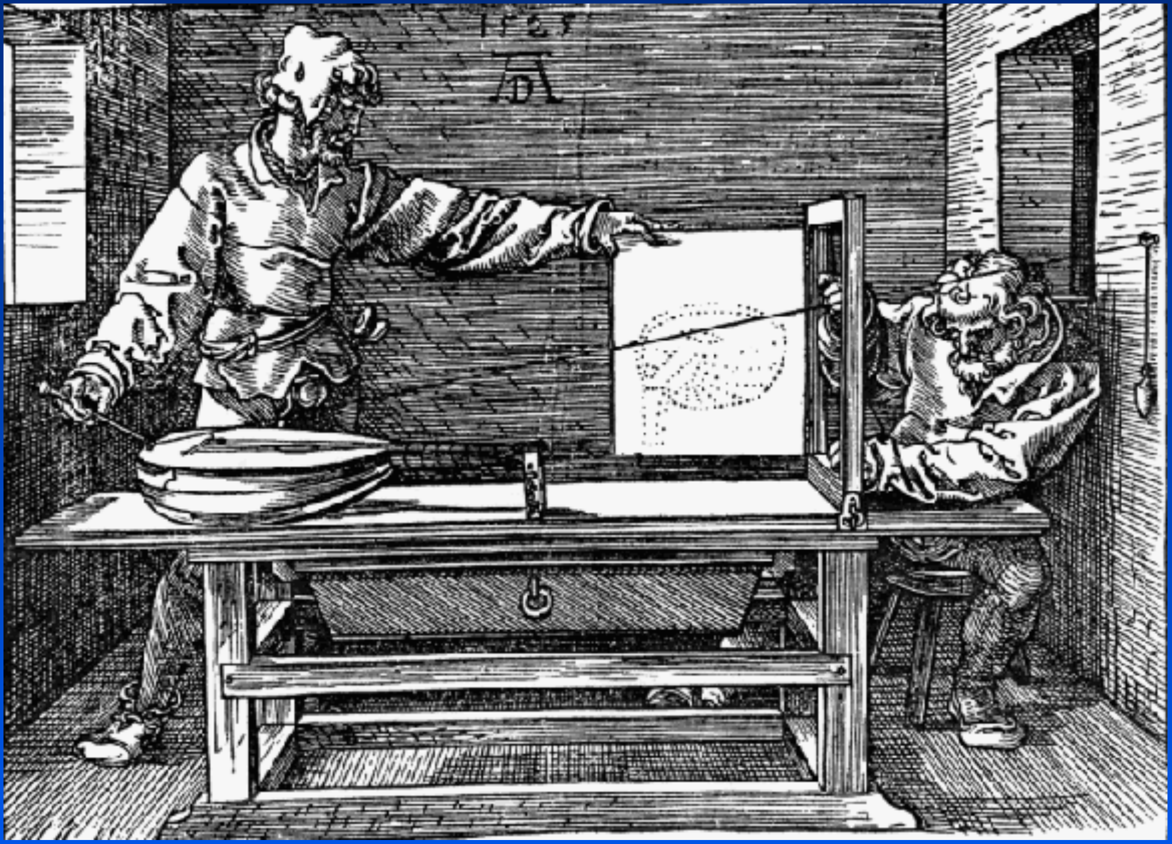
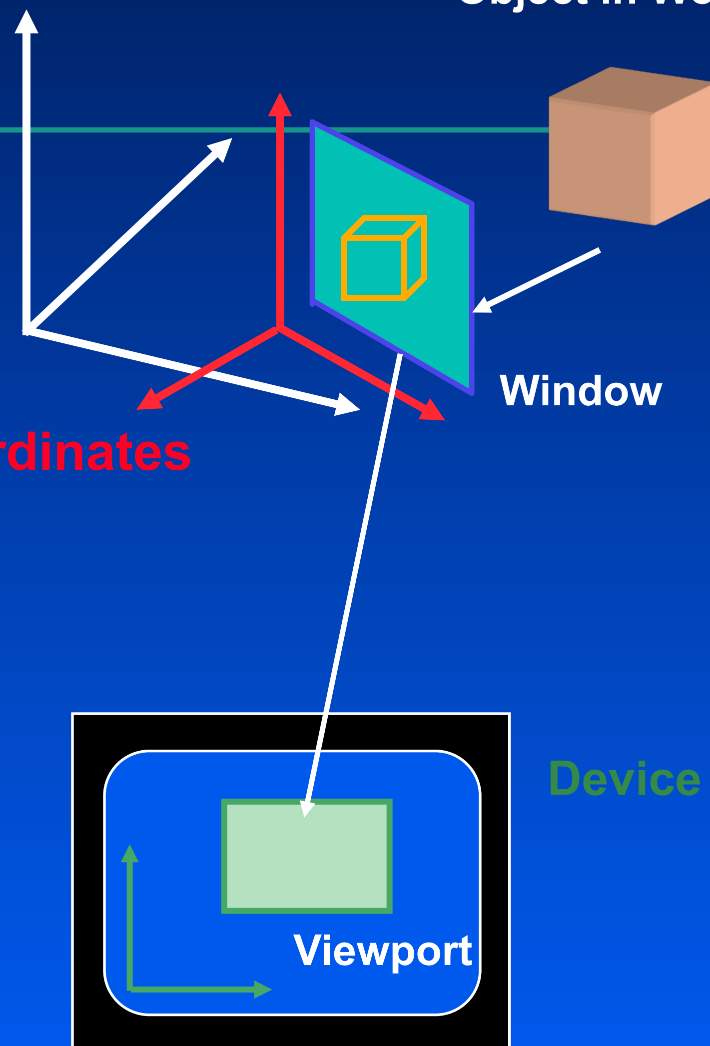
Viewing Transform

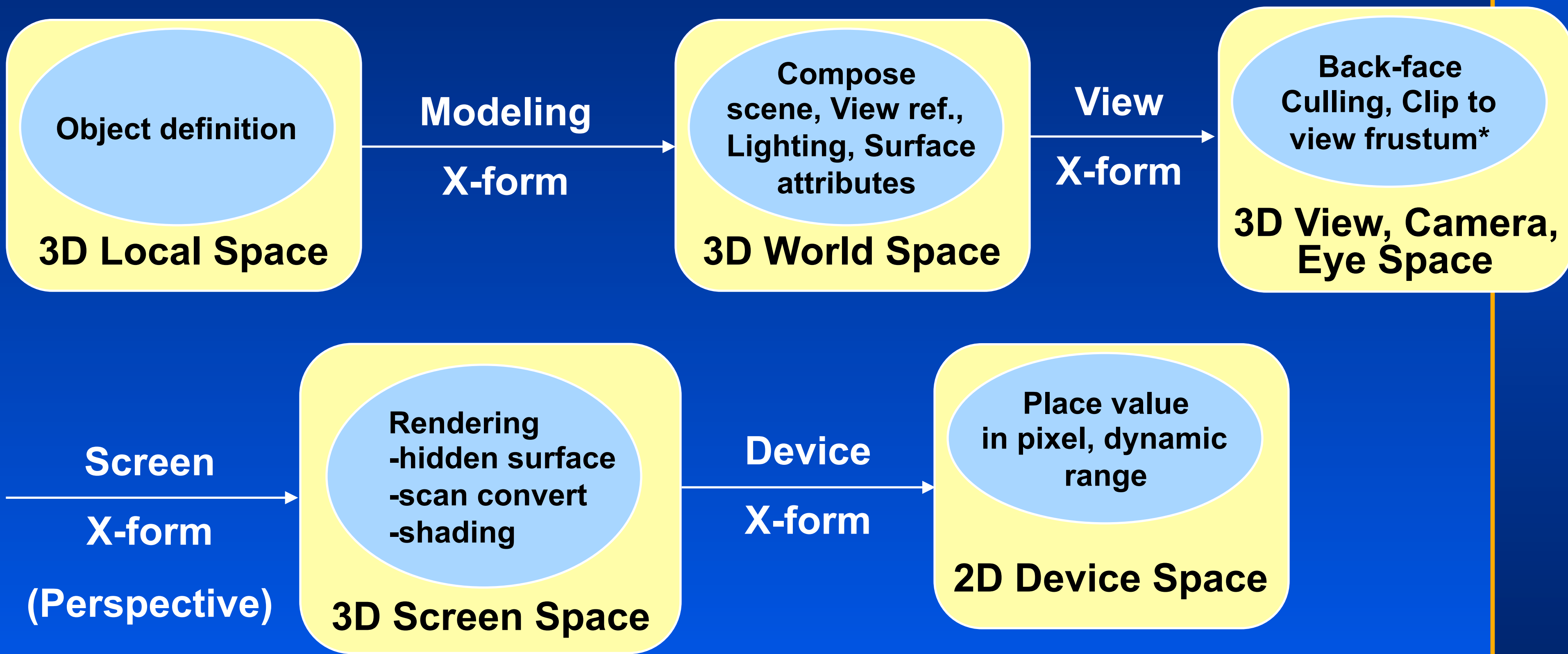


3D Viewing



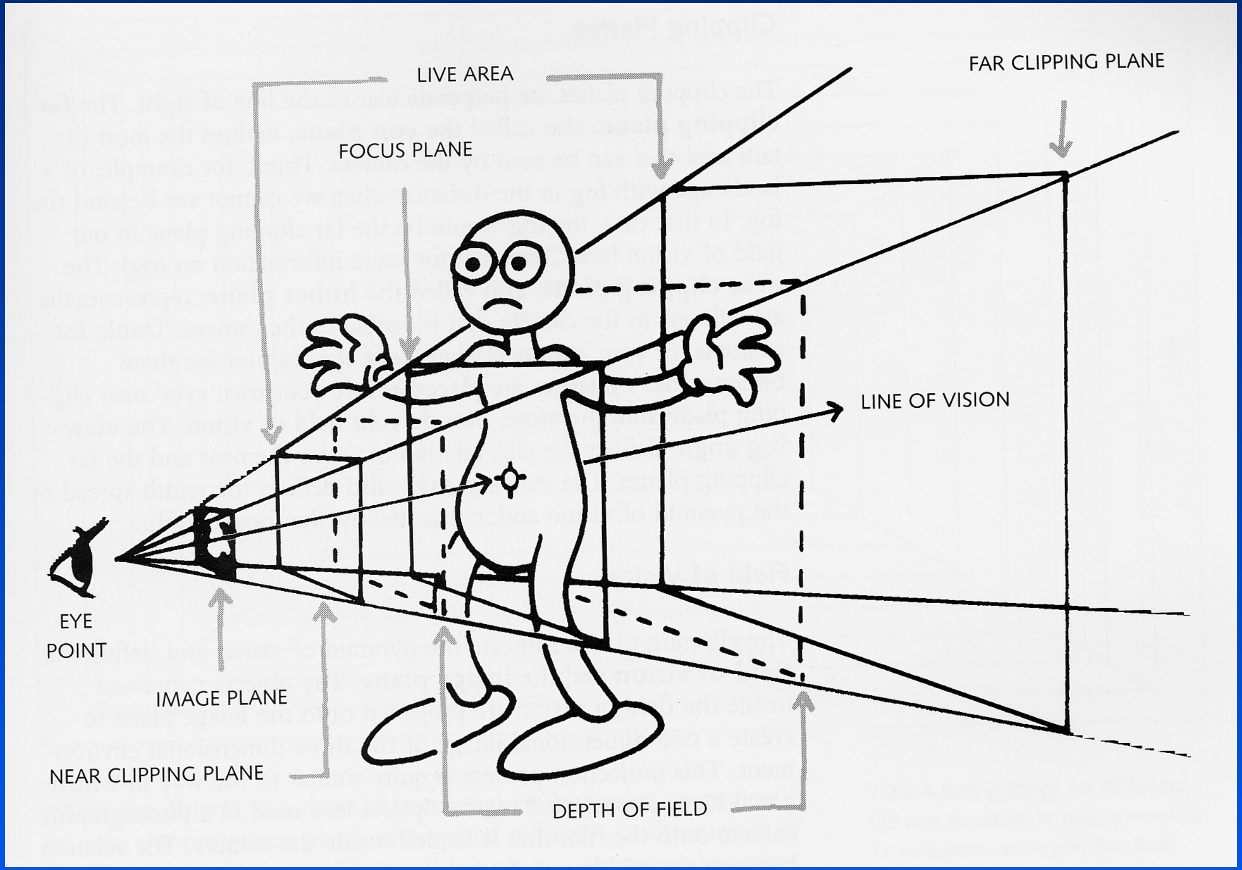
Daniel Keren

Rendering Pipeline: Spaces and Transformations



* Clipping can be done more simply in screen space

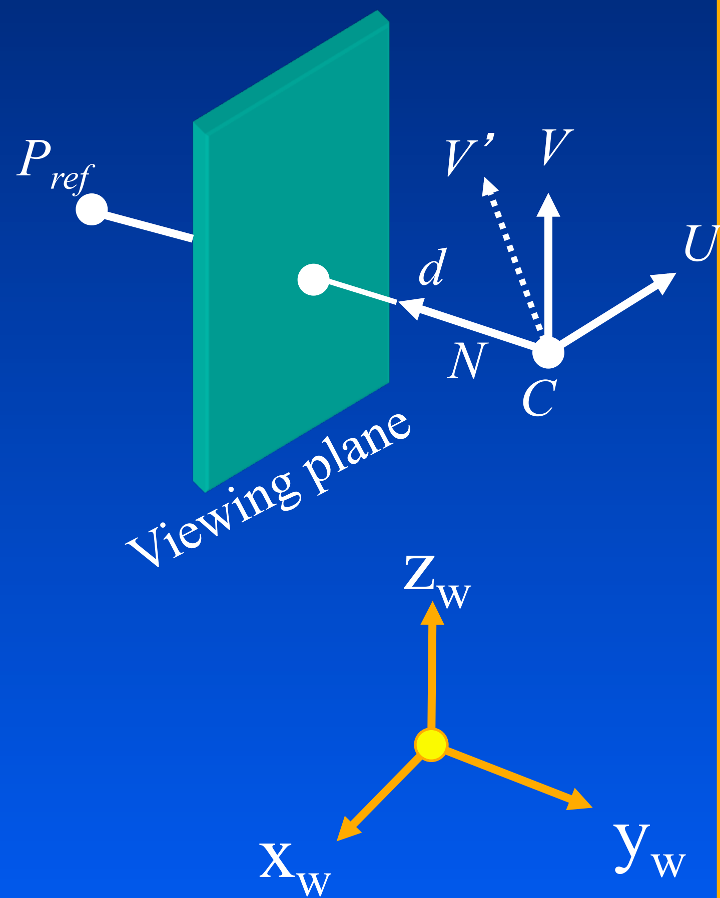
"Internal" camera parameters (about the camera itself)



Kerlow

“External” camera parameters (defined WRT world coordinate)

- Camera position C:
  center of projection and/or origin
  of camera coordinate system
  (this is used in book)
- Viewing direction N (positive z-axis)
  - Normal to view plane
- V : Y-direction of camera
  (Up vector direction of camera
  defined by V' the *up vector*)
- U vector for X-direction of camera
- View plane parallel to U-V plane



To define U,V,N

- Need to translate user defined quantities of C, P_{ref}, and V' into U, V, and N

$$N = \frac{P_{ref} - C}{\left|P_{ref} - C\right|} \quad ; \quad U = \frac{N \times V'}{\left|N \times V'\right|} \quad ; \quad V = U \times N$$

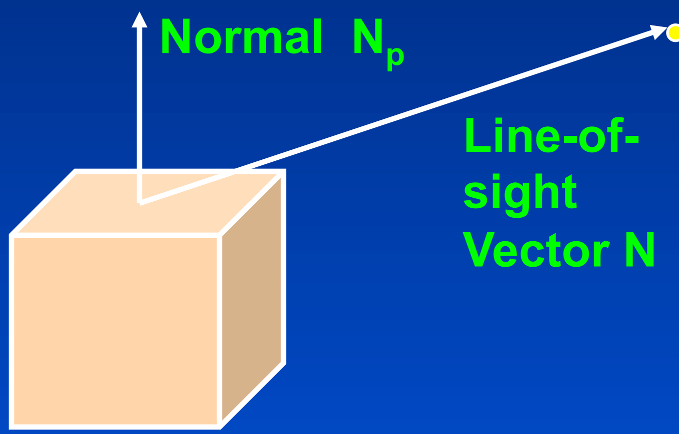
- Note the left handed coordinate system used here

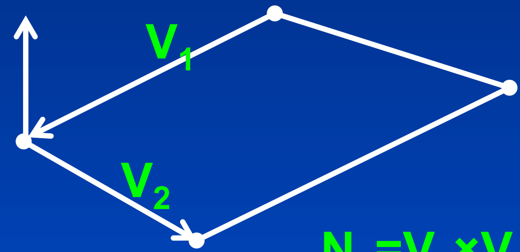
View X-Form from world space to camera space

- Translate so that viewing coordinate origin is coincident with world coordinate
- Rotate so that the axes coincide
  - Use the fact that rotation matrices are special orthogonal

$$M_{view} = \begin{bmatrix} U_x & U_y & U_z & 0 \\ V_x & V_y & V_z & 0 \\ N_x & N_y & N_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & -C_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = R \cdot T$$

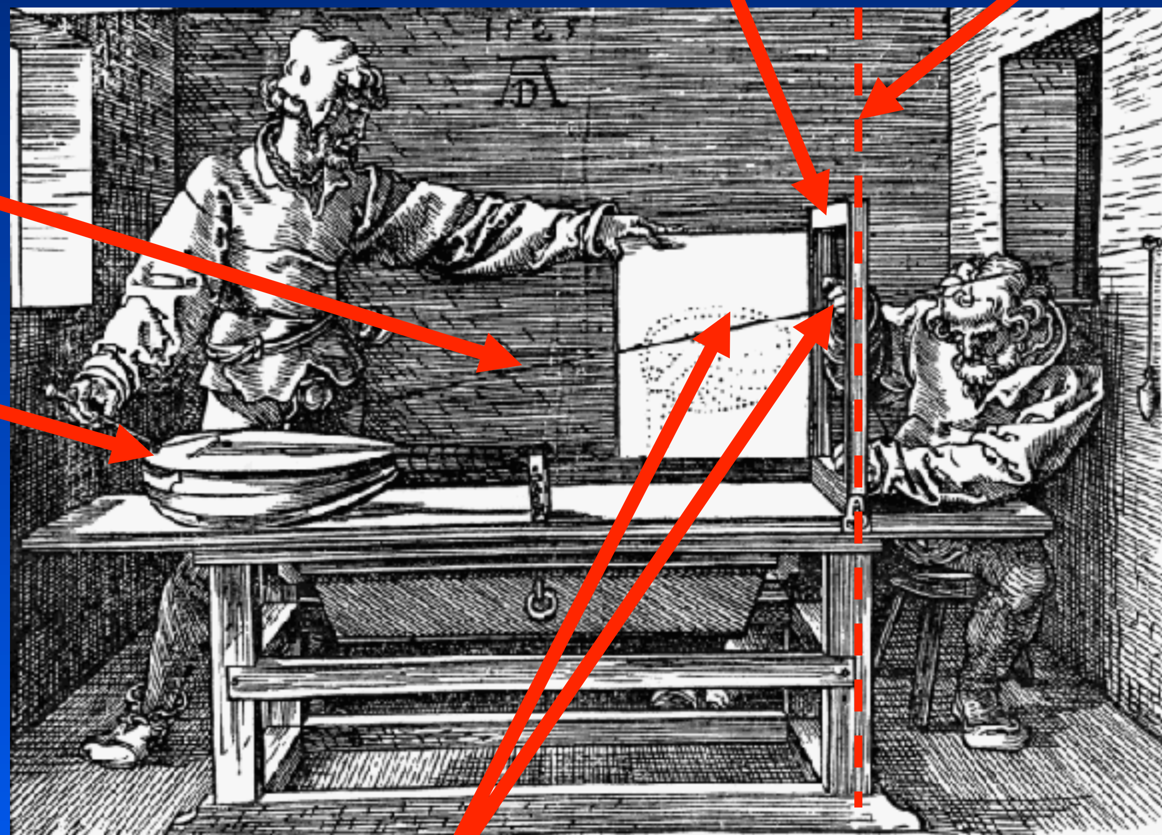
Back-face culling





- Line -of-sight Vector from any point on the plane to view point
- Visible if $N_p \cdot N > 0$
- Simpler after perspective (in screen space) since line-of-sight vector is parallel to Z-axis but need to remove as early as possible

(Oblique) Perspective projection transformation



Clipping window

View plane

Center of projection
(Projection ref. pt.)

Projector

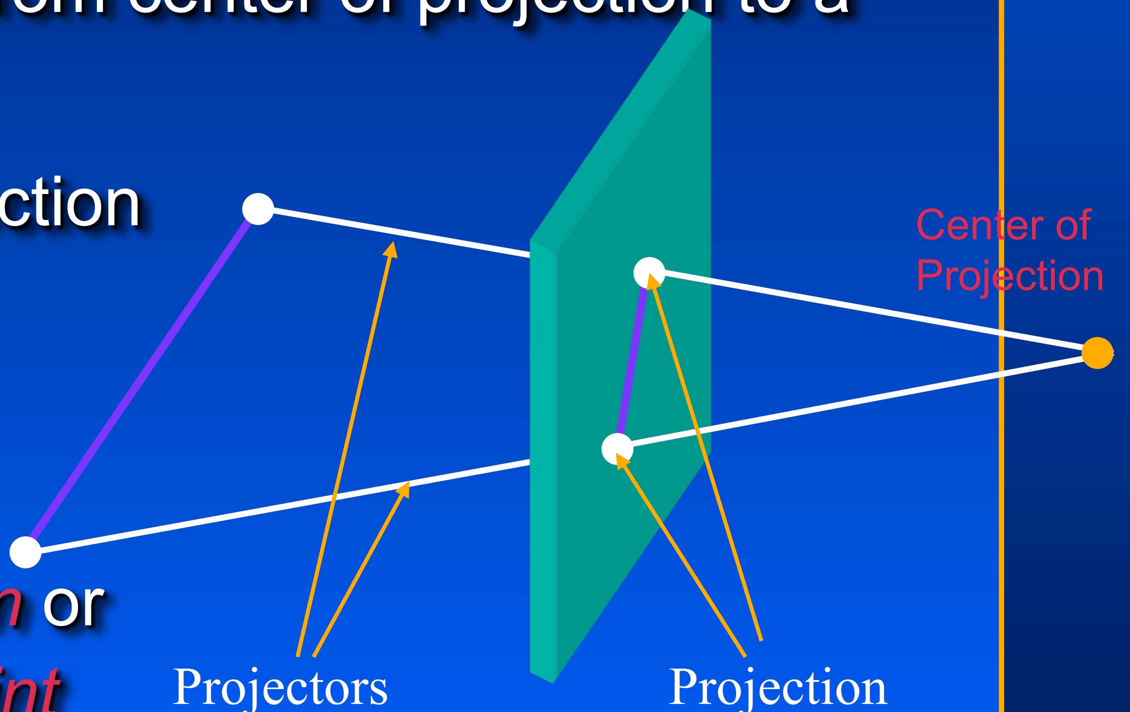
(x,y,z)

(x_p, y_p, z_p)

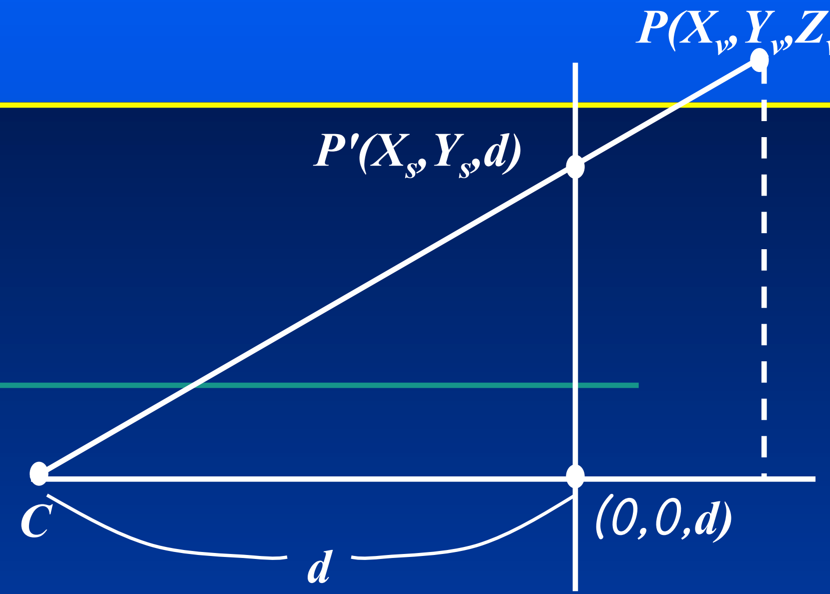
Albrecht Dürer (1529)

Projections

- Going from 3D to 2D require projection
- Projectors are vectors from center of projection to a point in the 3D world
- Projection is the intersection of a projector with the viewing plane
- Projectors emanate from *center of projection* or *projection reference point*



Screen Space



- Perspective transformation (use similar triangles)

$$x_s = \frac{d \cdot x_v}{z_v} = \frac{x_v}{z_v/d}, \quad y_s = \frac{d \cdot y_v}{z_v} = \frac{y_v}{z_v/d}, \quad z_s = d$$

- Corresponding matrix

$$M_{pers} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

- For a point (x_v, y_v, z_v):

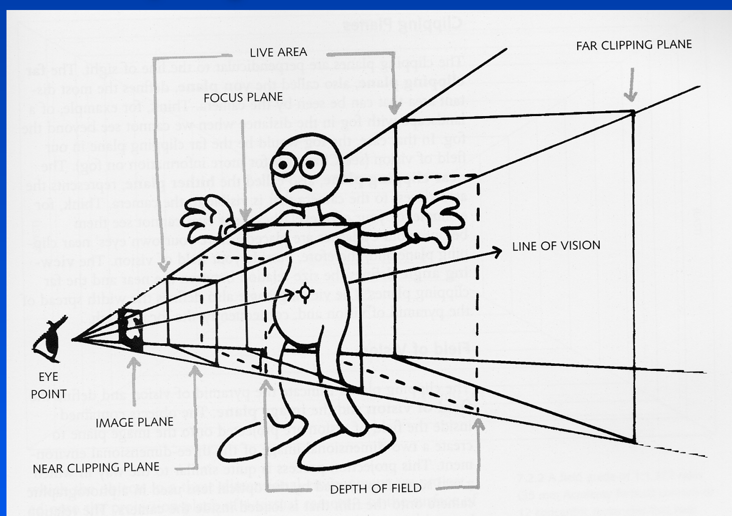
$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

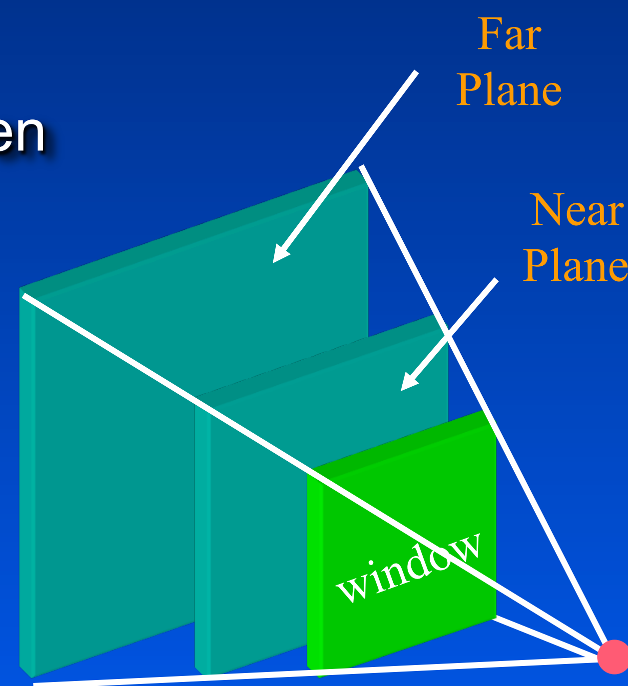
- In 3D (after divide by W):

$$\left(\frac{x}{z/d}, \quad \frac{y}{z/d}, \quad d \right)$$

View Volume

- View frustum defined by far and near clipping planes, clipping window, plus center of projection
- What is inside the view volume is seen and projected





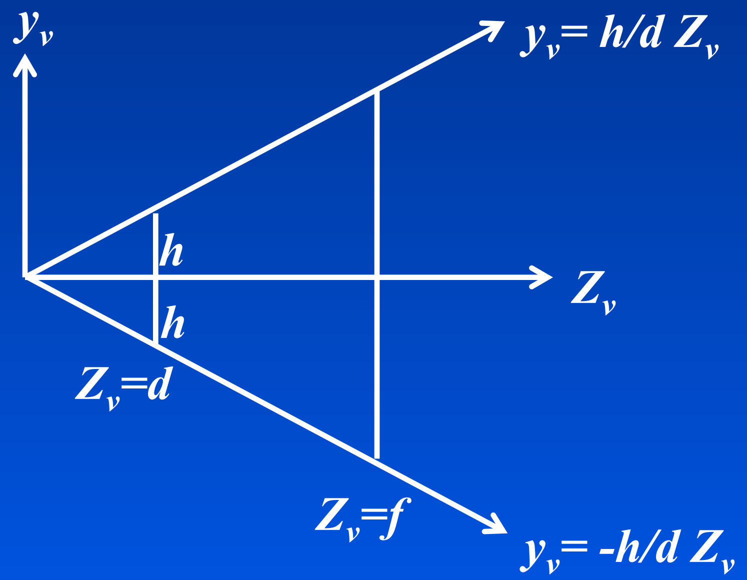
Viewing frustum♪

$$x_v = \pm \frac{h}{d} z_v$$

$$y_v = \pm \frac{h}{d} z_v$$

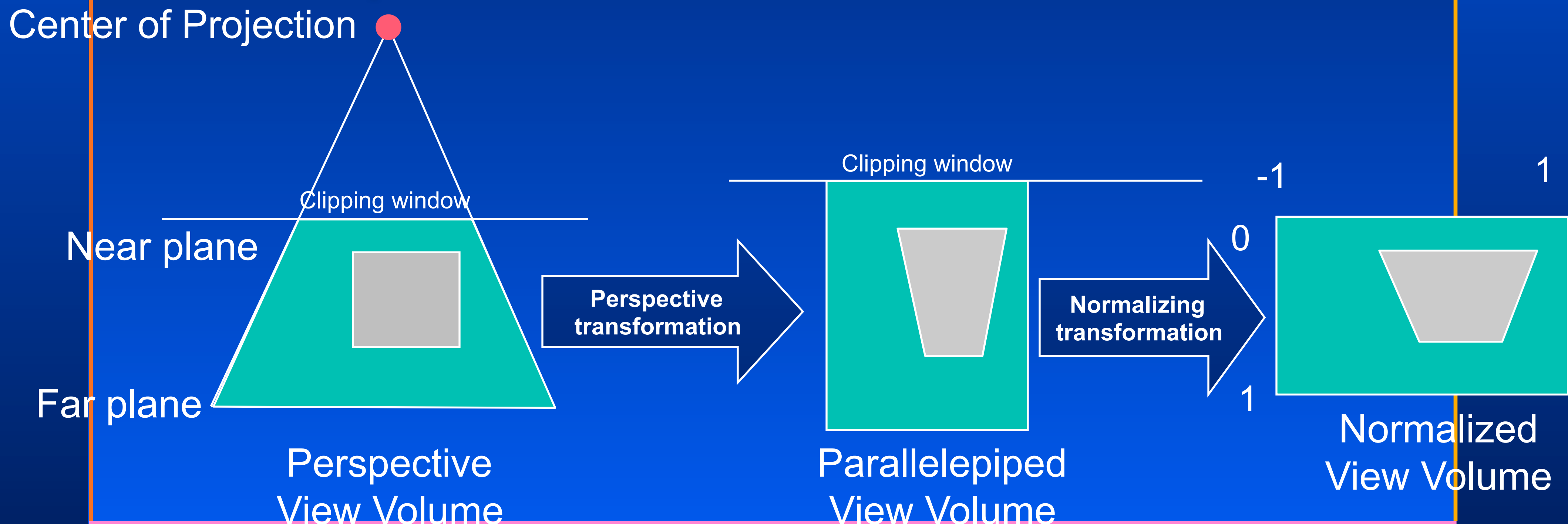
$z_v = d$ near clipping plan

$z_v = f$ far clipping plan

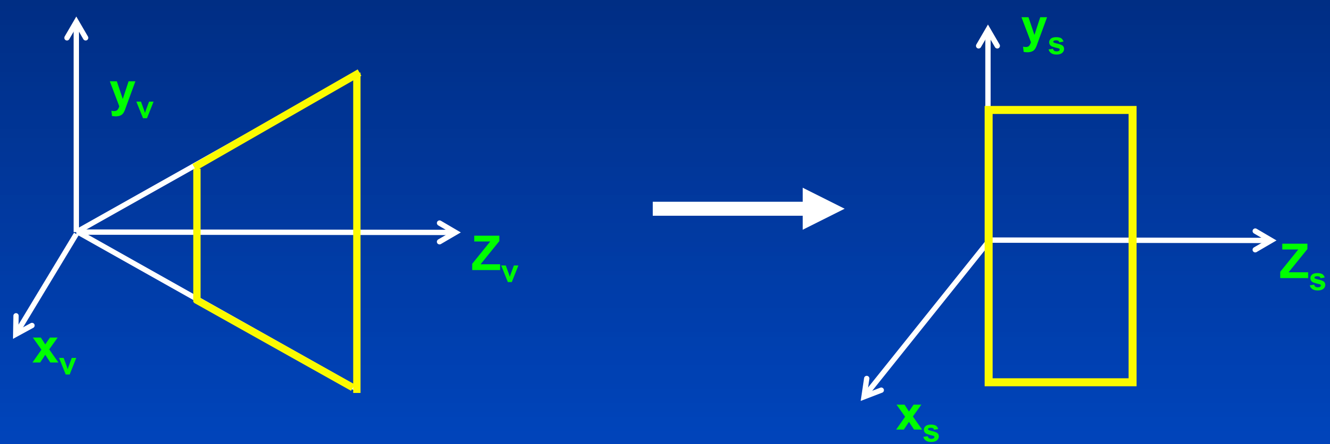


Effect of perspective projection transformation on view volume

- M_{pers} takes the perspective view volume and converts into the parallel view volume



Perspective transformation and hidden surface removal



- After perspective x-form, line-of-sight parallel to each other
- Hidden surface calculation on point with same x_s and y_s just compare their z_s
- Back face culling:
  z_s of normal vector > 0 $\Rightarrow$ Back facing

Derivation of Z_s: Transformation of viewing frustum

- Need Z_s to compare depths of objects
- Line transform to line and plane transform to plane in going from eye to screen space (can be shown)

$$\Rightarrow \quad z_s = A \pm \frac{B}{z_v}$$ A and B constants

- Choose B < 0 since z_v increases $\Rightarrow z_s$ increases
- $z_v \in [d \quad f]$ maps into $z_s \in [0 \quad 1]$

Perspective transform

- Perspective foreshortening plus transformation of truncated pyramid into canonical parallel view volume

$$x_s = d\,\frac{x_v}{h z_v} \qquad y_s = d\,\frac{y_v}{h z_v} \qquad z_s = f\,\frac{1 - d/z_v}{f - d}$$

$$x_s,\ y_s \in [-1,\ \ 1]\ ,\ \ z_s \in [0,\ \ 1]$$

- In homogeneous coordinates

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_{pers} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

$$M_{pers} = \begin{bmatrix} d/h & 0 & 0 & 0 \\ 0 & d/h & 0 & 0 \\ 0 & 0 & f/(f-d) & -df/(f-d) \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$x_s = X/W$$
$$y_s = Y/W$$
$$z_s = Z/W$$

Summary of transformations

- Model to screen:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_{pers} M_{view} M_{model} \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix}$$

- Device transformation takes x, y values between -1 and 1 and scales/translates to device dimensions

Clipping

- For reason of efficiency: put through rendering pipeline only those inside viewing frustum
- Transformation ill-defined outside viewing frustum
  - Objects in the negative w space map to viewing frustum
  - Singularity at z=0
- Clip in 4-D space
  - Before division by w

Clipping against viewing frustum

- Viewing frustum in homogeneous coordinates:

$$-W \le X \le W$$

$$-W \le Y \le W$$

$$0 \le Z \le W$$

- Sutherland-Hodgman algorithm
  - Each polygon clipped against each of the clipping rectangles in view volume

Next: Shading