CS5670: Computer Vision

Noah Snavely

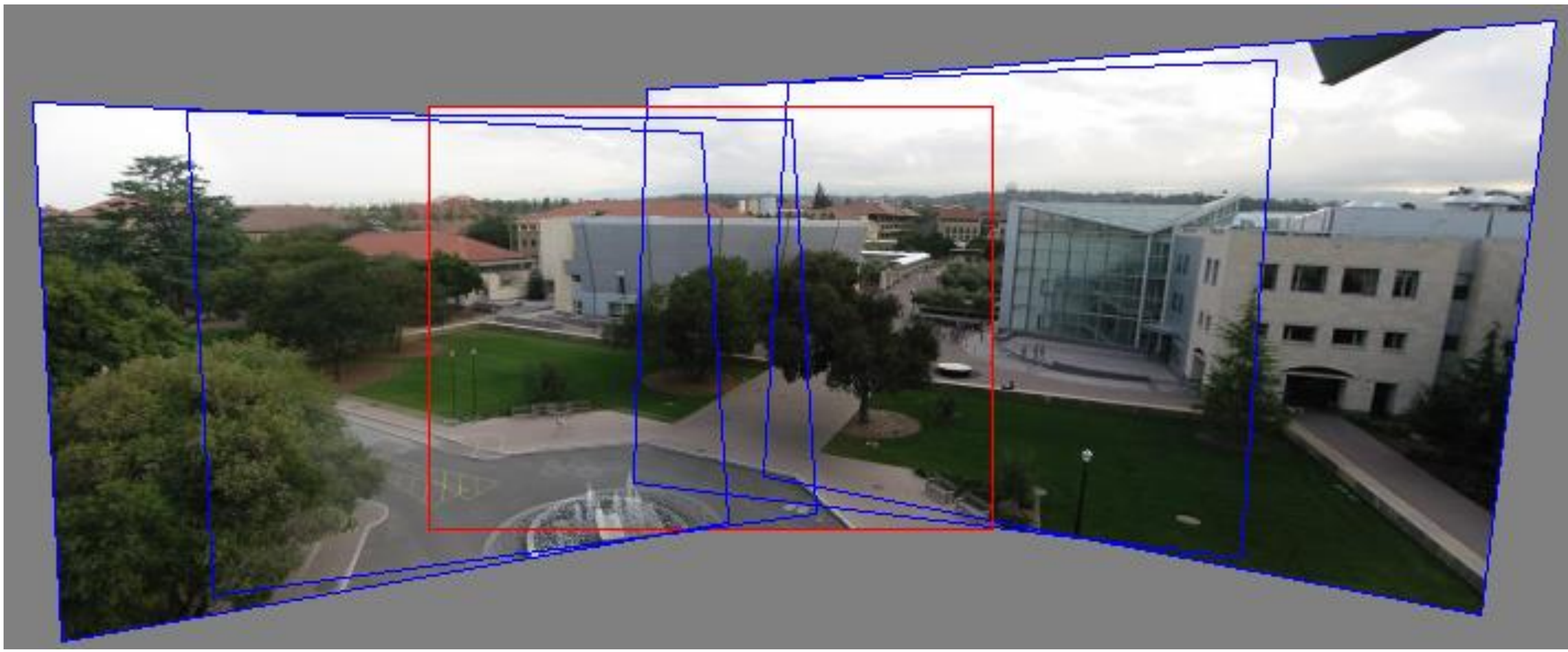
Lecture 10: Cameras

Source: S. Lazebnik

Announcements

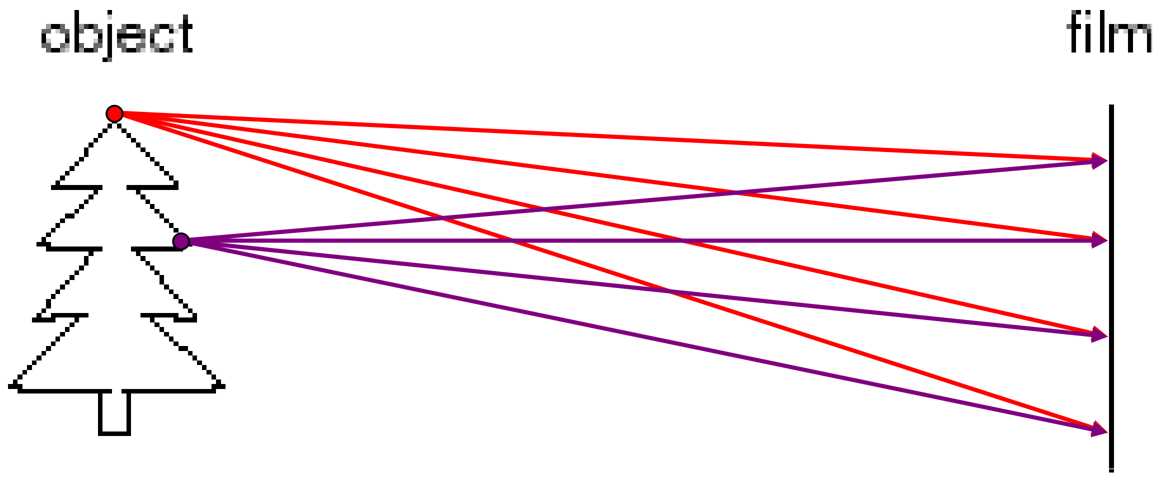
- Take-home midterm
  - To be distributed at the end of class this Wednesday, March 13
  - Due at the beginning of class on Wednesday, March 20
- Project 3: Panorama Stitching
  - Plan to release this week
  - Due on Friday, March 29

Can we use homographies to create a 360 panorama?



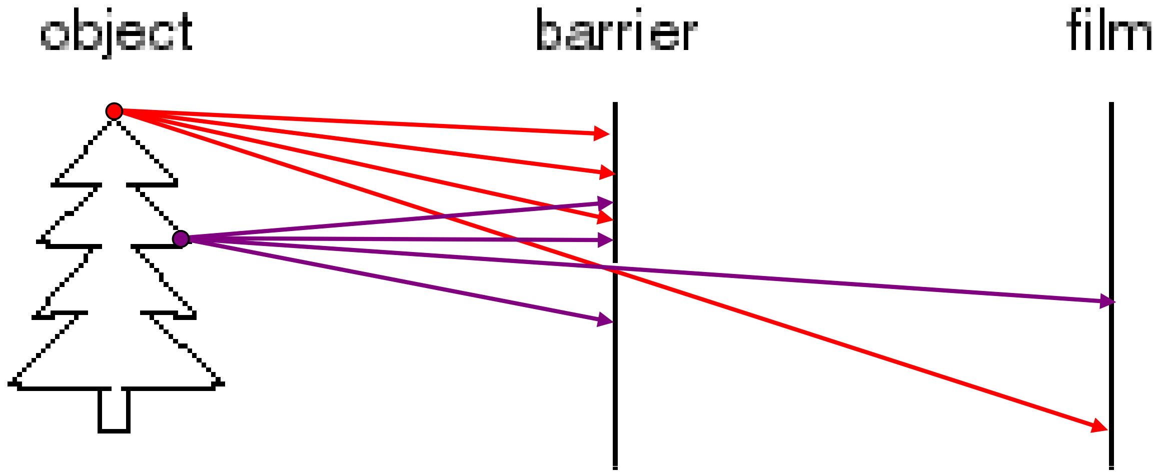
- In order to figure this out, we need to learn what a **camera** is

Image formation



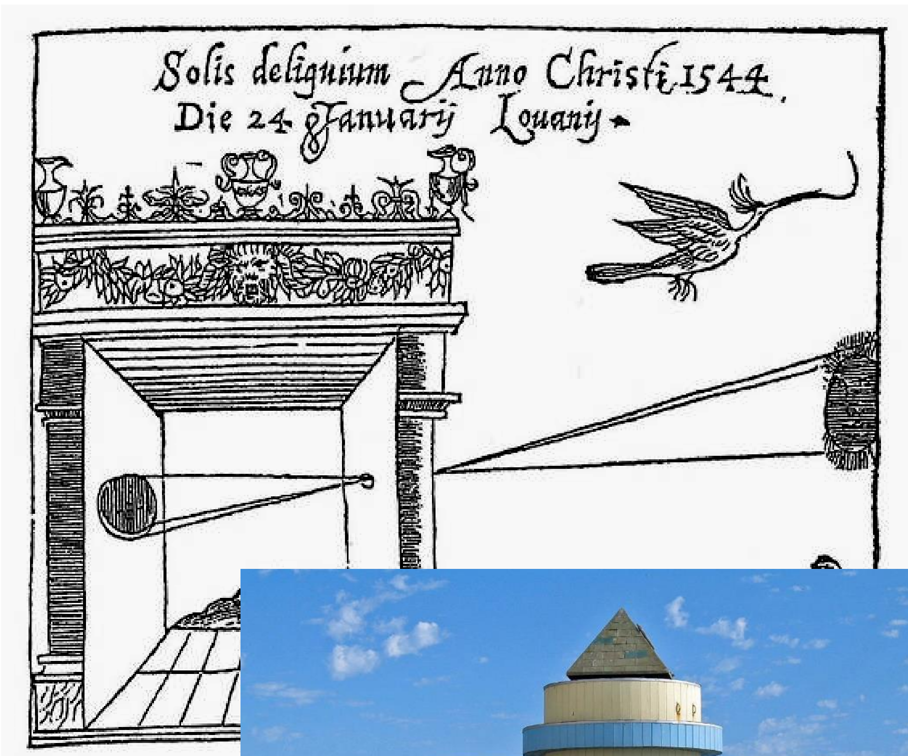
- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?
  - No. This is a bad camera.

Pinhole camera



- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the **aperture**
  - How does this transform the image?

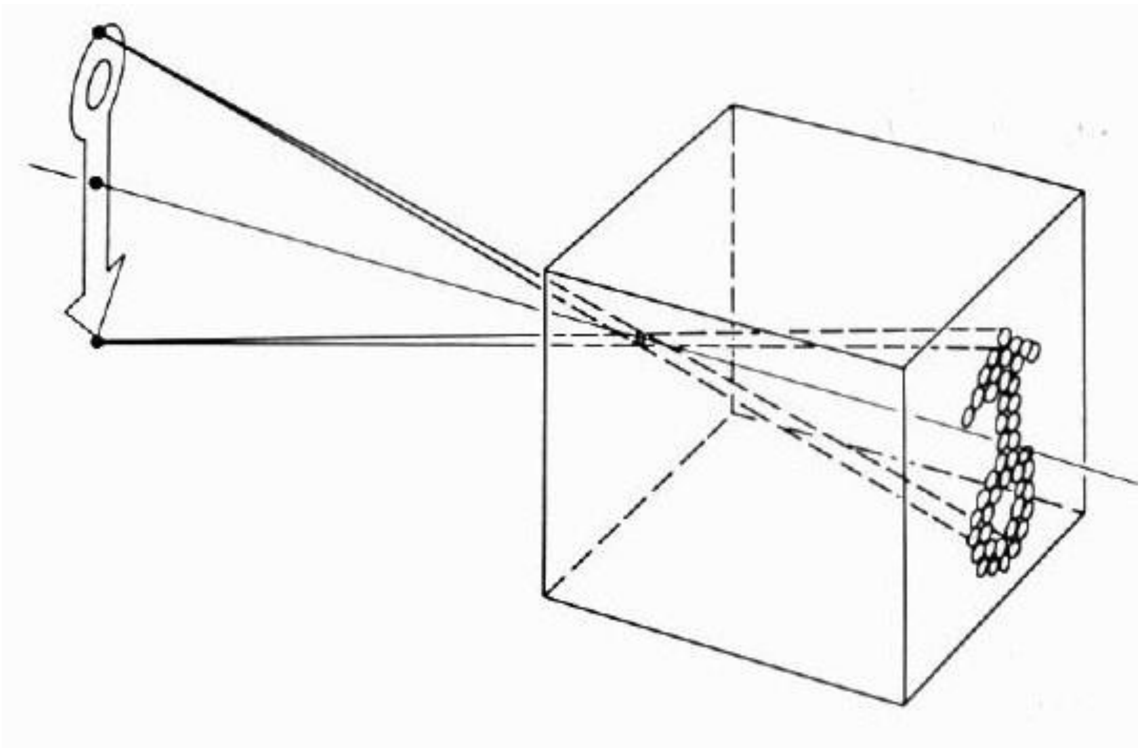
Camera Obscura



- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

Source: A. Efros

Camera Obscura



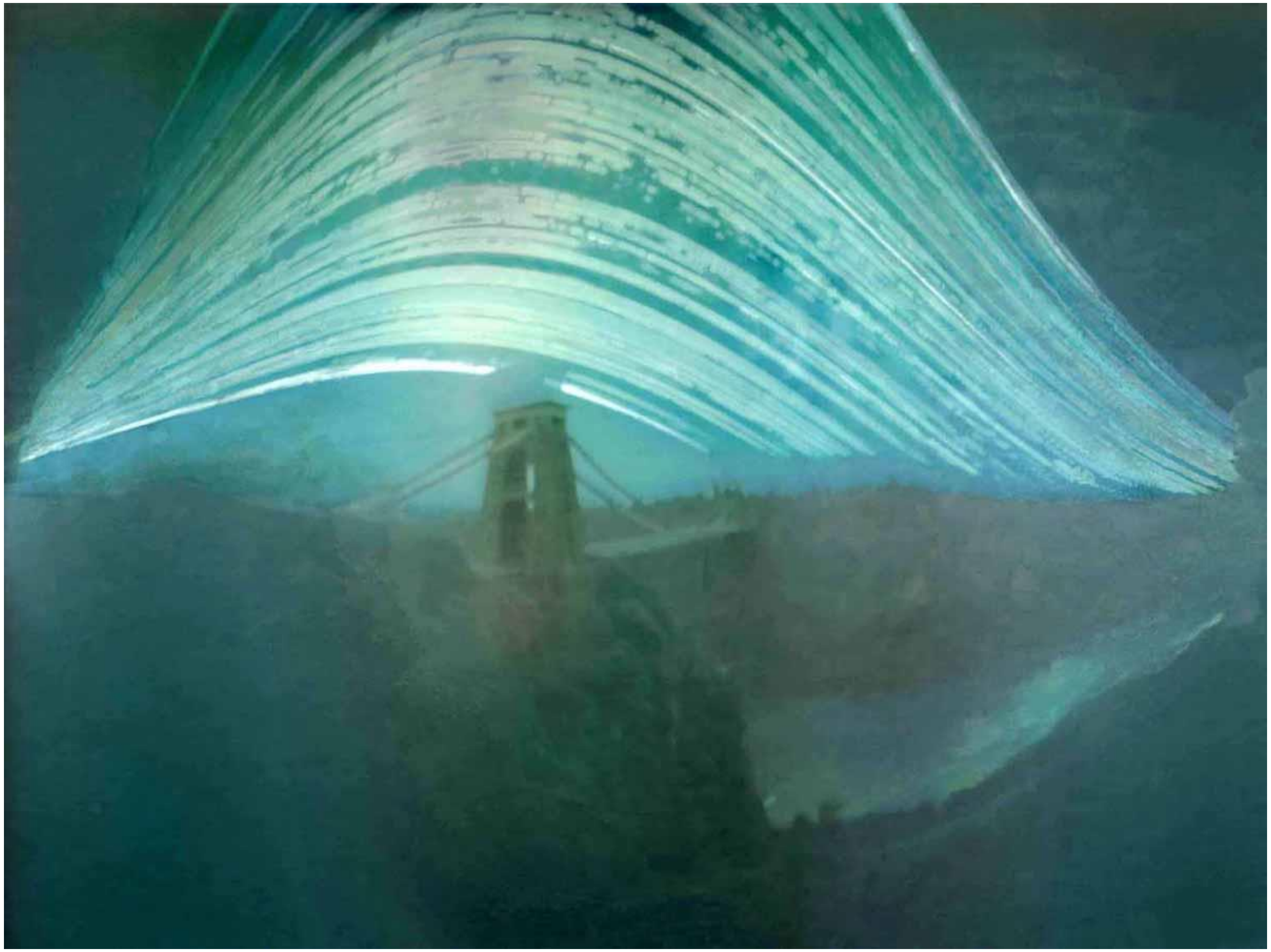
Home-made pinhole camera

Why so blurry?

http://www.debevec.org/Pinhole/

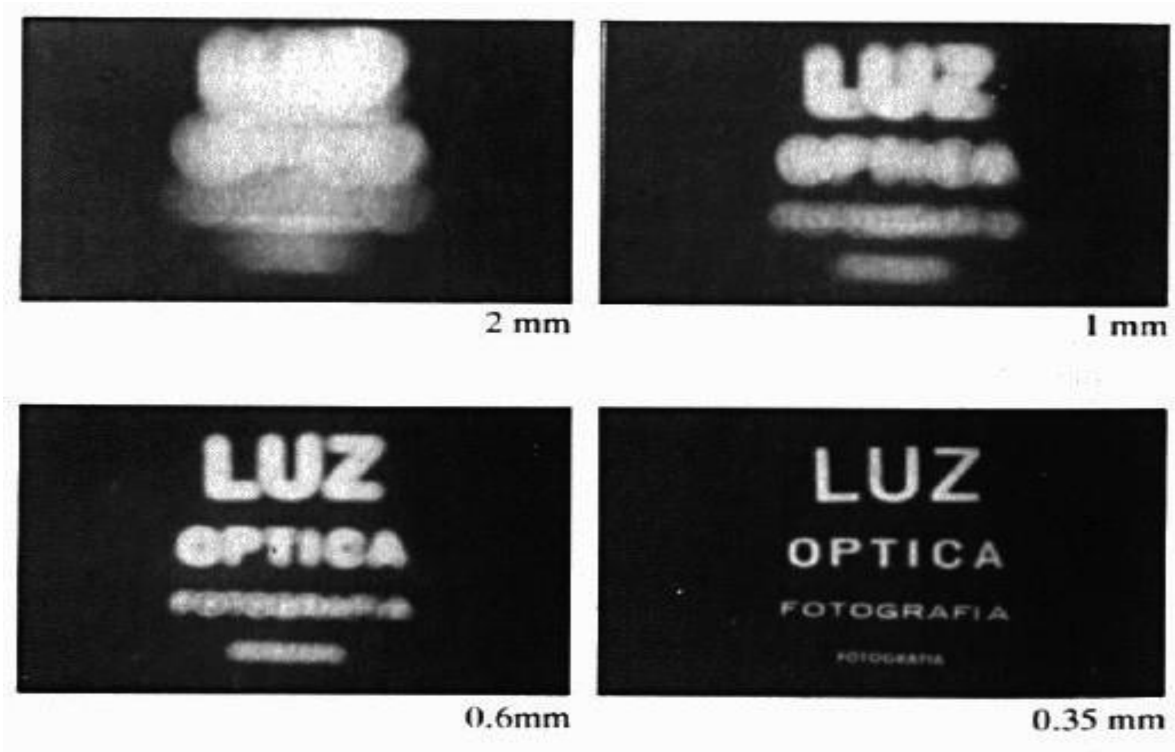
Slide by A. Efros

Pinhole photography



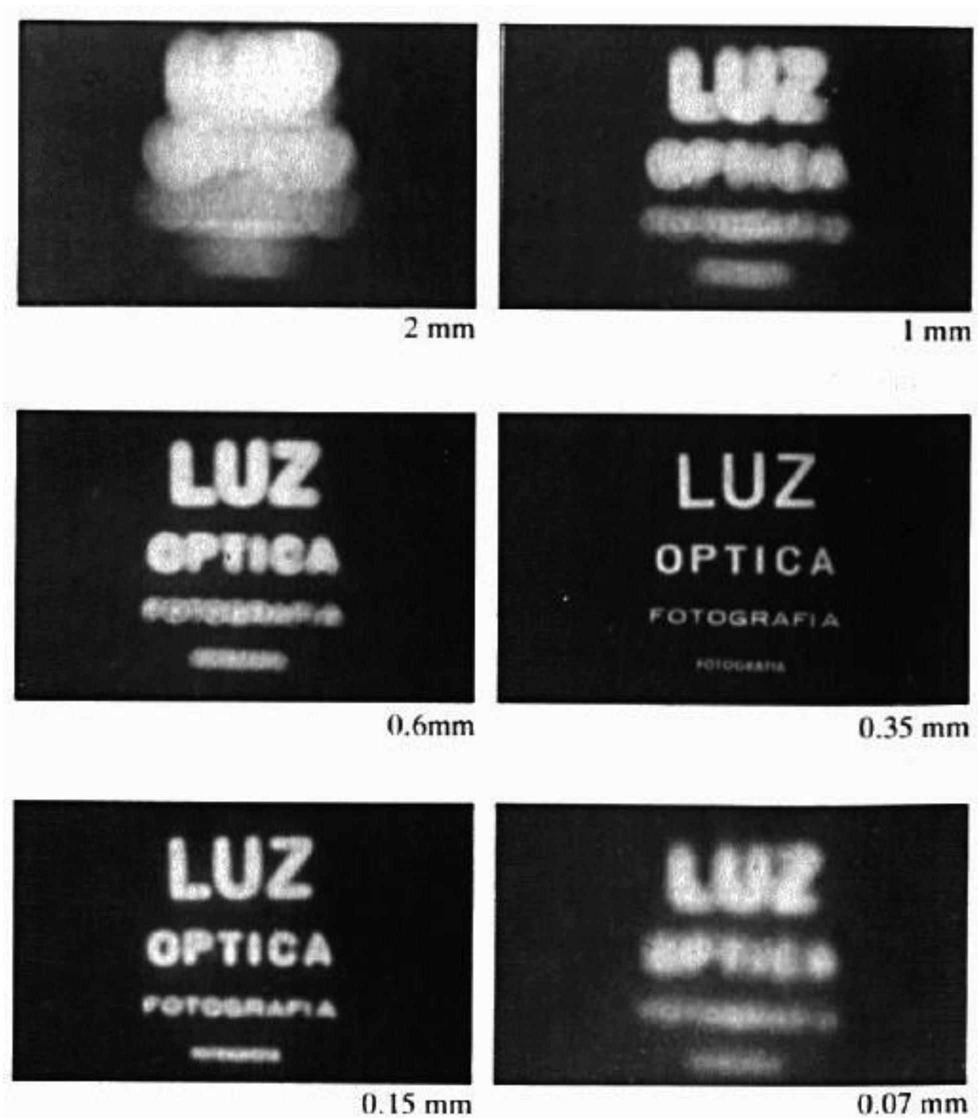
Justin Quinnell, The Clifton Suspension Bridge. December 17th 2007 - June 21st 2008
***6-month* exposure**

Shrinking the aperture

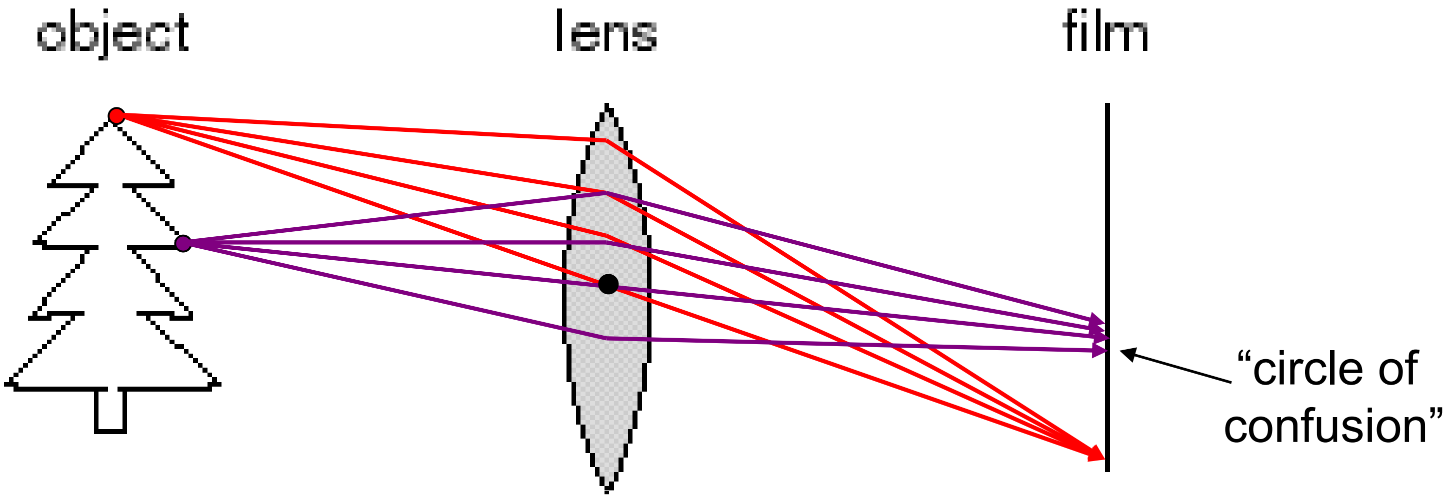


- Why not make the aperture as small as possible?
  - Less light gets through
  - *Diffraction* effects...

Shrinking the aperture

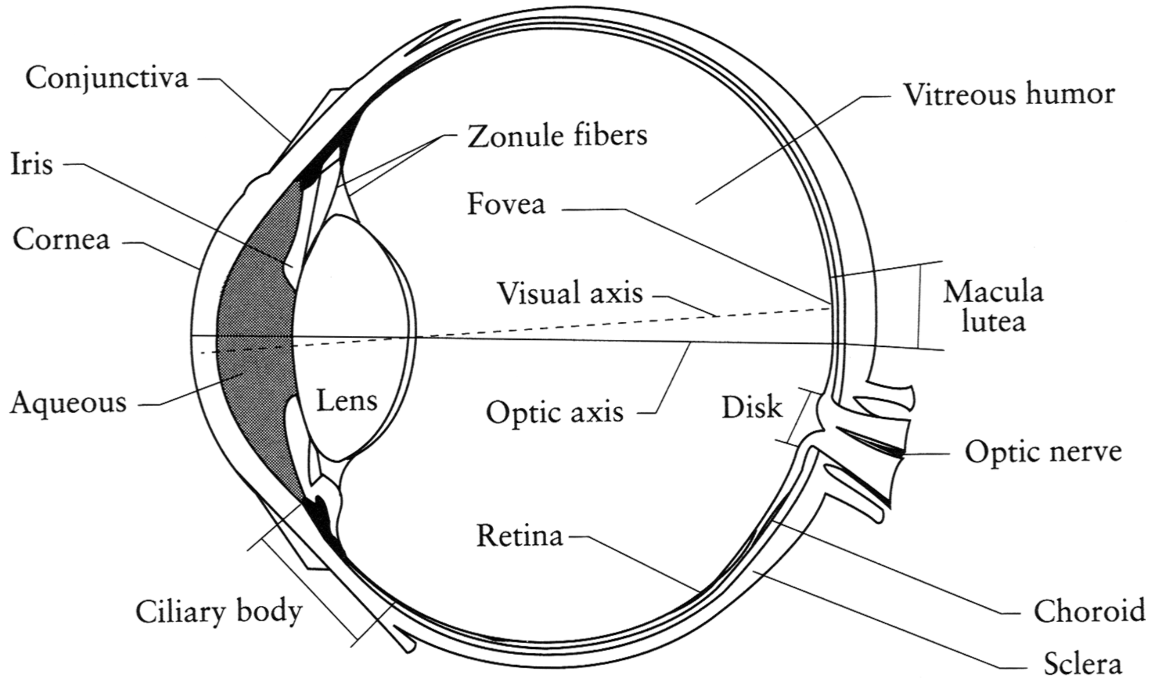


Adding a lens



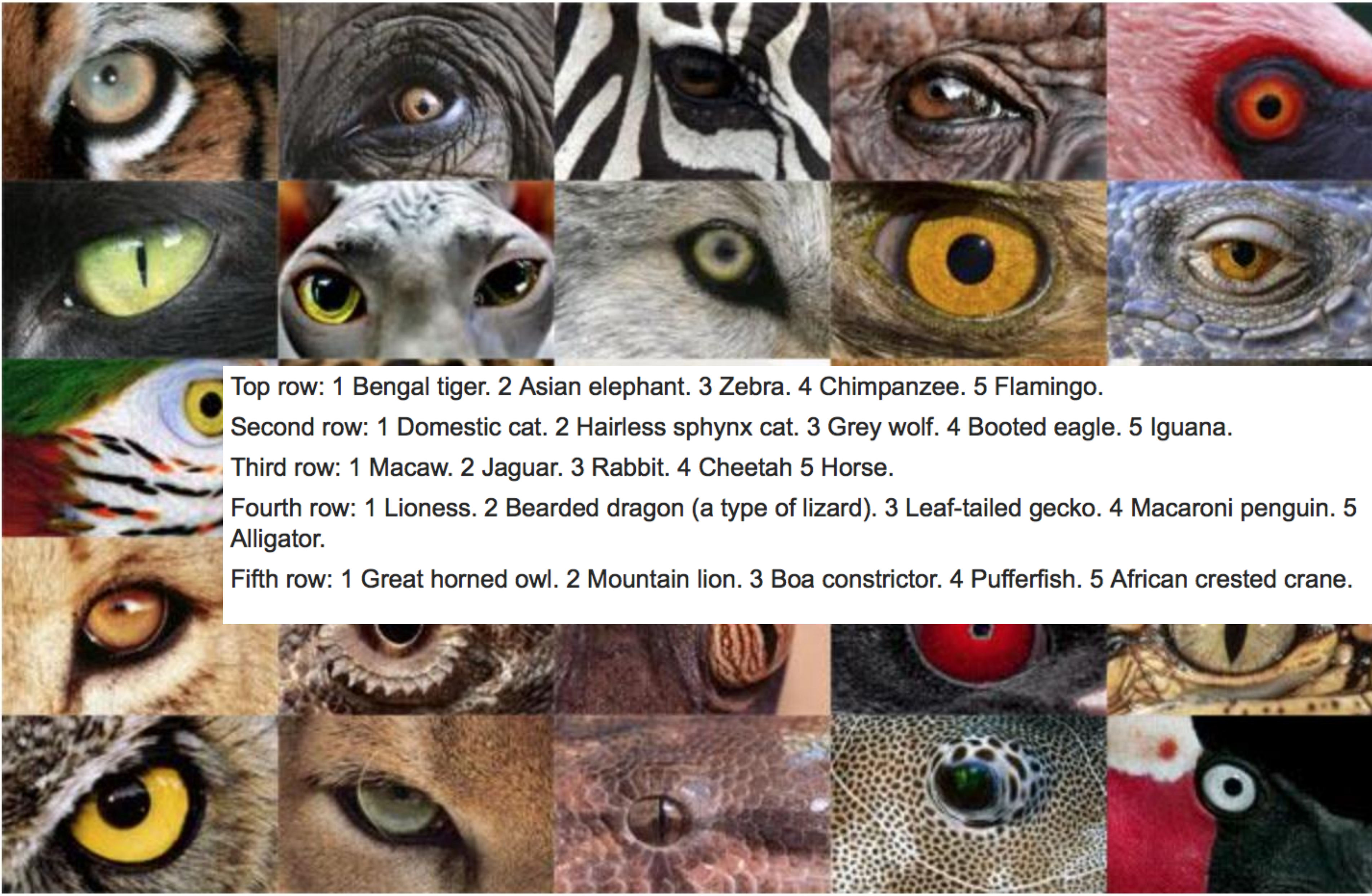
- A lens focuses light onto the film
  - There is a specific distance at which objects are "in focus"
    - other points project to a "circle of confusion" in the image
  - Changing the shape of the lens changes this distance

The eye



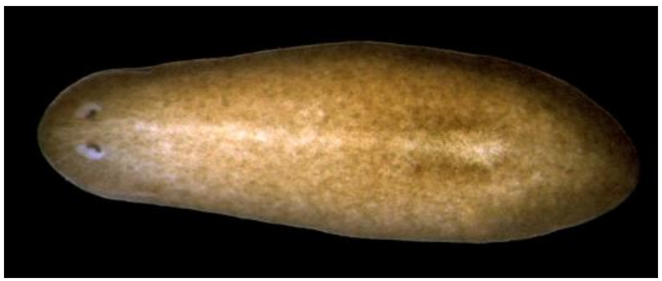
- The human eye is a camera
  - **Iris** - colored annulus with radial muscles
  - **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What's the "film"?
    - photoreceptor cells (rods and cones) in the **retina**

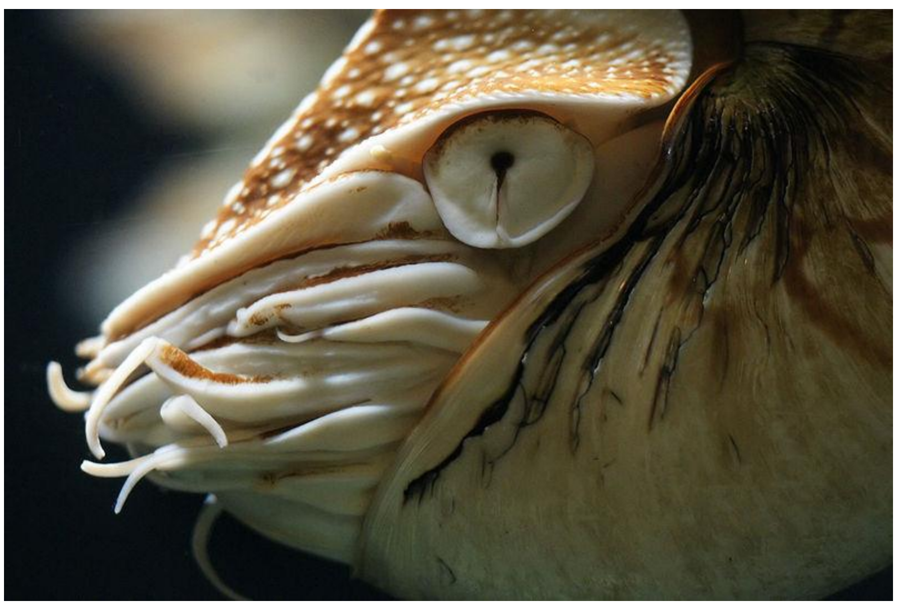
http://www.telegraph.co.uk/news/earth/earthpicturegalleries/7598120/Animal-eyes-quiz-Can-you-work-out-which-creatures-these-are-from-their-eyes.html?image=25



http://www.telegraph.co.uk/news/earth/earthpicturegalleries/7598120/Animal-eyes-quiz-Can-you-work-out-which-creatures-these-are-from-their-eyes.html?image=25

Eyes in nature: eyespots to pinhole



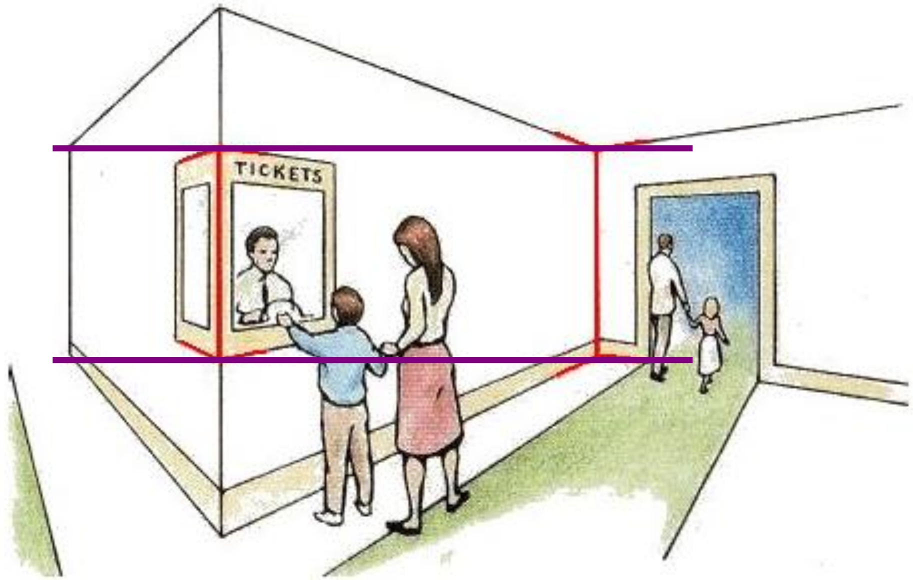


http://upload.wikimedia.org/wikipedia/commons/6/6d/Mantis_shrimp.jpg

Projection

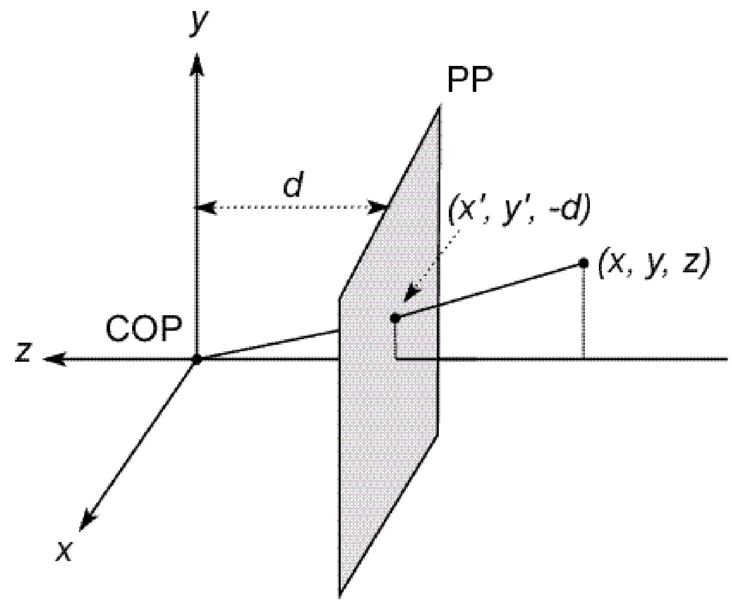
Projection

Müller-Lyer Illusion



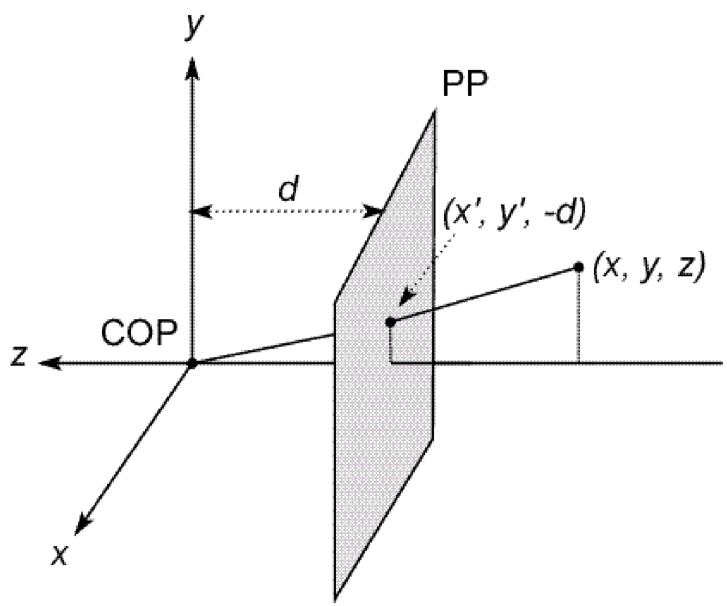
http://www.michaelbach.de/ot/sze_muelue/index.html

Modeling projection



- The coordinate system
  - We will use the pinhole model as an approximation
  - Put the optical center (**C**enter **O**f **P**rojection) at the origin
  - Put the image plane (**P**rojection **P**lane) *in front* of the COP
    - Why?
  - The camera looks down the *negative* z axis
    - we like this if we want right-handed-coordinates

Modeling projection



- Projection equations
  - Compute intersection with PP of ray from (x,y,z) to COP
  - Derived using similar triangles (on board)

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z},\ -d)$$

  - We get the projection by throwing out the last coordinate:

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$$

Modeling projection

- Is this a linear transformation?
  - no—division by z is nonlinear

Homogeneous coordinates to the rescue—again!

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Perspective Projection

Projection is a matrix multiply using homogeneous coordinates:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \quad -d\frac{y}{z})$$

divide by third coordinate

This is known as **perspective projection**

- The matrix is the **projection matrix**

- (Can also represent as a 4x4 matrix – OpenGL does something like this)

Perspective Projection

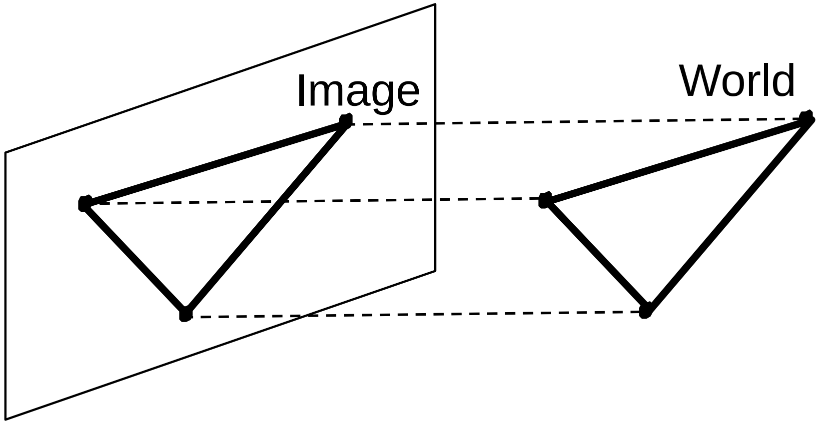
- How does scaling the projection matrix change the transformation?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \quad -d\frac{y}{z})$$

$$\begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -dx \\ -dy \\ z \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \quad -d\frac{y}{z})$$

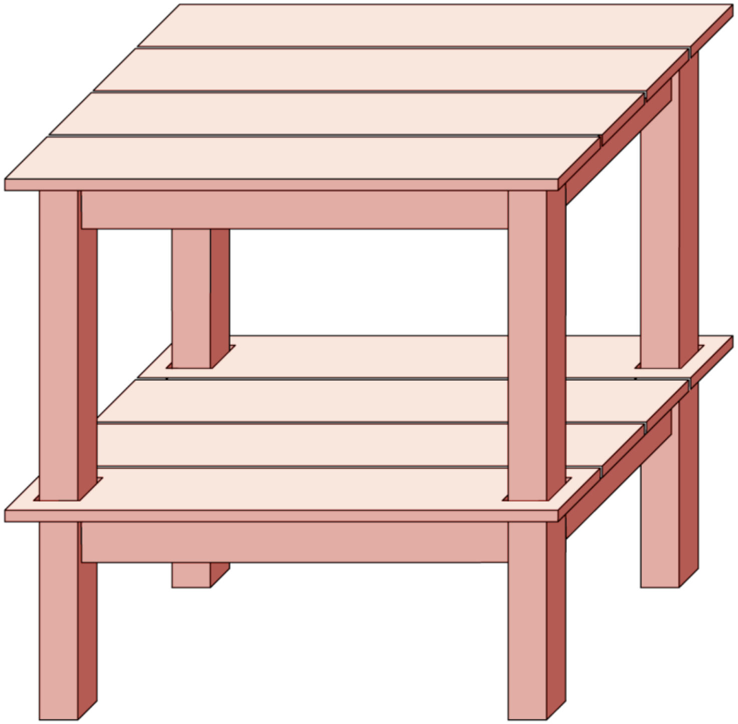
Orthographic projection

- Special case of perspective projection
  - Distance from the COP to the PP is infinite

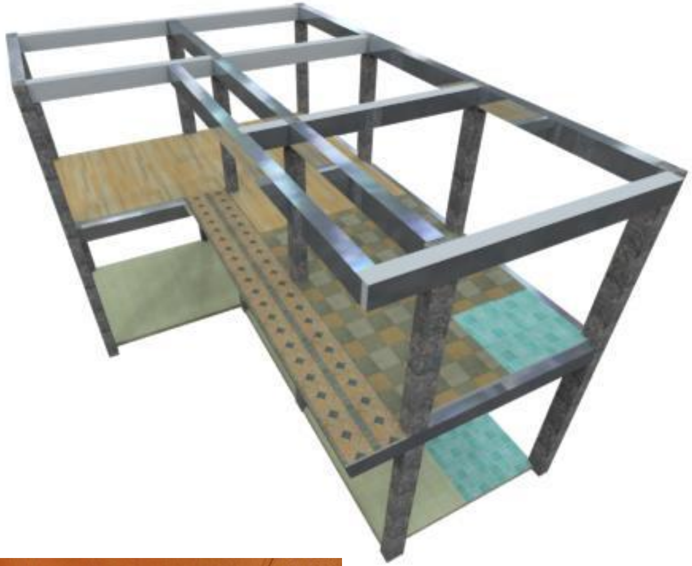


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Orthographic projection



Perspective projection



Variants of orthographic projection

- Scaled orthographic
  - Also called "weak perspective"

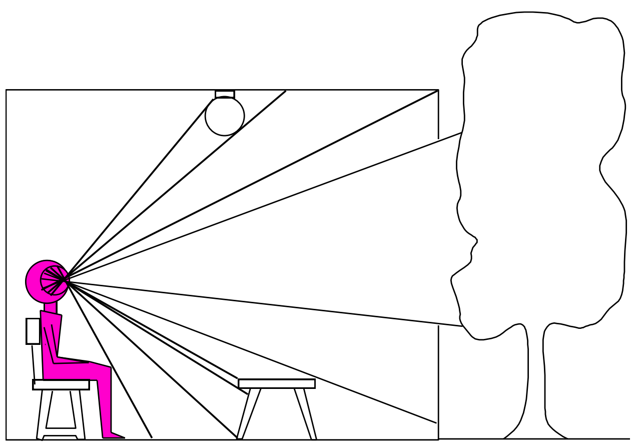
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/d \end{bmatrix} \Rightarrow (dx, dy)$$

- Affine projection
  - Also called "paraperspective"

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

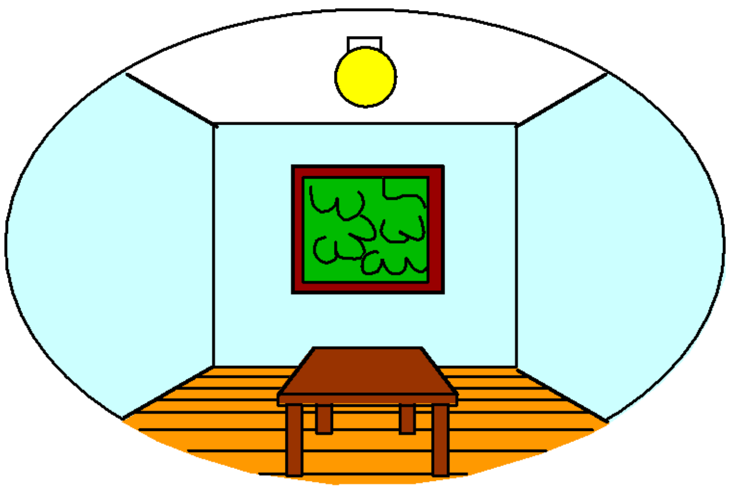
Dimensionality Reduction Machine (3D to 2D)

3D world



Point of observation

2D image



What have we lost?

- Angles
- Distances (lengths)

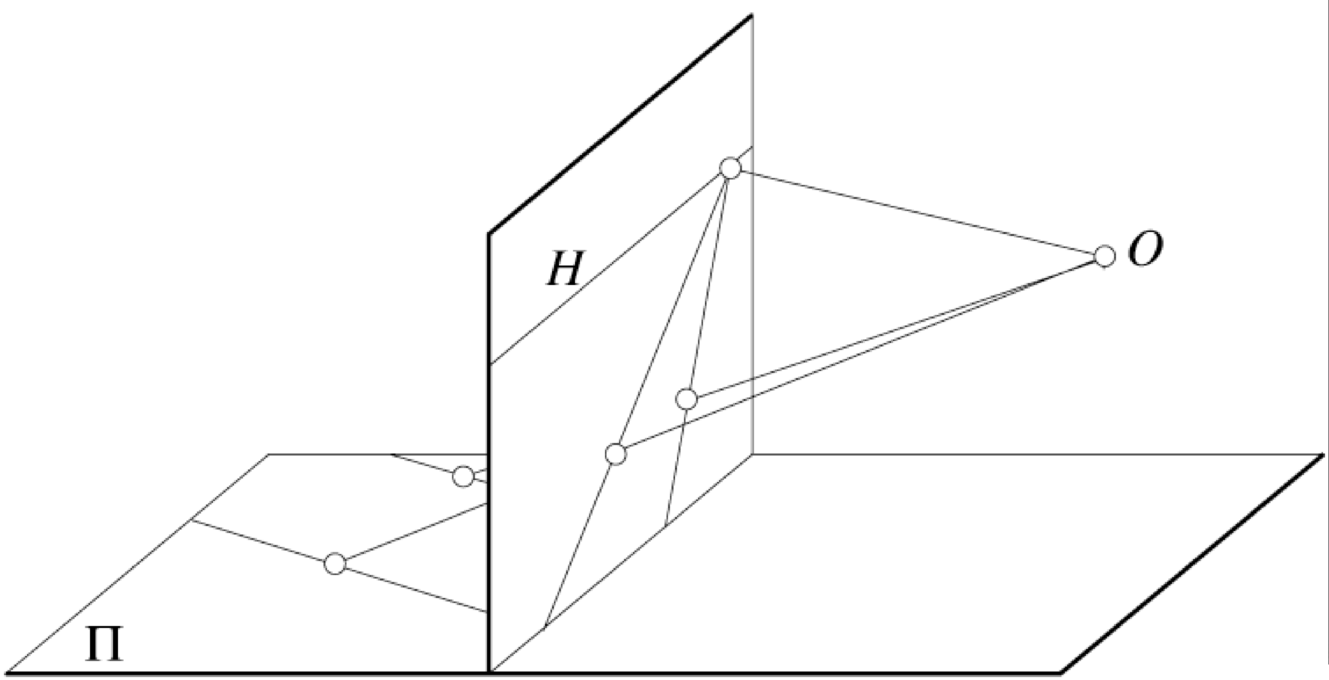
Slide by A. Efros

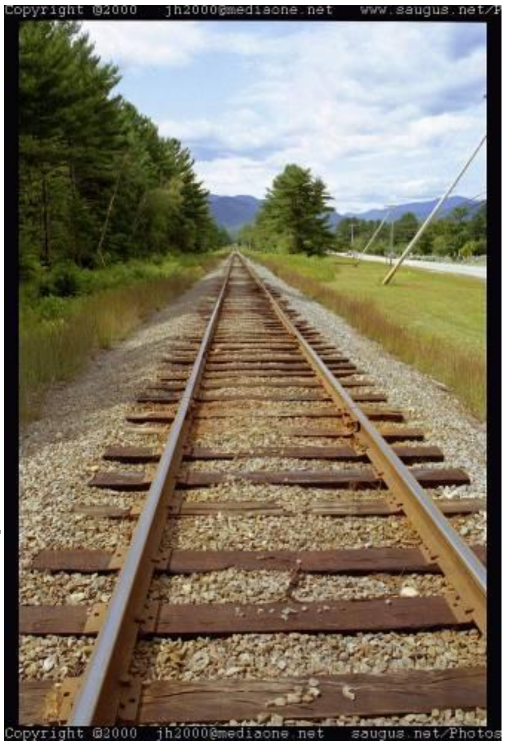
Projection properties

- Many-to-one: any points along same ray map to same point in image
- Points → points
- Lines → lines (collinearity is preserved)
  - But line through focal point projects to a point
- Planes → planes (or half-planes)
  - But plane through focal point projects to line

Projection properties

- Parallel lines converge at a vanishing point
  - Each direction in space has its own vanishing point
  - But parallels parallel to the image plane remain parallel

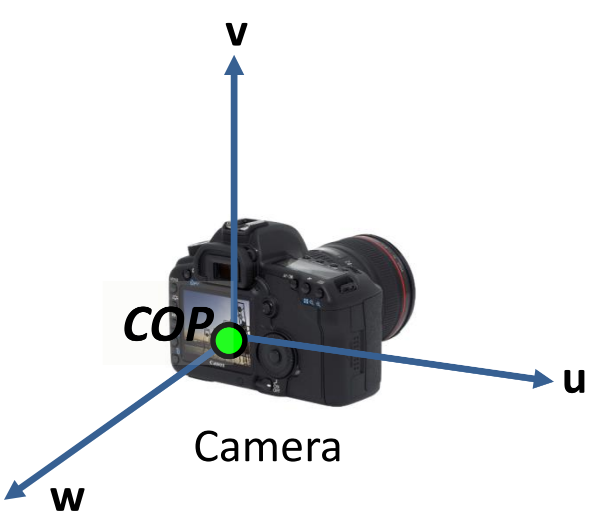




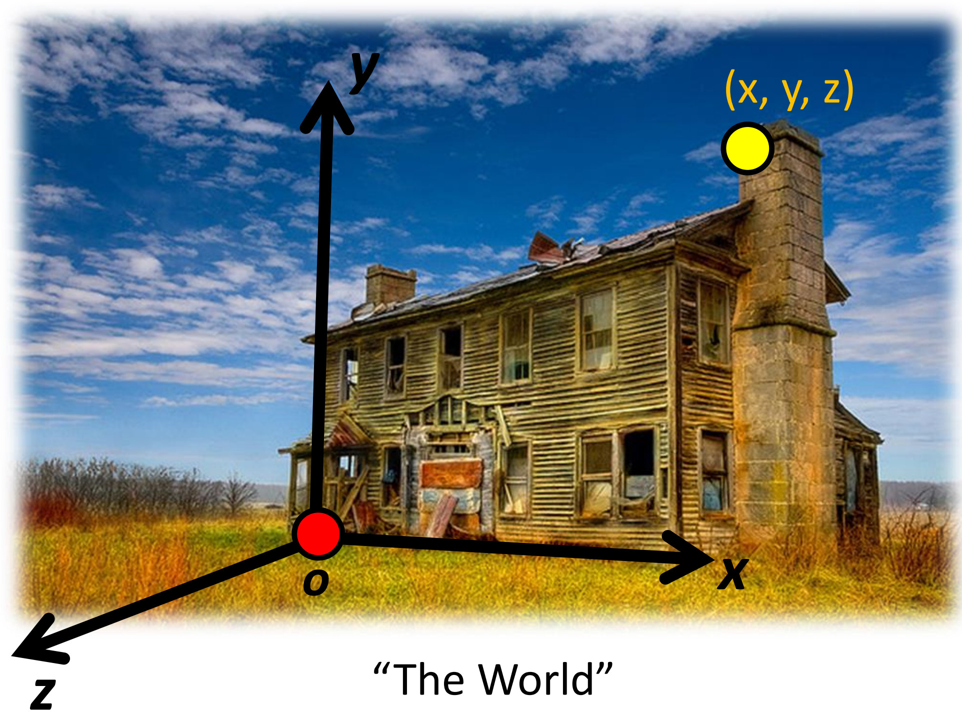
Questions?

Camera parameters

- How can we model the geometry of a camera?



Camera



“The World”

Two important coordinate systems:

1. *World* coordinate system
2. *Camera* coordinate system

How do we project a given point (x, y, z) in world coordinates?

Camera parameters

- To project a point (x,y,z) in *world* coordinates into a camera
- First transform (x,y,z) into *camera* coordinates
- Need to know
  - Camera position (in world coordinates)
  - Camera orientation (in world coordinates)
- Then project into the image plane to get a pixel coordinate
  - Need to know camera *intrinsics*

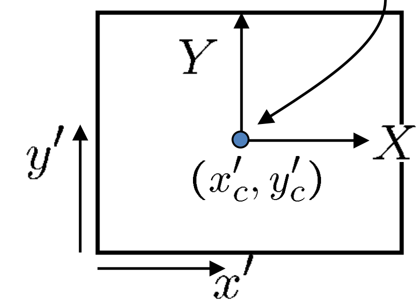
Camera parameters

A camera is described by several parameters

- Translation T of the optical center from the origin of world coords
- Rotation R of the image plane
- focal length f, principal point (x'_c, y'_c), pixel size (s_x, s_y)
- blue parameters are called “extrinsics,” red are “intrinsics”

Projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



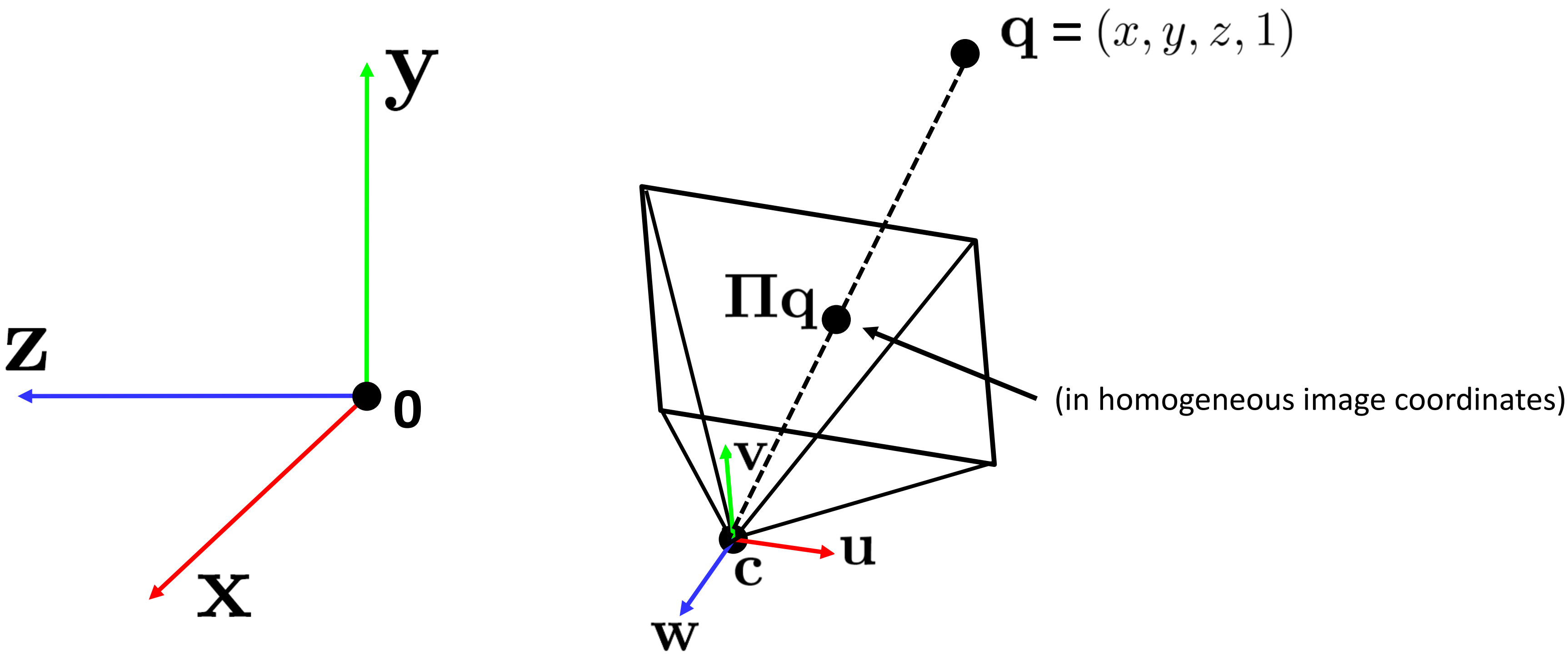
- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

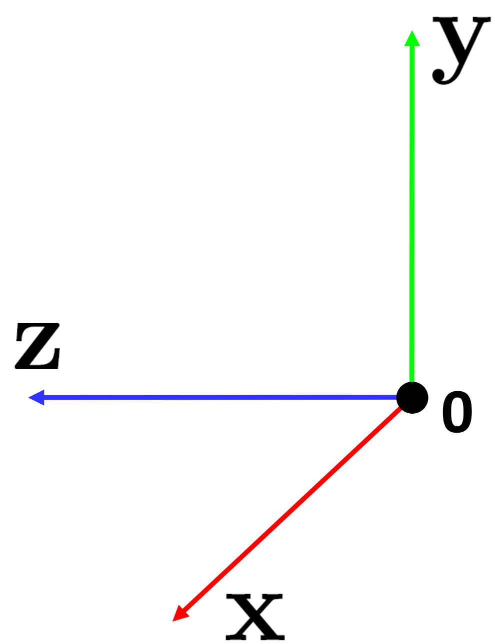
- The definitions of these parameters are **not** completely standardized
  - especially intrinsics—varies from one book to another

Projection matrix

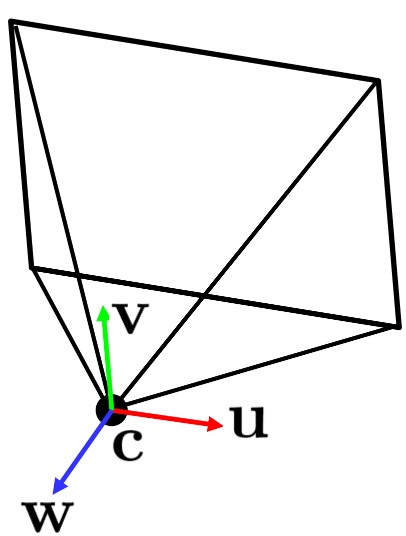


Extrinsics

- How do we get the camera to "canonical form"?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

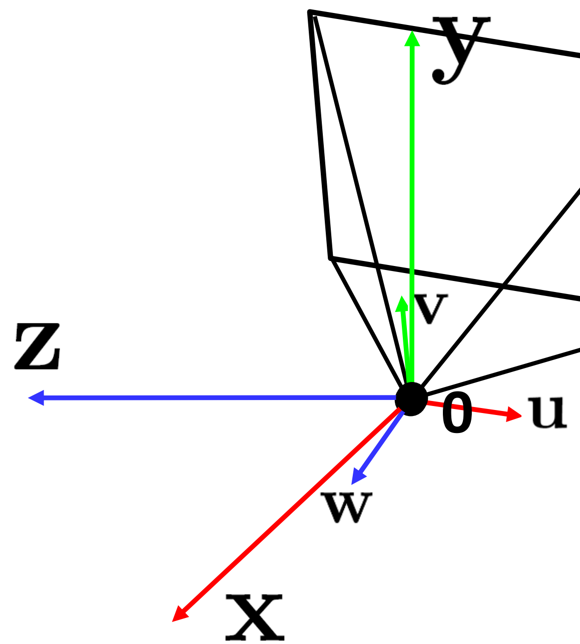


Step 1: Translate by -**c**



Extrinsics

- How do we get the camera to “canonical form”?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



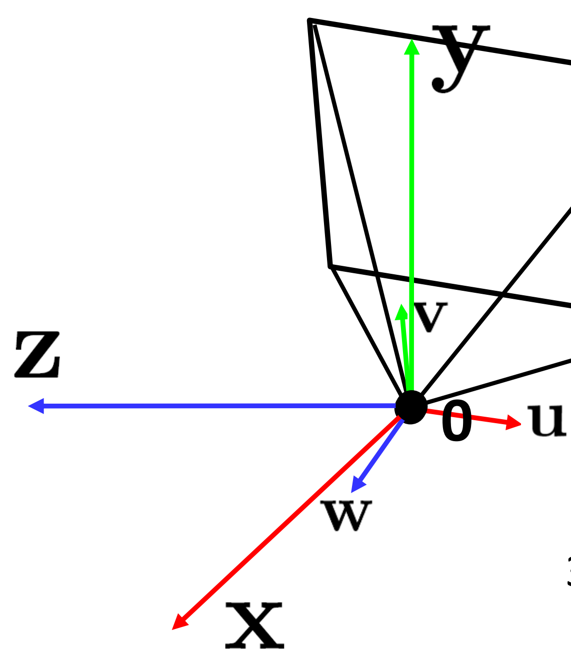
Step 1: Translate by -**c**

How do we represent translation as a matrix multiplication?

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Extrinsics

- How do we get the camera to “canonical form”?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



Step 1: Translate by -**c**

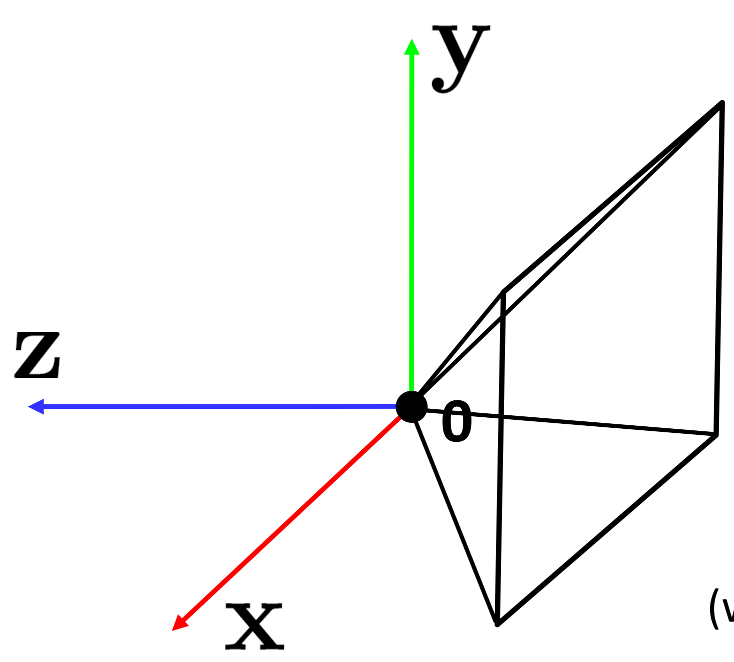
Step 2: Rotate by **R**

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

3x3 rotation matrix

Extrinsics

- How do we get the camera to "canonical form"?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



Step 1: Translate by -**c**

Step 2: Rotate by **R**

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

(with extra row/column of [0 0 0 1])

Perspective projection

$$\underbrace{\begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\mathbf{K} \\ \text{(intrinsics)}}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(converts from 3D rays in camera coordinate system to pixel coordinates)

in general, $\mathbf{K} = \begin{bmatrix} -f & s & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$ (upper triangular matrix)

α : **aspect ratio** (1 unless pixels are not square)

s : **skew** (0 unless pixels are shaped like rhombi/parallelograms)

(c_x, c_y) : **principal point** ((0,0) unless optical axis doesn't intersect projection plane at origin)

Focal length

- Can think of as "zoom"

24mm

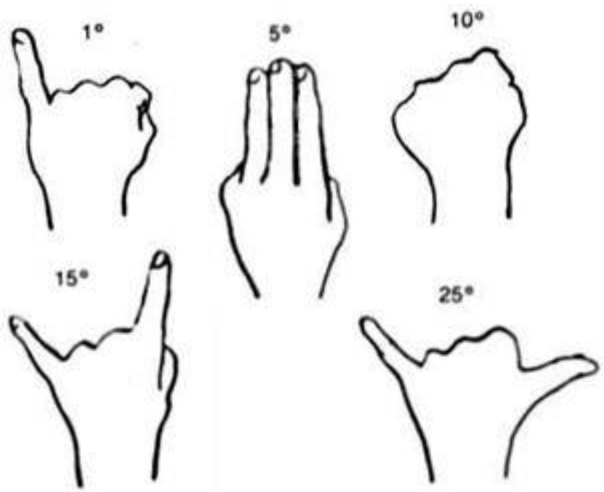
50mm

200mm

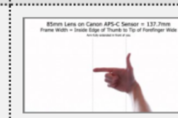
800mm

- Also related to *field of view*

Field of view

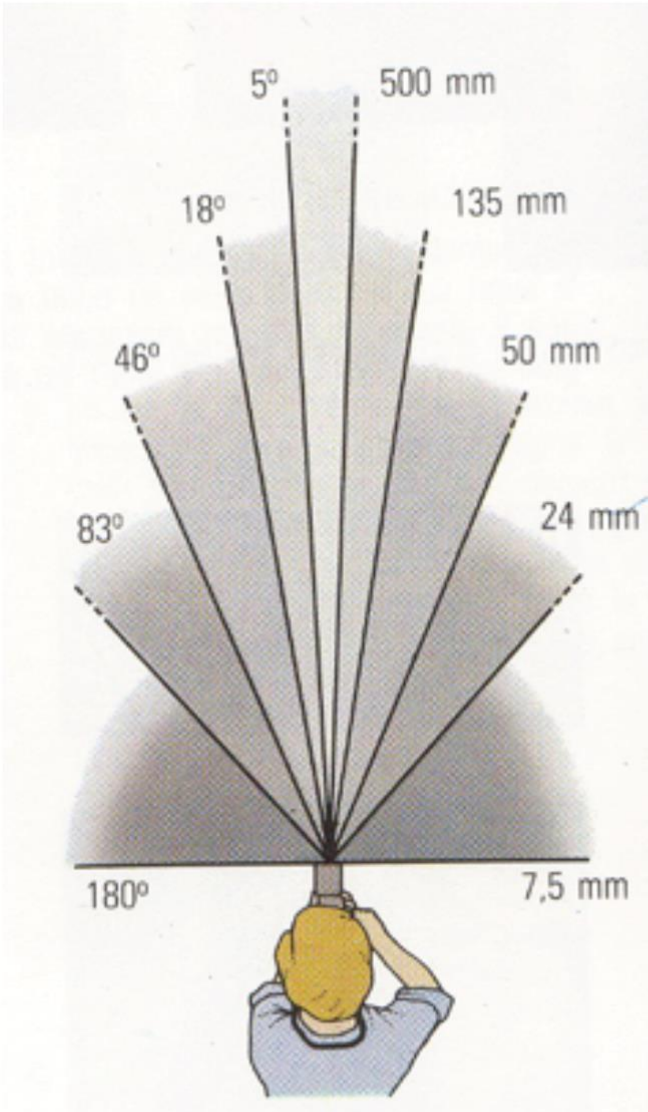


APS-C Crop Body Measurement Table

Lens	After 1.62 Multiplier	APS-C Sensor (1.62 lens multiplier) Canon 60D, 7D, 70D, T3i, T4i	Hand Positions
18mm	29.16mm	**Three hands wide** at full arms length.	
28mm	45.36mm	**Slightly less than two hands wide** at full arms length.	
35mm	56.7mm	**One hand + width of one fist** at full arms length.	
50mm	81mm	**One hand wide + width of thumb** at full arms length.	
55mm	89.1mm	**Slightly less than one hand wide** at full arms length.	
85mm	137.7mm	**Inside edge of thumb to tip of forefinger wide** with hand in "L" shape, thumb up.	

http://www.dslrsolutions.net/3298/using-your-hands-as-focal-length-calculator/

Focal length in practice



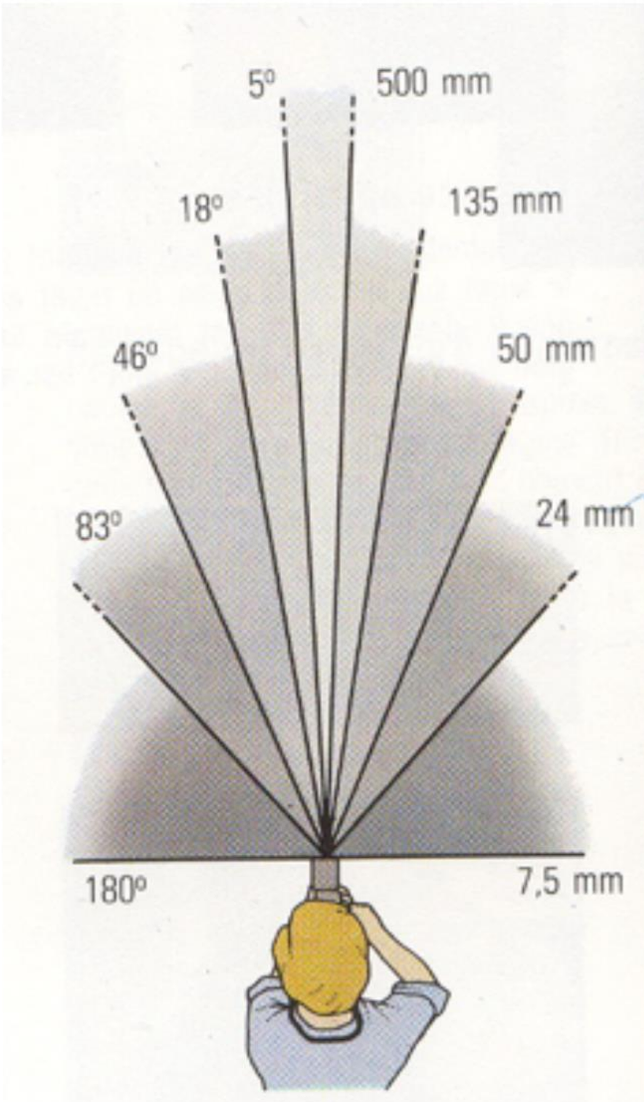
24mm

50mm

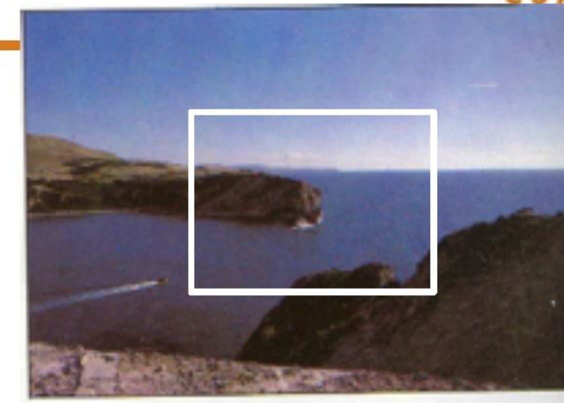
135mm

Fredo Durand

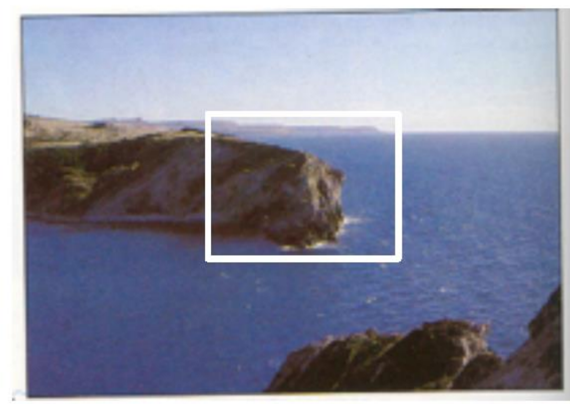
Focal length = cropping



24mm



50mm

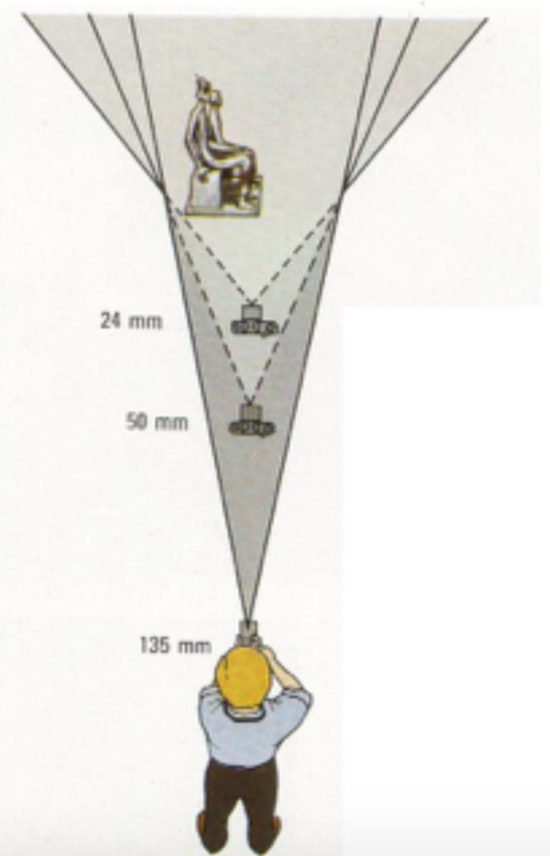


135mm

Fredo Durand

Focal length vs. viewpoint

- Telephoto makes it easier to select background (a small change in viewpoint is a big change in background.



Grand-angulaire 24 mm

Normal 50 mm

Longue focale 135 mm

Fredo Durand


Fredo Durand


Wide angle


Standard


Telephoto

http://petapixel.com/2013/01/11/how-focal-length-affects-your-subjects-apparent-weight-as-seen-with-a-cat/

Projection matrix

$$\mathbf{\Pi} = \underset{\text{intrinsics}}{\mathbf{K}} \underbrace{\underset{\text{projection}}{\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}} \underset{\text{rotation}}{\begin{bmatrix}\mathbf{R} & \begin{matrix}0\\0\\0\end{matrix}\\ 0\quad 0\quad 0 & 1\end{bmatrix}} \underset{\text{translation}}{\begin{bmatrix}\mathbf{I}_{3\times 3} & -\mathbf{c}\\ 0\quad 0\quad 0 & 1\end{bmatrix}}}$$

The **K** matrix converts 3D rays in the camera's coordinate system to 2D image points in image (pixel) coordinates.

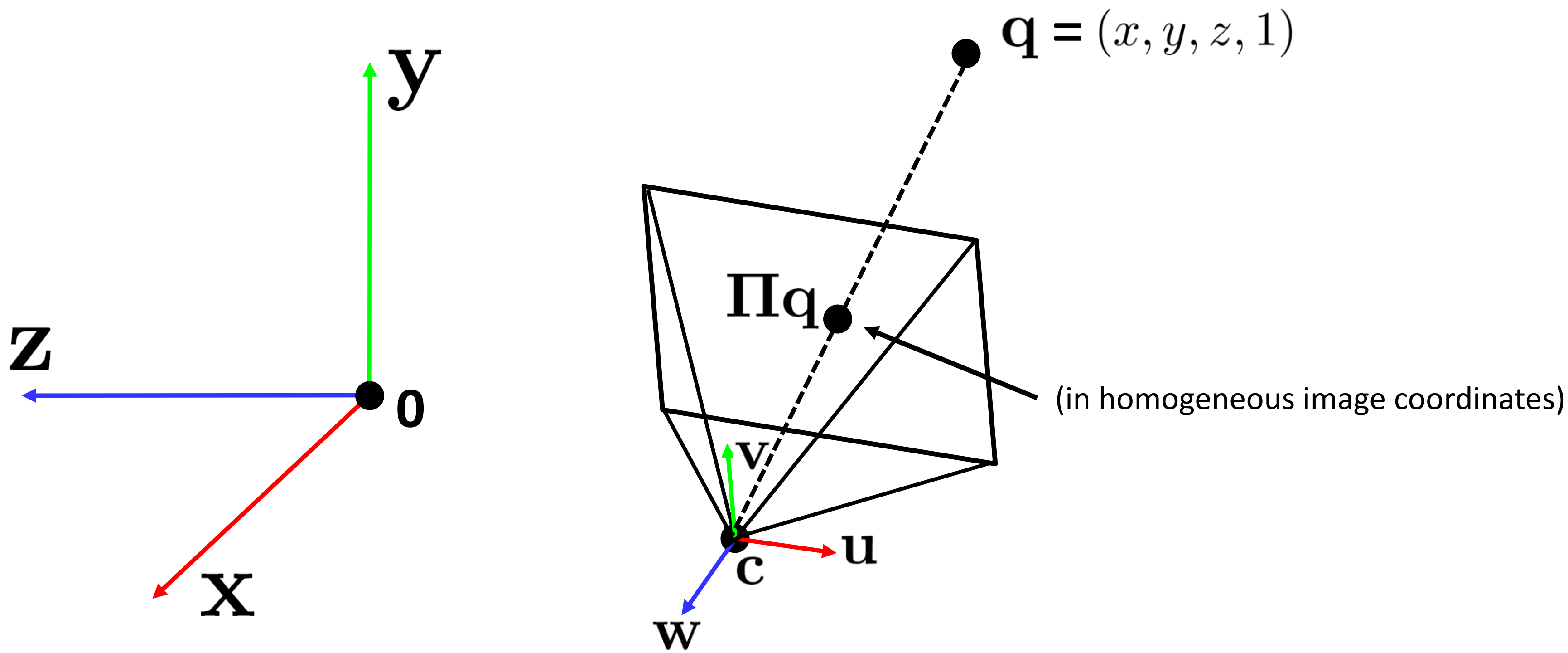
This part converts 3D points in world coordinates to 3D rays in the camera's coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

Projection matrix

$$\mathbf{\Pi} = \underset{\text{intrinsics}}{\mathbf{K}} \underbrace{\underset{\text{projection}}{\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}} \underset{\text{rotation}}{\begin{bmatrix}\mathbf{R} & \begin{matrix}0\\0\\0\end{matrix}\\ \begin{matrix}0&0&0\end{matrix} & 1\end{bmatrix}} \underset{\text{translation}}{\begin{bmatrix}\mathbf{I}_{3\times 3} & -\mathbf{c}\\ \begin{matrix}0&0&0\end{matrix} & 1\end{bmatrix}}}_{\left[\ \mathbf{R} \;\middle|\; \underbrace{-\mathbf{Rc}}_{(\mathbf{t}\text{ in book's notation})}\ \right]}$$

$$\mathbf{\Pi} = \mathbf{K}\left[\ \mathbf{R} \;\middle|\; -\mathbf{Rc}\ \right]$$

Projection matrix



Questions?

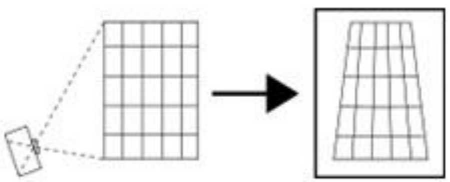
Perspective distortion

- Problem for architectural photography: converging verticals

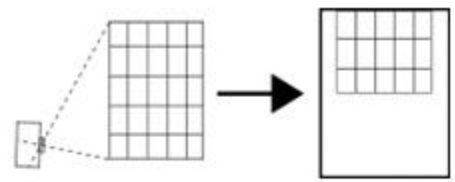
Source: F. Durand

Perspective distortion

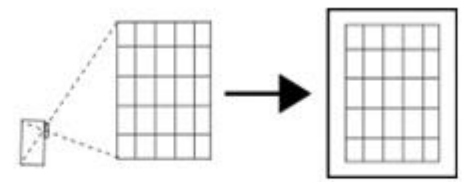
- Problem for architectural photography: converging verticals



Tilting the camera upwards results in converging verticals

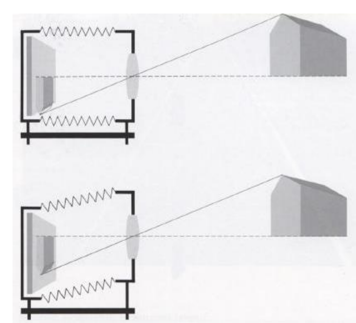


Keeping the camera level, with an ordinary lens, captures only the bottom portion of the building



Shifting the lens upwards results in a picture of the entire subject

- Solution: view camera (lens shifted w.r.t. film)

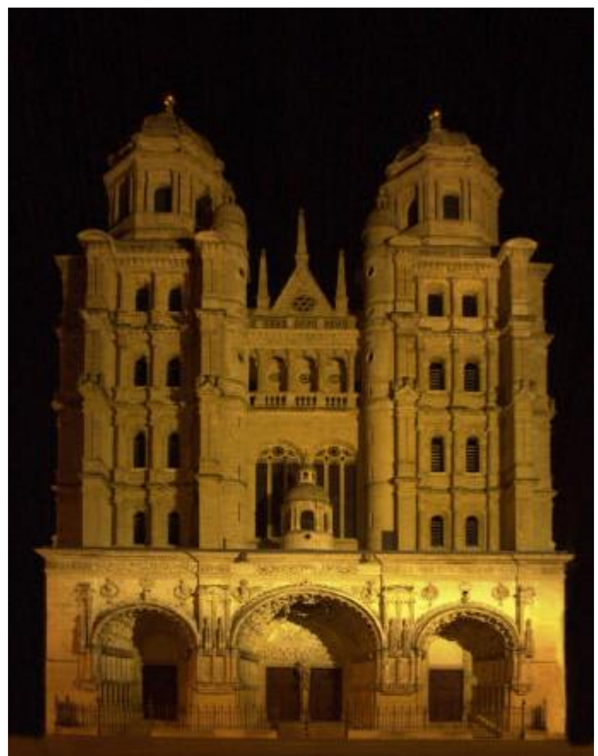


http://en.wikipedia.org/wiki/Perspective_correction_lens

Source: F. Durand

Perspective distortion

- Problem for architectural photography: converging verticals
- Result:



Source: F. Durand

Perspective distortion

- What does a sphere project to?

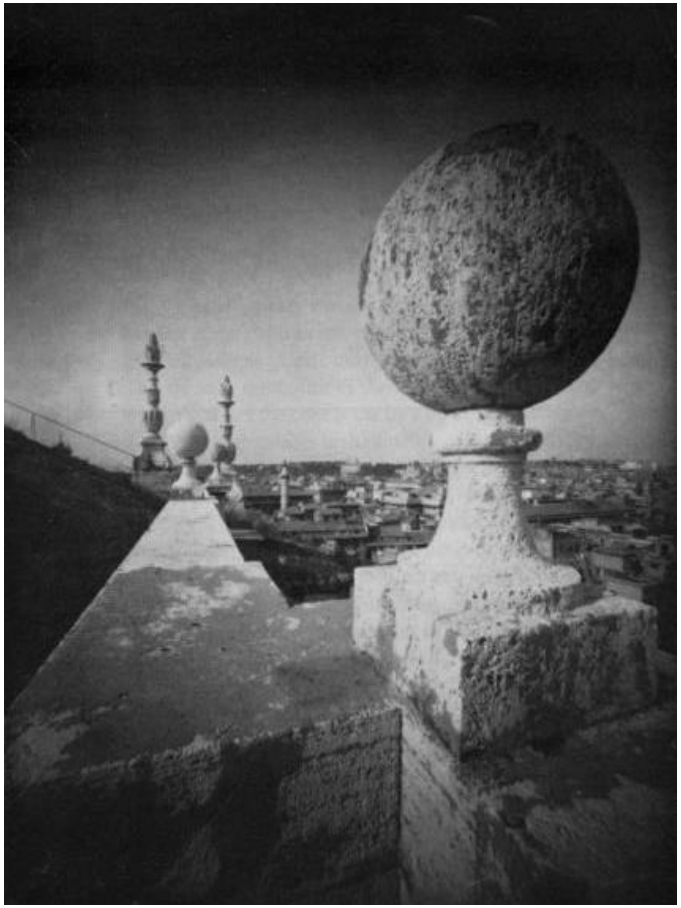
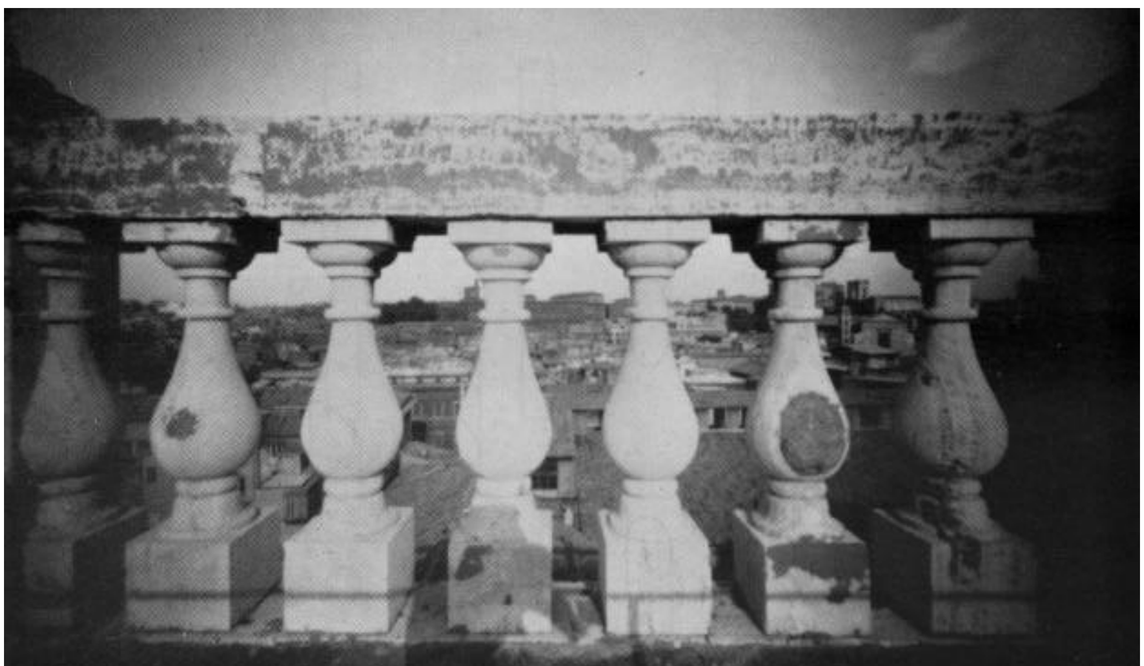
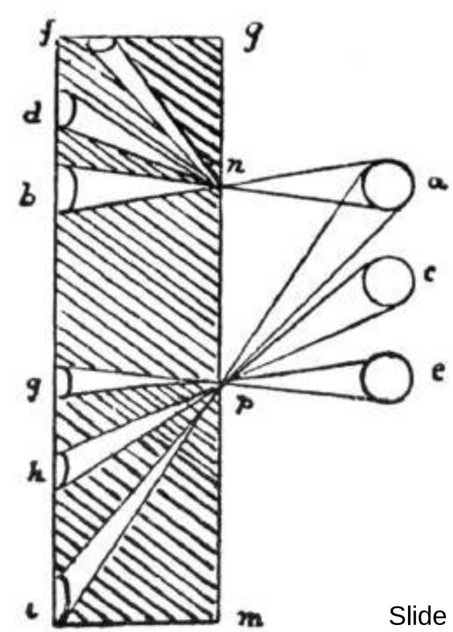


Image source: F. Durand

Perspective distortion

- The exterior columns appear bigger
- The distortion is not due to lens flaws
- Problem pointed out by Da Vinci

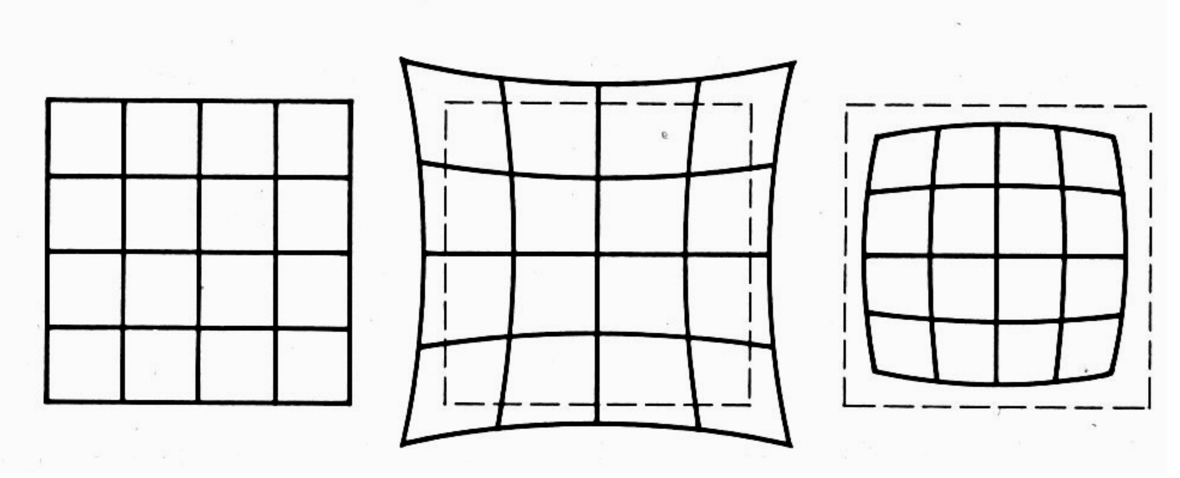




Slide by F. Durand

Perspective distortion: People

Distortion



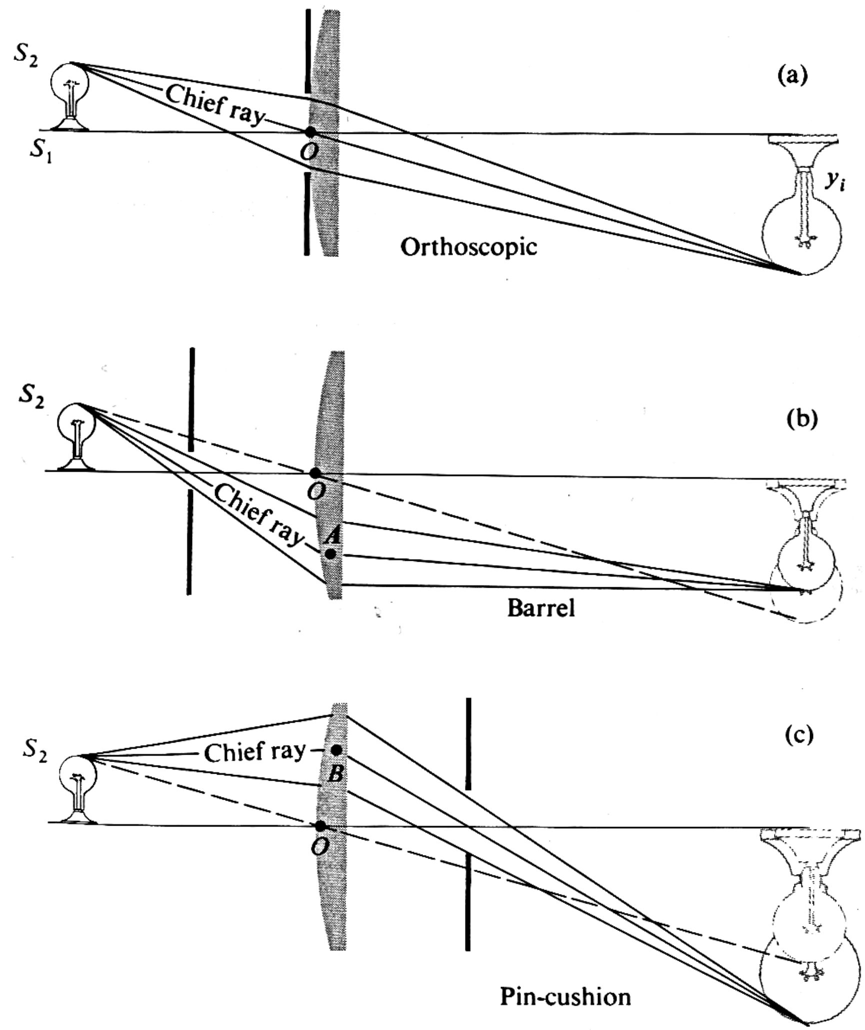
No distortion Pin cushion Barrel

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

Correcting radial distortion

from [Helmut Dersch]()

Distortion



S_2
Chief ray
S_1
O
(a)
y_i
Orthoscopic
S_2
(b)
O
Chief ray
A
Barrel
S_2
Chief ray
B
(c)
O
Pin-cushion

Modeling distortion

Project $(\widehat{x}, \widehat{y}, \widehat{z})$ to "normalized" image coordinates

$$
\begin{aligned}
x'_n &= \widehat{x}/\widehat{z} \\
y'_n &= \widehat{y}/\widehat{z}
\end{aligned}
$$

Apply radial distortion

$$
\begin{aligned}
r^2 &= {x'_n}^2 + {y'_n}^2 \\
x'_d &= x'_n(1 + \kappa_1 r^2 + \kappa_2 r^4) \\
y'_d &= y'_n(1 + \kappa_1 r^2 + \kappa_2 r^4)
\end{aligned}
$$

Apply focal length translate image center

$$
\begin{aligned}
x' &= f x'_d + x_c \\
y' &= f y'_d + y_c
\end{aligned}
$$

- To model lens distortion
  - Use above projection operation instead of standard projection matrix multiplication

Other types of projection

- Lots of intriguing variants…
- (I’ll just mention a few fun ones)

360 degree field of view…

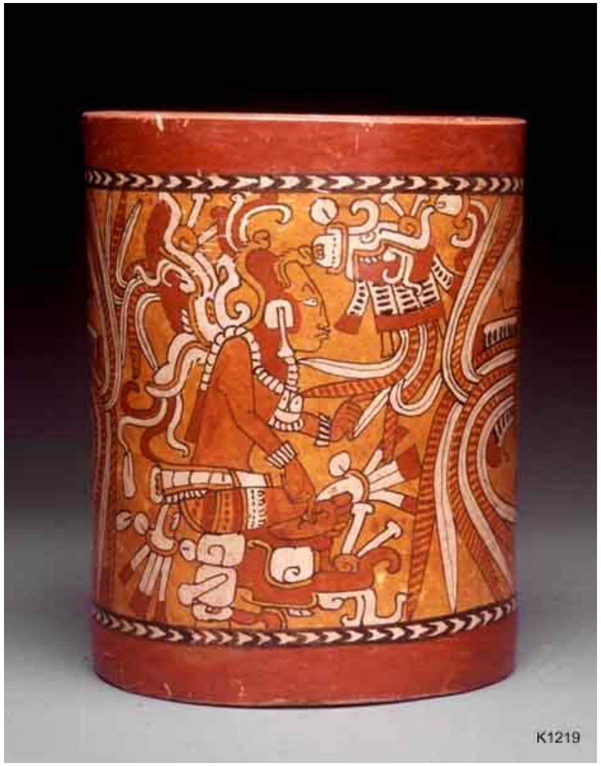
- Basic approach
  - Take a photo of a parabolic mirror with an orthographic lens (Nayar)
  - Or buy one a lens from a variety of omnicam manufacturers…
    - See http://www.cis.upenn.edu/~kostas/omni.html

Tilt-shift

http://www.northlight-images.co.uk/article_pages/tilt_and_shift_ts-e.html

Titlt-shift images from Olivo Barbieri
and Photoshop imitations

Rotating sensor (or object)



Rollout Photographs © Justin Kerr
http://research.famsi.org/kerrmaya.html

Also known as “cyclographs”, “peripheral images”