

使用方式：

程式執行檔：draw.exe

輸入檔案名稱：data.txt

輸出檔案名稱：Output.ppm

檔案輸出格式：2560 x 2048 ppm 檔

這次作業讀檔、畫直線、三角形的部分主要是由第一次作業衍生出來的，基本架構維持和第一次作業的一樣。不同的地方是，在畫圖前會把圖形作旋轉，分成四張圖片輸出到同一張圖檔上。分別是從上方看，側面看，旁邊看，還有從正面看。

作法是將所有三角形的點經由旋轉矩陣，首先對軸做旋轉，再對 x 軸做旋轉。如此就可將圖形轉成任何角度。上方的角度就對 x 軸旋轉 90 度。側面則是對 y 軸逆時針旋轉 90 度。

而在畫三角形的時候，先以 back face culling 將背面的三角形拿掉，然後再搭配 z buffer 將圖畫到 buffer 上。每片三角形的顏色則是將法向量與平行 z 軸的向量作 dot Dot 完以後放大 100 倍再加上 100，如此垂直 z 軸的三角形則會最亮。

直線方面，一樣是將圖形作旋轉之後輸出到 buffer 上，一樣以 buffer 來判斷線段的前後順序。而 line 的顏色則是以 rander 表示。

在讀檔的時候把所有點讀入 link list 後，要再讀入每個三角形的三點座標，此時需要去 search 之前的 link list 去把點抓出來。但在跑 Dragon (VRIP)這個 module 的時候，因為總共有 80 萬片三角形，幾乎 40 萬個點座標。如果以 linear search 的方式來搜尋的話，running time 為 $O(n)$ 。所以我將存座標的 link list 每 1000 個 node 把位址存到一個 index 裡，如此每次只要從對應的 index 點開始搜尋，worst case 為 1000 次，如此 running time 則減少為 $O(1)$ 。比對兩種方法所花的時間，沒有用 index 的時候，以 AMD Athlon 64 X2 3800+的 cup，跑將近三小時還跑步出圖。使用 index 後，30 秒內即可完成。

以下圖形為各 Models 跑出的結果，with line 為在輸出三角形時，每行第一個 x pixel 會圖成藍色，如此就可明顯看出 model 的三角形架構。附檔 draw3_line.exe 則為此執行檔

附檔說明：

draw.c hw 2 source code

draw.exe hw 2 執行檔

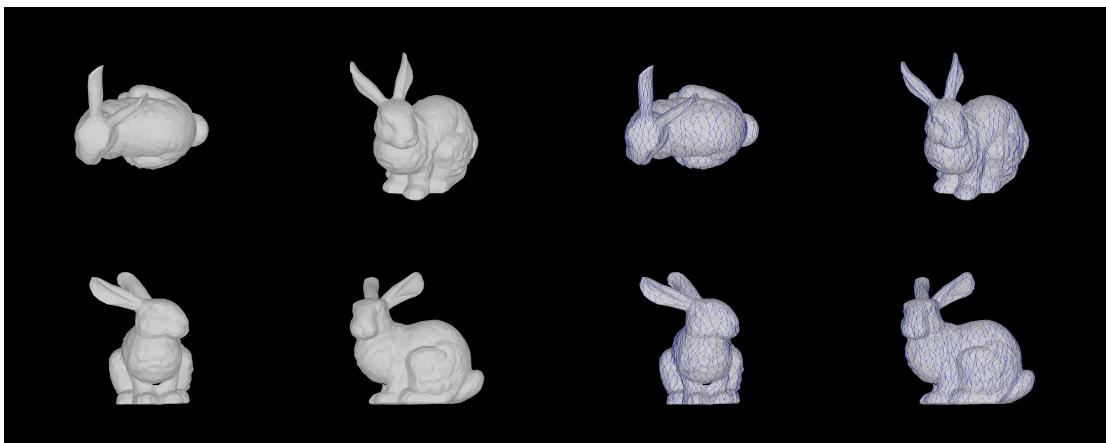
draw_3D_line.exe 輸出三角形外邊的執行檔

hw2 reports.doc reports

圖片：

Bunny

With line



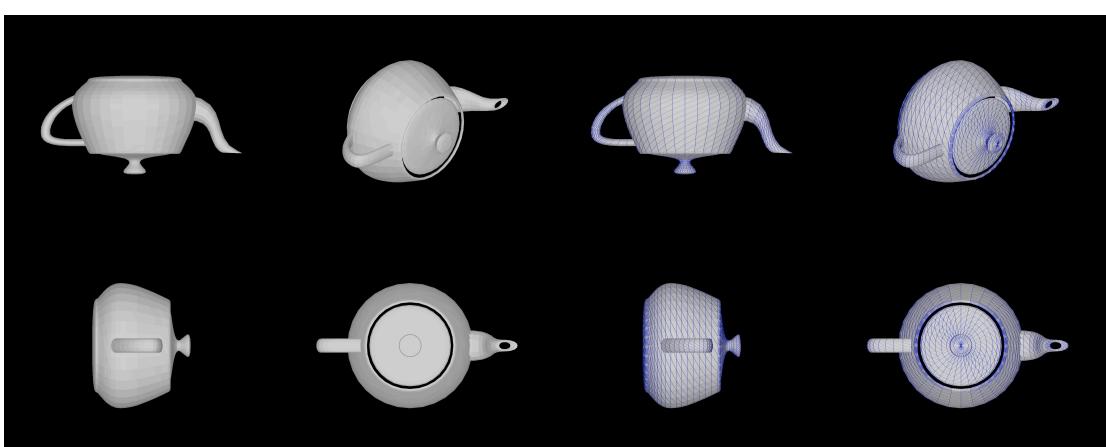
Dragon

With line



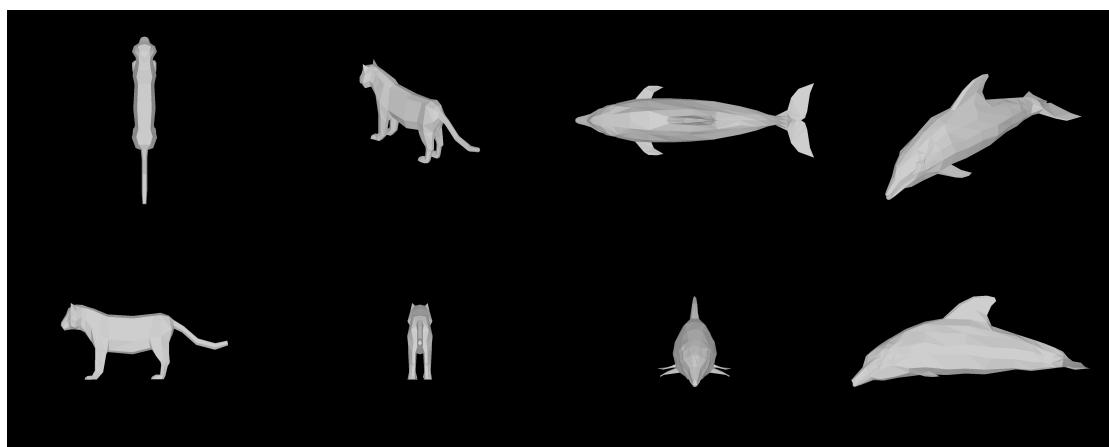
Teapot

With line



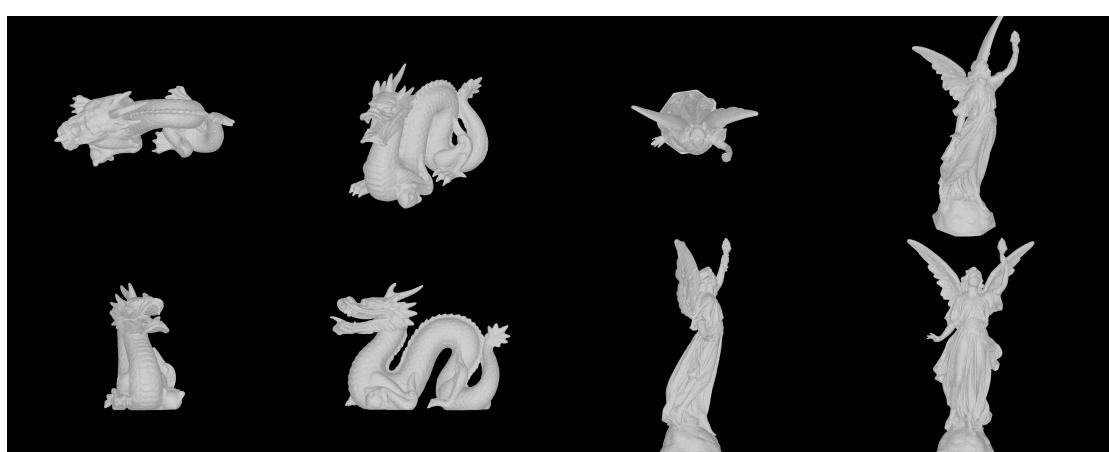
Tiger

Dolphin



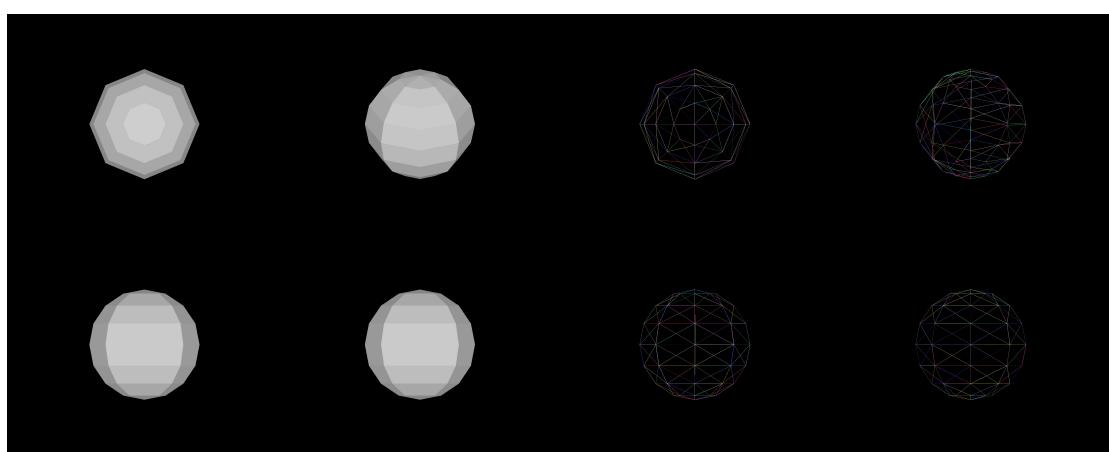
Dragon (VRIP)

Lucy



Sphere

Sphere line



Triangle

Line

