

基於虛擬實境之互動式 3D 醫學影像建模系統

*張財榮¹、陳冠因²

南臺科技大學資訊工程系

*trchang@stust.edu.tw

摘要

近年來由於虛擬實境 (virtual reality) 與 3D 列印的盛行，加上相關設備逐漸平價，近而促使相關應用普及。傳統的醫學影像是以 2D 方式呈現，但若能透過頭戴式立體顯示器 (head mounted display) 以虛擬實境立體顯示 (stereo display) 方式來呈現 3D 醫學影像會更加真實。本研究使用頭戴式立體顯示器 (HTC Vive) 配合體感控制器 (Vive Controller) 發展一套 3D 醫學影像互動系統。系統除了可以讓醫生可以直覺的方式觀察病患的患部，也可以讓使用者透過 3D 互動的方式圈選感興趣的組織部位以進行 3D 建模。建模時提供模型簡化演算法，在不影響模型品質下將模型的三角型數量減到最少，以提升顯示及計算效率。此外，也可將 3D 模型輸出成 STL 檔案格式，透過 3D 印表機輸出人體器官組織模型。本研究可以應用在醫學教學、手術模擬演練上，除了減少大體老師的需求也可以增進醫病信任關係。

關鍵詞：醫學影像、虛擬實境、模型簡化、3D 列印

3D Modeling of Medical Images Based on User Interactions in Virtual Reality

Tsai-Rong Chang, Guan-Yin Chen

Department of Computer Science and Information Engineering, Southern Taiwan University of Science and Technology

Abstract

The recent development of virtual reality and 3D printing technology, along with the ever-lowering costs of hardware, has popularized related applications. The conventional display of medical images is 2D-based. Yet, when it is upgraded to 3D-based one with devices of head-mounted-display and stereopsis aided by virtual reality, these images can be more vividly viewed by human subjects. In this study, we implemented a 3D-interactive system on medical images built upon the commercially available HTC Vive head-mounted-display, incorporating a kinesthetic-triggered Vive controller. This system could allow physicians to view more freely the bodily parts (both in healthy and diseased conditions), and to interactively outline the desired anatomical structures for 3D modeling. Optimization with simplified computation algorithms (minimizing the number of surface-constituent triangles) speeded up the 3D modeling procedure without jeopardizing the visual perceived quality of the model results. Moreover, the resultant 3D models of human anatomical parts had the output format of STL, which could be readily printed out as 3D objects. The present system we have developed can be applied widely to medical education and simulated surgery, easing the need for human corpses in the teaching of gross anatomy and enhancing patients' trust in physicians by facilitating their communication on surgical treatments.

Keywords: Medical Imaging, Virtual Reality, Model Simplification, 3D Printing

壹、前言

近年來由於虛擬實境 (virtual reality, VR)、擴增實境 (augmented reality, AR)、混合實境 (mixed reality, MR) 技術的成熟，圖學運算與呈現效能的提升，進而促使相關應用的逐漸普及。跨國投資銀行 Goldman Sachs 對 VR 及 AR 產業進行產值預估[1]，估計到 2025 年約 800 億美元，其規模將超過 TV 市場。另外根據研究諮詢公司 Industry ARC 的數據顯示，虛擬實境和擴增實境在醫療行業上的運用到 2020 年為止將達到 25.4 億美元。在這些相關的應用中以醫學領域最受到重視，其可以應用於醫學解剖教學、醫療決策、手術前規劃、患者教育及虛擬外科手術代替觀摩實習等領域。因此可以看出虛擬實境與擴增實境技術於醫學上的應用將是目前及未來極為重要的研究領域。目前市面上主要 VR 裝置有：HTC Vive (<https://www.vive.com/tw/>)、Meta QUEST 2 (<https://www.meta.com/tw/quest/products/quest-2/>)、Sony PlayStation VR (<https://www.playstation.com/zh-hant-tw/ps-vr/>)等。

VR 的特點在於提供沉浸式的體驗，因此對硬體配備的性能要求較高，成本也較高。VR 能給予媲美實際空間環境的體驗，因此在醫療方面的外科手術模擬、治療上的疼痛轉移 (pain distractions)，甚至心理諮詢時的畫面想像，VR 都是很好的應用工具，另外在交通工具的模擬駕駛、語言學習環境的建構，甚至是軍事訓練，VR 也都派得上用場。AR 則是在現實場景、環境中疊加入虛擬資訊和物件的技術，它的應用範圍比較廣泛，呈像的載具可以是手機、平板、眼鏡的鏡片，也可以是隱形眼鏡等任何具呈像功能的設備。由於 AR 呈像介面和設備的東西很多元，因此相關應用將會更豐富與普及。分析師預言，未來 AR 產值約為 VR 三到四倍的規模。VR 是從零打造出一個不存在的世界，AR 則是著重在與真實世界的結合，補強真實世界裡的資訊顯示與互動環境經驗。但真正具潛力的應該是 MR，因為它兼具 VR 與 AR 的特點，雖然目前其開發、應用的成熟度不如 VR 及 AR，但終究會是最後的贏家。MR 包含 AR 及 VR，將虛擬世界與真實世界混合在一起，使用者眼前所見到的環境同時包含有現實的物理實體，以及虛擬資訊同時呈現，目前市面上主要 MR 裝置有：Microsoft HoloLens 2 (<https://www.microsoft.com/zh-tw/hololens/>)、Magic Leap 2 (<https://www.magicleap.com/magic-leap-2/>)等。

而目前醫學影像技術的不斷進步，包括核磁共振攝影 (magnetic resonance, MR) 及多層螺旋 X 射線電腦斷層掃描 (multi-detector-row computed tomography, MDCT) 技術的發展成熟，使病患體內的結構可以高解析的序列影像來呈現。由於這些影像都是 2D 的並且數量龐大，醫師在查看患者身體內部狀況時都要一張張查看，不夠有效率。因此若能夠透過人機互動結合 3D 醫學影像立體顯示的方式，把拍攝的醫學影像以 3D 立體顯示 (stereo display)，醫師就可以很容易且快速的觀察病人的病因。透過 3D 醫學影像的真實顯示，也可以讓醫師透過虛擬實境技術模擬手術[2]，使用患者體內實際結構拍攝出來的序列影像建成的模型進行手術模擬。在這些相關應用中以醫學領域較為重要，其可以應用於醫學解剖教學、醫療決策、手術前規劃、患者教育及手術模擬等領域。在傳統的醫學影像中，影像是以二維 (two dimension, 2D) 呈現，但人體是以三維 (three dimensional, 3D) 空間組成。透過虛擬實境技術使醫學影像以高度擬真 3D 顯示，讓醫生可以更加理解人體構造與病患的患部。在教學方面，可以減少大體老師的需求更可以進行手術模擬，以減少手術失誤率，這是前所未有的突破。

另外傳統模型需要經過開模再製造，在這過程之中需要耗費不少時間與成本，如果成品需求量少會使得成本過高。但由於 3D 列印設備逐漸平價化，使得成品模型能客制化、成本也更低。3D 列印可以使用多種不同的原料列印，同時印出模型精密度高，使得應用更為廣泛。因此本研究希望透過以 VR 方式觀察 3D 醫學影像，將感興趣的組織圈選出來，再透過 3D 列印技術將組織模型列印出來，其可用於人體組織教學及說明用途。

本研究的分為兩個部分，涵蓋的範圍是從讀入醫學影像到模型建置與列印匯出，底下將闡述本研究的重點：

一、使用者互動介面

將讀入的多張 2D 醫學影像透過 3D 影像的渲染技術重構成 3D 醫學影像來協助使用者觀察影像，而

影像渲染方式分為兩種，分別為表面渲染（surface rendering）和立體渲染（volume rendering）。表面渲染只能用來顯示影像的表面資訊，需要儲存空間較少，速度較快。而立體渲染需要同時顯示物體的內外部的資訊，因此需要更多的儲存空間。在重構速度方面，由於硬體條件的限制，表面渲染比立體渲染快的多。但為了增進影像模型的真實性，本研究將同時採用表面渲染與立體渲染，立體渲染方面讓使用者可以選取出感興趣的組織，表面渲染則是下個部分會做介紹。

二、模型編修與匯出

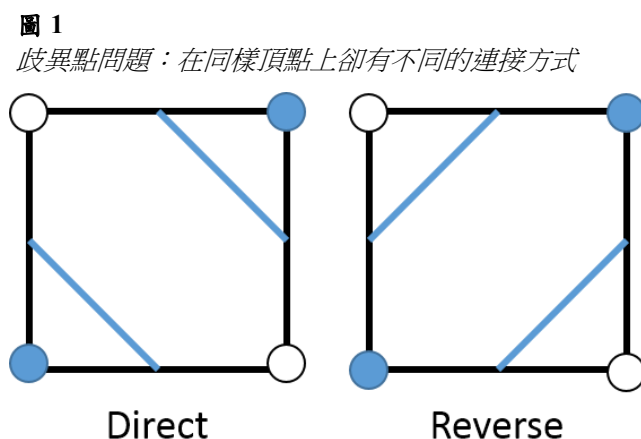
由上部分所述，將選取出的感興趣的組織透過表面渲染中的方塊演算法（marching cube）建立出 3D 立體模型，由於醫學影像本身為高解析度，所需的儲存空間較大，因此本系統將使用網格簡化（mesh simplify），在盡量不影響原始模型的型體的情況下，減少網格的數量，進而減少儲空間，並透過平滑的方式讓模型整體更平順、自然。在將處理過後的模型輸出成 STL 檔方便 3D 印表機輸出，做為醫學教學、手術模擬之用途。

貳、研究方法

本研究將發展一套互動式 3D 醫學影像顯示、建模與列印系統。首先使用 HTC Vive 及 Vive Controller 全套互動配備來研發 3D 醫學影像的體繪製（volume rendering）顯示與互動功能，讓使用者以 3D 互動的方式點選感興趣的組織部位，同時配合新增（+）與刪除（-）功能來選取感興趣組織的大部分 3D 部位。然後透過 3D 連通元件、膨脹、侵蝕進行感興趣組織 3D 區域生成，同時以組織染色方式於最大亮度投影法（maximum intensity projection）中呈現出來，讓使用者可以正確地切割出感興趣的組織部位。接著配合 3D 連通元件、膨脹、侵蝕依照影像像素的灰階值進行多視角人體組織輪廓擷取與建模編修，來區分人體各種不同的組織。最後使用表面繪製（surface rendering）建立出人體組織 3D 模型，並使用單色及彩色 3D 印表機印出等尺寸人體組織模型。因此本研究主要的研究方法將分為（1）3D 等值面提取演算法、（2）網格簡化演算法、（3）豪斯多夫距離、（4）3D 列印 STL 檔案格式，而各方向研究步驟詳述如下。

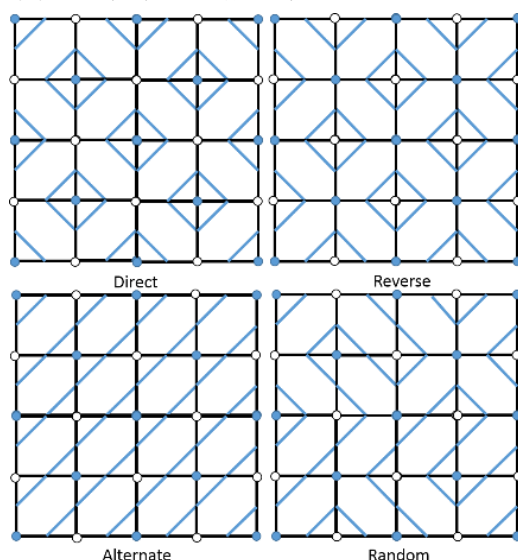
一、3D 等值面提取演算法（Dual Contouring）

傳統表面繪製演算法是使用行進方塊演算法，其計算方法雖然簡單，但是存在 2 個問題：（a）歧異點問題。（b）特徵點問題。[3]-[7]。以 2D 情況為例，歧異點問題在同樣頂點排列的情況下會有不同的連接方式，如圖 1 所示，假設藍色點代表物體內部，白色點代表外部，藍色線代表分割情形，就可以有如左圖的正面（direct）分割方式及右圖的反向（reverse）分割 2 種方式。



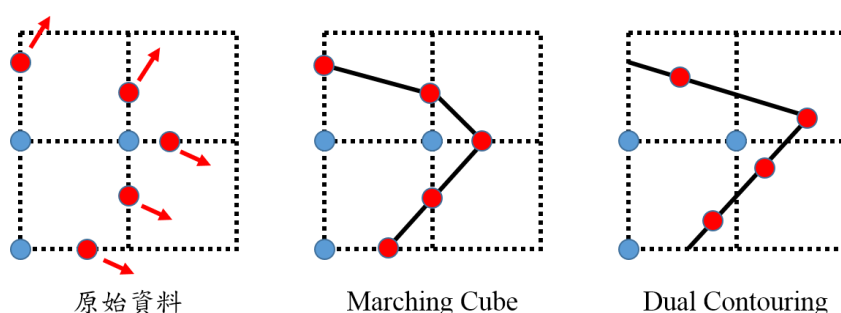
如圖 2 所示，對於圖 1 兩種連接方式，若有更多內部點（藍色）及外部點（白色）的相同排列，可能導致同個資料點會產生更多不一樣的結果，歧異點會導致模型的表面上產生漏洞。通常會發生歧異點的情況是在邏輯立方體上，該對角線的頂點如果同時大於（內部點）或小於（外部點）閾值的時候。

圖 2
歧異點所造成的可能結果



另一個問題就是特徵點問題，在 Marching Cubes 演算法中，組成三角形平面的頂點只會在取樣點的空間座標 X 軸或 Y 軸上。如果原始資料存在一些邊、角特徵的資料，使用 Marching Cubes 演算法的計算方式，可能會導致這些特徵點消失。而 Dual Contouring 演算法所計算出的交點座標與法向量，則可以使原本的特徵保留下來，如圖 3 所示。

圖 3
相同資料在不同演算法所建立出模型頂點分布情形



Dual Contouring 演算法與 Marching Cube 都是建構模型表面的演算法，但與 Marching Cube 相比，其使用交點與法向量來建構表面，可以補足上述所說的歧異點與特徵點的缺點。Dual Contouring 演算法先使用 Marching Cube 演算法計算出邏輯立方體輪廓上的交點與法向量，接著將交點與法向量代入二次誤差函數 $E(x)$ 找到最小值，該值為鄰近資料的交點，也就是構成網格平面的頂點座標，公式如下：

$$E(x) = \sum_i (n_i \cdot (x - p_i))^2 \quad (1)$$

其中 n_i 與 p_i 分別為邏輯立方體輪廓上的交點座標與法向量。 i 為邏輯立方體上的輪廓交點數量。接著將二次誤差函數改寫為矩陣形式如下：

$$E(x) = (Ax - b)^T (Ax - b) = x^T A^T A x - 2x^T A^T b + b^T b \quad (2)$$

其中矩陣 A 的列向量為交點的法向量 n_i ，向量 b 的每個元素為 $n_i \cdot p_i$ 。 $A^T A$ 為 3×3 矩陣， $A^T b$ 是一個大小為 3 的行向量， $b^T b$ 則是常數。將 $E(x)$ 透過正規表示式可以整理如下，其中 \hat{x} 為 x 的近似值：

$$A^T A \hat{x} = A^T b \quad (3)$$

但是[4]指出該方法會使數值不穩定，因此有學者提出使用 QR 矩陣分解來解決這個問題。使用 QR 矩陣分解計算出正交矩陣 Q 使得 Q 與[A b]相乘為上三角矩陣形式，表示如下：

$$\begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} x \\ x \\ x \\ 0 \\ \dots \end{pmatrix} = \begin{pmatrix} A' & b' \\ 0 & r \\ 0 & 0 \\ \dots & \dots \end{pmatrix} \quad (4)$$

其中A'為3×3的上三角矩陣，b'為大小為3的行向量，r為常數。誤差函數可以變化為：

$$\begin{aligned} (Ax - b)^T(Ax - b) &= (Ax - b)^T Q^T Q(Ax - b) \\ &= (QAx - Qb)^T(QAx - Qb) \\ &= (A'x - b')^T(A'x - b') + r^2 \end{aligned} \quad (5)$$

QR 分解[8]式子為：

$$\begin{aligned} A = QR &= [x_1 \quad x_2 \quad x_3] = [q_1 \quad q_2 \quad q_3] \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \\ &= [q_1 \quad q_2 \quad q_3] \begin{bmatrix} q_1^T x_1 & q_1^T x_2 & q_1^T x_3 \\ 0 & q_2^T x_2 & q_2^T x_3 \\ 0 & 0 & q_3^T x_3 \end{bmatrix} \end{aligned} \quad (6)$$

利用 $[q_1 \quad q_2 \quad q_3]$ 的正交規範性，組合權重 r_{ij} 即為 q_i 與 x_j 的內積， $r_{ij} = q_i^T x_j$ ，例如： $q_2^T x_3 = q_2^T(r_{13}q_1 + r_{23}q_2 + r_{33}q_3) = r_{23}q_2^T q_2 = r_{23}$ 。先透過 Gram-Schmidt 使用舊基底向量 x_i 算出新基底向量 y_i ， x_i 可寫為 y_1, \dots, y_i 的線性組合，如下：

$$\begin{aligned} x_1 &= y_1 \\ x_2 &= \frac{y_1^T x_2}{y_1^T y_1} y_1 + y_2 \\ x_i &= \frac{y_1^T x_i}{y_1^T y_1} y_1 + \frac{y_2^T x_i}{y_2^T y_2} y_2 \dots + \frac{y_{i-1}^T x_i}{y_{i-1}^T y_{i-1}} y_{i-1} + y_i \end{aligned} \quad (7)$$

根據上述公式計算出正交化 y_i 為：

$$y_i = x_i - \frac{y_1}{\|y_1\|^2} y_1^T x_i - \frac{y_2}{\|y_2\|^2} y_2^T x_i \dots - \frac{y_{i-1}}{\|y_{i-1}\|^2} y_{i-1}^T x_i \quad (8)$$

將正交化 y_i 單位化 q_i 則表示為：

$$q_i = \frac{y_i}{\|y_i\|} \quad (9)$$

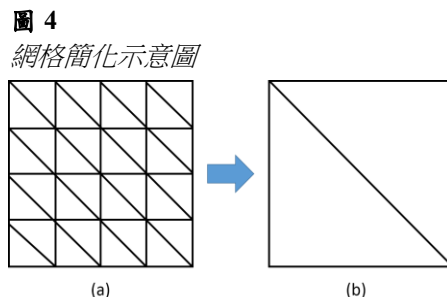
計算出 q_i 後，則可導出 Q 與 R 矩陣的值。經由上述介紹，Dual Contouring 計算步驟如下：

- (一) 建立 1 個 4×4 的零矩陣來存放 QR 分解的結果
- (二) 計算出每個體素邏輯立方體輪廓上的交點 p_i 與法向量 n_i 。
- (三) 將向量 $[x_{n_i}, y_{n_i}, z_{n_i}, \|p_i \cdot n_i\|]$ 放入 4×4 的零矩陣底部。
- (四) 通過 QR 分解得到大小 3×3 的上三角矩陣A'與向量b'。
- (五) 求線性方程式 $A'^T A'x = A'^T b' - A'^T b'c$ 的解，其中c是該體素的邏輯立方體中所有交點的重心位置。
- (六) 將求得的偏移量x加上重心位置c即為該體素的邏輯立方體中的頂點座標。
- (七) 如果計算得到的頂點座標在體素之外，那頂點座標用重心位置c來代替。
- (八) 找到每個相交的體素邊，將該周圍體素內的頂點連接產生 1 個四邊型平面。

二、網格簡化演算法 (Mesh Simplification)

使用 Marching Cube 所建構出來的模型大小會隨著影像大小張數遞增，組成該模型的三角形網格

(mesh) 數量會非常的龐大但不必要，大量的網格會給儲存、傳輸、編輯、渲染等待來困難，因此需要將網格進行有效的簡化。較有名的簡化方法有 Hoppe 和 QEM (quadric error metric) 兩種。如圖 4 所示，一個立方體的平面在 (a) 須使用 32 個三角形、96 個點 (三角形共有 3 個點，點會重複)，經過簡化後 (b) 只剩下 2 個三角形、6 個點，而且外型保持不變。網格簡化演算法的目的就是要減少網格的點與三角形的數量，經過簡化後模型外觀會接近原本的模型外觀。

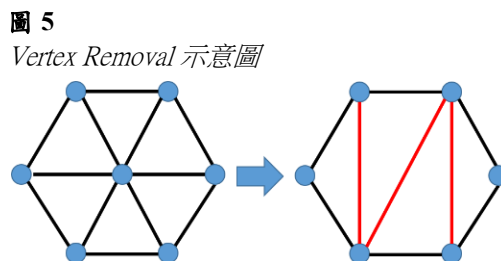


(一) 基礎簡化方法

基礎的簡化方法分為頂點移除 (vertex removal)、邊摺疊 (edge collapse)、半邊線摺疊 (half edge collapse) 三種[9]-[12]，簡述如下：

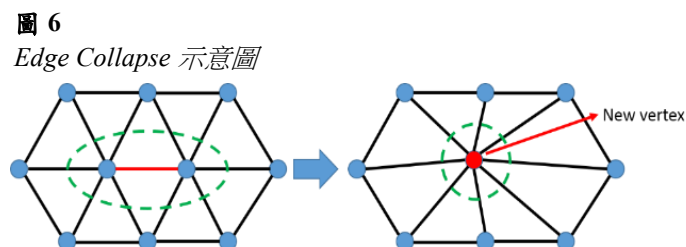
1. 頂點移除 (Vertex Removal)

如圖 5 所示，刪除任一頂點，再將刪除後的網格重新建構出新的邊及三角形。假設刪除前的頂點數為 k ，刪除後的頂點數變為 $k-1$ 。該網格簡化後的三角形由 6 個變成 4 個減少了 2 個，頂點由 7 個變成 6 個減少了 1 個，該方法會減少 1 個頂點並重新規劃原本與該頂點所連接的三角形。



2. 邊摺疊 (Edge Collapse)

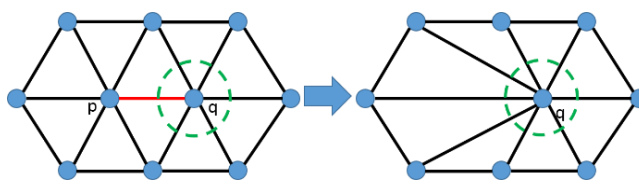
如圖 6 所示，將紅色標記的邊取其左右 2 個頂點的中心為新頂點。在將連接原始頂點的三角形點座標更新為新頂點。該網格簡化後的三角形由 10 個變成 8 個減少了 2 個，頂點由 10 個變成 9 個減少了 1 個，該方法會更新與原始頂點連接的三角形座標。



3. 半邊線摺疊 (Half Edge Collapse)

如圖 7 所示，選擇一個共邊的兩個頂點 p 、 q ，將 p 移動到 q 或 q 移動到 p ，則可將該邊合併為一頂點。原始連接 p 、 q 頂點的三角形改以合併後的頂點來取代。該網格簡化後的三角形由 10 個變成 8 個減少了 2 個，頂點由 10 個變成 9 個減少了 1 個，此方法並不會產生新的頂點。

圖 7
Half Edge Collapse 示意圖



但是使用該方法可能會遇到如圖 8 及圖 9 所示之問題，當頂點 p 往右與頂點 q 合併時，各三角形並無任何問題。但當頂點 q 往左與頂點 p 合併時會發生三角形重疊的現象，則須選擇另外兩點合併。

圖 8
頂點往右合併示意圖

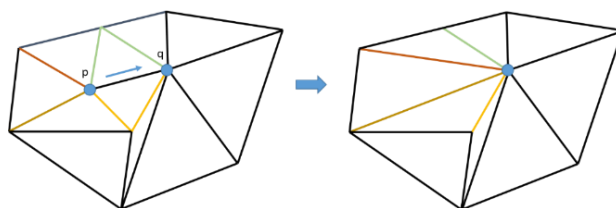
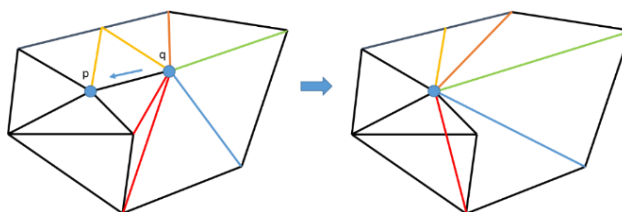


圖 9
頂點往左合併示意圖



上述三種方法的簡化流程大致皆分為以下步驟：

- (1) 記錄並整理點與周圍三角形的資料。
- (2) 決定簡化條件（取面積、法向量…等大小排序）。
- (3) 刪除最佳頂點並加入新頂點與更新三角形資料。
- (4) 持續步驟 3.直到三角形數量到達簡化目標。

(二)Quadric Error Metrics (QEM) 模型簡化演算法

QEM 模型簡化演算法是由 Michael Garland 所提出，有著計算速度快與可以保留原始模型特徵的優點，目前為止有許多研究都有用到這個演算法[12]–[17]。該模型簡化公式如下：

1. 計算出與該頂點連接的平面方程式，求出並記錄 a 、 b 、 c 、 d

$$ax + by + cz + d = 0 \text{ 且 } a^2 + b^2 + c^2 = 1$$

設三點座標 $p_1(x_1, y_1, z_1)$ 、 $p_2(x_2, y_2, z_2)$ 及 $p_3(x_3, y_3, z_3)$ ，平面法線與向量 $\overline{p_1p_2}$ 、 $\overline{p_1p_3}$ 垂直，此法向量 n 為：

$$\begin{aligned} \vec{n} = \overline{p_1p_2} \times \overline{p_1p_3} &= \begin{vmatrix} i & j & k \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} \\ &= ai + bj + ck = (a, b, c) \end{aligned} \quad (10)$$

$$\begin{aligned} a &= (y_2 - y_1) \times (z_3 - z_1) - (y_3 - y_1)(z_2 - z_1) \\ b &= (z_2 - z_1) \times (x_3 - x_1) - (z_3 - z_1)(x_2 - x_1) \\ c &= (x_2 - x_1) \times (y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \\ b &= -ax_1 - by_1 - cz_1 \end{aligned}$$

2. 計算每個平面的基本二次方誤差 (fundamental error quadric)，即 K_p ：

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (11)$$

$$p^T = [a, b, c, d]^T \quad (12)$$

3. 與該頂點連接的所有平面表示為：

$$\text{plane}(v) = (\text{plane}_1, \text{plane}_2, \dots, \text{plane}_n) \quad (13)$$

4. 將該頂點所有的基本二次方誤差 K_p 求和，即得到該點的基本二次方誤差之相加總合 Q ：

$$Q = \sum_{\text{plane}(v)} (K_p) = \sum_{\text{plane}(v)} (pp^T) \quad (14)$$

5. 簡化時須將兩個頂點 v_1 、 v_2 合併為 v_k 並計算合併後的基本二次方誤差之相加總合 Q ，即 Q_k ：

$$Q_k = Q_1 + Q_2 = \sum_{\text{plane}(v_1)} (K_{p1}) + \sum_{\text{plane}(v_2)} (K_{p2}) \quad (15)$$

$$v_k = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (16)$$

上述新座標 v_k 算式中的 q_{ij} 表示 Q_k 的第 i 列的第 j 個元素，如果 v_k 右式左方的矩陣不可翻轉則：

$$v_k = (1 - k)v_1 + kv_2, \text{ where } k \in [0,1] \quad (17)$$

上述 k 則是取0至1之間代入 $\text{Cost}(v_k)$ ，並取最小值即可。

6. 計算出 Q_k 的成本函數 $\text{Cost}(v_k)$ ：

$$\text{Cost}(v_k) = v_k^T Q_k v_k \quad (18)$$

綜上所述，QEM 整個計算流程如下：

1. 計算出網格上所有點的基本二次方誤差之相加總合 Q 。
2. 計算所有可能合併的點，並計算出 v_k 、 Q_k 。合併的條件為兩個頂點間有邊相連（如圖 10）或兩點間距離小於閾值 t （如圖 11）。

圖 10

合併點示意圖：兩個頂點間有邊相連

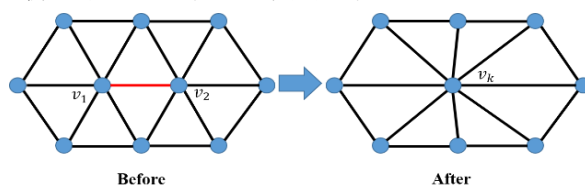
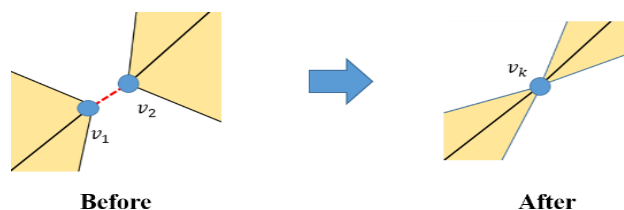


圖 11

合併點示意圖：兩點間距離小於閾值 t



3. 計算上述 v_k 的 $\text{Cost}(v_k)$ 並取最小值放入資料集中。
4. 排序並選出集中 $\text{Cost}(v_k)$ 最小的點 v_k ，再刪除該 v_k 所代表的 v_1 、 v_2 ，加入新座標 v_k 並更新集合內的資料。
5. 持續步驟4.直到三角形數量到達簡化目標。

三、豪斯多夫距離 (Hausdorff distance)

Hausdorff distance 又稱 Hausdorff metric 主要用於測量兩個 3D 模型之間的差異量，如果兩組集合上的每個點都與另一集合的點相近，那可說這兩組集合在 Hausdorff distance 是相近的[18]–[19]。該演算法為計算出兩個 3D 模型點集中一個模型點集合 X 中的任一點到另一個模型點集合 Y 中所有點，最後再從這些配對中找出最大值，公式如下：

$$d_H(X, Y) = \max\{\max_{x \in X} \min_{y \in Y} d(x, y), \max_{y \in Y} \min_{x \in X} d(x, y)\} \quad (19)$$

將它拆分為三個部分：

先找出點 x 在 Y 集中任一點的最小距離，如下：

$$d(x, Y) = \min_{y \in Y} d(x, y) \quad (20)$$

再由上述所示，找出 $d(x, Y)$ 中的最大值 $d(X, Y)$ ：

$$d(X, Y) = \max_{x \in X} d(x, Y) \quad (21)$$

最後整理公式的結果如下：

$$d_H(X, Y) = \max\{d(X, Y), d(Y, X)\} \quad (22)$$

舉例來說， $d_H(X, Y)$ 有一集合 X 其值為 {1,7}，另一集合 Y 其值為 {3,6}，欲計算出 $d_H(X, Y)$ 先代入公式 19，得知 $d(1, \{3,6\})=2$ ， $d(7, \{3,6\})=1$ ，再由公式 20、21，得知 $d(X, Y)=\max\{2,1\}=2$ 再由同一方法計算出 $d(Y, X)=\max\{2,1\}=2$ ，最後由公式 22 得到 $d_H(X, Y)$ 為 2。

四、3D 列印 STL 檔案格式

STL (STereoLithography) 是由 3D Systems 軟體公司創立、原本用於立體光刻電腦輔助設計軟體的檔案格式[20]。目前有許多軟體支援這種檔案格式，它被廣泛用於快速成型、3D 列印和電腦輔助製造 (CAM)。STL 檔案格式僅描述 3D 物體的表面幾何形狀，沒有顏色、材質貼圖或其它常見 3D 模型的屬性。STL 座標必須是正數，沒有尺度資訊，所以可以自定計量單位。

STL 檔案格式有文字和二進制兩種：文字格式的 STL 檔案是以文字方式儲存，該格式的缺點為生成的檔案容量較為龐大，但內容較二進制格式 STL 檔案直觀。文字格式 STL 檔案開頭以 solid 敘述當開頭，後面的 name 為檔案名稱。每一個三角形以 facet normal n_x 、 n_y 、 n_z 為開頭，以 endfacet 結尾。其中 n_x 、 n_y 、 n_z 是該三角形正規化後的法向量（使用右手定則），outer loop 記錄該三角形的三個頂點 v_1 、 v_2 、 v_3 的 3D 座標 $(v1_x, v1_y, v1_z)$ 、 $(v2_x, v2_y, v2_z)$ 及 $(v3_x, v3_y, v3_z)$ ，以 endloop 結尾，如圖 12 所示，其中 facet...endfacet 區段可以重複 n 次以記錄模型的所有三角形表面。而二進制格式 STL 檔案因簡潔且儲存容量較小因此比較常見，二進制 STL 檔用固定大小的位元組 (Bytes) 來定義三角形的幾何訊息，其三角形的法向量定義為右手法則。

圖 12

文字形式 STL 檔案格式

```
solid name

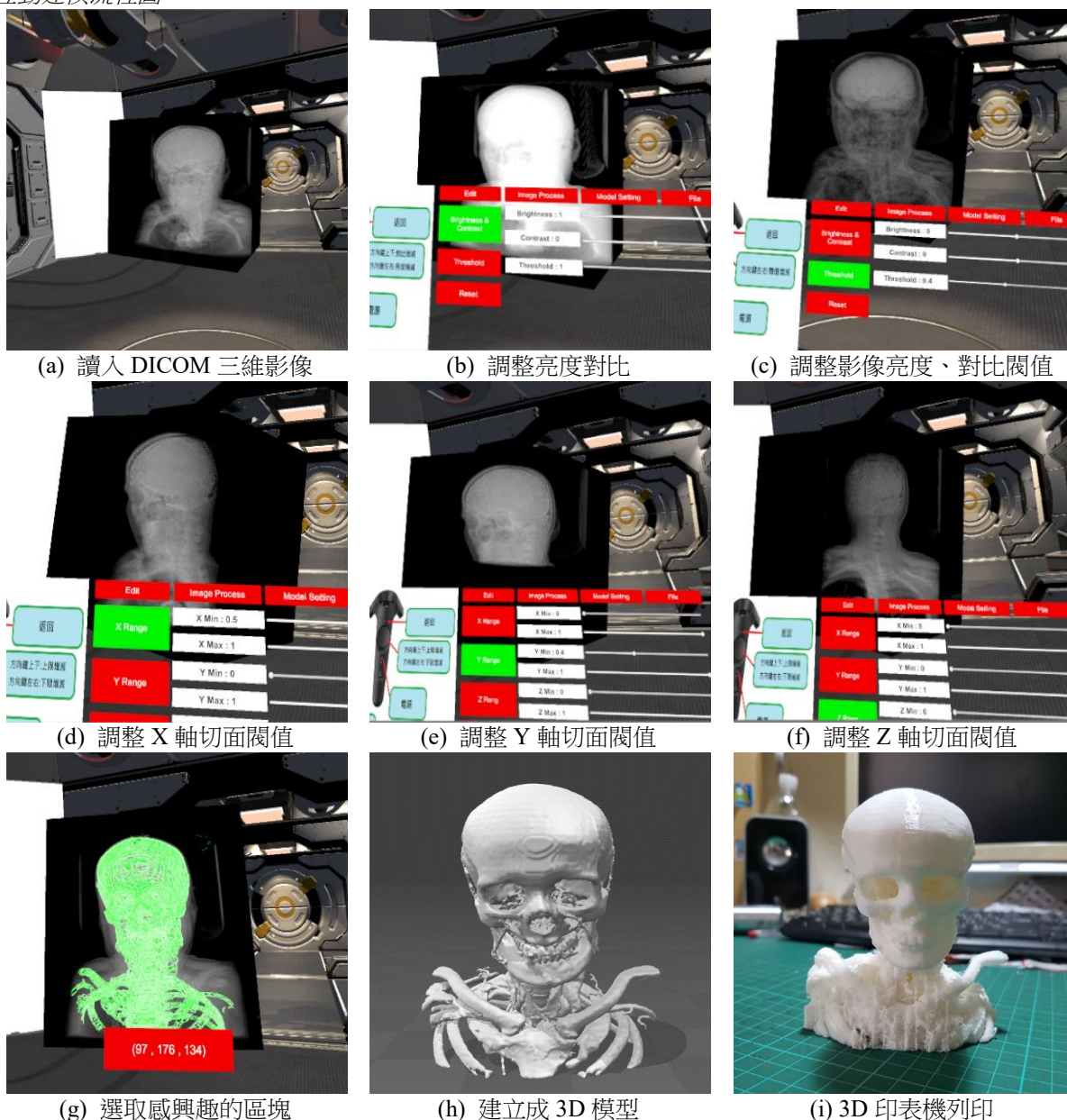
facet normal  $n_x$   $n_y$   $n_z$ 
  outer loop
    vertex  $v1_x$   $v1_y$   $v1_z$ 
    vertex  $v2_x$   $v2_y$   $v2_z$ 
    vertex  $v3_x$   $v3_y$   $v3_z$ 
  endloop
endfacet
endsolid name
```

五、互動建模流程

本研究使用 HTC Vive+Vive Controller 全套互動配備來研發 3D 醫學影像的體繪製 (volume rendering) 顯示與互動功能。首先將多張 DICOM 影像以體繪製配合虛擬實境技術顯示完整顯示出來，讓使用者以 3D 互動的方式點選感興趣的組織部位，同時配合新增 (+) 與刪除 (-) 功能來選取感興趣組織的大部分 3D 部位，接著透過 Marching Cube 演算法構出 3D 模型，同時提供使用者進行 3D 模型編修，最後將 3D 模型依照原始實際體素大小儲存成 3D 印表機通用的 STL 模型格式，圖 13 為一個互動建模流程範例。

圖 13

互動建模流程圖

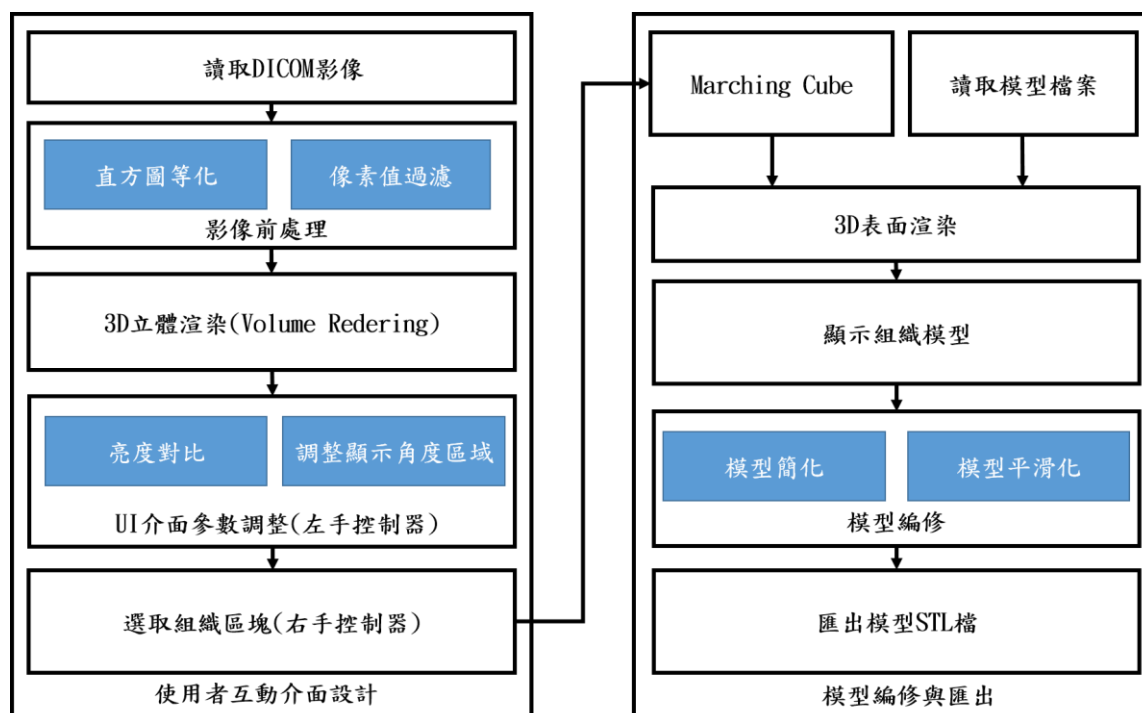


參、系統架構

一、系統概觀

本研究的互動式 3D 醫學影像建模系統主要分為兩個部分，分別為使用者互動介面設計與模型編修與匯出，如圖 14 所示。使用者互動介面部分主要目的是讓使用者讀取 DICOM 檔並建立起 3D 立體渲染影像，再透過 HTC Vive 以虛擬實境顯示出來，來增加真實感與人機互動。由於原始的 DICOM 影像可

圖 14
系統架構圖



能由 12 位元壓縮到 8 位元顯示辨識性較差，所以須透過影像直方圖等化與過濾掉底色（純黑色）的部分等影像前處理步驟增加可視性。而使用者（User Interface, UI）介面提供了參數調整區塊可以調整 3D 影像的亮度值與對比度，使某些特徵更明顯的顯示出來，方便使用者觀察與選取感興趣的組織，也可調整影像顯示的區間，例如原始影像為病患的全身圖，也可以設定僅顯示頭部區域。使用者調整參數後可以透過 Vive 的控制器伸入 3D 影像內作選取的動作，以便選取感興趣的區塊。模型編修與匯出的部分可以透過 Marching Cube 演算法將使用使用者互動介面所選取的組織區塊資料建立出 3D 模型或讀取 3D 檔案。模型編修則可以將 3D 模型進一步的簡化或平滑化，使模型的三角形數量變少或外觀更平順。最後將 3D 模型匯出為 STL 檔案格式，由於模型的三角形數量會與檔案大小成正比，所以簡化後的模型能使檔案較小與讀取較快。

二、使用者互動介面

原始的 DICOM 影像由 12 位元壓縮到 8 位元顯示，因此可讀性可能較差，須先經過影像前處理，例如直方圖等化、過濾純黑的像素值...等處理，讓 3D 影像顯示時能清楚看到物體樣貌。直方圖等化是用來增強影像對比度的，使影像的像素分布更均勻。該方法常用於背景與前景皆太亮或太暗的情況，例如：X 光影像，在等化後可以更清楚的看到人體的骨骼情況。直方圖等化的計算公式如下：

$$h(v) = \left(\sum_{i=\min}^v k(i) - k(\min) \right) \times \frac{L - 1}{M \times N - (k(\min))} \quad (23)$$

上述 $h(v)$ 為等化後的灰階值， $k(i)$ 為灰階值為 i 像素個數的統計值， \min 為直方圖中出現的最小灰階值， M 與 N 為影像的寬與高， L 為該影像的灰階級數，在常見的影像中 8 位元有 $2^8 = 256$ 階。

舉例來說，有一張圖的寬高為 5×5 ，影像灰階級數為 $2^3 = 8$ 階，像素值的統計表如表 1 所示。原始像素值為 3 經過直方圖等化後像素值變化如表 2 所示。

$$\begin{aligned} h(3) &= \left(\sum_{i=2}^3 k(i) - 3 \right) \times \frac{8 - 1}{5 \times 5 - 3} \\ &= (3 + 7 - 3) \times \frac{7}{22} \cong 2 \end{aligned} \quad (24)$$

表 1
直方圖統計範例

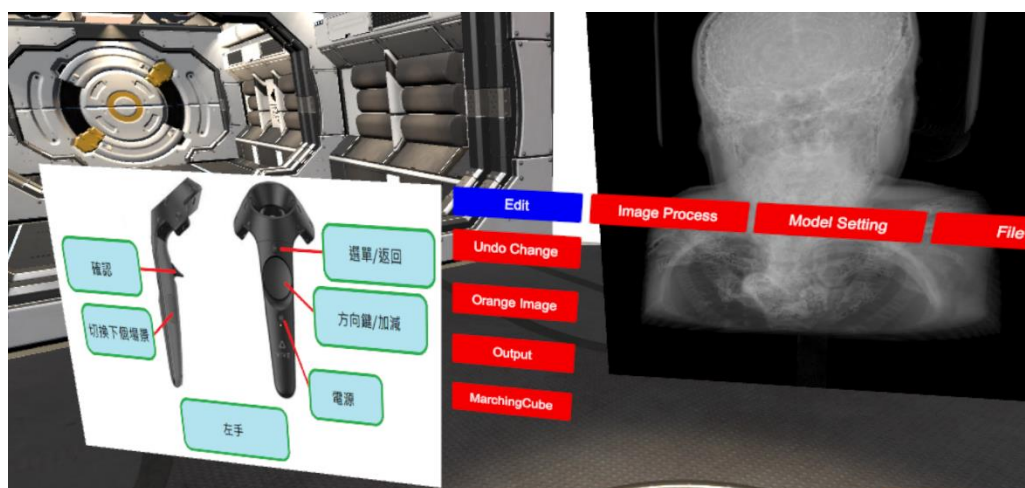
像素值	出現次數	像素值	出現次數
0	0	4	5
1	0	5	10
2	3	6	0
3	7	7	0

表 2
像素值等化前後對照

原始像素值	等化後像素值	原始像素值	等化後像素值
0	0	4	3.8 \cong 4
1	0	5	7
2	0	6	7
3	2	7	7

本系統使用 Unity (<https://unity.com/>)開發工具，將前處理後的 3D 影像陣列 color[,]寫入到一個 3D Texture 物件中，使用 Unity 中的 shader 物件將該 3D Texture 渲染在一個長方體上。使用者介面 (UI) 提供參數調整功能如圖 15 所示，本系統預設使用右手控制器來選擇感興趣的組織範圍，所以將 UI 依附在左手控制器上，同時加上操作說明。在對影像資料進行運算 (改變亮度對比或顯示區域) 後，將結果寫回 3D Texture 中，但使用該方法會因為計算量過大造成停頓，降低使用者操作的流暢度。而解決上述問題的方法為將影像處理程是改寫到 shader 程式中，使用 GPU 進行 shading 計算時先進行亮度對比及顯示區域上色後在進行 Ray Casting 計算，使用該方法能增加程式的流暢度，讓使用者調整影像處理參數時不會產生的停頓現象。

圖 15
使用者介面 UI 提供參數調整功能



如圖 16、圖 17 所示，本系統可以透過 UI 介面調整亮度與對比，而亮度與對比公式為：

$$\text{PixelValue} = \left(\frac{1.1 \times (C + 1)}{1 \times (1 - C)} \right) \times (\text{PixelValue} - 0.5) + 0.5 + B \quad (25)$$

其中 PixelValue 為該點像素值，介於 0 到 1 之間，以最常見的色階 256 色來說，須將像素值除 255 使值介於 0 到 1 之間。而 C 為對比度，介於 -1 到 1 之間。B 為亮度值，介於 -1 到 1 之間。

圖 16
調整亮度值的執行畫面

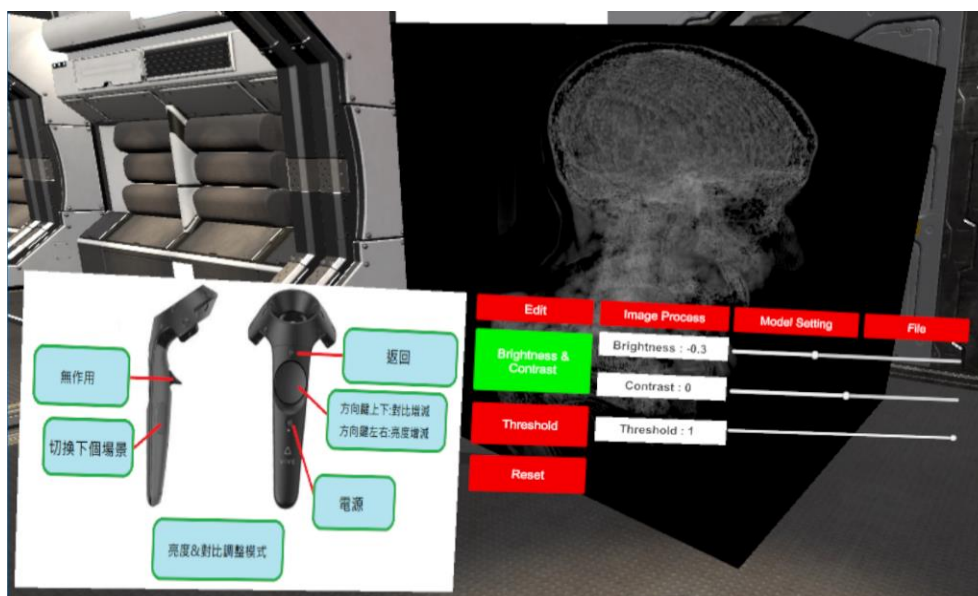
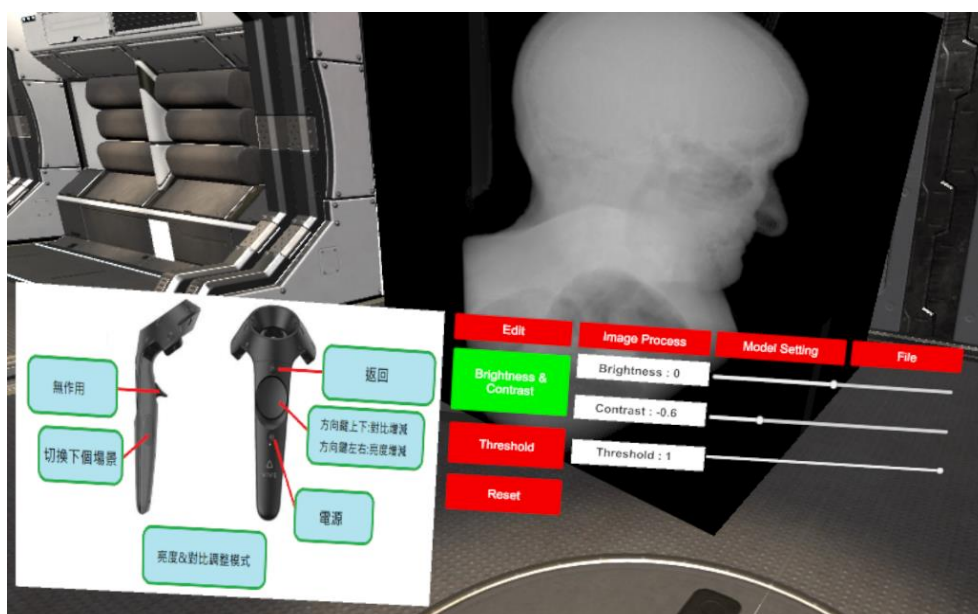
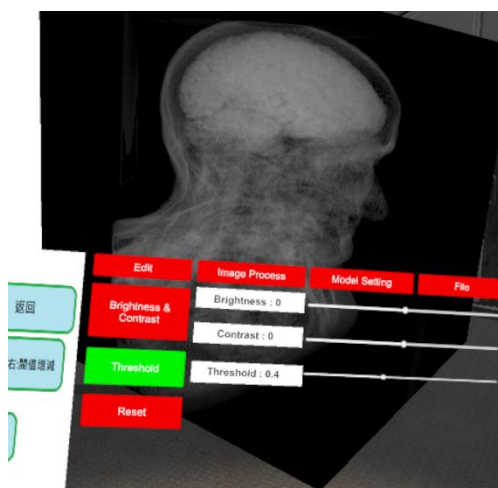


圖 17
調整對比度的執行畫面

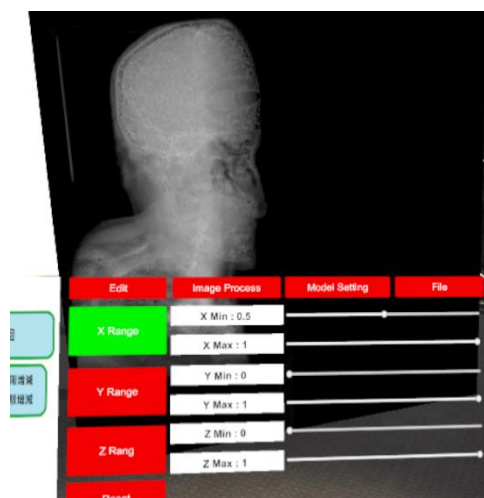


如圖 18 所示，(a) 為調整閾值 (Threshold) 範圍為 0.4 的執行畫面 (可調整範圍為 0 到 1 之間)，調整閾值數值可以過濾要顯示灰階值，像素灰階值需小於閾值才可顯示出來。另外 (b) (c) (d) 為調整 X、Y、Z 軸的顯示範圍的執行畫面，可以調整 X、Y、Z 軸的顯示範圍來顯示特定範圍的影像，任一軸範圍皆介於 0 到 1 之間。

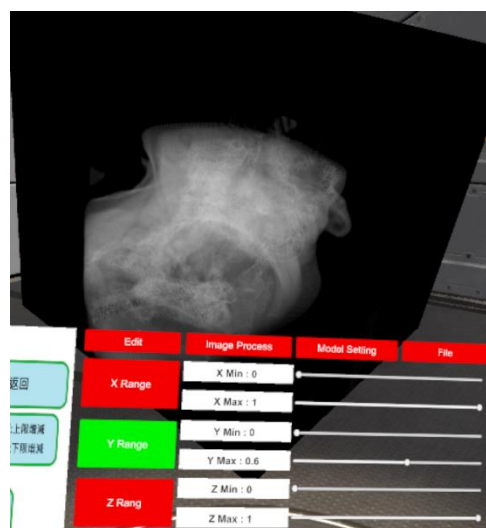
圖 18
調整各參數的結果



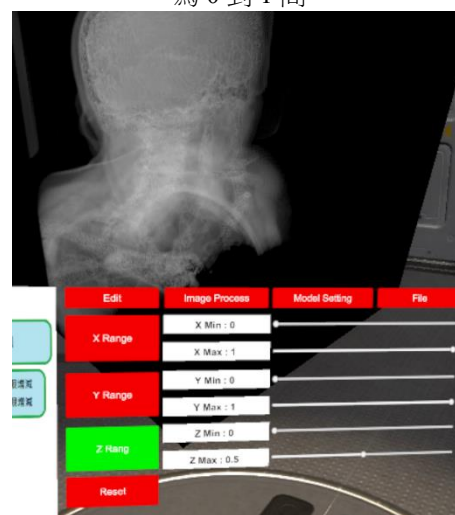
(a) 為調整閾值到 0.4 之結果



(b) 為 X 軸範圍介於 0.5 到 1 間，Y、Z 軸皆為 0 到 1 間



(c) 為 Y 軸範圍介於 0 到 0.6 間，X、Z 軸皆為 0 到 1 間



(d) 為 Z 軸範圍介於 0.5 到 1 間，X、Y 軸皆為 0 到 1 間。

如果使用者欲選取感興趣的區塊，可以將右手控制器伸入 3D 影像中直接按壓控制器的板機，程式會以板機壓下時控制器最前端為基準座標，轉換到 3D 影像內的相對應的點座標，藉由點座標可知所選取在 3D 影像陣列 $color[,]$ 中的像素位置，對該像素點進行 3D 像素連通演算法，要實作像素連通演算法須建立一個與原影像資料同大小的 3D 陣列，該陣列的用途用於標記，讓演算法得知該位置已經過計算，不需去尋找相鄰的像素。如圖 19、圖 20 所示，該演算法的步驟大概如下：

- (一) 建立一個 CheckNeighbor 方法來檢查原始影像特定座標點是否有標記過、是不是選取區塊的鄰居、是否將 N 連通的鄰居加入等待檢查的資料集中。如果沒有標記過就會檢查該座標是否跟選取區塊的顏色一樣或相近，並標記該座標與依照 N 連通將鄰居加入等待檢查的資料集中。
- (二) 透過一個迴圈不斷檢查待處理資料，直到等待檢查的資料集中無資料。

圖 19

像素連通演算法概述

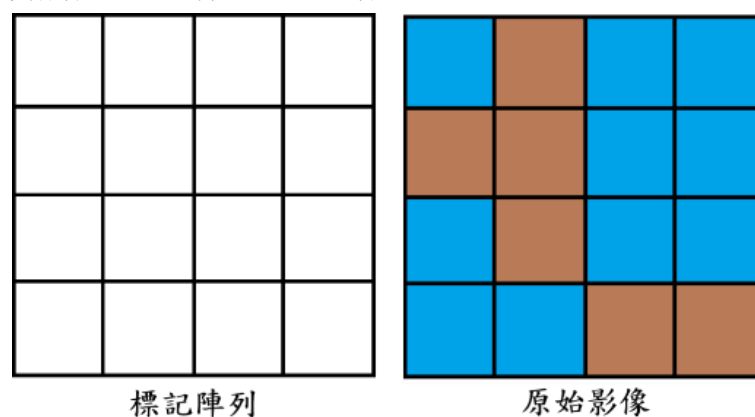
```

Color[,] OrgImage;//原始影像
Bool[,] Markdata;//標記
Color[,] resultImage;//結果影像
Color targetColor;//要找相鄰區塊的顏色值
Color MarkColor;//標記顏色
List<Vector3> list;//待處理資料
Void CheckNeighbor(Vector3 pos)
{
    if(Markdata[pos.x,pos.y,pos.z])//如果Markdata[pos.x,pos.y,pos.z]沒標記過
    {
        if(OrgImage[pos.x,pos.y,pos.z] ==targetColor)
        //如果OrgImage[pos.x,pos.y,pos.z]的顏色等於targetColor
        {
            resultImage[pos.x,pos.y,pos.z]= MarkColor;
            Markdata[pos.x,pos.y,pos.z]=true;//標記
            //依照N-連通將該座標加入list
            //ex. list.add(new Vector3(pos.x+1,pos.y+1,pos.z+1));
        }
    }
}
While(list>0)//一直檢查，直到list清空
{
    CheckNeighbor(list[0]);
    list. RemoveAt(0);
}

```

圖 20

實作像素連通演算法須建立的標記陣列示意圖



三、模型簡化及 3D 列印 STL 檔輸出

在 Unity 中加入 3D 模型有兩種方法，分別是 (a) .將上節所選取的資料透過 Dual Marching Cube 計算多個三角形資料，或是 (b) .將現有的記錄著多個三角形資料的檔案讀入 Unity 中。這裡要注意的是，Unity 中儲存三角形的網格 (mesh) 物件所能儲存的三角形數量上限為 65,534，如果超出該上限需要在另外建立多個 mesh 做儲存。另外，如果要將模型顯示出來，需要注意組成模型表面三角形的法向量。在 Unity 中是以左手法則的法向量來表示三角形正面，但 STL 檔案是以右手法則，在讀取檔案時需要注意否則會看不到模型。

在 Unity 上顯示模型時，如果該模型的三角型數量超過 65,534 個，Unity 會將該模型分割為多個 Mesh 物件。本系統是將各個 Mesh 物件做 QEM 簡化。如圖 21 所示，模型平滑化的功能為先取得各點鄰邊點

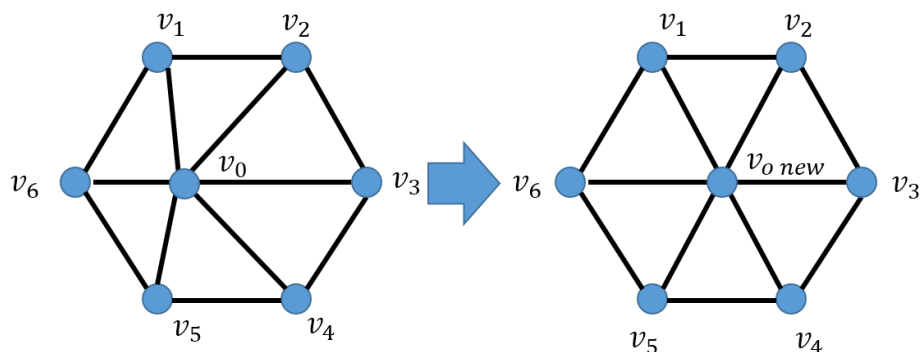
的資訊，再取平均為新點座標，其公式如下：

$$v_{new} = \frac{\sum_{i=0}^n v_i}{n+1} \quad (26)$$

其中 v_{new} 為平滑後的頂點座標， n 為舊頂點的鄰邊點數量， v_i 為鄰邊點的座標點。

圖 21

平滑化示意圖



將模型依使用者需求調整為適當的狀態後，可以將該模型的三角形座標點存成 STL 檔，再透過 3D 印表機印出 3D 模型，以應用在展示、教學...等用途。

肆、實驗結果

一、系統開發與環境設置

本系統所採用的設備及開發環境：

CPU：Intel Core(TM) i7-6700

RAM：16GB

顯示卡(GPU)：NVIDIA GEFORCE GTX 1070

作業系統：Windows 10 64 位元

開發環境：Unity 5.6、Visual Studio 2017

開發語言：C#

二、簡化結果之探討與比較

目前美國、中國、韓國皆有一個完整人體影像的專案，假如以男性影像資料作比較：Visible Human Project (VHP) 的切片影像尺寸為 2048×1216，電腦斷層 (CT) 影像尺寸為 512×512，核磁共振 (MRI) 影像尺寸為 256×256。Chinese Visible Human (CVH) 的切片影像尺寸為 3072×2048，電腦斷層影像尺寸為 512×512，核磁共振影像尺寸為 512×512。Visible Korean Human (VKH) 的切片影像尺寸為 3040×2008，電腦斷層影像尺寸為 505×276，核磁共振影像尺寸為 505×276。本實驗的資料取自 Visible Korean Human[21] 所提供的男性影像資料，影像內容如圖 22 所示分為切片影像、電腦斷層影像、核磁共振影像、以及使用不同顏色來標記各個器官區塊影像。本實驗針對四個器官，分別為：皮膚、心臟、肺部、腦部進行 3D 重構及簡化，詳細部位將在下方個別說明。

由於原始影像尺寸較大，重構出來的 3D 模型三角形數量太過龐大，因此實驗時先將原始影像等比縮小至 256×256，同時原始影像資料的間隔為 5 取樣，所以可以使計算更為快速。雖然如此，但建構出的模型三角形數量仍然很大，為了加速 3D 顯示速度，需要對模型進行簡化，但如何在模型精細程度與三角形數量間求得一個平衡點是很重要的。本研究希望於透過計算出最佳的簡化情形，為了減少逐步簡化所耗費的大量計算時間，最佳的定義為在原始與簡化的模型間，找出兩者模型間的點座標差距不大，並且盡量減少三角形的數量的狀況，演算法流程為：

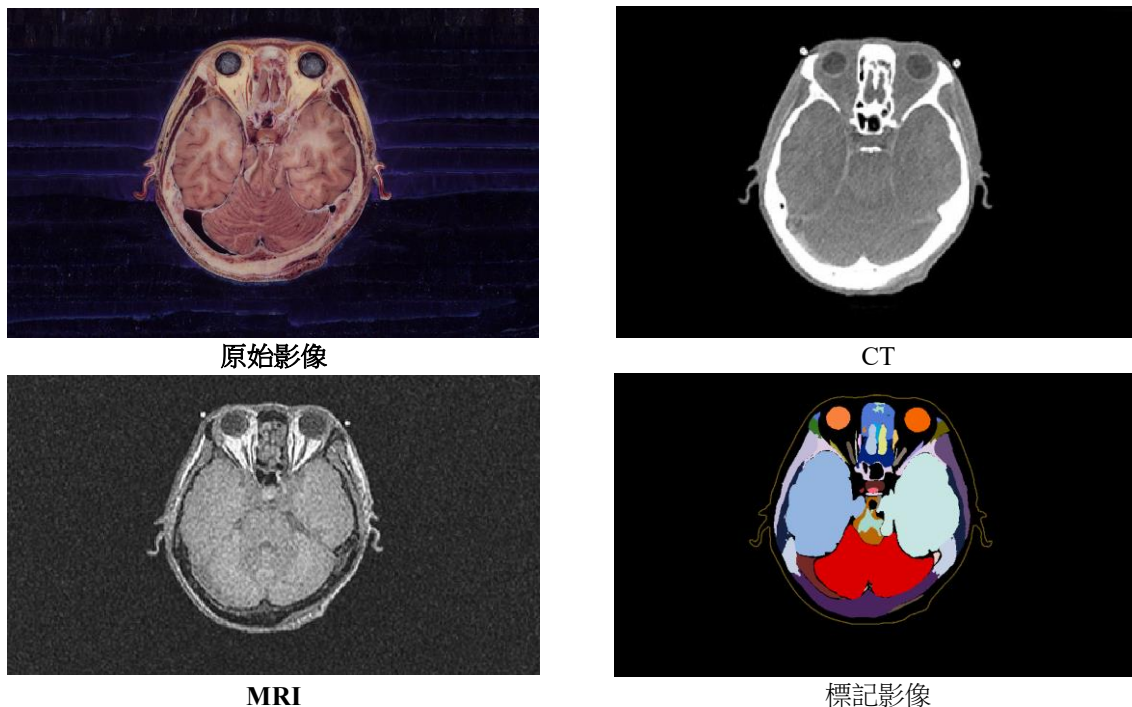
1. 取模型的 10 個簡化樣本 (簡化率 0%、10%、...、90%)，計算簡化後三角形數量與原模型之 Hausdorff

distance。

- 透過 Spline 內插計算出模型在預估簡化率為 0%、1%、2%、...、100%後的三角形數量與原模型之 Hausdorff distance，找出簡化過程中斜率變化最大的位置即為最佳的簡化模型百分比。

圖 22

影像資料分為原始影像、CT、MRI、標記影像四種

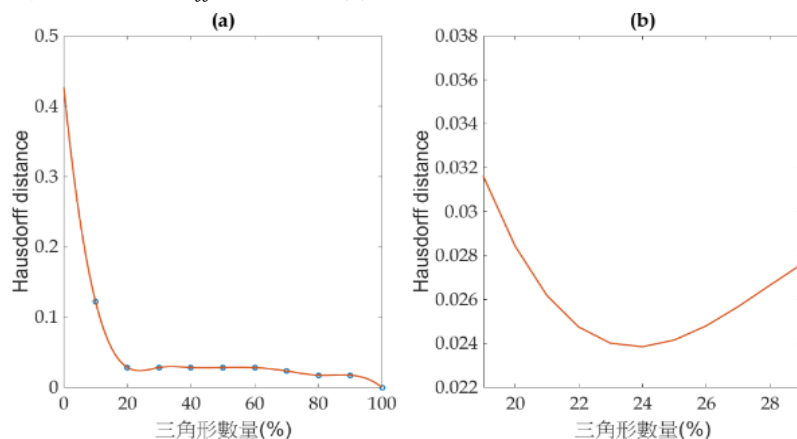


(一) 心臟模型

根據標記影像，該部位在原始影像所包含的部位為 pericardial cavity、Pulmonary、Superior pulmonary、Inferior pulmonary，原始影像大小為 $442 \times 338 \times 282$ ，縮小後的影像大小為 $256 \times 196 \times 163$ 。將 10 個簡化樣本的模型與該樣本的三角形數量與 Hausdorff distance 透過 Splines 內插計算出模型的三角形簡化率在 0~100% 的情況，並透過二次微分計算出斜率變化最大的位置，找出最佳簡化百分比，再透過此數值讓簡化演算法實作出最佳的簡化模型，如圖 23 所示。最佳化的模型簡化率為 76%，三角形數量為 97,868 個，Hausdorff distance 為 0.028438，如圖 24 所示。

圖 23

三角形數量與 Hausdorff distance 的折線圖

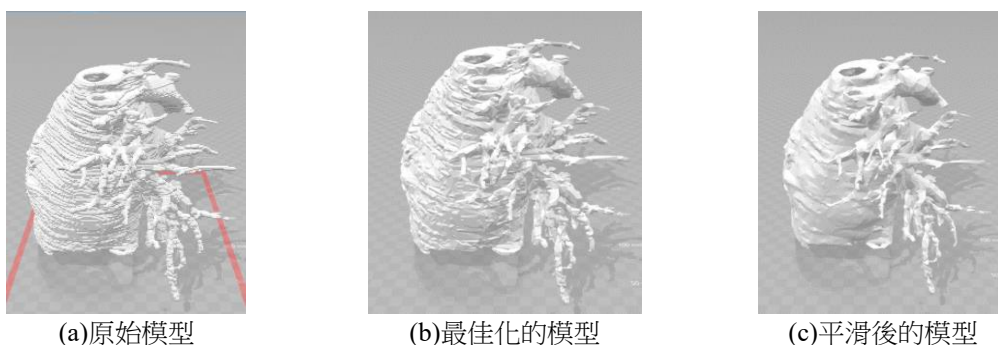


(a) 樣本透過 Spline 內插計算出三角形數量 0~100% 的折線圖

(b) 斜率變化最大的位置在三角形數量為 24% (簡化率 76%) 的位置

圖 24

最佳化的心臟模型，簡化率為 76%，三角型數量為 97,868，Hausdorff distance 為 0.028438



(a)原始模型

(b)最佳化的模型

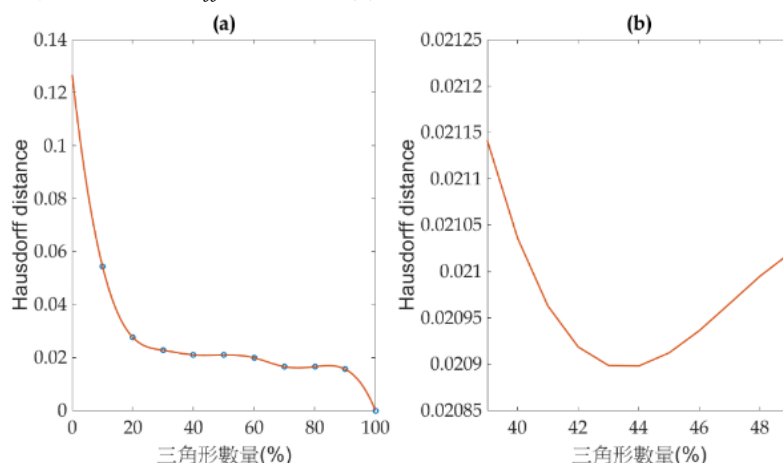
(c)平滑後的模型

(二) 肺部模型

根據標記影像，該部位在原始影像所包含的部位為 Lung，原始影像大小為 $613 \times 388 \times 650$ ，縮小後的影像大小為 $241 \times 153 \times 256$ 。將 10 個簡化樣本的模型與該樣本的三角形數量與 Hausdorff distance 透過 Splines 內插計算出模型的三角形簡化率在 0~100% 的情況，並透過二次微分計算出斜率變化最大的位置，找出最佳簡化百分比，再透過此數值讓簡化演算法實作出最佳的簡化模型，如圖 25 所示。最佳化的模型簡化率為 56%，三角型數量為 175,584 個，Hausdorff distance 為 0.017027，如圖 26 所示。

圖 25

三角型數量與 Hausdorff distance 的折線圖



(a) 樣本透過 Spline 內插計算出三角形數量 0~100% 的折線圖

(b) 斜率變化最大的位置在三角形數量為 44% (簡化率 56%) 的位置

圖 26

最佳化的肺部模型，簡化率為 56%，三角型數量為 175,584，Hausdorff distance 為 0.017027



(a)原始模型

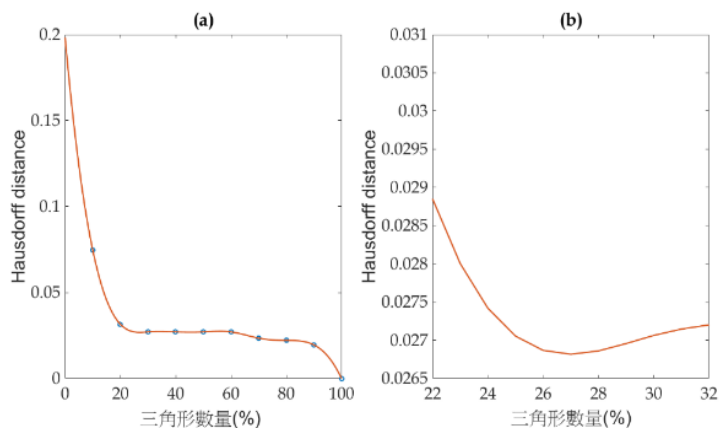
(b)最佳化的模型

(c)平滑後的模型

(三) 腦部模型

根據標記影像，該部位在原始影像所包含的部位為 Frontal lobe、Parietal lobe，原始影像大小為 $318 \times 330 \times 214$ ，縮小後的影像大小為 $247 \times 256 \times 166$ 。將 10 個簡化樣本的模型與該樣本的三角形數量與 Hausdorff distance 透過 Splines 內插計算出模型的三角形簡化率在 0~100% 的情況，並透過二次微分計算出斜率變化最大的位置，找出最佳簡化百分比，再透過此數值讓簡化演算法實作出最佳的簡化模型，如圖 27 所示。最佳化的模型簡化率為 73%，三角型數量為 159,640 個，Hausdorff distance 為 0.027063，如圖 28 所示。

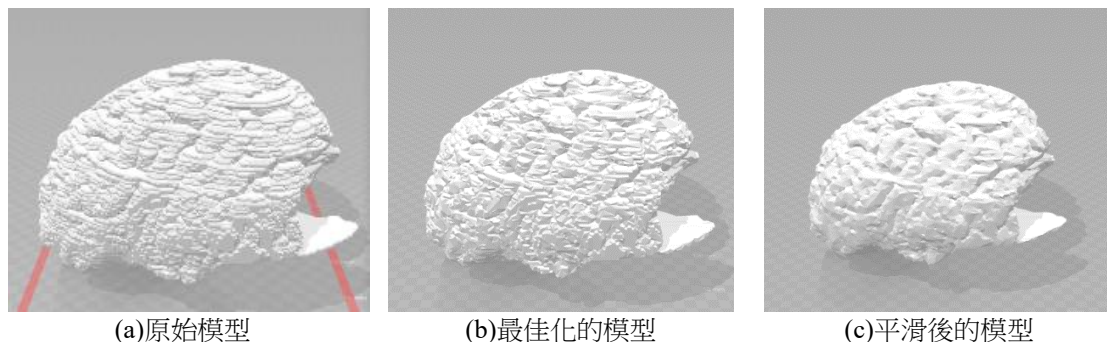
圖 27
三角型數量與 Hausdorff distance 的折線圖



(a) 樣本透過 Spline 內插計算出三角形數量 0~100% 的折線圖
(b) 斜率變化最大的位置在三角形數量為 27% (簡化率 73%) 的位置

圖 28

最佳化的腦部模型，簡化率為 73%，三角型數量為 159,640，Hausdorff distance 為 0.027063



(a) 原始模型

(b) 最佳化的模型

(c) 平滑後的模型

伍、結論

本研究的主要目的可以分成兩個部分，分別為 3D 醫學影像互動系統與最佳模型簡化程度演算法。主要是希望透過 3D 醫學影像與虛擬實境結合讓人體組織以更逼真方式呈現，以增進使用者對於人體結構組織的認知。在模型編修與簡化方面，由於 Marching Cube 所建立出來的 3D 模型有的三角型數量龐大，會占據許多儲存空間。本研究提出透過 Hausdorff distance 與 Spline 在不影響模型品質下將模型的三角型數量減到最少，以提升顯示計算效率。本研究未來可以朝手術模擬的方向前進，期待可以改進目前醫學教學環境，透過虛擬實境技術提供更為真實的教學體驗。

參考文獻

- [1] Virtual & Augmented Reality: The Next Big Computing Platform? (2016/2/9) <http://www.goldmansachs.com/our-thinking/pages/virtual-and-augmented-reality-report.html>
- [2] Sutherland, L.M., Middleton, P.F., Anthony, A., Hamdorf, J., Cregan, P., Scott, D., and Maddern, G.J. (2006). Surgical simulation, *Ann Surg*, 243(3), 291–300. <https://doi.org/10.1097%2F01.sla.0000200839.93965.26>
- [3] 算法小丑 (2016/6/22)。三維等值面提取演算法 (Dual Contouring)。博客園。
<https://www.cnblogs.com/shushen/p/5607833.html>
- [4] Cook, R., Hourvitz, L., and SIGGRAPH (1995). *Computer graphics*. ACM SIGGRAPH.
- [5] Zhang, N., Hong, W., and Kaufman, A. (2004). Dual contouring with topology-preserving simplification using enhanced cell representation. In *IEEE Visualization 2004* (pp. 505–512).
- [6] Schaefer, S., Ju, T., and Warren, J. (2007). Manifold dual contouring. *IEEE Trans Vis Comput Graph*, 13(3), 610–619.
- [7] Bloomenthal, J. (1988). Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5(4), 341–355.
- [8] Ccjou (2010/4/22). Gram-Schmidt 正交化與 QR 分解, 線代啟示錄, <https://ccjou.wordpress.com/2010/04/22/gram-schmidt-%e6%ad%a3%e4%ba%a4%e5%8c%96%e8%88%87-qr-%e5%88%86%e8%a7%a3/>
- [9] Park, I., Shirani, S., and Capson, D.W. (2005). Area of surface as a basis for vertex removal based mesh simplification. In *2005 IEEE International Conference on Multimedia and Exp* (pp. 4–).
- [10] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1993). Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 19–26). <https://doi.org/10.1145/166117.166119>
- [11] Hoppe, H. (1996). Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (pp. 99–108). <https://doi.org/10.1145/237170.237216>.
- [12] Garland, M. and Heckbert, P.S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 209–216). <https://doi.org/10.1145/258734.258849>
- [13] Li, G., Wang, W., Ding, G., Zou, Y., and Wang, K. (2012). *The edge collapse algorithm based on the batched iteration in mesh simplification*. In *2012 IEEE/ACIS 11th International Conference on Computer and Information Science* (pp.356–360).
- [14] WZFish0408 (2017/8/19). QEM 網格簡化。簡書。<https://www.jianshu.com/p/2bf615c38165>
- [15] Li, Y. and Zhu, Q. (2008). A new mesh simplification algorithm based on quadric error metrics. In *2008 International Conference on Advanced Computer Theory and Engineering* (pp. 528–532).
- [16] Wang, W., Zhou, M., Shui, W., Wu, Z., and Zhou, Y. (2016). An improved quadric error metrics algorithm based on triangle strips. In *2016 IEEE International Conference on Computational Electromagnetics (ICCEM)* (pp. 149–151).
- [17] Li, L., He, M., and Wang, P. (2010). Mesh simplification algorithm based on absolute curvature-weighted quadric error metrics. In *2010 5th IEEE Conference on Industrial Electronics and Applications* (pp. 399–403).
- [18] Li, T., Pan, Q., Gao, L., and Li, P. (2017). A novel simplification method of point cloud with directed Hausdorff distance. In *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 469–474).

- [19] Zhang, D., Zou, L., Chen, Y., and He, F. (2017). Efficient and accurate Hausdorff distance computation based on diffusion search. *IEEE Access*, 6, 1350–1361.
- [20] Xu, H., Jing, W., Li, M., and Li, W. (2016). A slicing model algorithm based on STL model for additive manufacturing processes. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 1607–1610).
- [21] Park, J.S., Chung, M.S., Hwang, S.B., Shin, B.-S., and Park, H.S. (2006). Visible Korean human: Its techniques and applications. *Clin Anat*, 19(3), 216–224. <https://doi.org/10.1002/ca.20275>