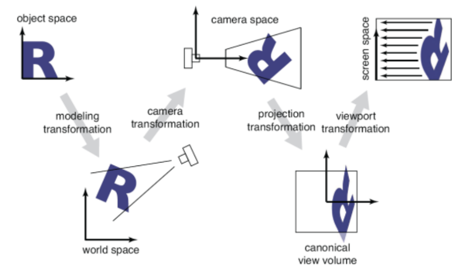
Perspective

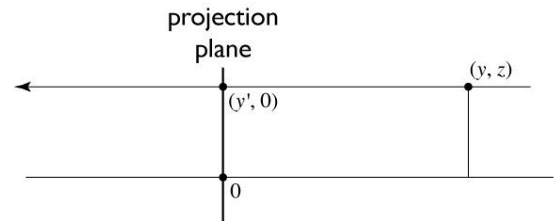
COMP 557

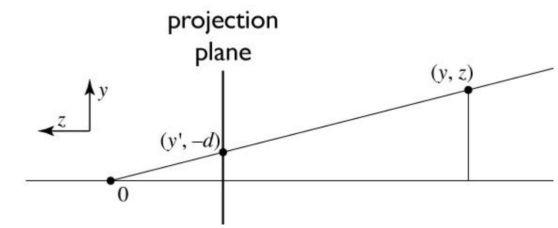
Paul Kry

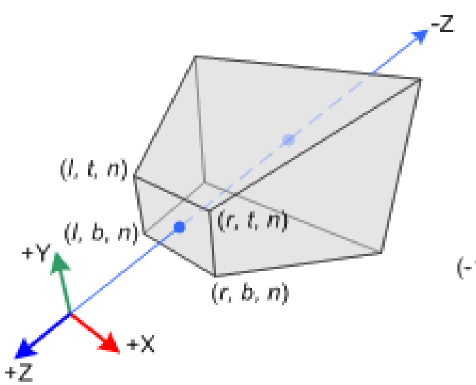
Review

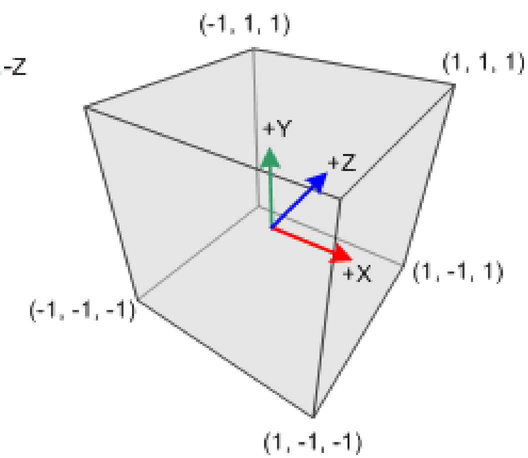
- Pipeline of transformations
- Windowing transformations
- Orthographic projection matrix
- Perspective projection matrix
- Canonical view volume, Normalized Device Coordinates
- What goes where after perspective projection?



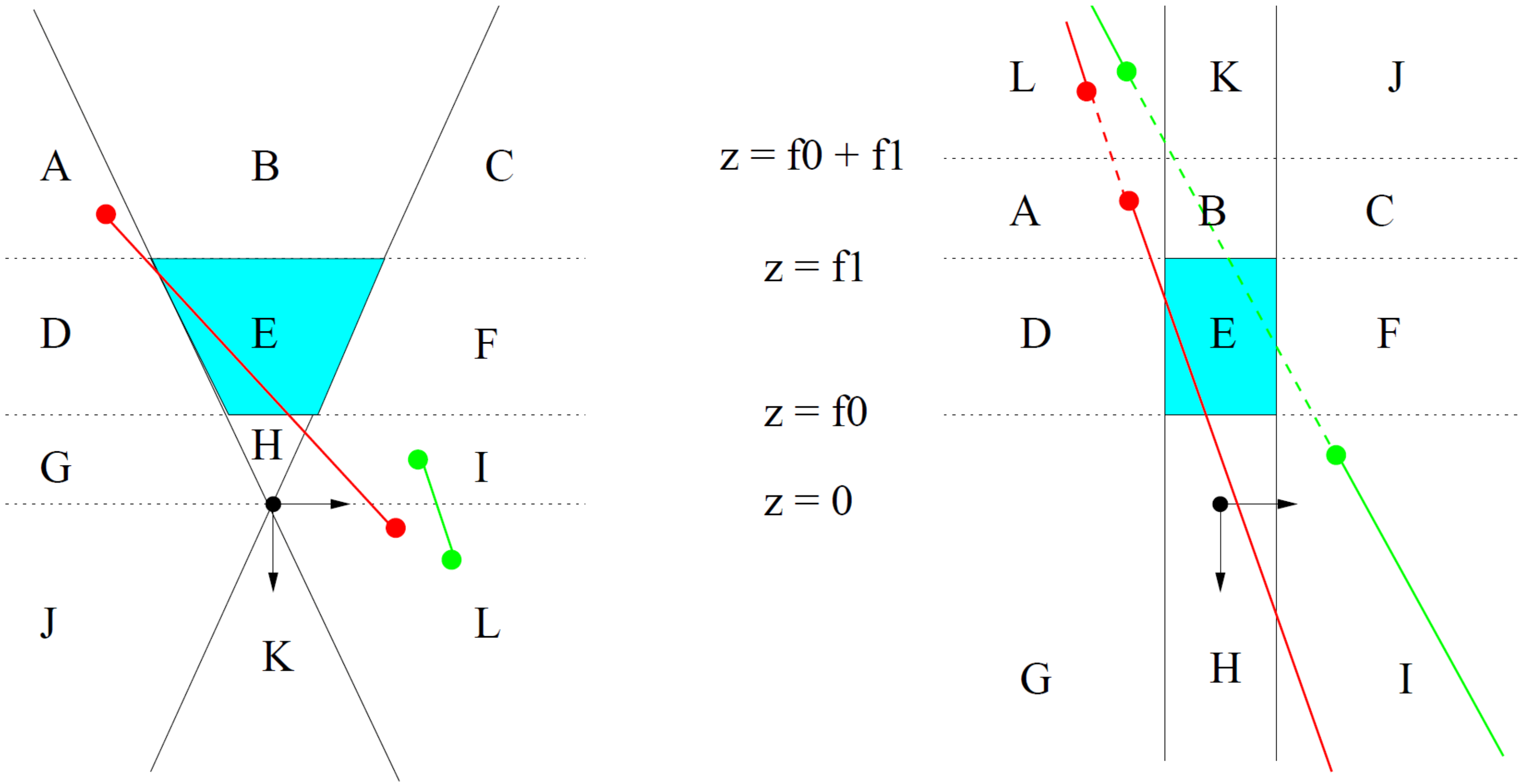








What goes where?



Camera coordinates (non-normalized) projection coordinate

Notice implications for clipping (i.e., discarding) geometry!

Recall: Perspective transformation chain

- Transform into world coords (modeling transform, M_m)
- Transform into eye coords (camera xf., $M_{\text{cam}} = F_c^{-1}$)
- Perspective matrix, P
- Orthographic projection, M_{orth}
- Viewport transform, M_{vp}

$$\mathbf{p}_s = \mathbf{M}_{\text{vp}}\mathbf{M}_{\text{orth}}\mathbf{P}\mathbf{M}_{\text{cam}}\mathbf{M}_{\text{m}}\mathbf{p}_o$$

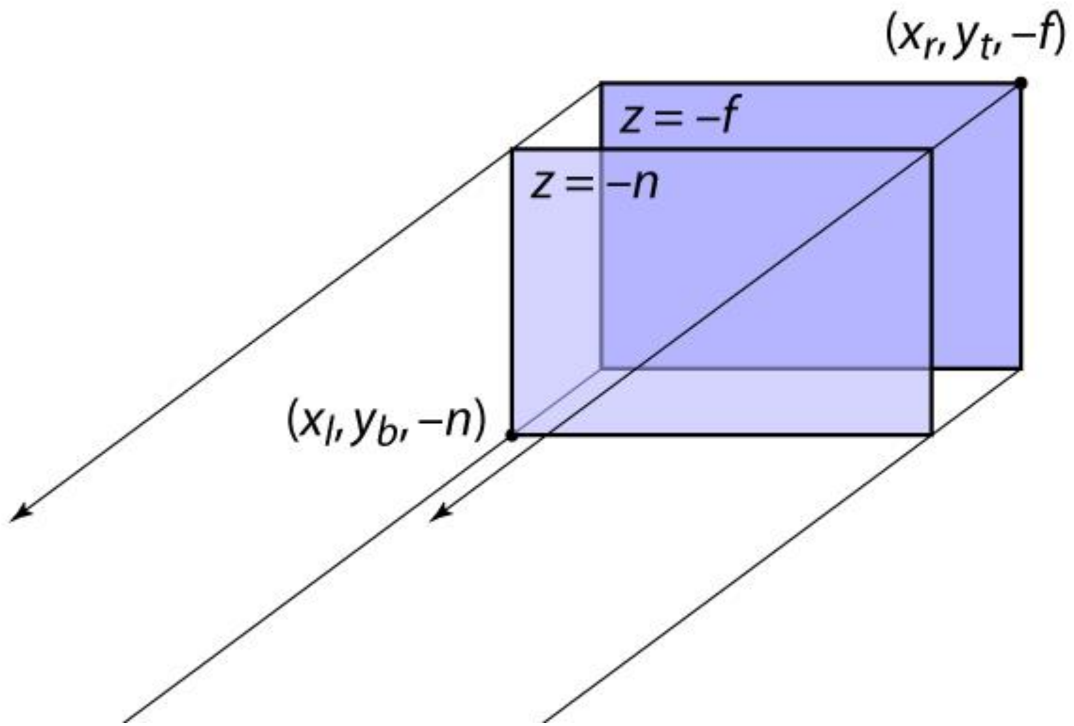
$$\begin{bmatrix} x_s \\ y_s \\ z_c \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x-1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \boxed{\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}} \boxed{\mathbf{M}_{\text{cam}}\mathbf{M}_{\text{m}}} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

OpenGL set with glViewport

OpenGL projection matrix

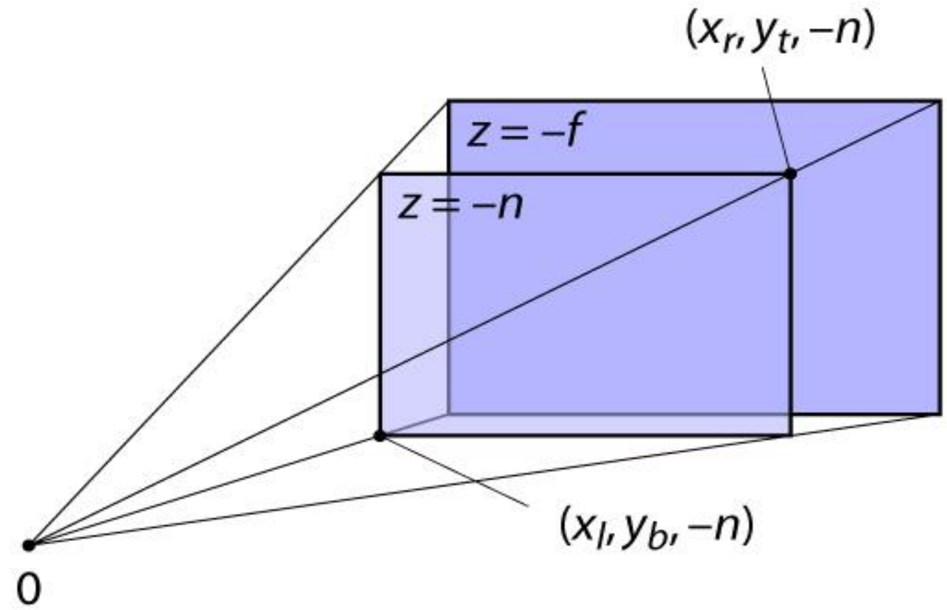
OpenGL modelview matrix

OpenGL view frustum: orthographic



Note OpenGL puts the near and far planes at $-n$ and $-f$ so that the user can give positive numbers

OpenGL view frustum: perspective



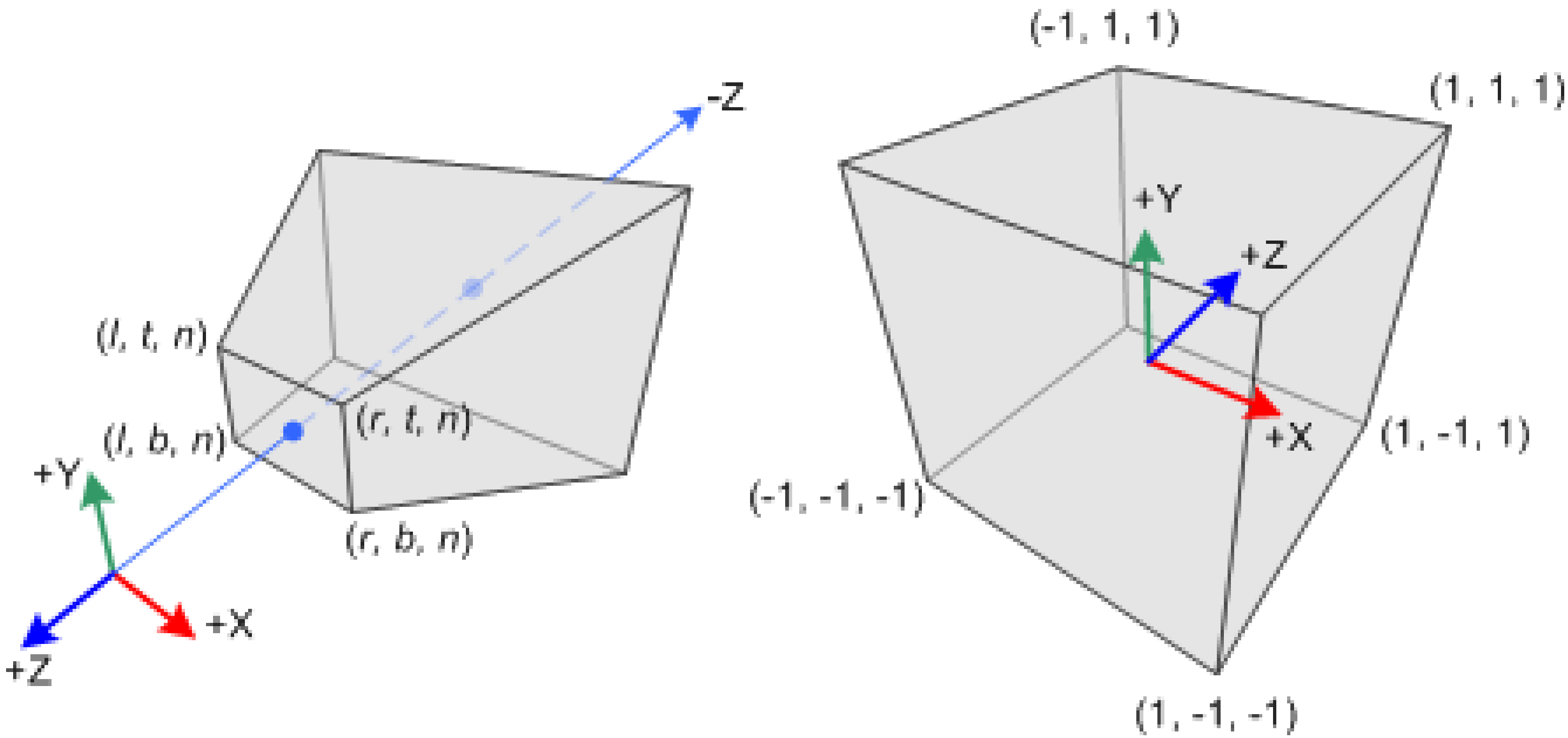
Note OpenGL puts the near and far planes at $-n$ and $-f$ so that the user can give positive numbers

gluPerspective(fovy, aspect, near, far)

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

Defined on near plane

Frustum to Normalized Device Coordinates (NDC)



Note switch to left handed coordinates!

Frustum applications (explained on blackboard)

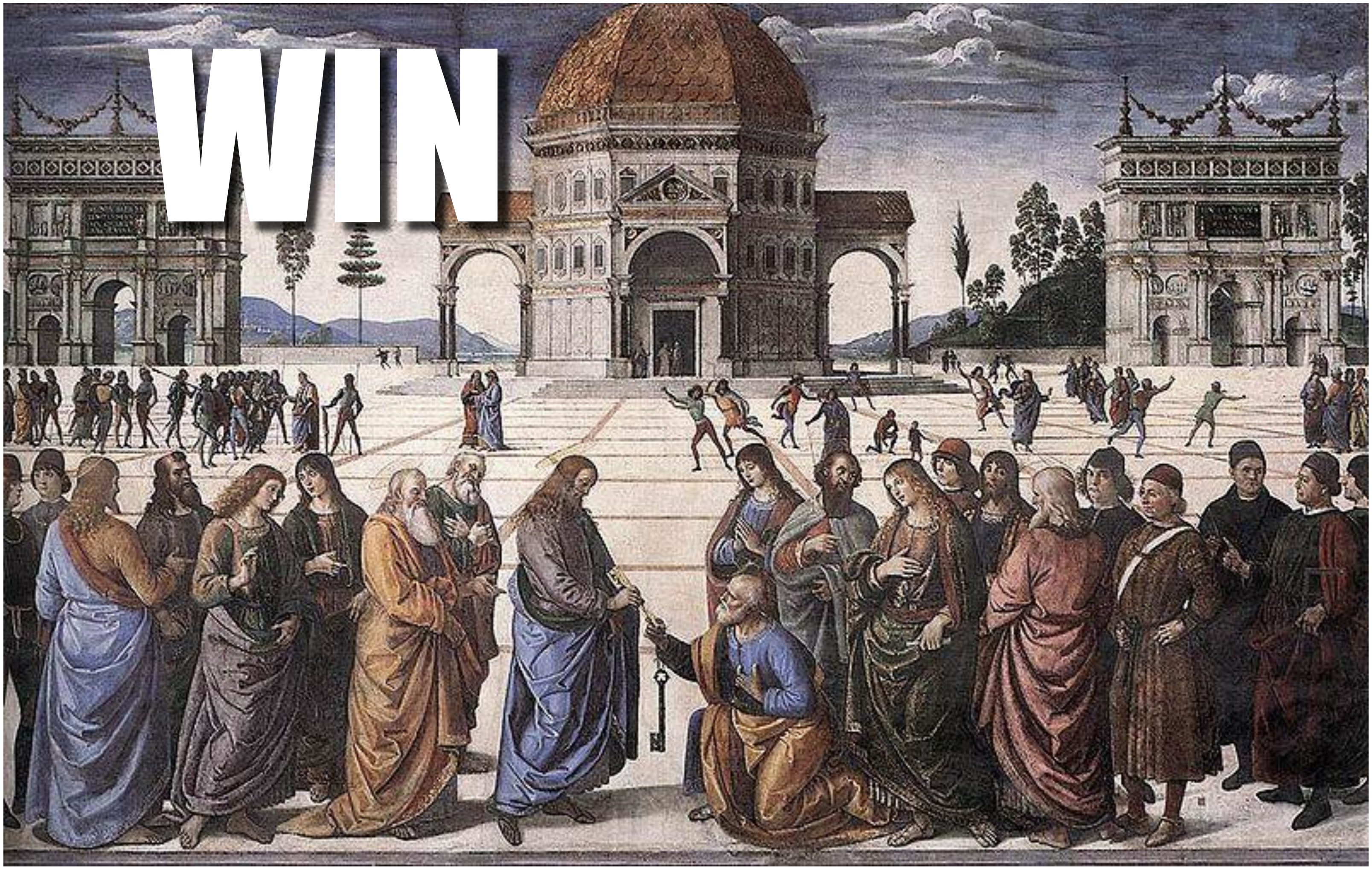
- Tiled rendering
  - Render very high resolutions
- 3D viewing
  - Left eye right eye
- Depth of field
  - Accumulating multiple render passes
- Shifted perspective (more later)

History of Projection

Let's look at examples in art…

15th century illustration from the Old French translation of William of Tyre's Histoire d'Outremer

Pietro Perugino, fresco at the Sistine Chapel (1481–82)



History of projection

- Ancient times: Greeks wrote about laws of perspective
- Renaissance: perspective is adopted by artists

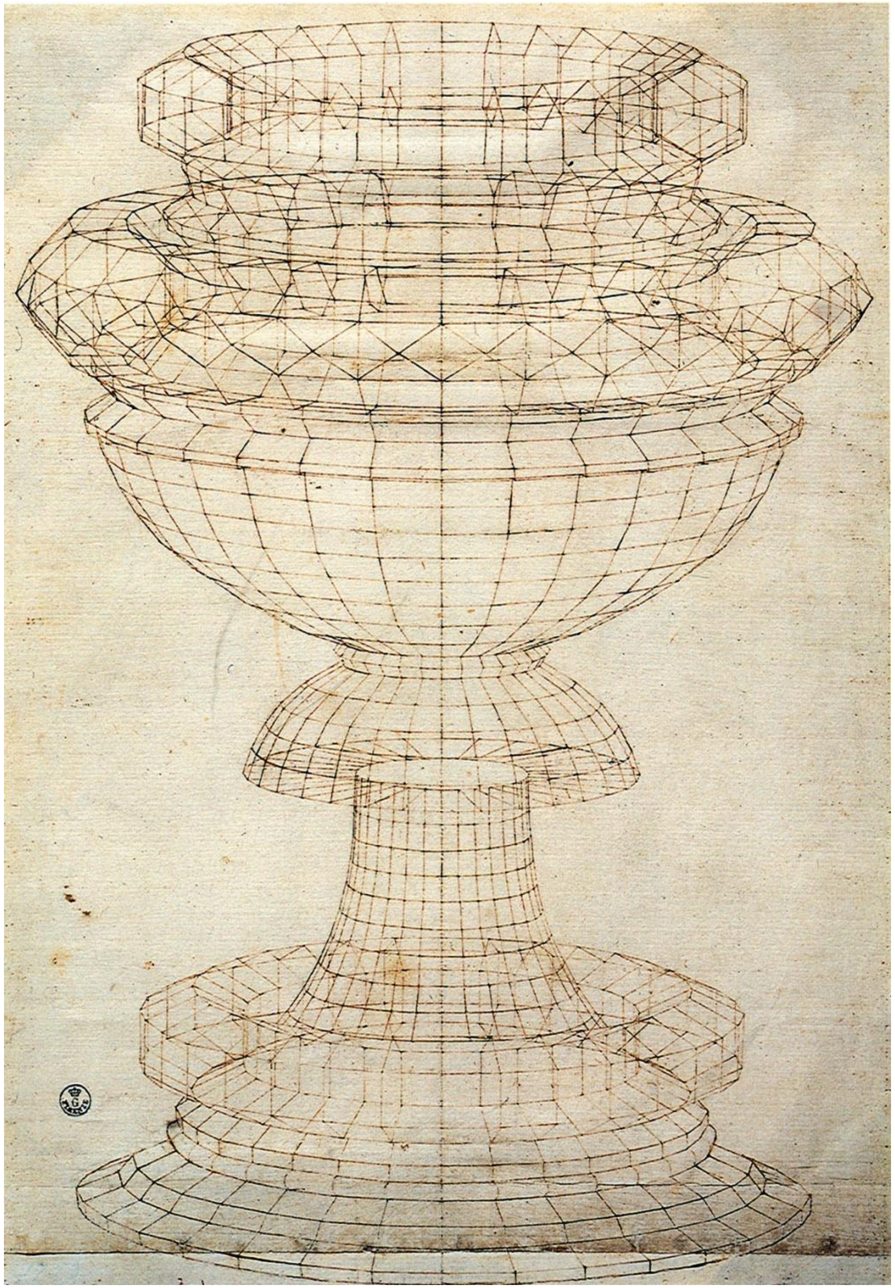
Duccio c. 1308

History of projection

- Later Renaissance: perspective formalized precisely

da Vinci c. 1498

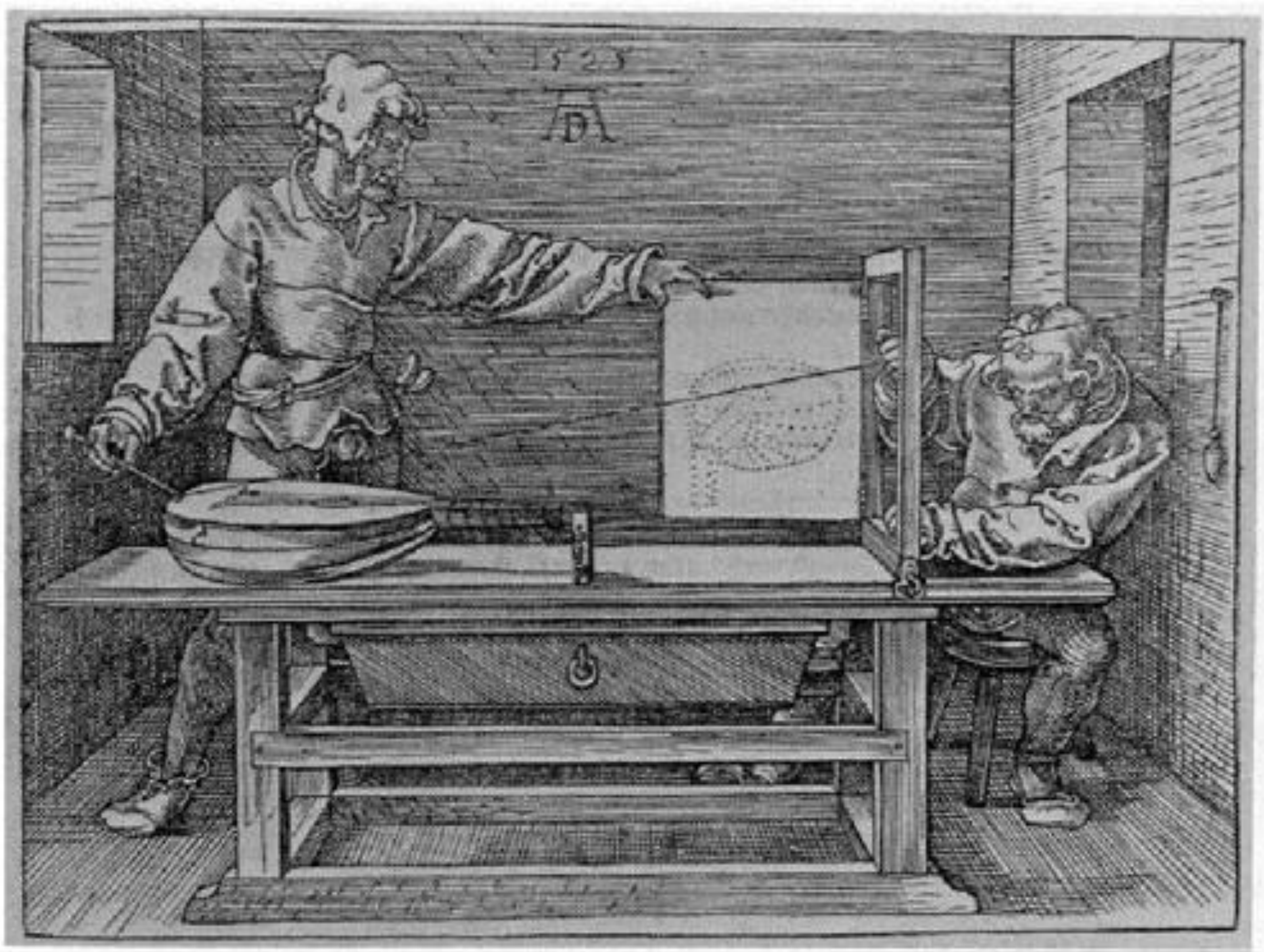
Paolo Uccello (1397-1475)



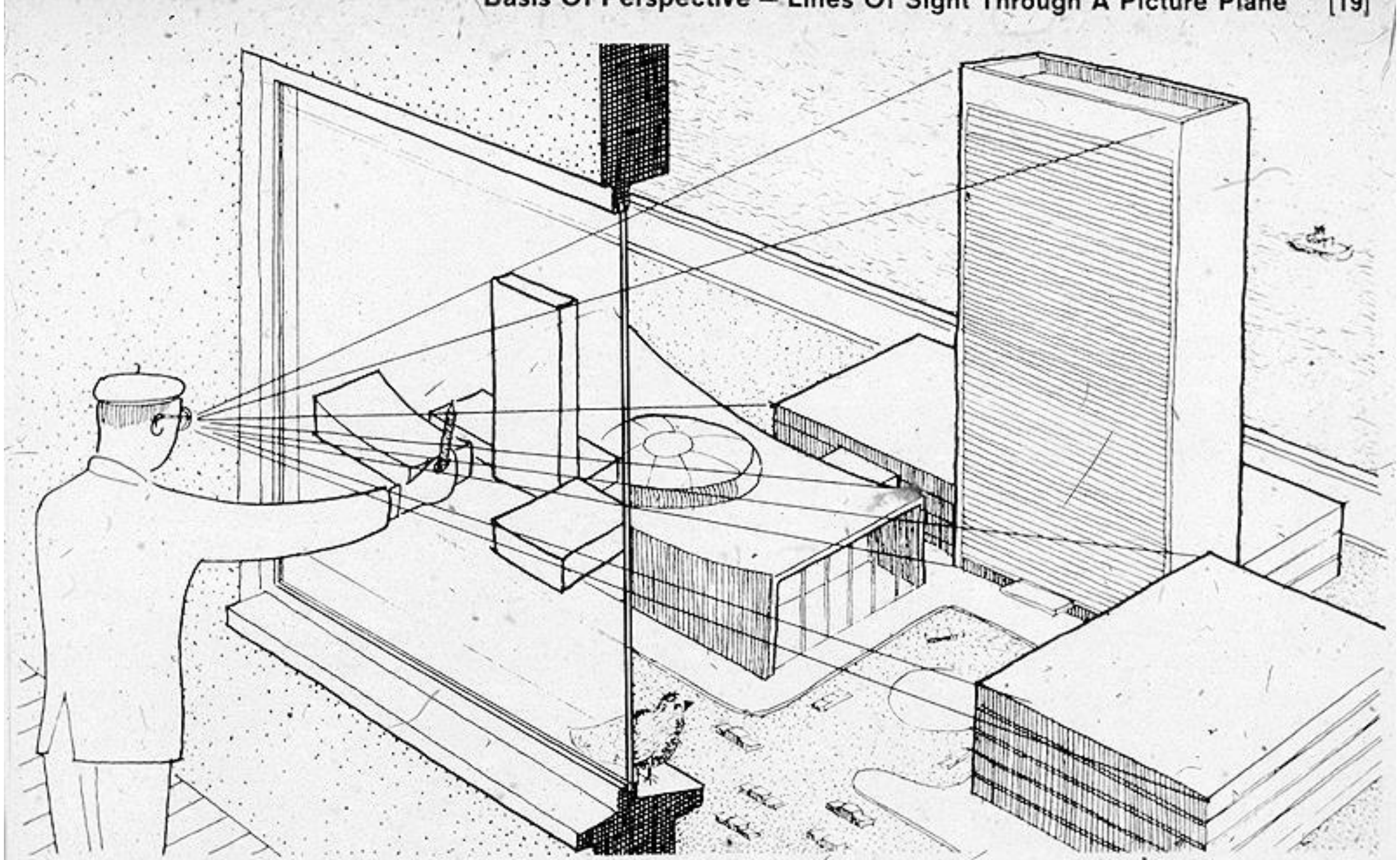
Plane projection in drawing

Artist Drawing a Lute, woodcut from Unterweysung der Messung mit dem Zyrkel und Rychtscheyd, 1525

[Carlbom & Paciorek 78 http://doi.acm.org/10.1145/356744.356750]



Plane projection in drawing

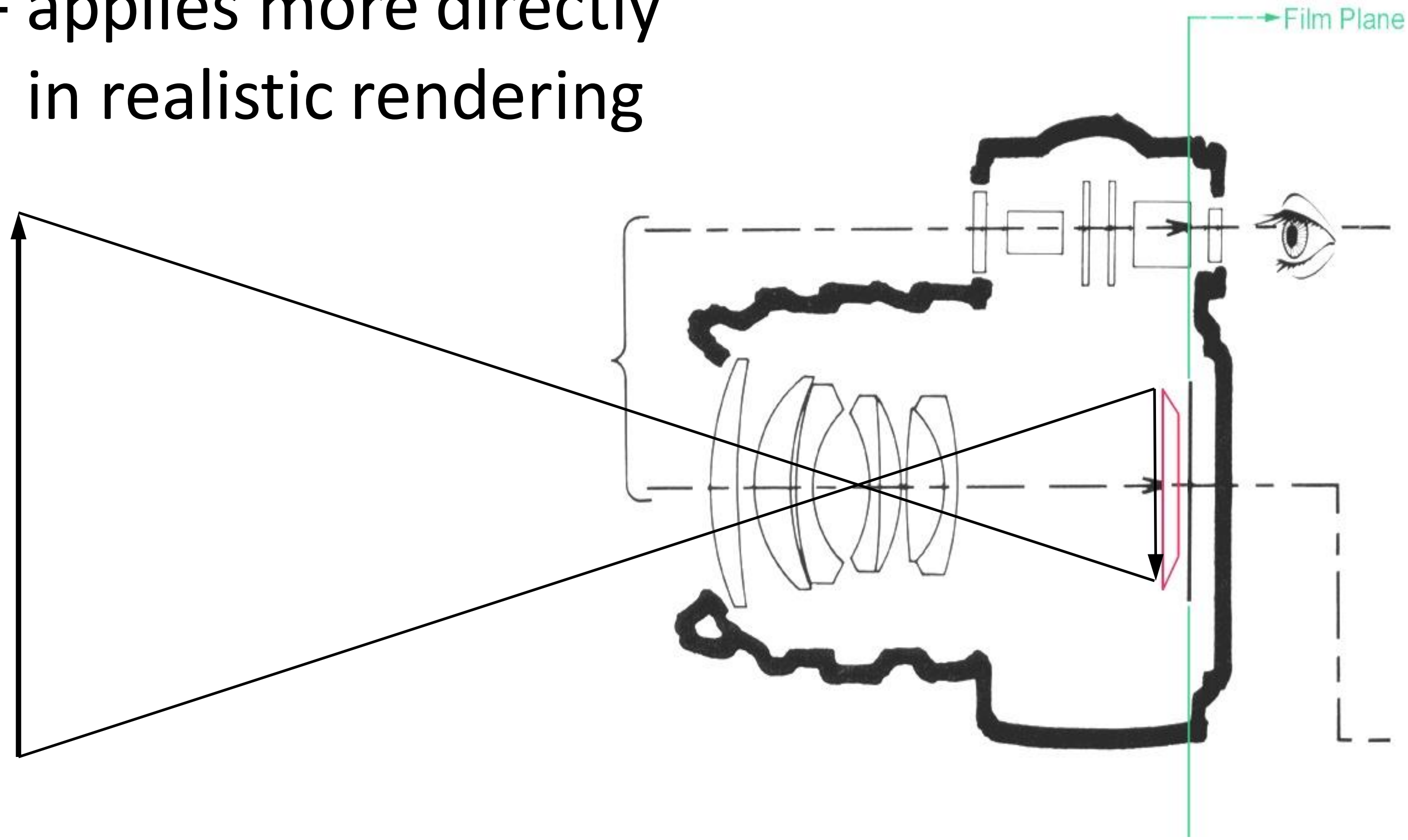


The concept of the picture plane may be better understood by looking through a window or other transparent plane from a fixed viewpoint. Your lines of sight, the multitude of straight lines leading from your eye to the subject, will all intersect this plane. Therefore, if you were to reach out with a grease pencil and draw the image of the subject on this plane you would be "tracing out" the infinite number of points of intersection of sight rays and plane. The result would be that you would have "transferred" a real three-dimensional object to a two-dimensional plane.

[Cornell CS 417 Spring 2002]

Plane projection in photography

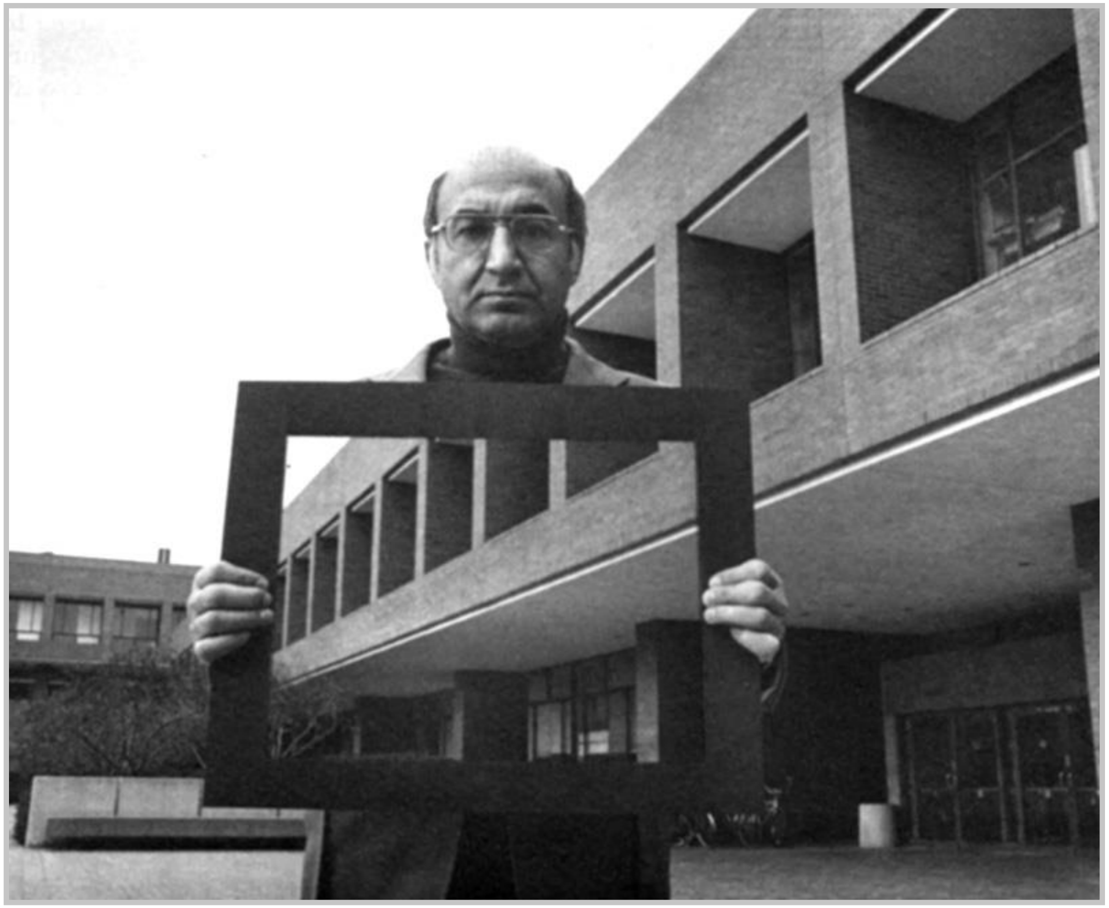
- This is another model for what we are doing
  - applies more directly in realistic rendering



[Source unknown]

Plane projection in photography

Questions: How was this picture made? What influences our perception of depth?

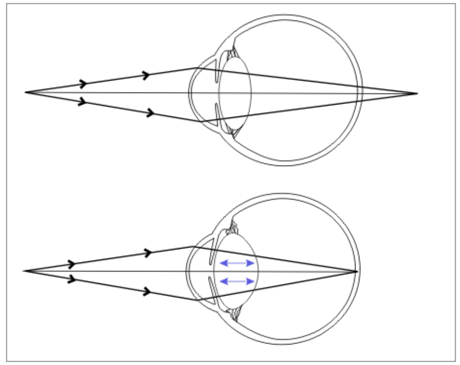


[Richard Zakia]

Tilt Shift Photography

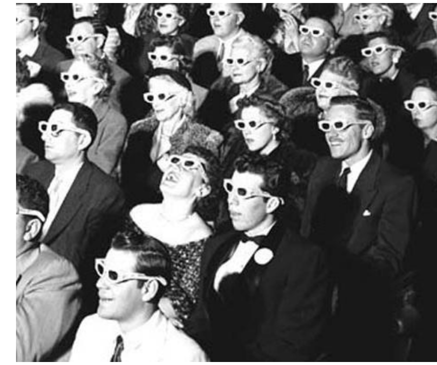
How many depth perception cues can you think of?

defocus



accommodation

vergence



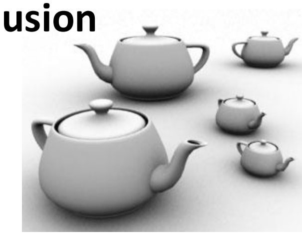
stereopsis

reflection

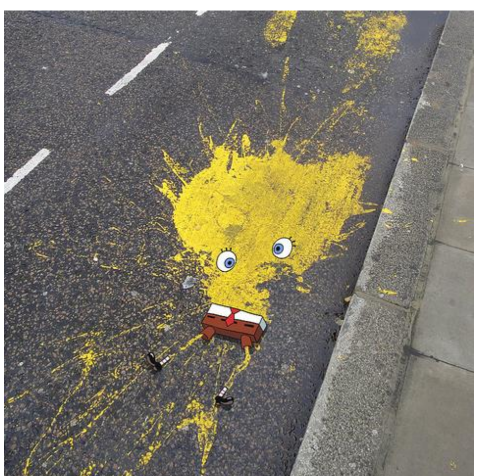
shadows

fog / atmospheric

occlusion

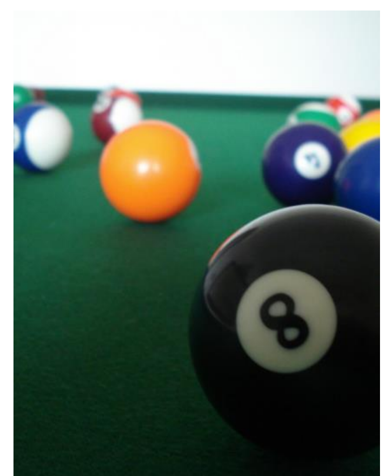


shading



texture gradient

perspective

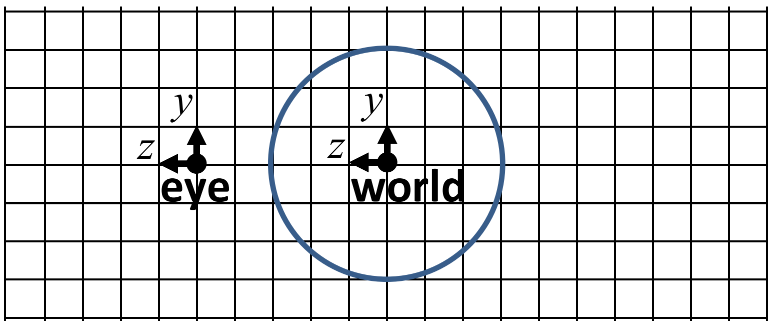


height in image

size familiarity

Review and more information

- OpenGL Red Book chapter 3 gives a good overview of how viewing and projective transformations are set in OpenGL
  - http://www.glprogramming.com/red/chapter03.html


y
z
eye
y
z
world