

國立暨南國際大學資訊工程學系

碩士論文

建構在模型辨識原形上的認知型 LOD 系統

The Perceptual LOD System based on the Recognition

Prototypes of Models

指導教授：陳履恆 博士

研究生：劉鎮宇

中華民國九十七年六月

致謝

研究生的生活，兩年很快就過去了，回想我這兩年來的作為，我幾乎都在實驗室裡活動，在此地我也學到很多的東西，尤其是學長們，幾乎會不厭其煩的回答我們的疑問，讓我們可以學習更廣泛的知識。特別要感謝指導教授陳履恆博士在研究上的指引，讓我們在遇到問題時，適時的糾正，而不至於長期陷入嚴重的瓶頸。

再者，要感謝實驗室的各位伙伴，因為有你們，這兩年半的研究生活過得很愉快，感謝 404 的學長、同學、學弟，感謝小溫學長用心的帶我進入 LOD 的研究；感謝 pointer 在我寫程式遇到問題時能給我寶貴的意見；感謝文祥可以常常帶我出去遊玩；感謝 bird、沛宇、小蘭&瑪利、宜靜、panda、小俞、阿俊、小龍等等因為有你們，研究生生活有人可以聊天、陪伴而能在苦悶或失意時繼續大力往前邁進，也感謝學弟妹們在口試時的各種事項的準備與安排。

在實驗室中其實是很歡樂的，尤其在學弟妹們進來後，整個實驗室人數大增，因此我們更容易常常舉辦聚餐和活動，讓我們在研究之閒，可以放鬆心情和降低壓力。最後，我希望這個實驗室可以長存，讓我未來工作之餘，可以常常回來探望實驗室狀況和探望老師，也可以當作未來學長、同學或學弟妹們的聚會之地。

論文名稱：建構在模型辨識原形上的認知型 LOD 系統

校院系： 國立暨南國際大學資訊工程學系

頁數： 47

畢業時間：九十七年/六月

學位別： 碩士

研究生： 劉鎮宇

指導教授：陳履恆 博士

論文摘要

現今電腦圖學的領域中，LOD(level of Detail)是經常於電腦圖學相關應用中被採用的技術之一，LOD 為當三維模型隨著視點距離的增加而對模型的網格做簡化以減少運算的技術，在我們的 LOD 研究中，我們加入了 Human Perceptual System (HPS)的要素，目的在於當簡化模型面時以人類視覺的認知來維持模型的特徵，使的模型在簡化的過程都有較佳的可辨認性。

在我們的研究中，我們修改了傳統 QEM 的簡化方式。傳統的 QEM 以權重平均的方式逐步將模型簡化，直到模型面數為零；因此我們提出了認知型 LOD 系統來修改簡化的方法，並且採用 HPS (Human Perceptual System) 的概念。

為了更進一步改進我們的認知導向性 LOD 系統，我們分析複數相關模型產生的資料，來建立出 HPS 的認知原型。其中思想在於模型簡化時，將模型的特徵部位保留，使特徵不在簡化的過程中被破壞而影響人類視覺上對模型的辨認。

此外，模型投射的面積相對於簡化的程度也是我們研究的主題之一，模型的面積也影響了觀察者對模型的認知能力。最後，我們利用原型(prototype)算出模型簡化的權重，來改善簡化的策略。有了我們提出的方法，我們可以保留重要的特徵，使的簡化到最低程度時，保持其可辨認性。

關鍵字：LOD, Simplification, Prototype

Title of Thesis : The Perceptual LOD System based on the Recognition Prototypes of
Models

Name of Institute : National Chi Nan University, Dept. of Information Management

Pages : 47

Graduation Time : June 2008

Degree Conferred : Master

Student Name : Zhen-Yu Liu

Advisor Name : Dr. Lieu-Hen Chen,

Abstract

Level-of-detail (LOD) is one of the significant techniques which widely applied in the field of computer graphics. LOD is a kind of technique that a mesh of 3D model is simplified according to the distance between viewer and model in a scene. We combine Human Perceptual System to our LOD research in order to using human's reorganization of sight to retain model's characteristic for simplifying model. It makes simplified models more identifiable.

In this research, we implemented our perceptual LOD system based on the QEM algorithm. The traditional methods reduce the complexity of model gradually until there is no surface left, mainly based on the geometric consideration. Therefore, we proposed our perceptual LOD system by modifying the mesh simplification sequence of QEM and adopting the concept of Human Perceptual System.

In order to further improve our perceptual LOD system, by extracting and analyzing the possible prototype from multiple related models, the cognitive prototypes in human perceptual system (HPS) are constructed. Our central idea is to maintain the object visually recognizable during the model simplification stage, by reserving the important features according to the characteristics of the cognitive prototype.

In addition, we take the size of projection area, which influence model's level of

detail, into consideration. The area has influence on user's recognizing ability for model. Finally, we use the weighting values which are determined from the prototypes to improve the simplifying strategy. With the proposed method, we can preserve the important features of HPS while model is simplified, so the simplified model can remain recognizable under lower LOD.

Keyword : LOD, Simplification, Prototype

目錄

致謝.....	2
論文摘要	3
Abstract.....	4
目錄.....	6
圖片目錄	7
圖表目錄	8
第一章 緒論	9
1.1 研究背景與動機.....	9
1.2 研究目的.....	9
第二章 相關研究	11
2.1 Human Perceptual System	11
2.2 Prototype	13
2.3 模型簡化演算法.....	15
第三章 系統架構與演算法	18
3.1 決定面績和模型之剩餘面數的關係.....	19
3.2 決定模型的起始面數.....	20
3.3 限制最低能簡化到的面數.....	21
3.4 限制模型能離視點的最遠距離.....	26
第四章 研究結果	29
第五章 結論	44
第六章 參考文獻	46

圖片目錄

圖 1 擁有很多樹模型的場景.....	10
圖 2 模型軀幹分類視窗.....	13
圖 3 模型隨著距離變化做化簡.....	18
圖 4 一開始擁有最大面數的 woman 模型.....	20
圖 5 一開始就做適當簡化的 woman 模型.....	21
圖 6 Dragon 模型由 2328 個三角形簡化到 116 個三角形.....	23
圖 7 Starfish 模型由 1048 個面簡化到 14 個面.....	25
圖 8 三角錐(不含底面)和三角柱(不含前後面).....	26
圖 9 由 pixel 構成的矩形.....	26
圖 10 Man 模型隨視點到模型的距離而簡化.....	28
圖 11 輸入模型 (man & woman class).....	29
圖 12 模型骨架.....	30
圖 13 模型依骨架的分段.....	30
圖 14 以很多球構成的過去研究結果.....	32
圖 15 Human 模型、骨架及其抽象化結果.....	34
圖 16 Dog 模型、骨架及其抽象化結果.....	35
圖 17 Horse 模型、骨架及其抽象化結果.....	37
圖 18 Turtle 模型、骨架及其抽象化結果.....	38
圖 19 Frog 模型隨視點到模型的距離而簡化.....	41
圖 20 Horse 模型隨視點到模型的距離而簡化.....	43

圖表目錄

表 1 本研究的實驗平台.....	29
表 2 人類群組原型資料表.....	31

第一章 緒論

1.1 研究背景與動機

在 3D 電腦圖學的應用中，場景越來越精細，也越來越複雜，場景中的模型往往由數十萬個面所構成，因此漸漸有人想出 Level of Detail(LOD)的方法來減少模型的計算成本，然而模型也有很多種類，例如地形圖[1]、動物、植物[2]、細胞等等的模型，其分別廣泛了地理學、生物學、植物學、醫學等等領域，因此 LOD 的應用範圍非常的大。

無論哪一種領域，在 3D 的電腦圖學的應用裡，最關鍵的是我們眼睛看到並且理解的資訊，經過人類的研究和分析後，才能成為領域上的知識。反之，從視覺心理學的領域著手，來針對 3D 電腦圖學領域的各種計算，其出來的結果理應會是視覺上效果較佳的結果。

1.2 研究目的

每一種動物都有其代表性的特徵，我們將每種動物之骨架資訊記錄下來，並把同種類的動物模型之骨架資訊做統計，來成為此種動物的原型(prototype)，之後讀進同種類的模型，就會知道其原型特徵而給予該模型權重，模型就可以依此權重做適當的簡化動作。我們得到的原型只有其數據，如何將統計出來的原型環以原貌則是另一個研究主題。

當模型可以做簡化後，就可以決定模型要在哪個時機做多少簡化的程度。在我們眼睛看到的部份，有些模型成像比較遠，有些比較近，近的模型當然會比較清楚，所以模型也會呈現比較複雜，遠的模型則相反(如圖 1)；在一個場景裡面，有些景物特別顯著，其模型也會呈現較複雜。有了這些概念，接下來就要研究，模型在一定的狀態，會簡化到多少程度？模型越遠，人類能夠辨認的程度也會變小，但是模型簡化到一定程度，模型的特徵就會消失，此時就會改觀人類對此模型的認知，因此，我們還必須限制模型能簡化到的程度。



圖 1：擁有很多樹模型的場景[2]

第二章 相關研究

1 Human Perceptual System

視覺感知

人類的感知系統，主要是由人類的五感刺激腦部的運作，而得到認知結果，五感分為視覺、聽覺、嗅覺、味覺、觸覺，除此之外，人類由生以來的經驗也會成為影響認知的要素，在我們的研究裡面，主要是以視覺的認知系統為主，因為電腦圖學的成像被決定的是眼睛看到的東西。

我們參考了[3]的描述，為了探討視覺認知的問題，主要分成了三個階段：

- 首先，視覺認知(看到並理解)牽涉到形狀、形體、顏色、輪廓、對比和動作的基本分析。這些元素由我們的眼睛中的末梢神經系統去感受。接著，以物理能量形式存在的電磁波訊號被轉換為電器化學訊號，在傳送到視皮質做進一步的處理。在後面這個階段裡，視覺認知和更高層次的處理過程才開始產生。
- 第二，這些根本的資訊被組織成基本的形體，而這些基本的形體是更高層次處理的基礎，而且大部分是不須事先學習或經驗就可以察覺。基本形體的例子就是『圖形和底色』，其中物體(也就是圖形)從背景(底色)中突顯出來，如同文字(圖形)從紙張(底色)中突顯出來。
- 第三，這些基本形體和我們的長期記憶產生連結，而我們儲存在長期記憶中對世界的一切了解，賦予了這些基本形體意義。有時候，我們把這個資訊處理模式的最後階段稱之為更高層次的認知，倒不是因為這個過程比較細膩，而是因為它是之前所有『低階』過程的結果。除此之外，我們腦部的思維會將我們的注意力導向視覺畫面的特定部分，給予顯著的特徵或個人感興趣的地方較多注意力。最後，腦部把更多的資訊加諸在這些原始的視覺印象上，進而對最初所接收到單純的視覺刺激，賦予了更豐富的意義。

視覺感知方面的議題

之前感知方面 LOD 的研究有 Martin Reddy[4]所提出的在一場景中因模型與模型間依視點不同的情況下產生不同的細緻程度，依視點所在位置與模型的各種考量因素保留視點所在位置而簡化可被忽略的面以達到 LOD 的目的，作者稱此為 View dependent simplification(VDS)，所考量的性質除了與視點的距離 LOD 外，還有因模型大小差異的 LOD、習慣性忽略外圍 LOD(Eccentricity LOD)、模型移動速率 LOD(Velocity LOD)、場景深度 LOD(Depth of field LOD)，以下一一做介紹：

距離 LOD：依距離差異產生細緻度的差異是 LOD 發展的基本概念，隨著模型離視點越遠則模型越不需保留其精細程度而簡化。

模型大小 LOD：在視覺感知的情況下，一場景中的兩個模型大小不同，通常會將較小的模型直覺上認為是較遠的，因此將較小的模型加以簡化，又有一種情況為兩個大小不同的模型與視點的距離相同，但人類視覺通常第一眼較會先集中在較大的模型上，因此場景中模型的大小差異可產生不同的 LOD。

習慣性 LOD：在人類視覺感知上通常會把注意力放在場景中間範圍，如場景周圍則會習慣性的忽略，因此可假設一視點射線剛好穿過場景中心點，而其他視線與中心視線會產生一視角的差異，以此視角的差異程度將在其位置的模型做 LOD。

速率 LOD：人類視覺對移動中的物體相較於靜止中的物體較無法辨認其周圍精細度，尤其是當模型移動速率增加則更加明顯，因此可依模型移動速率而將模型外圍俟對同程度的 LOD。

場景深度 LOD：從人類視覺的研究上來說，視覺上對物體能清楚辨別精細度事實上有個距離範圍，此範圍稱作 Panum's fusional area，模型離觀察者太近或太遠皆會造成視覺上的模糊，因此模型在此範圍外的空間皆可以簡化。

以上性質皆建立在場景或一個模型之間對視點的差異上，應用這些特性來對

整個場景以及模型做不同程度的簡化，需假設視點在畫面的中心點來進行實作，習慣性 LOD、速率 LOD 用(Contrast Sensivity Function, CSF)來做為簡化的尺度。

2 原型(Prototype)

原型生成

原型生成的方式是讀取一組同類型的模型先需將模型的各軀幹和主體做分類，分類的方式為程式自動找出分支後再加以手動分類為模型各個部位如圖 2，而同一組要做原型的模型皆需要做此分類，而由程式找出 default branch 的方法必須依賴 DCG[5]骨架的架構，找的方法是先找出骨架裡的 end point，以 end point 為起點沿著骨架往回找，直到找到 joint point 則停止搜尋，將找到的骨架曲線做標記，依此步驟即可找出模型的分支。

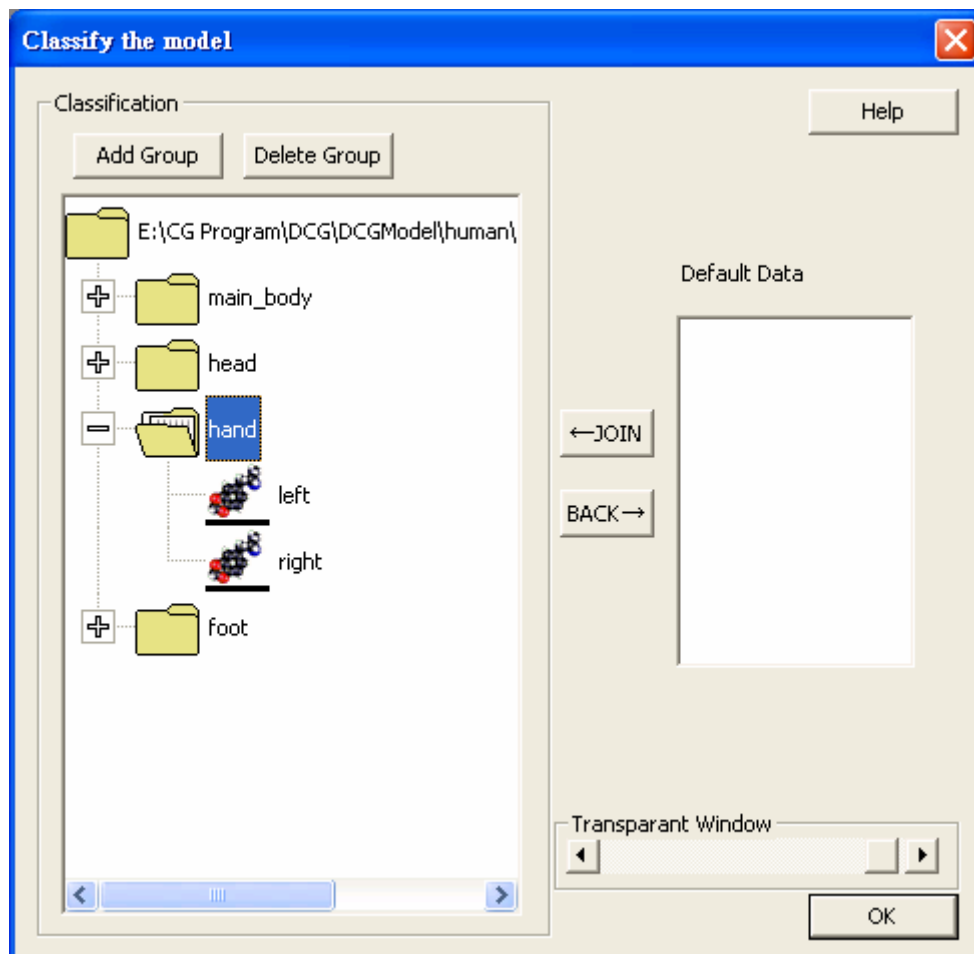


圖 2：模型軀幹分類視窗

接下來將各分類群組做統計，統計該類別中的原型需要有的必要條件，即是原型所應擁有的主體和軀幹種類與個數，考量方式為讀入 N 個模型，其中模型 x 中擁有軀幹 A，表示為 $x.branch(A)=1$ ，分辨其他的模型中是否軀幹 A，流程結束可以得到軀幹 A 的參考層級(referring level 以下簡稱 RL)，RL 可以以平均方式表示為：

$$RL = ((\sum_{k=1}^n a_k .branch (A)) / N) * 100 \% , N > 1$$

如此可以經由判斷知道軀幹 A 於該類別模型的必要性，將模型各自軀幹比對過後可以得出該類別原型的基本架構，可說是該類別可被正確辨認的必要條件。除此之外，為了能更清楚表達生成模型的辨認度，光有必要條件只能了解其基本架構，容易有兩不同類別的模型有相同的架構的情形出現，因此還需要加入其他和模型相關的特徵，軀幹和主體的各別粗細的簡單幾何特性可呈現出該模型的大略幾何形狀，測量的方式為藉由模型的骨架來計算以 domain points 到模型表面的 domain ball 半徑來求得粗細，求得各個模型中每個軀幹的粗細後，即要從中取得每個軀幹與主體最具代表性的粗細用在表示原型，不過在統計之前因每個模型的測得實際數據皆不同，因此必需先予以正規化。

正規化的用途為取得各個模型中其一軀幹的比例以方便做統計，正規化的方法為軀幹相對於主體的比例，如此才可以將單一模型軀幹的長度和粗細求出相對數據，而各個模型的主體資料數據則直接取其實際數據來做統計，經過正規化後每個軀幹各自有其粗細與長度的數據資料，接著的統計方法為常態分配(Normal distribution)，求出各軀幹與主體最具代表性的數據為該軀幹或主體的粗細與長度資訊，用所求得的資訊來建構原型。

原型的用途

I. 可得出模型的辨認度：依原型的"參考層級"來做為一項尺度，顯示出模型經簡化後的辨認程度，參考層級為原型經模型緬取出的各個分支和主體分別做有或無的統計，以平均方式得出單一分支中的對此原型值得參考程度，如生成原型的每一個模型皆具中某一分支則參考層級為 100%，如採樣 10 個模型中只有 9 個模型有該分支則參考層級為 90%，以此類推，依據參考層級可以知道某一模型可被正確辨認的必要條件為達到該原型的各個分支的參考層級，當模型簡化至失去參考層級的其中一個模型分支時，則可被正確辨認的程度就可能大為降低。

II. 最低簡化程度：依統計出辨認程度可知道在可以被正確辨認該模型的情況下最低可簡化的程度，除此之外，藉由原型統計出每項分支相對主體的半徑常態分佈可以來知道各分支的重要程度來做權重上的分配。

III. AI 的模型辨識：可依照模型原型來對任一模型做辨識，此非 LOD 的範疇，但為未來值得發展的一項議題。

3 模型簡化演算法

Vertex clustering

Vertex clustering 演算法最早由 Rossignac 和 Borrel 於 1992 年提出，做法為先將三維模型的點給予權重的分配，分配方式是將三角形所佔面積較大的三點以及凸起曲面上的轉折點給予較高的重要性，當進行簡化時，重要性較低的點靠往重要性較高的點，與之合併，這樣執行下來有些三角形會消失而達到簡化效果，此種方法的好處是實作容易，有效率且是最快的模型簡化演算法之一，但缺點是每次簡化的結果都不同且無法同步，也無法復原到較高解析度的程度，為此 Kok-Lim Low and Tiow-Seng Tan[6]提出了改進的方法，即每個點分配好權重後，加以排序，再將每個細胞(cell)中取得一最重要的點，將細胞內其他的點刪除，這樣即可避免之前所提出的錯誤，然而，此種改進方法只適合用在面數較少

的模型，如模型太大則執行時間冗長是本方法最大的缺點。

Vertex decimation

此演算法最早同樣於 1992 年提出，Vertex decimation 的做法為先將點做分類，再設定一個尺度標準，如點的尺度值超過尺度標準則將點做刪去，再將原本和刪去點有關的面加以重新分割，此種方法之後有很多演算法延伸和改進，並且長期被應用而成為 Visualization ToolKit[7]的一部分，此方法的好處是即使執行簡化的模型很大還是可以很快速地運作，且使用起來方便，但是在簡化過後模型精確度有其限制。

Quadric Error Metrics

QEM(Quadric Error Metrics)演算法於 1997 年由 Michael Garland 和 Paul Heckbert[8]所共同發表，QEM 為連續式 LOD 以動態的方式對模型中兩點成一線的邊(edge)做簡化，將模型兩相鄰的點分為一組，並用矩陣的資料形態計算其二次曲面誤差(quadric error)，將每一組的二次曲面誤差值推積成樹(heap tree)，在動態執行模型簡化時先選擇從二次曲面誤差值較低的一組邊簡化，一組兩點的邊做完刪去的動作後將這兩點合併並直接將誤差值相加而成為一新的點，並重新對模型計算二次曲面的誤差並排序成樹，重新選擇二次曲面誤差值較低的邊做簡化，直到簡化至該模型目標層級，QEM 演算法在 LOD 的相關研究歷程中可做為一項里程碑，此演算法的運算結果結合了速度、精確度和整體的健全性突破之前的各種模型簡化方法，但它也不是沒有缺點，用矩陣為資料結構的二次曲面誤差當需要增加其他屬性時則需要增加矩陣的維度，例如一頂點(Vertex)中的所存的陣列基本資料有座標值(x,y,z)，如需增加顏色資料則陣列資料形態則需改為(x,y,z,r,g,b)，如增加貼圖則為(x,y,z,tu,tv)以此類推，當矩陣維度增加，動態計算時的時間和記憶體的成本將會大幅度增加，並且在實作上難以對所增加的維度加以有效地控制，針對這些問題，之後路續有幾篇論文提出解決計算複雜度和節省記憶體的方法[9][10][11]。

Reverse simplification

由 Dima Brodsky 和 Ben Watson 於 2000 年發表[12]，結合了 vertex clustering 和 QEM 演算法再產生出一新的方法，做法為將模型依向量改變量做分群，最多用三個平面將模型分成八部分，每一部分依向量改變量來做排序，再將向量改變量最多的區塊再做一次分群，照這樣的方法重覆執行直到達到預計的分割數量，分割的每個群組用 QEM 的方法做簡化，簡化到指定的點數量，RSIMP(Reverse simplification)的速度比 QEM 快，但結果沒有比 QEM 好，此方法的特點為適合大型模型的簡化。

第三章 系統架構與演算法

提出這個研究的目的，是希望模型在漸漸遠離觀物窗時，配合 Prototype 決定的權重，讓模型做簡化三角形面數的動作，使得模型在遠處時，可以減少儲存記憶體消耗，尤其當多個模型存在於觀物窗時，更可以有效的節省空間；另外雖然模型離視點越遠，模型三角形個數則越少，但相對於觀物者(或稱使用者)，因為遠處的模型可視度有限，因此不會影響其真確性。針對上述的目的，我們主要研究的部份，是提出幾個方法來計算：當模型遠離觀物窗時，其三角形數應該化簡到的程度為何？並且不輕易的讓觀物者看到模型簡化後的破裂、失真等等視覺上的錯誤。

為了解決這類問題，過去有針對 Height Fields 模型提出的方法[13]，以及針對 3D 動畫的研究[14][15]，我們則是針對動物模型來研究。首先提出的方法是用最直觀的線性方法以離視點的距離當作參數來決定簡化的變數，並定義一個最終的距離，模型再遠也不會超過這個距離，此時我們就可以用差值法來計算模型目前位置對應的化簡。

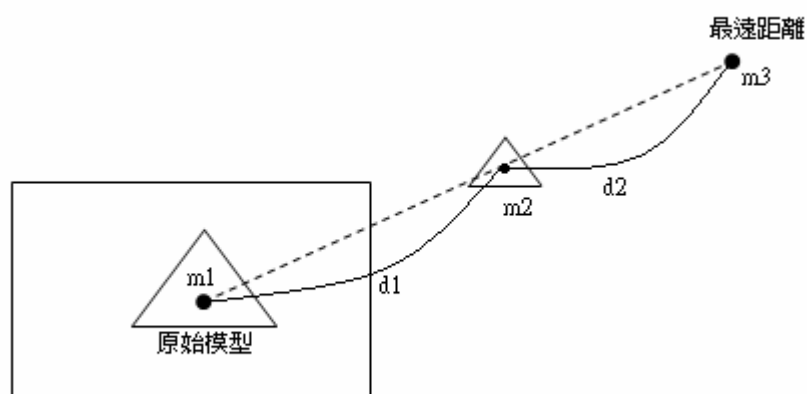


圖 3：模型隨著距離變化做化簡

參考圖 3 其公式如下： $m2 = \frac{d1 * m3 + d2 * m1}{d1 + d2}$ ，其中 $m1$ 表示原始模型三角形數目， $m3$ 表示最低能簡化到的三角形數目，因此 $m2$ 算出來的結果代表此模

型在距離變化後的三角形數目，也就是說模型將會簡化到剩下此三角形數目。

雖然線性解法很直觀，運算也很簡單，但是其最遠距離的定義也是一個問題，此距離太近會難以顯出本目的之效果，太遠則因人的視覺力有限，會造成多餘的運算，所以距離的取捨也是問題關鍵所在，還有最低能簡化到的三角形面數也是一個問題，這將是另一個研究的主題。

除了線性以距離為參數的方法外，我們利用 DCG 產生的 Segment 並以面積為主要參數的方法來決定簡化的公式和最低能簡化到的程度，此時的面積，指的是模型投射到螢幕畫布所佔的面積，其單位為 pixel。因此面積剩餘多少，直接影響模型簡化到的剩餘面數，我們以這方法作為研究的主题。研究的主题順序如下：

1. 決定面積和模型之剩餘面數的關係
2. 決定模型的起始面數
3. 限制最低能簡化到的面數
4. 限制模型能離視點的最遠距離

3.1 決定面積和模型之剩餘面數的關係

我們的目的是模型隨著遠離視點而簡化面數，而隨著和視點的遠離，模型呈現在螢幕的面積也會隨著減少，因此我們以面積為主要參數，來影響模型的簡化。為了決定面積和簡化的關係，我們考慮到一個三角形面最小能被視為一個面的面積為 4.5 pixel(3.4 會提到)，意思是說剩餘面數：面積 = 1：4.5，不過呈現在螢幕上的面積是只考慮模型正面，所以比例為剩餘面數：面積 = 0.5：4.5，也就是 1：9。由這比例可以知道，當模型遠離視點時，模型會以當下的面積除以 9，來當作簡化後的剩餘面數，因此就能達到連續性的簡化效果。

3.2 決定模型的起始面數

有了我們決定的面積和簡化關係後，還必須克服一些問題，以圖 4 為例，模型的起始面數為 64855，而此模型離視點最近的時候面積為 33174 pixel，此時的面數相對於面積的比例呈現過多的狀態，因此我們另模型在一開始的時候，就先計算其應該簡化到的面數，以上面的例子來說，雖然模型有 64855 個面，但因為面積為 33174 pixel，所以一開始會直接簡化到 3486 個面，使比例成為 1:9，如圖 5。反之，如果模型起始的面數為過少的情形發生，我們將不給予簡化，只有當模型做遠離視點的動作後，其面積符合當前的面數時，才會開始簡化，舉例來說，某個模型起始面數為 800，其離視點最近時的面積為 200000 pixel，因無法讓模型繁化，所以此時不作任何動作，唯有當模型開始遠離視點，直到面積大約為 7200 pixel 之後，就會達到開始簡化的起始點，因此若繼續遠離使的面積為 5400 pixel，此時模型會簡化到 600 個面數。

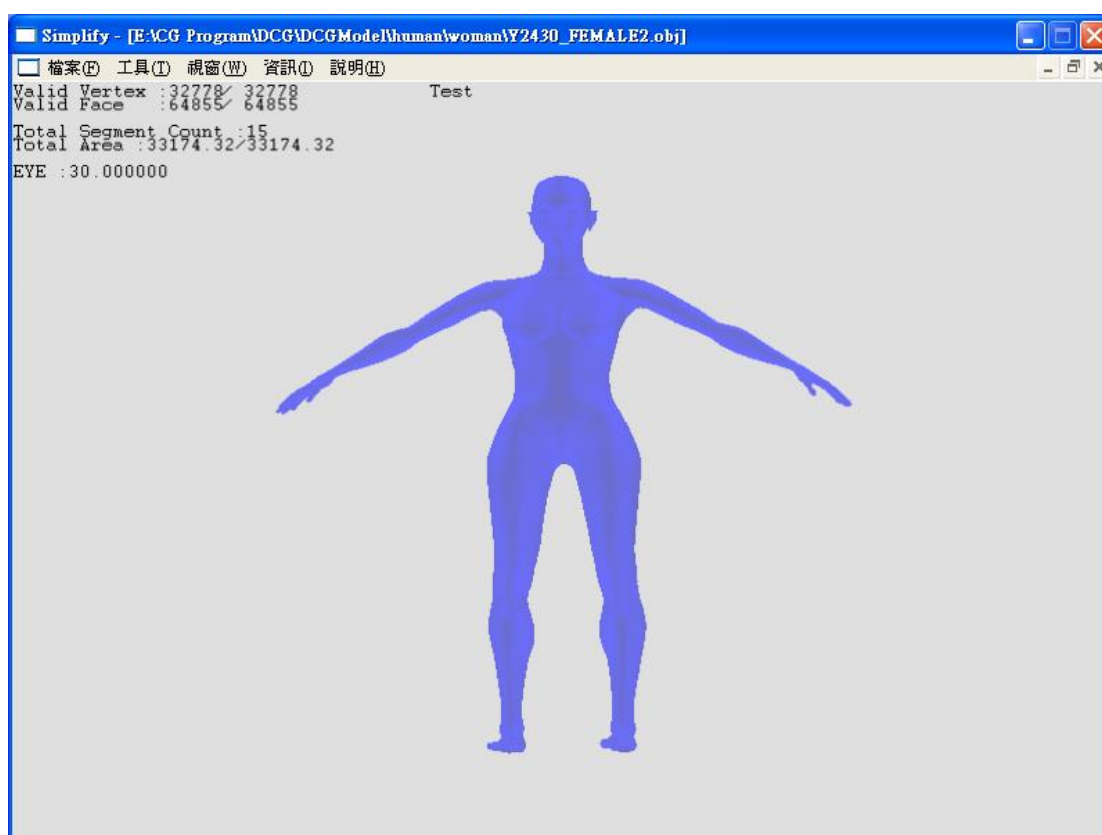


圖 4：一開始擁有最大面數的 woman 模型

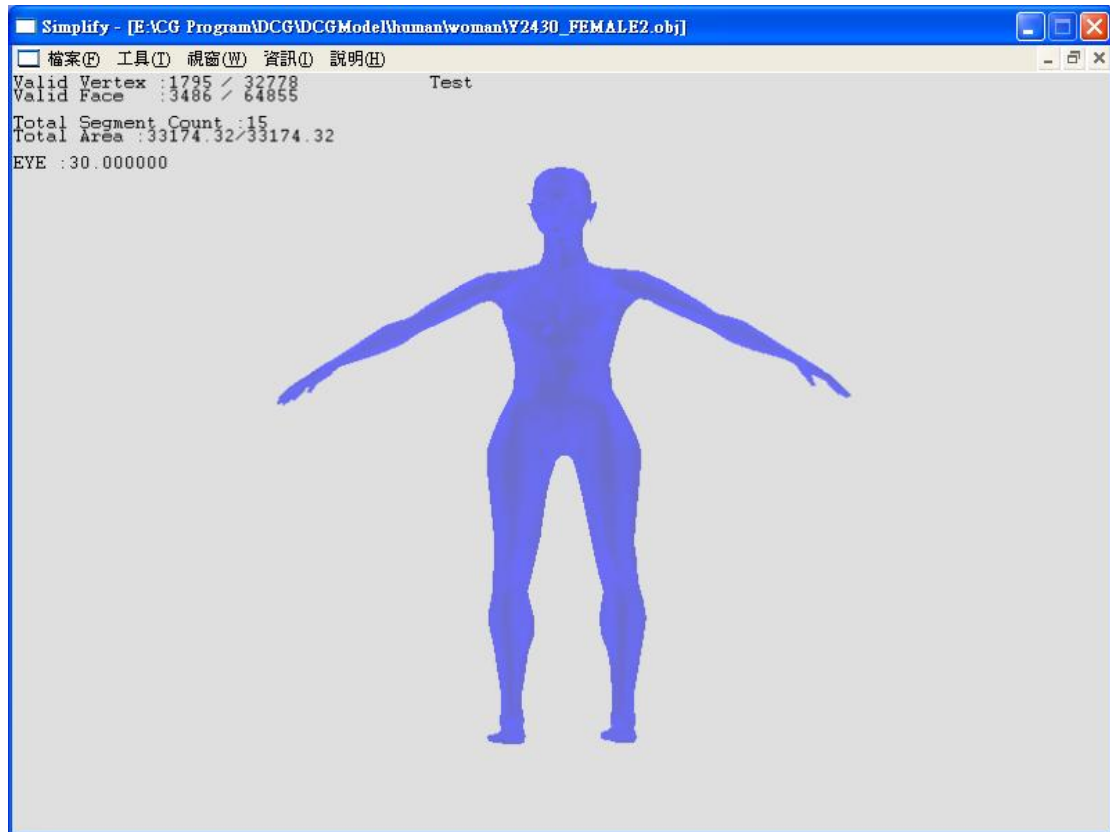
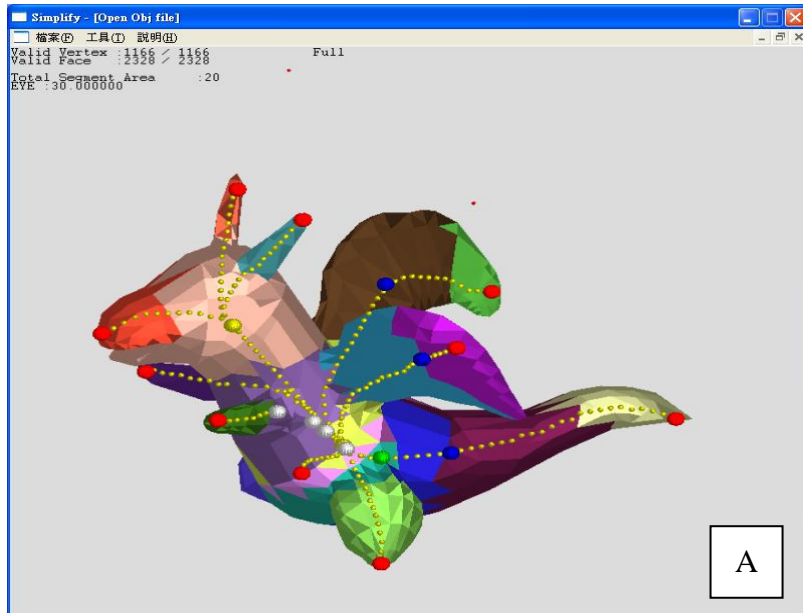


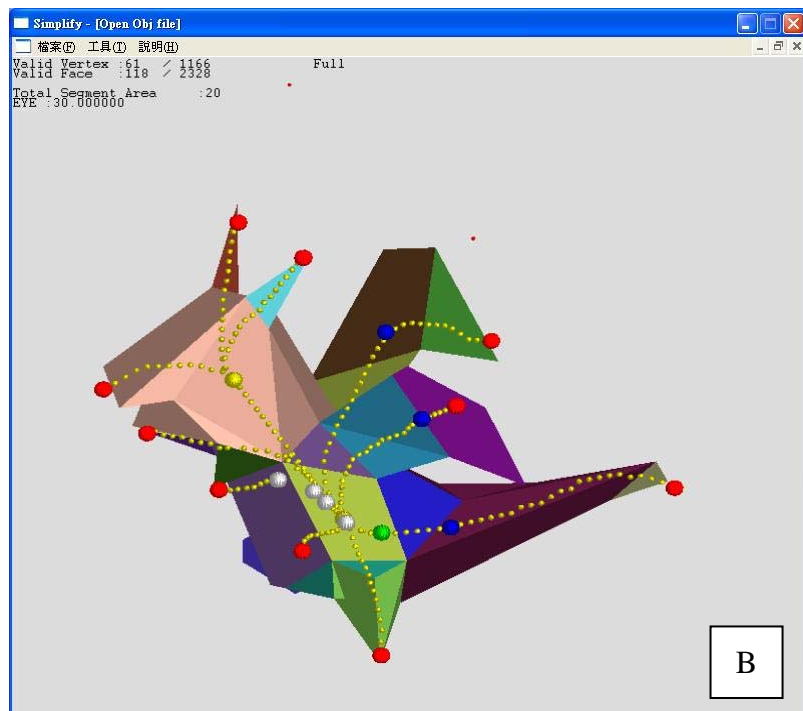
圖 5：一開始就做適當簡化的 woman 模型

3.3 限制最低能簡化到的面數

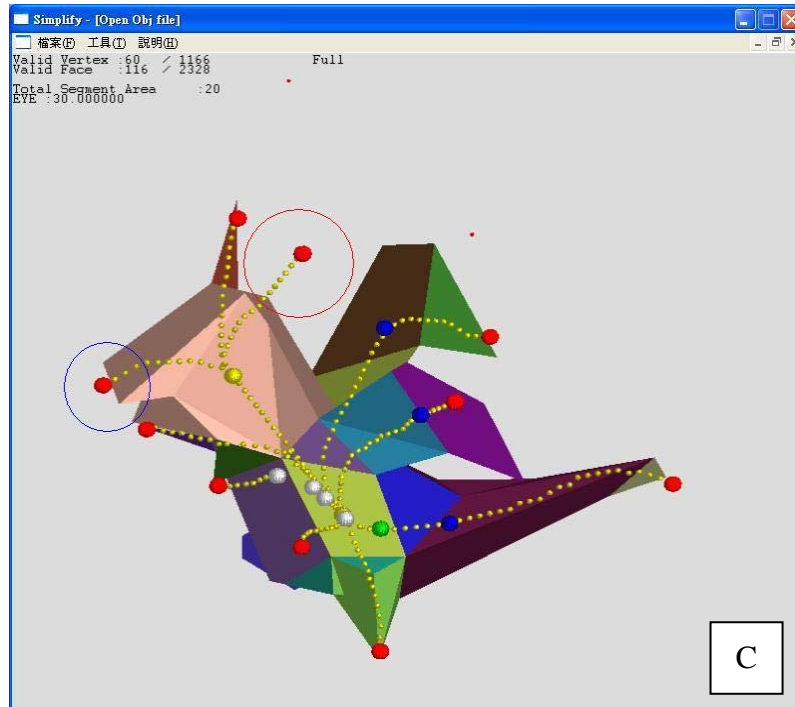
任何 LOD 方法的化簡，當化簡到達一個限度時，其模型必定會變成讓使用者無法辨認其原始形狀，一般的作法是手動調整參數，讓模型酌量還原成多變形數較多的狀態，使其特徵點明顯而且容易讓人辨認形狀，針對這一點我們提出一個問題，是否可以讓電腦判斷模型的簡化之最低程度？這是這個主題主要探討的問題。



含 2328 個面之 dragon 模型



含 118 個面之 dragon 模型

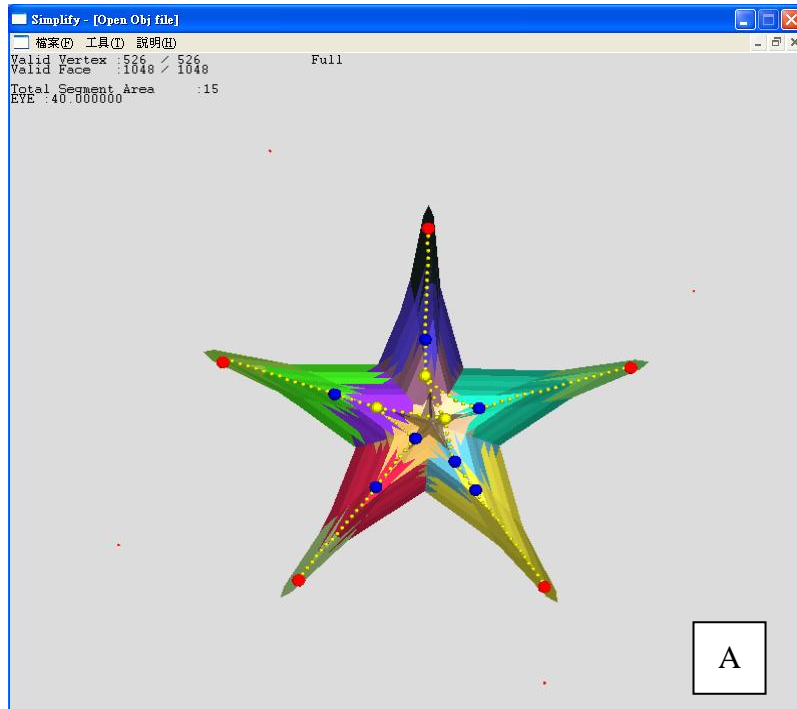


含 116 個三角形之 dragon 模型

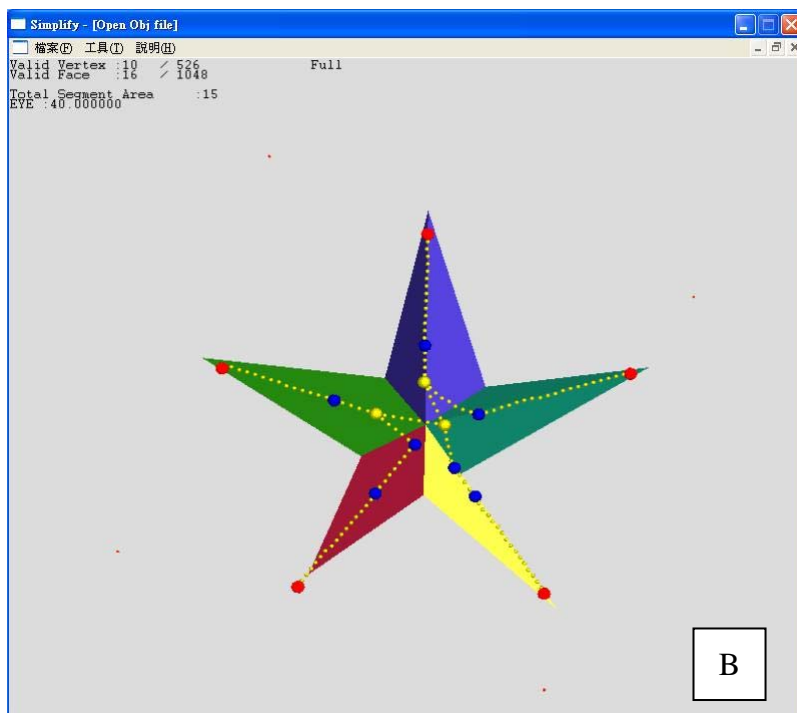
圖 6：Dragon 模型由 2328 個三角形簡化到 116 個三角形

由圖 6 來看，模型從(A)開始簡化到剩(B)三角形時，其模型依然可以辨認出為 dragon 的形狀，不過在化簡到(C)時，(C)中的紅圈的 end point 區域之三角形幾乎不存在了，使得 dragon 失去了耳朵的特徵，此時我們稱(B)的狀態為模型在可辨識情況下的化簡之最低程度，另外在(C)中的藍圈部份，雖然在早前化簡時，已經化簡掉此 end point 區域大部分的三角形，但是其三角形數目依然足夠到不至於破壞特徵，所以此時依然可繼續化簡，不構成視覺上的錯誤。

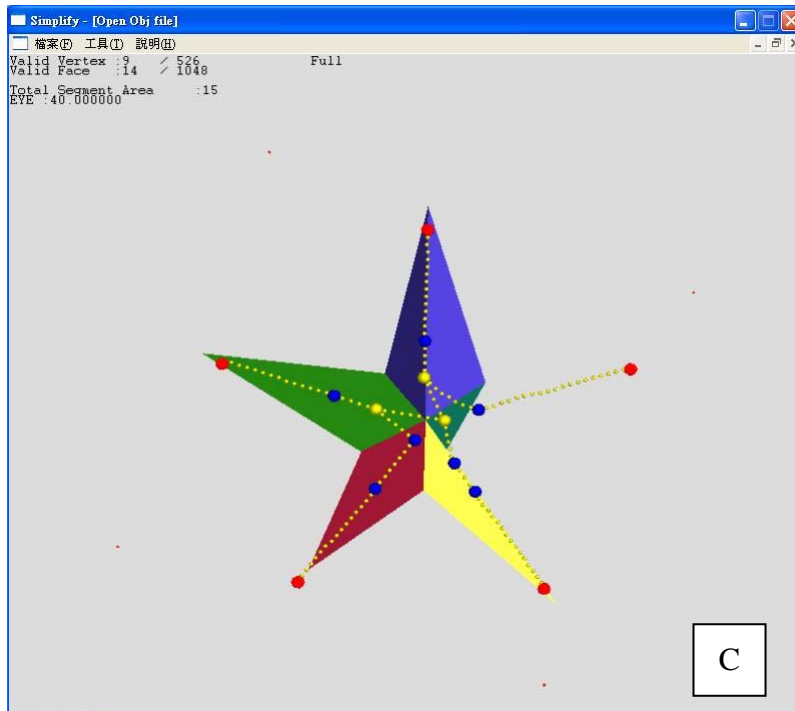
依照我們目前的研究，可以透過前面講的方法建立各類模型的 prototype，而當模型要做簡化的當時，可以去參考此 prototype 的數據，給予各特徵部位適當的權重，以達到第二章提到的 Characteristic weight optimization。不過雖然有了權重分配，依然還不能決定 (B)的狀態之模型。



含 1048 個三角形之 starfish 模組



含 16 個三角形之 starfish 模組



含 14 個三角形之 starfish 模組

圖 7：Starfish 模型由 1048 個面簡化到 14 個面

為了解決這個問題，首先我們利用 DCG 產生的 Segment 數據，知道了模型每個 domain ball 的分布，而每個 domain ball 的 Segment 關係到特徵存在與否，也就是說，假如某個 domain ball 的 Segment 之面數過少，則模型將會呈現破裂。因此，我們假設每個 End Point 的 Segment 最小可以由三角錐構成(圖 8 的 A)，此三角錐有 3 個三角形面，Connection Point 的 Segment 最小可以由三角柱構成(圖 8 的 B)，此三角柱有 6 個三角形面，而 Joint Point 的 Segment 則最小可以由一個 Box 構成，Box 共有 12 個三角形面，最後我們再依 Domain ball 的不同，把最小可以構成的面數加總起來，來當作模型最低能簡化到的面數，所以當模型簡化到剩此面數時，模型將不會繼續簡化，以保持模型的可辨度。在此狀況下，模型依然可以繼續遠離視點直到視點的限制達到為止，途中不會進行簡化。

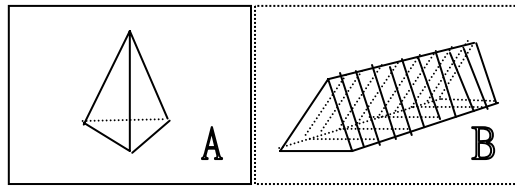


圖 8：三角錐(不含底面)和三角柱(不含前後面)

3.4 限制模型能離視點的最遠距離

當模型一直遠離視點，其投影在螢幕的面積也會隨著減少，但是當模型離視點遠到一個程度，其投影在螢幕的範圍將小到無法辨認形狀，因此我們必須限制此距離，來避免發生這種情況。

假設一個面在螢幕上的顯示以 4.5 pixel 為最小可辨認此面的面積，而模型剩餘的面數中，粗略估計正面和背面各佔一半面數，因此我們設定以剩餘面數的一半乘以 4.5 為模型最小能達到的面積，也就是說當模型的面積達到此限制時，將不能再遠離視點。以上提到的最小面積為 Chapter 3.3 所提出的最低能簡化到之面數，每個面以最小 4.5 pixel 的理由如下。

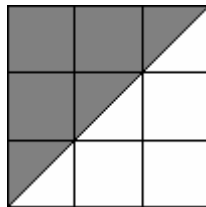
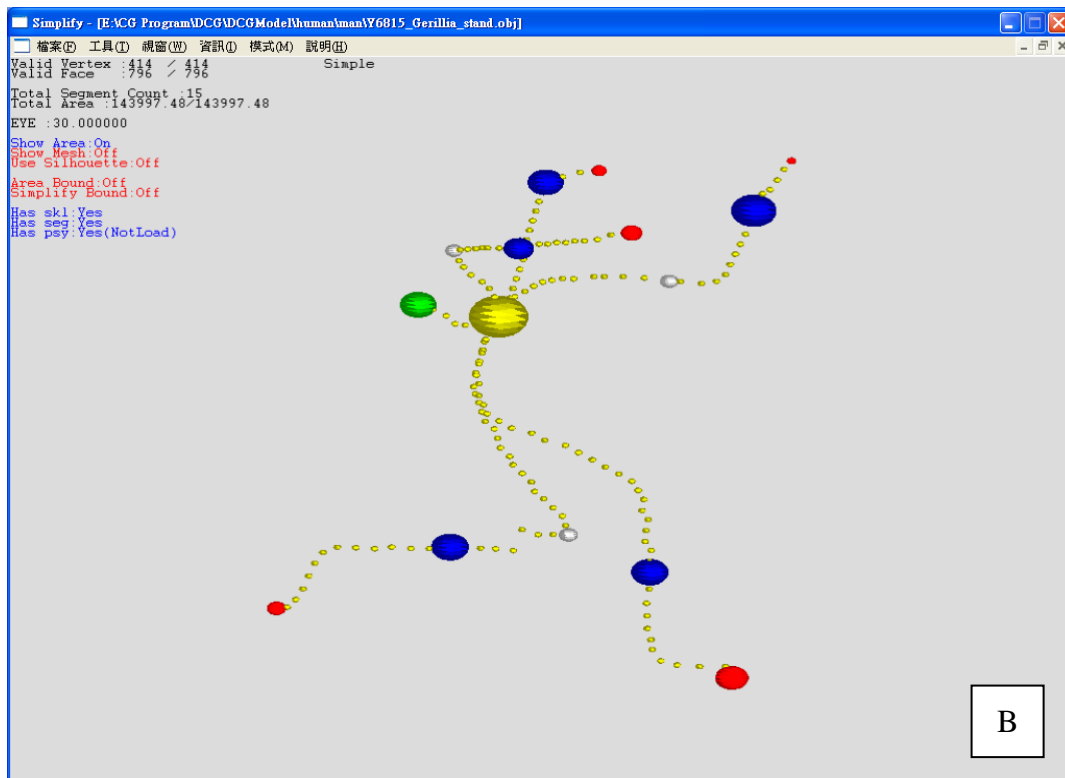
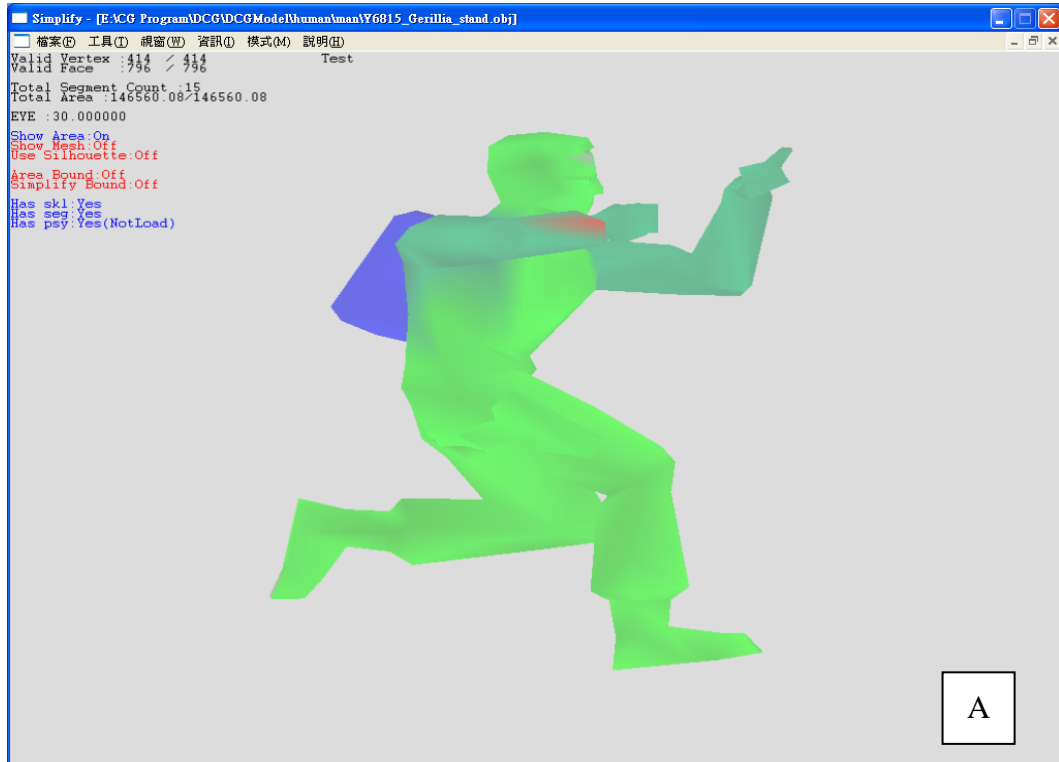


圖 9：由 pixel 構成的矩形

從圖 9 可以知道，一個矩形在可辨認其為矩形的條件下之最小面積約為 9 pixel(如果再小下去就變成 $2*2=4$ 個 pixel，呈現的幾乎是兩個點)，此時兩個面共約為 9 pixel，因此我們以每個面 4.5 pixel 來當作最小可辨認此面的面積。以 Man 模型為例，此模型剛讀入時含有 796 個面和 25 個 segment(圖 10 的 A)，其中有 5 個 End points、8 個 Connection points 和 1 個 Joint point，因此這模型最低能簡化到剩 $5*3 + 8*6 + 1*12 = 75$ 個面，此時 3.3 的限制將啟動(圖 10 的 D)，模型可隨

著視點繼續拉遠而但不簡化，圖 10 的 C 可以看到模型簡化到最小並拉近視點的結果。而當模型離視點到一個程度，也就是此時模型投影到螢幕的面積小於 $75 / 2 * 4.5 = 168.75$ pixel 時，視點距離的限制將啟動，模型離視點的距離將不能再繼續增加。



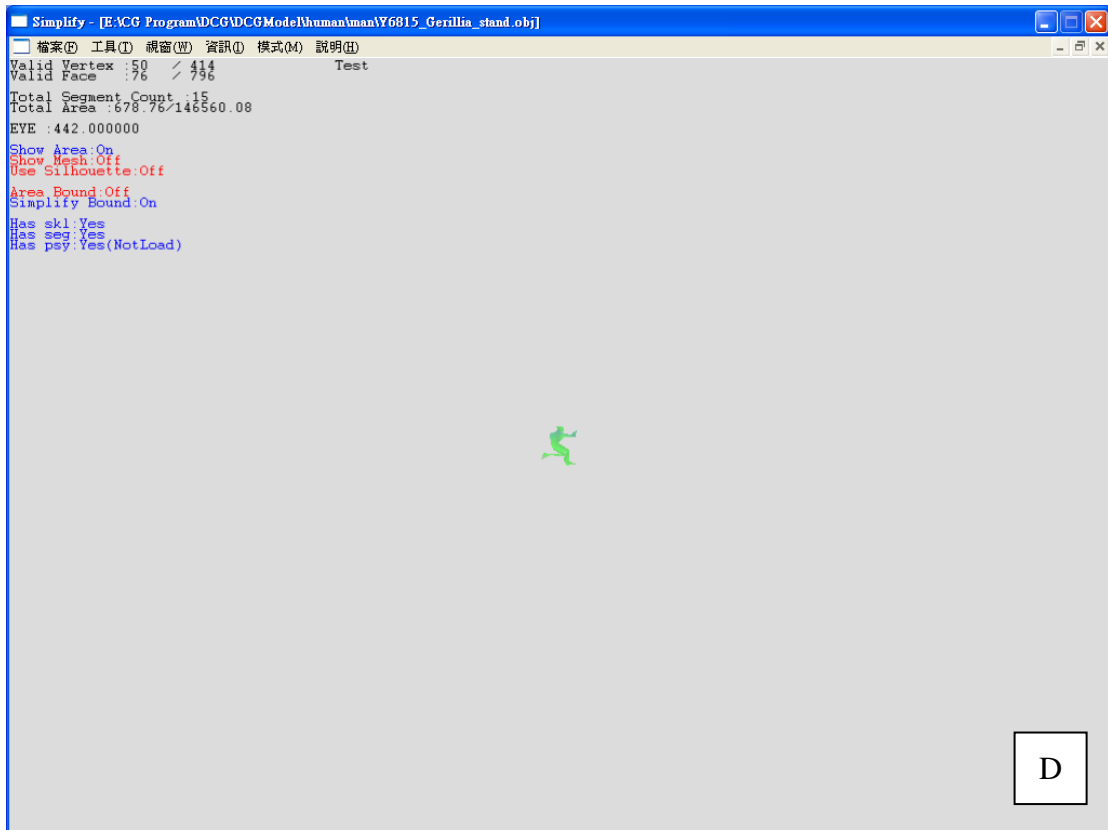
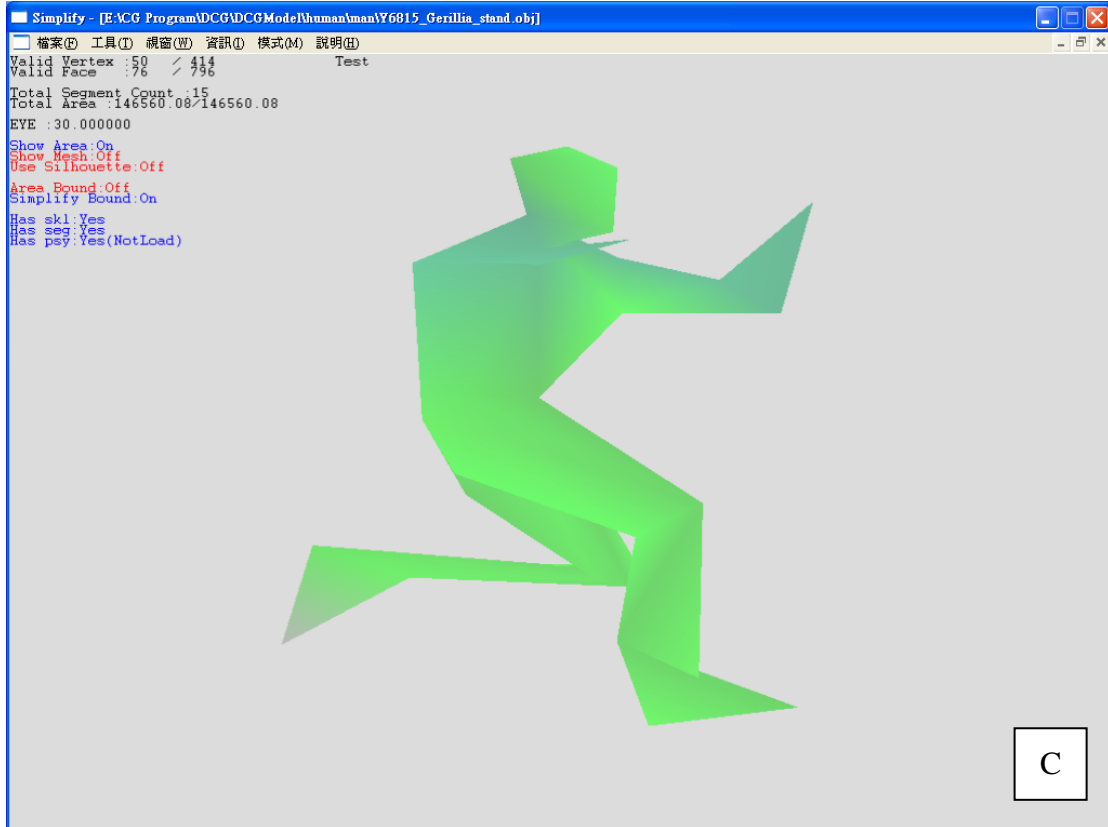


圖 10 : Man 模型隨視點到模型的距離而簡化

第四章 研究結果

表一、本研究的實驗的平台：

CPU	Intel Pentium CPU 3.2GHz
RAM	2GB
顯示卡	NVIDIA GeForce FX 5900XT
作業系統	Microsoft Windows XP pro
開發平台	MS Visual C++ .Net2003 +DirectX 9.0c

研究結果依產生順序可分為：

1 **DCG 骨架生成結果**：骨架生成後的資料包函的資料結構有 end points、connection points 和 joint points 三種主要連接點座標以及骨架連接點之間線段的點座標和模型依骨架分段的分段(segments)，如圖 11 至圖 13。



圖 11：輸入模型 (man & woman class)



圖 12：模型骨架

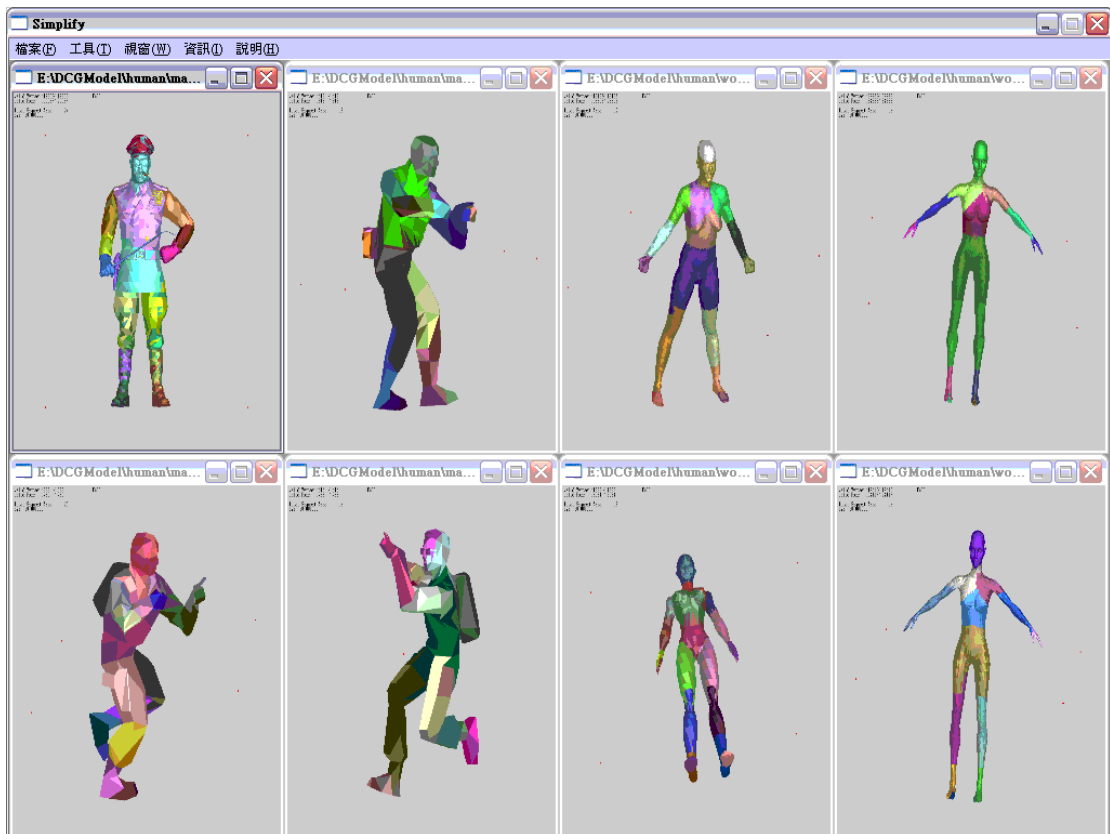


圖 13：模型依骨架的分段

2 原型生成結果：原型生成主要的資料有代表性骨架的骨架座標、模型分支數和模型各分支的粗細和長度在常態分佈的資訊，模型在讀入時自動依骨架截取出模型的分支(branches)，但要將分支的各個意義做分類目前還需要以手動的分式，因此需將模型分支分類後產生一組分類資料後才可順利將原型生成，分類視窗如上圖 2，將模型分類好之後即可產生原型，如下圖 15。

建構原型的方法，我們使用模型上的頂點，首先產生抽象化的結果，來決定原型的基本幾何形狀。接下來再依照產生原型的數據，調整上述的基本幾何形狀，以便符合原型的特性。利用此方式來取代過去的原型構成(圖 14)的好處，是讓原型有體積的數據，可以當做未來原型和原始模型抽象化結果的分析。

目前我們的研究，只完成模型的抽象化結果，而利用原型的數據調整抽象化幾何圖形的部分還未完成。表二為原型以讀入人類類別中的模型做計算，所計算出的原型資訊。

	Referring level	Radius average	Radius deviation	Radius maximum	Radius minimum
main body	1	0.0557273	0.023851	0.085422	0.026989
head	1	0.485804	0.320716	1.22263	0.13061
left hand	1	0.346429	0.0708726	0.447249	0.229835
right hand	1	0.34721	0.122449	0.524394	0.113264
left foot	1	0.516251	0.142559	0.743563	0.325953
right foot	1	0.498876	0.158239	0.722533	0.239932

表二、人類群組原型資料表

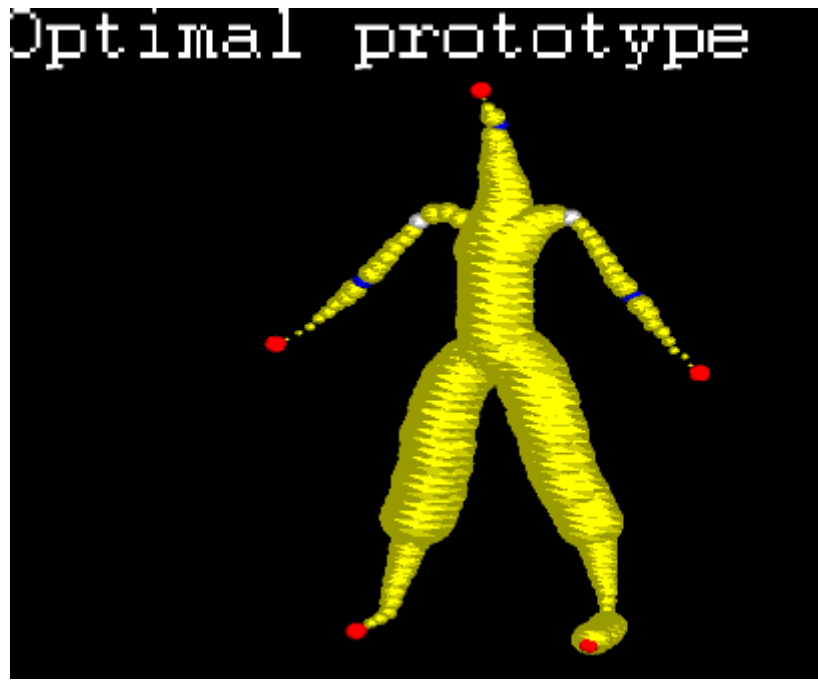
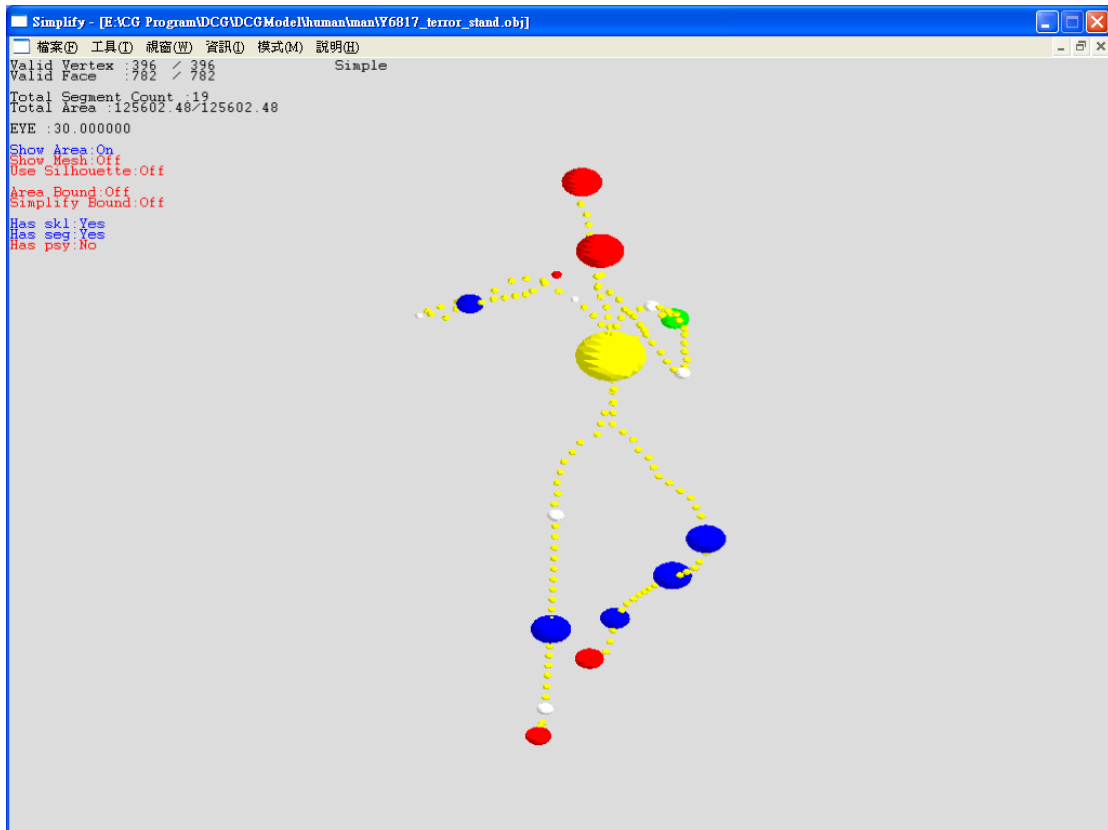
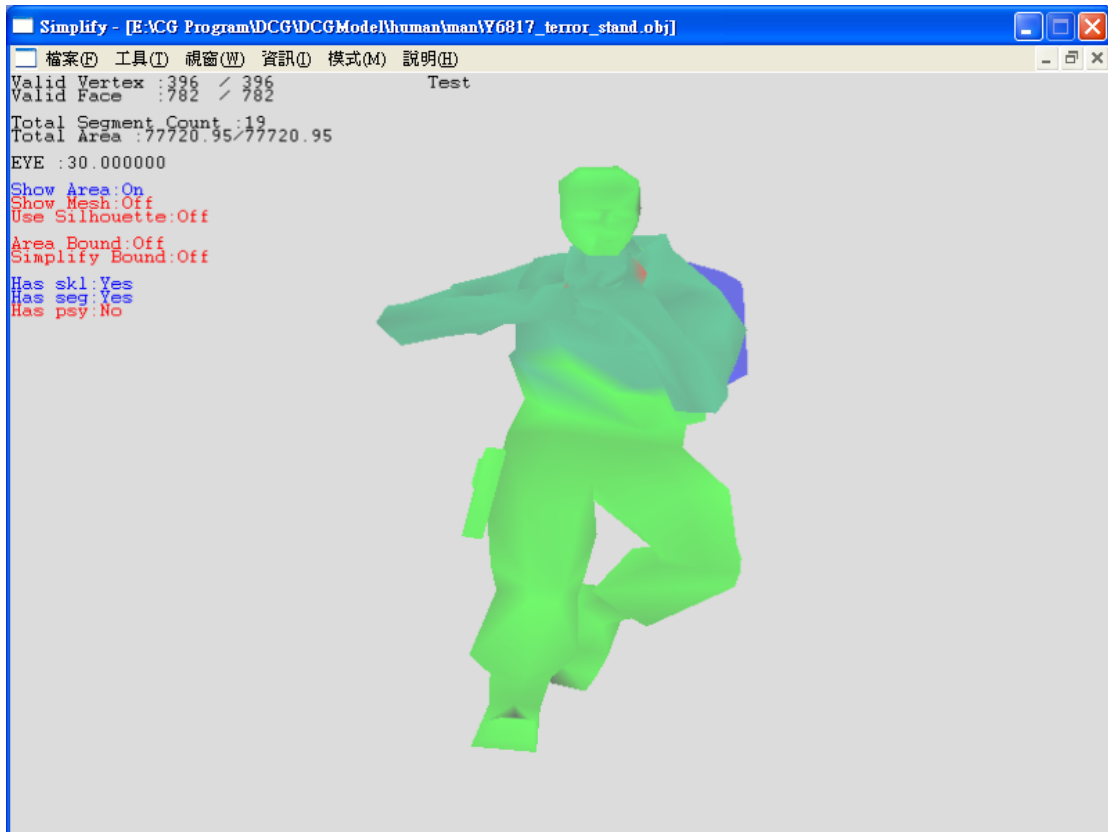


圖 14：以很多球構成的過去研究結果



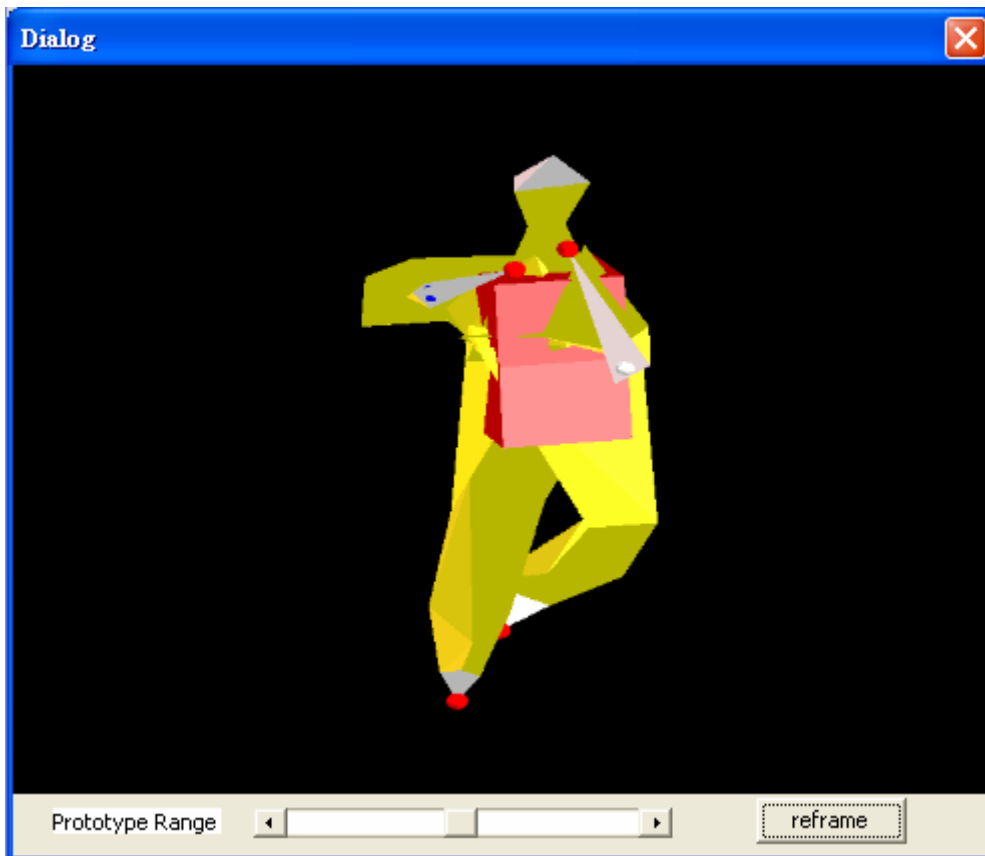
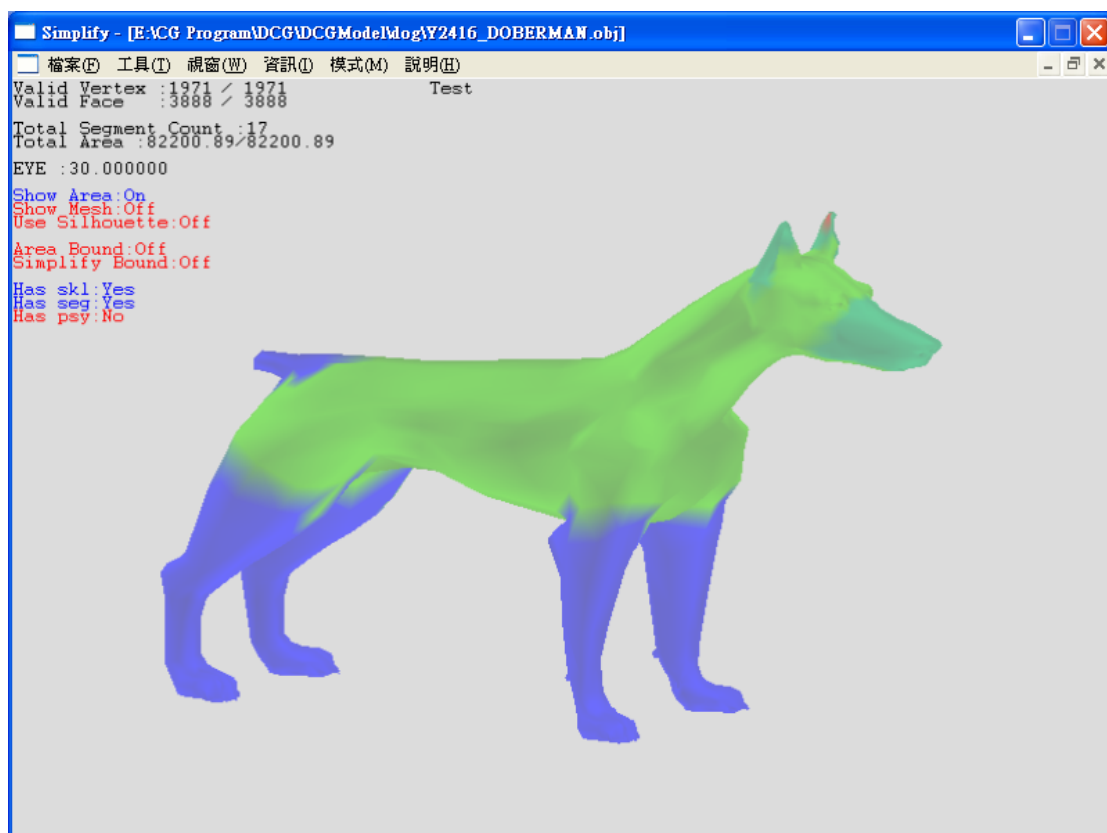


圖 15：Human 模型、骨架及其抽象化結果



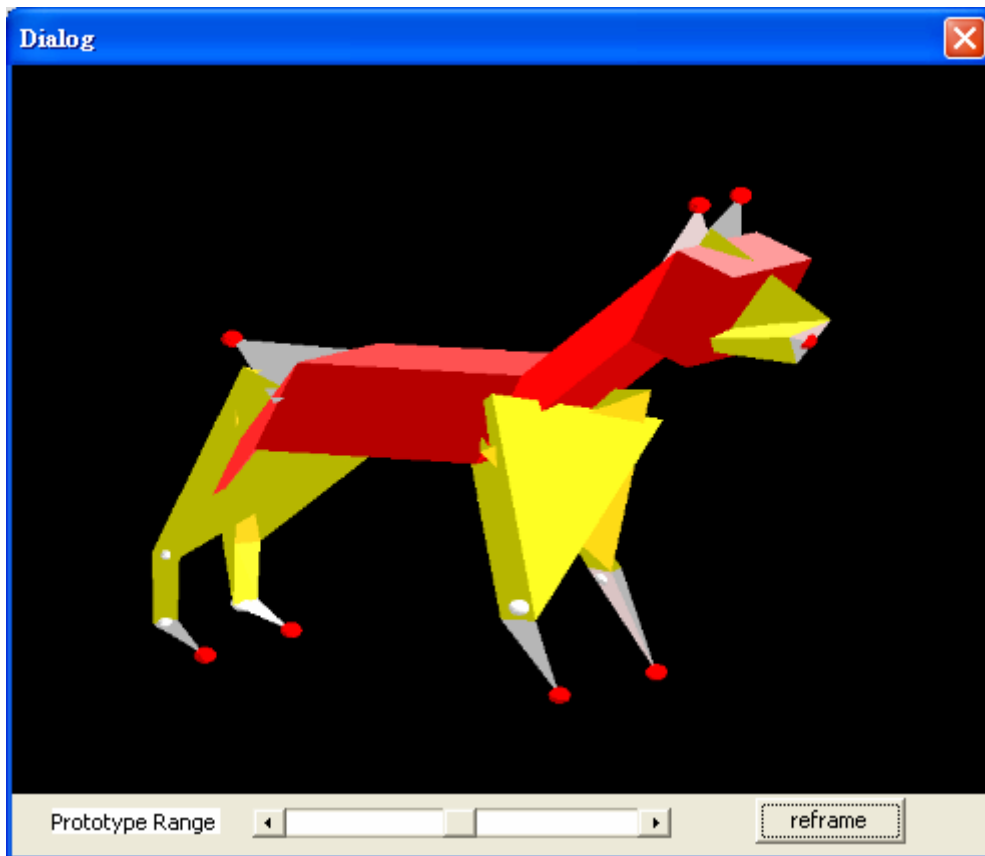
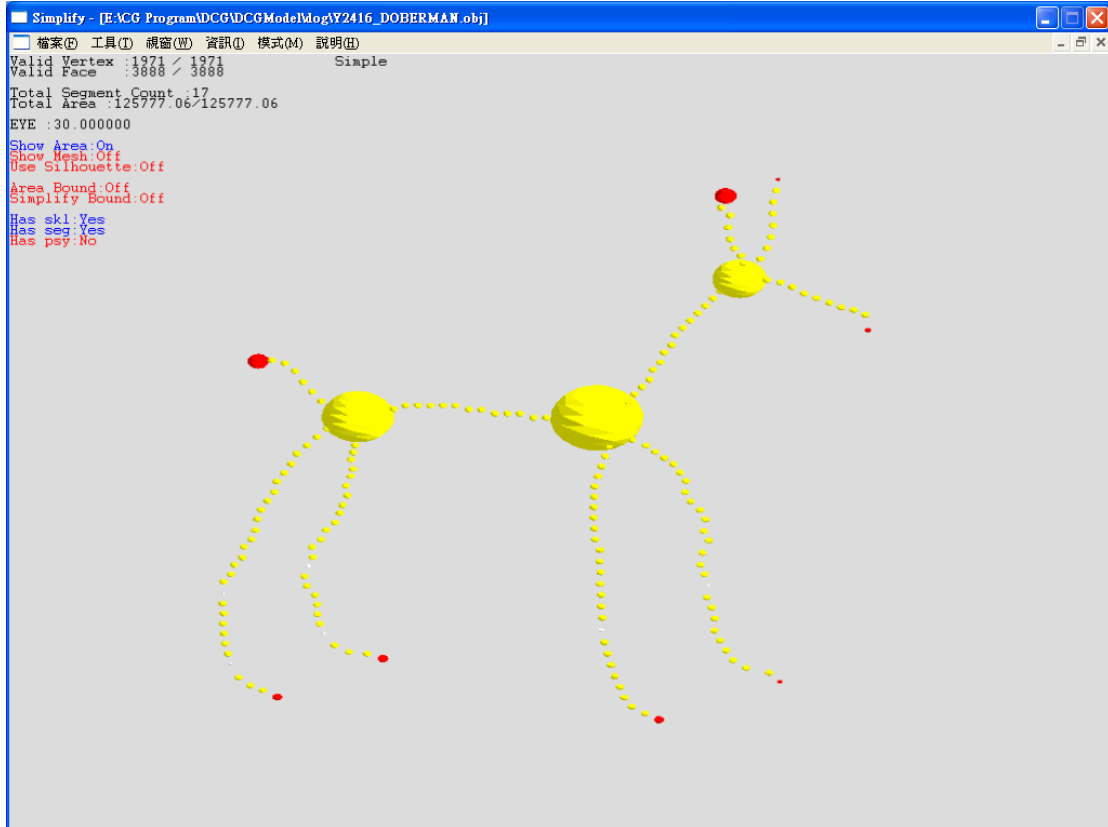
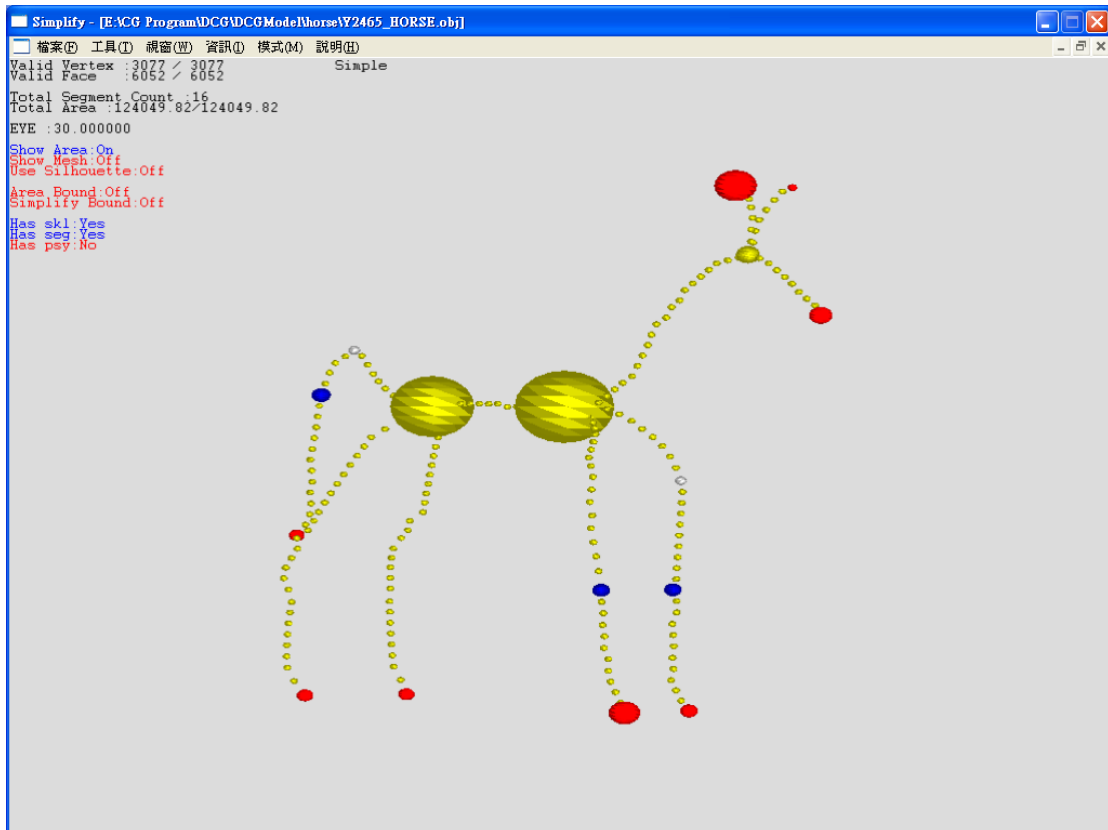
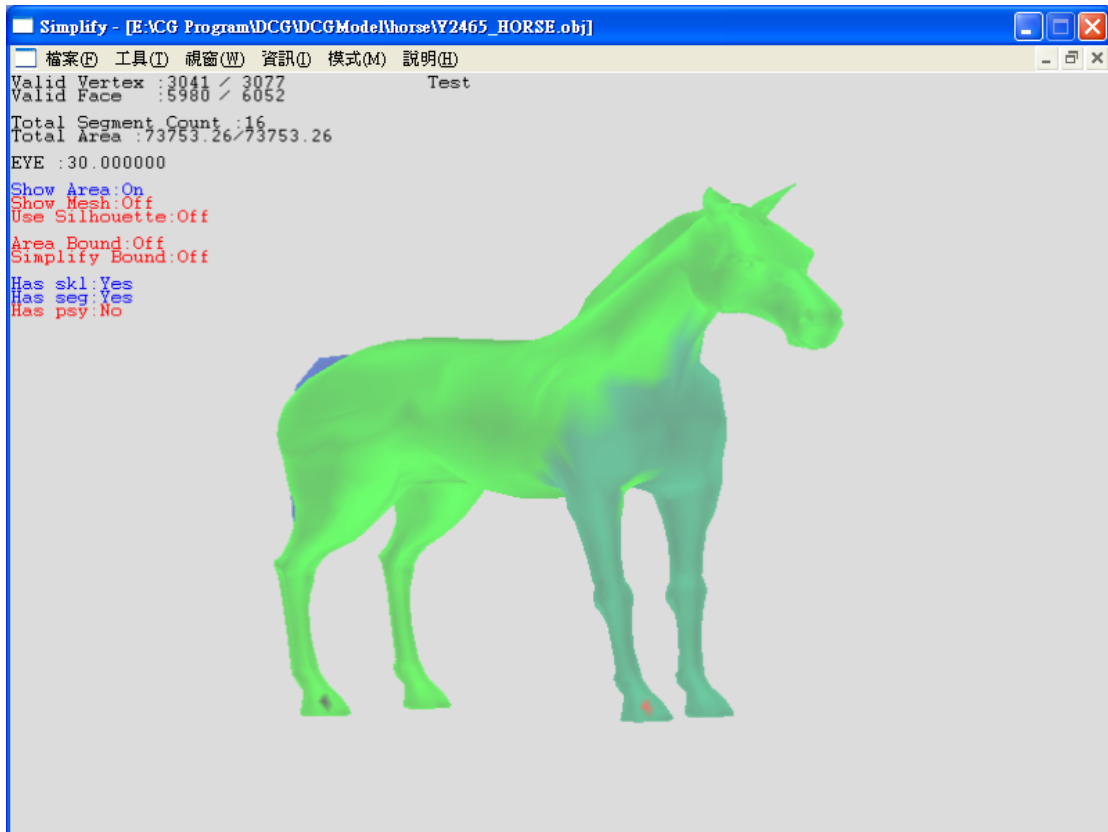


圖 16 : Dog 模型、骨架及其抽象化結果



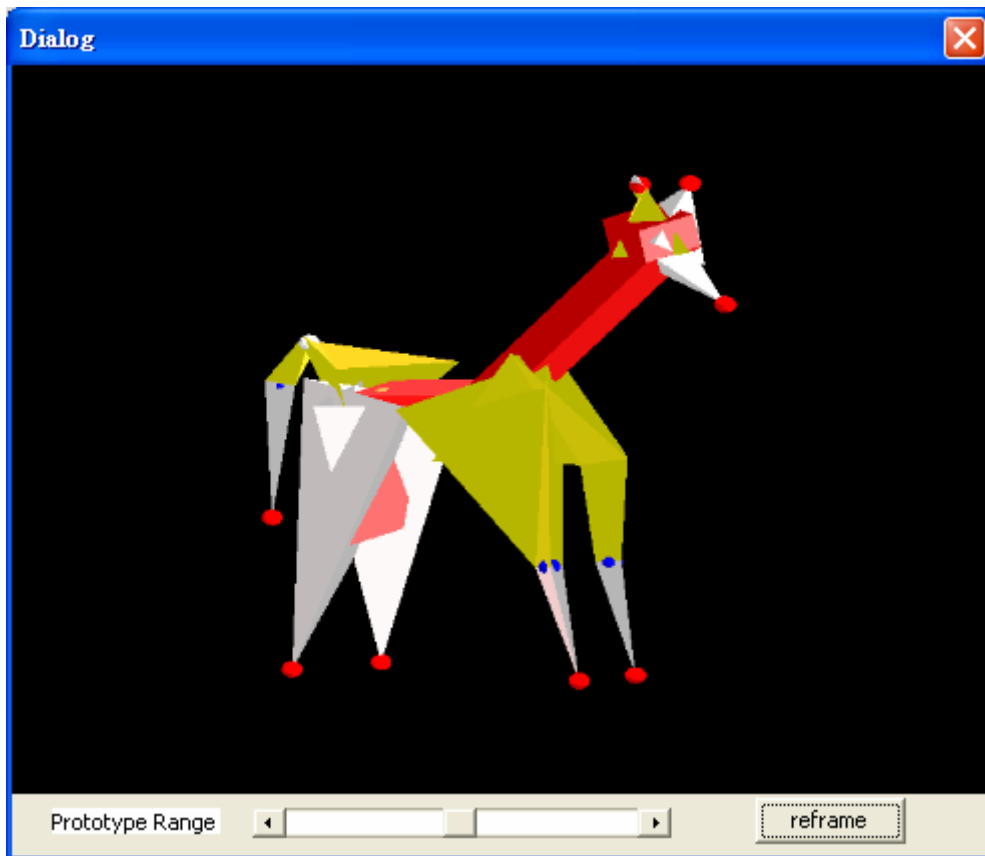
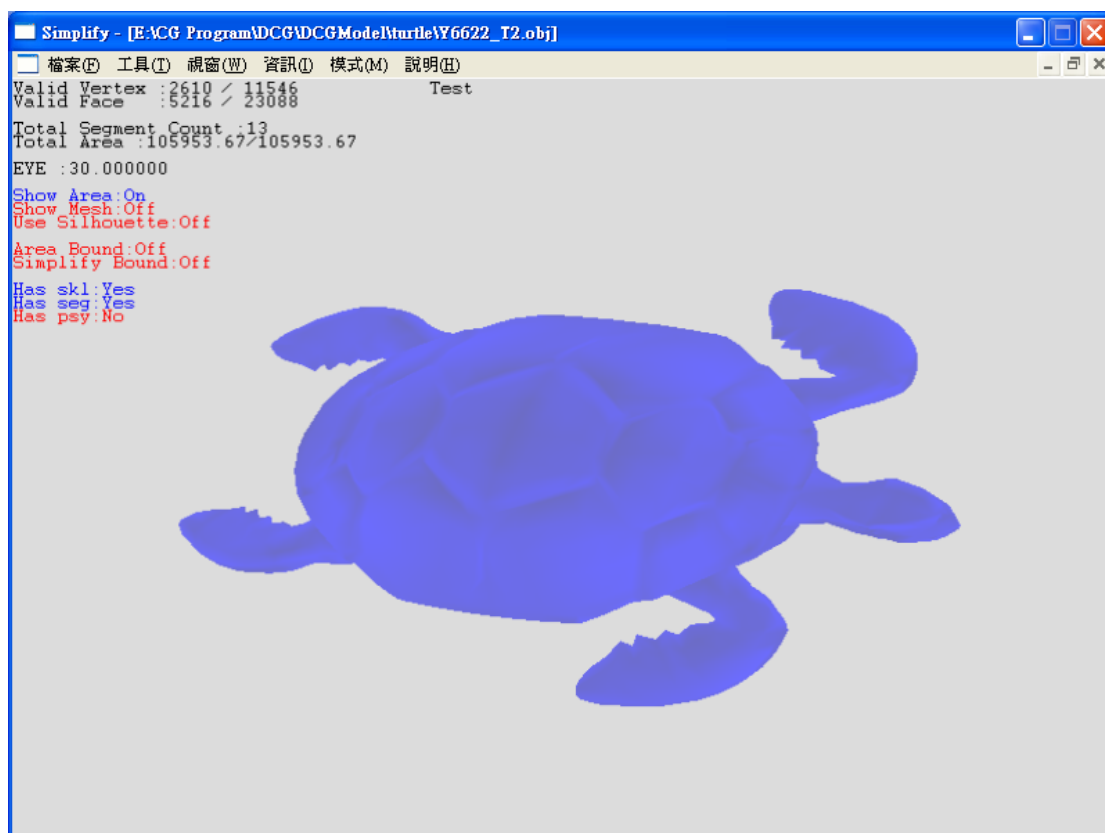


圖 17：Horse 模型、骨架及其抽象化結果



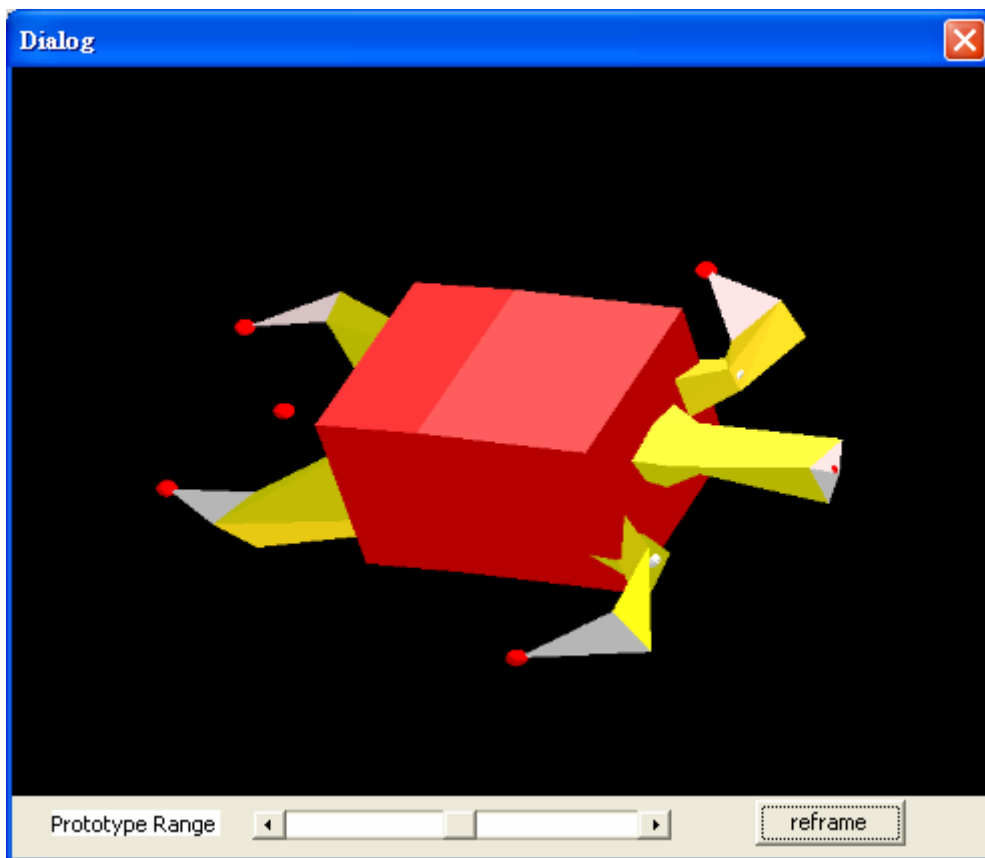
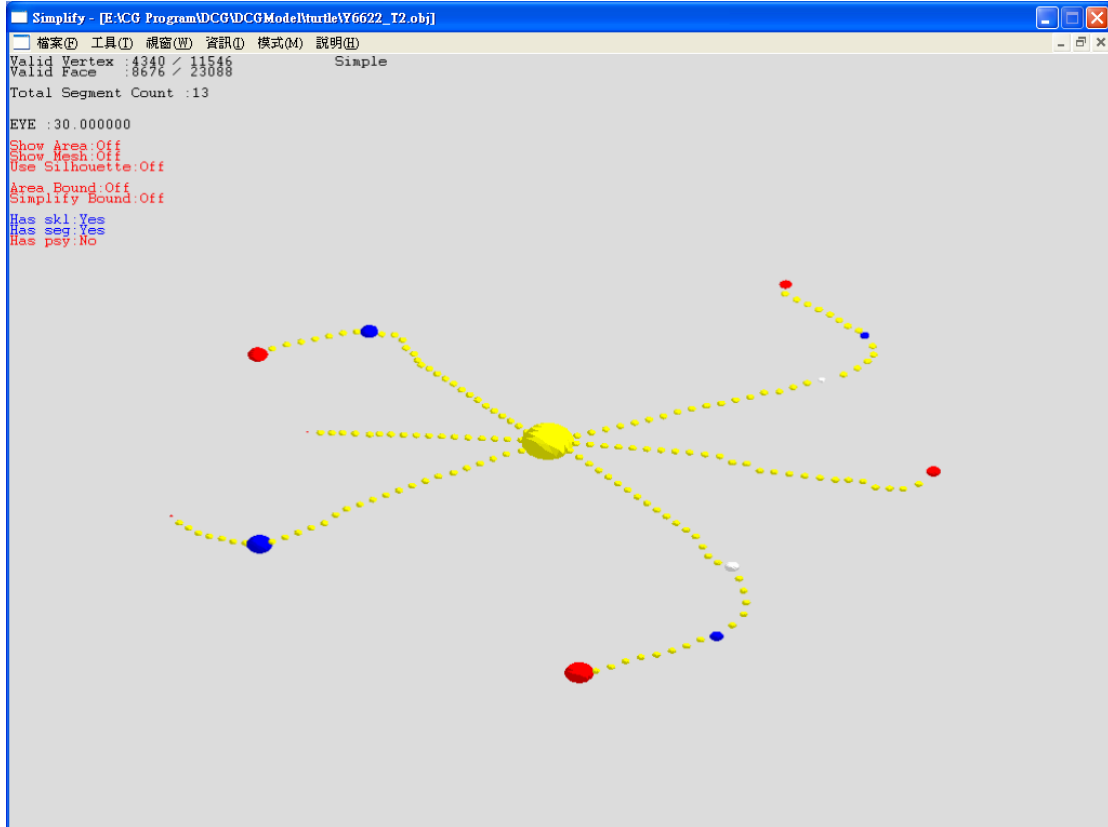
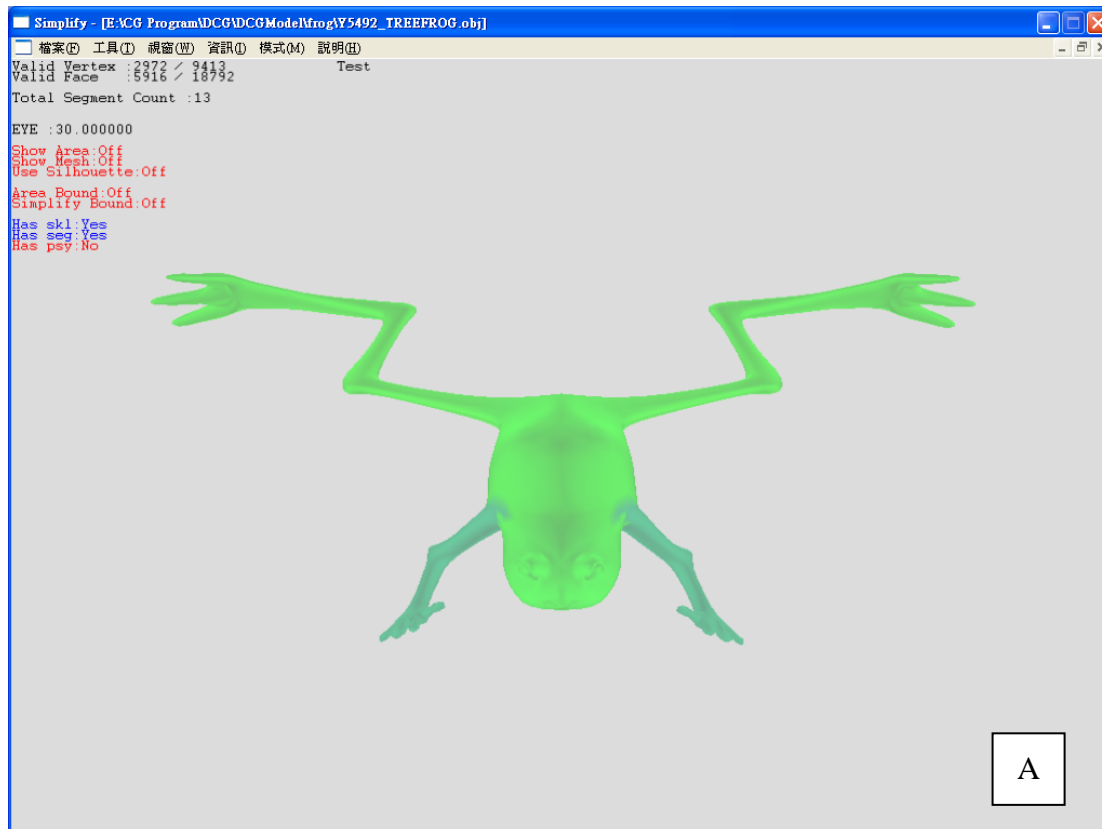
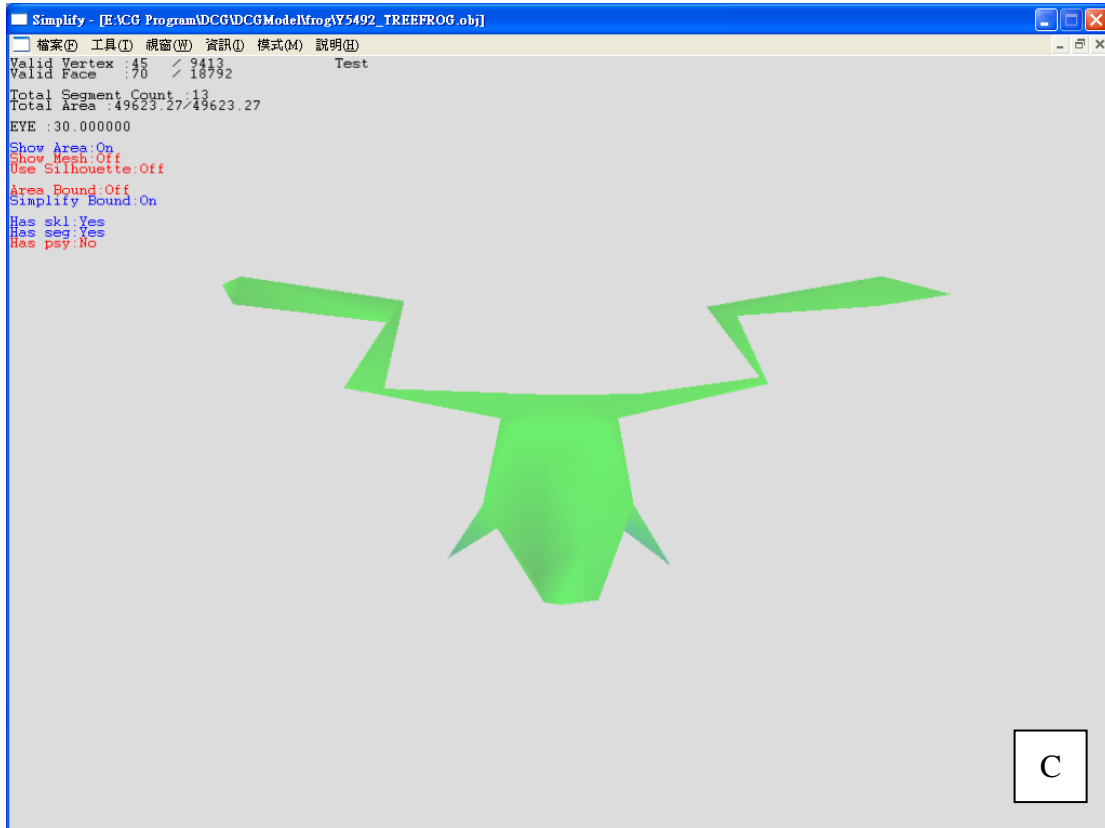
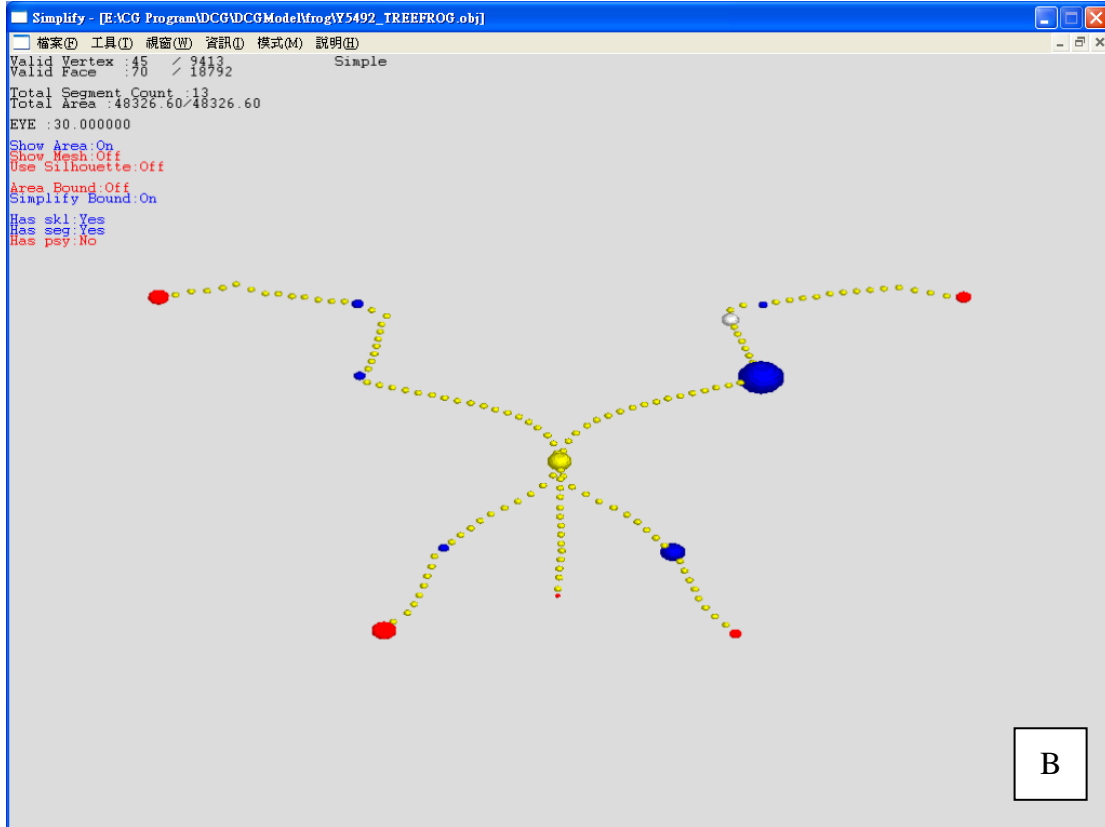


圖 18：Turtle 模型、骨架及其抽象化結果

3 隨視點變化而簡化的成果：影響模型簡化的參數為模型投影在螢幕所佔的面積，而影響面積變化的則是視點到模型座標系統中心點的距離，因此模型將會依照使用者改變視點距離而改變模型面積，進而影響模型簡化的程度。除了上圖 10 之 man 模型的例子外，我們再以 frog 和 horse 模型為例子來展示。





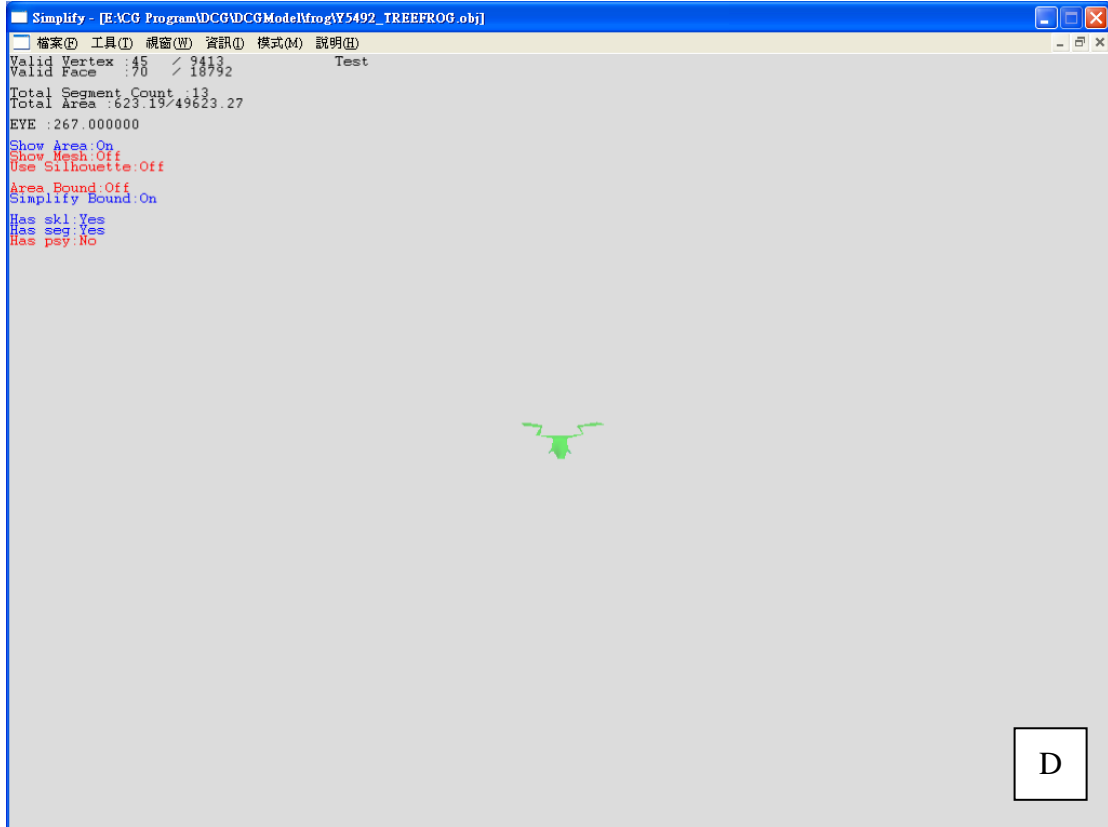
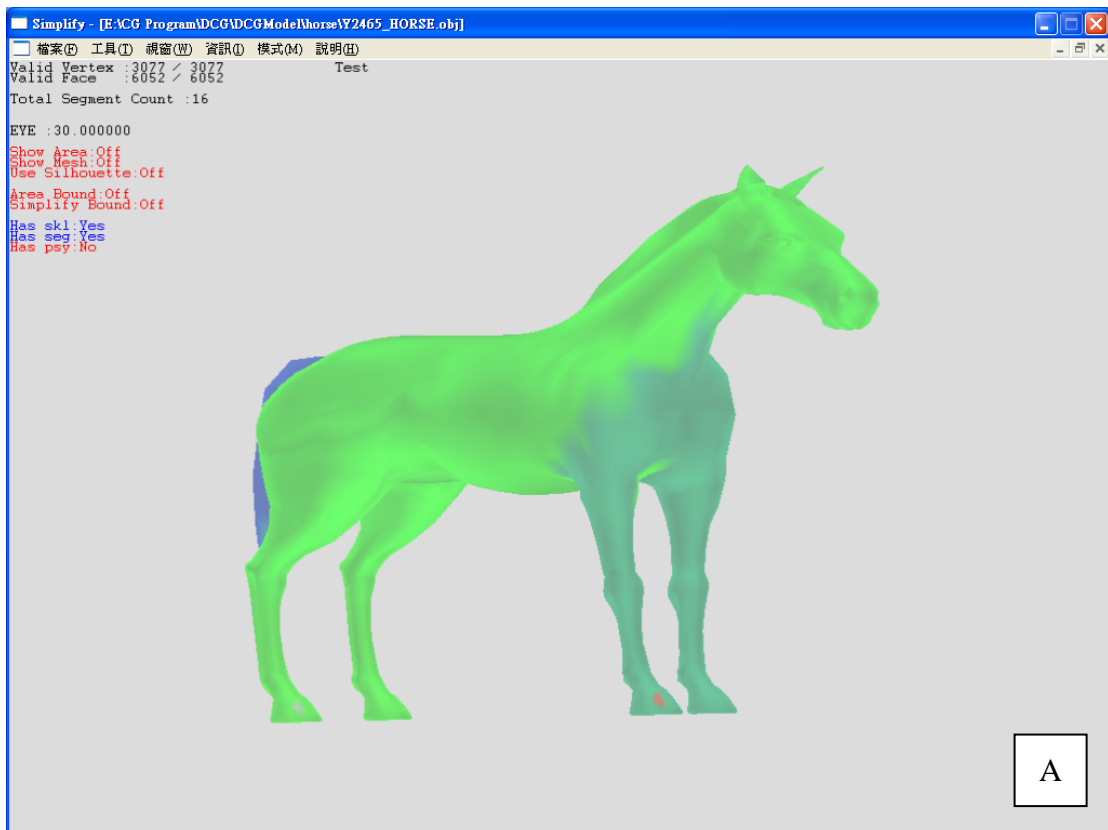
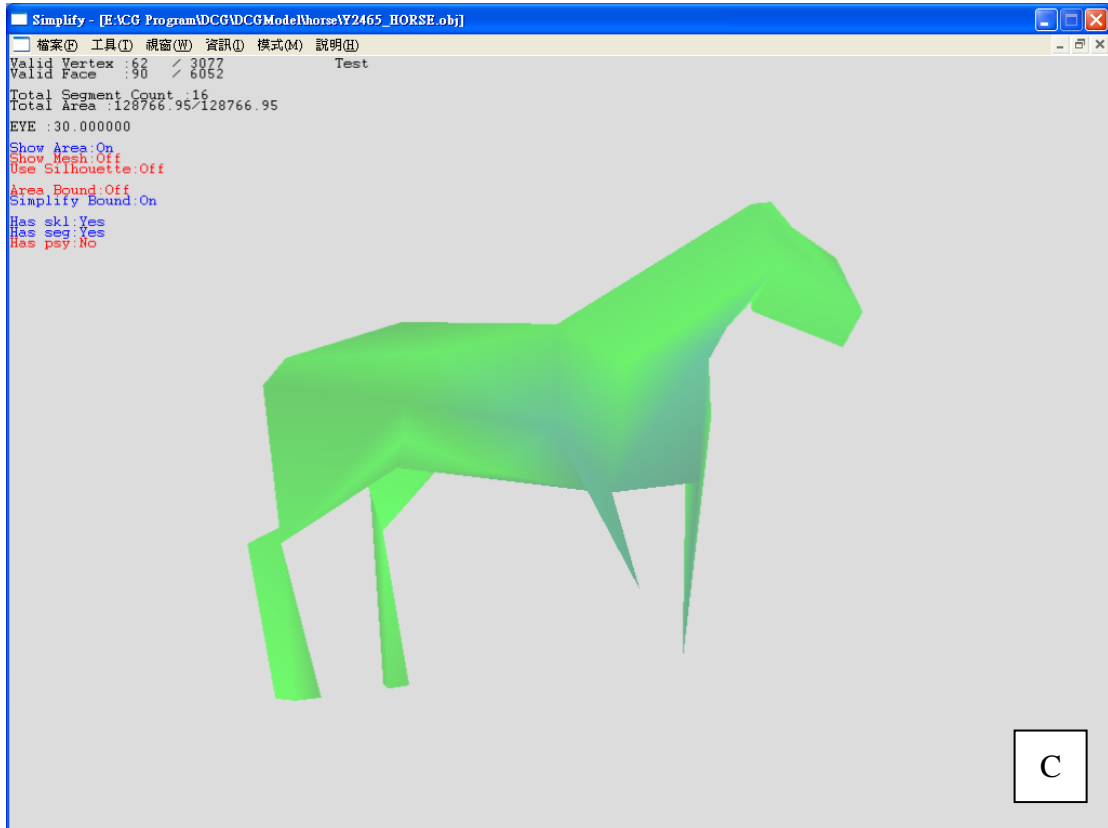
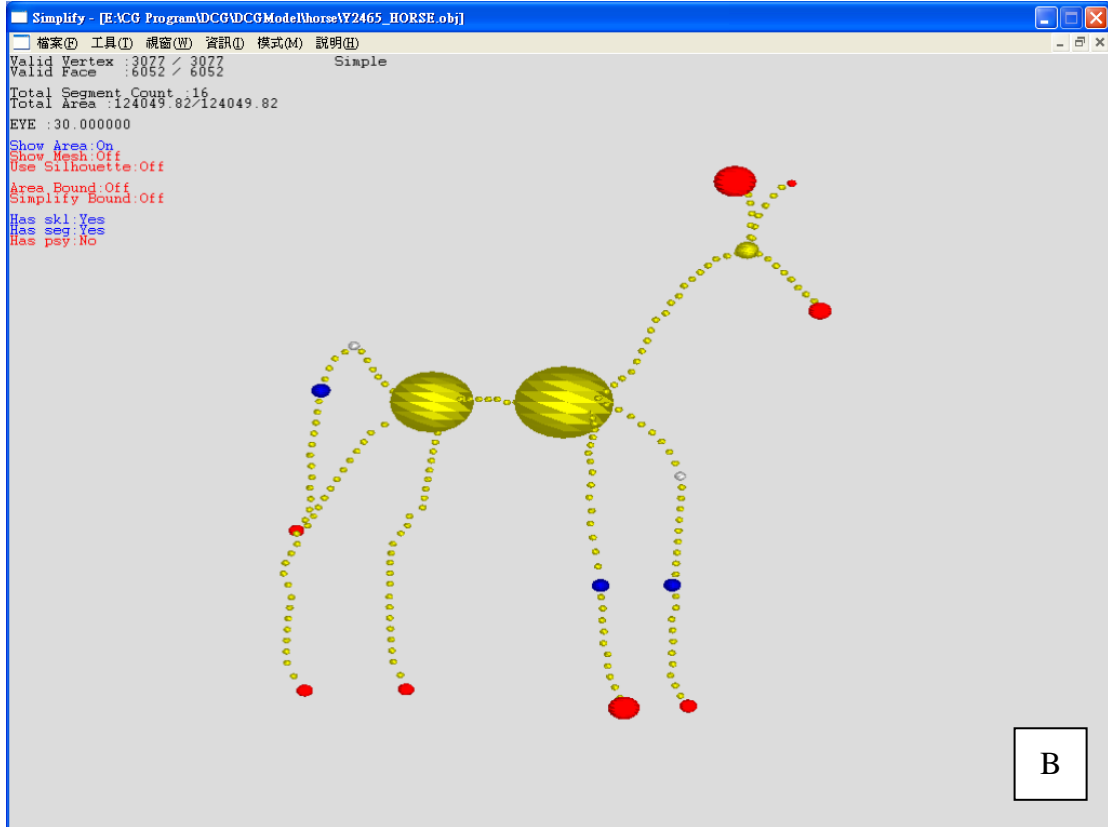


圖 19：Frog 模型隨視點到模型的距離而簡化





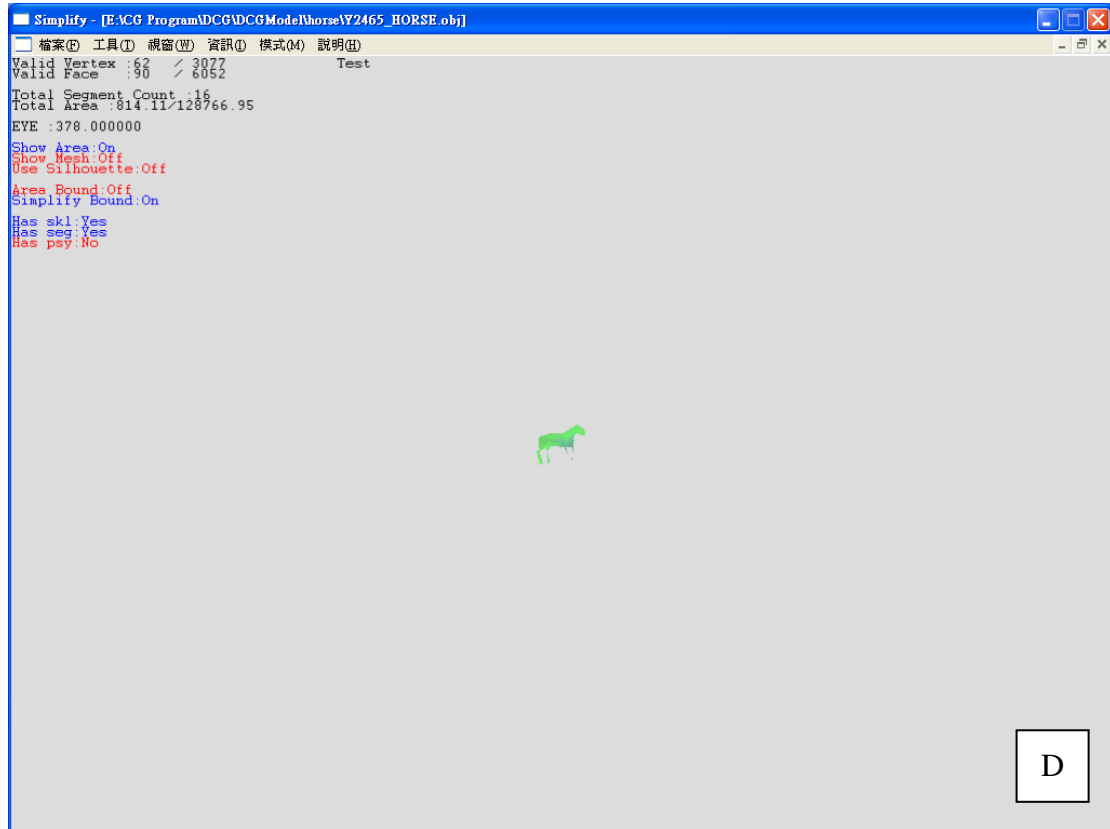


圖 20：Horse 模型隨視點到模型的距離而簡化

4 Demonstration

連續的簡化結果請參照 Demonstration movies。

Demonstration movies 可以在以下網址下載

<http://cgda.csie.ncnu.edu.tw/gallery.htm>

第五章 結論

本研究將 HPS 導入到 LOD SYSTEM 中，使的過去考慮幾何、材質等等的出發點，擴大到更符合人類來觀測並辨識 3D 物件時的模型簡化。之後我們賦予傳統的 QEM 手法，在給定各頂點權重時，具有更自然、更合理的策略。本研究還更進一步提供一個系統化的工具，幫助使用者分析 3D 模型的特性，進而建立同一種類模形的原型。透過原型，可以決定 LOD 合理的最小解析度。同時，也可以作為將 3D 模型抽象化的一個方向。

本研究的主題以感知的方式為基礎實作模型的細緻層級(LOD, level of detail)，所謂細緻層級為將模型依其在環境中的改變(如和視點的距離)而將模型以減少模型面數和精細度，來達到節省運算和儲存資源的技巧，細緻層級的實作有許多種方式，本研究採用連續式 LOD 的演算法來進行實作，QEM(Quadric error metrics)演算法為代表性的 LOD 演算法，以幾何圖形的特性來對整個 3D 模型做平均的連續性簡化。就感知的層面，主要有三個部分，第一部分，模型的分段是由使用者半手動方式分類到認知的群組，例如兩隻手的分段分配到手的群組、頭的分段分配到頭的群組，依照使用者視覺認知的分配後，集合所有同類型已分類好的模型，最後即形成一個經認知分類而產生的原型(Prototype)。第二部分，原型的成像我們是以三角錐來構成模型分支的尖端部份，然後以三角柱來構成整個分支，最後再以立方體來構成整個身體；依此成像的效果，希望可以讓使用者把此簡單圖形構成的模型，認知成原來的模型，因此即為我們原型的定義。第三部分，模型隨著視點拉遠會因投影在螢幕的面積縮小而簡化，其中視點最遠距離的限制，為模型呈現在視點正面的面數乘以 4.5pixel，此值的理由為，人類對電腦螢幕呈現圖形的面積，若要認知成三角形面，則最少要 4.5pixel。

雖然我們利用 HPS 來決定模型的原型和簡化，但是視覺認知很容易有主觀意識，隨著使用者不同，可能會有不同結果。針對以上三個認知問題，第一部分，模型的分類，以四隻腳的動物為例，有些人會把四隻腳設為同一群組，有些人則

分成前後腳，此兩種分類法的結果不相同，此為主觀認知的不同。第二部分，由簡單圖形所構成的原型，極有可能會有不同動物但原型相似的結果，以同為四腳著地的哺乳類動物狗和虎為例，其原型結果很相似，因此認知此原型為原來模型的可能性就變很主觀。第三部分，我們定義一個三角形面呈現在電腦螢幕能被辨認成三角形的最小 pixel 面積為 4.5pixel，不過因人的視力不同，此值可能過小，而無法被辨認成三角形，所以帶有主觀結果。

Prototype 的部份，目前 prototype 的數據只有 radius 和長度，因此還必須考慮可能的參考數據加入到裡面，讓 prototype 的數據更完善；Prototype 的抽象化結果，目前的構成方式還未必能夠被辨認成原來的模型，而且 dcg 產生的骨幹資訊有很多各式各樣的案例，因此我們還需要改善 prototype 的構造和方法，讓抽象化結果更符合當下的模型。

第六章 參考文獻

- [1] Wolfgang Heidrich, Philipp Slusallek, Hans-Peter Seidel, “Real-Time Generation of Continuous Levels of Detail for Height Fields”
- [2] Javier Lluch, Emilio Camahort, Roberto Vivó, “An Image-Based Multiresolution Model for Interactive Foliage Rendering”
- [3] Robert L. Solso 著; 梁耘塘編譯, “視覺藝術認知: Cognition and the Visual Arts”
- [4] Reddy, M. Perceptually Optimized 3D Graphics. IEEE Computer Graphics and Applications. Vol.21(5).pp.68-75.2001.
- [5] Fu-Che Wu, Wan-Chun Ma, Rung-Huei Liang, Bing-Yu Chen, Ming Ouhyoung, “Domain Connected Graph: the Essential Skeleton of a 3D Shape”,
<http://graphics.csie.ntu.edu.tw/DCG/>
- [6] [1997ACM] Model Simplification Using Vertex Clustering. Kok-Lim Low+ and Tiow-Seng Tan+
- [7] VTK Web Site: <http://public.kitware.com/VTK/what-is-vtk.php>
- [8] M. Garland, P.S. Heckbert, “Surface Simplification Using Quadric Error Metrics”, SIGGRAPH '97 Conf. Proc., T. Whitted, ed., Page(s):209-216, Aug. 1997.
- [9] Hoppe, H. H. New Quadric Metric for Simplifying Meshes with Appearance Attributes. Proceedings of IEEE Visualization '99. pp.59-66 1999.
- [10] Hoppe, H. Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering. Proceedings of IEEE Visualization '98. pp.35-42. 1998.
- [11] Lindstrom, P and G Turk. Fast and Memory Efficient Polygonal Simplification. Proceedings of IEEE Visualization '98. pp.279-286. 1998.
- [12] D. Brodsky, B.A. Watson (1999). R-Simp: model simplification in reverse . Graphics Interface 99 conference poster abstracts, 39-40 (London, Canada, June).
- [13] Wolfgang Heidrich, Philipp Slusallek, Hans-Peter Seidel, “Real-Time Generation of Continuous Levels of Detail for Height Fields”, Graphische Datenverarbeitung

(IMMD9)

[14] Ming Fan Ueng and Jung Hong Chuang, "Terrain Rendering With View-Dependent LOD"

[15] Takahito Tejima and Yasushi Kuno, "Incorporating "Level Of Detail" into Programming Languages Aimed at Real-Time Application"