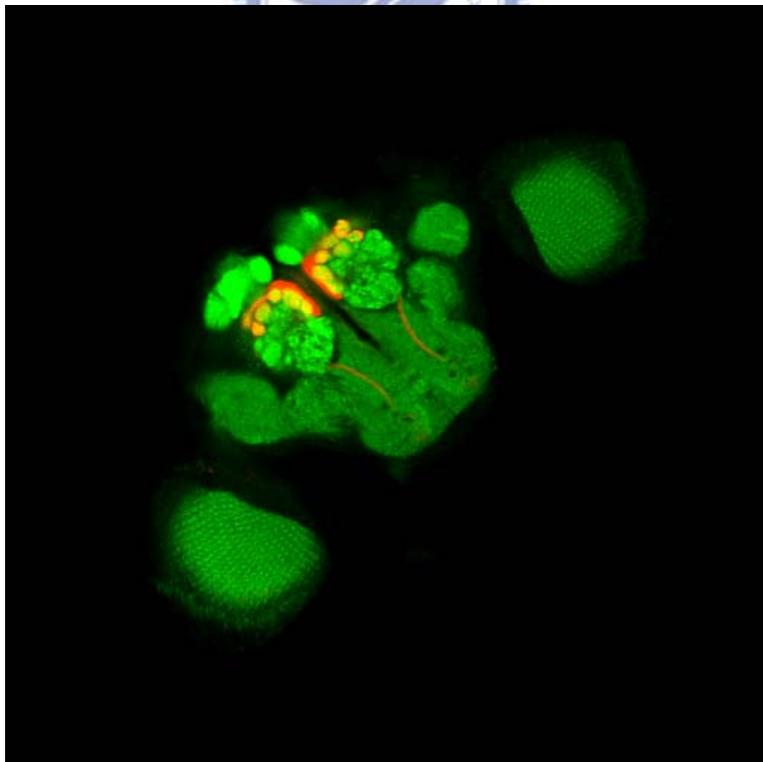
圖【4-1】



圖【4-2】

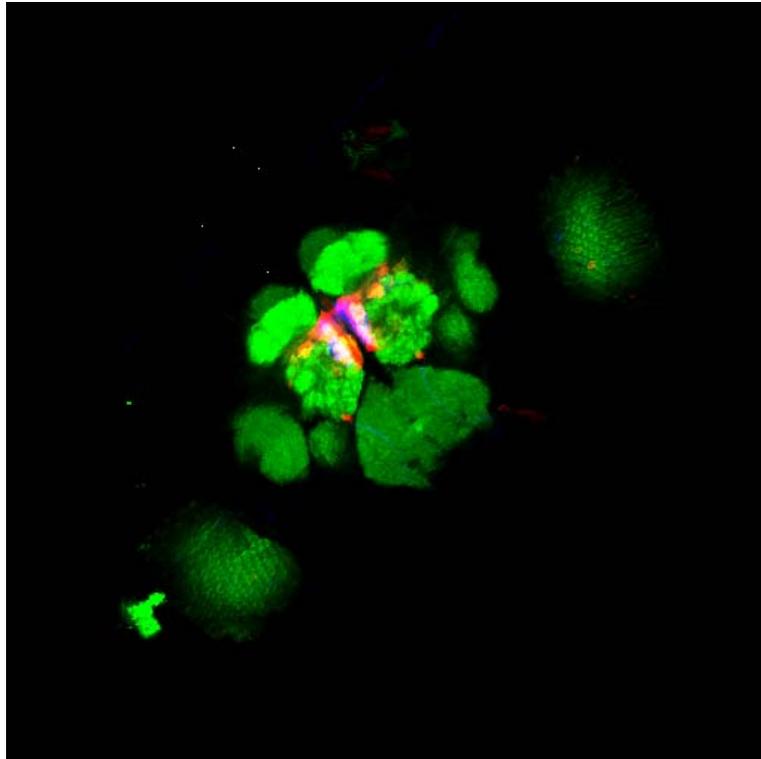


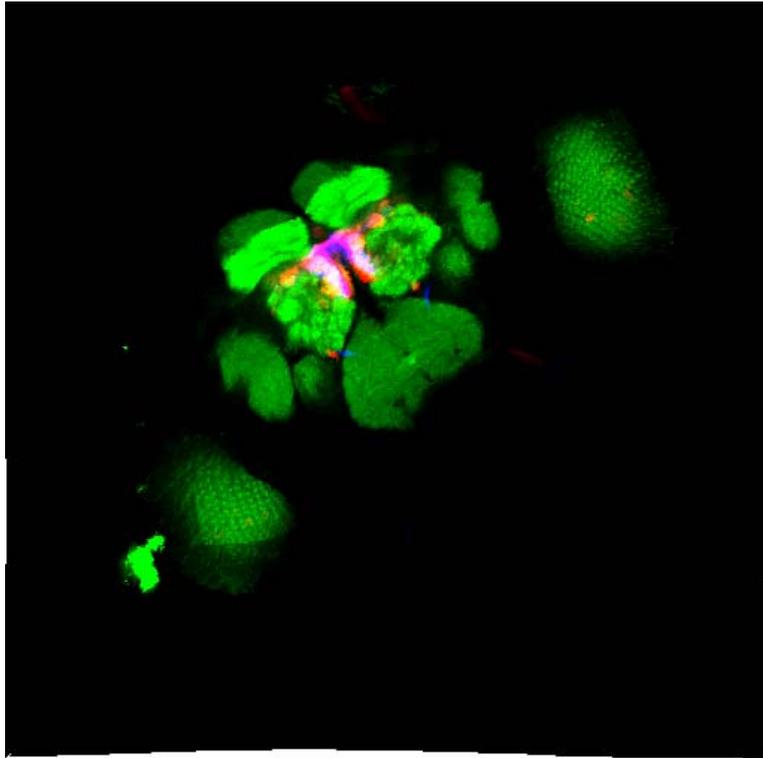
圖【4-3】



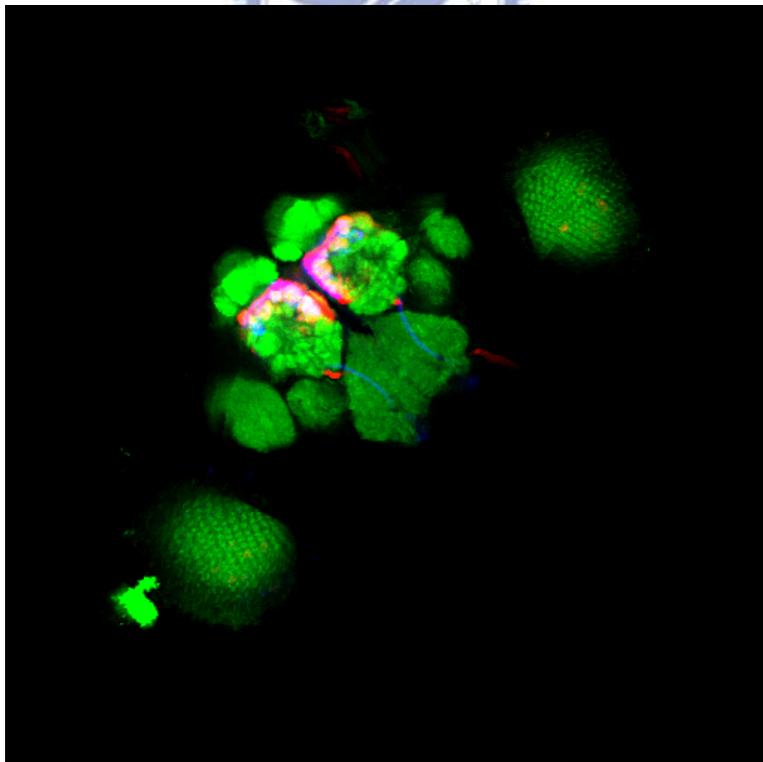
圖【4-4】

而接下的圖【4-5】到圖【4-7】為由 001 Morphing 到 002、003、004 的結果圖。
而【4-8】為 004 Morphing 到 001 的結果。

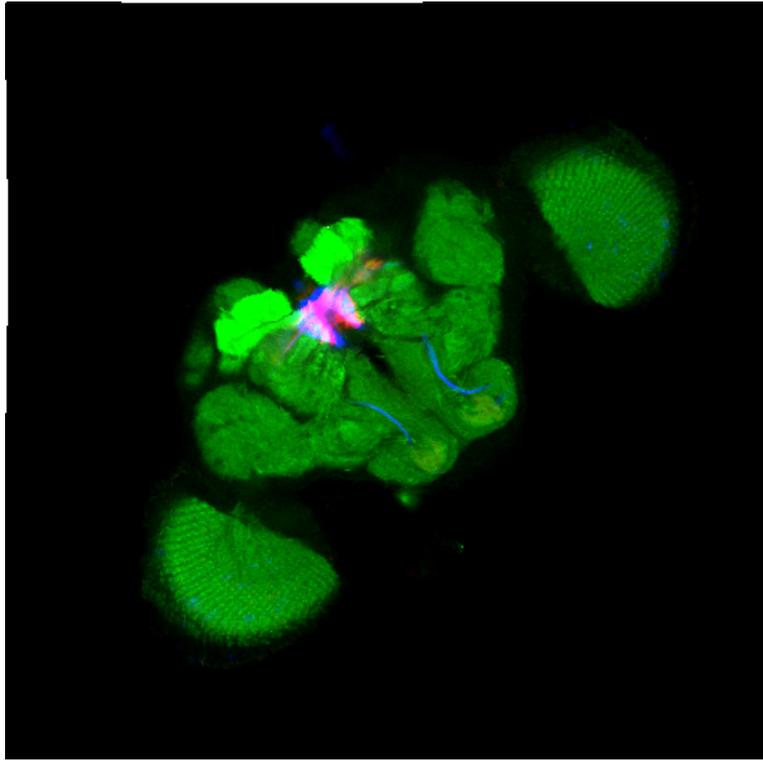




圖【4-6】

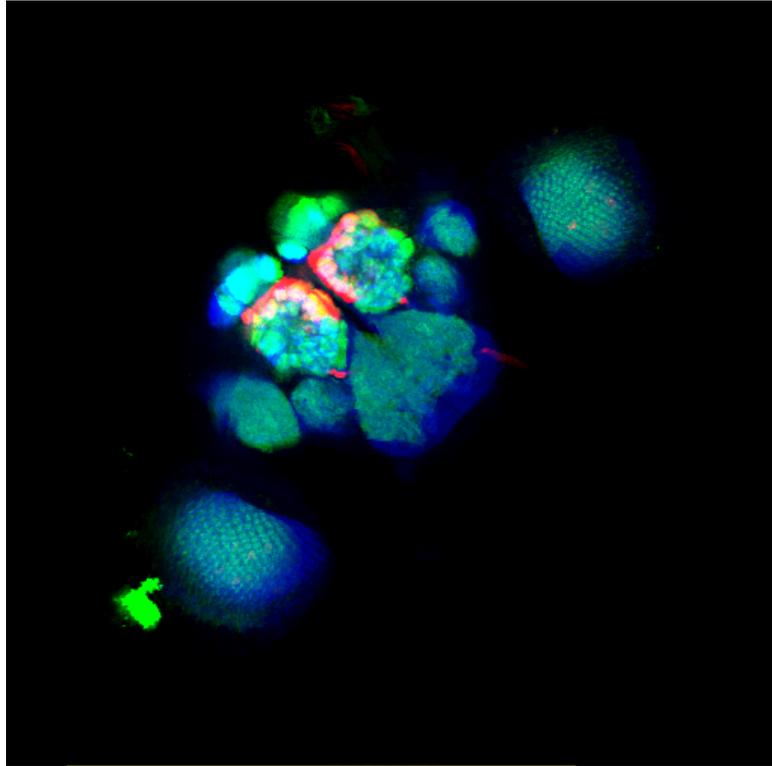


圖【4-7】



圖【4-8】

在這些結果裡面，紅色、綠色為 Source 的 2 個 channel，藍色部分為 Target 的神經 channel。在下圖【4-9】為 001 Morph 到 004 的圖，不過圖中的藍色部分為 004 的全腦 channel。



圖【4-9】

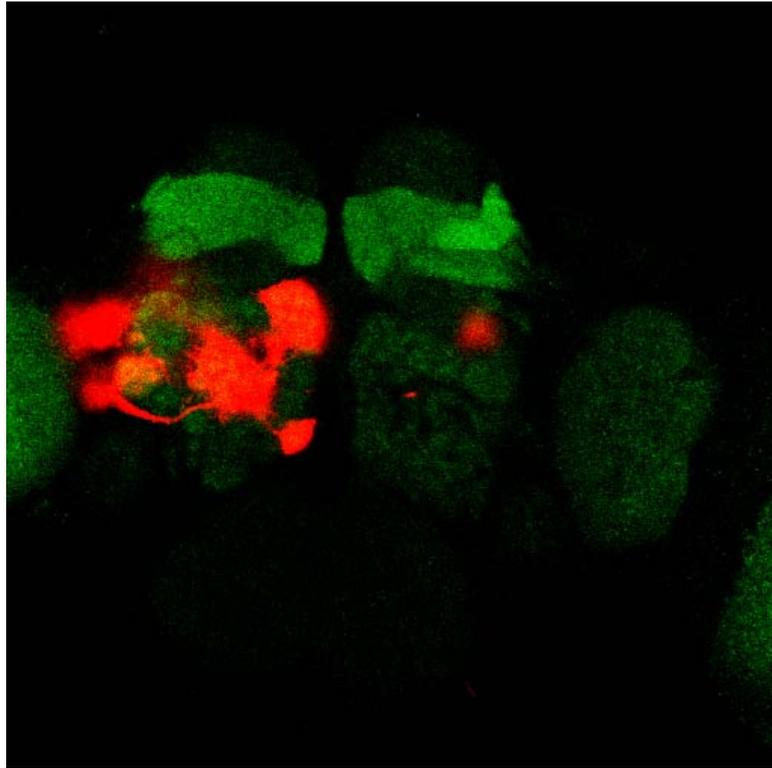
從上面的結果可以看出在 morphing 過後，原本神經的位置會因為特徵點的影響而改變位置，連同周遭附近的點也是，這樣一來可以清楚的觀察出神經是否在同一地方；此外，morphing 後的結果，整體來看輪廓上依然是維持住，而且看起來也相當連續。

下表為 Morphing 的時間

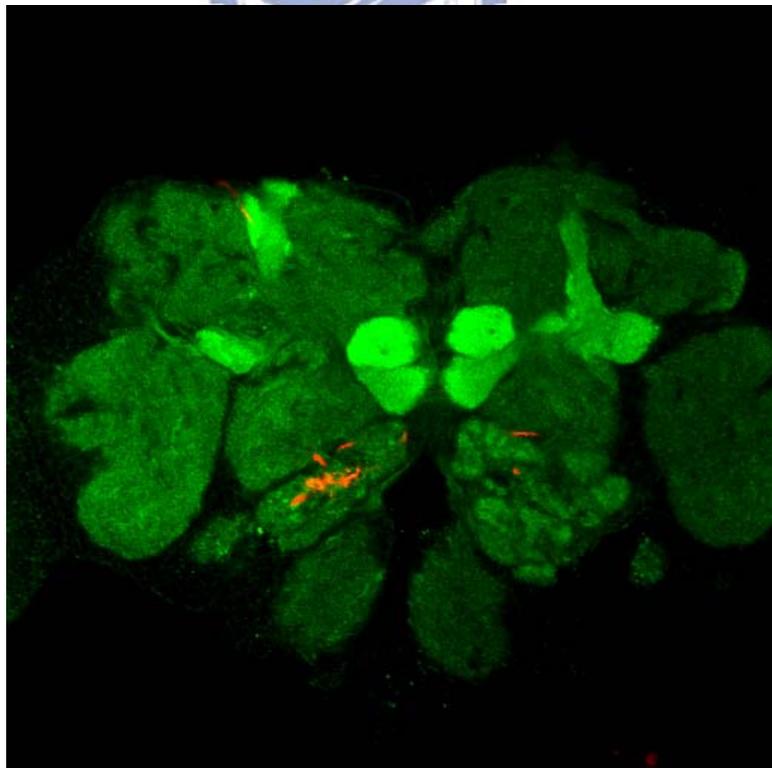
組別	時間 (單位：秒)
001 morph 到 002	3995
001 morph 到 003	5209
001 morph 004	4626
004 morph 001	5016

由上面的表格可以這個方法會花費大量的時間去做運算，這是因為必須去對 Target 上的每一個點去找到對應點，此外因為 Delaunay Triangulation 會產生較多的 Cell 數目，在找尋點在哪個 Cell 內部的計算上也會花去不少時間。所以花費的時間會跟這兩個因素有關。

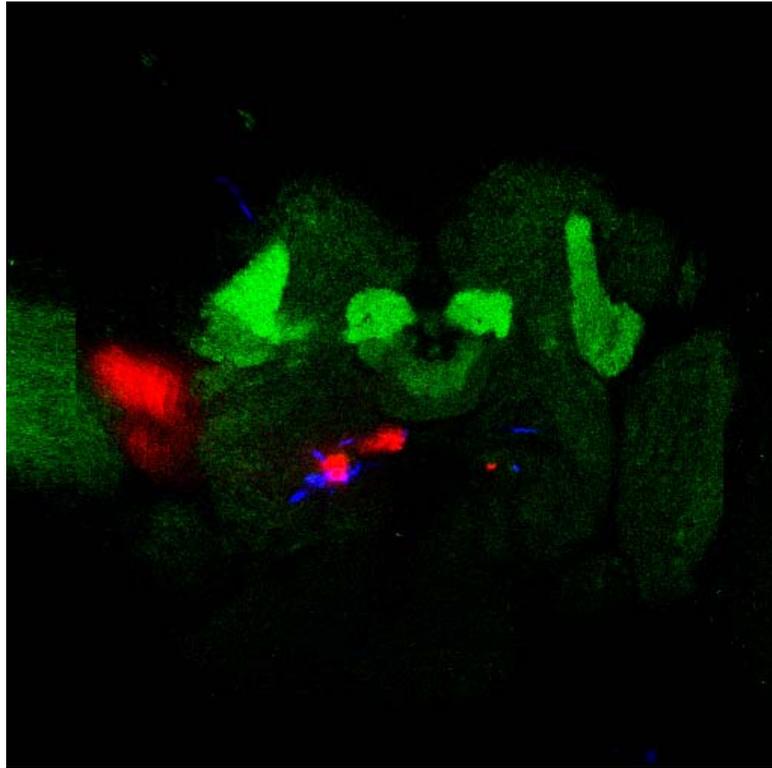
對於原本兩組 data 是歪斜，並且是上下顛倒的，Warp 之後也有不錯的結果。下面的圖【4-10】為 Source，圖【4-11】為 Target，圖【4-12】為 Result，在 Result 中的藍色部分是 Target 的神經位置，可以看出這組神經的位置在 Result 中都在應該在的位置上。這是因為在我們 Cell 的 Warp 裡面就有包括 translation、rotation 跟 scaling 的作用，可以把 data 反轉跟旋轉成跟 Target 同樣的模式，加上權重的影響，會把特徵點附近的點都拉在一起。



圖【4-10】

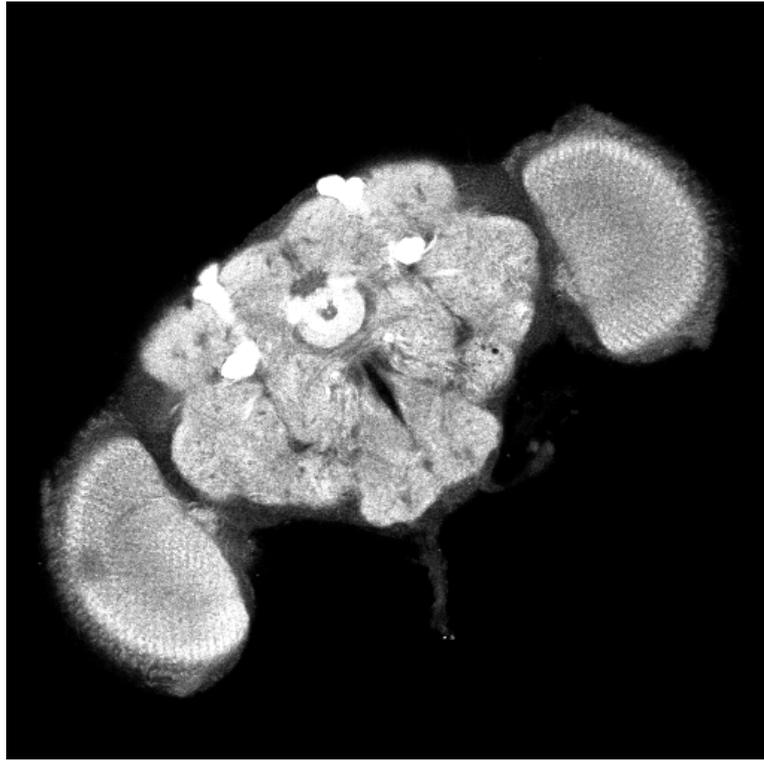


圖【4-11】

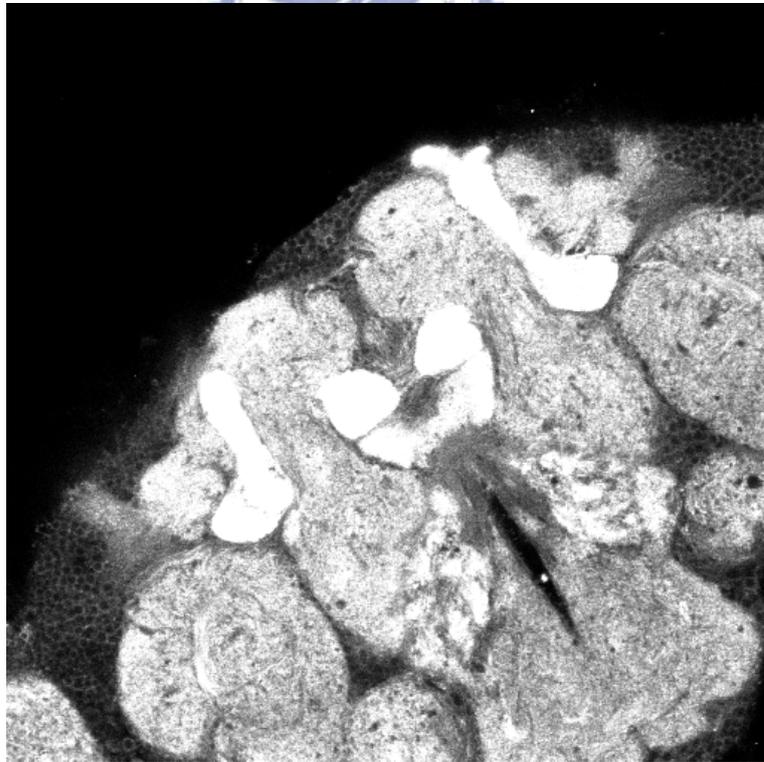


圖【4-12】

下面的圖【4-13】到【4-15】為部分腦跟全腦的 Morphing。圖【4-13】為 Target，大小為 512 x 512 x 71；圖【4-14】為 Source，大小為 512 x 512 x 115，圖【4-15】為 Morphing 後的結果。



圖【4-13】

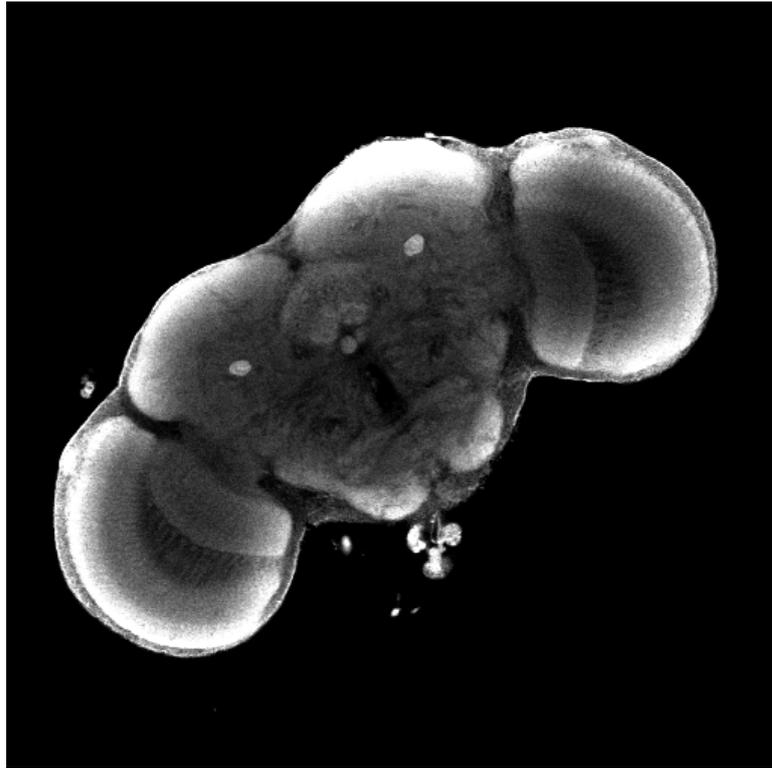


圖【4-14】

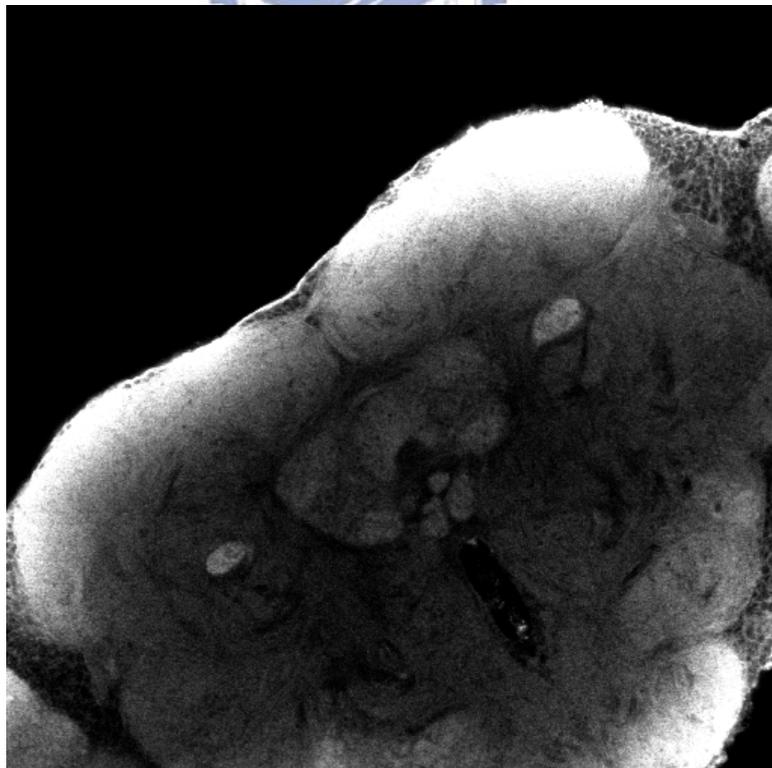


圖【4-15】

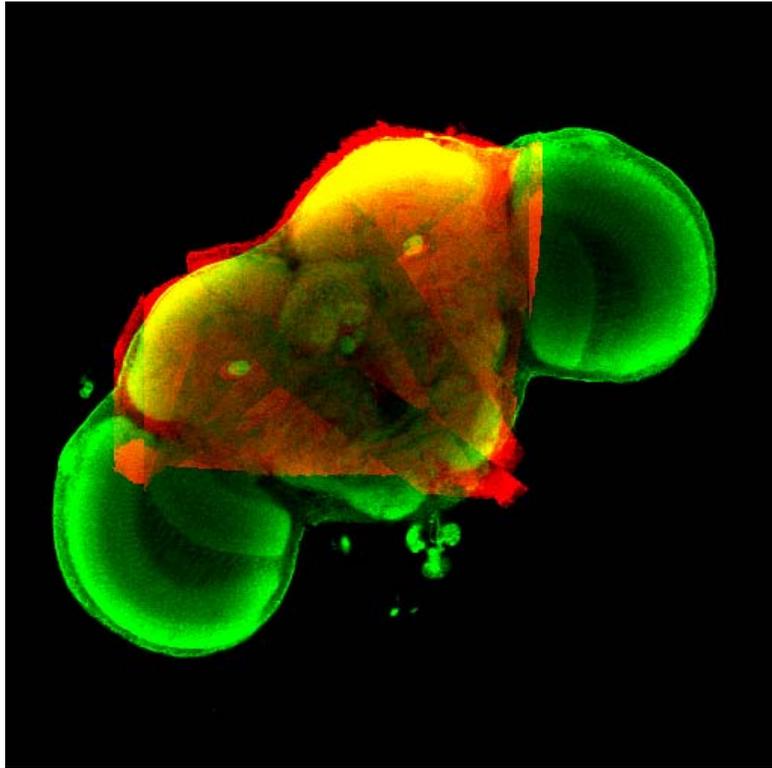
大致上部分腦 Morphing 過去，輪廓上上是沒有太大改變，但是在連續性有點被破壞，尤其是當部分腦跟全腦上掃描的張數比太大時，連續性會維持不住，這一點也許是因為在 down sampling 時，只對 X、Y 方向上作，而 Z 方向上沒有做，這樣有可能造成三角化的結果變成銳角；此外也因為當張數比太大時，精細度不同的關係，會讓我們特徵點的選取誤差放大，也許讓結果不連續。在下面的圖【4-16】為 Target，大小為 512 x 512 x 60；圖【4-17】為 Source，大小為 512 x 512 x 160，圖【4-18】為 Morphing 後的結果，紅色部分是 Morphing 的結果，綠色是 Target 的原來狀況。可以看出在張數比過大時，不連續的狀況會更嚴重。



圖【4-16】



圖【4-17】



第五章

未來展望

在我們提出的方法中，蠻依賴特徵點選取的正确性。當 Source 跟 Target 上對應出現錯誤時，也就是對錯的狀況發生，會讓整個結果失真。未來也許可以用好的 Registration 的演算法來從 Target 中選取一點，就可以找出在 Source 中最相似的點。此外在連續性上，在 Warp 步驟裡選取點的方法，也許可以改成用 RBF (radial basis function) 來實作，這樣可以在邊界上維持在 C^2 的連續性。這樣一來也可以讓結果看起來比較柔順。

此外，之後若是把 Warp 的結果跟 Volume Rendering 結合，就可以把 Warp 的結果以用 3D 的結構型式來觀察，更可以看出 Result 跟 Target 之間的差異，未來能建立資料庫管理，加上的網路的應用，可以讓使用者能線上上傳 Data 到伺服器，跟定義的標準腦來做 Warp，最後再以 Volume Rendering 展現結果，如此一來可以讓這個系統更加完善。

參考文獻

- [1] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In Computer Graphics, vol 26(2), pp 35–42, New York, NY, July 1992. Proceedings of SIGGRAPH 92.
- [2] Apostolos Lierios, Chase D. Garfinkle, and Marc Levoy. Feature-Based Volume Metamorphosis. In Computer Graphics, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 449-456, Proceedings of SIGGRAPH '95
- [3] Tim McInerney and Demetri Terzopoulos. Deformable Models in Medical Image Analysis: A Survey. in *Medical Image Analysis*, 1(2):91–108, 1996.
- [4] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics* 23, 3 (Aug.), 399-405, 2004.
- [5] Jun-yong Noh and Ulrich Neumann. Expression cloning. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, CA, 2001. 277~284
- [6] J. C. Carr, R. K. Beatson, J.B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. in Proceedings of ACM SIGGRAPH, Los Angeles, CA, 2001, pp. 67-76
- [7] Zhunping Zhang, Lifeng Wang, Baining Guo and Heung-Yeung Shum. Feature-Based Light Field Morphing. *ACM Trans on Graphics*, Vol. 21, No. 3,

2002 (Siggraph '02 Proceedings).

[8] Katherine M. Simonson, Steven M. Drescher and Franklin R. Tanner. A statistics-based approach to binary image registration with uncertainty analysis .IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 29 (1): 112-125 JAN 2007

[9] 蔡明倫，「三度空間腦部結構校準」，交通大學資訊科學所碩士論文，2002

