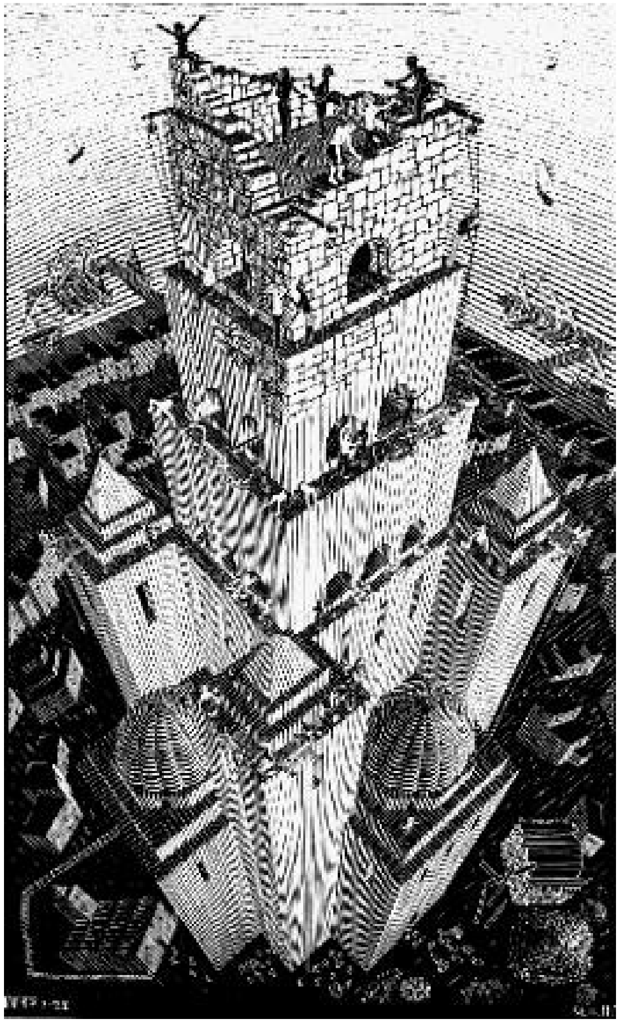
Viewing in 3D – Part II

Foley & Van Dam, Chapter 6

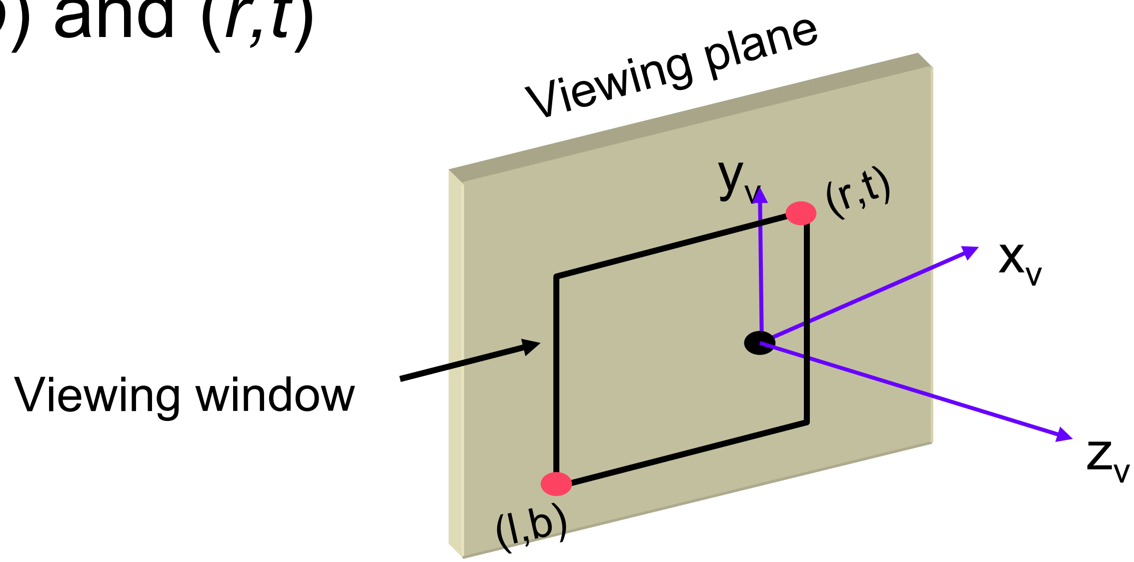


Viewing in 3D – Part II

- Viewing Window
- Viewing Volume
- Defining the Viewing Volume
- Canonical Viewing Volume
- Viewport Transformation

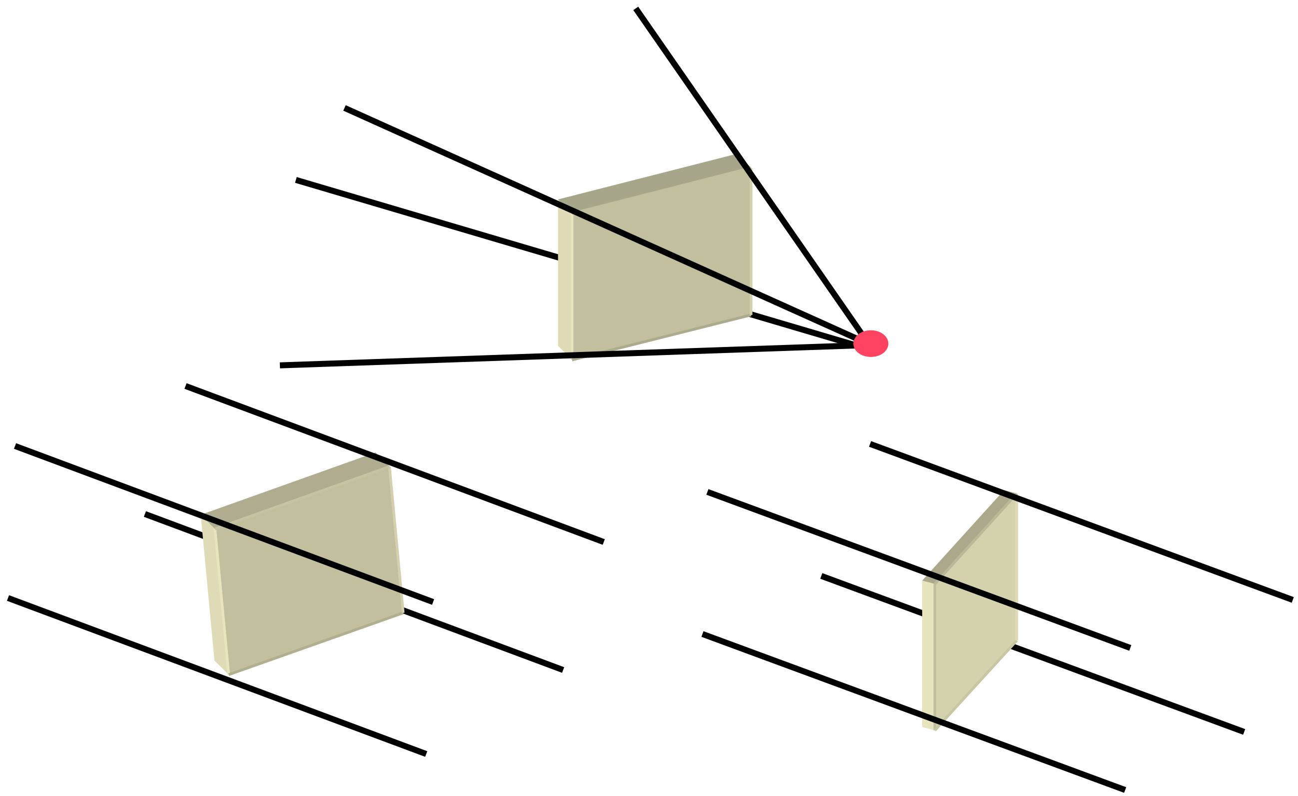
Viewing Window

- After objects were projected onto the viewing plane, an image is taken from a *Viewing Window*
- A viewing window can be placed *anywhere* on the view plane
- In general the view window is aligned with the viewing coordinates and is defined by its extreme points: (*l*,*b*) and (*r*,*t*)



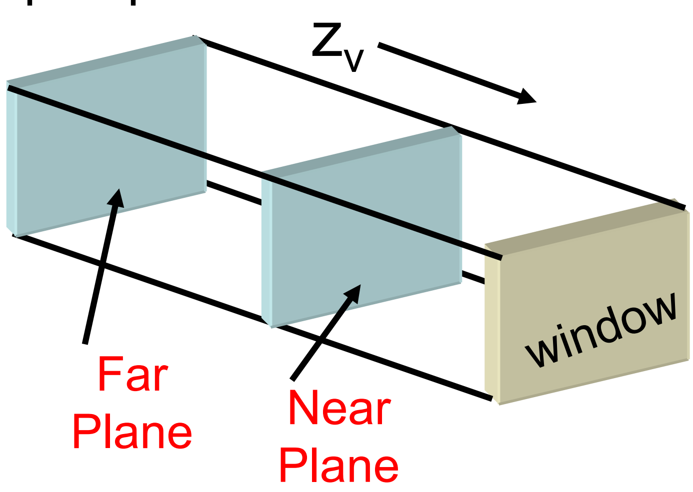
Viewing Volume

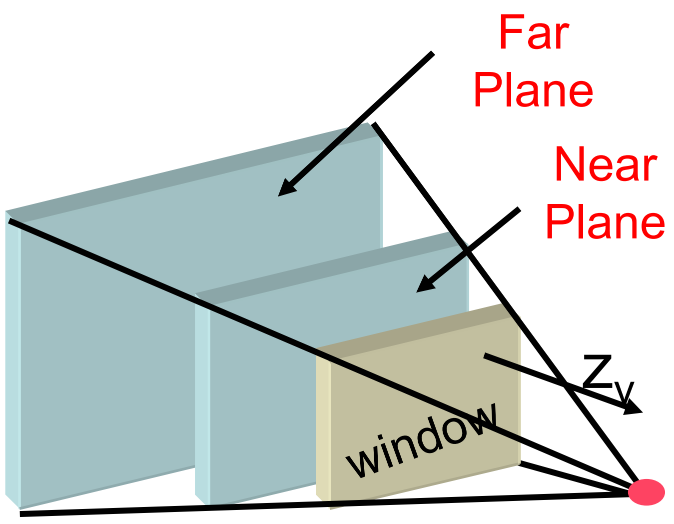
- Given the specification of the *viewing window,* we can set up a *Viewing Volume*
- Only objects inside the viewing volume will appear in the display, the rest are clipped



Viewing Volume

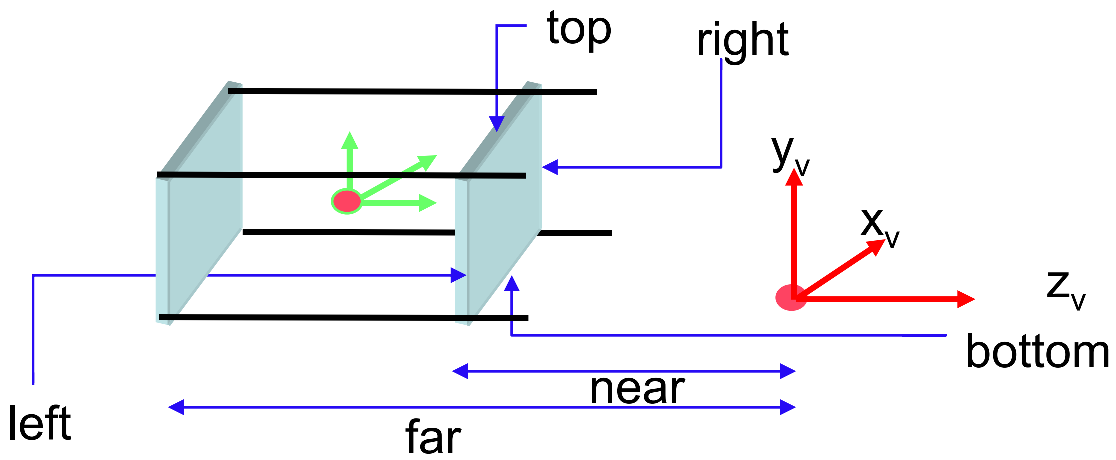
- In order to limit the infinite viewing volume we define two additional planes: *Near Plane* and *Far Plane*
- Only objects in the bounded viewing volume will appear
- The near and far planes are parallel to the viewing plane and specified by z_{near} and z_{far}
- A limited viewing volume is defined:
  - For orthographic: a rectangular parallelpiped
  - For oblique: an oblique parallelpiped
  - For perspective: a frustum





Defining the Viewing Volume

- Definition for Orthographic Projection:

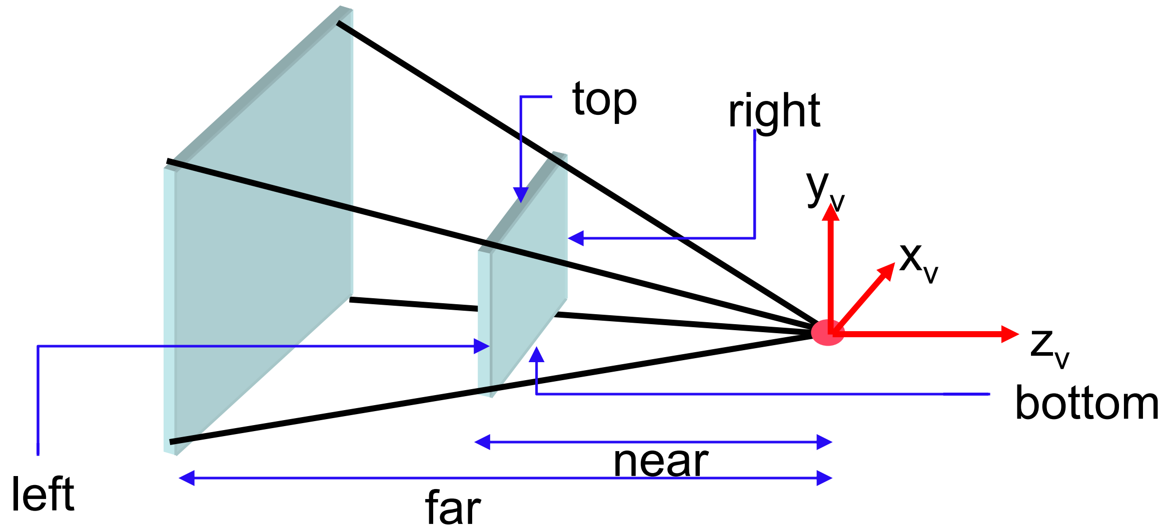



```
glMatrixMode(GL_PROJECTION) ;
glLoadIdentity();   /* camera “looks” down */
glOrtho(left, right, bottom, top, near, far);
```

- Note: In all definitions, *near* and *far* are distances (always positive). Near = 50 means that the near plane intersect the z-axis at z = -50

Defining the Viewing Volume

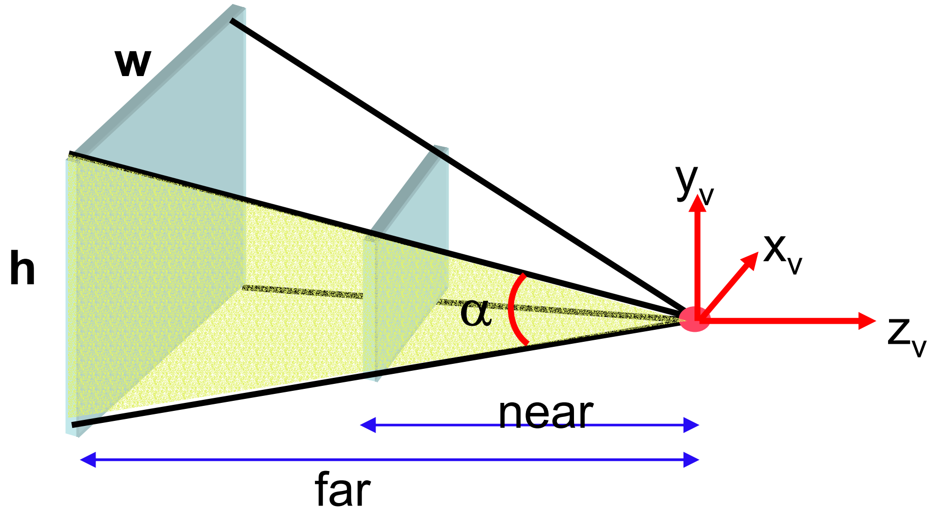
- Definition for general Perspective Projection:




```
glMatrixMode(GL_PROJECTION) ;
glLoadIdentity();
glFrustum(left, right, bottom, top, near, far);
```

Defining the Viewing Volume

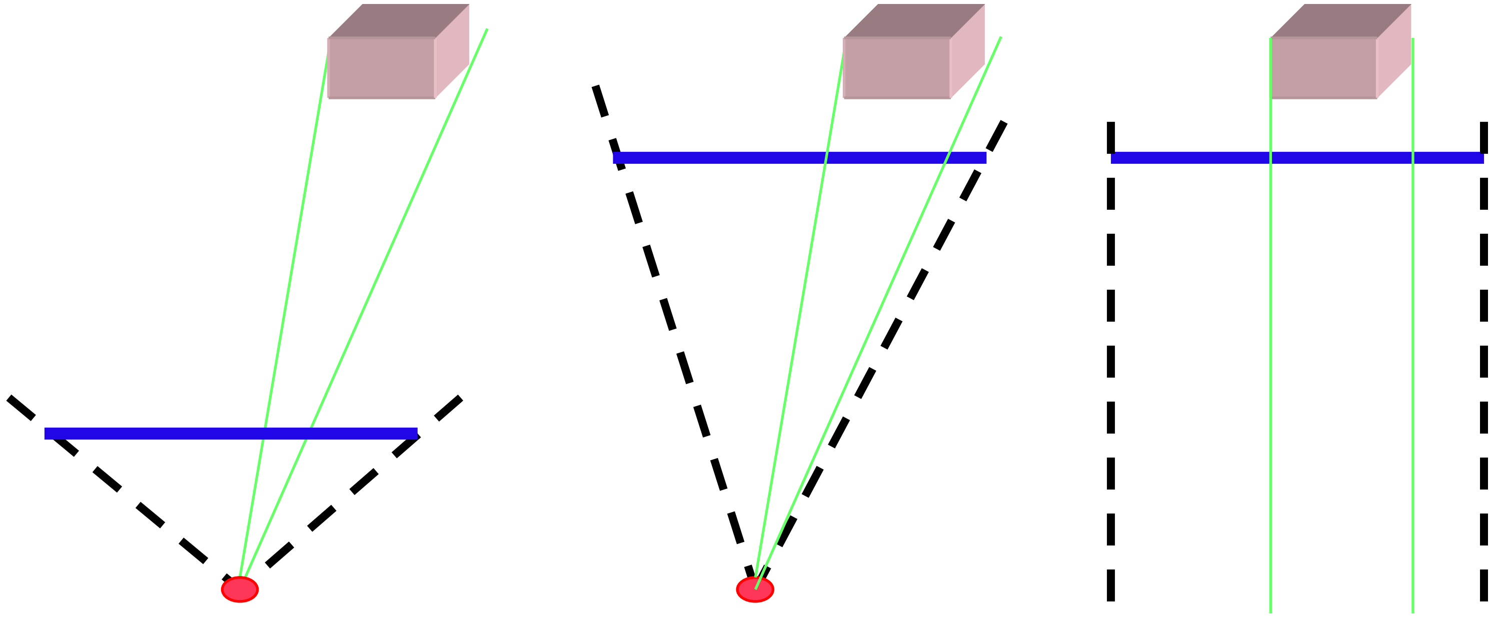
- Definition for a Standard Perspective Projection:




```
glMatrixMode(GL_PROJECTION) ;
glLoadIdentity();
aspectRatio = w/h;
gluPerspective(viewAngle, aspectRatio, near, far);
```

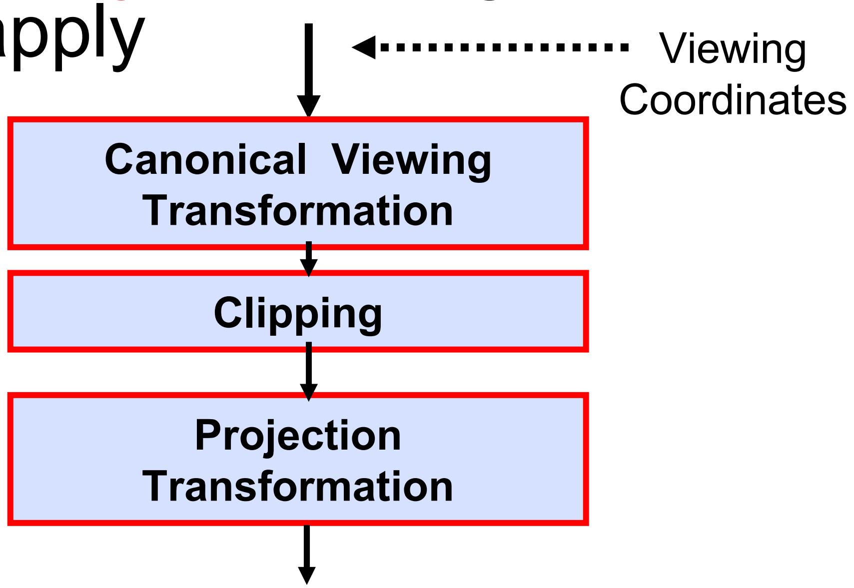
Position of the Viewing Plane

- **Parallel projection:** viewing-plane positioning does not affect the projected image
- **Perspective projection:** viewing-plane positioning does affect the projected image



Canonical Viewing Volume

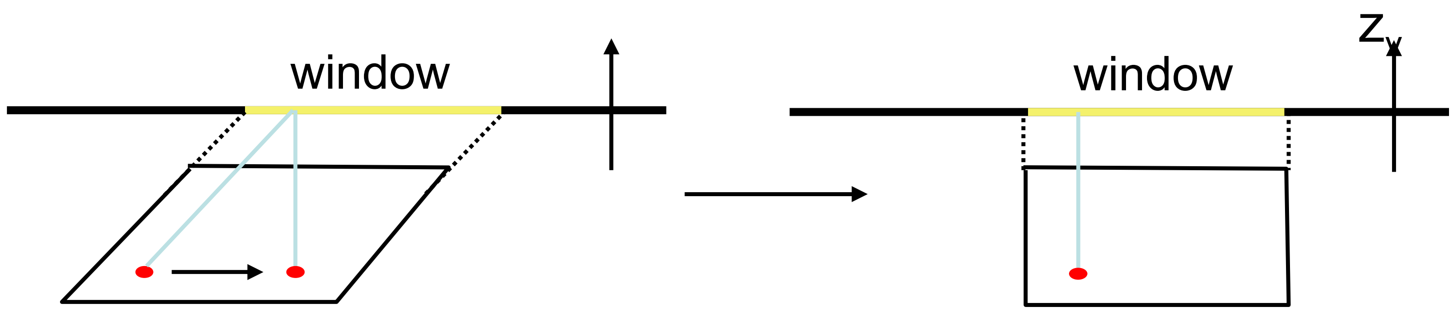
- In order to determine the objects that are seen in the viewing window we have to clip objects against six planes forming the view volume
- Clipping against arbitrary 3D planes requires considerable computation
- For fast clipping we transform the viewing volume into a *canonical viewing volume* against which clipping is easy to apply



Canonical Volume: Parallel Projection

- Depth-preserving shear

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a\cos\phi & 0 \\ 0 & 1 & a\sin\phi & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} = \begin{bmatrix} x_v + z_v a\cos\phi \\ y_v + z_v a\sin\phi \\ z_v \\ 1 \end{bmatrix}$$



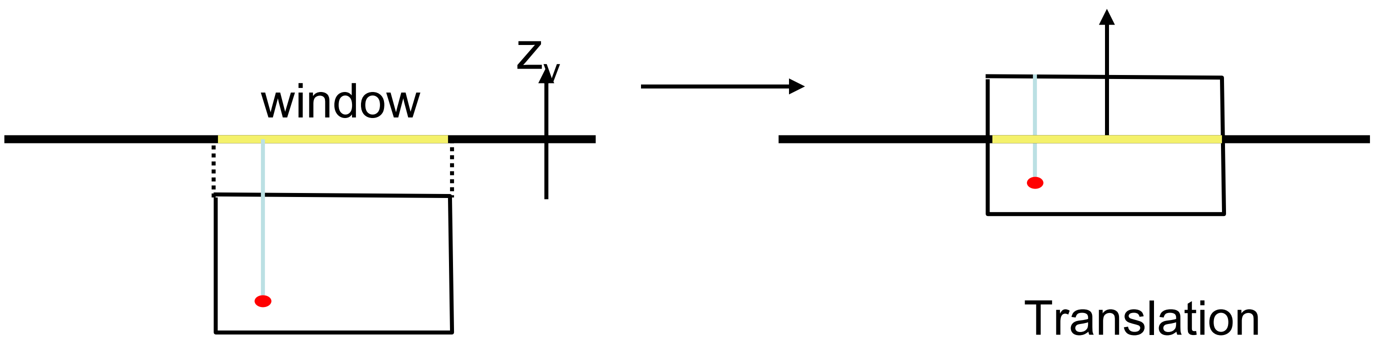
Oblique Projection

Shear

Canonical Volume: Parallel Projection

- Translation

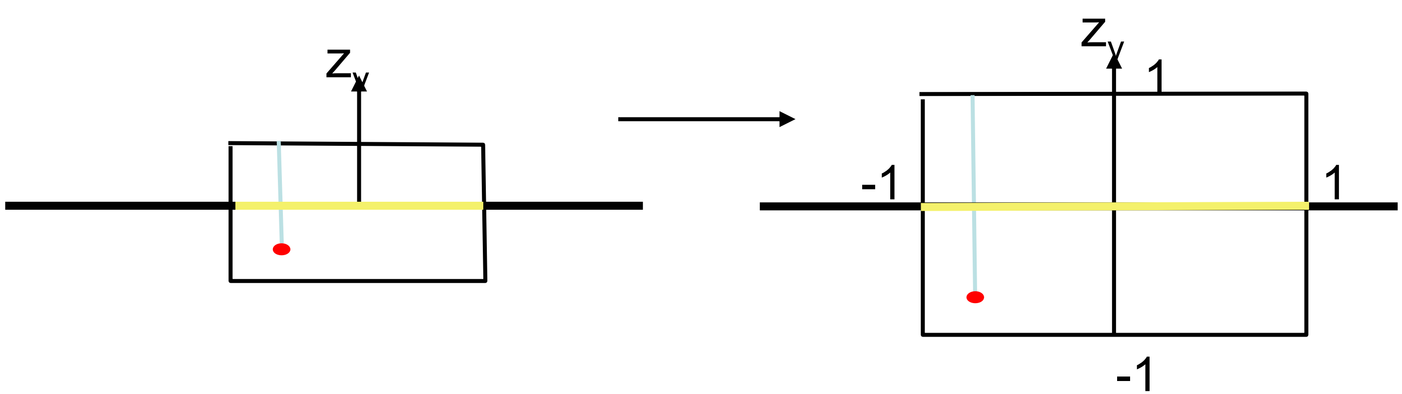
$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} x_c - \frac{r+l}{2} \\ y_c - \frac{t+b}{2} \\ z_c + \frac{f+n}{2} \\ 1 \end{bmatrix}$$



Canonical Volume: Parallel Projection

- Scaling

$$\begin{bmatrix} x''_c \\ y''_c \\ z''_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{-2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot x'_c}{r-l} \\ \frac{2 \cdot y'_c}{t-b} \\ \frac{-2 \cdot z'_c}{f-n} \\ 1 \end{bmatrix}$$



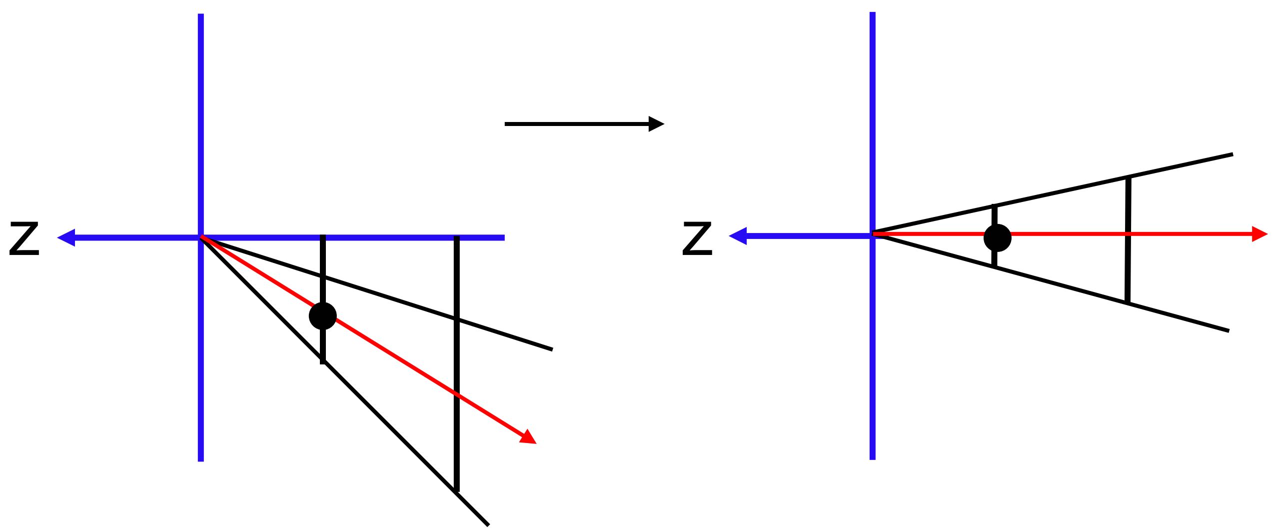
scaling

Canonical Volume: Perspective Projection

- Depth-preserving shear

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{r+l}{2n} & 0 \\ 0 & 1 & \frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

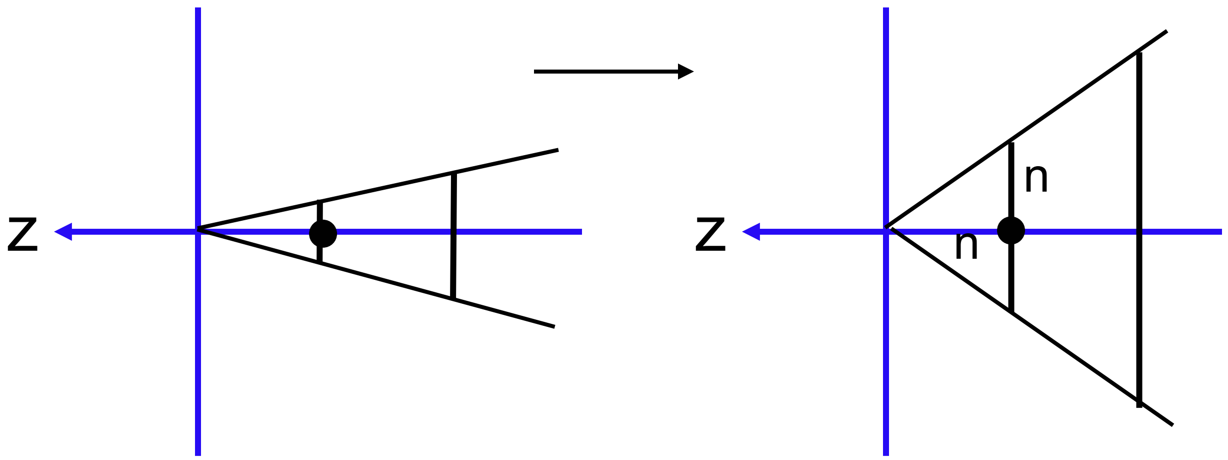
$$\begin{bmatrix} 1 & 0 & \frac{r+l}{2n} & 0 \\ 0 & 1 & \frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r+l}{2} \\ \frac{t+b}{2} \\ -n \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -n \\ 1 \end{bmatrix}$$



Canonical Volume: Perspective Projection

- Scaling

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$



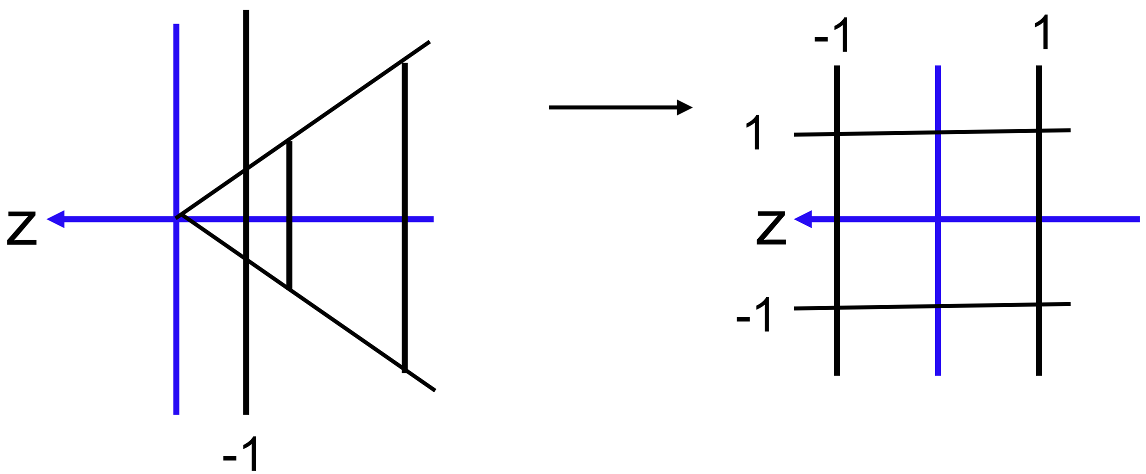
Canonical Volume: Perspective Projection

- Perspective transformation

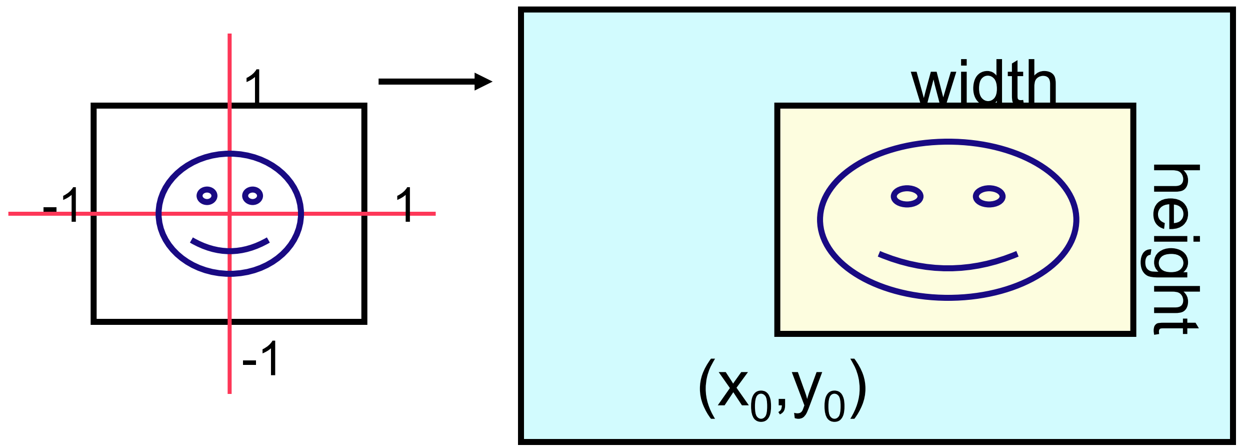
$$\begin{bmatrix} x''_c \\ y''_c \\ z''_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -2\frac{fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix}$$

$$\frac{x''}{1} = \frac{x'}{-z'} \Rightarrow x'' = \frac{x'}{-z'} \qquad \frac{y''}{1} = \frac{y'}{-z'} \Rightarrow y'' = \frac{y'}{-z'}$$

$$[0 \quad 0 \quad -n \quad 1] \to [0 \quad 0 \quad -1 \quad 1] \qquad [0 \quad 0 \quad -f \quad 1] \to [0 \quad 0 \quad 1 \quad 1]$$



Viewport Transformation



$$\begin{pmatrix} x_{win} \\ y_{win} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{width}{2} & 0 & 0 \\ 0 & \frac{height}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c'' \\ y_c'' \\ 1 \end{pmatrix}$$

glViewport(x_0, y_0, width, height) ;
/* You can have multiple viewports on your window */