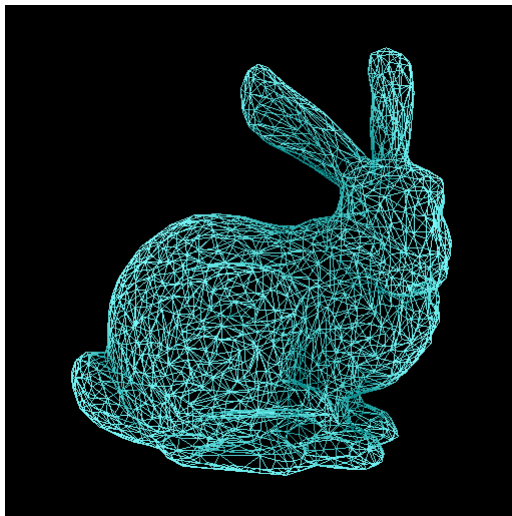


3DMM 2014

# Assignment #1: Rasterization

# Requirement

- You are required to write a C/C++ program that can accept triangles input vertices.
- The program should include the line and triangle rasterization algorithm mentioned in the class.



*Wireframe mode*



*Solid mode*

# Requirement

- Use orthogonal projection for ease of viewing projection.
- Use back-face culling to improve the performance.
- Use Z-buffer to correctly remove hidden surfaces.
- Draw models with color assigned. If the model does not provide any vertex color information, then assign the vertex depth as the color of corresponding vertex.
- D3D and OpenGL APIs are not allowed in this assignment.
- Set the frame size to **512x512** for each model.

# Input Format

- Triangles

```
Vertices                                     // Vertex Identifier
(no. of vertices)                          // Vertex count
x0 y0 z0
x1 y1 z1
x2 y2 z2
...
Colors                                     // Optional vertex color identifier. It could be color_rgb or color_rgba
(no. of vertices)                          // Vertex color count
r0 g0 b0                                  // Example of color_rgb
r1 g1 b1
r2 g2 b2
...
Triangle_list                             // Primitive identifier. It could be triangle_list, triangle_strip, or triangle_fan
(no. of triangles)                         // Triangle count
1 3 2                                     // First triangle
0 1 2                                     // Second triangle
0 2 3
7 6 5
7 5 4
...
```

# Output Format

- Portable pixmap format(\*.ppm)
  - [http://en.wikipedia.org/wiki/Netpbm\\_format](http://en.wikipedia.org/wiki/Netpbm_format)
  - Viewer: XnView, IrfanView...
- Example:

```
P3           // ppm file header for ASCII color format
300 200      // Image width, height
255         // Max intensity
12 34 45 33 127 42 98 209 125 ...
           // R1 G1 B1 R2 G2 B2... interleaved
```

# Hints

- Not every model has the same scale. You have to **normalize** first and then apply some transformation (scaling, translation, rotation) for better visual result.
  - Make sure the object faces forward.
  - Translate the model center to proper location, such as image center.

# Report

- Describe your rasterization algorithm.
- Describe your parameters used in transformation for each model.
- Evaluate performance of rasterization with/without culling.
- Show your rasterized results using wireframe mode and solid mode. (12 models \* 2 modes = 24 images)

# Submission

- **Deadline: 2014/3/31 11:59 pm**
- Compress all required files as a single ZIP or RAR file.
  - **Naming rule:**  
3DMM\_HW1\_R02943001.zip  
3DMM\_HW1\_R02943001\_v2.zip
- **FTP:**
  - **140.112.174.82**
  - Port: **2021**
  - Account: The same as the one used in the course website.
- **Required files**
  - Report, in **PDF** format
  - Source code (C/C++)
  - A readme, to describe how to execute your program.
- **Any further question, please contact TA**
  - 塗偉志 [wctu@media.ee.ntu.edu.tw](mailto:wctu@media.ee.ntu.edu.tw)