

Class 1-4

Basic Transformation

3-D rotation

$$rotation - z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$rotation - x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$rotation - y(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

because we make z rotates to x the y-rotation is different

Transformation Expansion

Because of translation transformation we need to expand one dimension

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \end{bmatrix}$$

The x and y all move 1 in this example

Rodrigues Transformation

We can use this transformation to rotate by any given axis

$$R = \cos\theta * I + (1 - \cos\theta) \begin{bmatrix} k_1 & k_2 & k_3 \\ k_2 & k_1 & k_3 \\ k_3 & k_3 & k_1 \end{bmatrix} + \sin\theta \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}$$

Viewing Transformation

1. Modeling Transformation

The combination of basic transformations(translation,rotation,scaling,shearing)

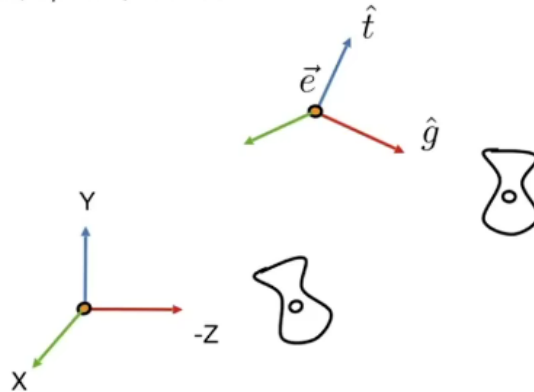
2. View\Camera Transformation

We need to remove the camera to the global origin

GAME101-Webinar bilibili

View / Camera Transformation

- Transform the camera by M_{view}
 - So it's located at the origin, up at Y, look at -Z
- M_{view} in math?
 - Translates e to origin
 - Rotates g to -Z
 - Rotates t to Y
 - Rotates $(g \times t)$ To X
 - Difficult to write!



15

Lingqi Yan, UC Santa I

1. Move the camera to the origin

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_e & 0 & 1 & 0 & -y_e & 0 & 0 & 1 & -z_e & 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Modify the angle to make three vectors corresponding to three axes

As the transformation is difficult, we can think of the inverse transformation first

$$R_{view}^{-1} = \begin{bmatrix} x_{\hat{g} \times \hat{t}} & x_t & x_{-g} & 0 & y_{\hat{g} \times \hat{t}} & y_t & y_{-g} & 0 & z_{\hat{g} \times \hat{t}} & z_t & z_{-g} & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

so that the original matrix is

$$R_{view} = \begin{bmatrix} x_{\hat{g} \times \hat{t}} & y_{\hat{g} \times \hat{t}} & z_{\hat{g} \times \hat{t}} & 0 & x_t & y_t & z_t & 0 & x_{-g} & y_{-g} & z_{-g} & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The final result is

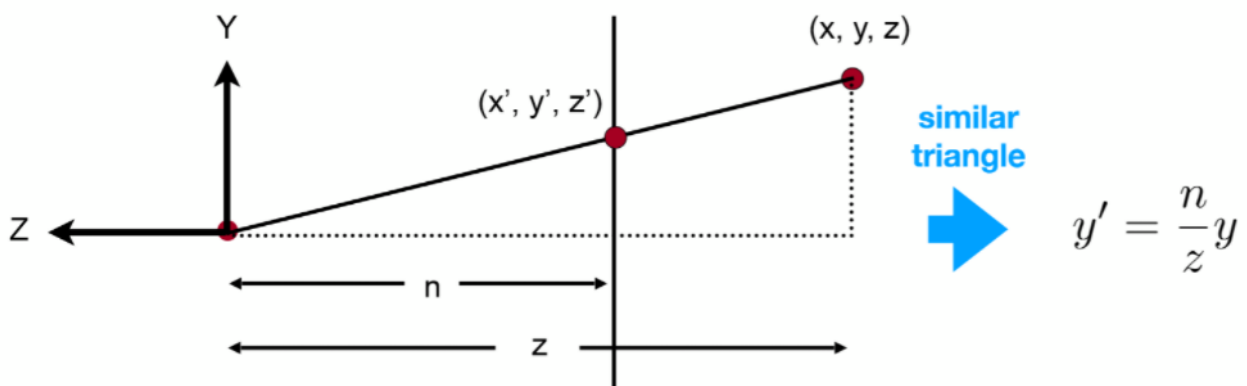
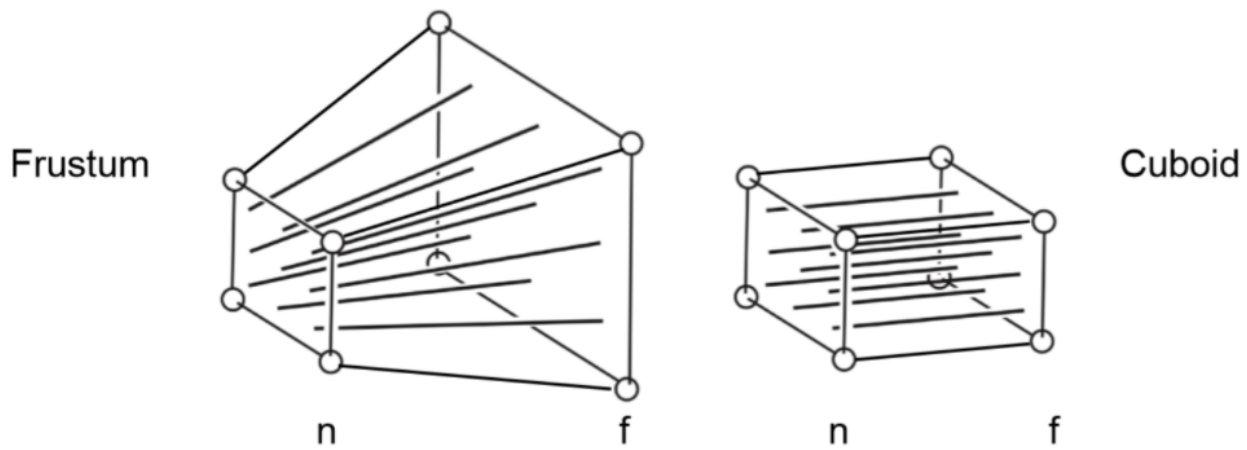
$$V = R_{view}T$$

3. Projection Transformation

3.1 Orthographic Projection Transformation

$$M_{ortho} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 & \frac{2}{t-b} & 0 & 0 & 0 & \frac{2}{n-f} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} & 0 & 1 & 0 & -\frac{t+b}{2} & 0 & 0 & 1 & -\frac{n+f}{2} & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2 Perspective Projection Transformation



Now we can get the initial vector

$$M_{persp \rightarrow ortho} [x \ y \ z \ 1] = \left[\frac{nx}{z} \ \frac{ny}{z} \ unknown \ 1 \right] == [nx \ ny \ unknown \ z]$$

Then we know that: 1. The coordinate on near plane will not change 2. The z on far plane will not change

Then we can know the initial matrix:

$$M_{persp \rightarrow ortho} = \begin{bmatrix} n & 0 & 0 & 0 & 0 & n & 0 & 0 & 0 & 0 & A & B & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$An+B=n^2 \quad Af+B=f^2$$

$$M_{persp \rightarrow ortho} = \begin{bmatrix} n & 0 & 0 & 0 & 0 & n & 0 & 0 & 0 & 0 & n+f & -fn & 0 & 0 & 1 & 0 \end{bmatrix}$$