

e

Simple perspective projection

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix} \Rightarrow \begin{cases} x' = \frac{d}{z}x \\ y' = \frac{d}{z}y \\ z' = \frac{d}{z}z = d \end{cases}$$

This achieves a simple perspective projection onto the view plane z = d

but we've lost all information about z!

<whiteboard>

Perspective Projection

$$P = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad z' = (n+f) - \frac{nf}{z}$$

















OpenGL Perspective Viewing

glFrustum(xmin,xmax,ymin,ymax,near,far)



Using Field of View

With glFrustum it is often difficult to get the desired view gluPerpective(fovy, aspect, near, far) often provides a better interface





Clipping after the perspective transformation can cause problems



[Shirley, Marschner]

Hidden Surface Removal

Hidden Surface Removal





"painter's algorithm" draw primitives in back-to-front order



[Wikimedia Commons]

Occlusion



"painter's algorithm" draw primitives in back-tofront order

> **problem**: triangle intersection

Occlusion

"painter's algorithm" draw primitives in back-tofront order

> **problem**: occlusion cycle

test depth on a pixel by pixel basis

red drawn last



at each pixel, record distance to the closest object that has been drawn in a *depth* buffer









http://www.beyond3d.com/content/articles/41/

Backface culling: another way to eliminate hidden geometry



Hidden Surface Removal in OpenGL

glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glEnable(GL_DEPTH_TEST);
glEnable(GL_CULL_FACE);

For a perspective transformation, there is more precision in the depth buffer for z-values closer to the near plane