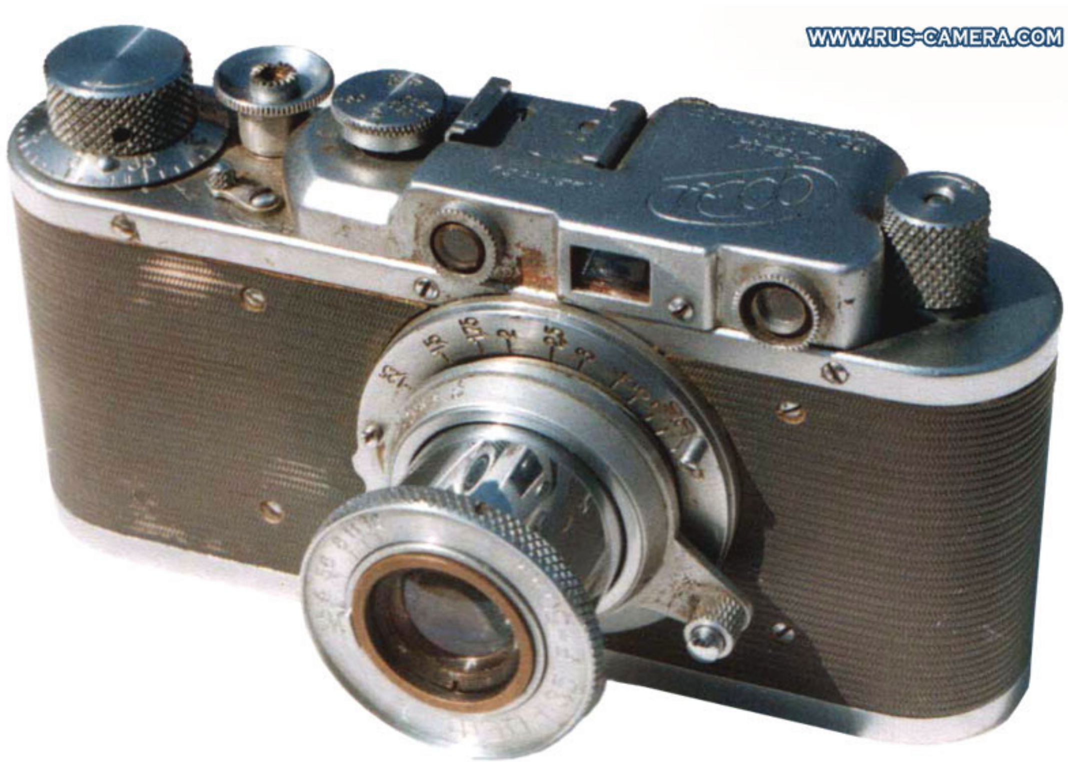
CS4670 / 5670: Computer Vision

KavitaBala

Lecture 18: Cameras



Source: S. Lazebnik

Announcements

- Prelim next Thu
  - Everything till Monday

Where are we?

- Imaging: pixels, features, …
- Scenes: geometry, material, lighting
- Recognition: people, objects, …

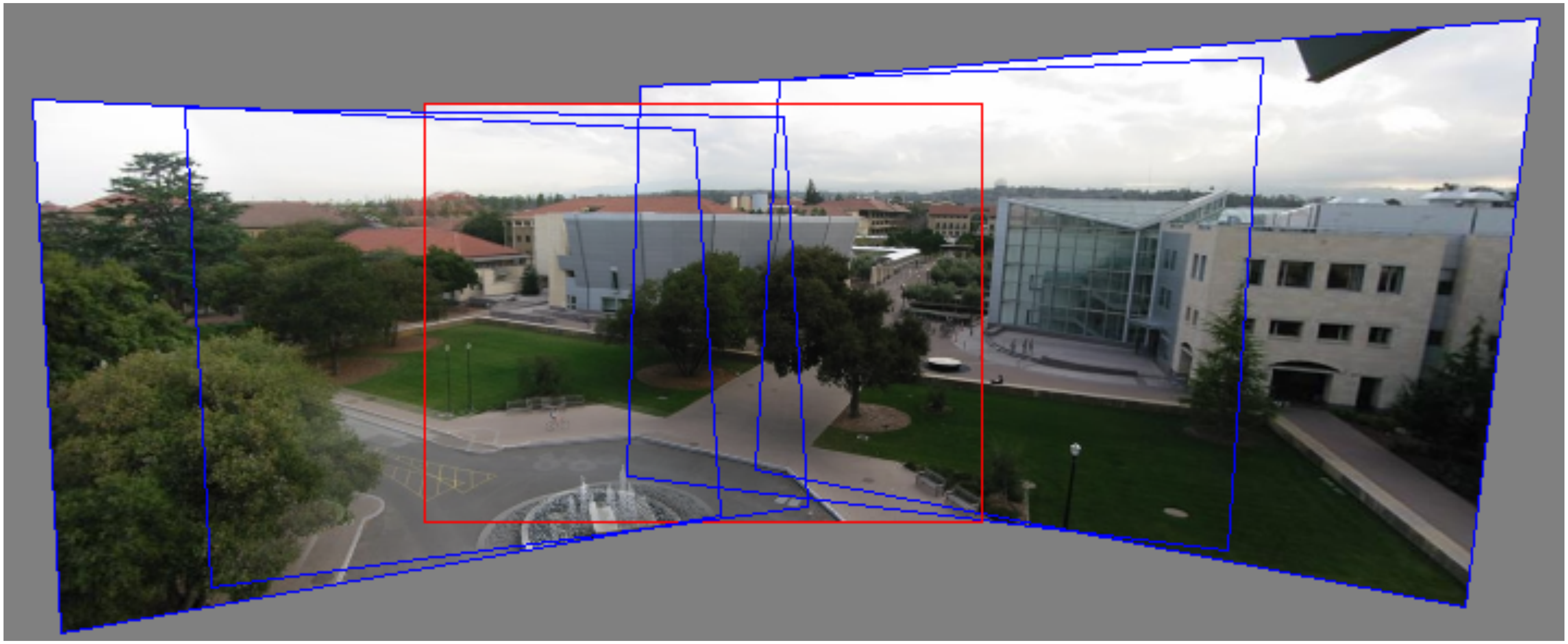
Reading

- Szeliski 2.1.3-2.1.6

Panoramas

- Now we know how to create panoramas!
- Given two images:
  - Step 1: Detect features
  - Step 2: Match features
  - Step 3: Compute a homography using RANSAC
  - Step 4: Combine the images together (somehow)
- What if we have more than two images?

Can we use homographies to create a 360 panorama?

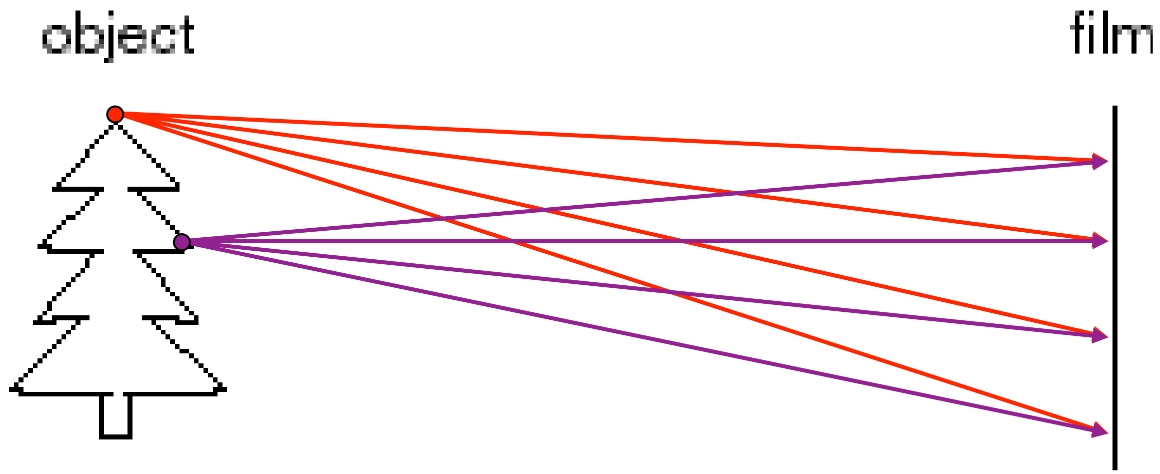


- To figure this out, we need to learn what a **camera** is

360 panorama

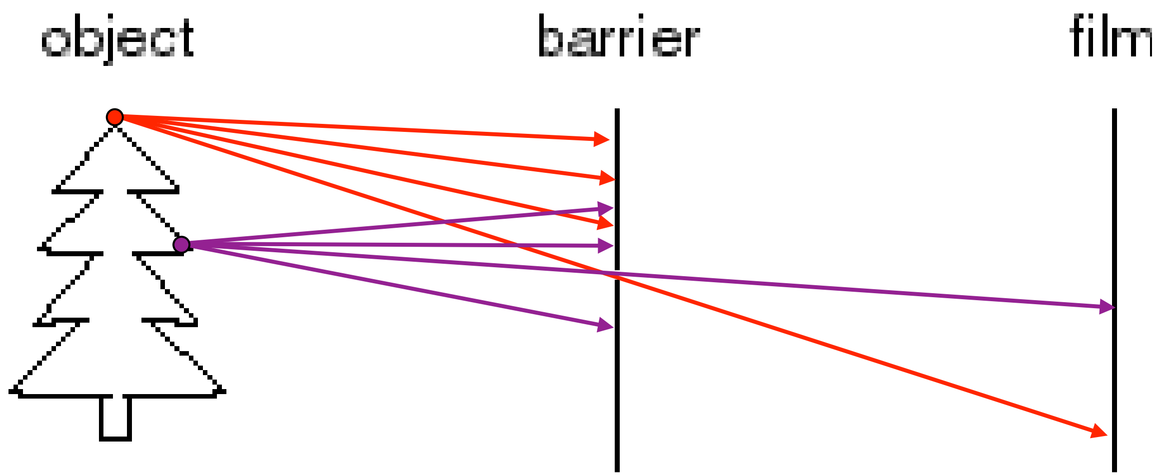

WWW.RUS-CAMERA.COM

Image formation



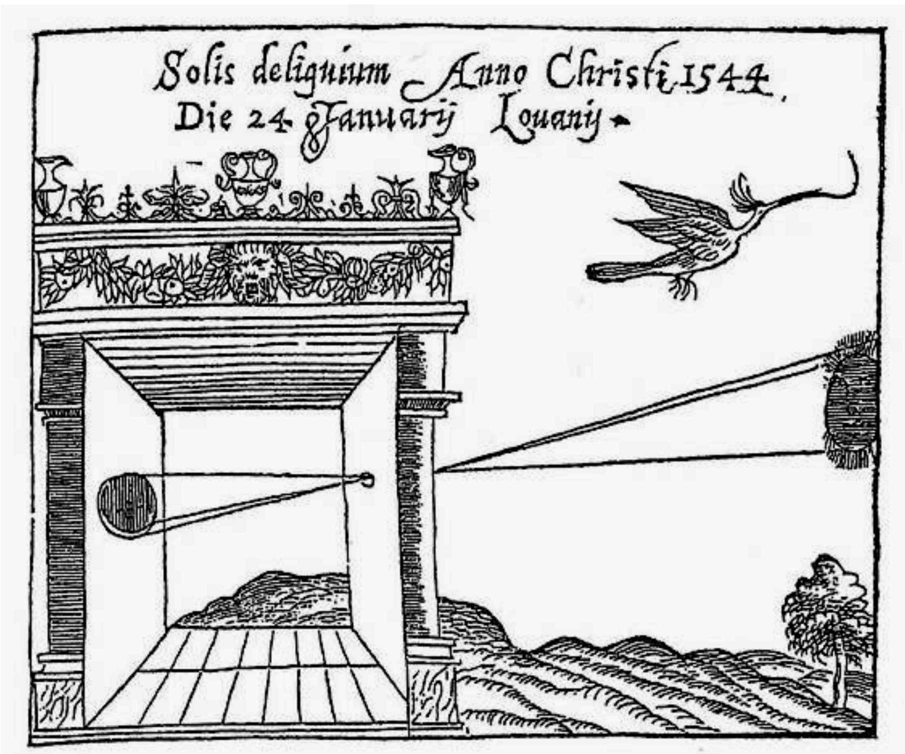
- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

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

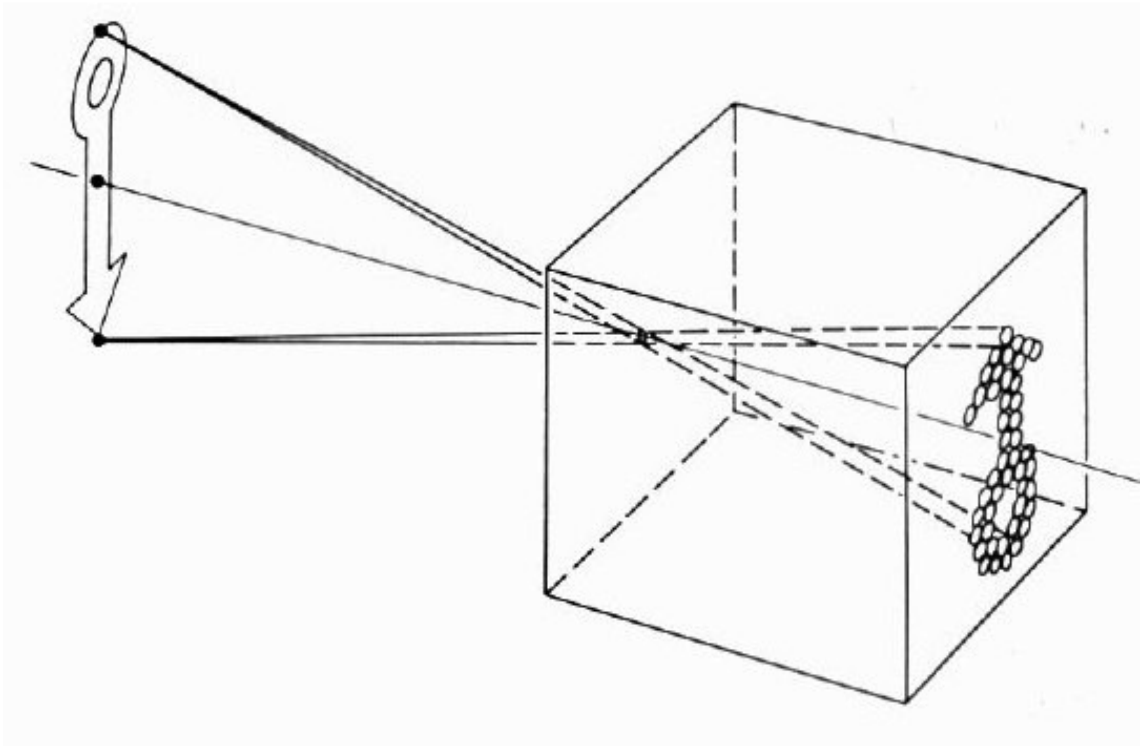


Gemma Frisius, 1558

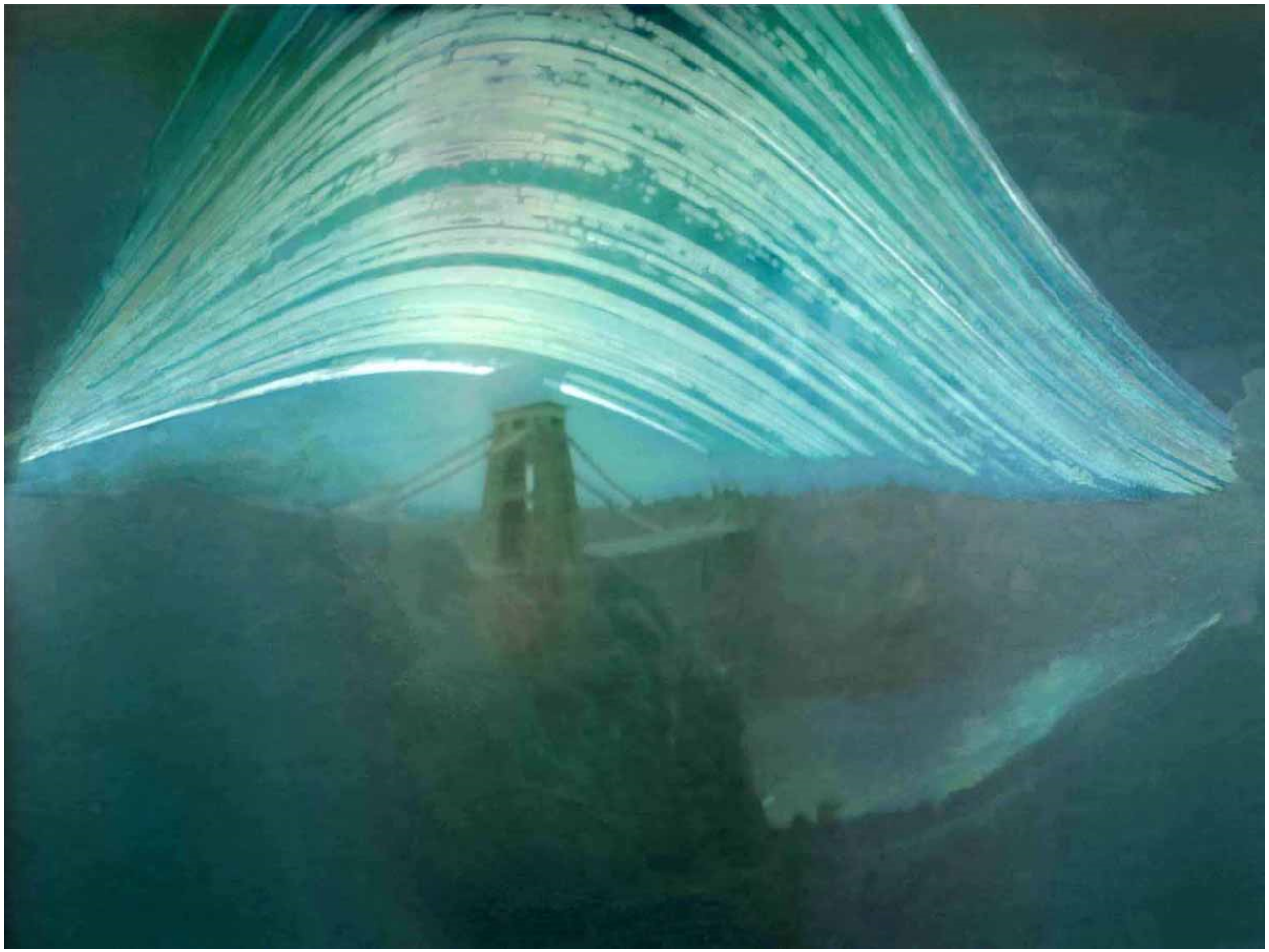
- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

Source: A. Efros

Camera Obscura

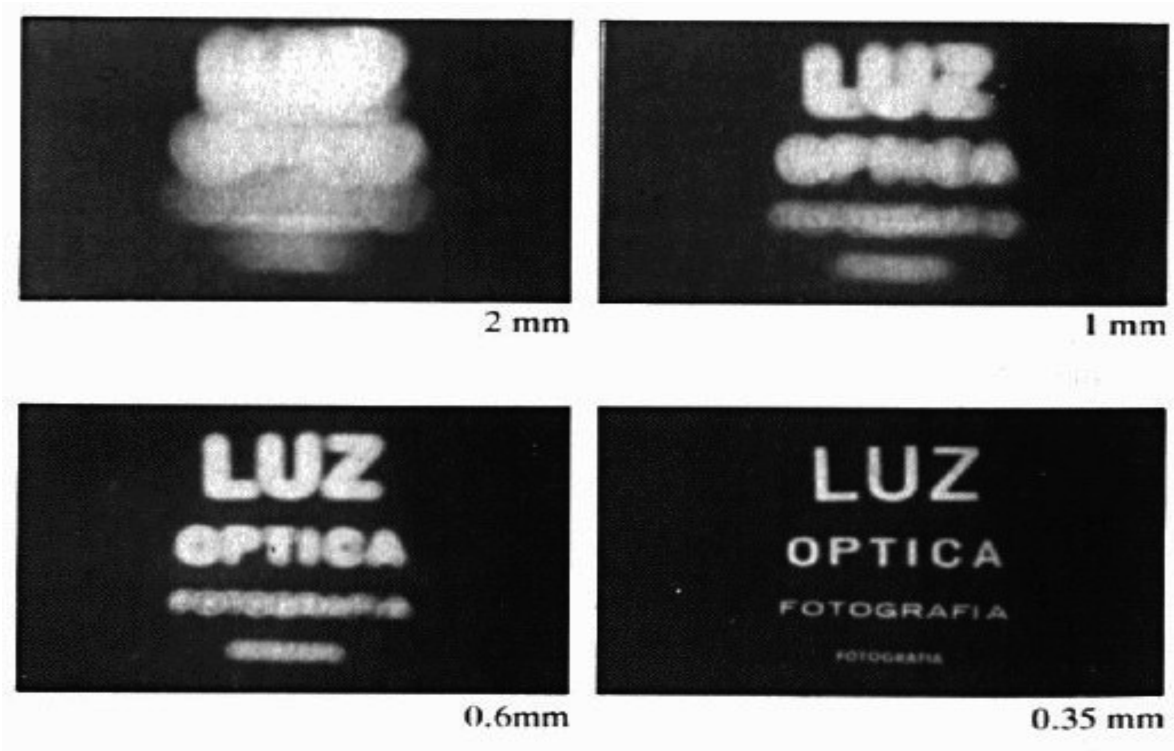


Pinhole photography



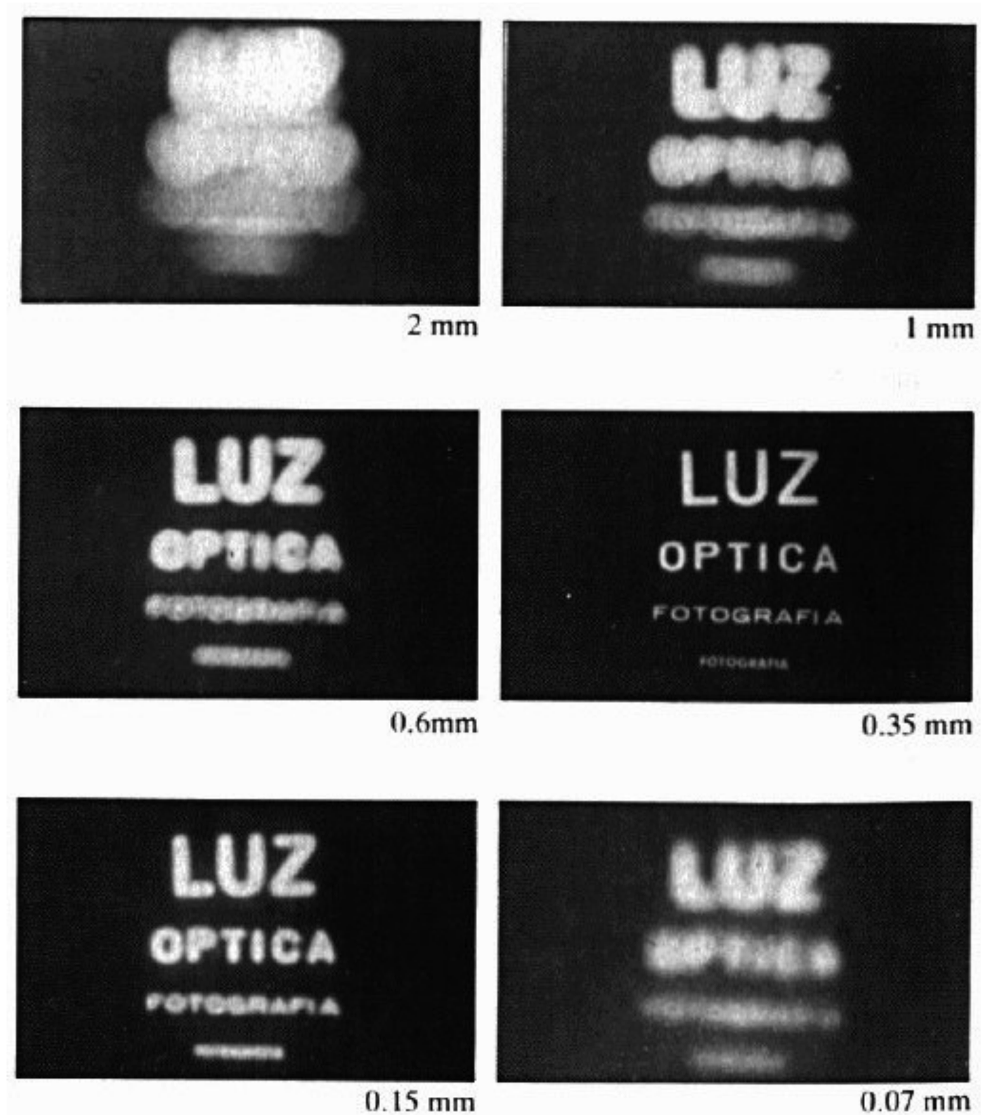
Justin Quinnell, The Clifton Suspension Bridge. December 17th 2007 - June 21st 2008
***6-month* exposure**

Shrinking the aperture

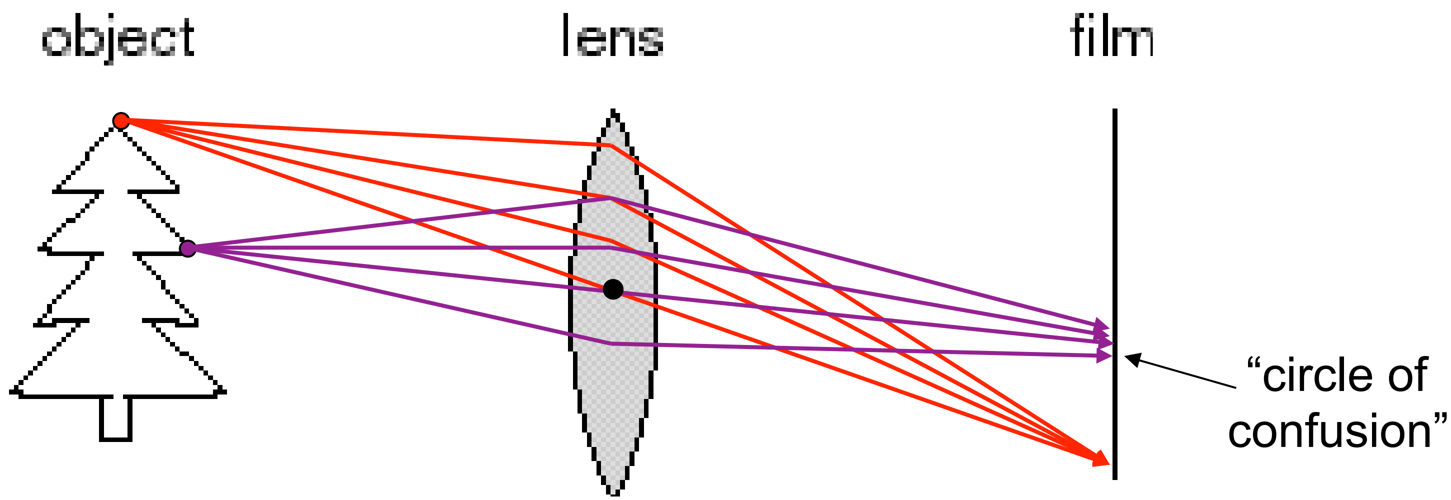


- Why not make the aperture as small as possible?
  - Less light gets through
  - *Diffraction* effects...

Shrinking the aperture

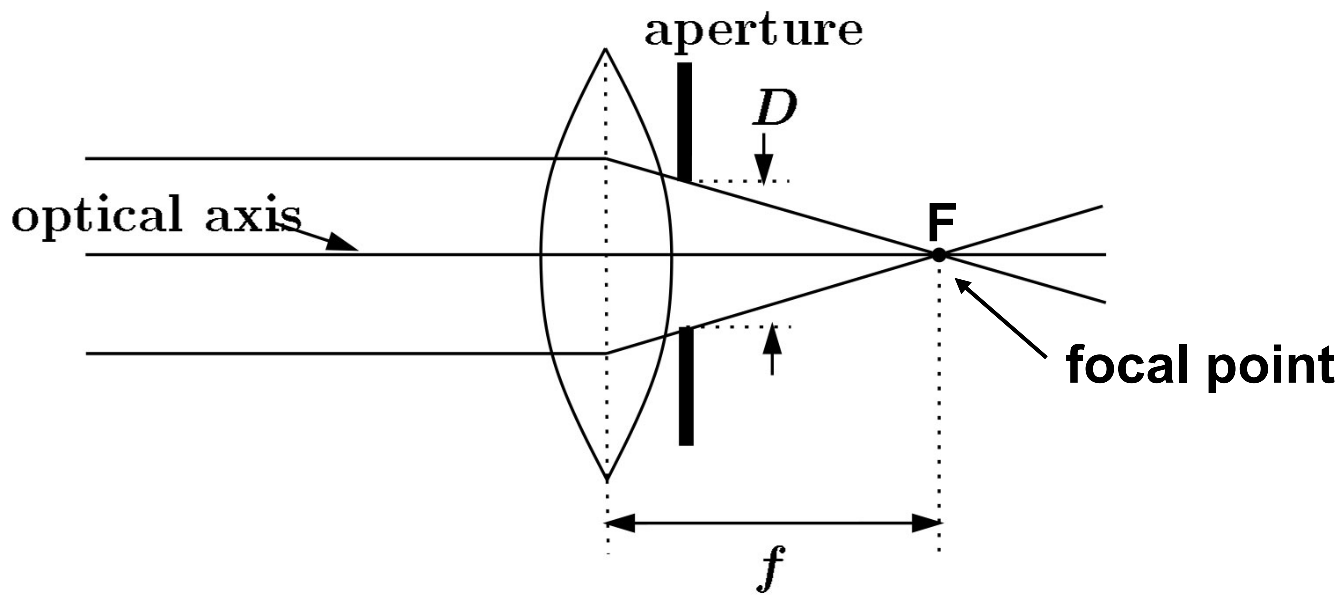


Adding a lens



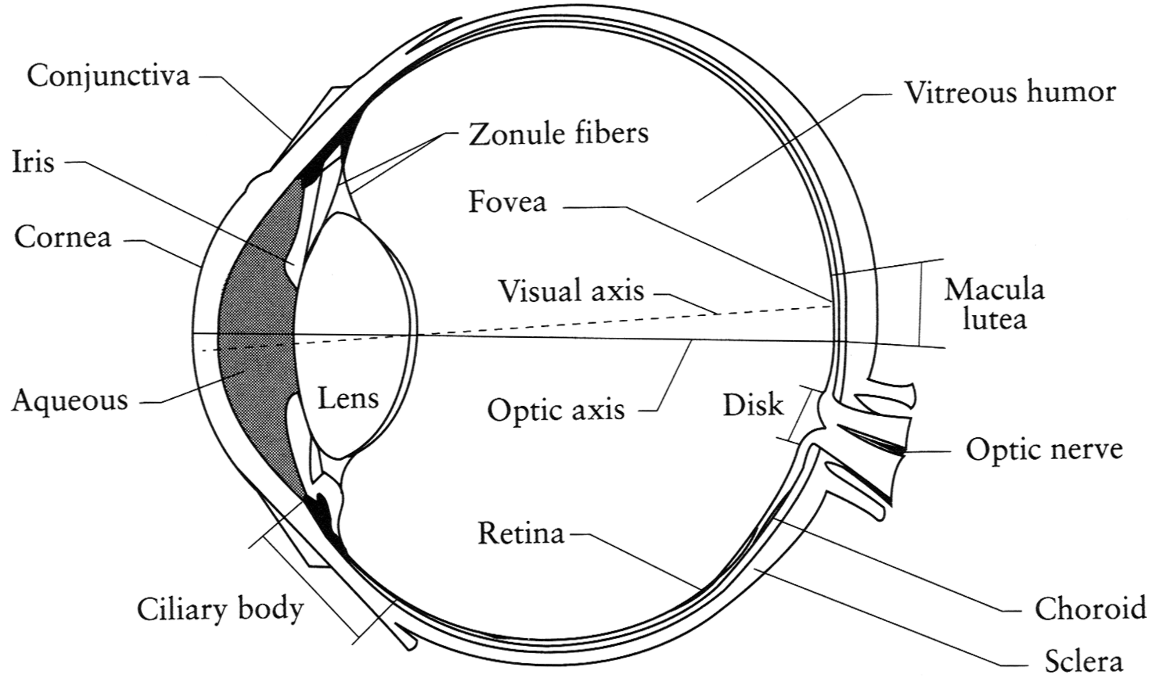
- A lens focuses light onto the film
  - There is a specific distance at which objects are "in focus"
    - other points project to a "circle of confusion" in the image
  - Changing the shape of the lens changes this distance

Lenses



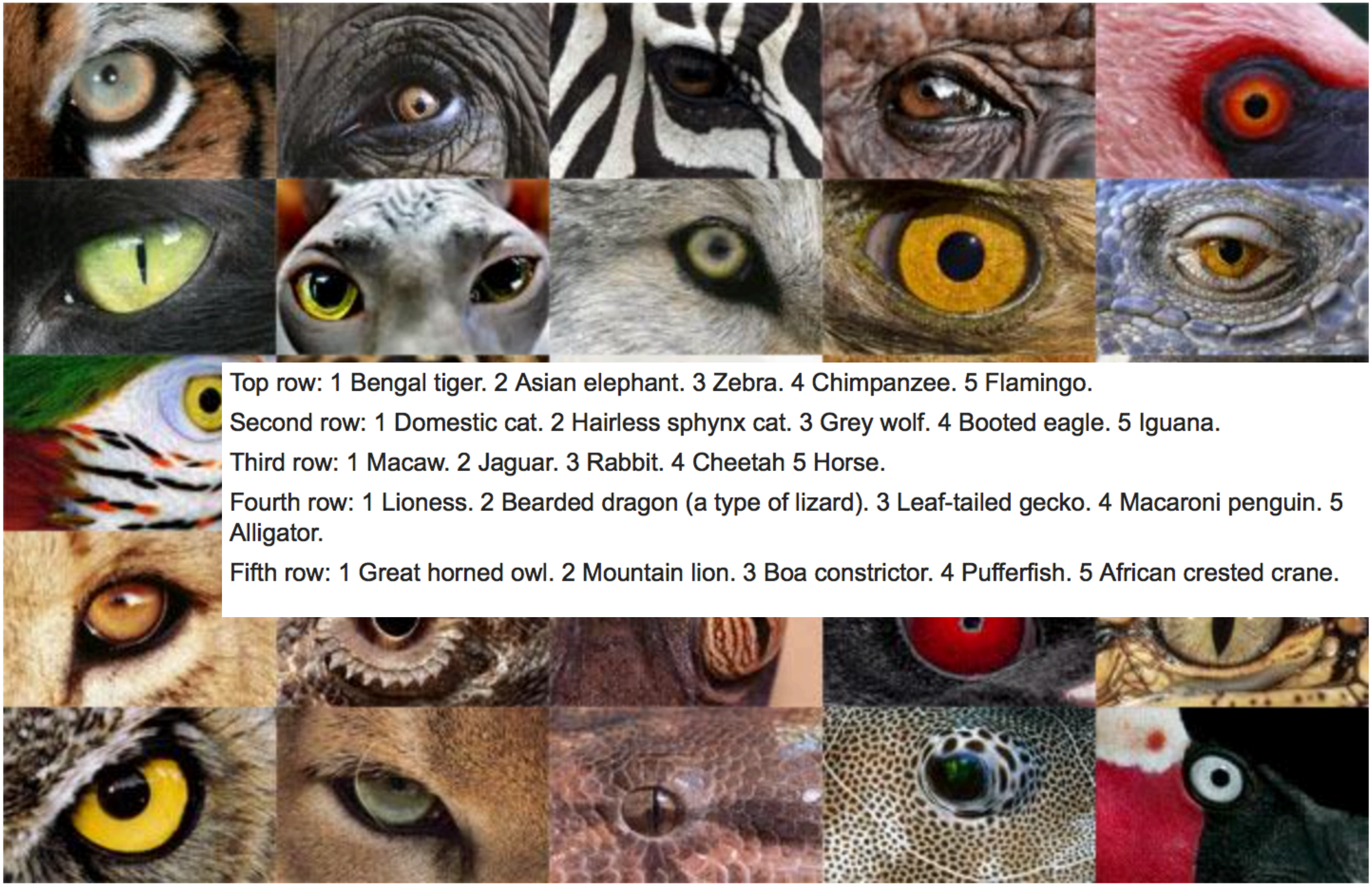
- A lens focuses parallel rays onto a single focal point
  - focal point at a distance *f* beyond the plane of the lens (the *focal length*)
    - *f* is a function of the shape and index of refraction of the lens
  - Aperture restricts the range of rays
    - aperture may be on either side of the lens
  - Lenses are typically spherical (easier to produce)

The eye



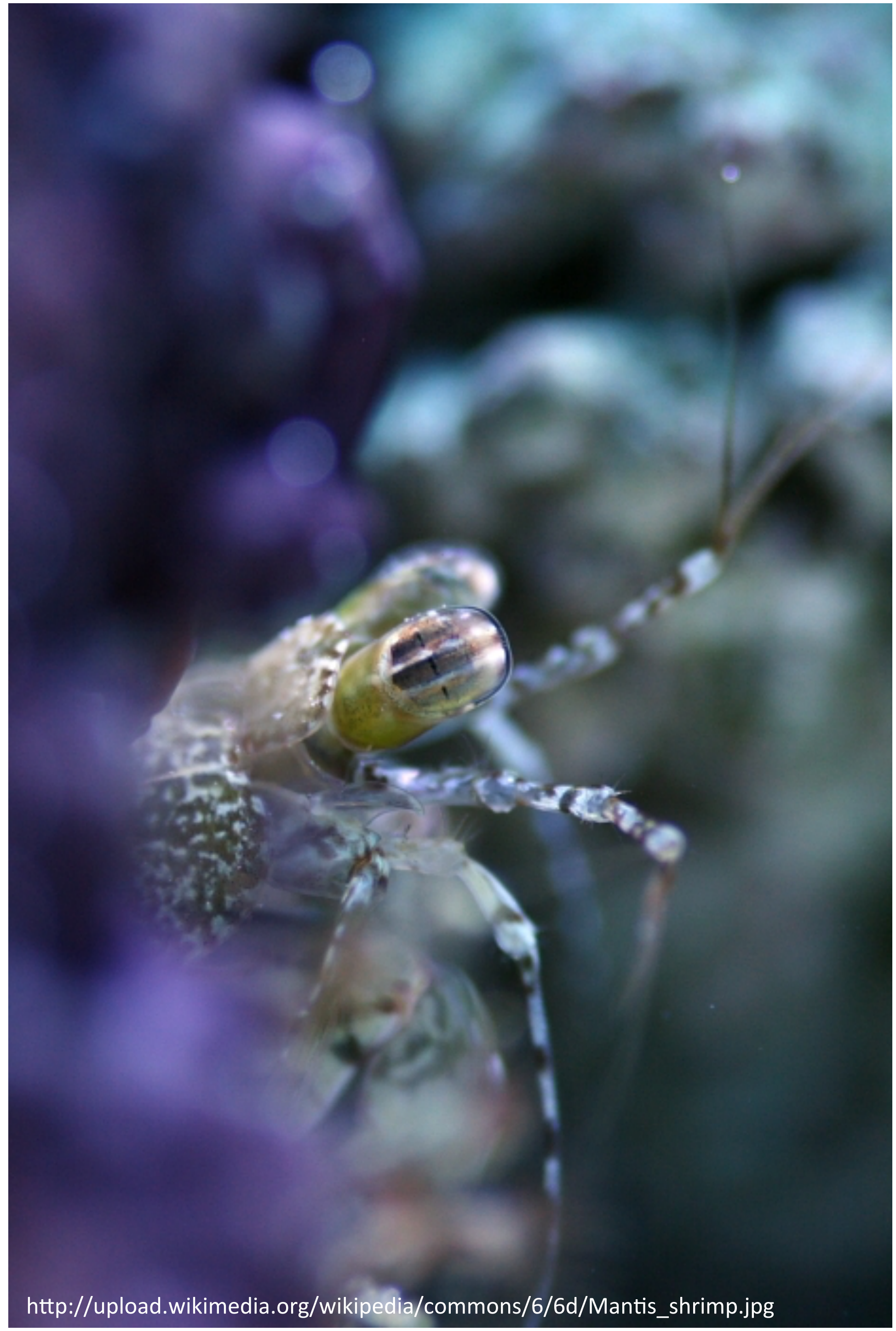
- The human eye is a camera
  - **Iris** - colored annulus with radial muscles
  - **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What's the "film"?
    - photoreceptor cells (rods and cones) in the **retina**

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

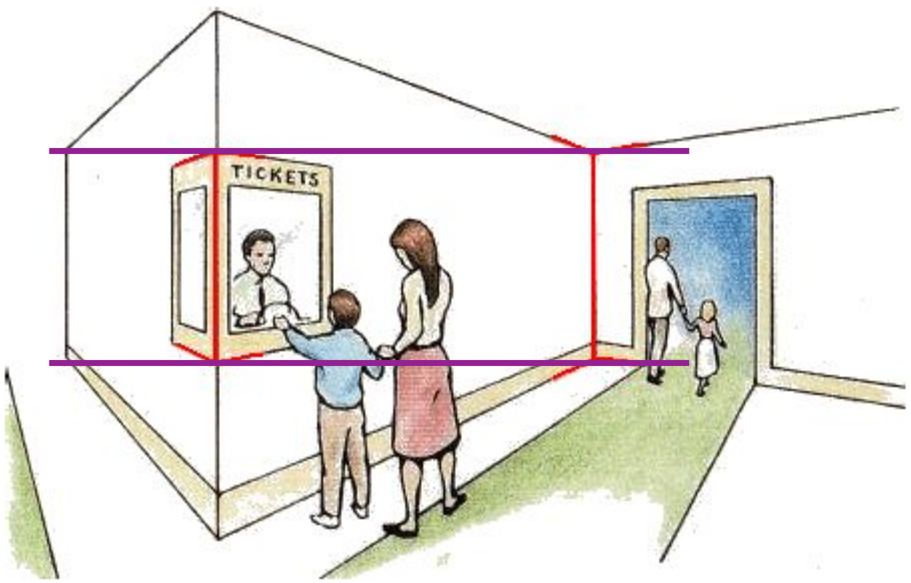


Projection


MAKE POVERTY HISTORY
JULIAN BEEVER
CoolOpticalIllusions.com

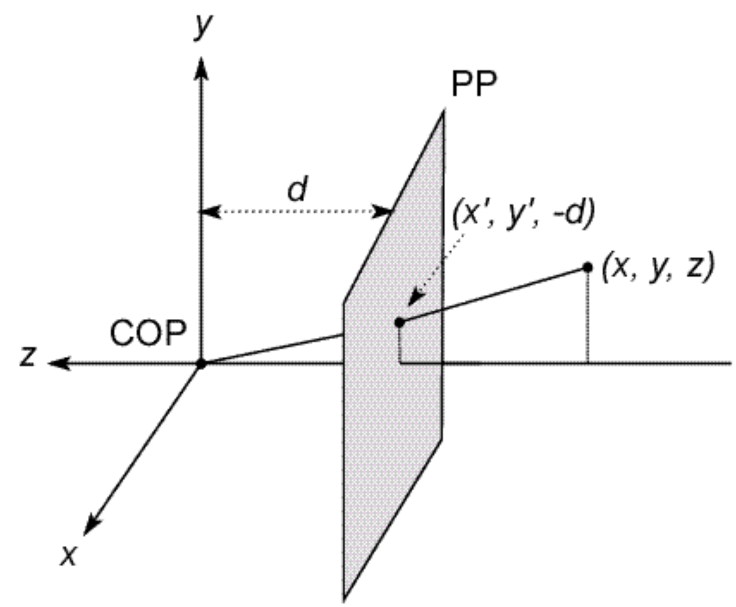
Projection

Müller-Lyer Illusion



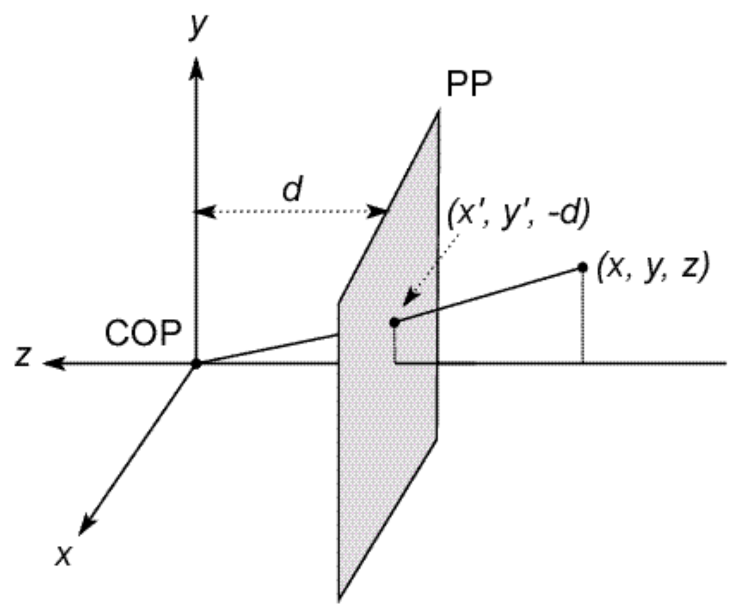
http://www.michaelbach.de/ot/sze_muelue/index.html

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 need 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)

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

- The screen-space or image-plane projection is therefore:

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

Modeling projection

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

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

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

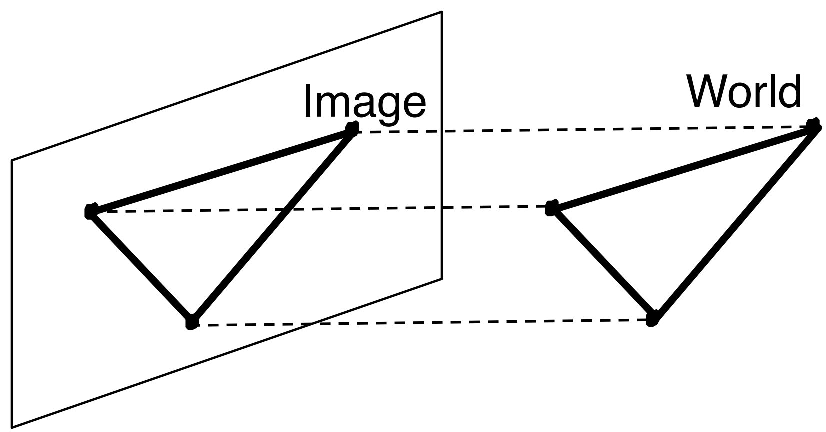
- How does scaling the projection matrix change the transformation?

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

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

Orthographic projection

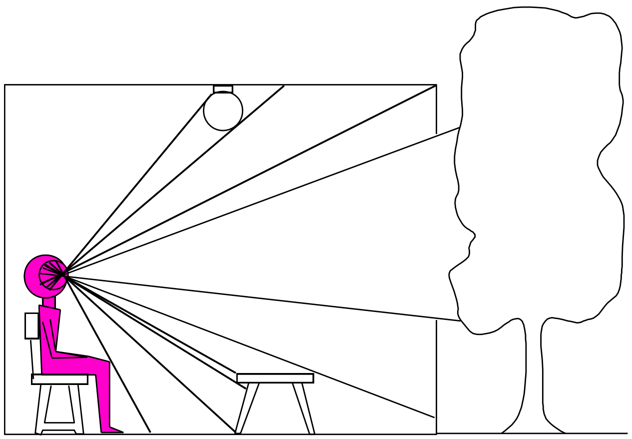
- Special case of perspective projection
  - Distance from the COP to the PP is infinite



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

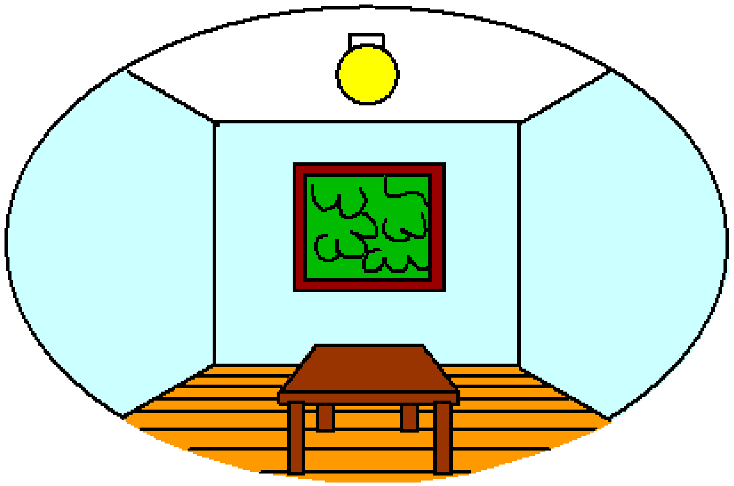
Dimensionality Reduction Machine (3D to 2D)

3D world



Point of observation

2D image



What have we lost?

- Angles
- Distances (lengths)

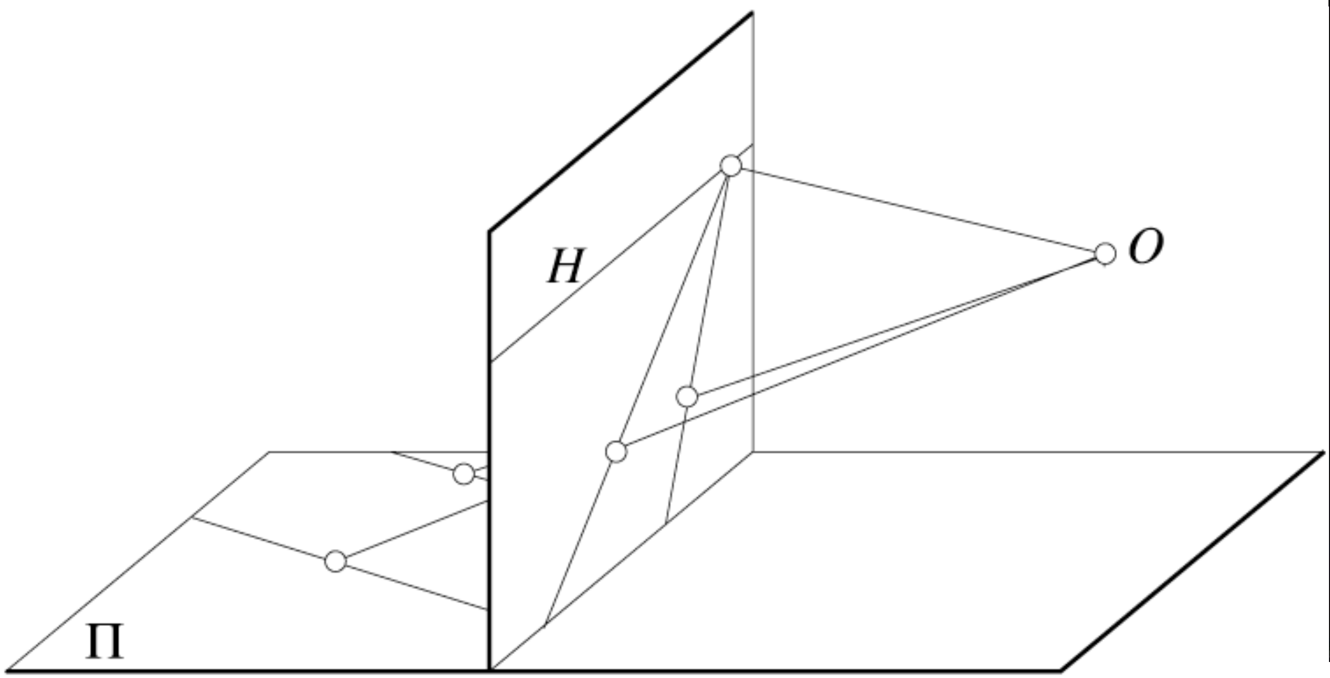
Slide by A. Efros

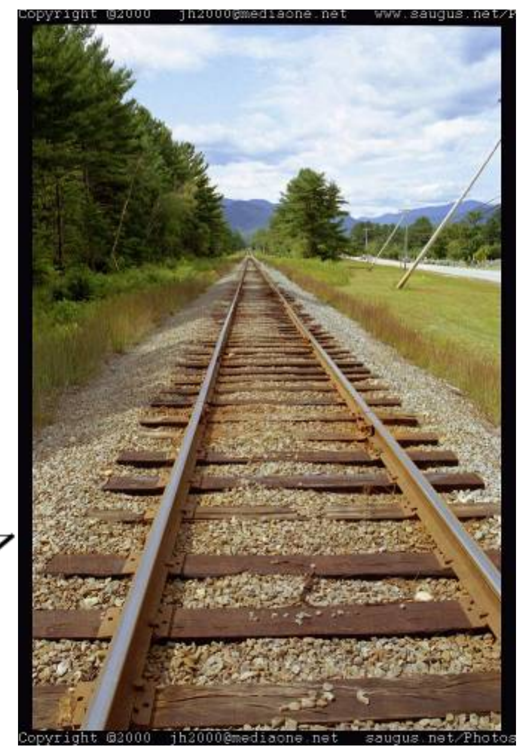
Projection properties

- 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

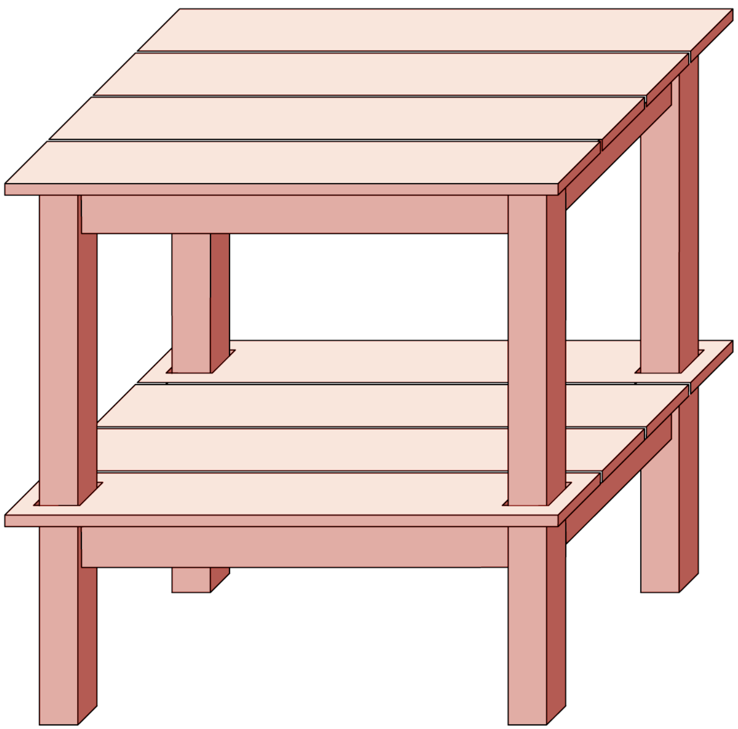
Projection properties

- 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

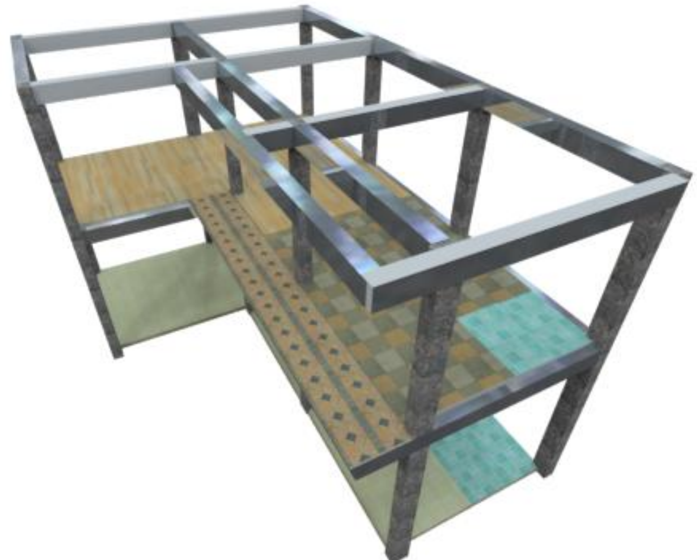




Orthographic projection



Perspective projection

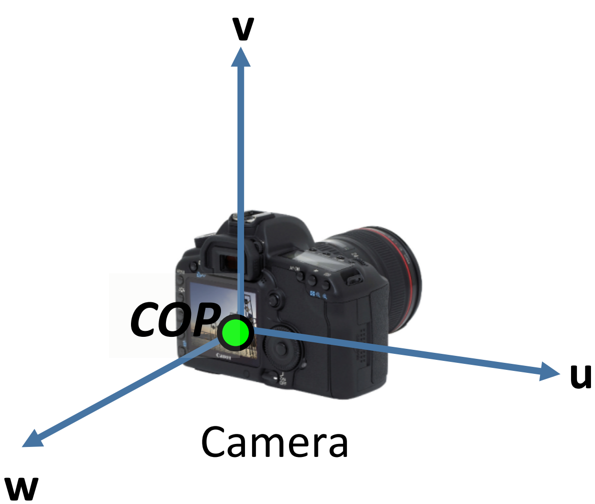


Camera parameters

- How many numbers do we need to describe a camera?

- We need to describe its *pose* in the world
- We need to describe its internal parameters

A Tale of Two Coordinate Systems



Camera

"The World"

Two important coordinate systems:
1. *World* coordinate system
2. *Camera* coordinate system

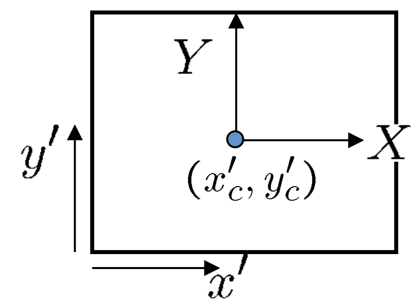
Camera parameters

- 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
  - Need to know camera *intrinsics*
- These can all be described with matrices

Camera parameters

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

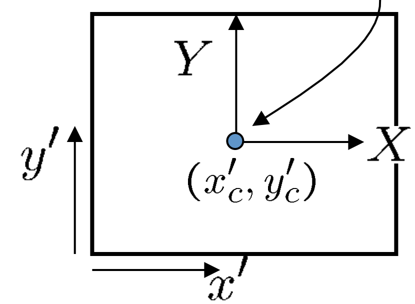
Camera parameters

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

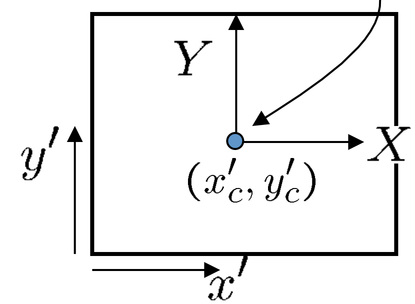
Camera parameters

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

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



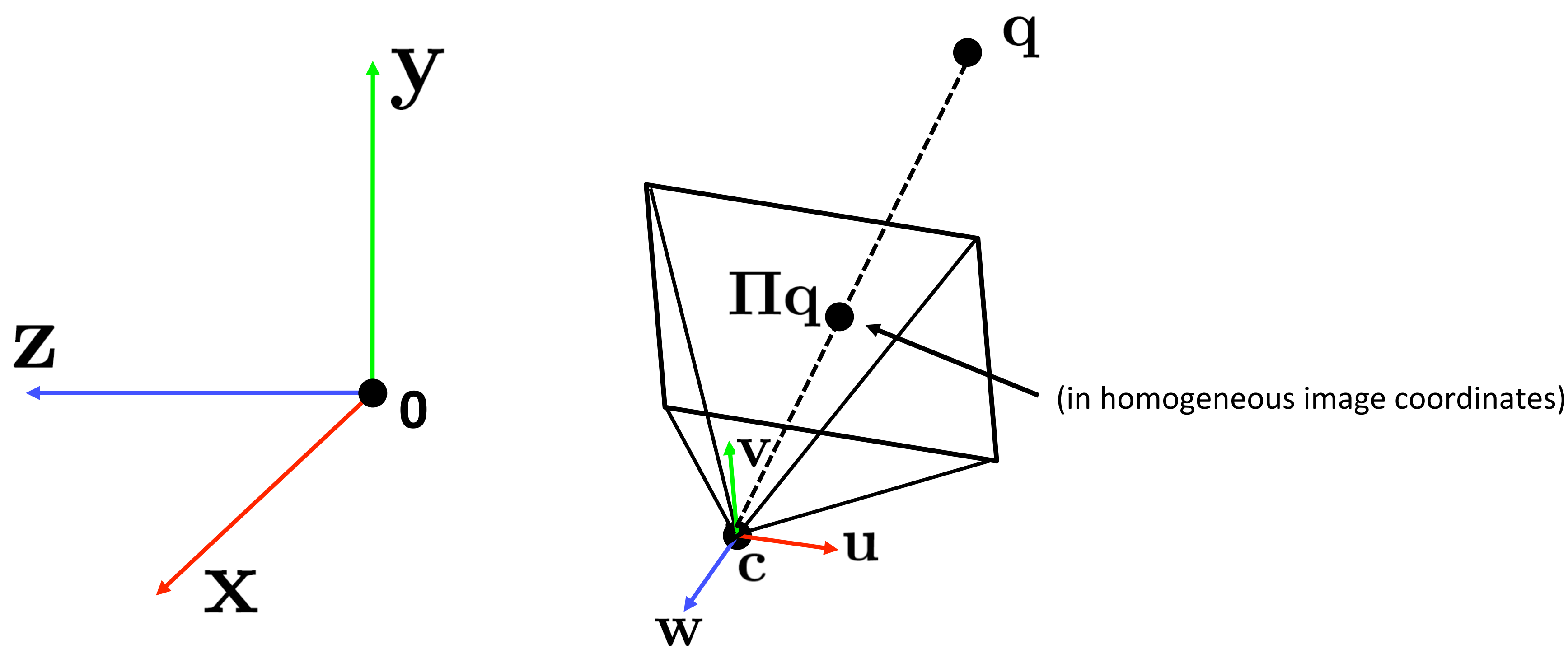
- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

identity matrix

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

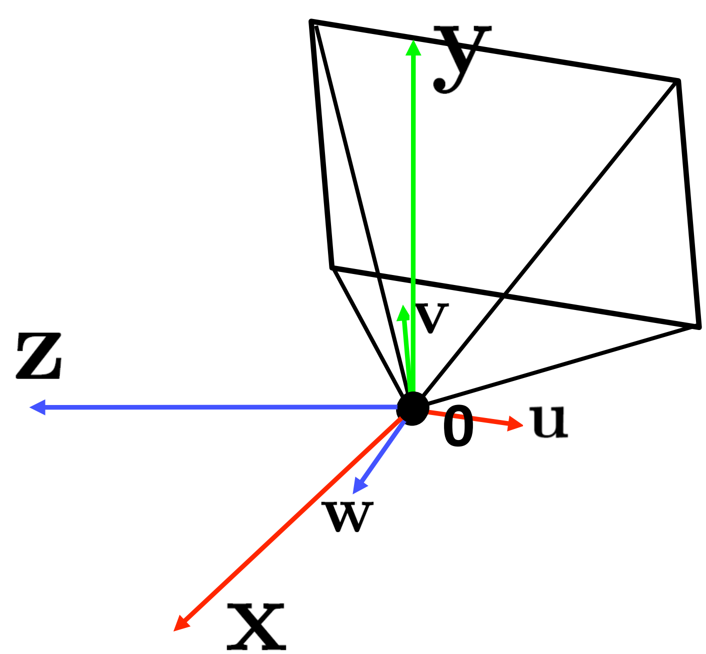
- The definitions of these parameters are **not** completely standardized
  - especially intrinsics—varies from one book to another

Projection matrix



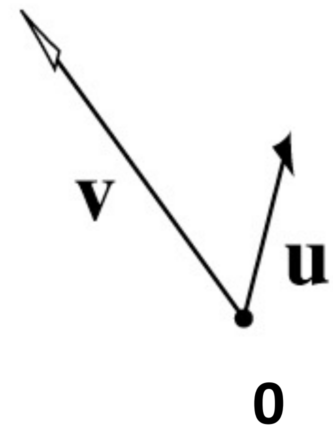
Extrinsics

- 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)



Affine change of coordinates

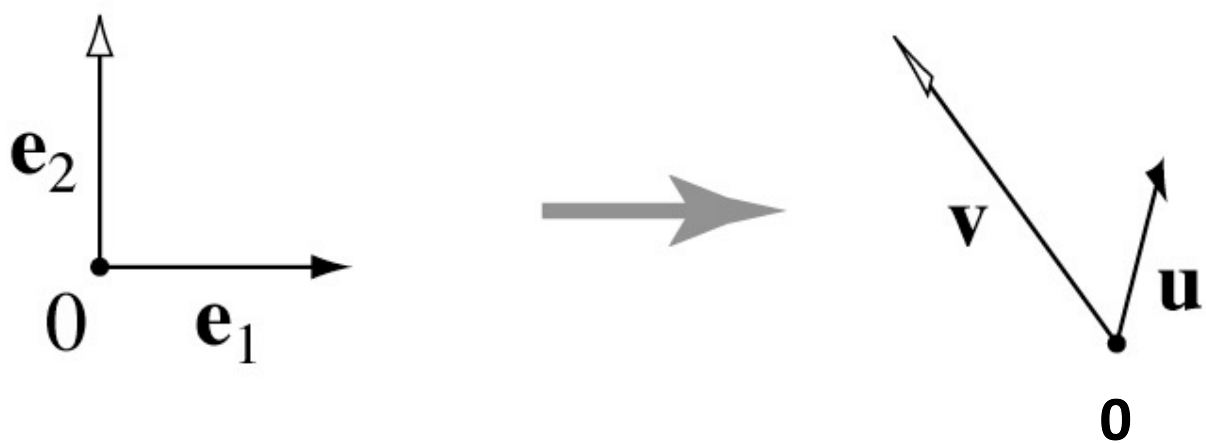
- Coordinate frame: point plus basis
- Need to change representation of point from one basis to another
- Canonical: origin (0,0) w/ axes e1, e2



Recap from 4620

Another way of thinking about this

- Change of coordinates



Recap from 4620

Coordinate frame summary

- Frame = point plus basis
- Frame matrix (frame-to-canonical) is

$$F = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix}$$

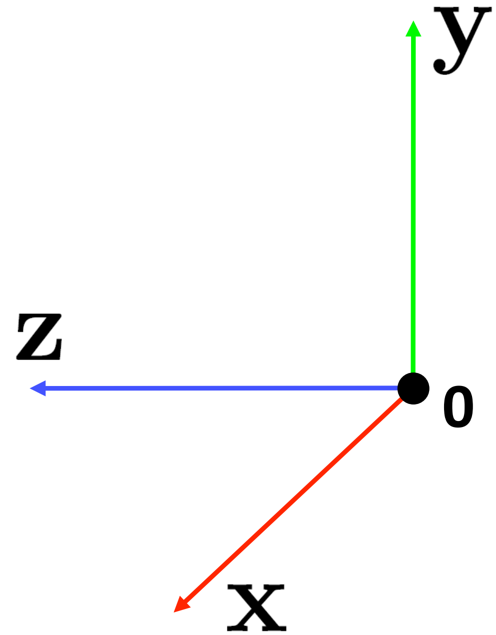
- Move points to and from frame by multiplying with *F*

$$p_e = F p_F \quad p_F = F^{-1} p_e$$

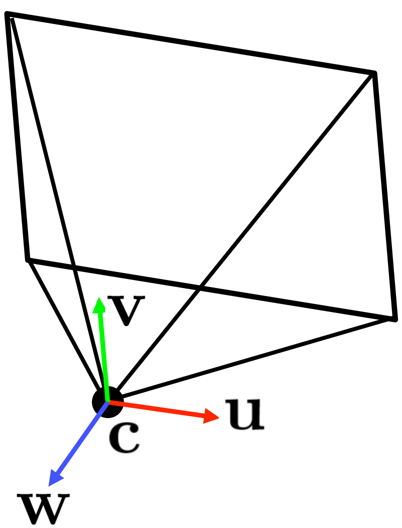
Recap from 4620

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)

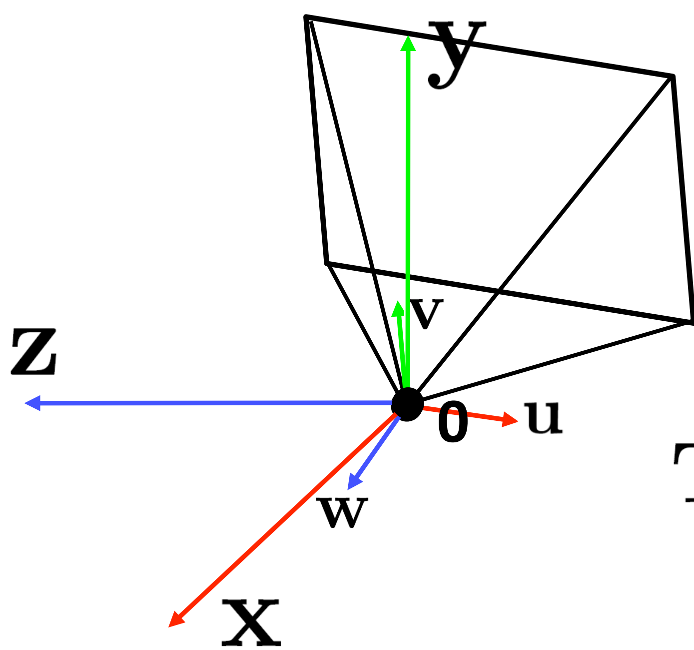


Step 1: Translate by -**c**



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)

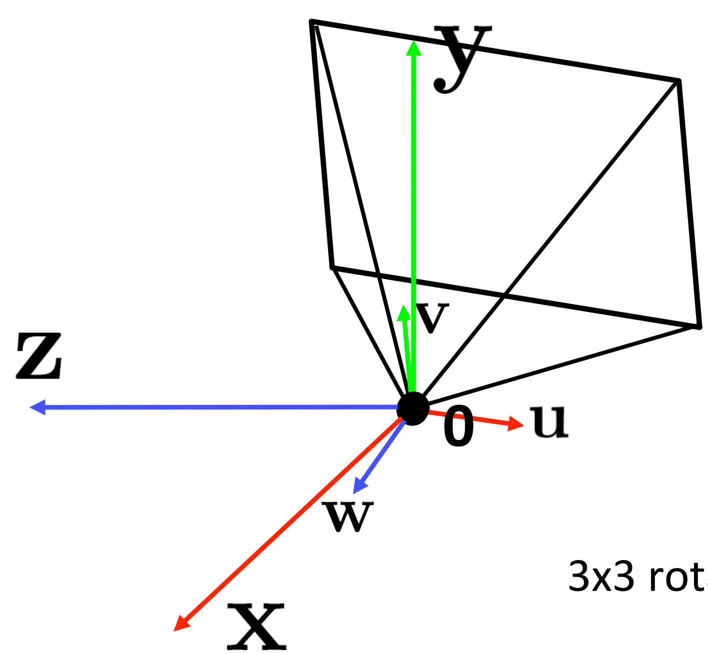


Step 1: Translate by -**c**

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{3\times3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

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

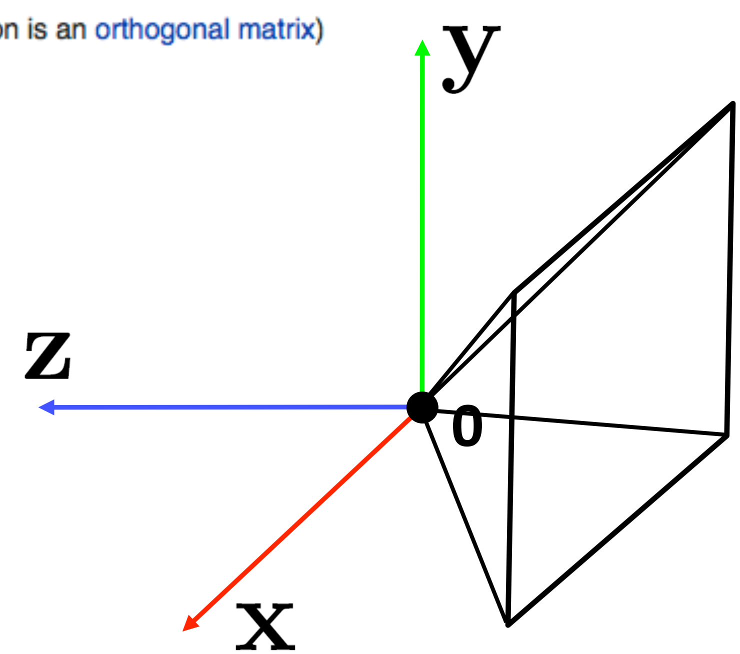
- How do we get the camera to "canonical form"?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

For any rotation matrix R acting on $\mathbb{R}^n$,

- $R^T = R^{-1}$ (The rotation is an orthogonal matrix)

Step 1: Translate by -**c**

Step 2: Rotate by **R**



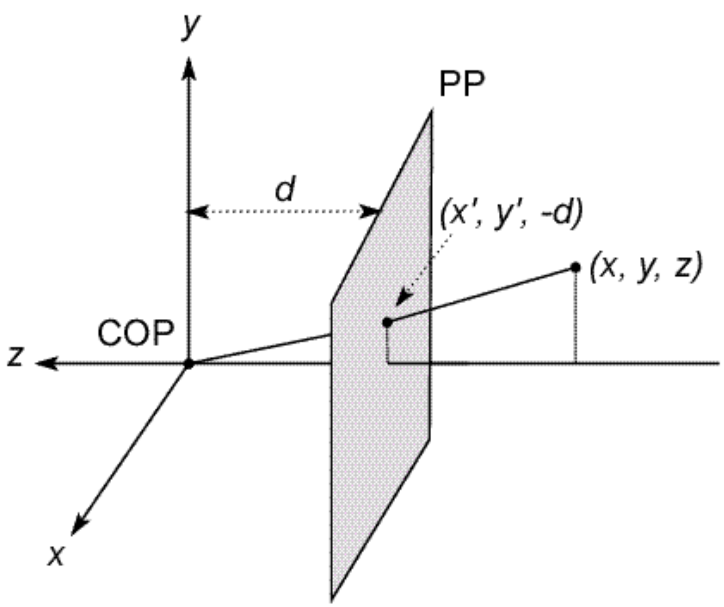
$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

Perspective projection

$$\underbrace{\begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(intrinsics)

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



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

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)

Projection matrix

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

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