Computer Graphics (CS 543)
Lecture 6c: Derivation of Orthographic Projection

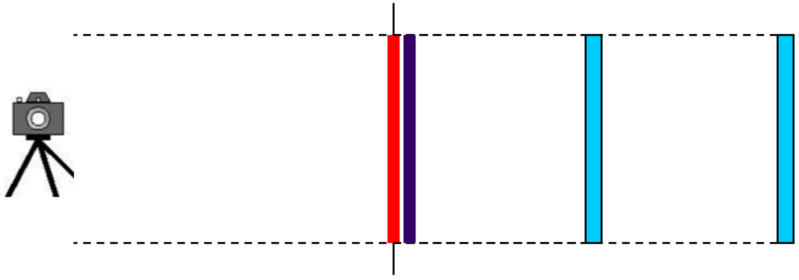
Prof Emmanuel Agu

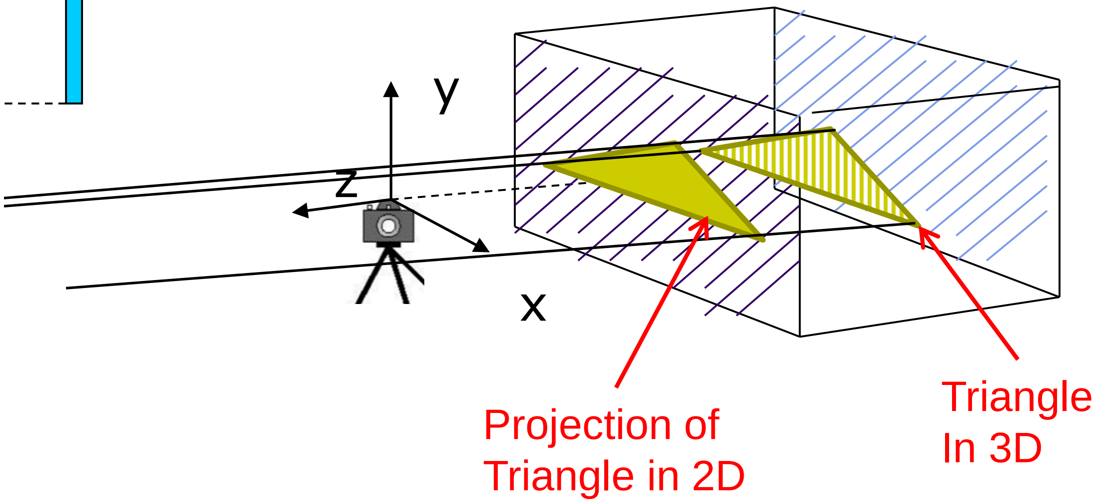
Computer Science Dept.

Worcester Polytechnic Institute (WPI)

Orthographic Projection

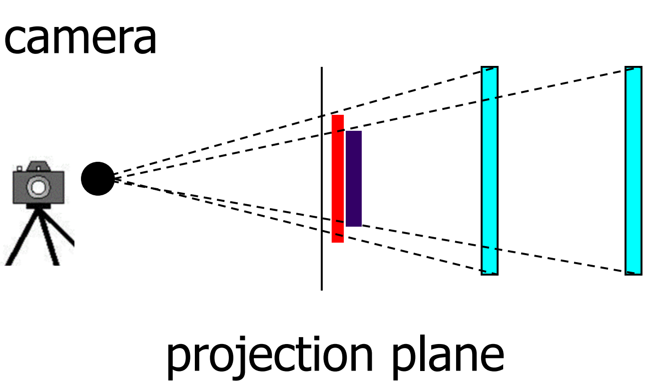
- How? Draw parallel lines from each object vertex
- The projection center is at infinite
- In short, use (x,y) coordinates, just drop z coordinates

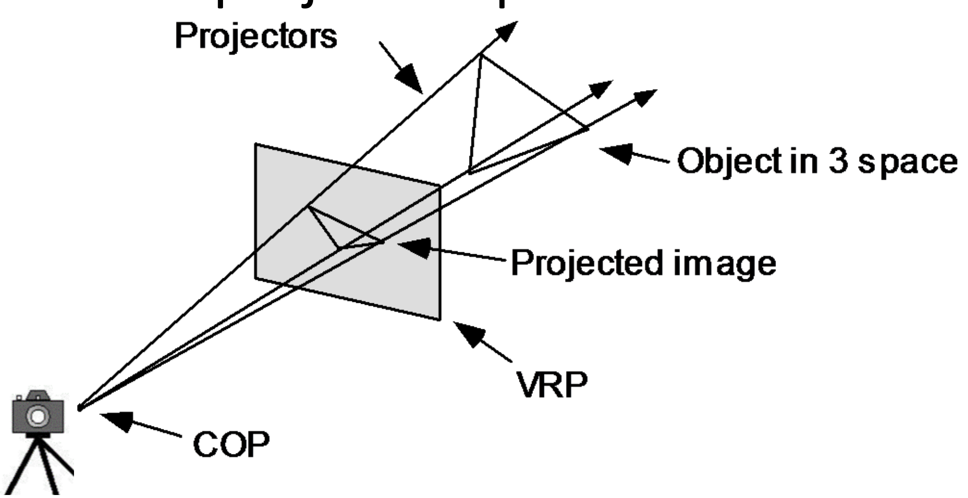




Perspective Projection

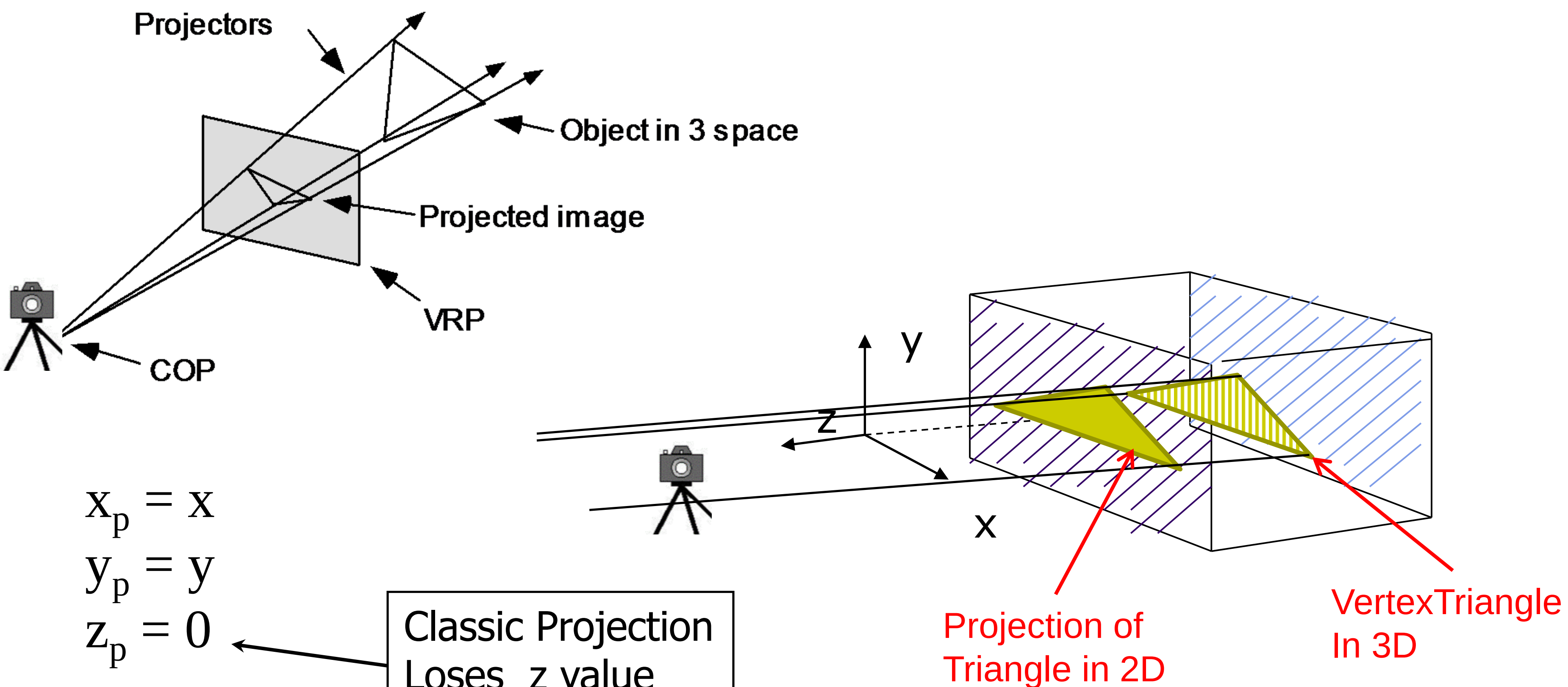
- After setting view volume, then projection transformation
- Projection?
  - **Classic:** Converts 3D object to corresponding 2D on screen
  - How? Draw line from object to projection center
  - Calculate where each intersects projection plane





The Problem with Classic Projection

- Keeps (x,y) coordintates for drawing, drops z
- We may need z. Why?

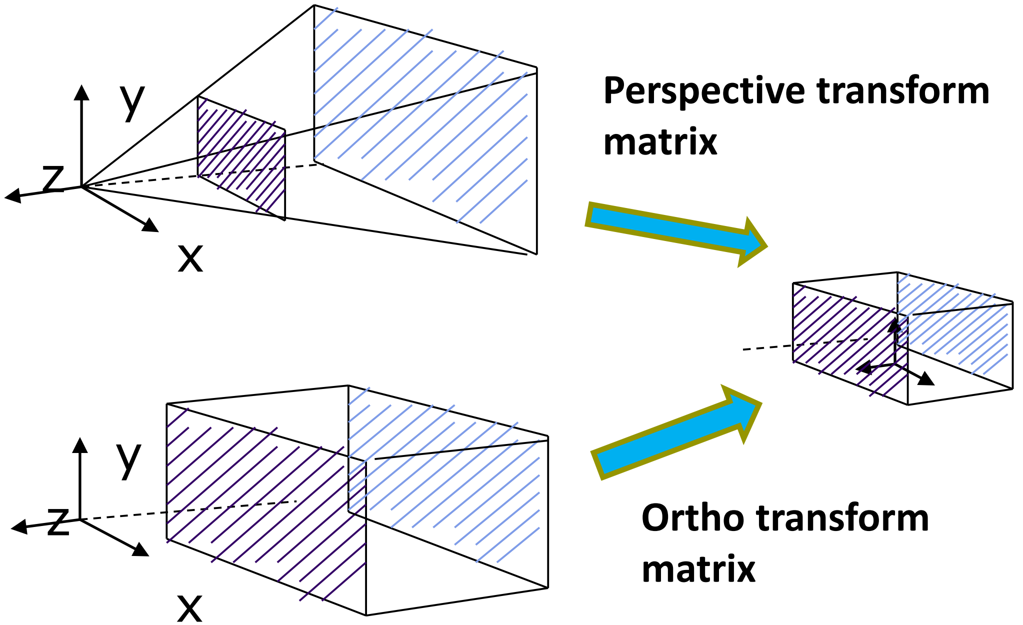


$x_p = x$
$y_p = y$
$z_p = 0$

Classic Projection
Loses z value

Normalization: Keeps z Value

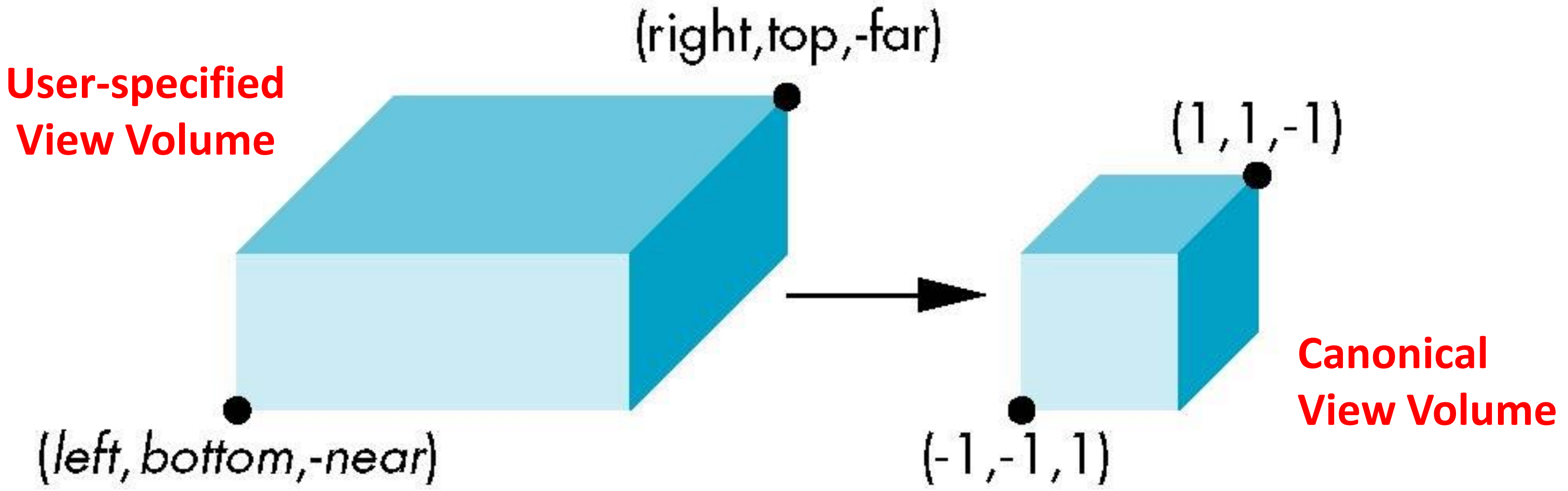
- Most graphics systems use *view normalization*
- **Normalization:** convert all other projection types to orthogonal projections with the ***default view volume***



Default view volume
Clipping against it

Parallel Projection

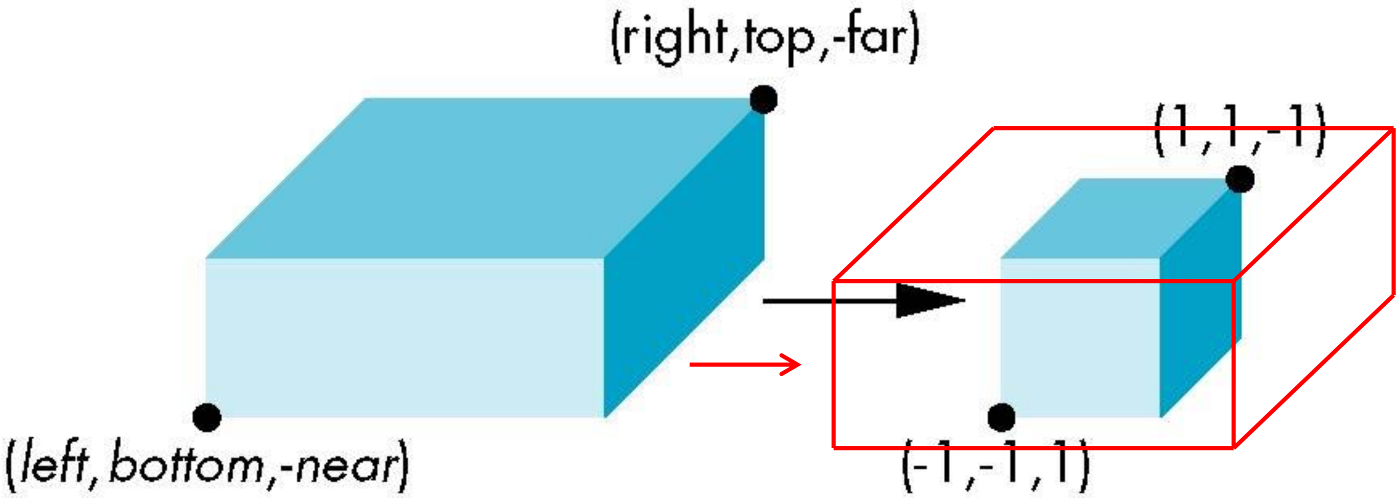
- **normalization** ⇒ find 4x4 matrix to transform **user-specified view volume** to **canonical view volume (cube)**



glOrtho(left, right, bottom, top,near, far)

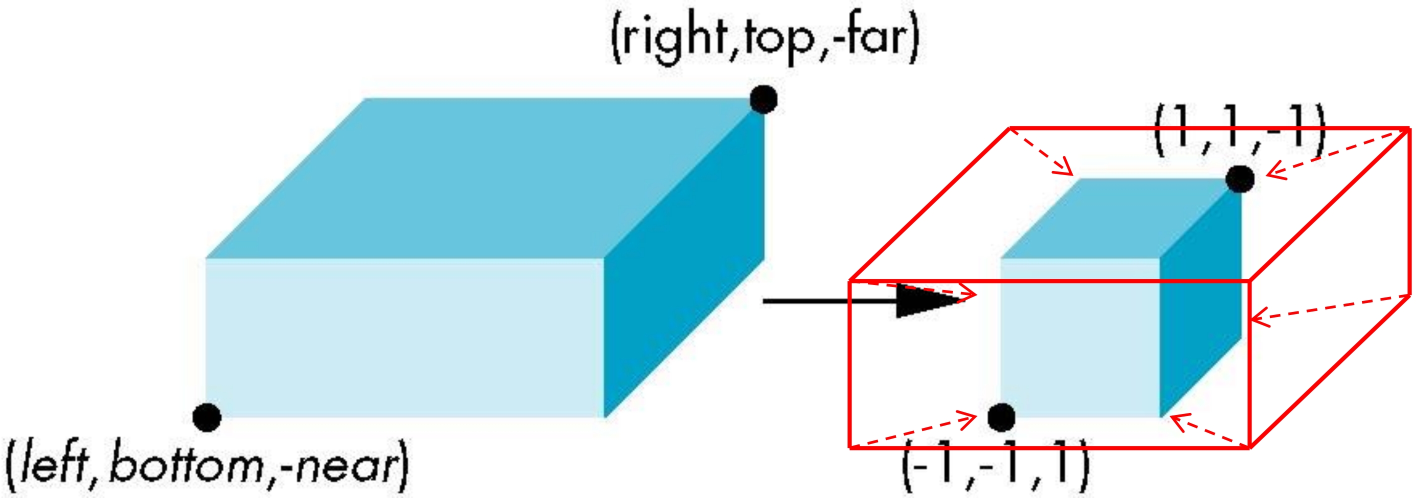
Parallel Projection: Ortho

- Parallel projection: 2 parts
  1. **Translation:** centers view volume at origin



Parallel Projection: Ortho

2. **Scaling:** reduces user-selected cuboid to canonical cube (dimension 2, centered at origin)



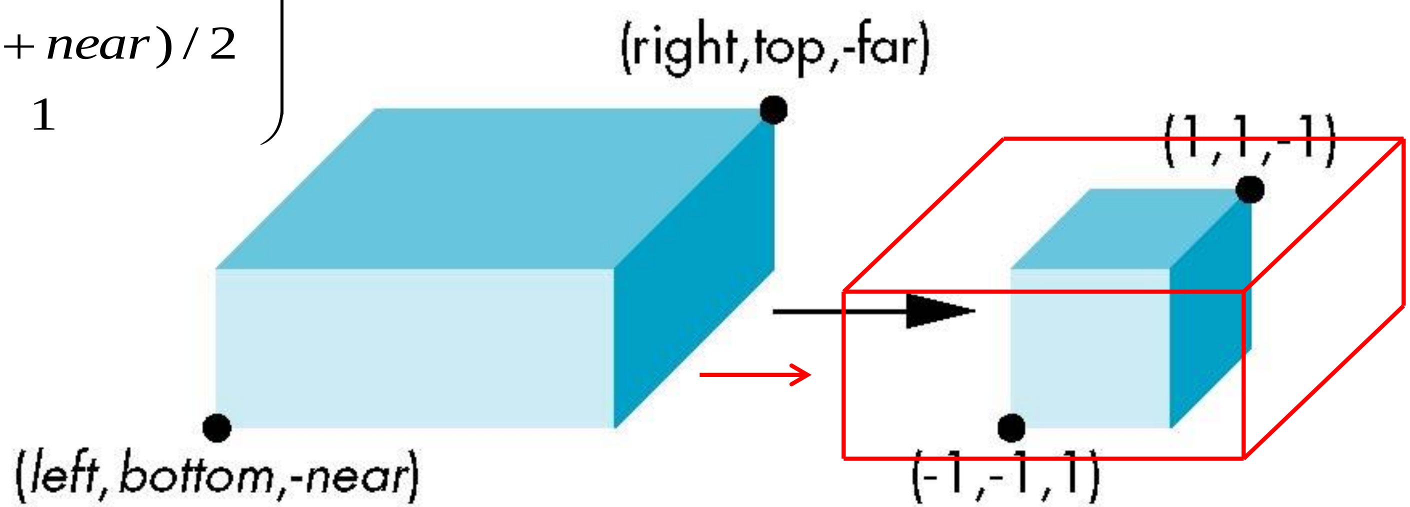
Parallel Projection: Ortho

- Translation lines up midpoints: E.g. midpoint of x = (right + left)/2
- Thus translation factors:

  -(right + left)/2, -(top + bottom)/2, -(far+near)/2

- Translation matrix:

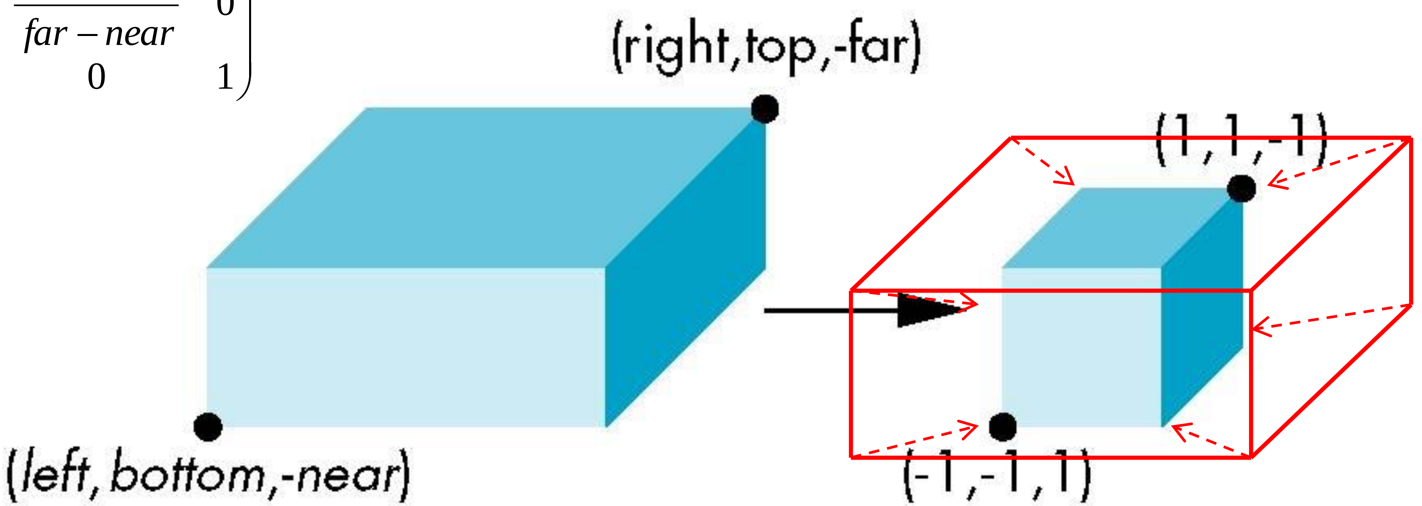
$$\begin{pmatrix} 1 & 0 & 0 & -(right+left)/2 \\ 0 & 1 & 0 & -(top+bottom)/2 \\ 0 & 0 & 1 & -(far+near)/2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Parallel Projection: Ortho

- Scaling factor: ratio of ortho view volume to cube dimensions
- Scaling factors: 2/(right - left), 2/(top - bottom), 2/(far - near)
- Scaling Matrix M2:

$$\begin{pmatrix} \frac{2}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2}{top-bottom} & 0 & 0 \\ 0 & 0 & \frac{2}{far-near} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Parallel Projection: Ortho

Concatenating **Translation** x **Scaling**, we get Ortho Projection matrix

$$\begin{pmatrix} \frac{2}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2}{top-bottom} & 0 & 0 \\ 0 & 0 & \frac{2}{far-near} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & -(right+left)/2 \\ 0 & 1 & 0 & -(top+bottom)/2 \\ 0 & 0 & 1 & -(far+near)/2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{P} = \mathbf{ST} = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right-left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{2}{near-far} & \frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Final Ortho Projection

- Set $z = 0$
- Equivalent to the homogeneous coordinate transformation

$$\mathbf{M}_{\text{orth}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Hence, general orthogonal projection in 4D is
  $\mathbf{P} = \mathbf{M}_{\text{orth}}\mathbf{ST}$

References

- Interactive Computer Graphics (6th edition), Angel and Shreiner
- Computer Graphics using OpenGL (3rd edition), Hill and Kelley