Tutorial 3: Viewing and Projection

About Assignment 1

- 36 github repository
- 6 email package

- Demonstration will be at 9:00 pm-10:30 pm, Mar 21 (Sunday), in Room 202A, SIST No. 2 Building.
- You should bring your laptop with you. Make sure that your code can be run successfully.

View/Camera Transformation

- Transform from camera to world Coordinate
- Transform from world to camera coordinate
- How to define the M_view Matrix

Scaling

$$\begin{bmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & S_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} S_1 \cdot x \\ S_2 \cdot y \\ S_3 \cdot z \\ 1 \end{pmatrix}$$

Translation

$$\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + T_x \\ y + T_y \\ z + T_z \\ 1 \end{pmatrix}$$

Trans & Scale

$$Trans.Scale = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Lookat

- using 3 perpendicular axes you can create a matrix with those 3 axes plus a translation vector and you can transform any vector to that coordinate space by multiplying it with this matrix.

$$LookAt = \begin{bmatrix} R_x & R_y & R_z & 0 \\ U_x & U_y & U_z & 0 \\ D_x & D_y & D_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & -P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Translation

$$T = \begin{bmatrix} 0 & 0 & 0 & eye_x \\ 0 & 0 & 0 & eye_y \\ 0 & 0 & 0 & eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- P(X,Y,Z) in camera coordinate, calculate p(x’,y’,z’) in world coordinate

$$R = \begin{bmatrix} s[0] & u[0] & -f[0] & 0 \\ s[1] & u[1] & -f[1] & 0 \\ s[2] & u[2] & -f[2] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$TR * P = p'$$

$$view = (T * R)^{-1} = R^{-1} * T^{-1} = R^T * T^{-1}$$

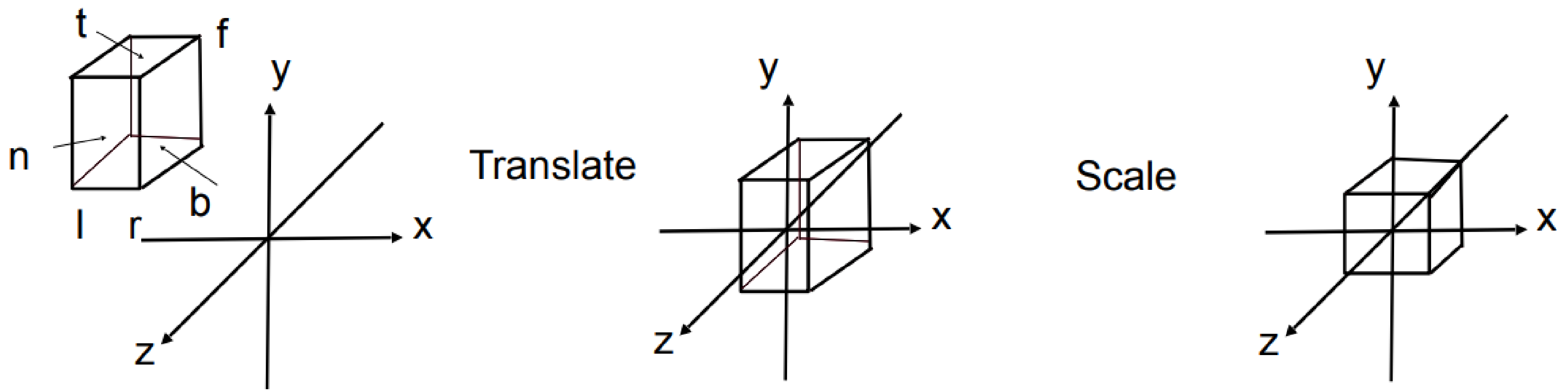
$$R^T = \begin{bmatrix} s[0] & s[1] & s[2] & 0 \\ u[0] & u[1] & u[2] & 0 \\ -f[0] & -f[1] & -f[2] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P = (T * R)^{-1} p'$$

Orthographic Projection

- Transformation matrix?
  - Translate (**center** to origin) first, then scale (length/width/height to **2**)

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



Projection Transform

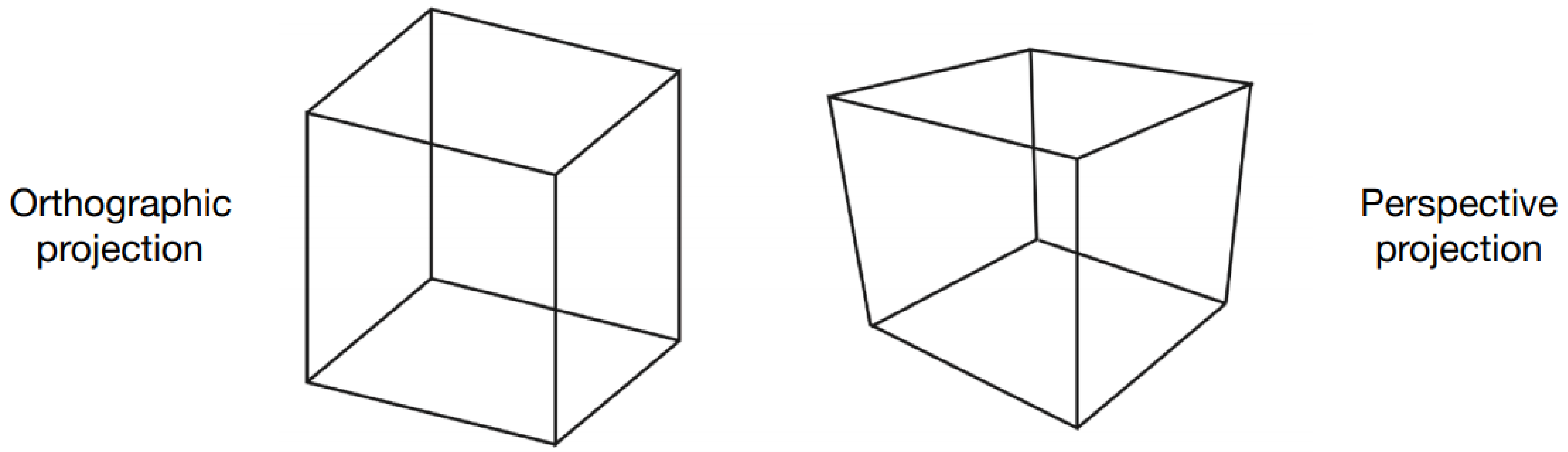
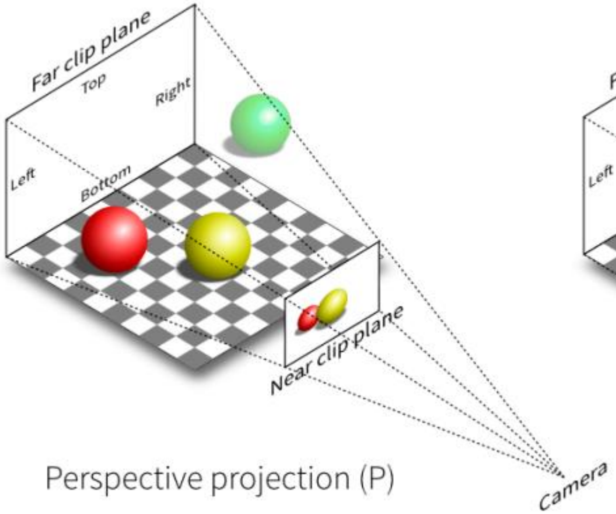


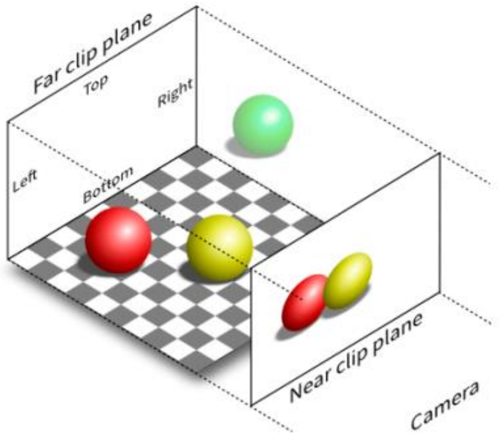
Fig. 7.1 from Fundamentals of Computer Graphics, 4th Edition

Projection Transform

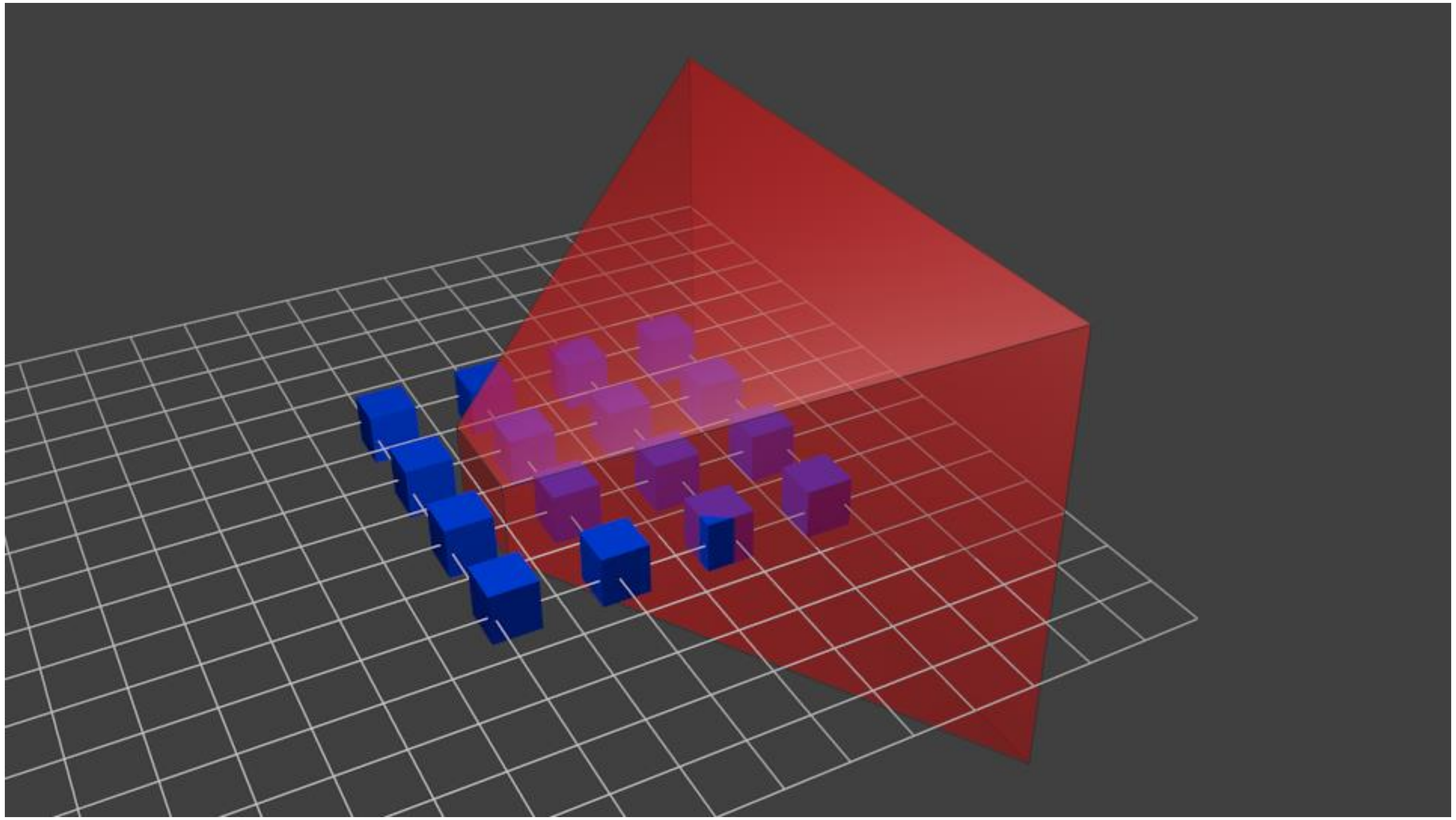
- Perspective projection vs. orthographic projection



Perspective projection (P)

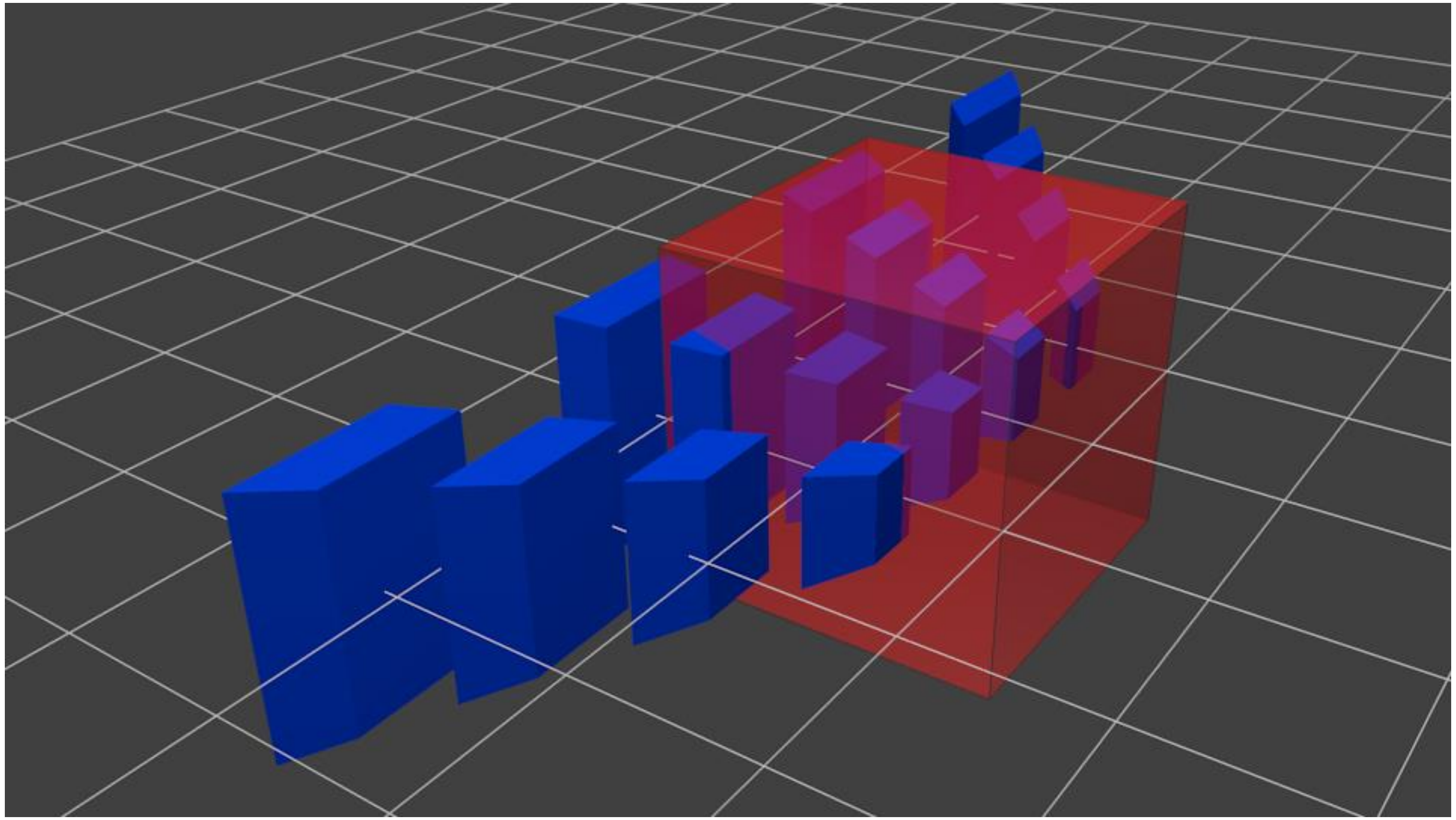


Orthographic projection (O)

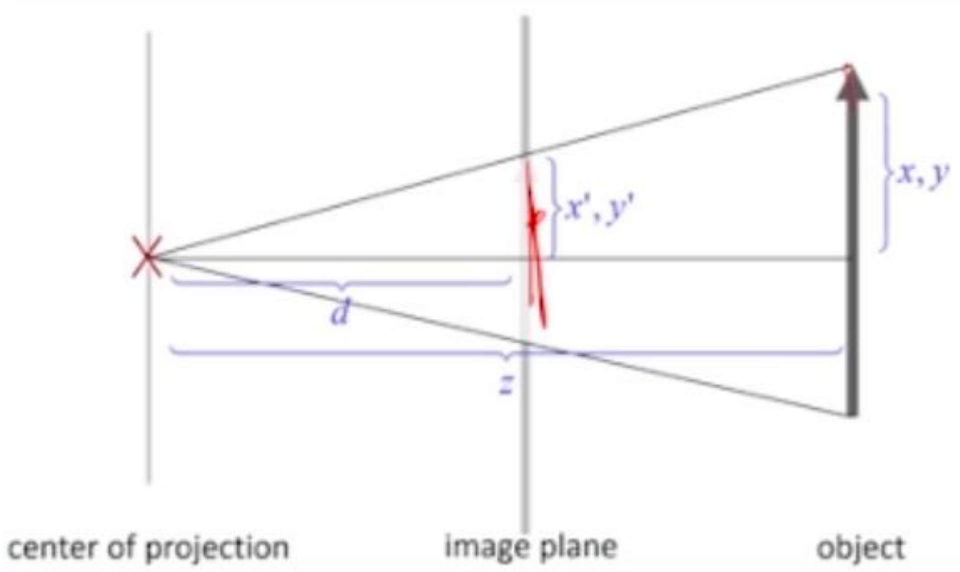


www.opengl-tutorial.org

After multiply by Projection Matrix



Calculate View matrix and Projection Matrix



$$x' = d\frac{x}{z}, \quad y' = d\frac{y}{z}$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ \omega' \end{pmatrix} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

How to get the persp->ortho matrix?

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \Longrightarrow \begin{pmatrix} \frac{n}{z}x \\ \frac{n}{z}y \\ \text{unknown} \\ 1 \end{pmatrix} \Longrightarrow^{\times z} \begin{pmatrix} nx \\ ny \\ \text{still unknown} \\ z \end{pmatrix}$$

$$M_{persp\to ortho} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ ? & ? & ? & ? \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

For near plane

- Any point on the near plane will not change

$$M^{(4\times4)}_{persp\to ortho}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ \text{unknown} \\ z \end{pmatrix} \xrightarrow{\text{replace z with n}} \begin{pmatrix} x \\ y \\ n \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} x \\ y \\ n \\ 1 \end{pmatrix} == \begin{pmatrix} nx \\ ny \\ n^2 \\ n \end{pmatrix}$$

- So the third row must be of the form (0 0 A B)

$$\begin{pmatrix} 0 & 0 & A & B \end{pmatrix}\begin{pmatrix} x \\ y \\ n \\ 1 \end{pmatrix} = n^2$$

n^2 has nothing to do with x and y

For far plane

- What do we have now?

$$\begin{pmatrix} 0 & 0 & A & B \end{pmatrix} \begin{pmatrix} x \\ y \\ n \\ 1 \end{pmatrix} = n^2 \quad \Rightarrow \quad An + B = n^2$$

- Any point's z on the far plane will not change

$$\begin{pmatrix} 0 \\ 0 \\ f \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ 0 \\ f \\ 1 \end{pmatrix} == \begin{pmatrix} 0 \\ 0 \\ f^2 \\ f \end{pmatrix} \quad \Rightarrow \quad Af + B = f^2$$

Final Perspective matrix

$$M_{persp\to ortho} \quad \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$M_{persp} = M_{ortho} M_{persp\to ortho}$$