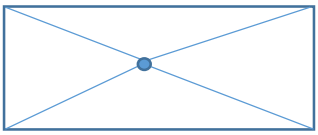
Lab Exercise Date 14-03-2019

1. Write a C program to implement iterative Gouraud shading algorithm for a polygon surface.



2. Perform orthographic projection onto y=0, z=0, and x=0 for a cube centered at origin. Transformation matrix are :

T_x=
$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

T_y=
$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

T_z=
$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

3. Perform a two point perspective projection on a cube with centers of projection at x=-10 and y=-10 projected onto the z=0 plane. Tranformation matrix is

$$\begin{matrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Also p=-(1/x) and q= -(1/y)

4. Perform a three point perspective projection on a cube with centers of projection at x=-10, y=-10 and z= 10 plane. Vanishing Point is x=10, y=10, and z=-10. Tranformation matrix is

$$\begin{matrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{matrix}$$

Also p=-(1/x), q= -(1/y) and r= -(1/z)

Instructions

Steps for Gouraud Shading

1. Determine the *average* unit normal vector at each vertex of the polygon surface.
2. Apply lightening model at each vertex to compute color at that position.
3. Linearly interpolate the vertex intensities over the projected area of the polygon.

Iterative algorithm for Gouraud Shading

1. Determine the average unit normal vector at each vertex of the polygon surface.
2. Let C_R=color of the rightmost edge pixel on the i^{th} scan line, x_r = the x-coordinate of the rightmost edge pixel, C_L = color of the leftmost edge pixel on the i^{th} scan line and x_l = the x-coordinate of the leftmost edge pixel. Compute and store the scan line constant $C_i = (C_R - C_L)/(x_r - x_l)$ for each scan line that lies within the projected area of the polygon.
3. Let C_T = color of the topmost vertex pixel of the j^{th} left edge, y_t = the y- coordinate of the topmost vertex pixel of the edge, C_B = color of the lowermost vertex pixel of the j^{th} left edge and y_l = the y-coordinate of the lowermost vertex pixel of the edge. Compute the edge constant $C_j = (C_B - C_T)/(y_t - y_l)$ for all the left edges of the polygon.
4. Initialize a temporarily variable *color* = 0.0.
5. **For** each scan line *i* within the polygon area starting from the top **do**
   For each pixel on the same scan line within the projected surface area, starting from the left **do**
   if it is vertex pixel **then**
   color = color computed using lightening model.
   else if it is the pixel on the left edge then
   color= color of the pixel of the previous scan line and same edge + C_j
   else
   color = color of previous pixel on the same scan line + C_i
   end if
   end for
6. **end for**