Projective Geometry and Camera Models

Computer Vision
CS 543 / ECE 549
University of Illinois

Derek Hoiem

Note about HW1

- Out before next Tues

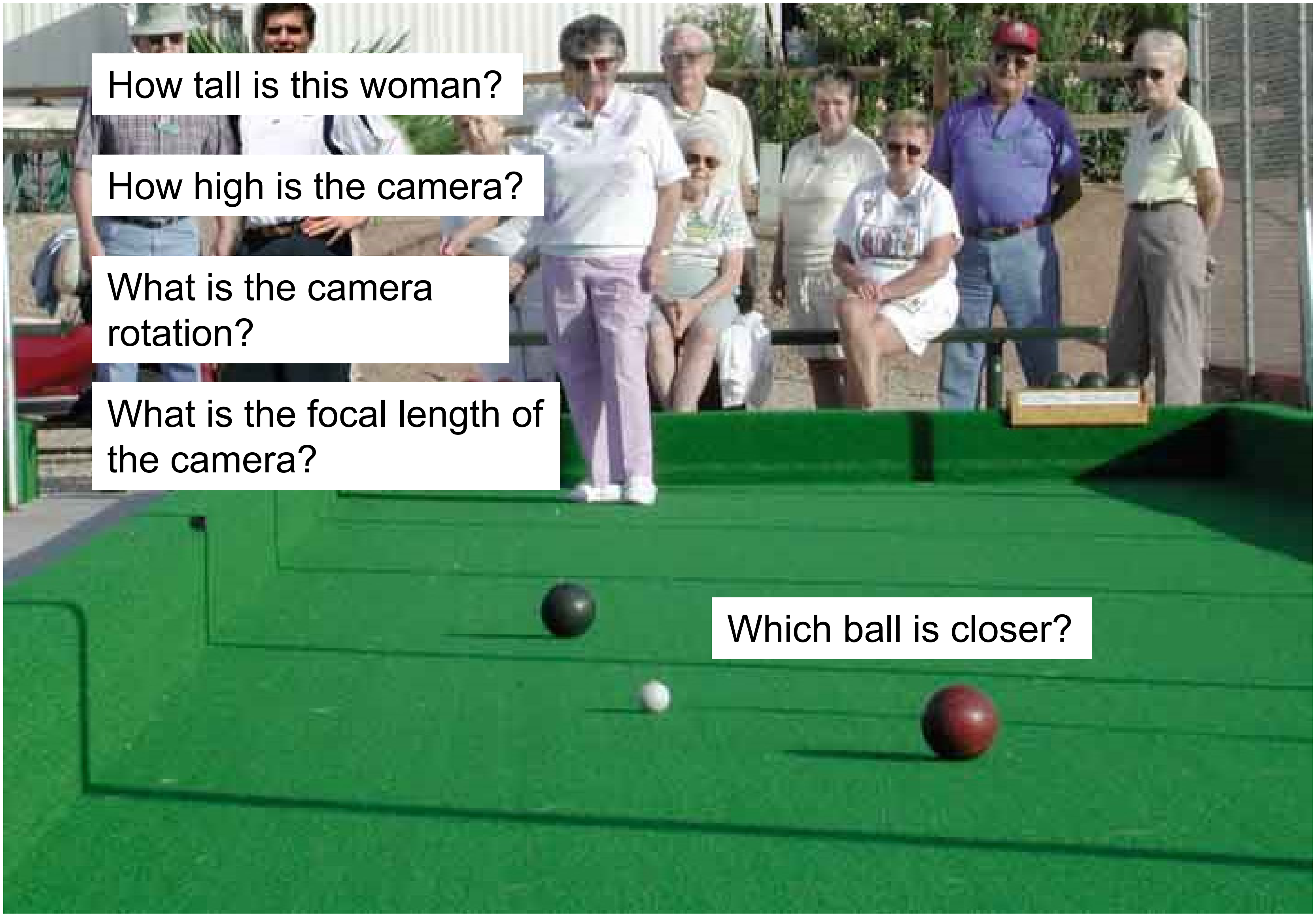
- Prob1: covered today, Tues
- Prob2: covered next Thurs
- Prob3: covered following week

Last class: intro

- Overview of vision, examples of state of art

- Logistics

Next two classes: Single-view Geometry

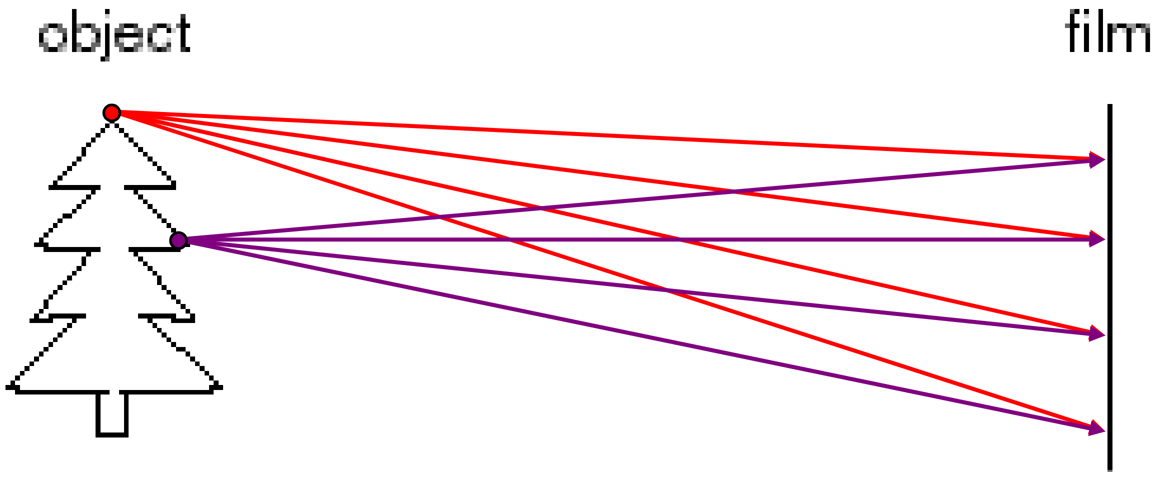


Today's class

Mapping between image and world coordinates

- Pinhole camera model
- Projective geometry
  - homogeneous coordinates and vanishing lines
- Camera matrix
- Other camera parameters

Image formation

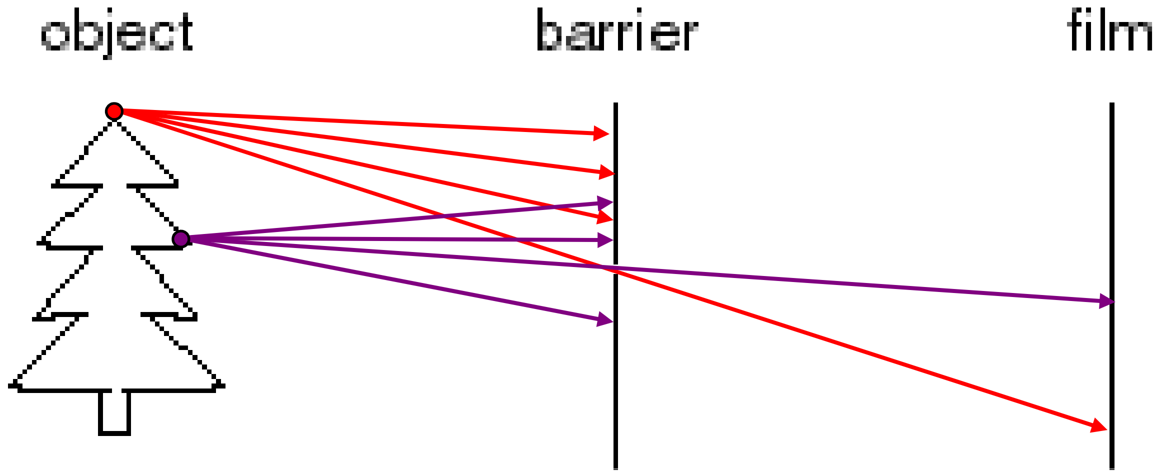


Let's design a camera

- Idea 1: put a piece of film in front of an object
- Do we get a reasonable image?

Slide source: Seitz

Pinhole camera

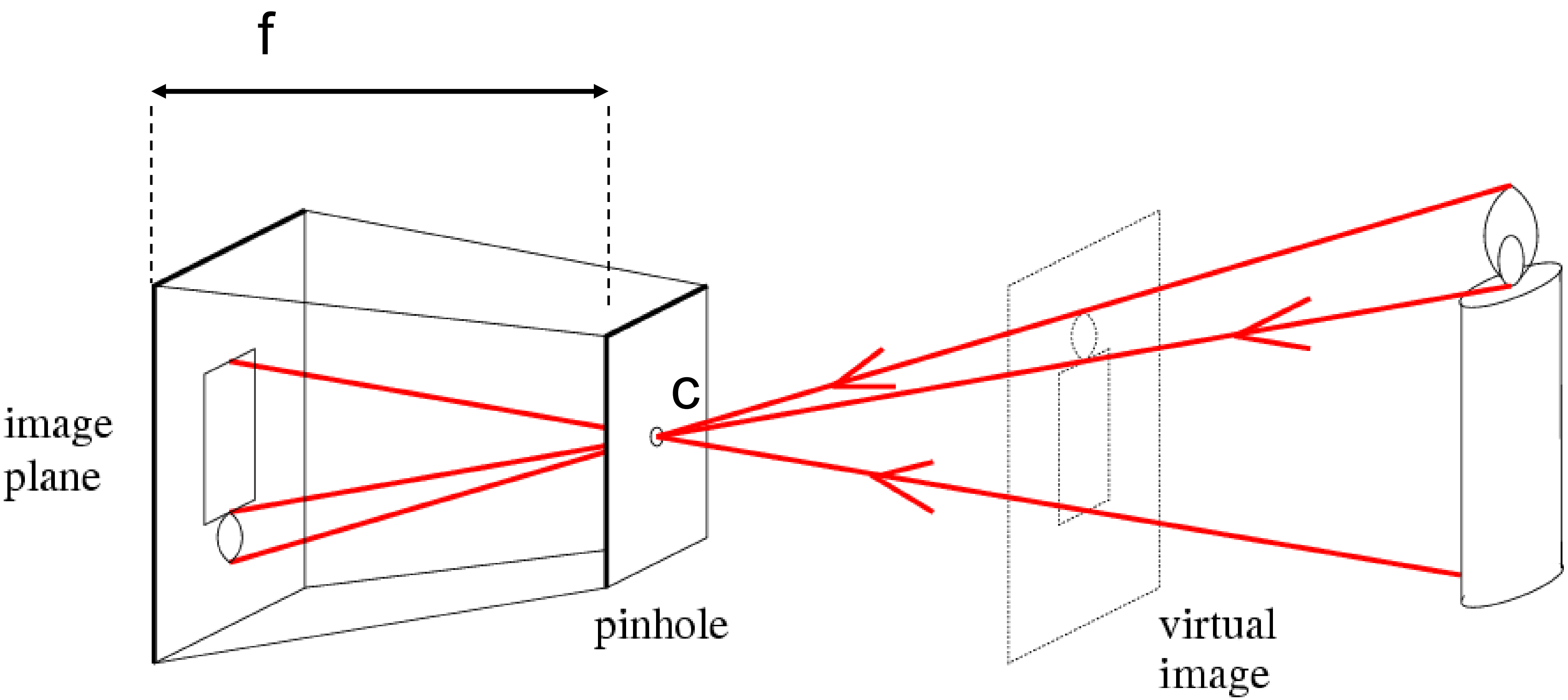


Idea 2: add a barrier to block off most of the rays

- This reduces blurring
- The opening known as the **aperture**

Slide source: Seitz

Pinhole camera



f = focal length
c = center of the camera

Figure from Forsyth

Camera obscura: the pre-camera

- First idea: Mo-Ti, China (470BC to 390BC)

- First built: Alhacen, Iraq/Egypt (965 to 1039AD)

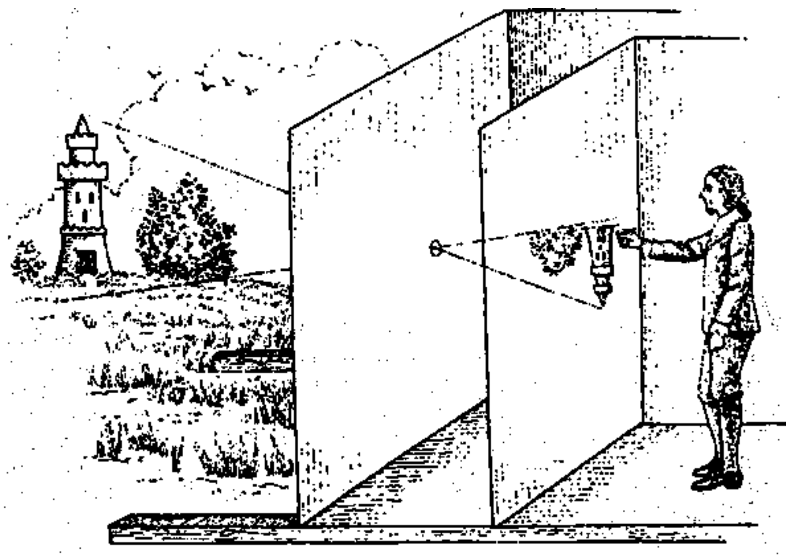


Illustration of Camera Obscura

Freestanding camera obscura at UNC Chapel Hill

Photo by Seth Ilys

First Photograph

First photograph

- Took 8 hours on pewter plate

Joseph Niepce, 1826

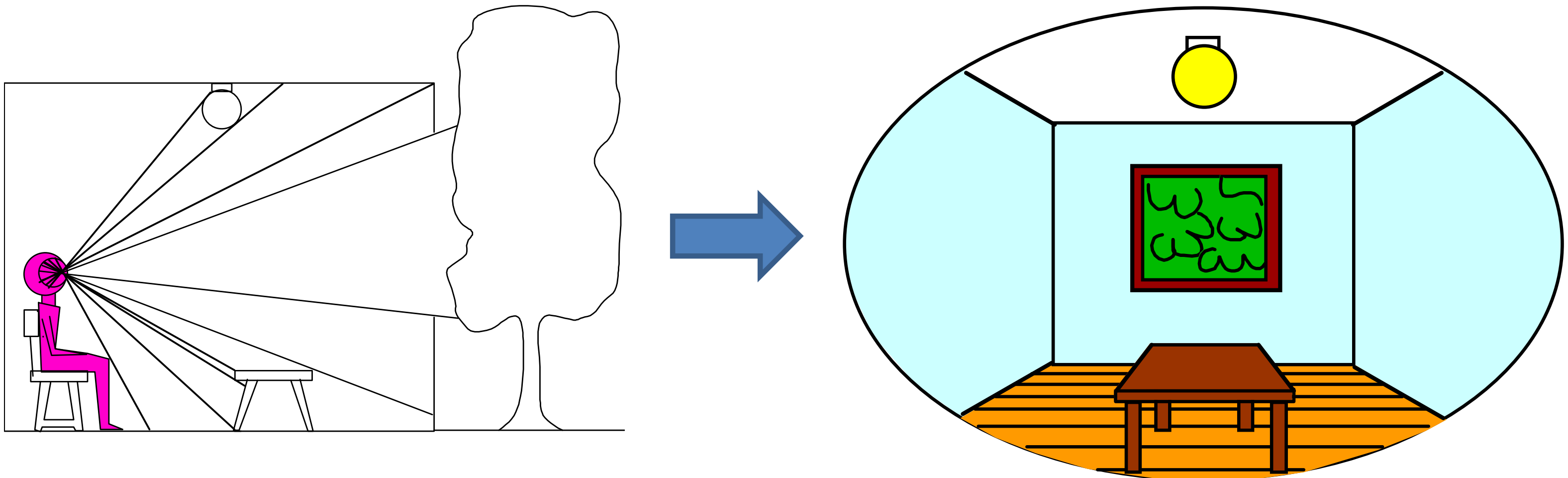
Photograph of the first photograph

Stored at UT Austin

Dimensionality Reduction Machine (3D to 2D)

3D world

2D image



Point of observation

Figures © Stephen E. Palmer, 2002

Slide source: Seitz

Projection can be tricky…

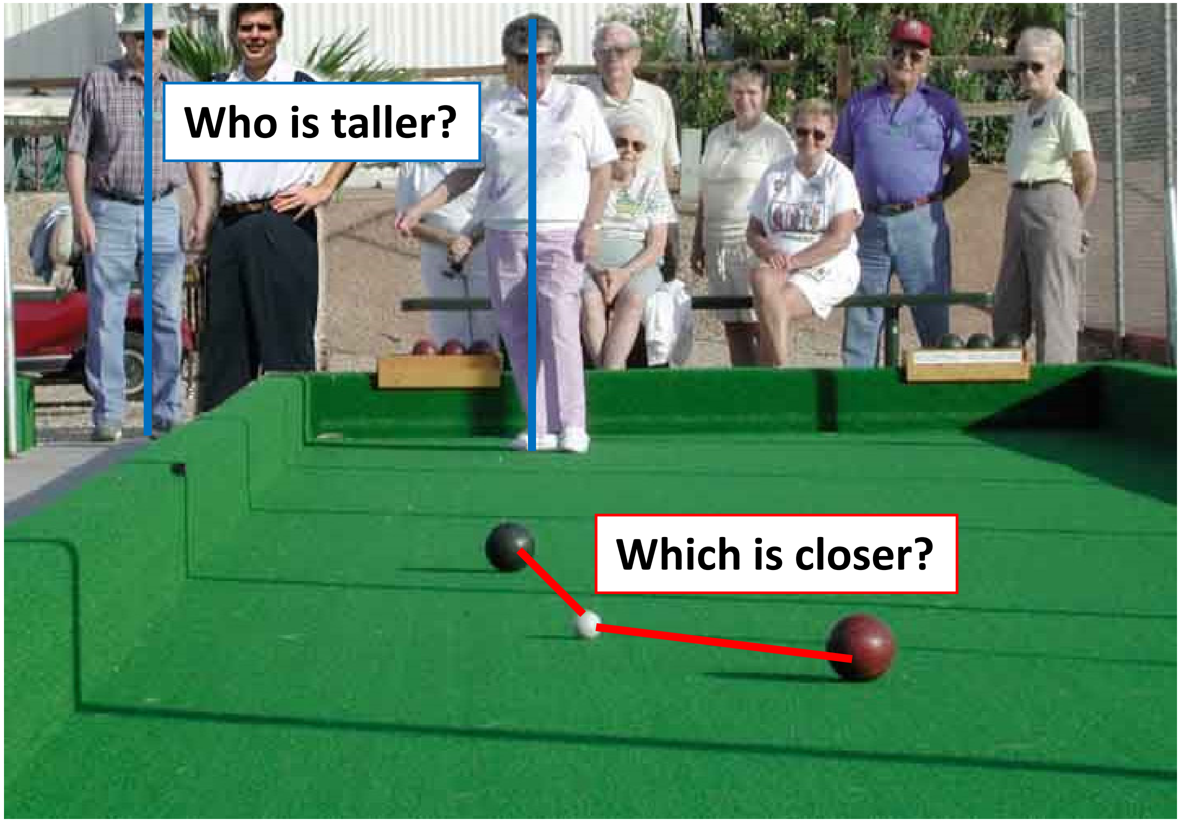
Slide source: Seitz

Projection can be tricky…

Projective Geometry

What is lost?

- Length



Length is not preserved

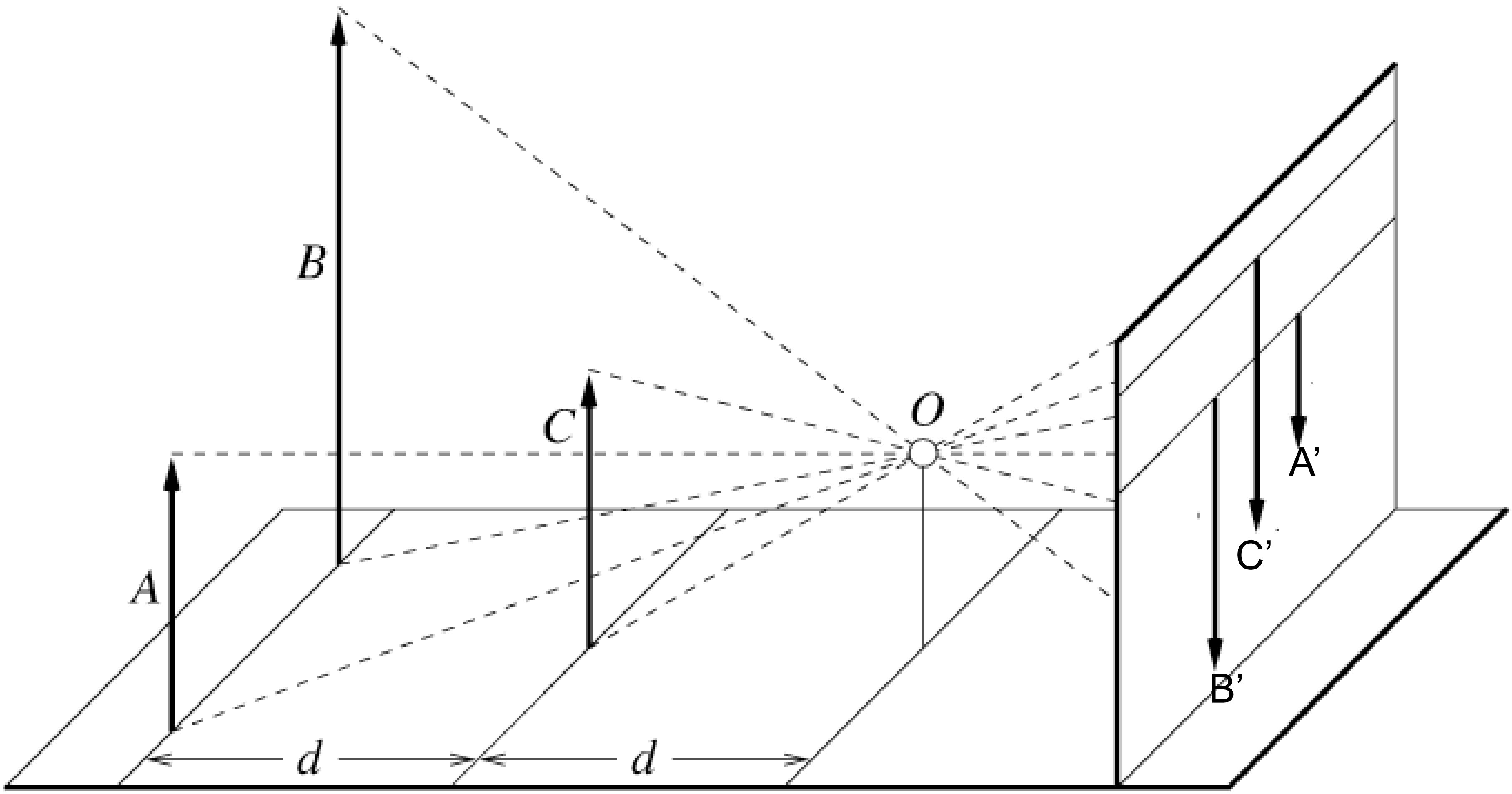
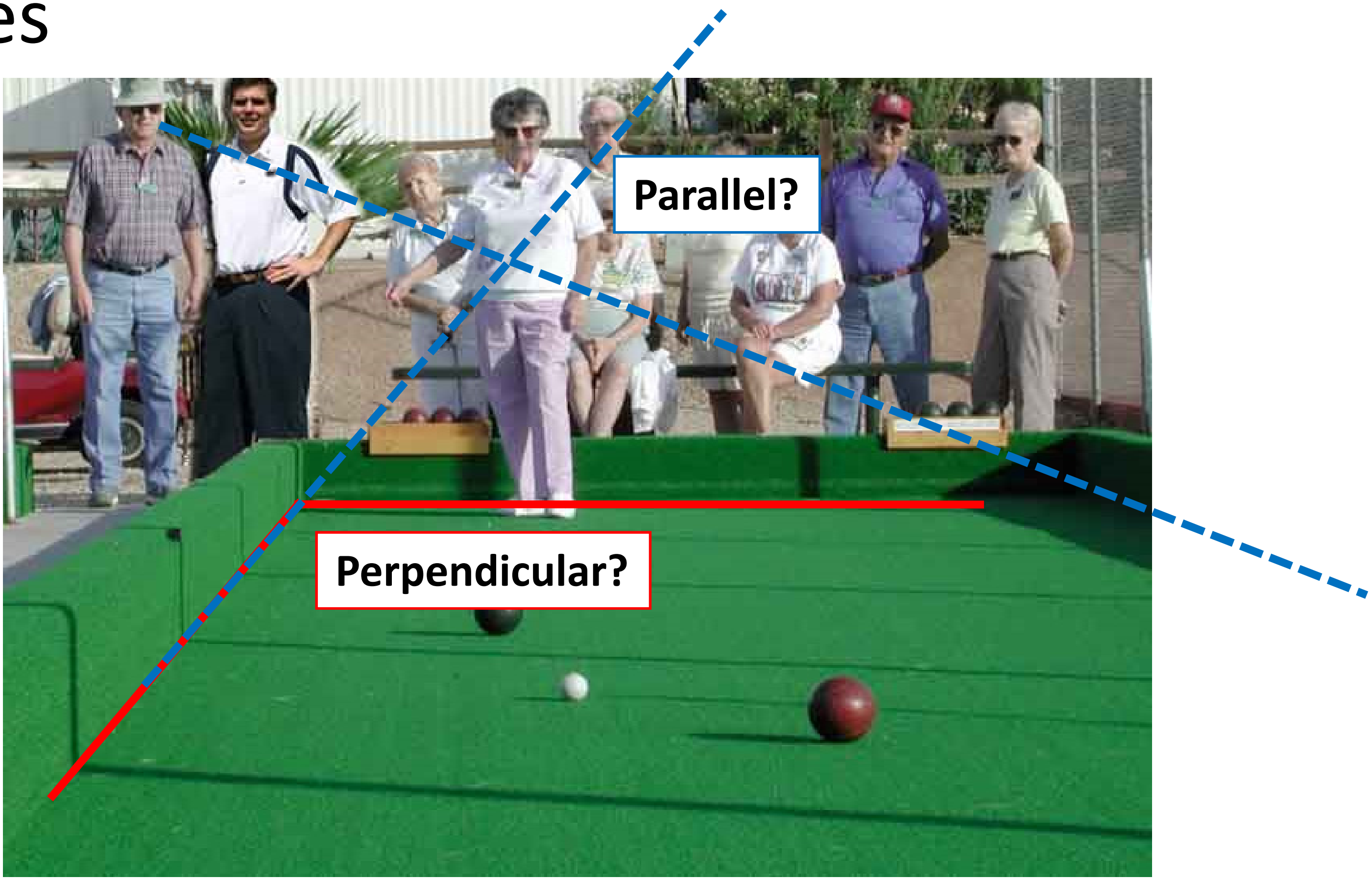


Figure by David Forsyth

Projective Geometry

What is lost?

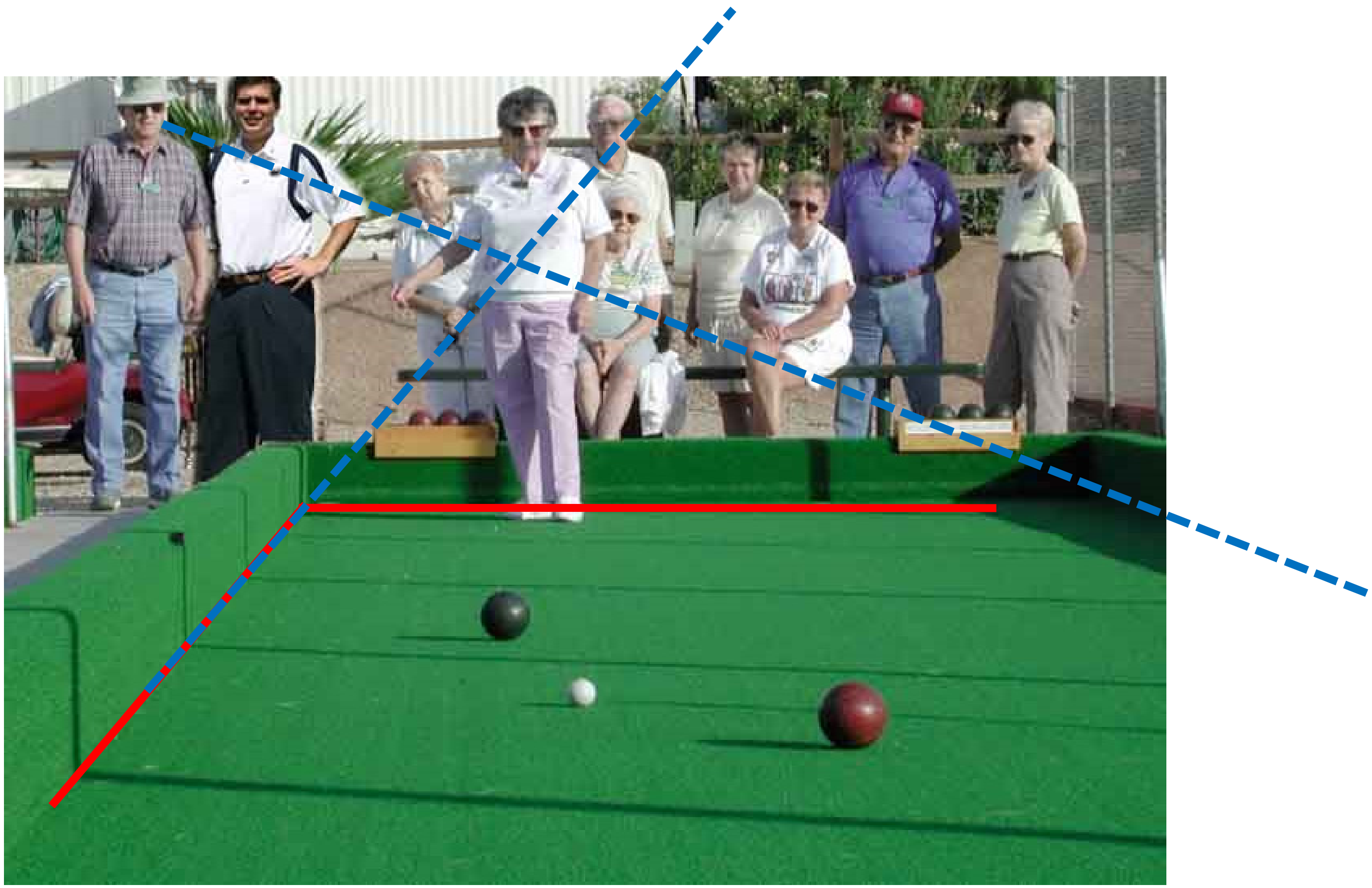
- Length
- Angles



Projective Geometry

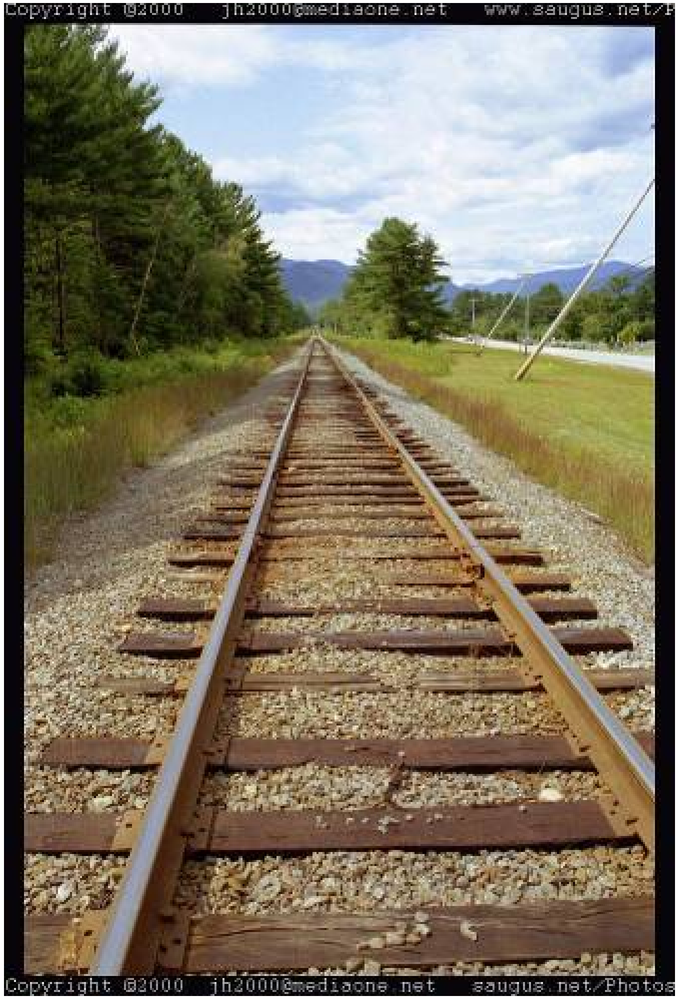
What is preserved?

- Straight lines are still straight

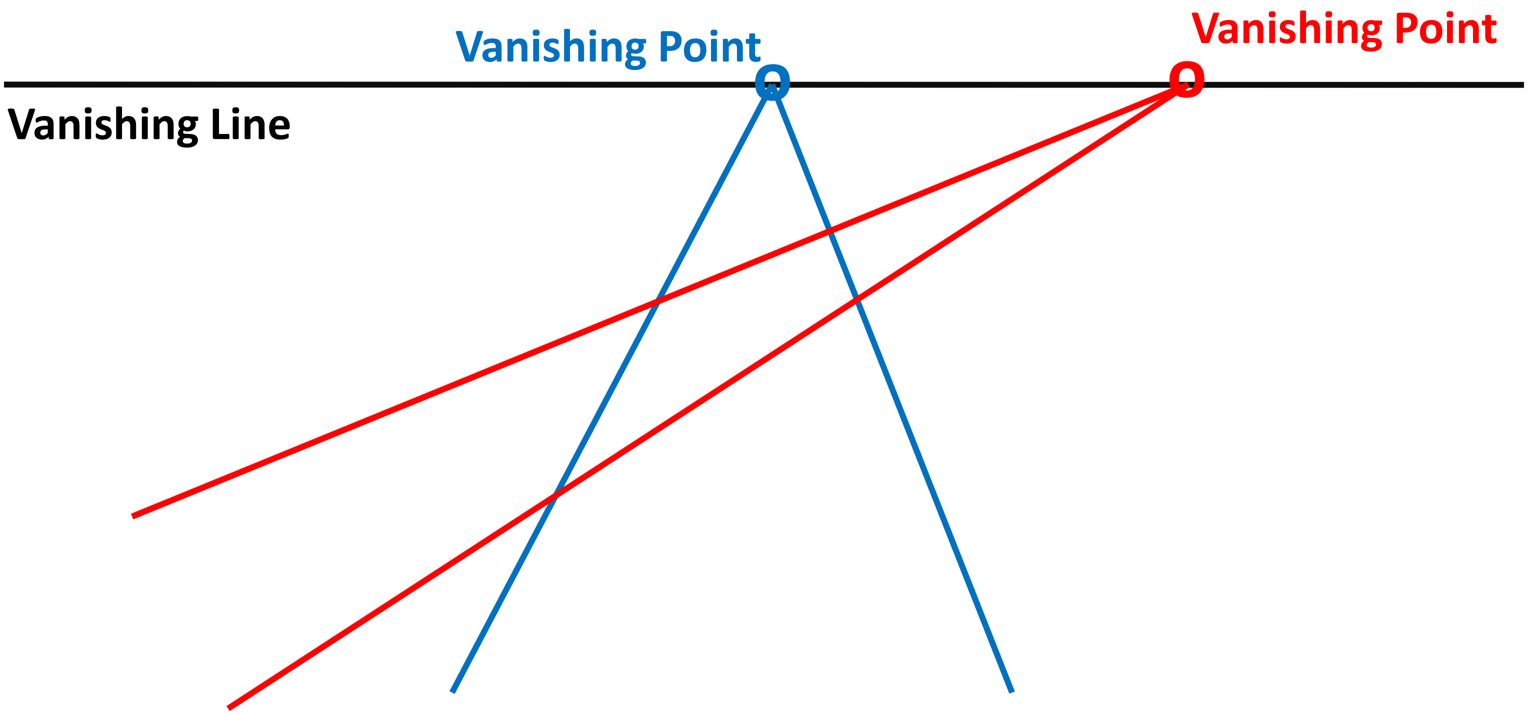


Vanishing points and lines

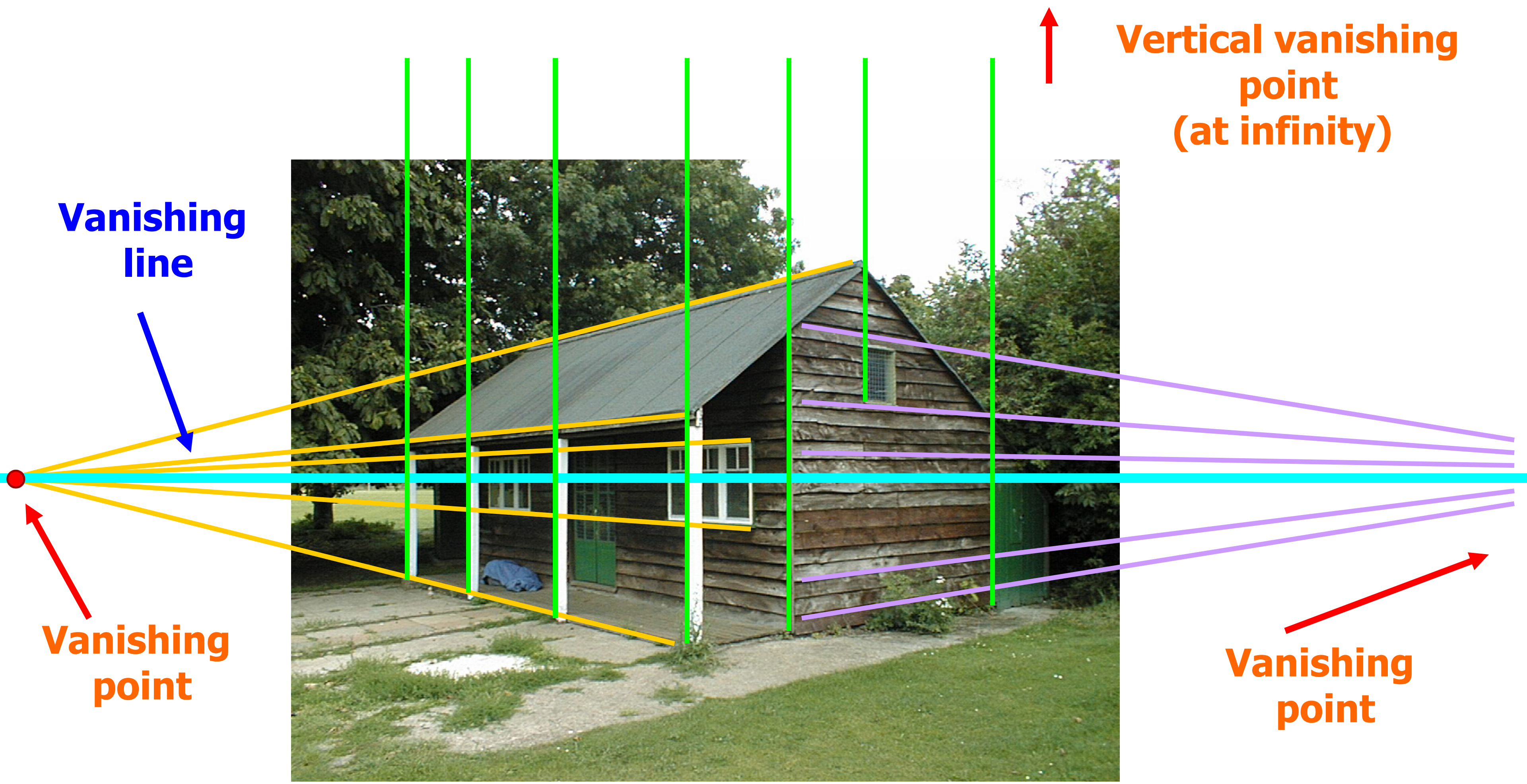
Parallel lines in the world intersect in the image at a "vanishing point"



Vanishing points and lines



Vanishing points and lines

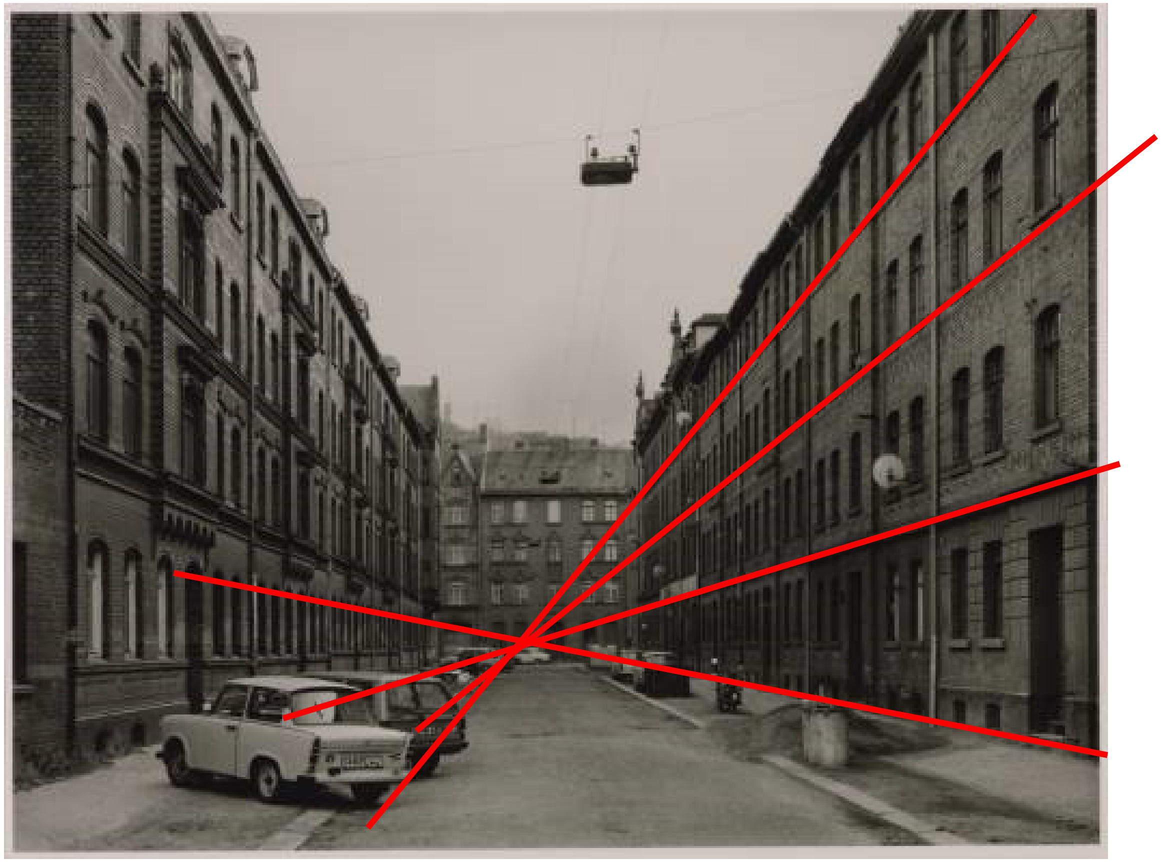


Slide from Efros, Photo from Criminisi

Vanishing points and lines

Photo from online Tate collection

Note on estimating vanishing points



Use multiple lines for better accuracy

… but lines will not intersect at exactly the same point in practice

One solution: take mean of intersecting pairs

… bad idea!

Instead, minimize angular differences

(more vanishing points on board)

Vanishing objects

Projection: world coordinates$\rightarrow$image coordinates

(work on board)

Homogeneous coordinates

Conversion

Converting to *homogeneous* coordinates

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

homogeneous image coordinates

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

homogeneous scene coordinates

Converting *from* homogeneous coordinates

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

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

Homogeneous coordinates

Invariant to scaling

$$k\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

Homogeneous Coordinates

Euclidean Coordinates

Point in Euclidean is ray in Homogeneous

Basic geometry in homogeneous coordinates

- Line equation: $ax + by + c = 0$

$$line_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix}$$

- Append 1 to pixel coordinate to get homogeneous coordinate

$$p_i = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$$

- Line given by cross product of two points

$$line_{ij} = p_i \times p_j$$

- Intersection of two lines given by cross product of the lines

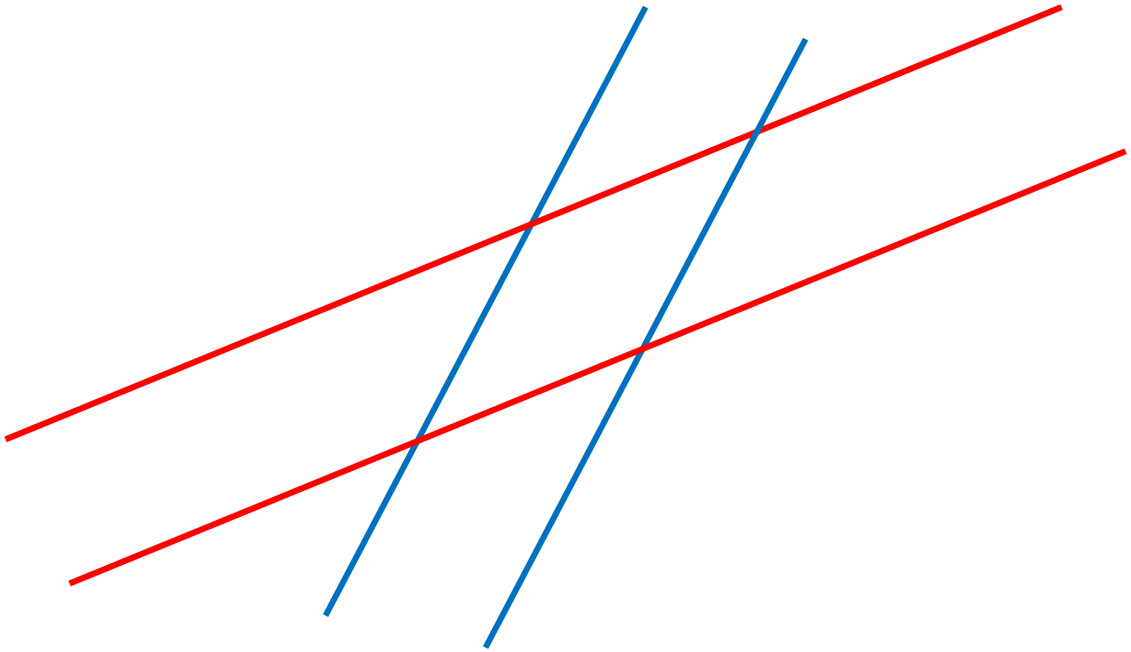
$$q_{ij} = line_i \times line_j$$

Another problem solved by homogeneous coordinates

Intersection of parallel lines

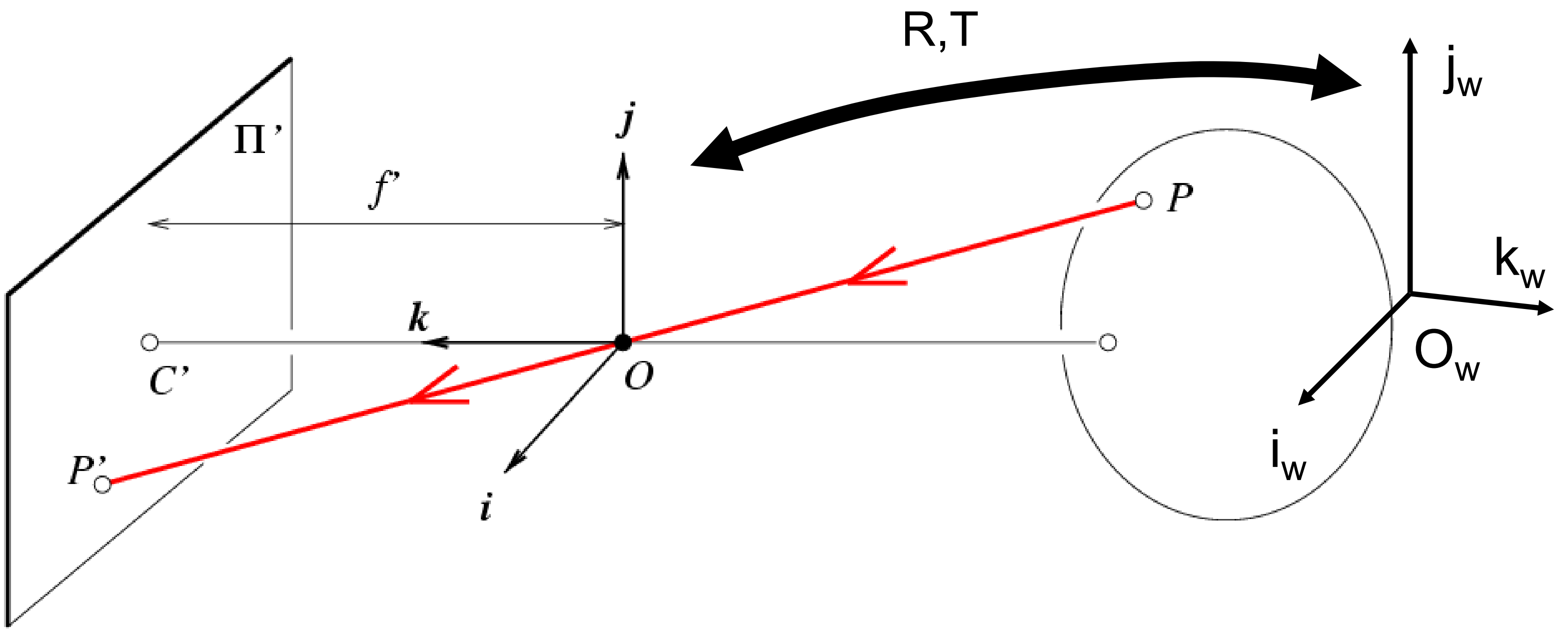
Euclidean: (Inf, Inf)
Homogeneous: (1, 1, 0)

Euclidean: (Inf, Inf)
Homogeneous: (1, 2, 0)



Slide Credit: Saverese

Projection matrix



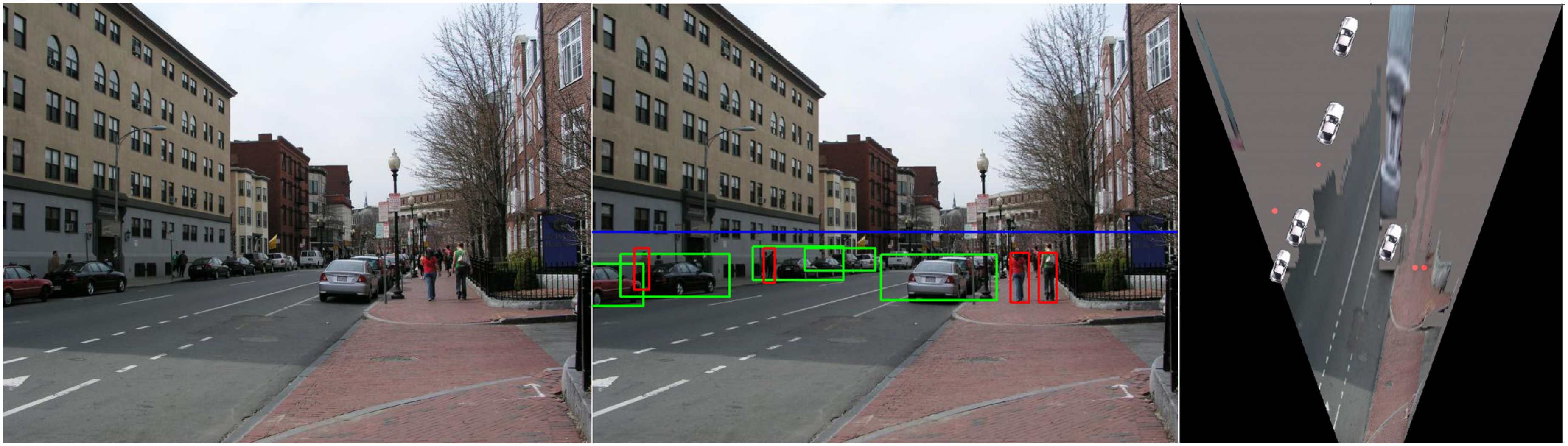
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
X: World Coordinates: (X,Y,Z,1)

Interlude: when have I used this stuff?

When have I used this stuff?

Object Recognition (CVPR 2006)

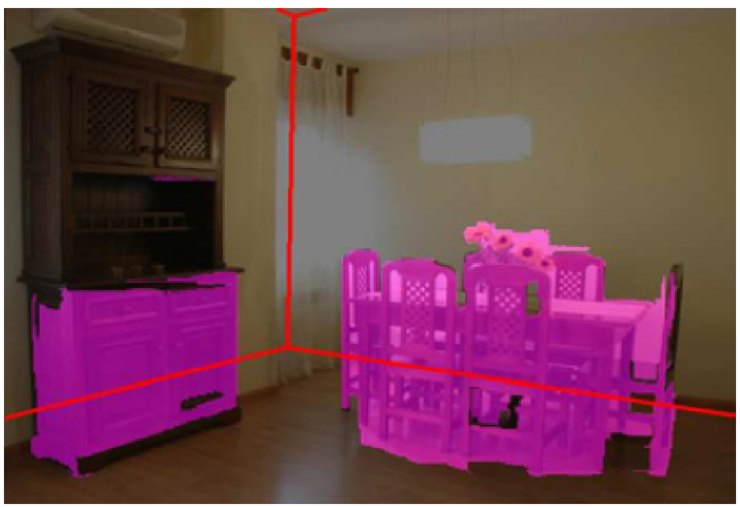


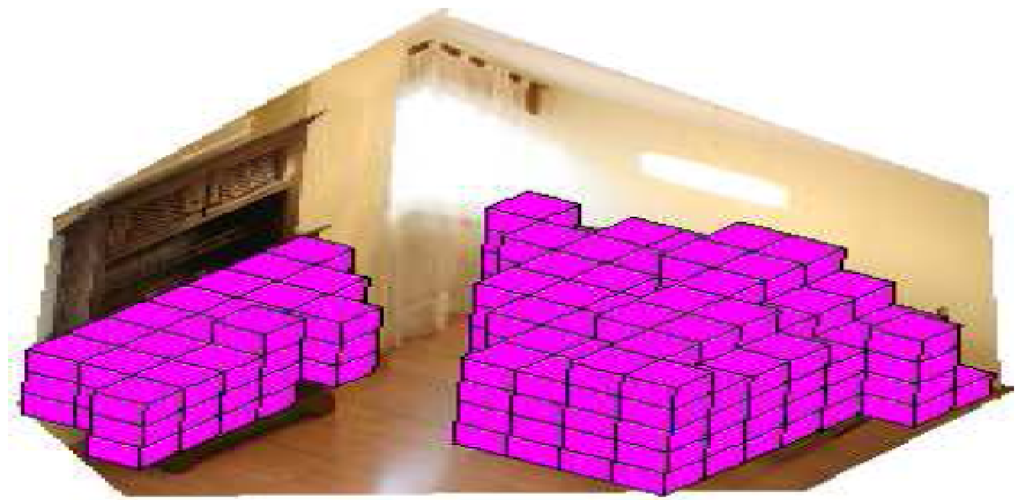
When have I used this stuff?

Single-view reconstruction (SIGGRAPH 2005)

When have I used this stuff?

Getting spatial layout in indoor scenes (ICCV 2009)





When have I used this stuff?

Inserting photographed objects into images (SIGGRAPH 2007)

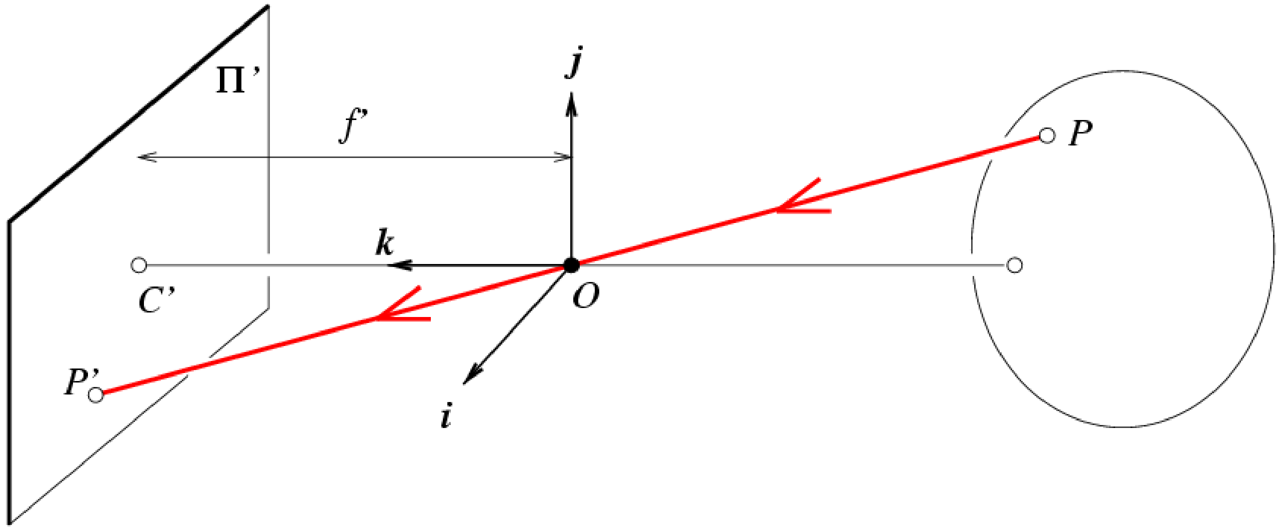
Original

Created

When have I used this stuff?

Inserting synthetic objects into images (submitted)

Projection matrix



Intrinsic Assumptions

- Unit aspect ratio
- Optical center at (0,0)
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow k\begin{bmatrix}u \\ v \\ 1\end{bmatrix} = \begin{bmatrix}\boxed{\begin{matrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1\end{matrix}}_{\mathbf{K}} & \begin{matrix}0 \\ 0 \\ 0\end{matrix}\end{bmatrix}\begin{bmatrix}x \\ y \\ z \\ 1\end{bmatrix}$$

Slide Credit: Saverese

Remove assumption: known optical center

Intrinsic Assumptions

- Unit aspect ratio
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \quad \Rightarrow \quad k\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Remove assumption: square pixels

Intrinsic Assumptions
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow k\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Remove assumption: non-skewed pixels

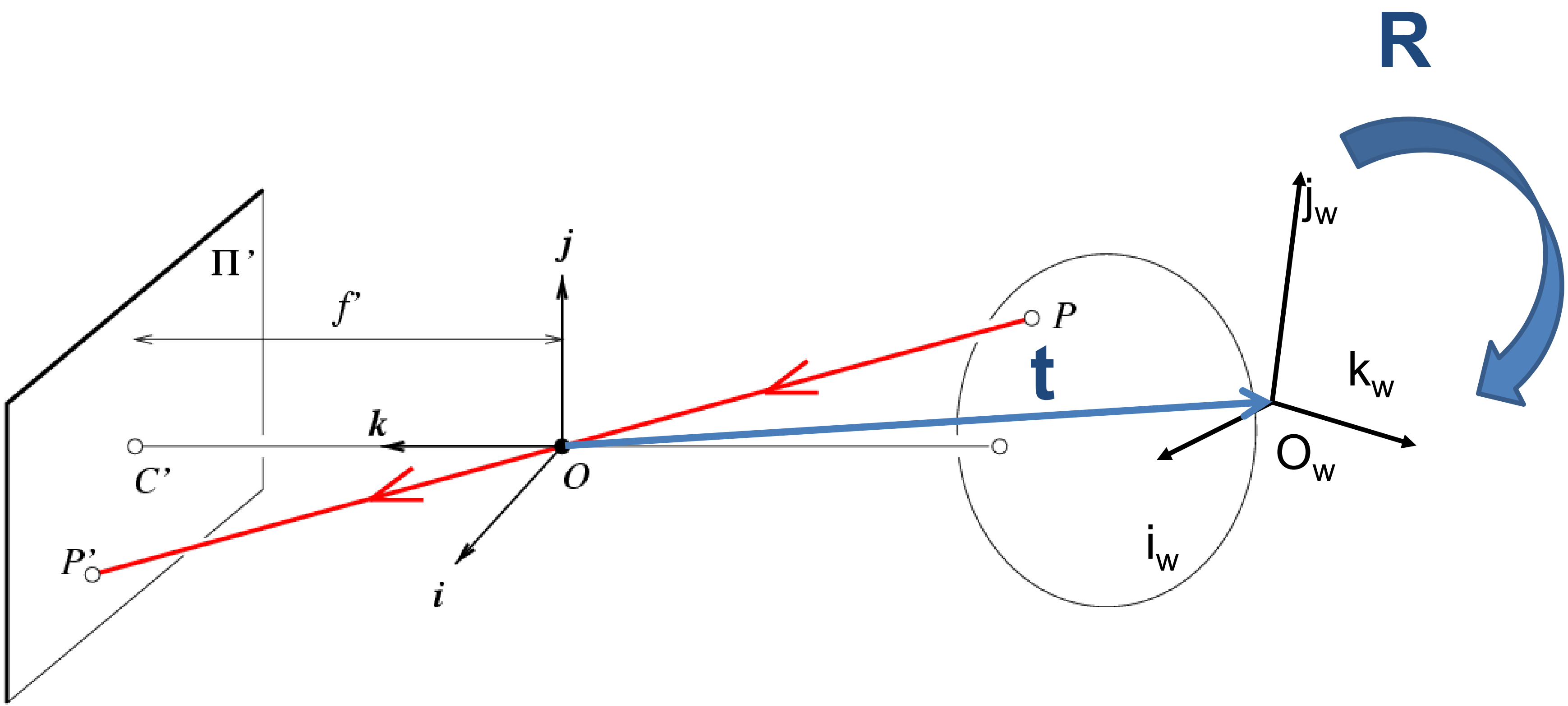
Intrinsic Assumptions

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow k\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Note: different books use different notation for parameters

Oriented and Translated Camera



Allow camera translation

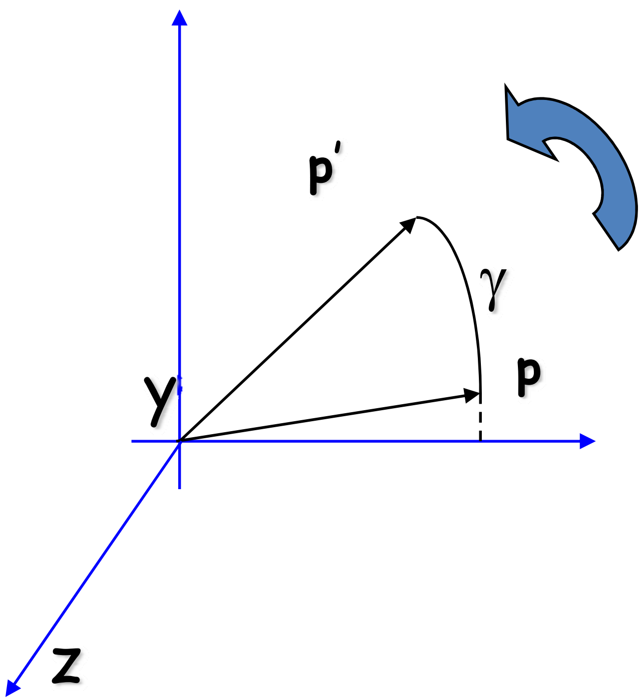
Intrinsic Assumptions

Extrinsic Assumptions
- No rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{t}\end{bmatrix}\mathbf{X} \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Allow camera rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

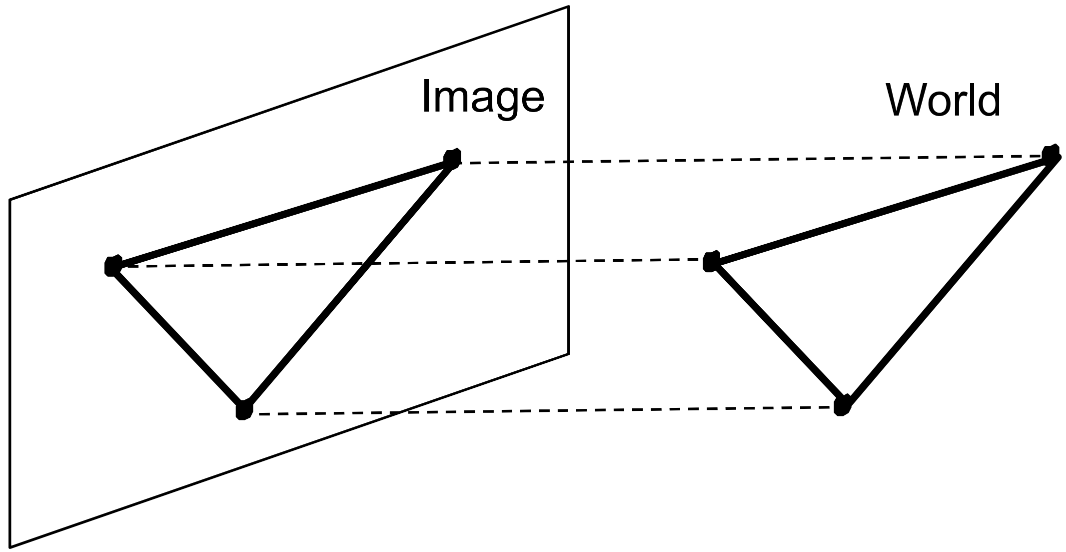
Degrees of freedom

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$k\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \overset{5}{\begin{bmatrix}\alpha & s & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}}\overset{6}{\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}}\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}$$

Scaled Orthographic Projection

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



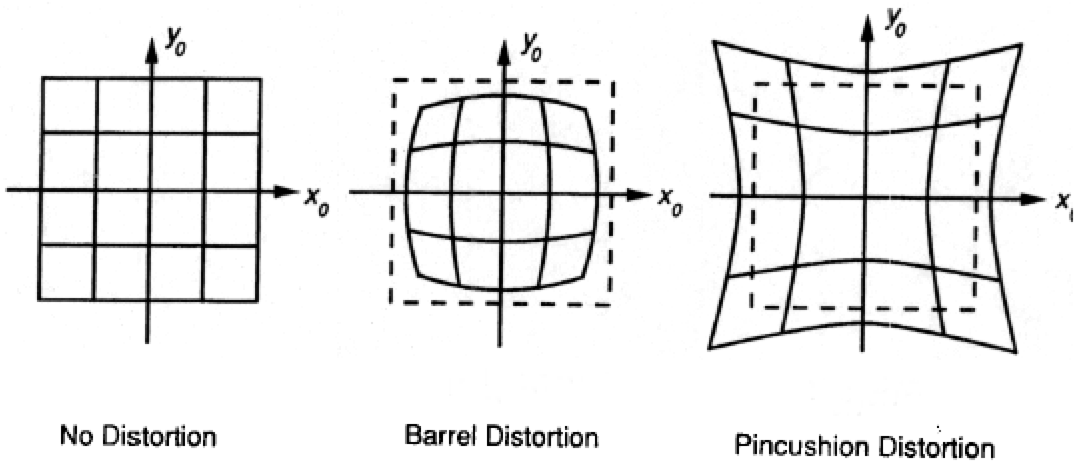
  - Also called "parallel projection"
  - What's the projection matrix?

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Slide by Steve Seitz

Other things to be aware of

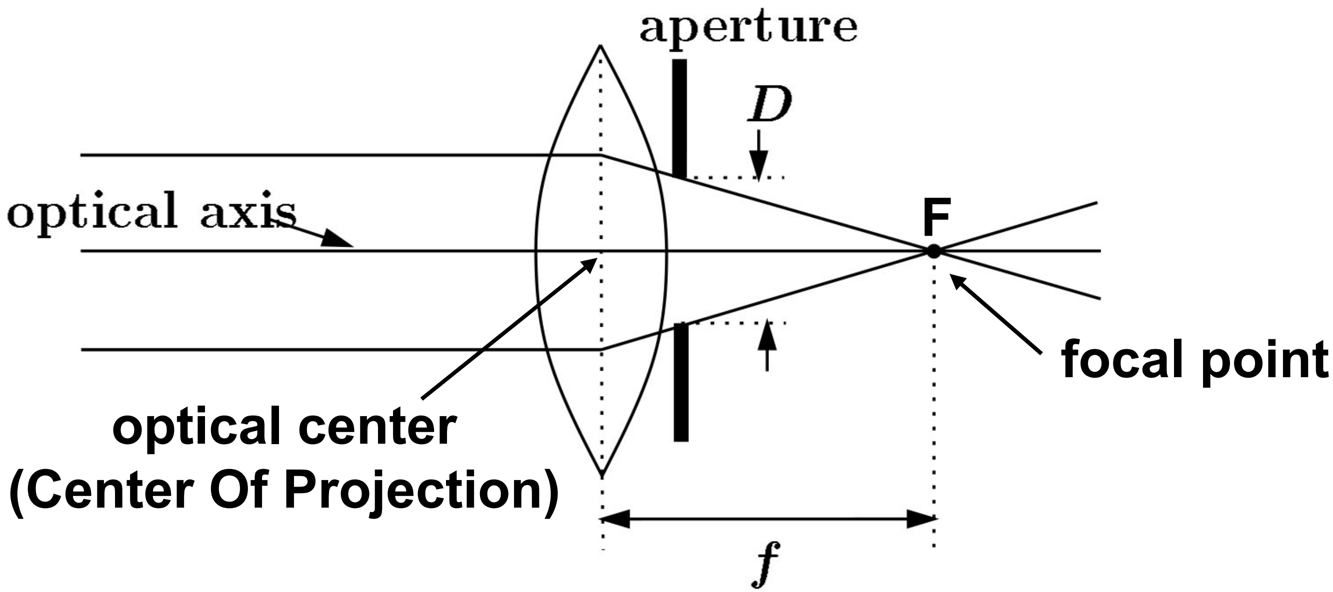
Radial Distortion



Corrected Barrel Distortion

Image from Martin Habbecke

Focal length, aperture, depth of field

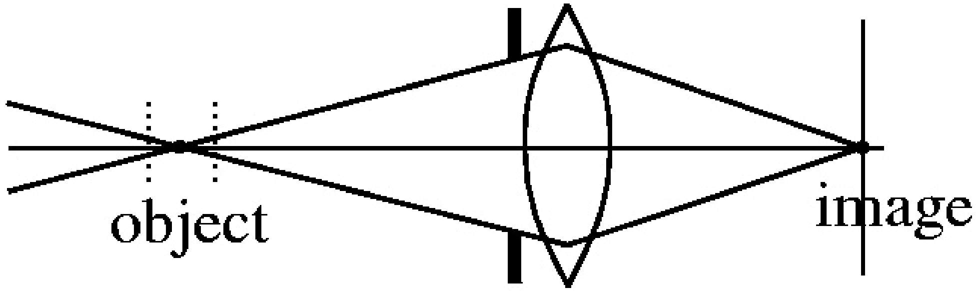


A lens focuses parallel rays onto a single focal point

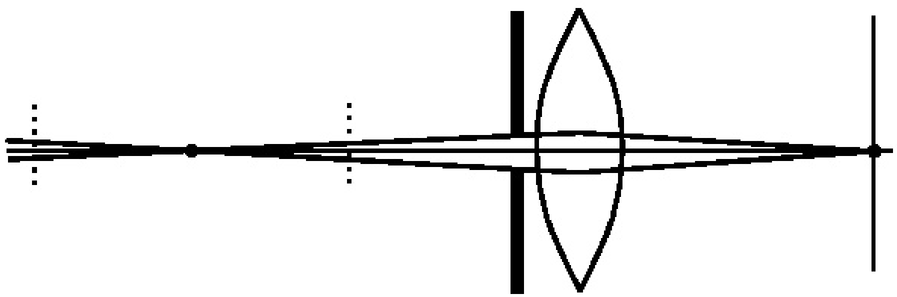
– focal point at a distance *f* beyond the plane of the lens
– Aperture of diameter D restricts the range of rays

Slide source: Seitz

Depth of field



f / 5.6



f / 32

Changing the aperture size or focal length affects depth of field

Flower images from Wikipedia http://en.wikipedia.org/wiki/Depth_of_field

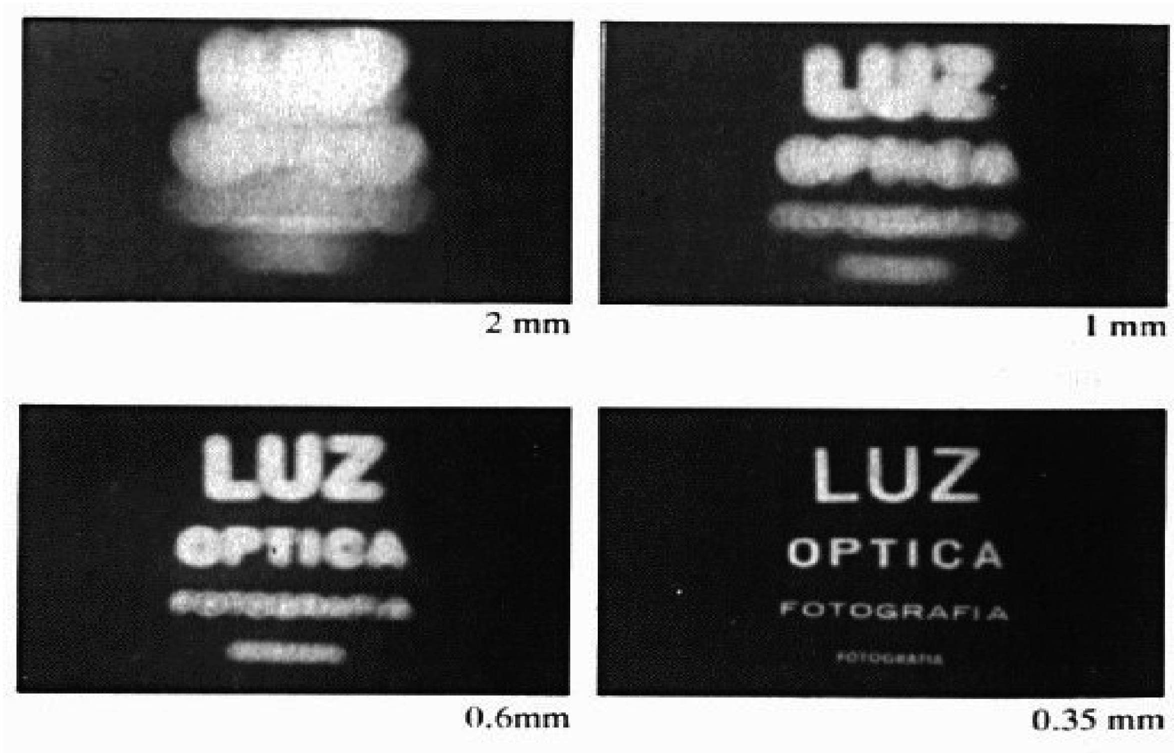
Slide from Efros

Varying the aperture

Large apeture = small DOF

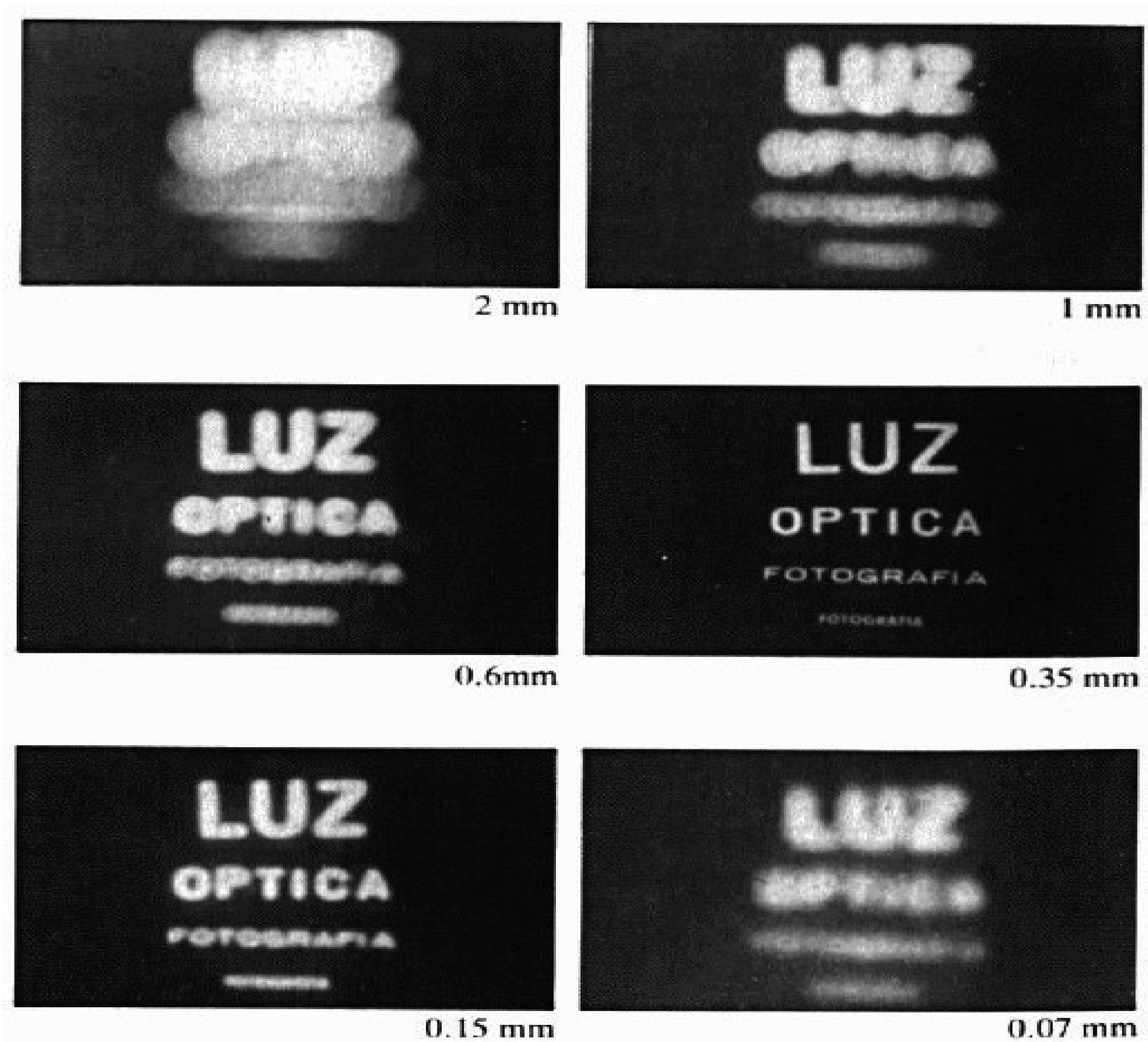
Small apeture = large DOF

Shrinking the aperture



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

Shrinking the aperture



Slide by Steve Seitz

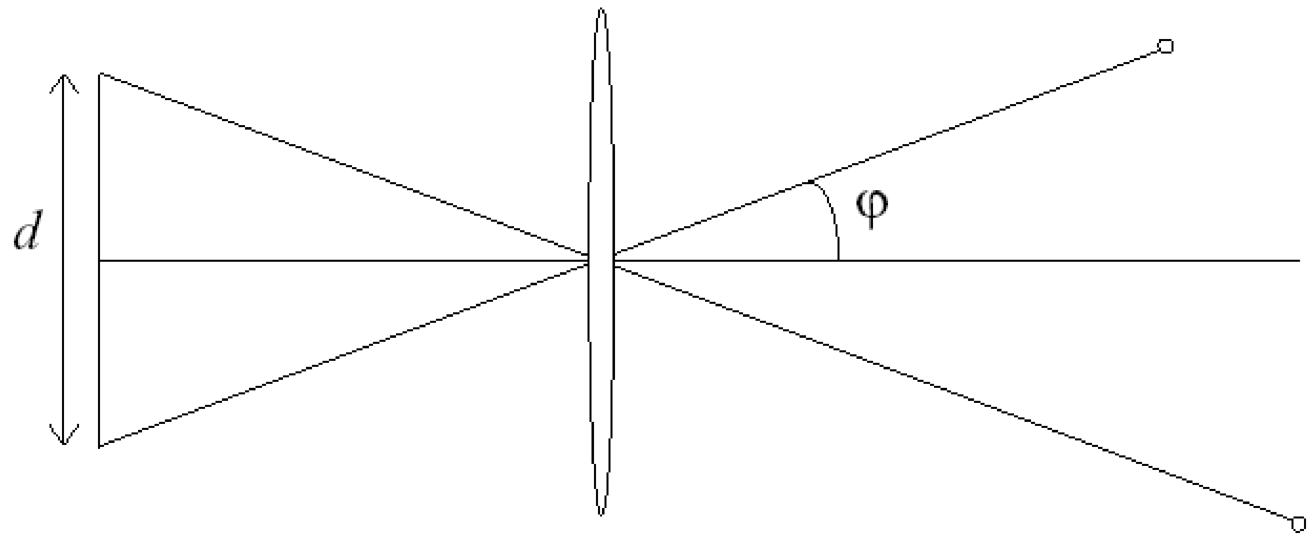
Relation between field of view and focal length

Field of view (angle width)

Film/Sensor Width

$$fov = \tan^{-1} \frac{d}{2f}$$

Focal length

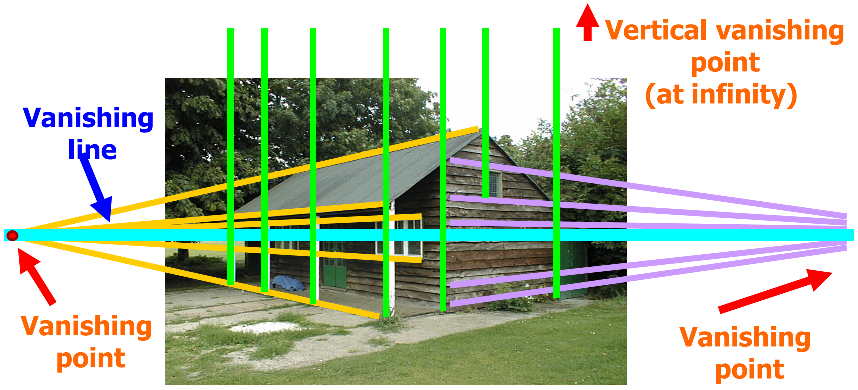


Dolly zoom (or "Vertigo effect")

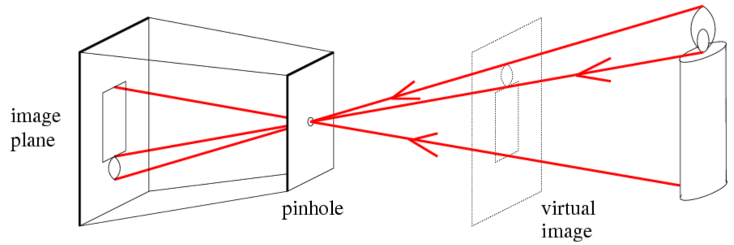
http://www.youtube.com/watch?v=Y48R6-iIYHs

Things to remember

- Vanishing points and vanishing lines



- Pinhole camera model and camera projection matrix



$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

- Homogeneous coordinates

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

Next class

- Applications of camera model and projective geometry
  - Recovering the camera intrinsic and extrinsic parameters from an image
  - Recovering size in the world
  - Projecting from one plane to another (if time allows)

Questions