Computer Graphics Note

Yichen, Di

December 4, 2023

1 Working with light

Energe adds according to $E(\lambda) = E_1(\lambda) + E_2(\lambda)$

YUV color space: luminance, U and V, can be obtained by $\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = P \begin{bmatrix} R \\ G \\ B \end{bmatrix}$

We perceive brightness intensity differences better at lower (as opposed to higher) light intensities, so logarithmic compression is often used.

2 Virtual world

Rotation matrix: $R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$ $R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$,$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

In rotation, line segments, angles and shapes are preserved.

Scaling matrix: $S = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}$

homogenous coordinates: point $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \to \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \vec{x_H}$, vector $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \to \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ 0 \end{bmatrix} = \vec{u_H}$

rotation $R' = \begin{bmatrix} & & & 0 \\ & R & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ translation $T' = \begin{bmatrix} & & & t_1 \\ & I & & t_2 \\ & & & t_3 \\ 0 & 0 & 0 & t_4 \end{bmatrix}$, $T' \begin{bmatrix} \vec{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \vec{x} + \vec{t} \\ 1 \end{bmatrix}$

Screen space projection: $x' \frac{hx}{z}$, $y' = \frac{hy}{z}$. Because $\frac{1}{z}$ is nonlinear, we write the result as

$$\begin{bmatrix} x'w \\ y'w \\ z'w \\ w \end{bmatrix} = \begin{bmatrix} h & 0 & 0 & 0 \\ 0 & h & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$z' = n + f - \frac{nf}{z}$ can be used to compute occlusion/transparency.

3 Triangles

Rasterization: the process of transforming the vertices to screen space.

$\vec{n}$ is the outward normal of the edge and p_0 one of the endpoints of the edge, then $(\vec{p} - \vec{p_0}) \cdot \vec{n}$ with p,p_0 on the same plane means that p is on the interior side of the edge.

If on the interior sides of all three edges, then p is inside the 2D triangle.

Color each pixel using the triangle that has the smallest z' value.

For p inside triangle, $p = \alpha_0 p_0 + \alpha_1 p_1 + \alpha_2 p_2 = \vec{p_2} + \beta_0(p_0 - p_2) + \beta_1(p_1 - p_2) = p_2 + \beta_0 u + \beta_1 v = p_2 + \alpha_0 u + \alpha_1 v$

After perspective projection: $p' = \alpha'_0 p'_0 + \alpha'_1 p'_1 + \alpha'_2 p'_2$, $\begin{bmatrix} \alpha'_0 \\ \alpha'_1 \end{bmatrix} = \begin{bmatrix} \frac{z_0}{z_p}\alpha_0 \\ \frac{z_1}{z_p}\alpha_1 \end{bmatrix}$, $\frac{1}{z} = \alpha'_0 \frac{1}{z_0} + \alpha'_1 \frac{1}{z_1} + \alpha'_2 \frac{1}{z_2}$

4 Ray tracing