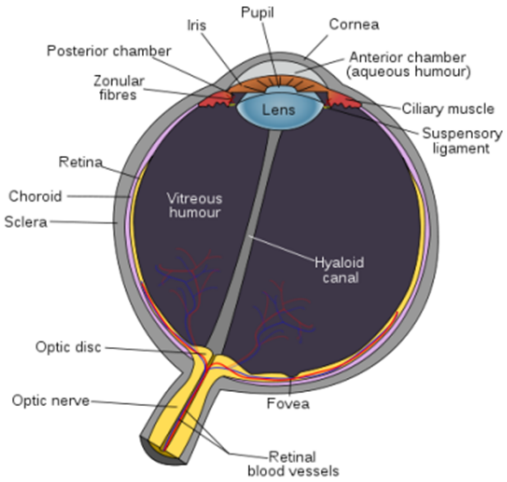
Camera Models

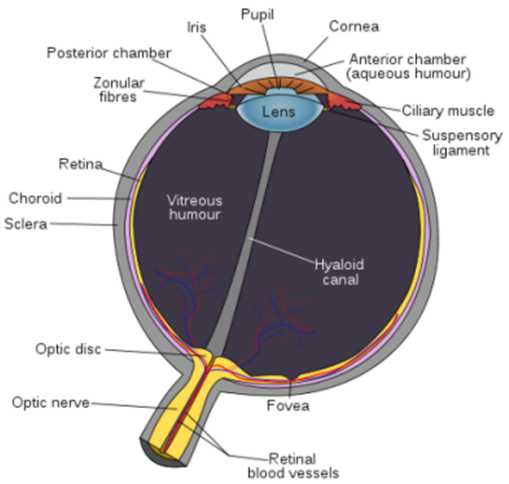
Fall 2023

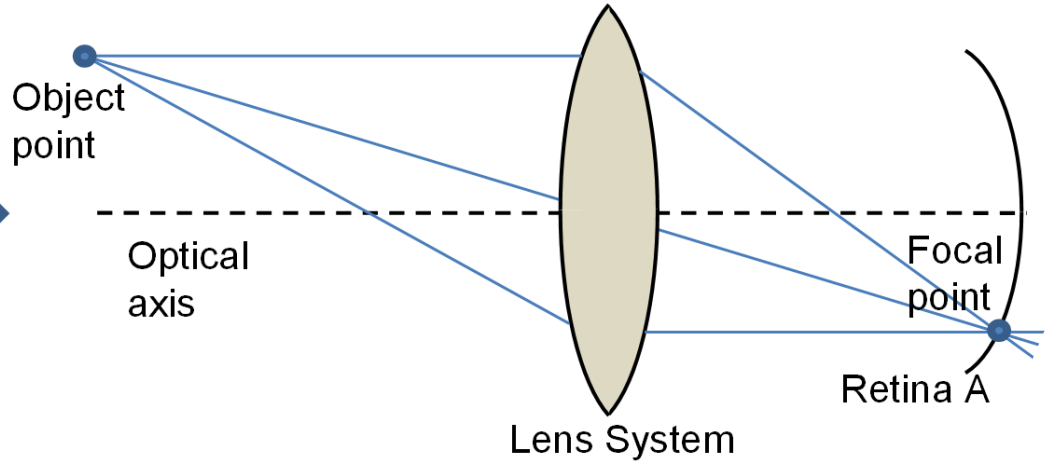
Daniel G. Aliaga
Department of Computer Science
Purdue University

Human Eye

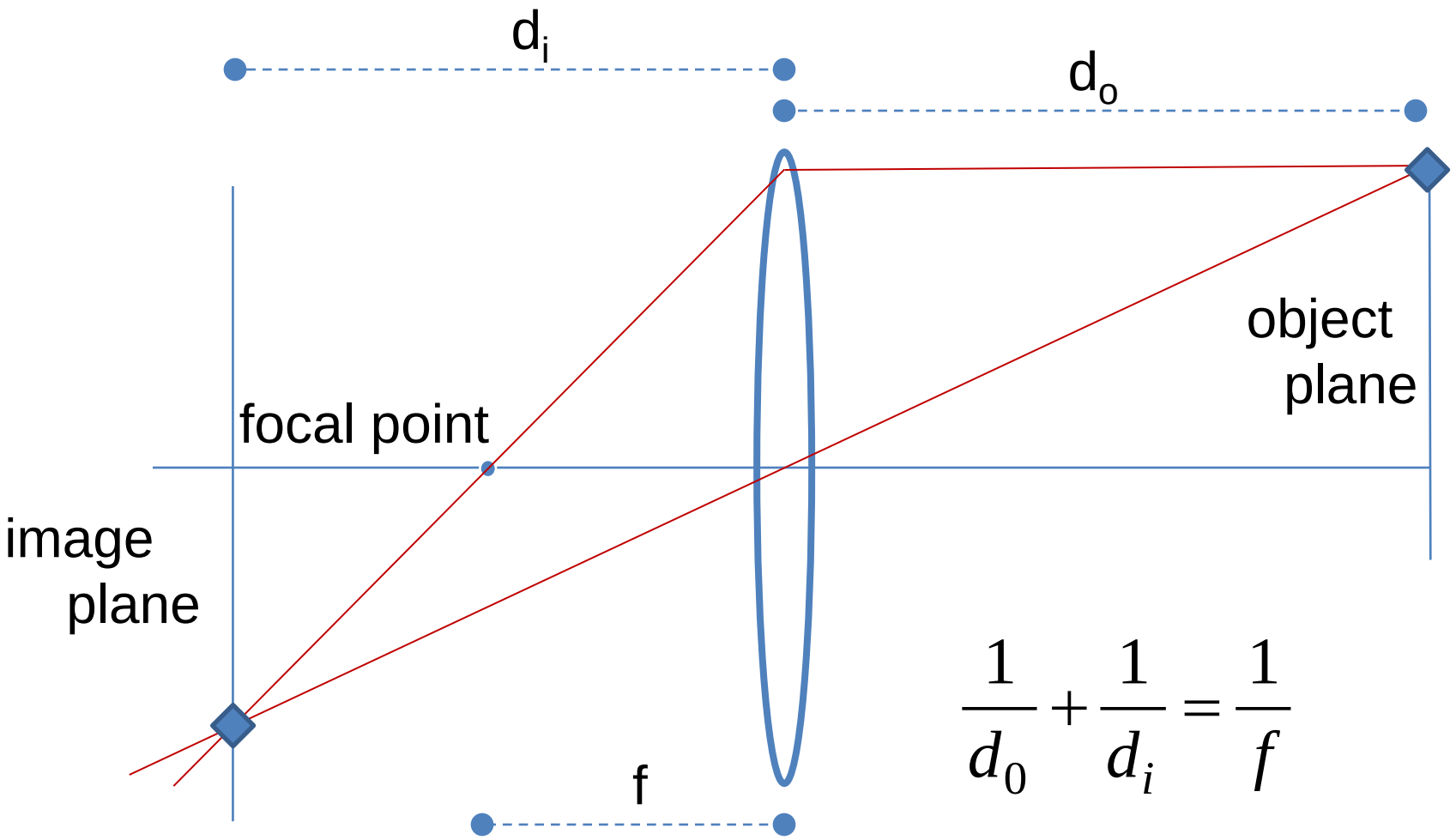

Pupil
Iris
Cornea
Posterior chamber
Anterior chamber
(aqueous humour)
Zonular
fibres
Lens
Ciliary muscle
Suspensory
ligament
Retina
Choroid
Vitreous
humour
Sclera
Hyaloid
canal
Optic disc
Optic nerve
Fovea
Retinal
blood vessels

Thin Lens System



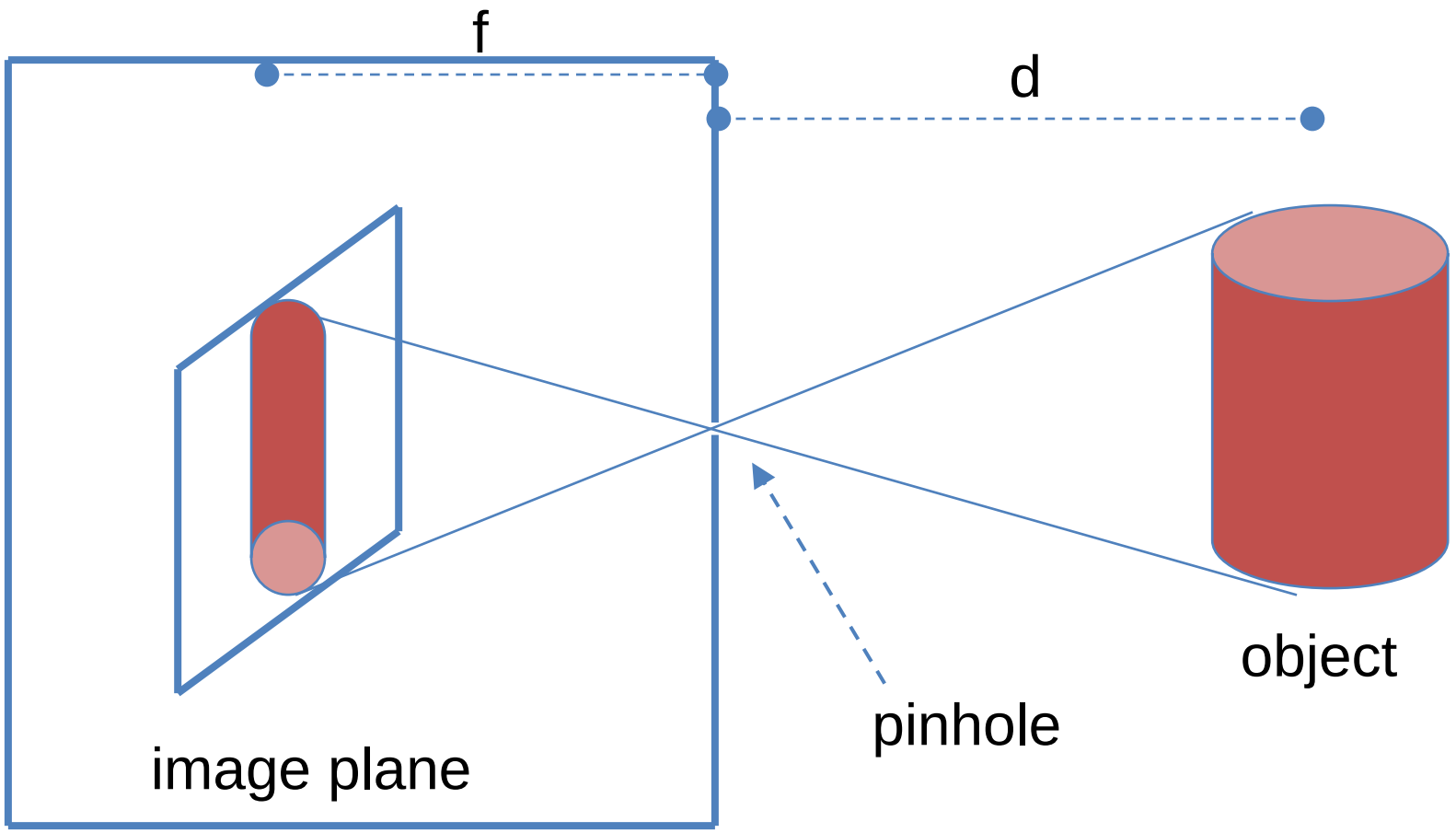


Thin Lens Equation

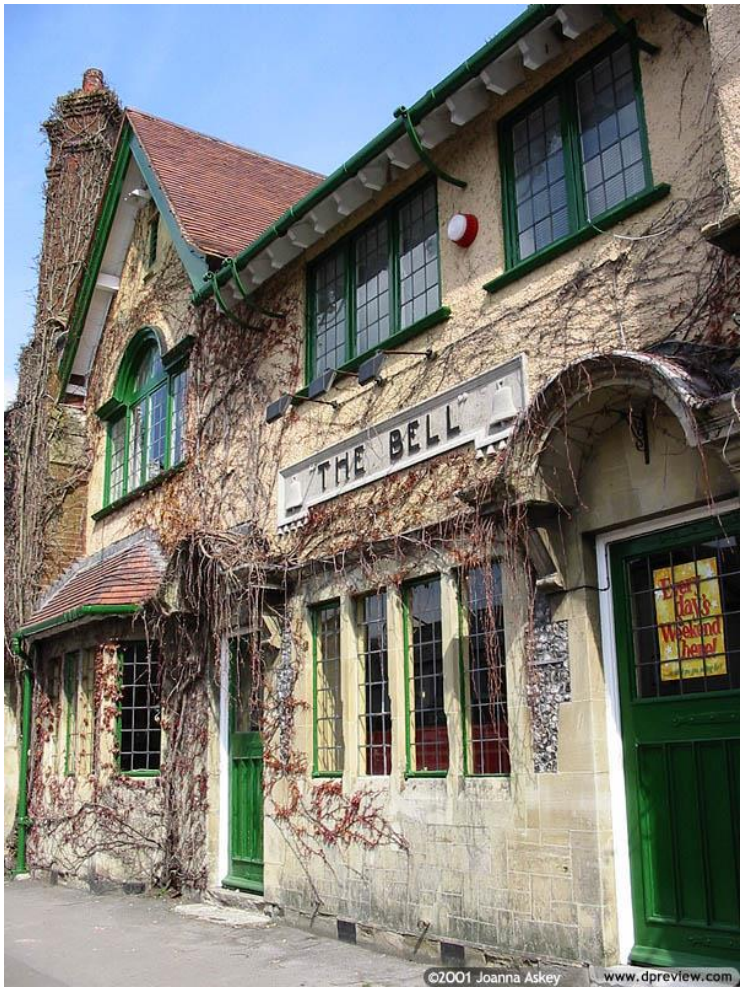


$$\frac{1}{d_0} + \frac{1}{d_i} = \frac{1}{f}$$

Pinhole Camera Model



“Pinhole” Camera Image

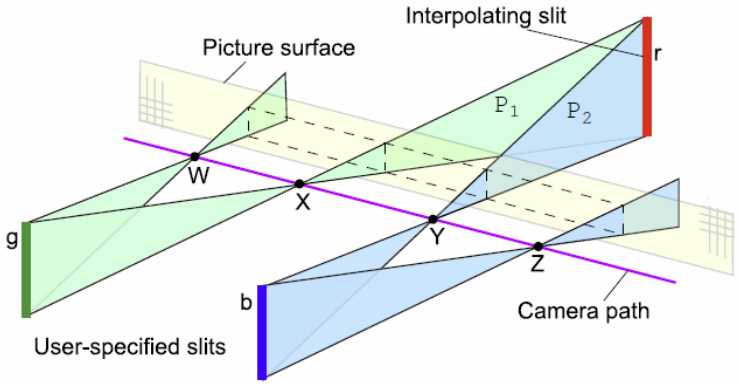


Digression: Non-Pinhole Camera Models…

- Why restrict the camera to a pinhole camera model?

Multiperspective Imaging

Hand-crafted



semi-automated…

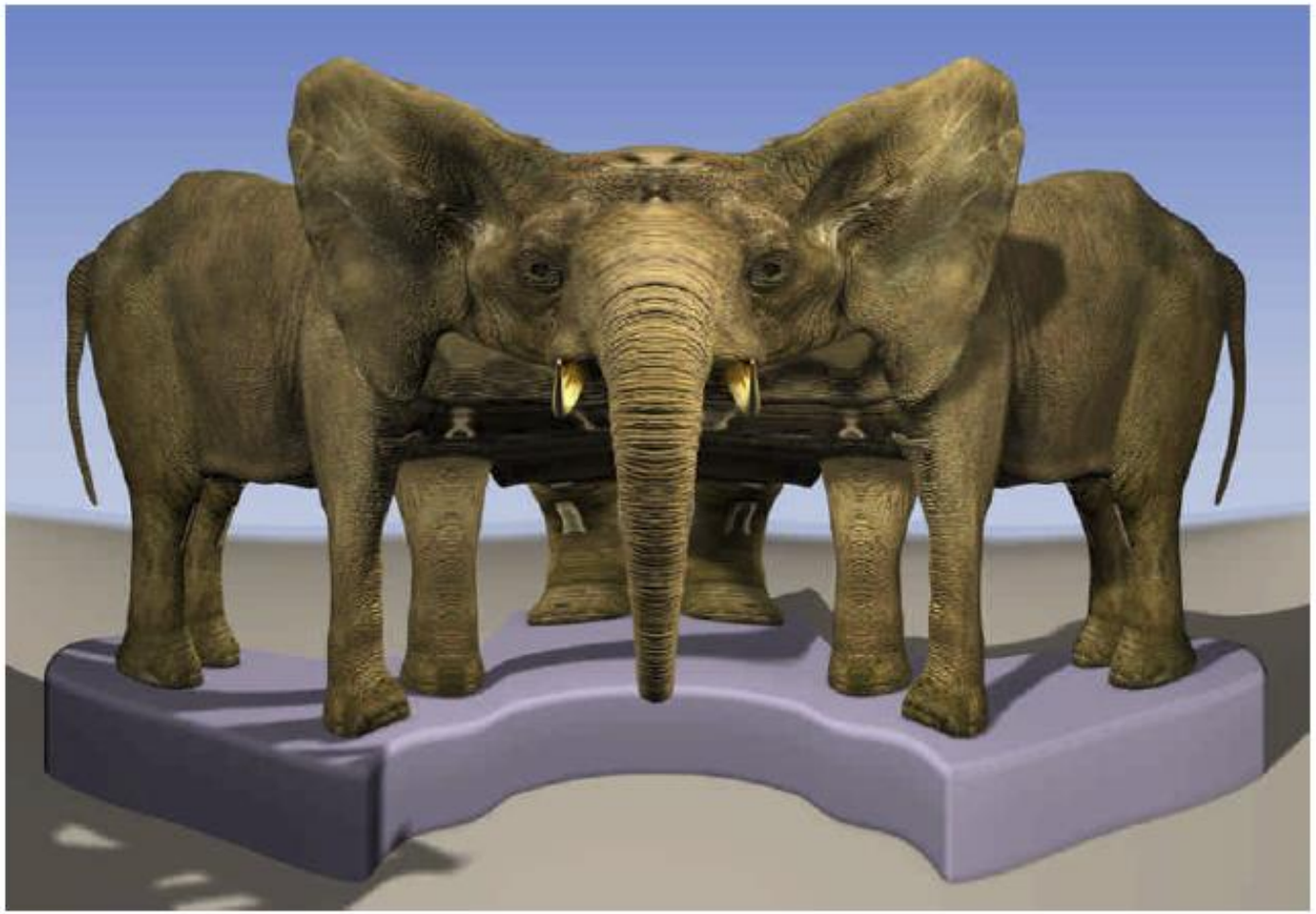
to produce this…

[Roman-Vis04]

Multiperspective Imaging

[Seitz-CGA03]

Multiple COP Images



[Rademacher-SIG98]

Multiple COP Images

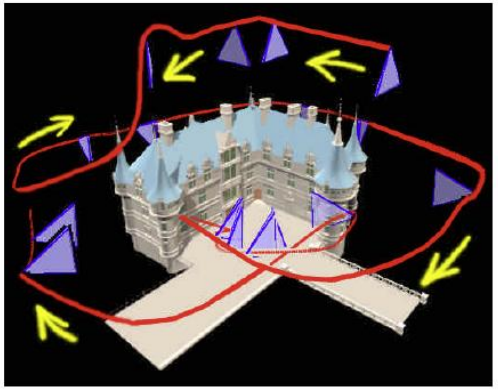


Figure 5 Castle model. The red curve is the path the camera was swept on, and the arrows indicate the direction of motion. The blue triangles are the thin frusta of each camera. Every 64th camera is shown.

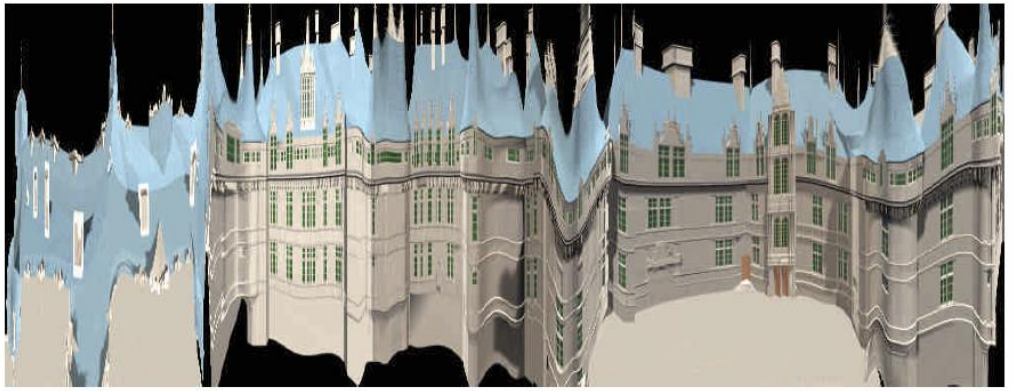


Figure 6 The resulting 1000×500 MCOP image. The first fourth of the image, on the left side, is from the camera sweeping over the roof. Note how the courtyard was sampled more finely, for added resolution.

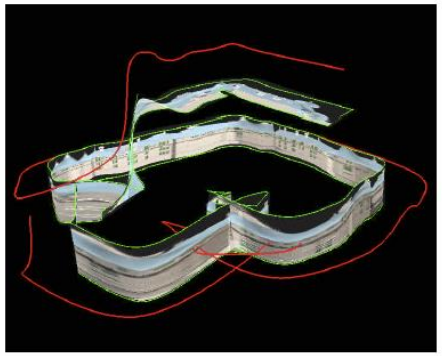


Figure 7 The projection surface (image plane) of the camera curve.

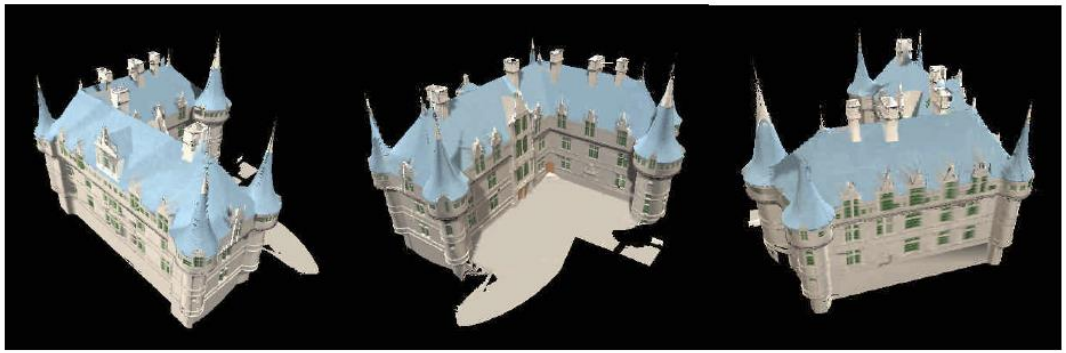


Figure 8 Three views of the castle, reconstructed solely from the single MCOP image above. This dataset captures the complete exterior of the castle.

[Rademacher-SIG98]

Multiperspective Imaging for Cel Animation

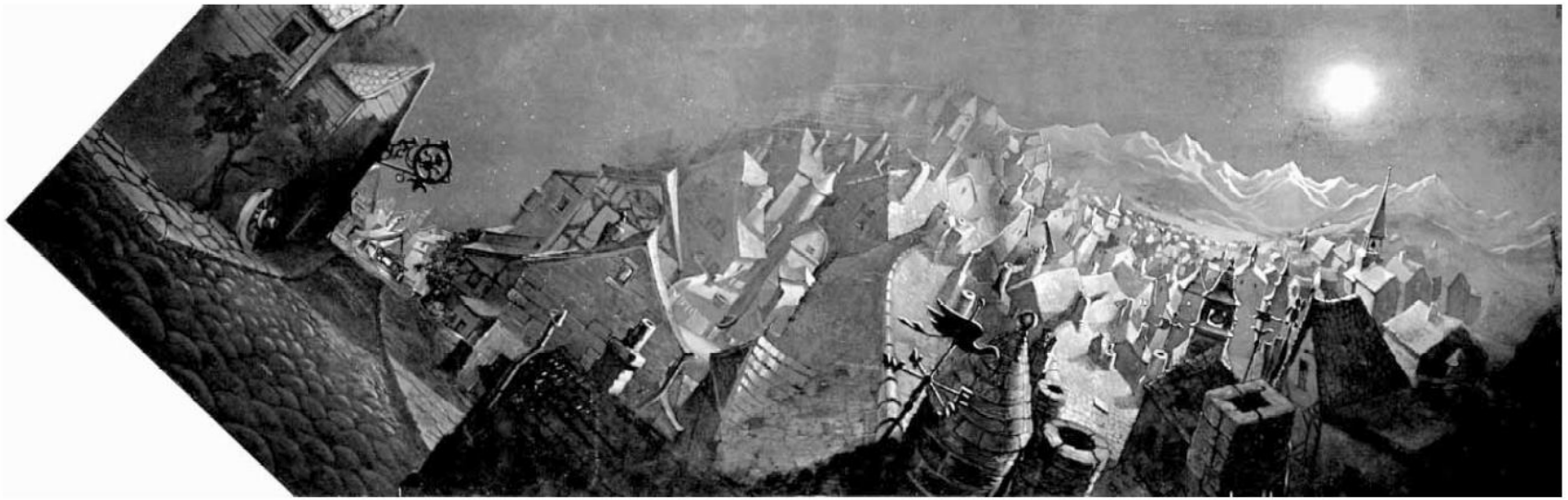
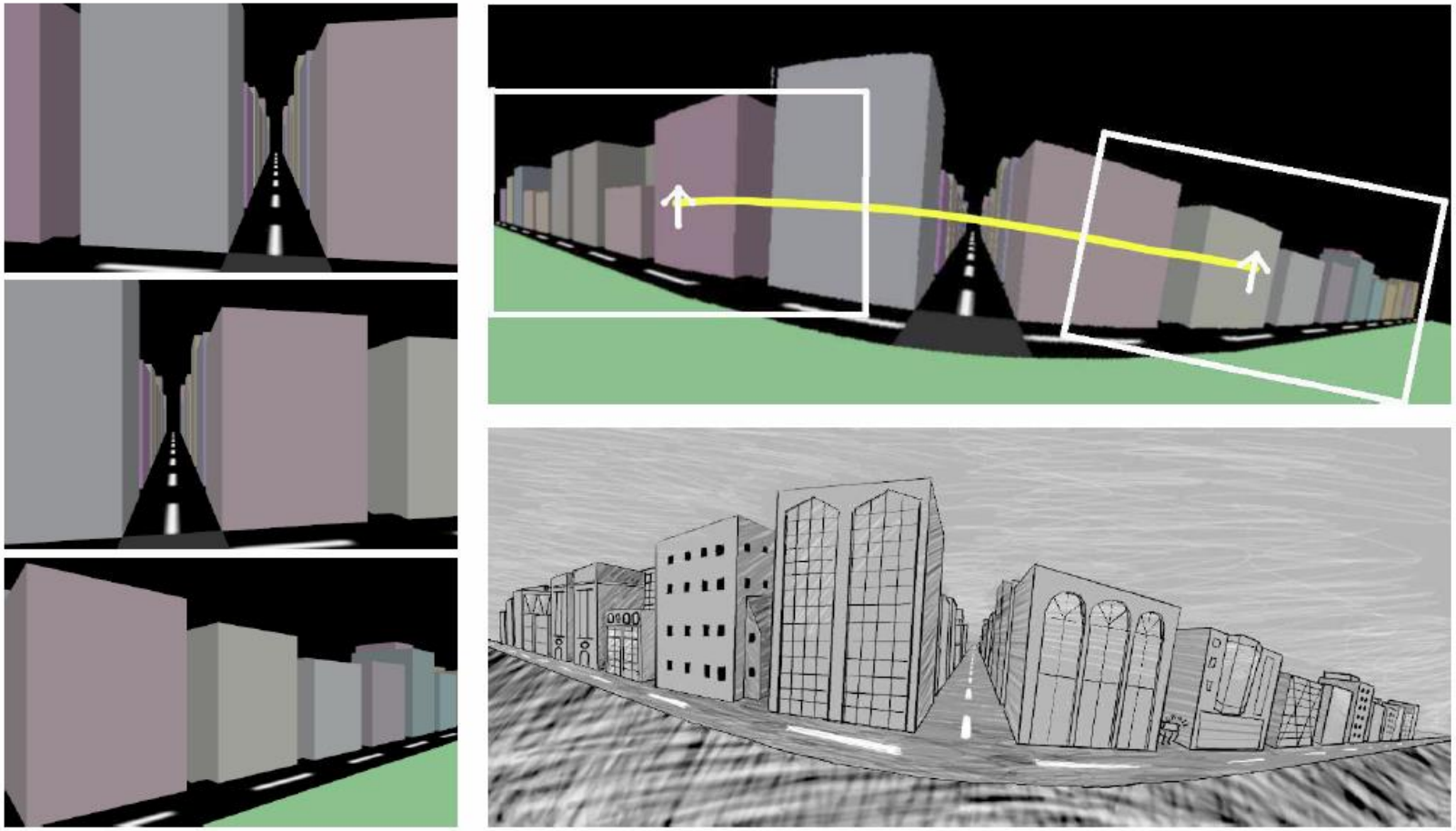


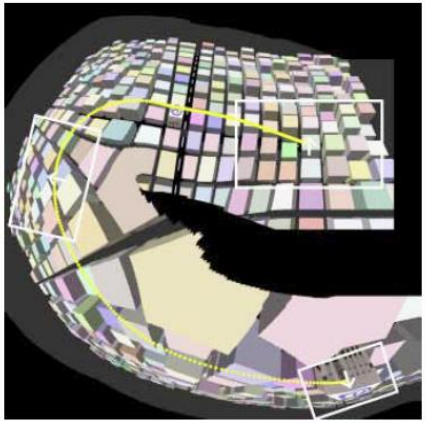
Figure 1 A multiperspective panorama from Disney's 1940 film *Pinocchio*. (Used with permission.)

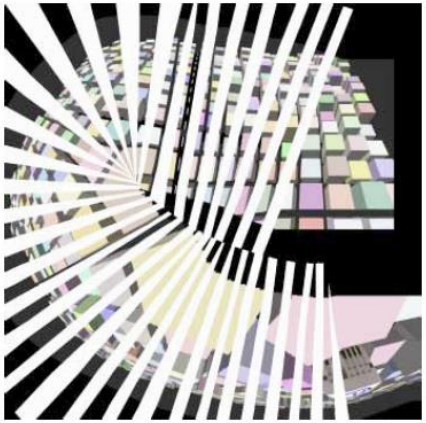
Multiperspective Imaging for Cel Animation

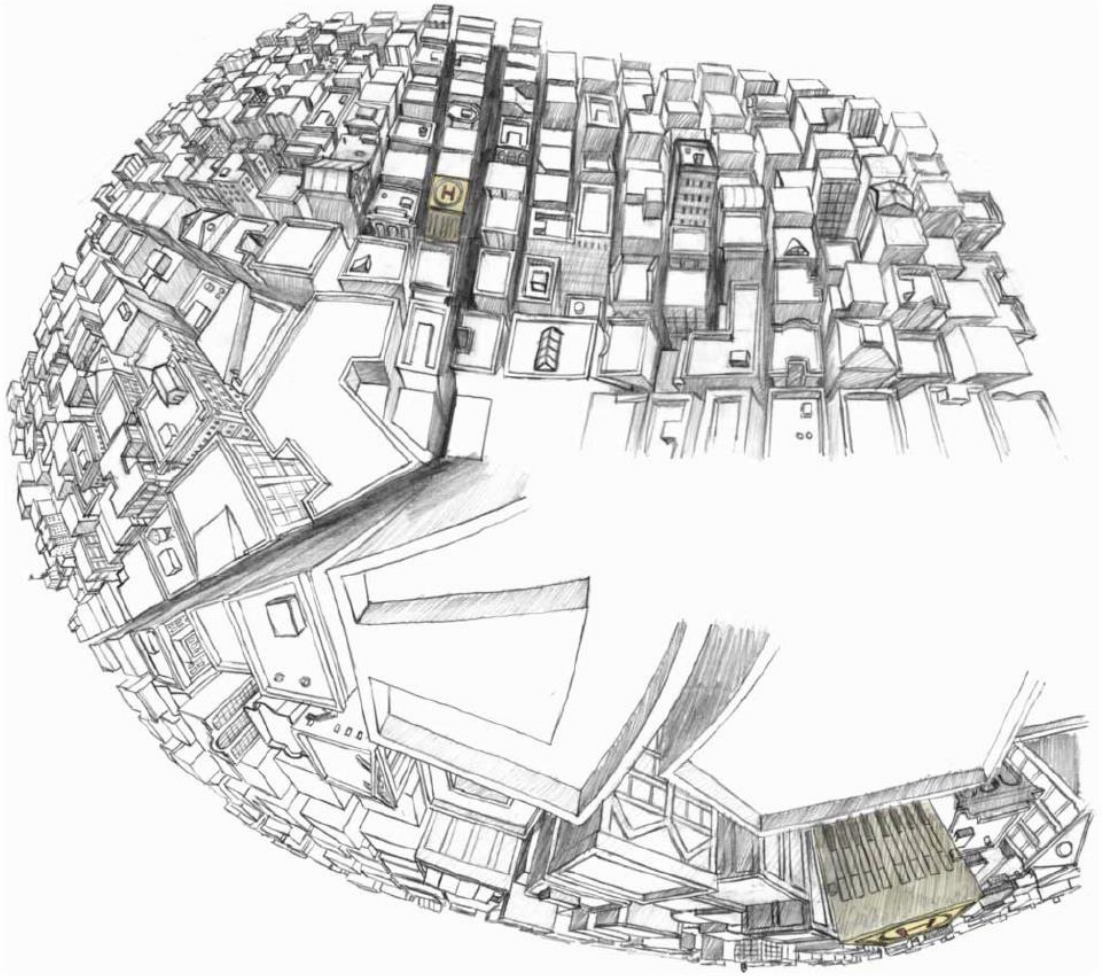


[Wood-SIG97]

Multiperspective Imaging for Cel Animation

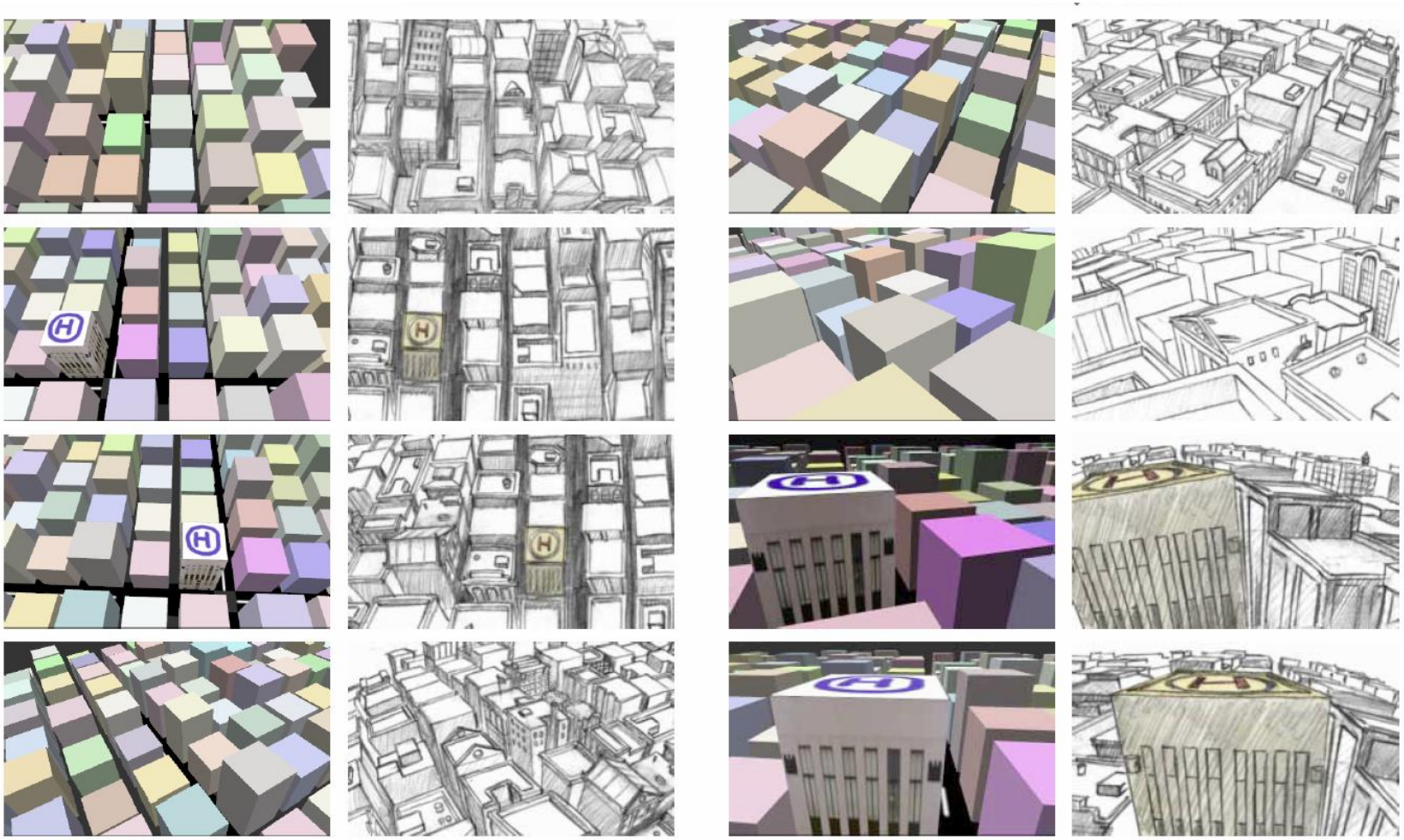






[Wood-SIG97]

Multiperspective Imaging for Cel Animation



[Wood-SIG97]

Occlusion-Resistant Cameras

Input images

Output images

[Aliaga-CGA07]

Occlusion Cameras

SIMULATOR IMAGES

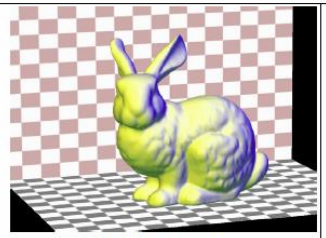
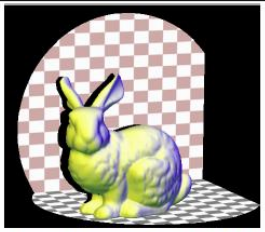


Figure 1 Depth image (DI).



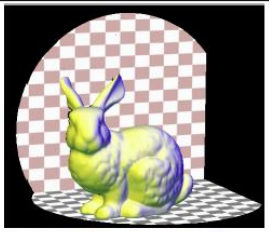


Figure 2 Images rendered from DI and OCRI, viewpoint 4" left of reference viewpoint.

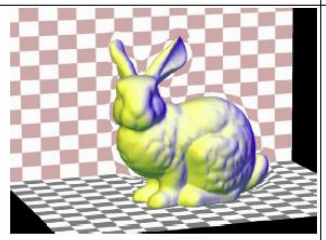
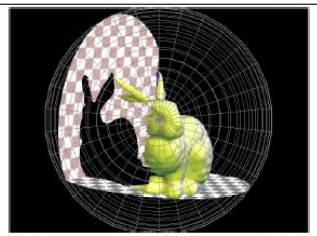


Figure 3 OCRI.



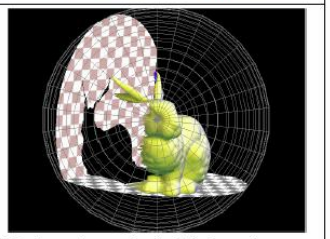
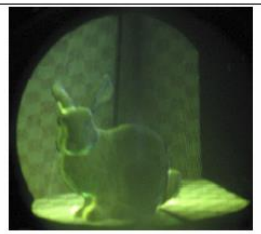
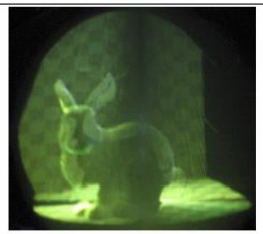


Figure 4 Images rendered from DI and OCRI. Wireframe shows spherical display volume.

PHOTOGRAPHS OF 3D DISPLAY





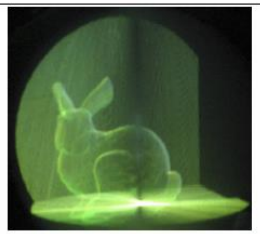
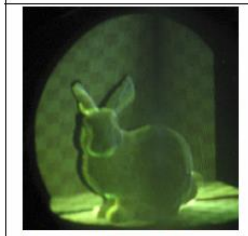


Figure 5 3D images rendered from DI (*left*), OCRI (*middle*), and original geometric model (*right*), all photographed from reference view.



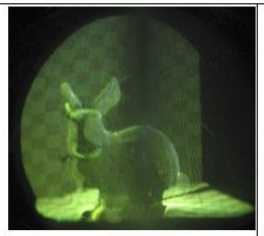
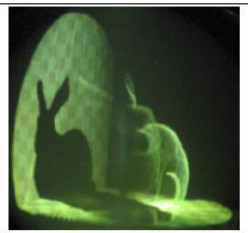


Figure 6 DI and OCRI 3D images from viewpoint translated 4" left.



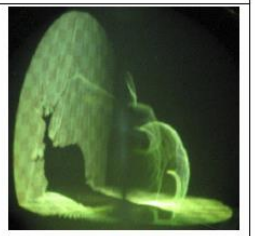
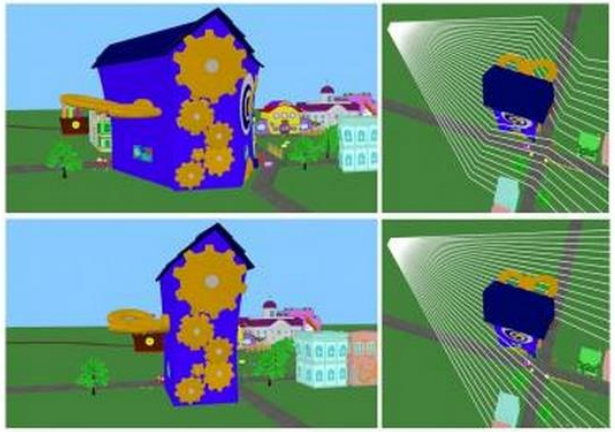


Figure 7 DI and OCRI 3D images from side view.

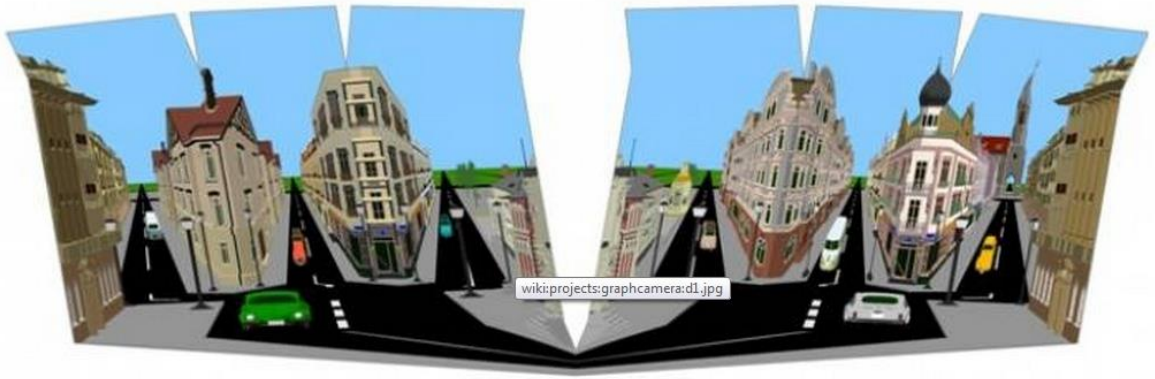
[Popescu-JDT06]

Graph Cameras

Portal-based graph camera image (top left and fragment right) and PPC image for comparison (bottom left)



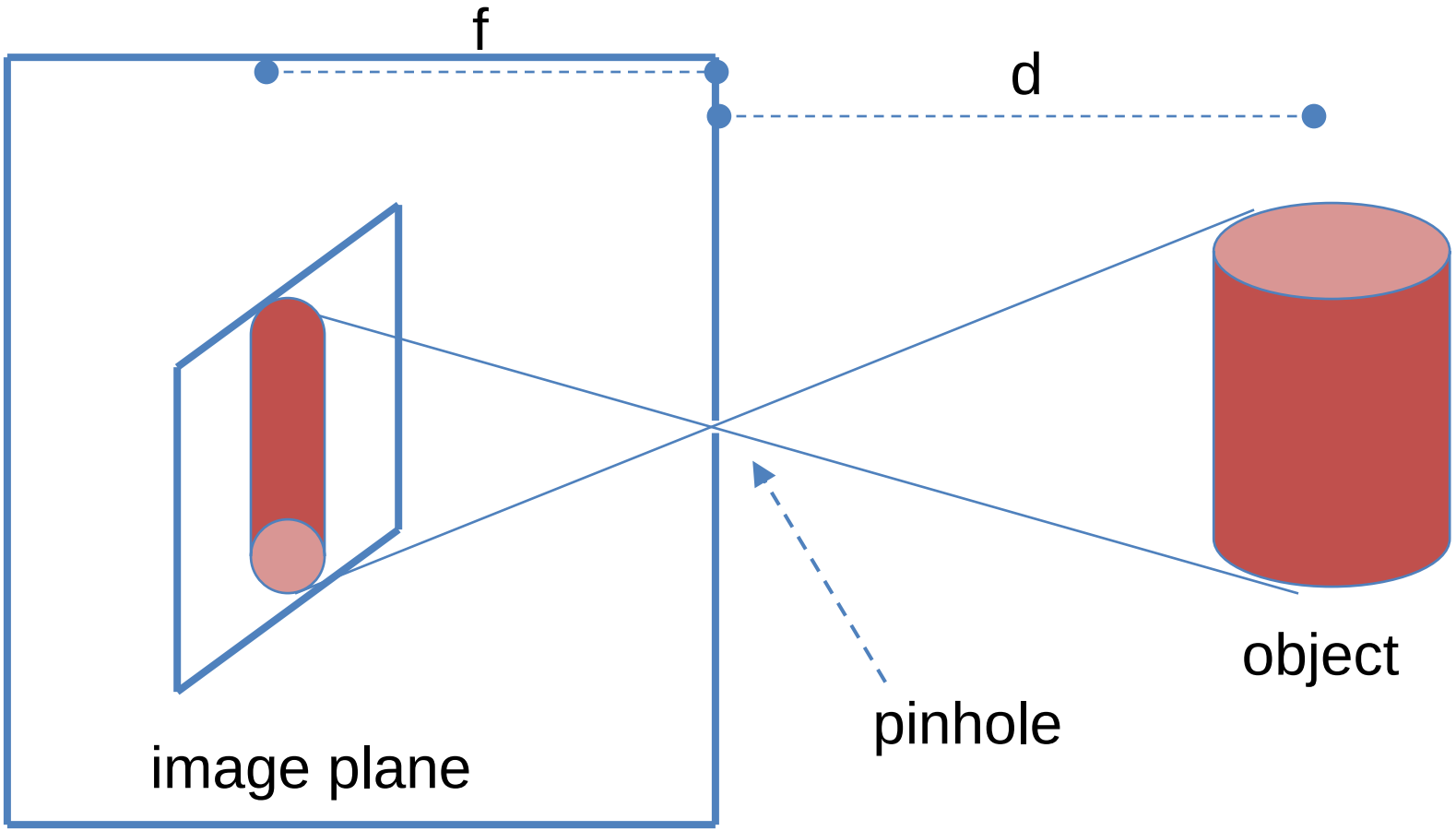
Occluder-based graph camera image (top left), PPC image for comparison (bottom left), and ray visualizations (right)



Enhanced street-level navigation

[Popescu-SIGA09]

Let's get back on track: Pinhole Camera Model…



Photographic Camera History 101

- First photos:
  - 1826:
    - https://www.youtube.com/watch?v=sOkd8ObhN_M
- First videos:
  - 1874
    - https://www.youtube.com/watch?v=VC-yTYyE2w0
- Femto camera (10^{12} frames per second)
  - 2013
    - https://www.youtube.com/watch?v=EtsXgODHMWk

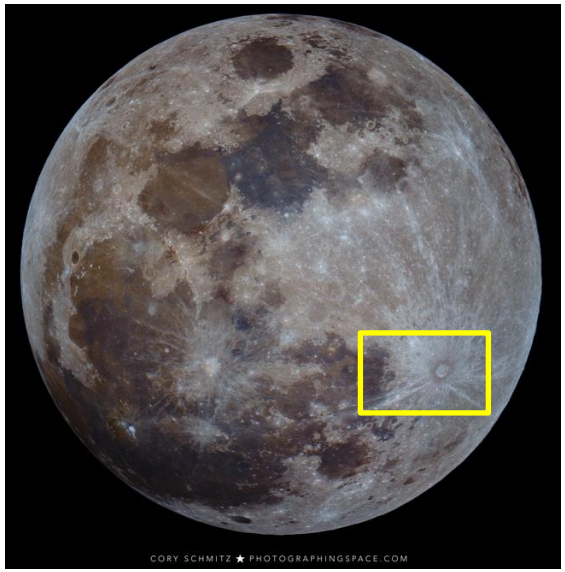
Camera Resolution

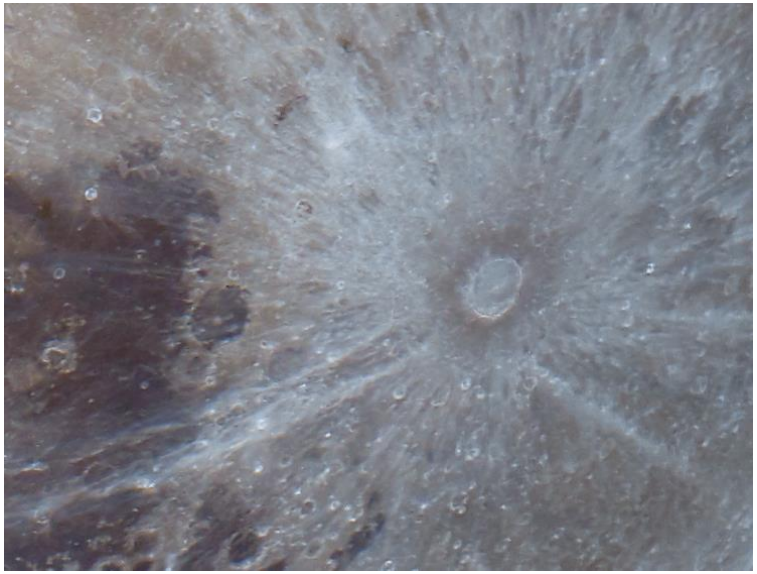
- First digital camera (~1985): 0.01 MP (100x100)
- Consumer ware: up to 100 MP
- Gigapixel camera (assembled pics): ~300 GP

Camera Resolution

0.25 MP

Camera Resolution





100 MP

(astrophotography)

Camera Resolution

100 MP

Camera Resolution

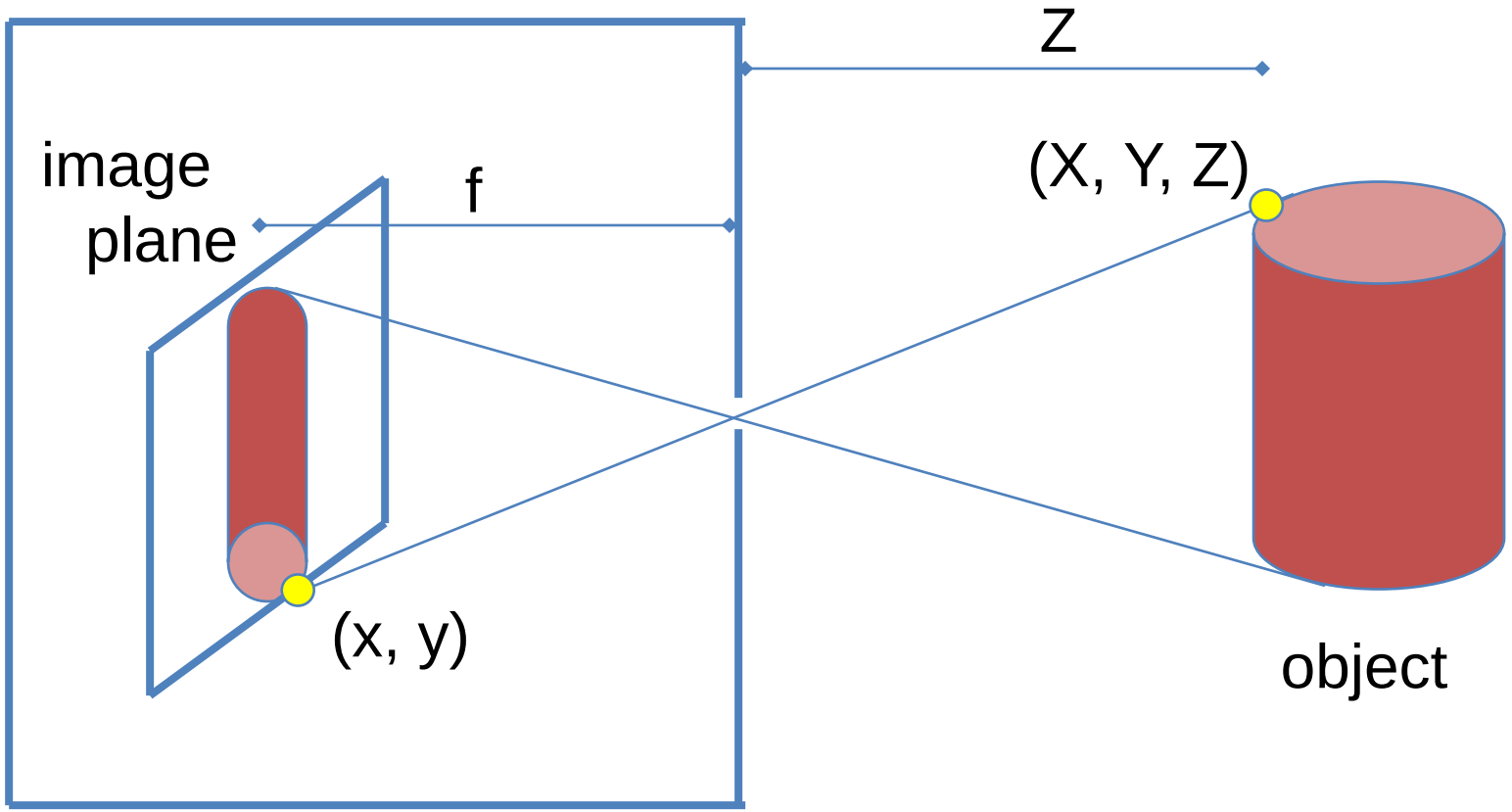
272 GP

Used 12,000 images….Calibration is crucial!!!!

Camera Resolution

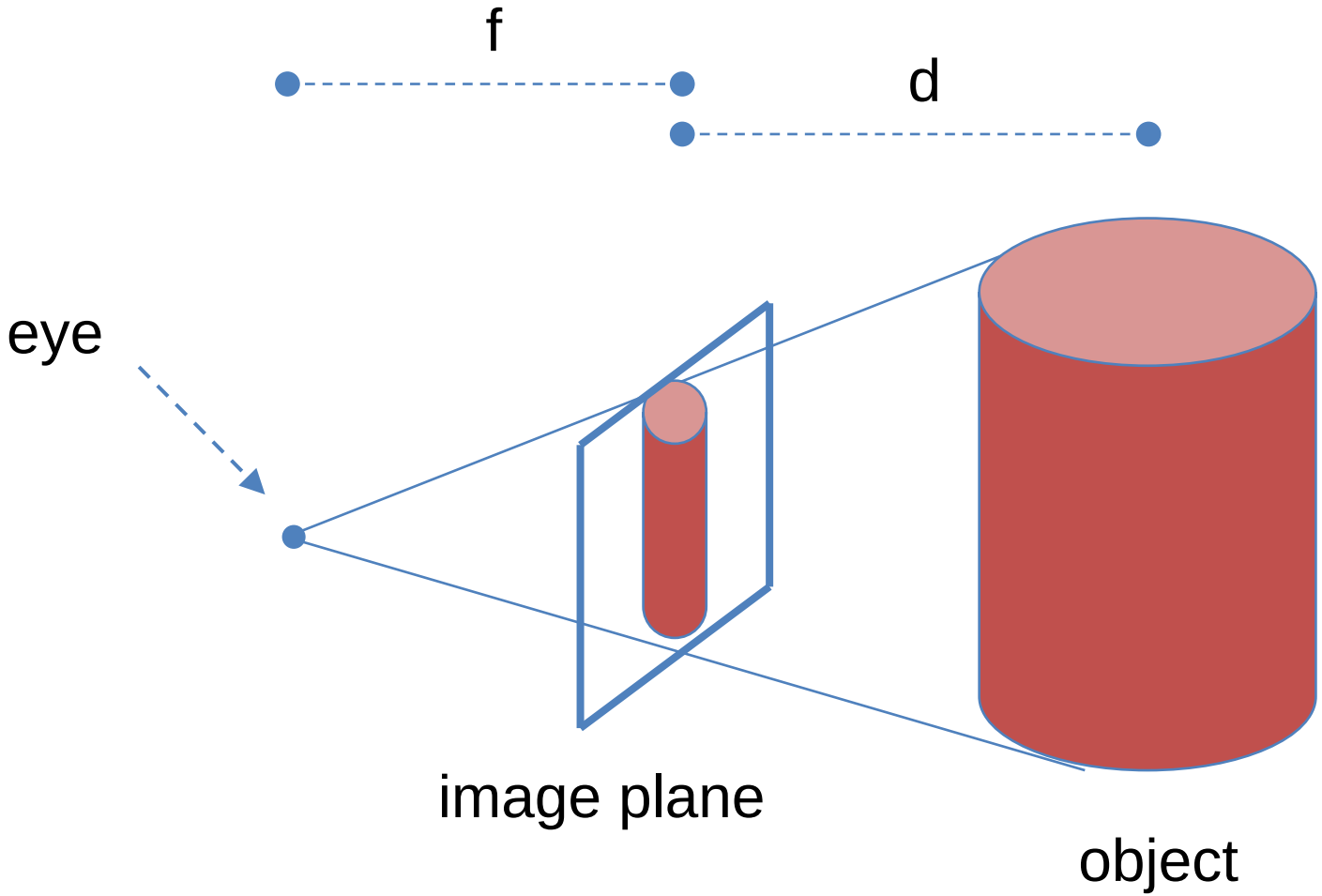
- London (565,000 x 565,000 pixels)
  - https://www.dpreview.com/articles/7683341128/320-gigapixel-photo-of-london-is-the-worlds-largest-panoramic-photo

What is perspective projection?

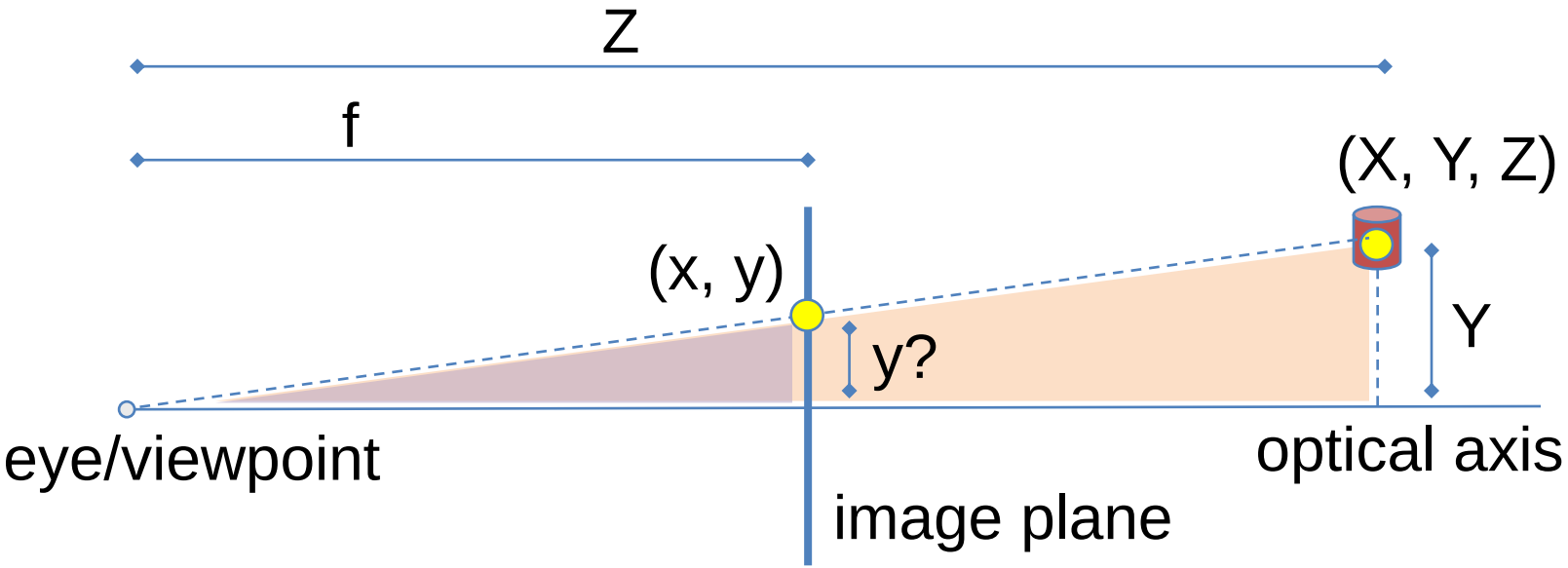


$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

“Computer Graphics” Pinhole Camera Model



What is perspective projection?



$$\frac{y}{f} = \frac{Y}{Z} \quad \Rightarrow \quad y = f\frac{Y}{Z} \quad \& \quad x = f\frac{X}{Z}$$

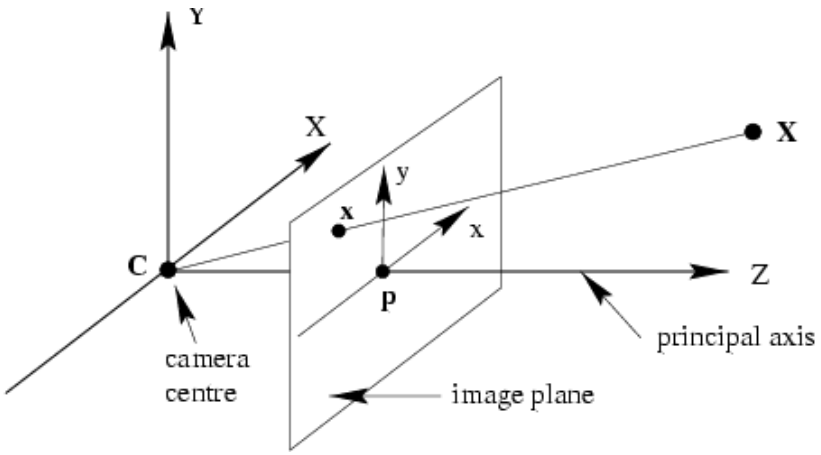
Perspective Camera Parameters

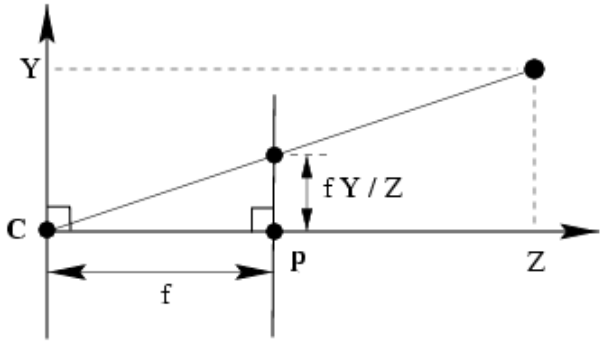
- Intrinsic/Internal
  - Focal length f
  - Principal point (center) p_x, p_y
  - Pixel size s_x, s_y
  - (Distortion coefficients) $k_1, \ldots$
- Extrinsic/External
  - Rotation ϕ, φ, ψ
  - Translation t_x, t_y, t_z

Perspective Camera Parameters

- Intrinsic/Internal
  - Focal length f
  - Principal point (center) (=middle of image)
  - Pixel size (=1, irrelevant)
  - (Distortion coefficients) (=0, assuming no bugs ☺)
- Extrinsic/External
  - Rotation ϕ, φ, ψ
  - Translation t_x, t_y, t_z

Focal Length

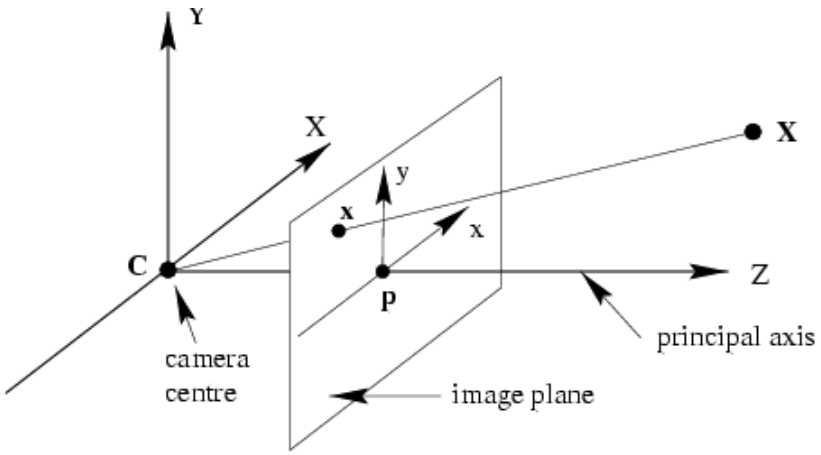


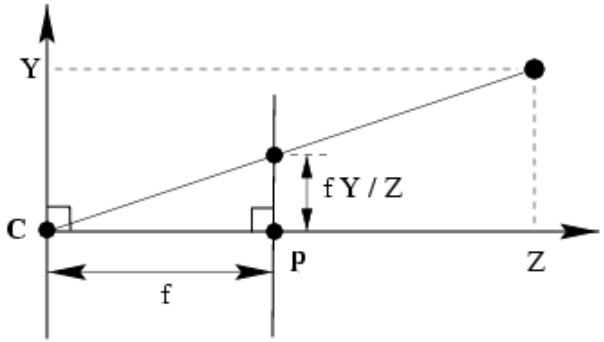


Assume $c = 0$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix} \Leftarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Focal Length

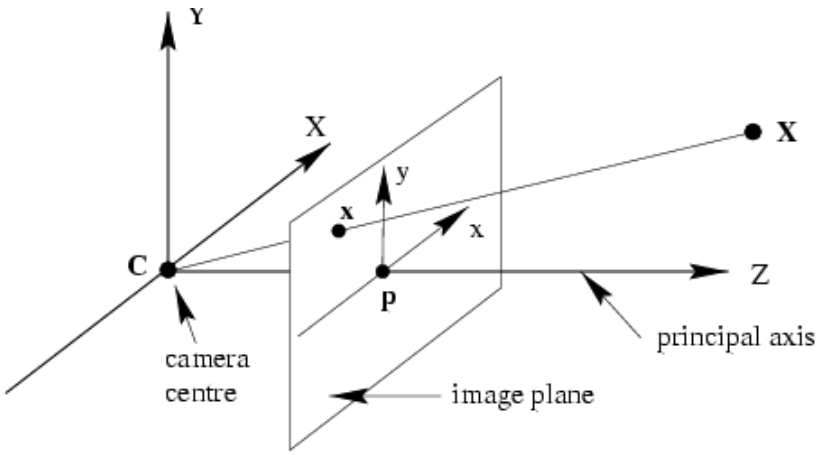


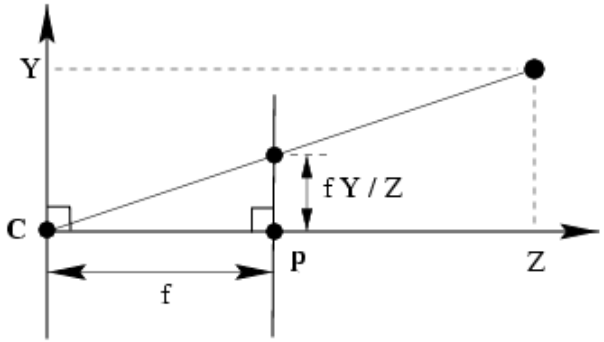


Assume $c \neq 0$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} fX'/Z' \\ fY'/Z' \end{pmatrix} \Leftarrow \begin{pmatrix} fX' \\ fY' \\ Z' \end{pmatrix} = \begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Focal Length

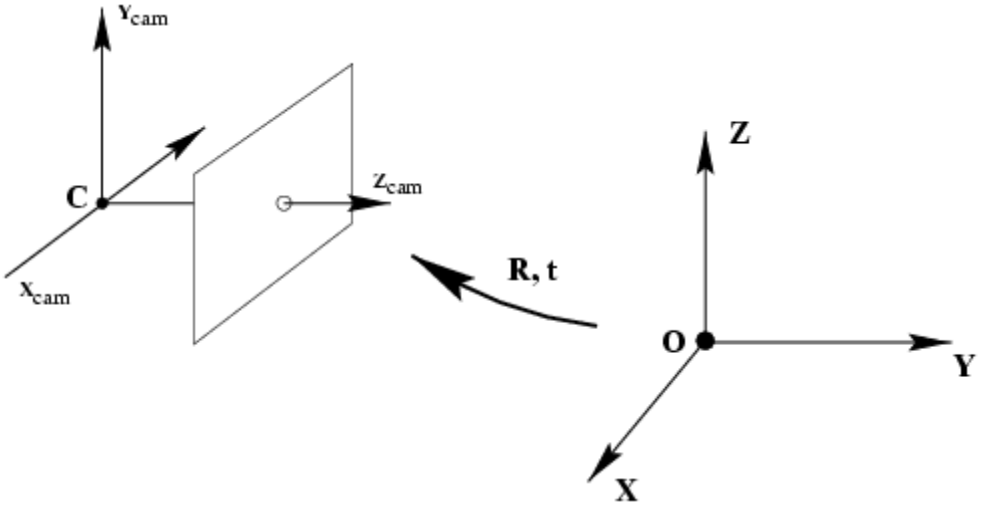




In general, where M is some 4x4 matrix:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} fX'/Z' \\ fY'/Z' \end{pmatrix} \Leftarrow \begin{pmatrix} fX' \\ fY' \\ Z' \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} M_{4x4} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

World to Camera Matrix *M*



$$\tilde{x}_c = R(\tilde{X} - C)$$
$$\tilde{x}_c = R\tilde{X} - RC$$
$$-t$$

$$\tilde{x}_c = \begin{bmatrix} \mathrm{R} & \mathrm{t} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

M

$$R = R_x R_y R_z$$

3x3 rotation matrices

$$t = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$$

translation vector

Perspective Projection Process

- Thus, given $\tilde{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$

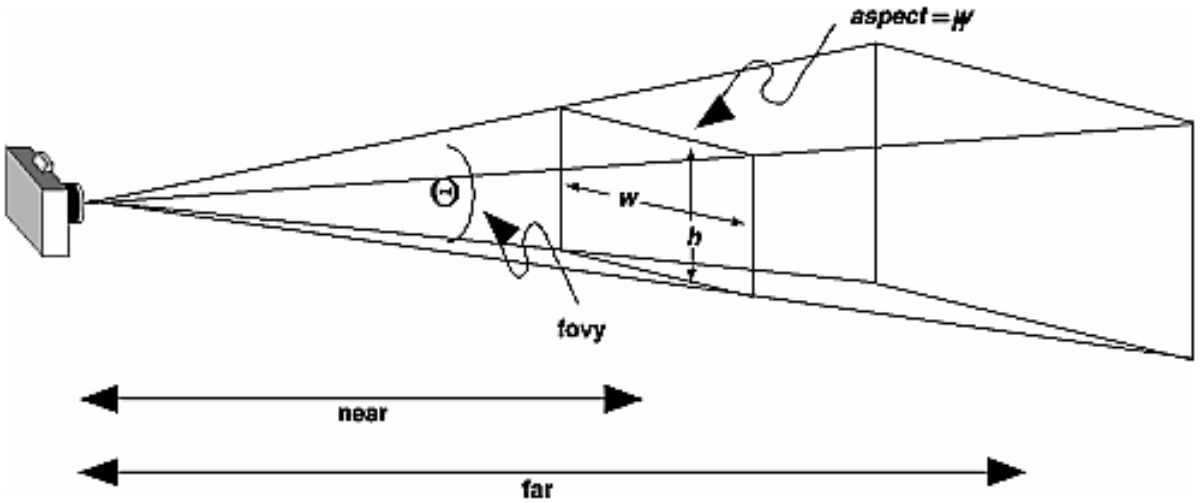
…the perspective projection is

$$\tilde{x}_p = PM \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \tilde{x}_{p_x} / \tilde{x}_{p_z} \\ \tilde{x}_{p_y} / \tilde{x}_{p_z} \end{bmatrix}$$

OpenGL Equivalent

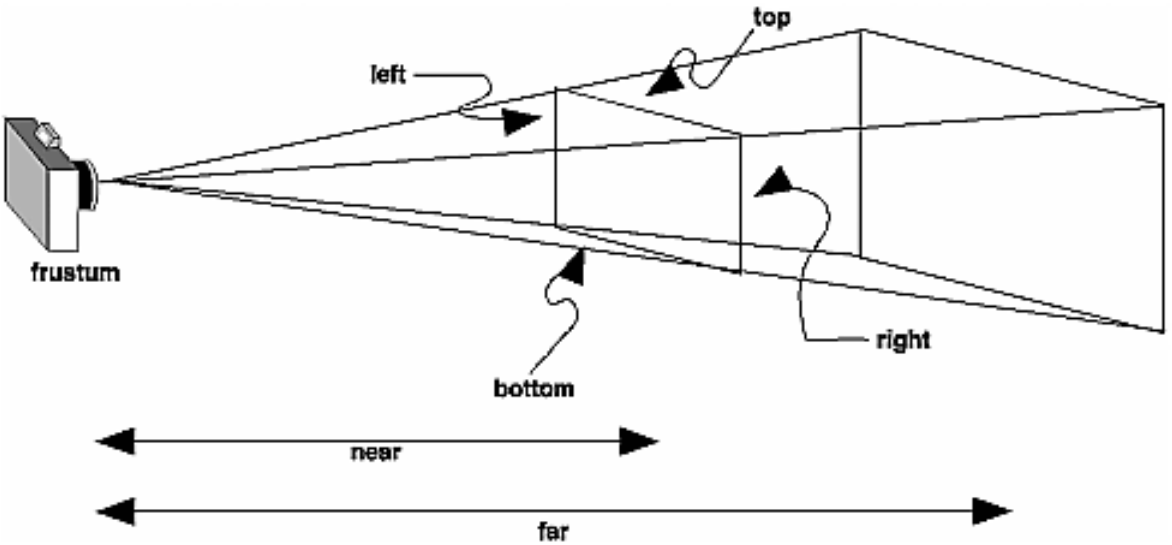

```
…
glMatrixMode(GL_PROJECTION);
…
gluPerspective(60, 1.0, 0.1, 1000.0);
…
glMatrixMode(GL_MODELVIEW);
…
glTranslatef(tx,ty,tz);
glRotatef(rx,1,0,0);
glRotatef(ry,0,1,0);
glRotatef(rz,0,0,1);

/* or glLoadMatrixf(mat); */
…
```

Projection Transformations



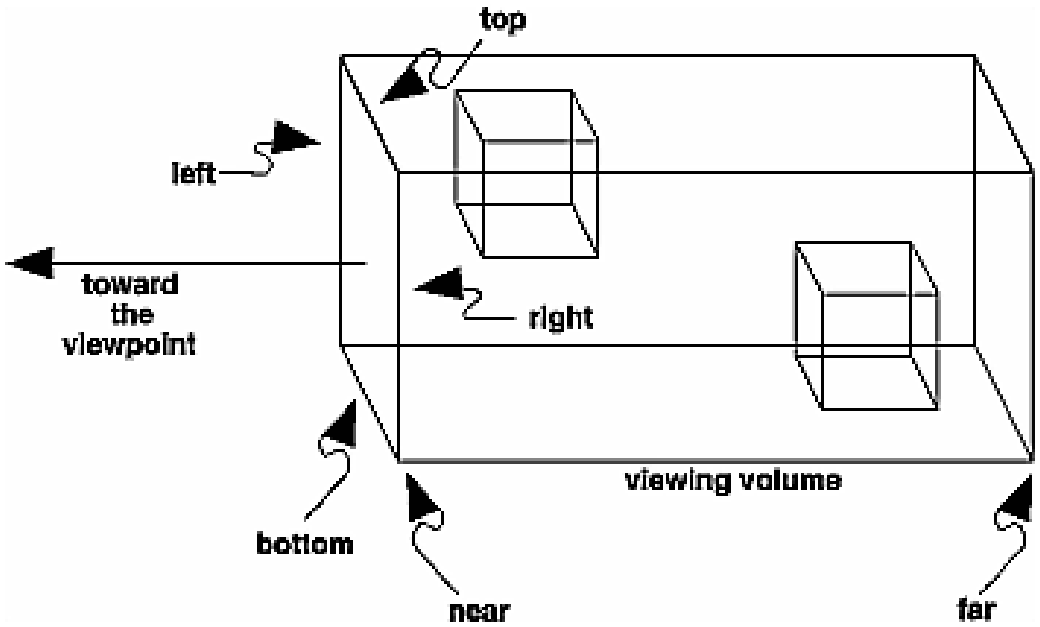

```
void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble
   near, GLdouble far);
```

Projection Transformations



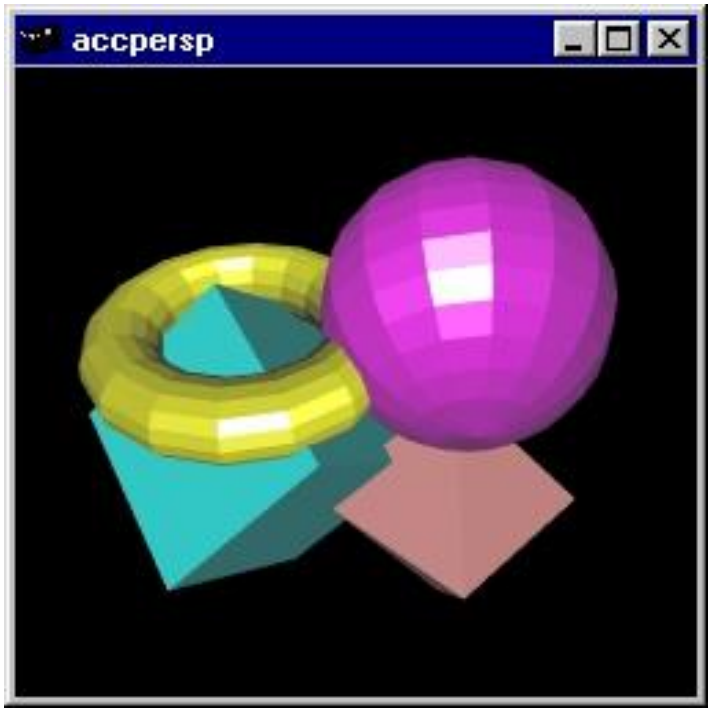

```
void glFrustum(GLdouble left, GLdouble right, GLdouble
  bottom, GLdouble top, GLdouble near, GLdouble far);
```

Projection Transformations




```
void glOrtho(GLdouble left, GLdouble right, GLdouble
  bottom,
  GLdouble top, GLdouble near, GLdouble far);

void gluOrtho2D(GLdouble left, GLdouble right,
  GLdouble bottom, GLdouble top);
```

Example OpenGL Rendering



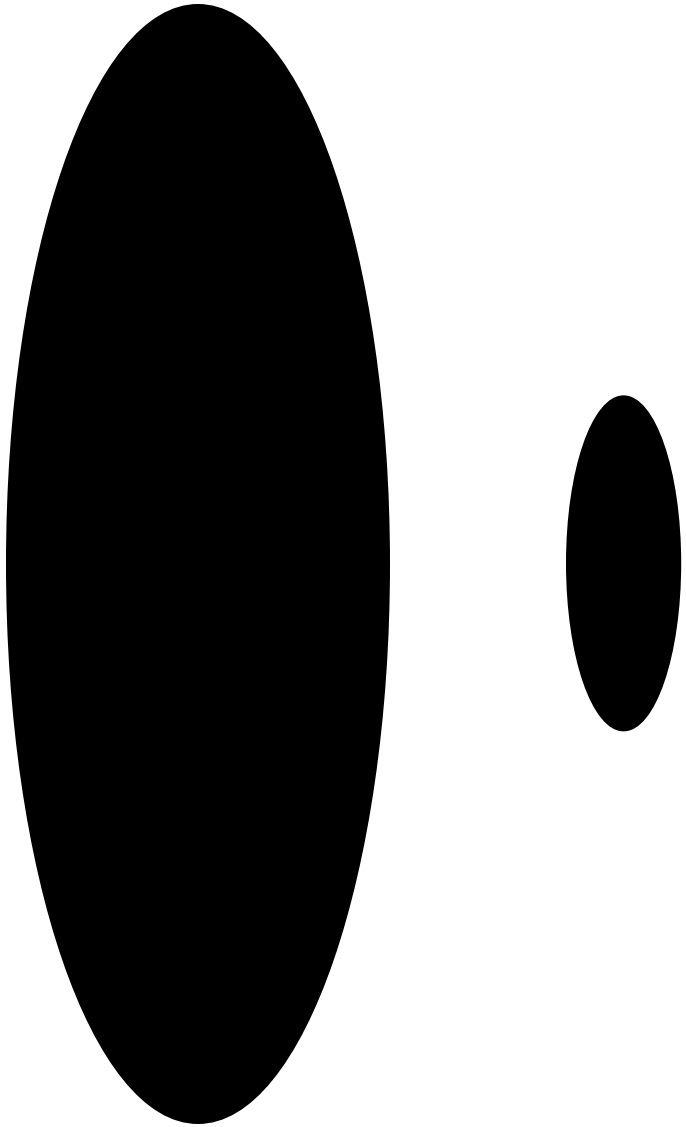
- Perspective is “mathematically” accurate…

(3D) Perception

- There is lot more to it…

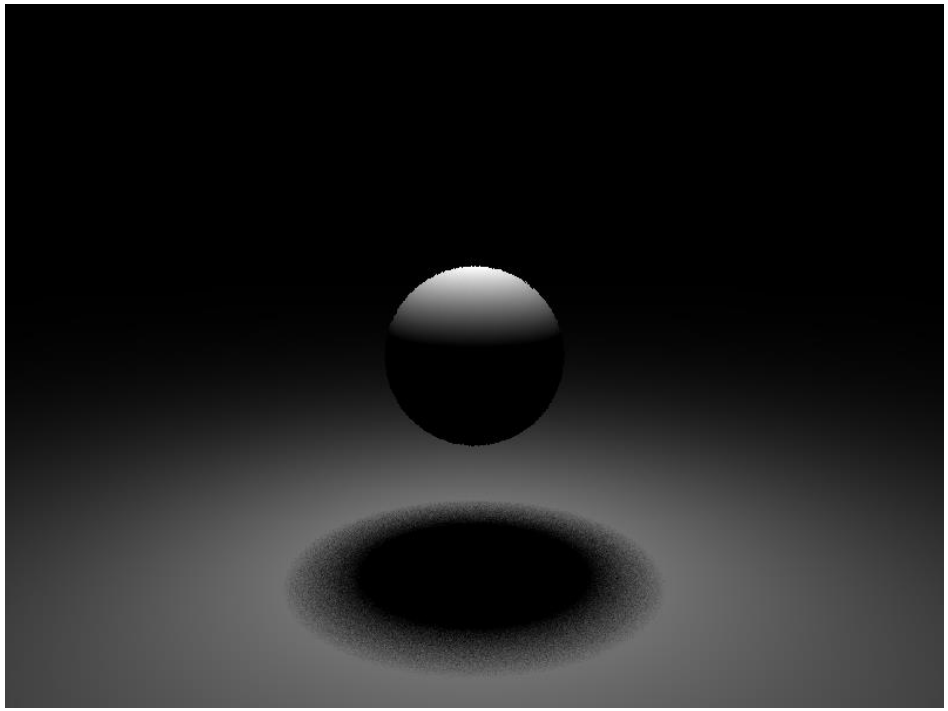
Perception

- Size-distance relationship
  - Same size object at two distances…



Perception

- Importance of ground plane, shadows…



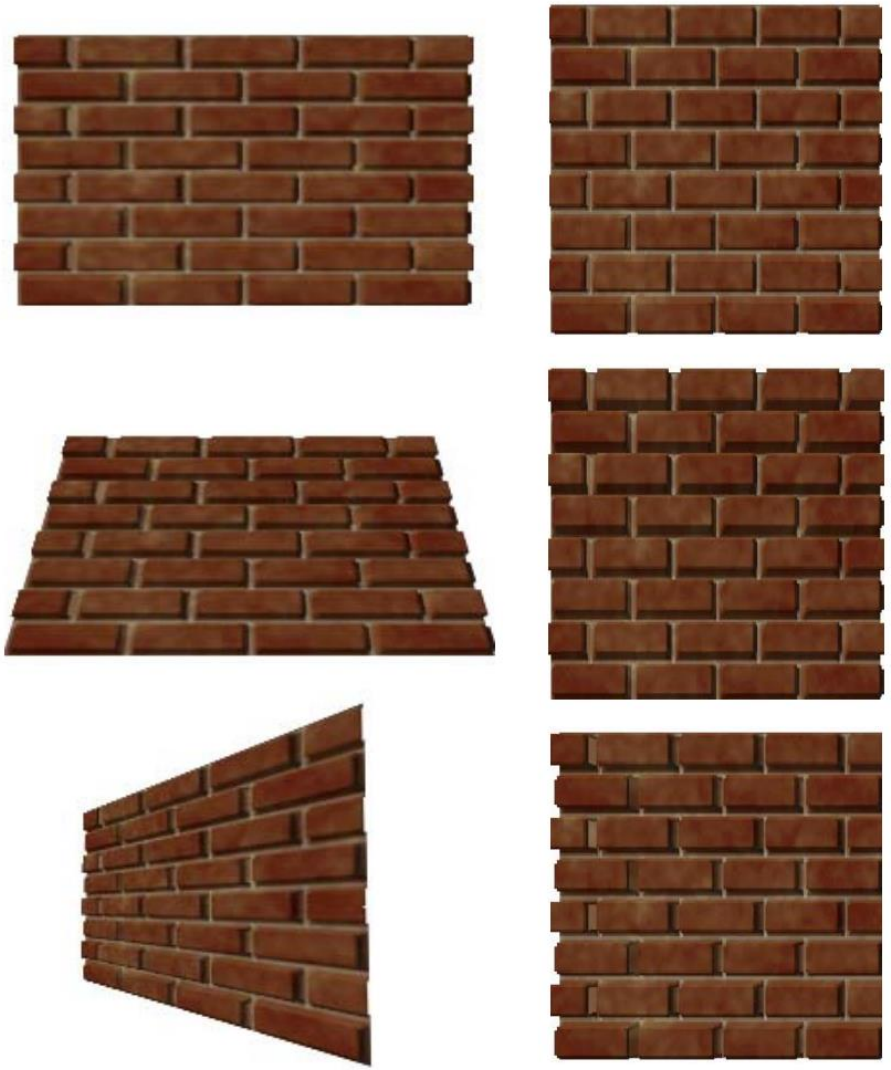
Perception

- Alto Relief

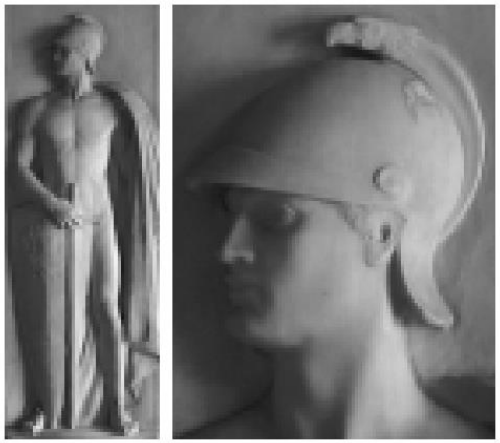
- Sunken Relief

(Relief Images)

- We will see this type of things later…



Perception: Bas-Relief



Perception: Bas-Relief

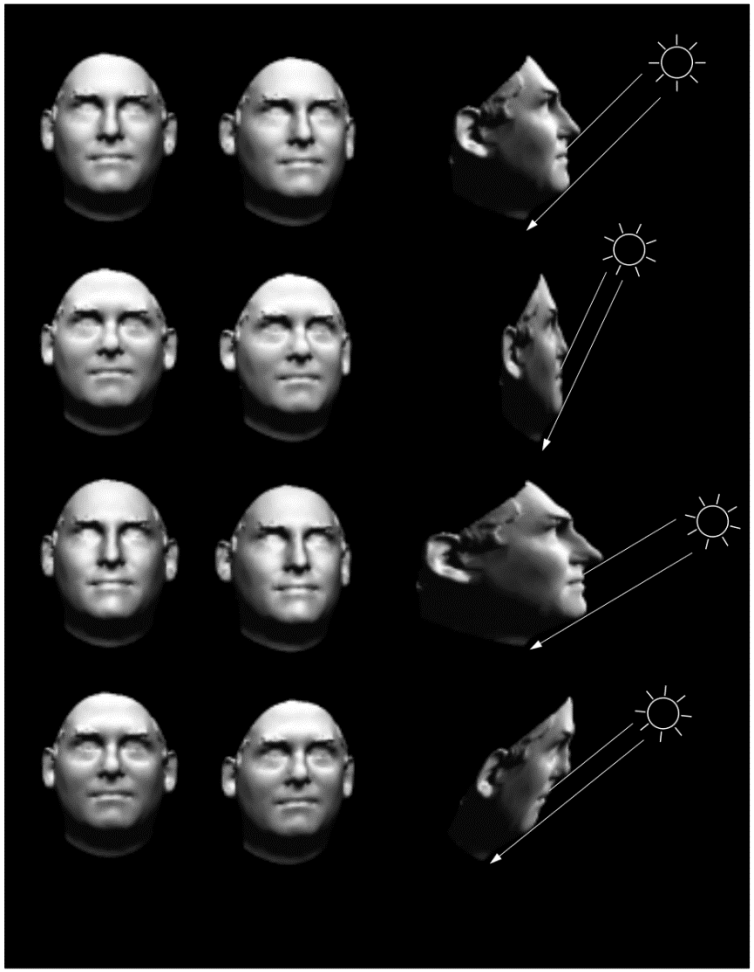
Perception: Bas Relief Ambiguity

$$NL^T = C$$

or

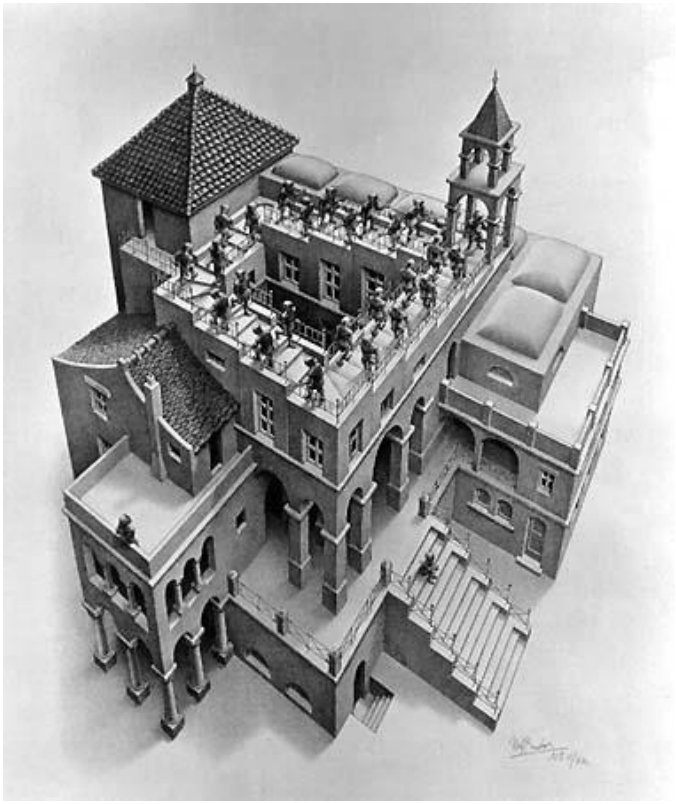
$$NRR^T L^T = C$$
$$NGG^{-1} L^T = C$$

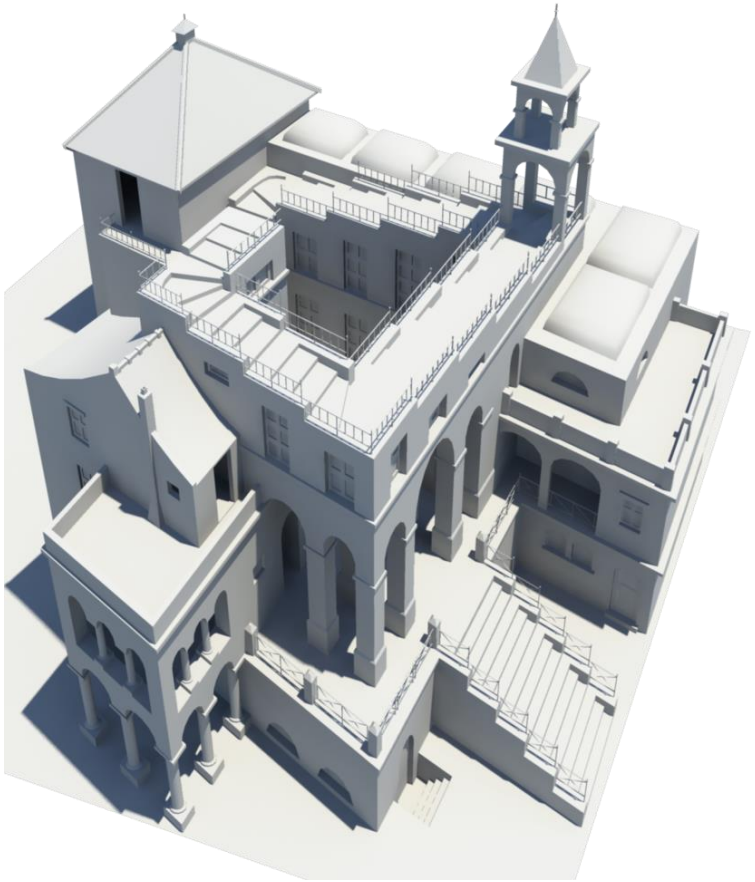
??



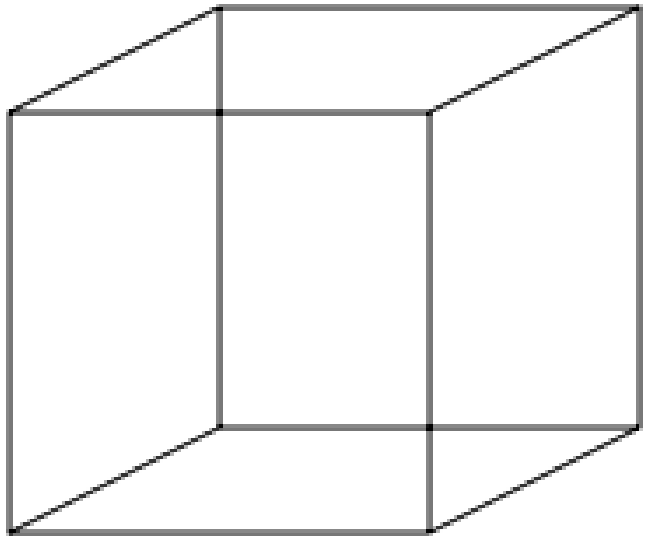
Perception: Inconsistencies

- Escher

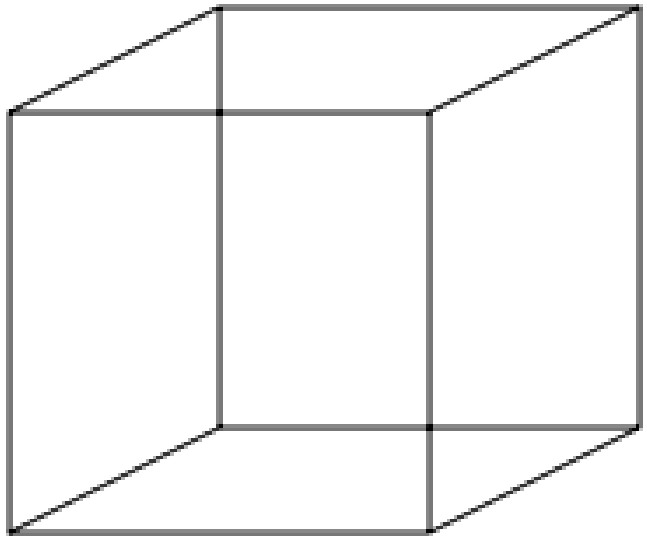


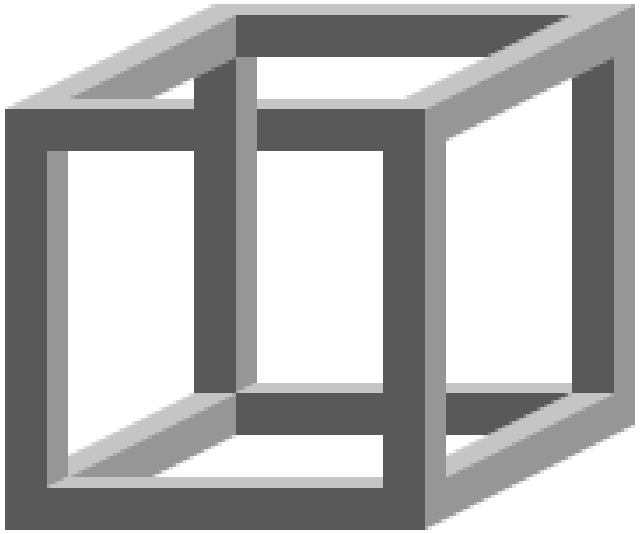


Perception: Nekker Cube

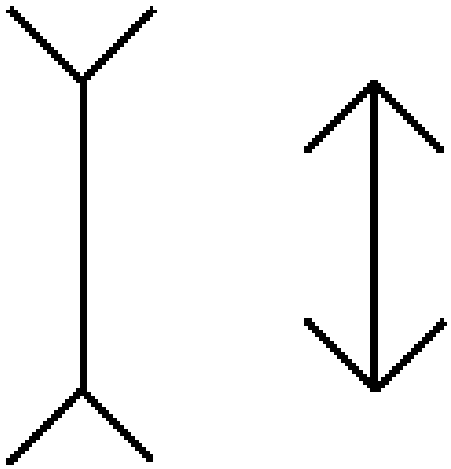


Perception: Nekker Cube

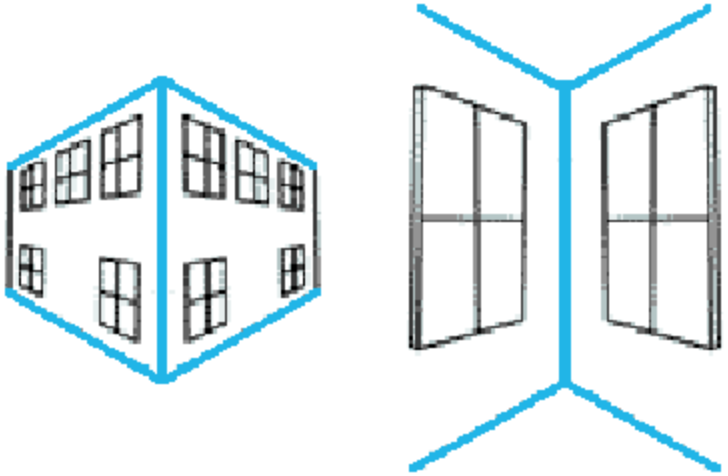




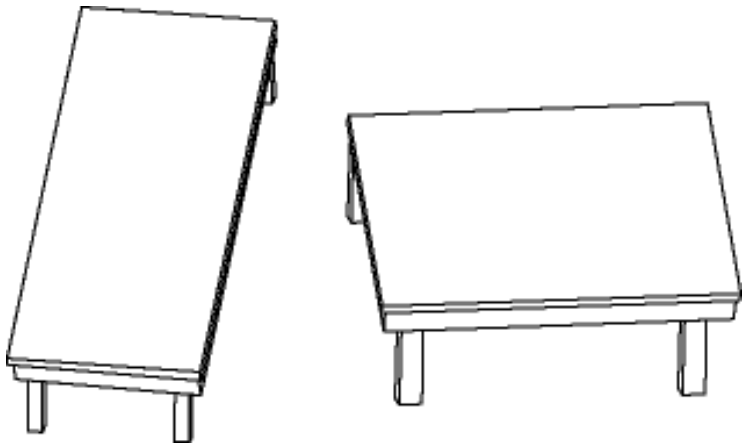
Perception: Muller-Lyer Illusion



Perception: Muller-Lyer Illusion



Perception: Muller-Lyer Illusion



Back to Projection Matrices…

- Basic Perspective Projection:

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Basic Orthographic:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Projection Matrix

- Problem (maybe?):
  - The formulation give does not pass z, or depth, though the transformation pipeline
  - What can we do?

IDEAS????

Options

- Trivial changes?
- Funky matrices…
- Re-mapping options…

Projection Matrices

- Passthrough Z Perspective Projection:

$$\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{If } (Z = n), then\ z = \left(n + f - \frac{nf}{n}\right) = n$$

$$\text{If } (Z = f), then\ z = \left(n + f - \frac{nf}{f}\right) = f$$

Projection Matrices

- To map [-1,+1] to viewing frustum (l,r,b,t,n,f):

$$\begin{bmatrix} 2n/(r-l) & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2n/(t-b) & 0 & -(t+b)/(t-b) \\ 0 & 0 & 2nf/(n-f) & -(n+f)/(n+f) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$