Computer Graphics (CS 543)
Lecture 7 (Part 1): Projection (Part I)

Prof Emmanuel Agu

Computer Science Dept.

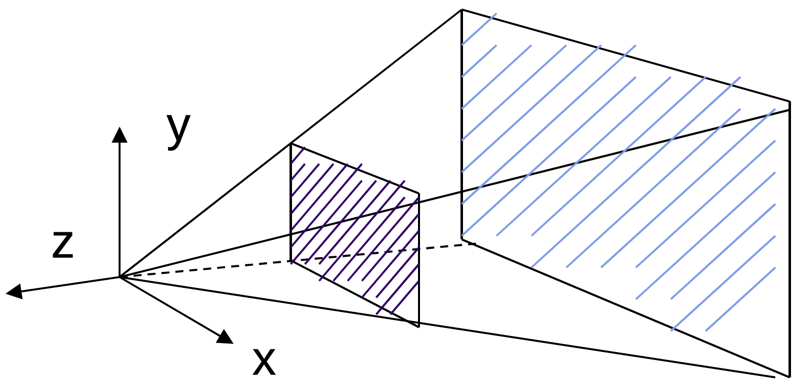
Worcester Polytechnic Institute (WPI)

Reminder

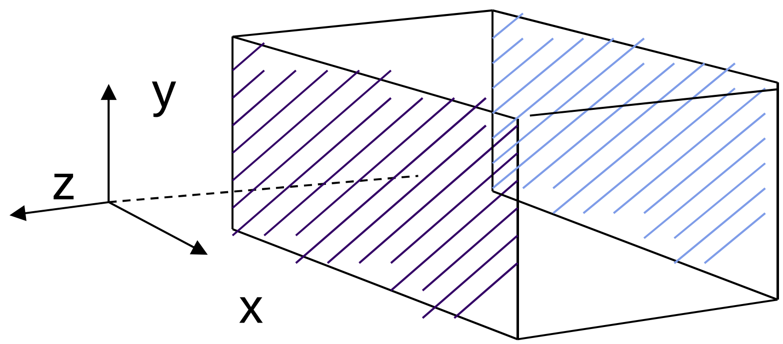
- No class next week Tuesday (Term break)!

Projection Transformation

- Projection? map the object from 3D space to 2D screen



Perspective: **Perspective()**



Parallel: **Ortho()**

Default Projections and Normalization

- What if you user does not set up projection?
- Default OpenGL projection in eye (camera) frame is orthogonal (Ortho( ));
- To project points within default view volume

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

Homogeneous Coordinate Representation

default orthographic projection

$$x_p = x$$
$$y_p = y$$
$$z_p = 0$$
$$w_p = 1$$

Vertices **before** Projection

Vertices **after** Projection

$$\mathbf{p}_p = \mathbf{M}\mathbf{p}$$

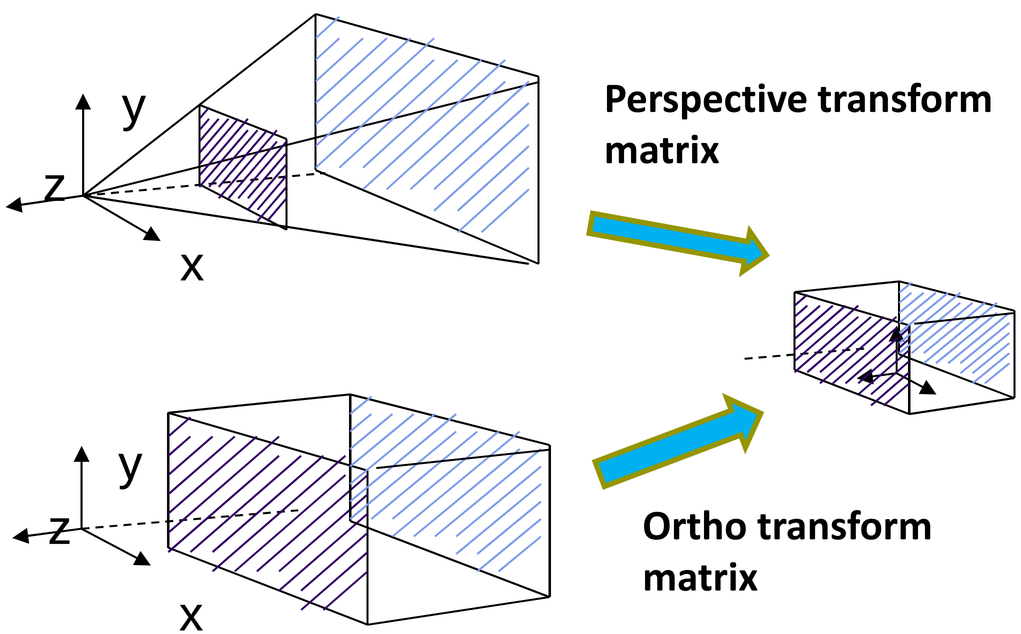
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Default Projection Matrix

In practice, can let **M = I,** set the *z* term to zero later

Normalization

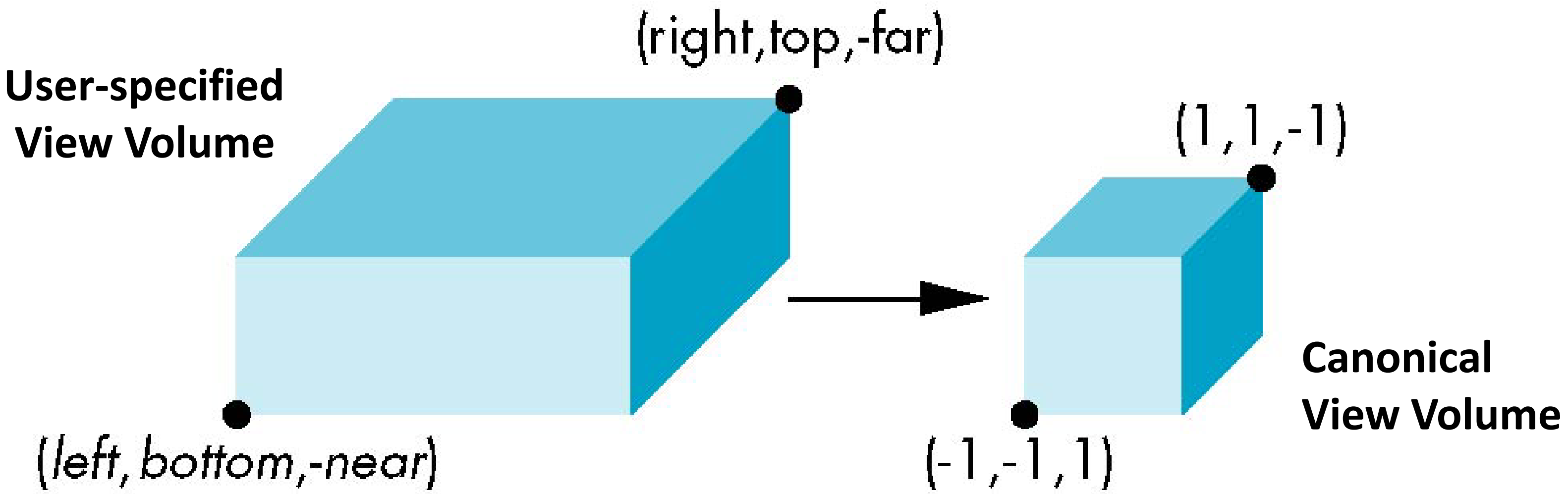
- Most graphics systems use *view normalization*
- **Normalization:** convert all other projection types to orthogonal projections with the default view volume



Default view volume
Clipping against it

Parallel Projection

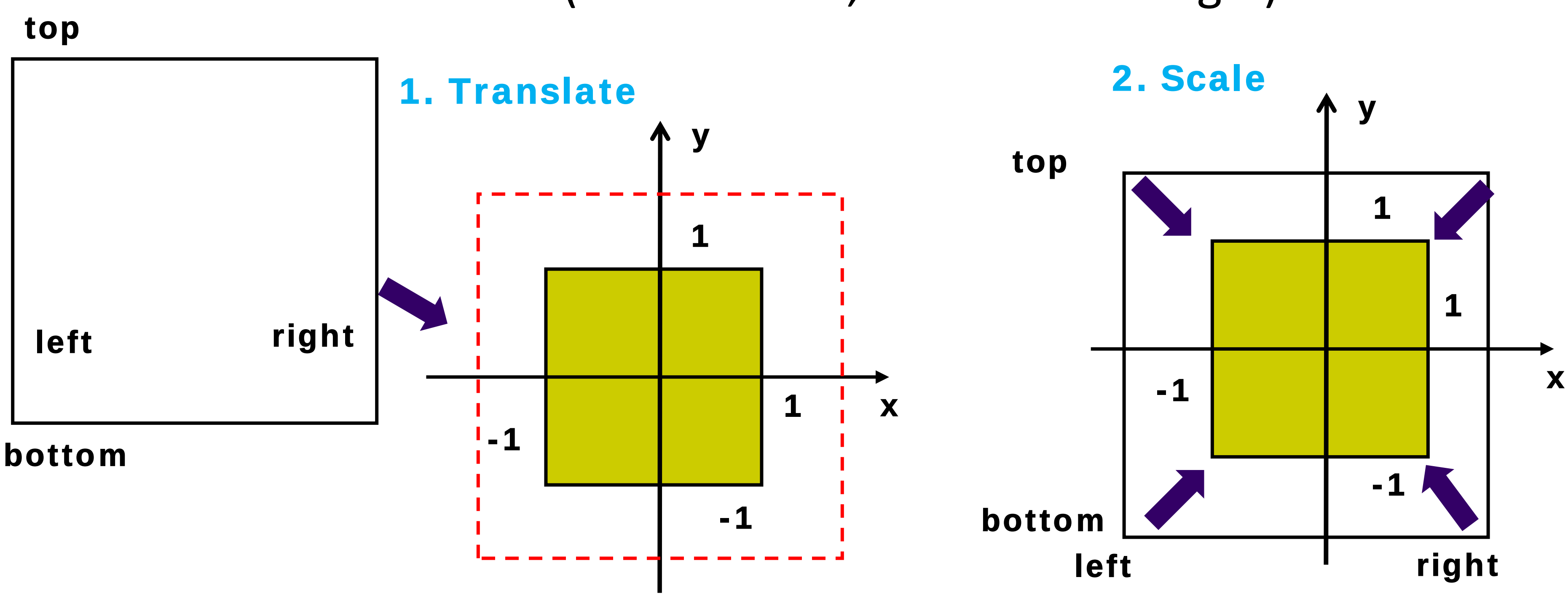
- **Approach:** Project everything in the visible volume into a **canonical view volume (cube)**
- **normalization** ⇒ find 4x4 matrix to convert specified view volume to default



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

Parallel Projection: Ortho

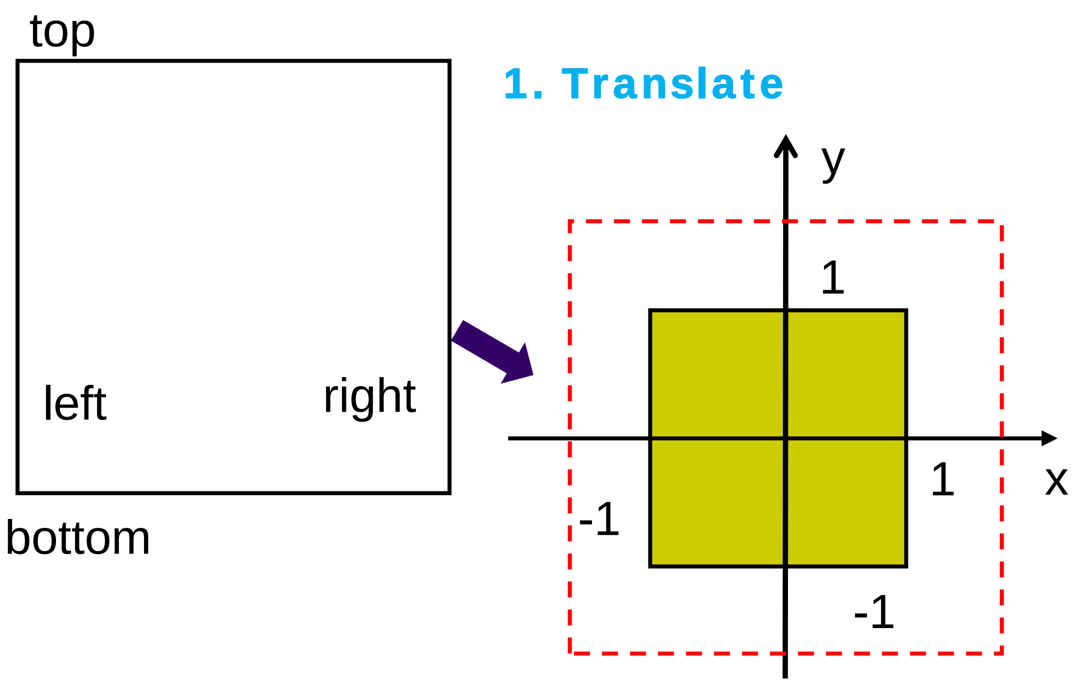
- Parallel projection can be broken down into two parts
  1. **Translation:** which centers view volume at origin
  2. **Scaling:** which reduces cuboid of arbitrary dimensions to canonical cube (dimension 2, centered at origin)



Parallel Projection: Ortho

- Translation sequence moves midpoint of view volume to coincide with origin:
- E.g. midpoint of x = (right + left)/2
- Thus translation factors along (x, y, z):

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



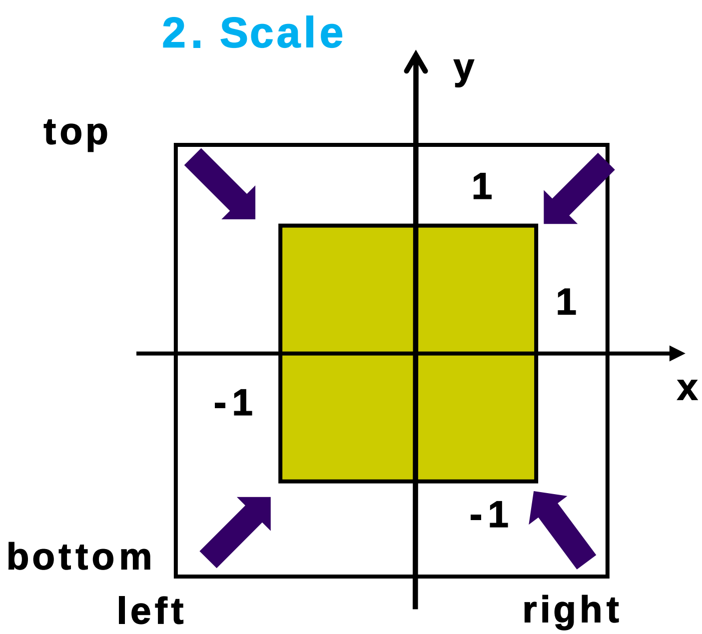
And translation matrix M1:

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

Parallel Projection: Ortho

- Scaling factor is ratio of cube dimension to Ortho view volume dimension
- Scaling factors along (x, y, z):

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



And 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 M1xM2, we get transform matrix used by glOrtho

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

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

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