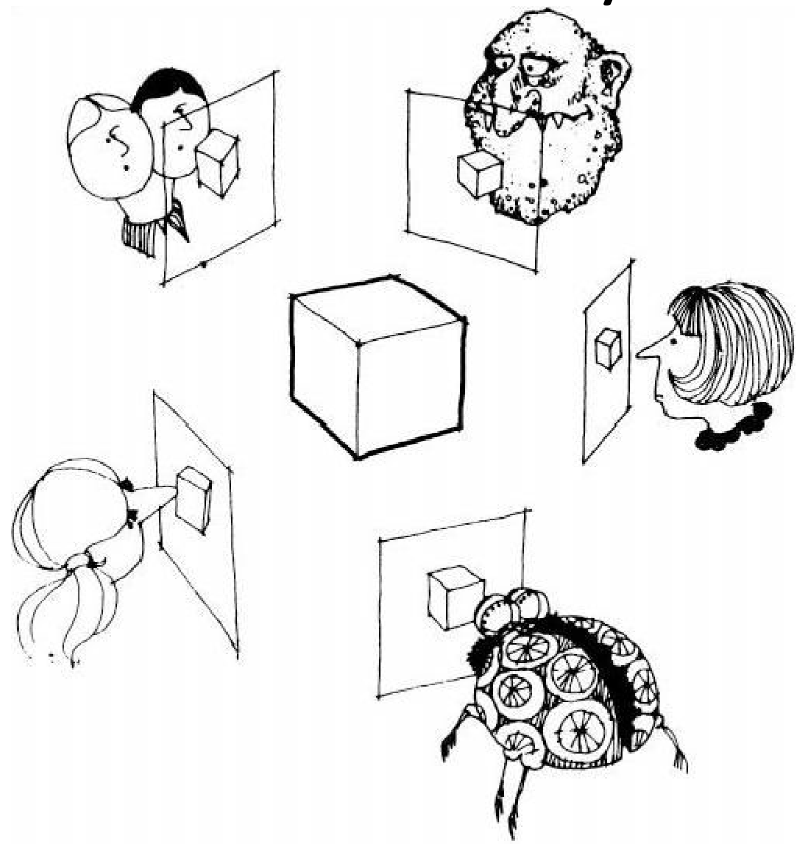
Computer Vision
Camera Models and Projective Geometry

Dr. Mrinmoy Ghorai

Indian Institute of Information Technology
Sri City, Chittoor

Camera Models and Projective Geometry

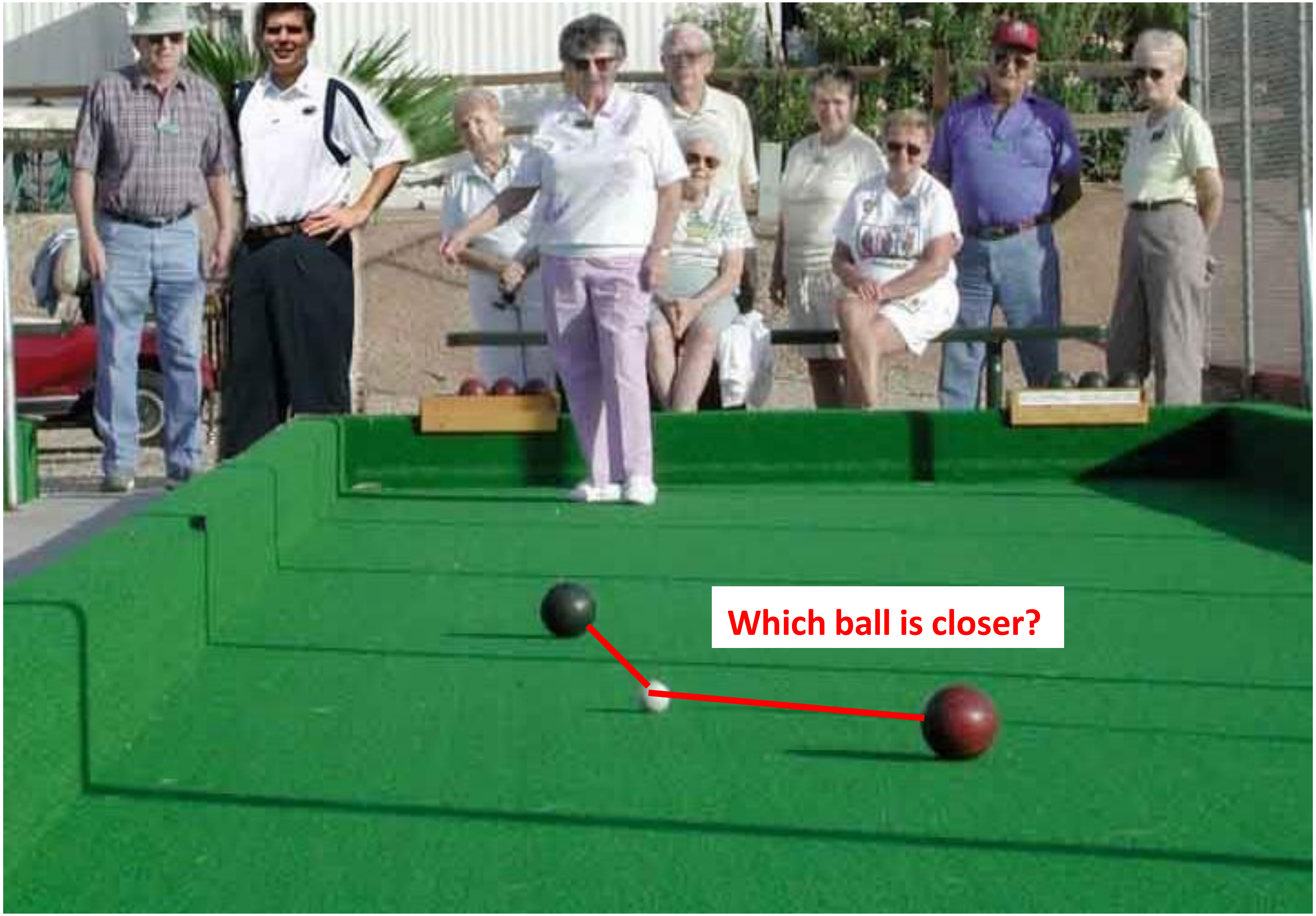


Perspective and 3D Geometry

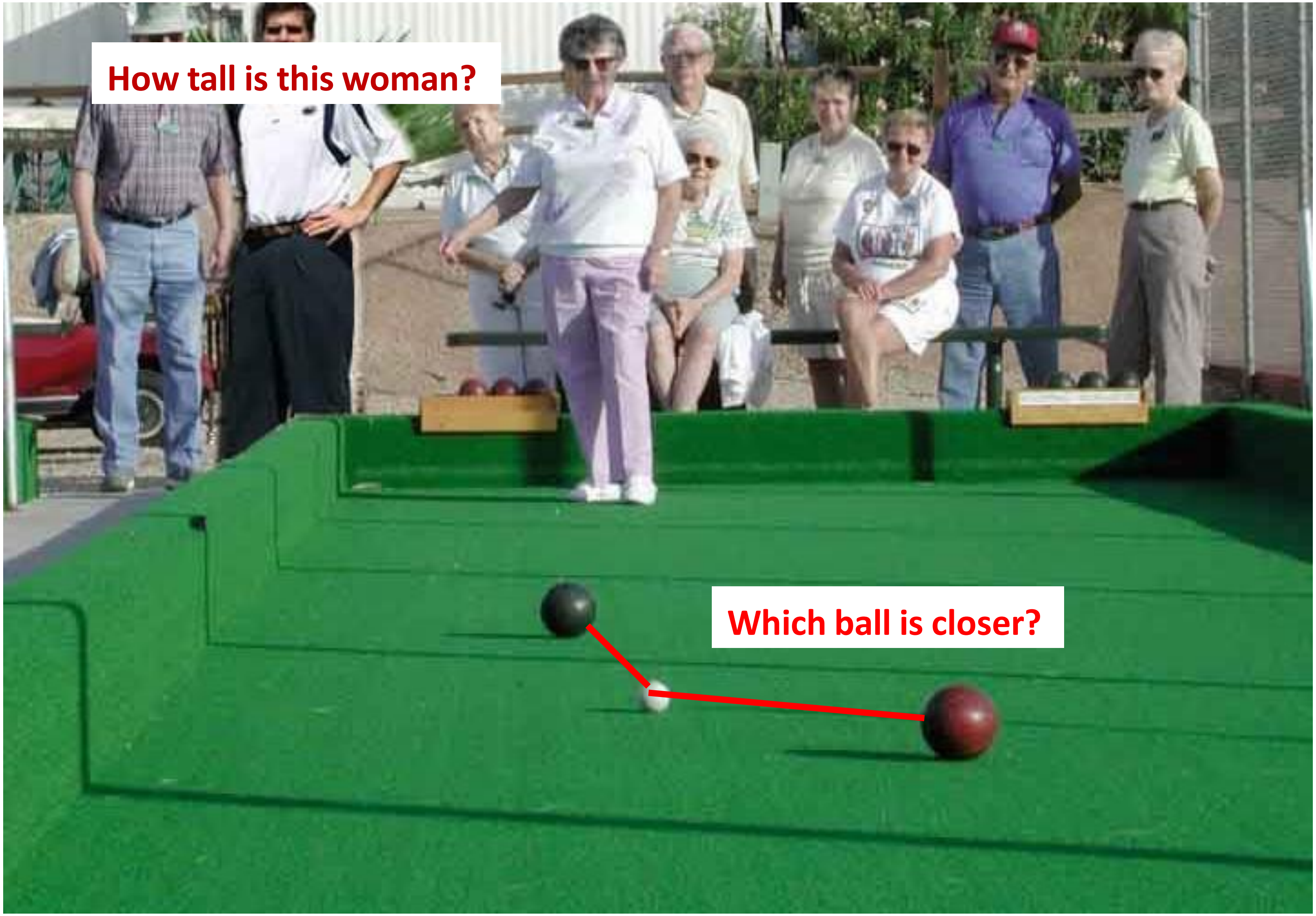
- **Camera models and Projective geometry**
  - What's the mapping between image and world coordinates?
- **Projection Matrix and Camera calibration**
  - What's the projection matrix between scene and image coordinates?
  - How to calibrate the projection matrix?
- **Single view metrology and Camera properties**
  - How can we measure the size of 3D objects in an image?
  - What are the important camera properties?
- **Photo stitching**
  - What's the mapping from two images taken without camera translation?
- **Epipolar Geometry and Stereo Vision**
  - What's the mapping from two images taken with camera translation?
- **Structure from motion**
  - How can we recover 3D points from multiple images?

Next few classes: Single-view Geometry

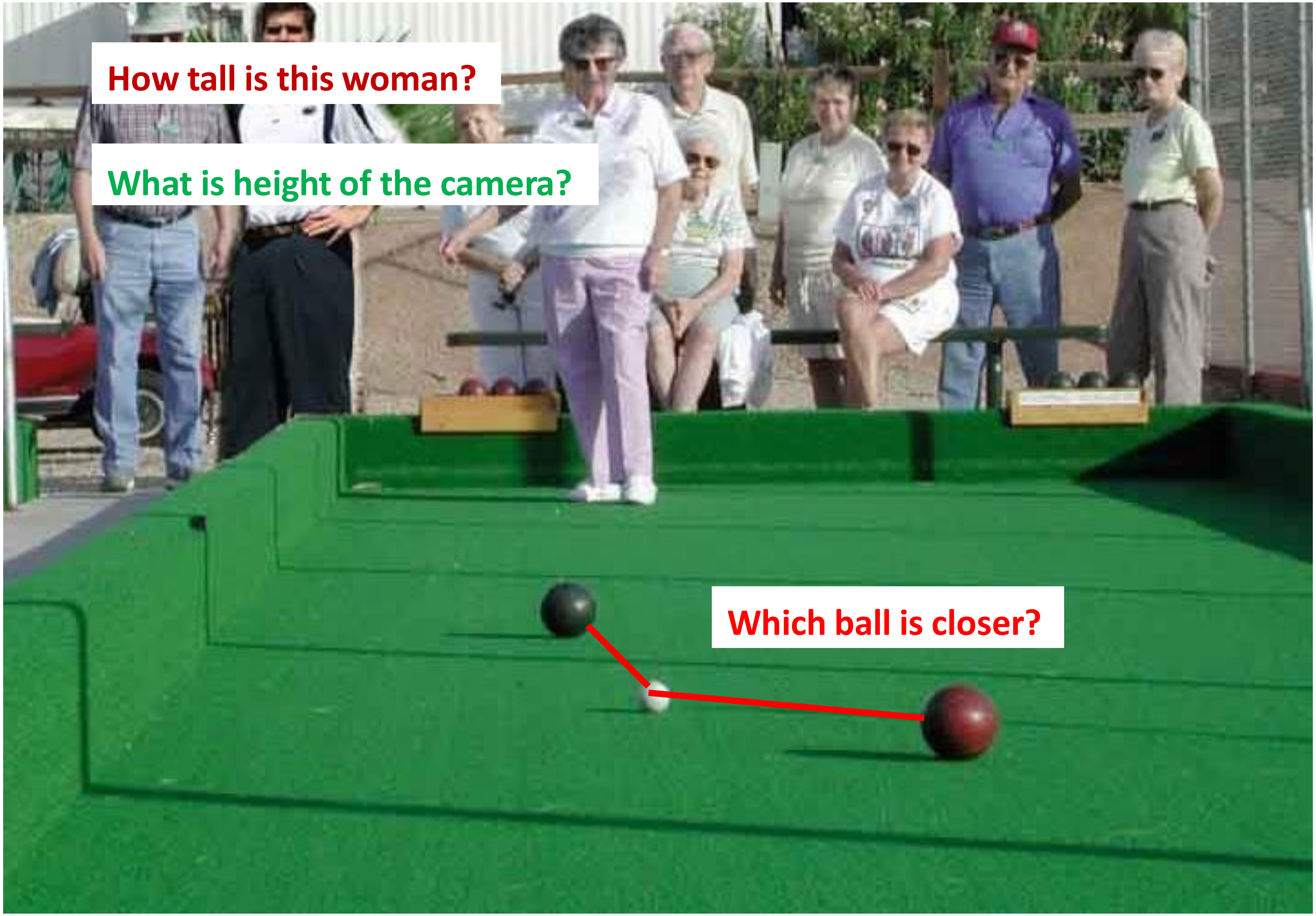
Next few classes: Single-view Geometry



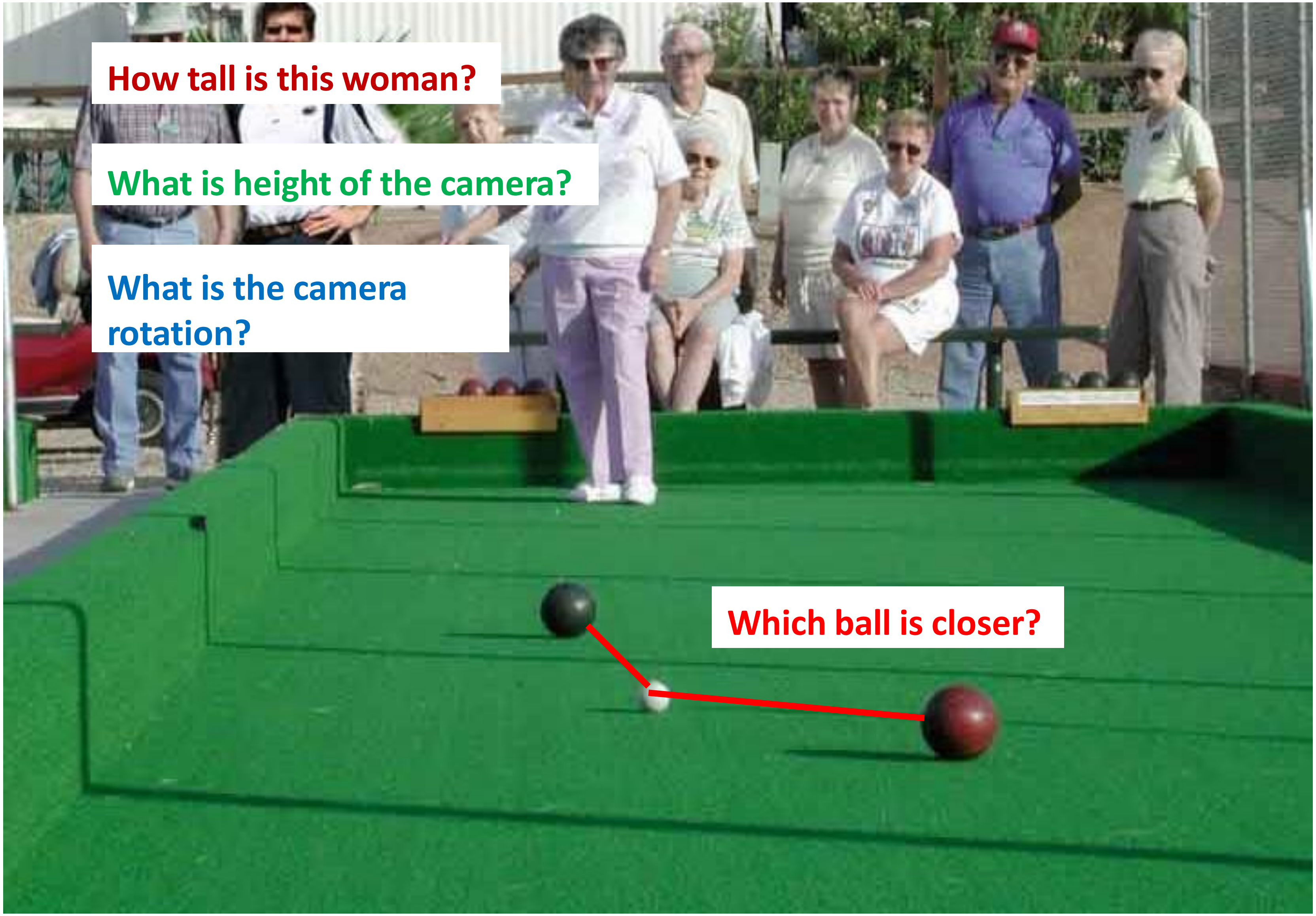
Next few classes: Single-view Geometry



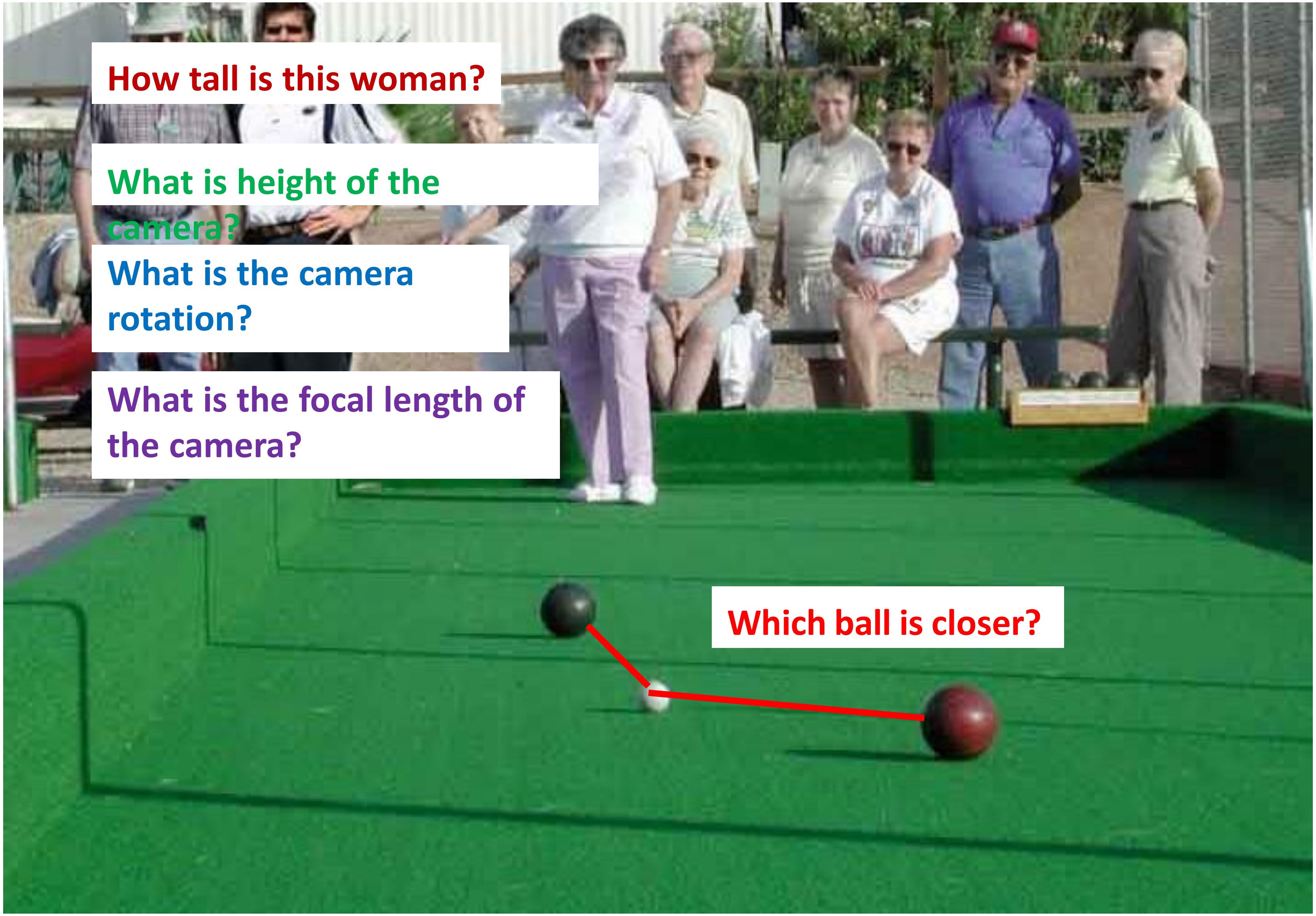
Next few classes: Single-view



Next few classes:
Single-view

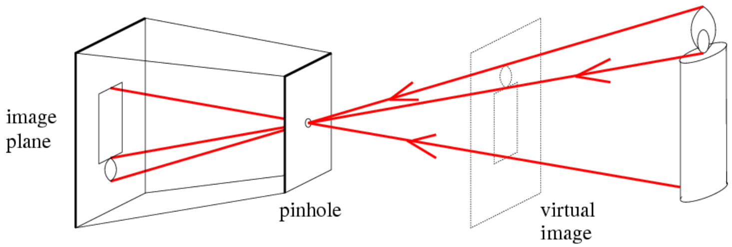


Next few classes: Single-view

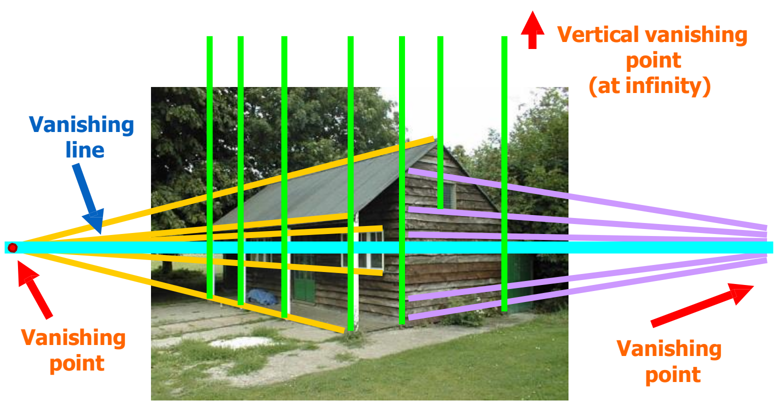


Today's class
Mapping between image and world coordinates

- Pinhole camera model



- Projective geometry
  - Vanishing points and lines



What is an Image?

- Up until now: a function –a 2D pattern of intensity values
- Today: a 2D projection of 3D points

Slide modified from: Bobick

Image formation

Let’s design a camera

–Idea 1: put a piece of film in front of an object

Slide source: Seitz

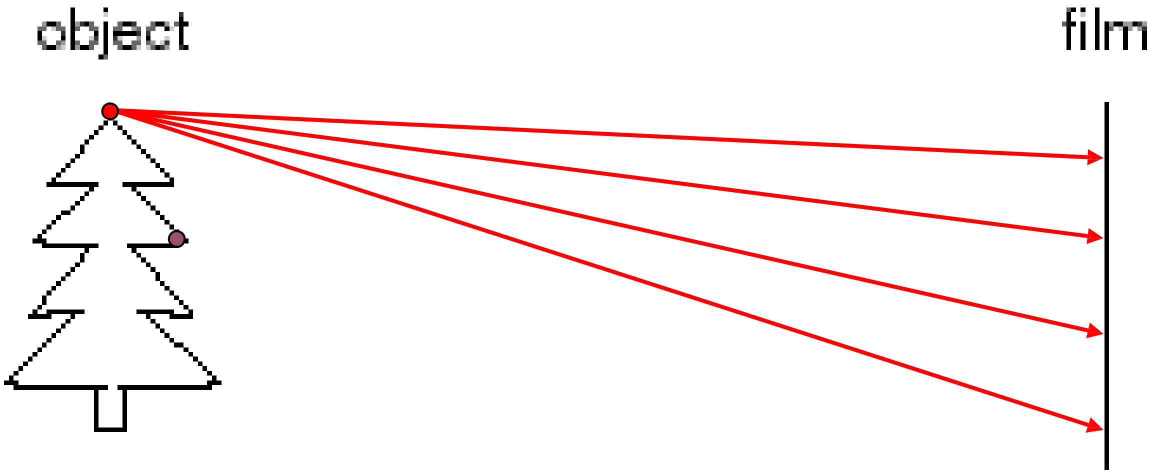
Image formation

Let’s design a camera

–Idea 1: put a piece of film in front of an object

Slide source: Seitz

Image formation

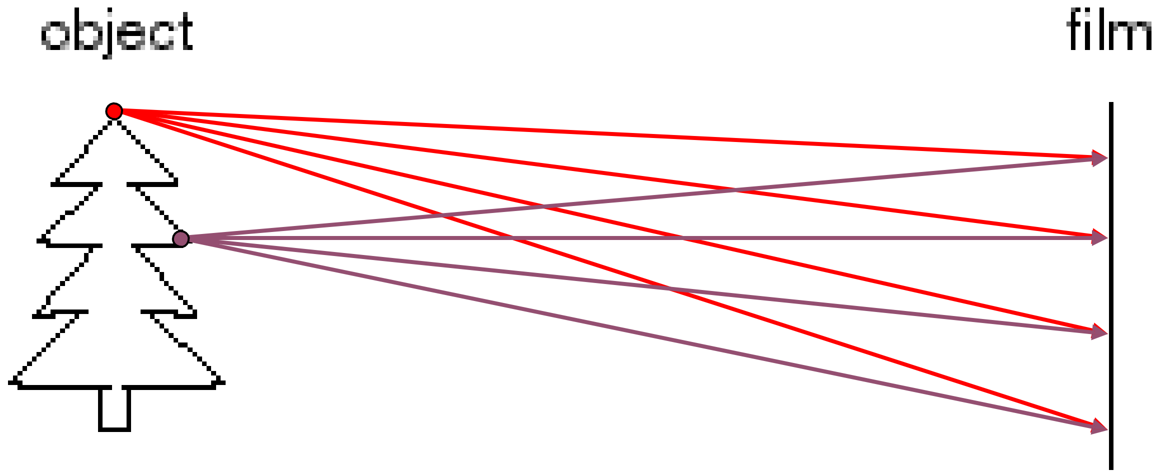


Let's design a camera

–Idea 1: put a piece of film in front of an object

Slide source: Seitz

Image formation

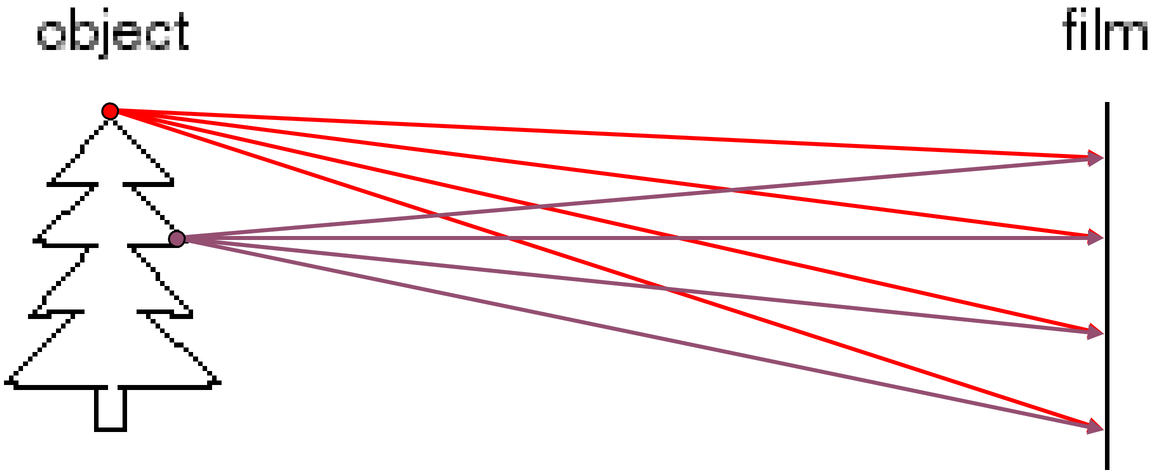


Let's design a camera

- Idea 1: put a piece of film in front of an object

Slide source: Seitz

Image formation



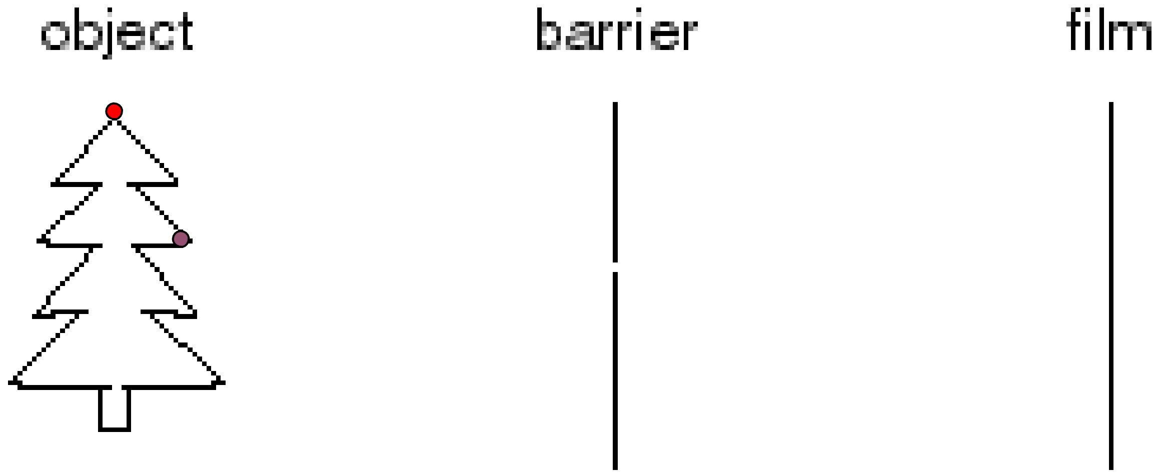
Do we get a reasonable image?

Let's design a camera

–Idea 1: put a piece of film in front of an object

Slide source: Seitz

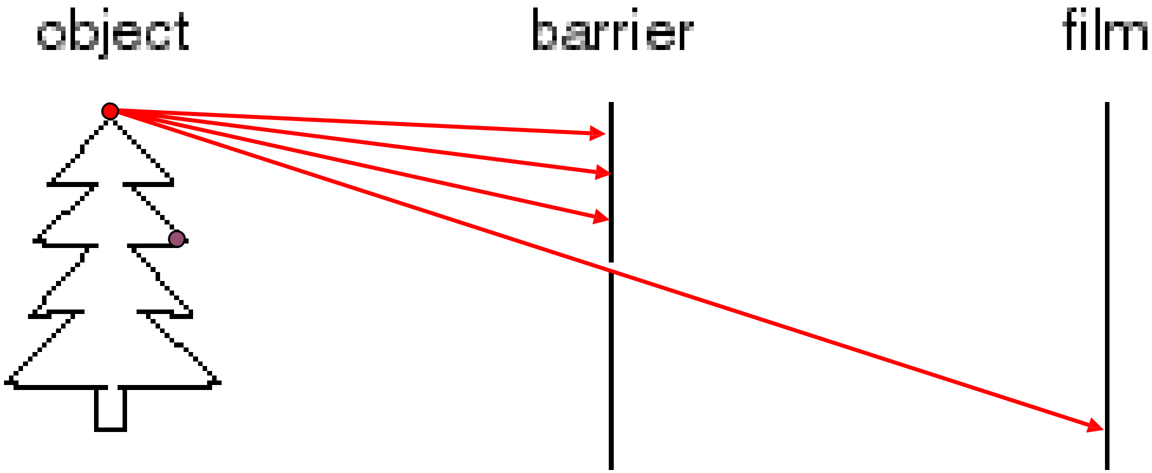
Pinhole camera



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

Slide source: Seitz

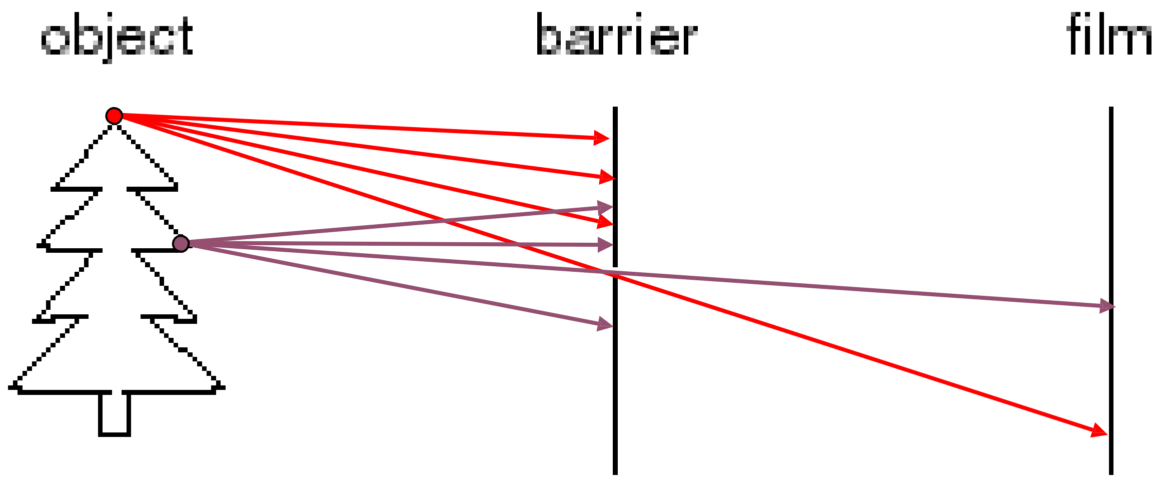
Pinhole camera



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

Slide source: Seitz

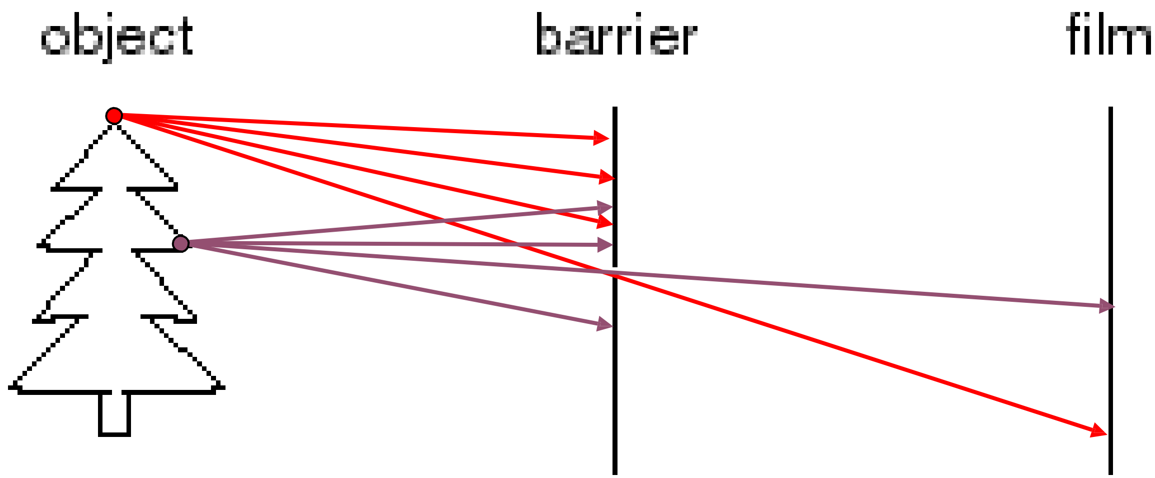
Pinhole camera



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

Slide source: Seitz

Pinhole camera

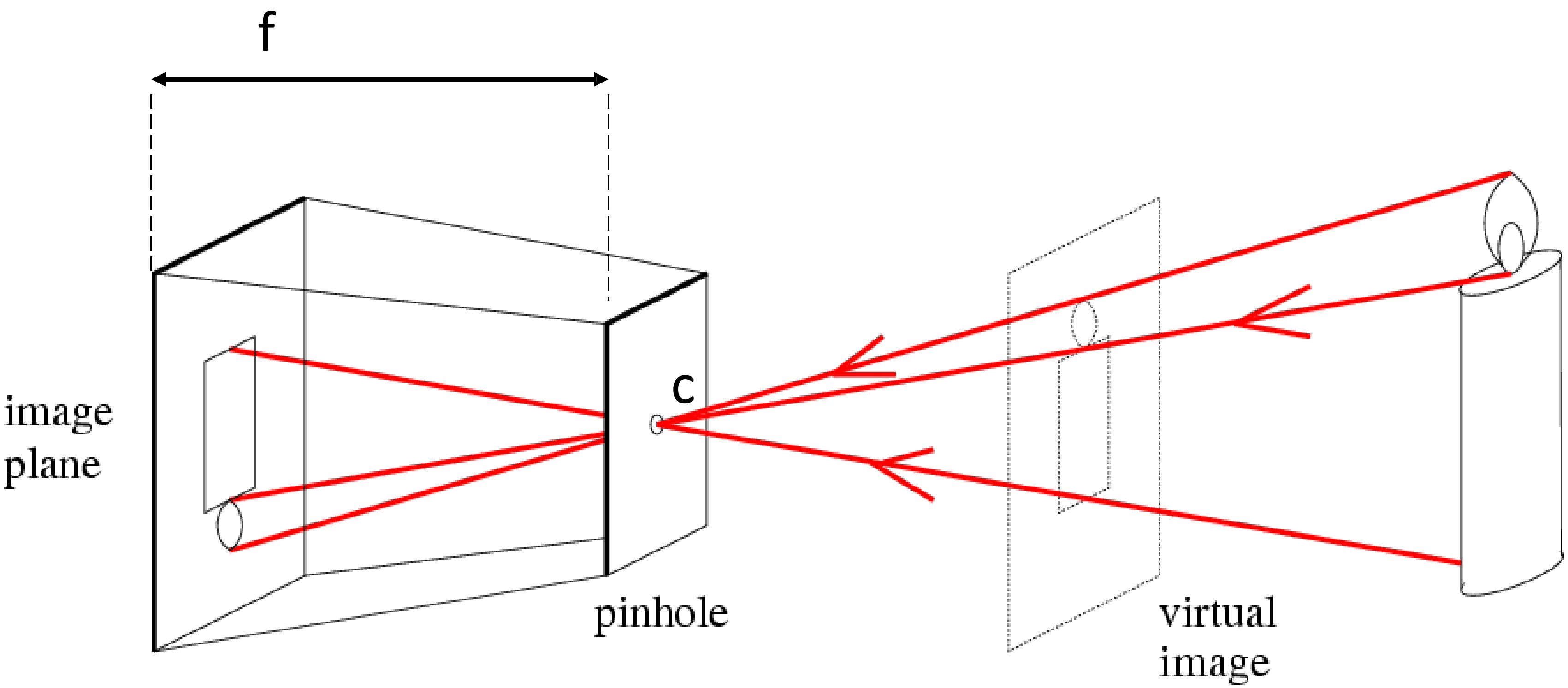


Idea 2: add a barrier to block off most of the rays

- 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: Alhazen, Iraq/Egypt (965 to 1039AD)

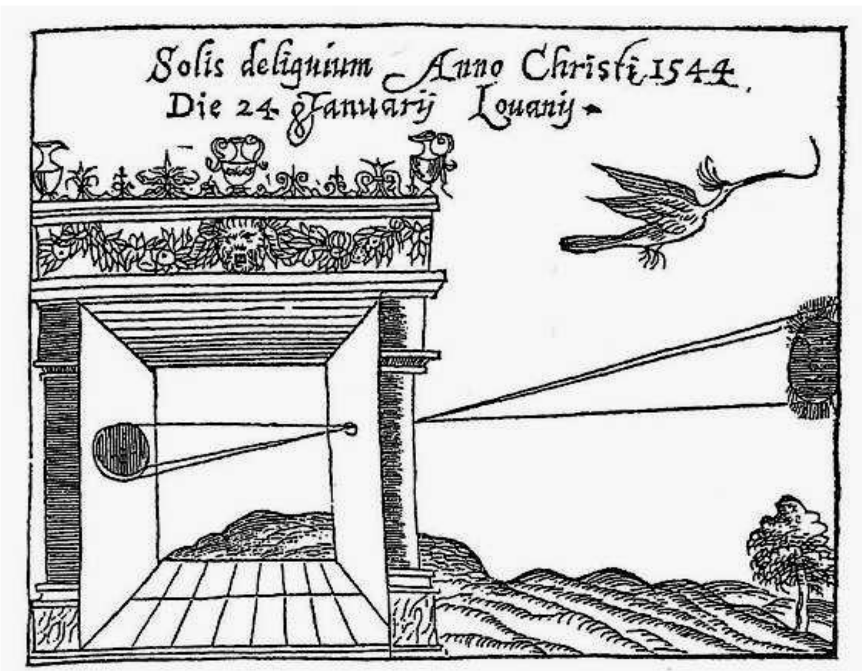


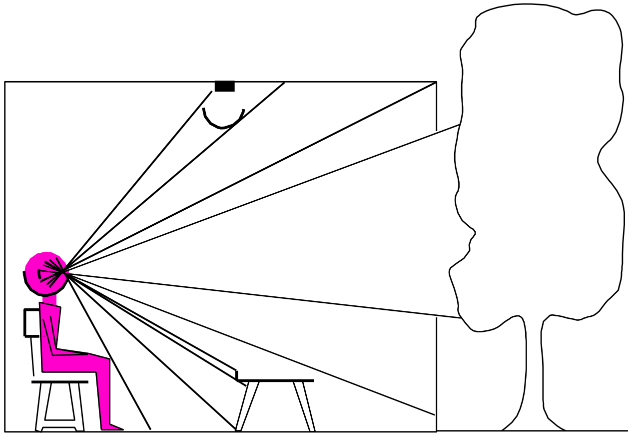
Illustration of Camera Obscura

Freestanding camera obscura at UNC Chapel Hill

Photo by Seth Ilys

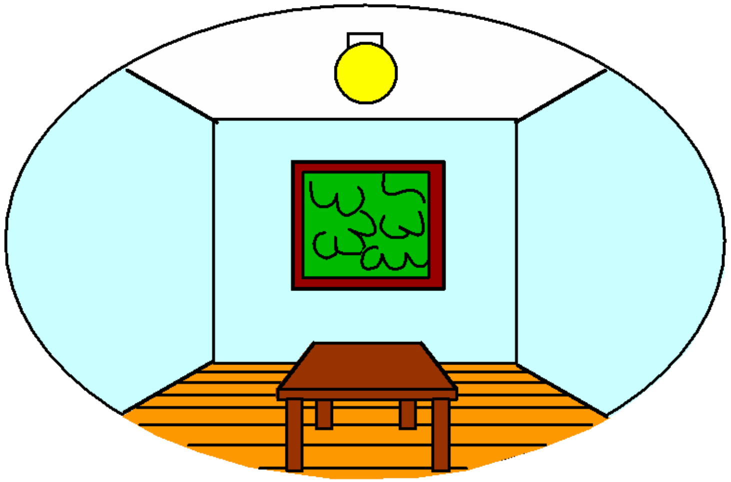
Dimensionality Reduction Machine (3D to 2D)

3D world



Point of observation

2D image



Slide source: Seitz

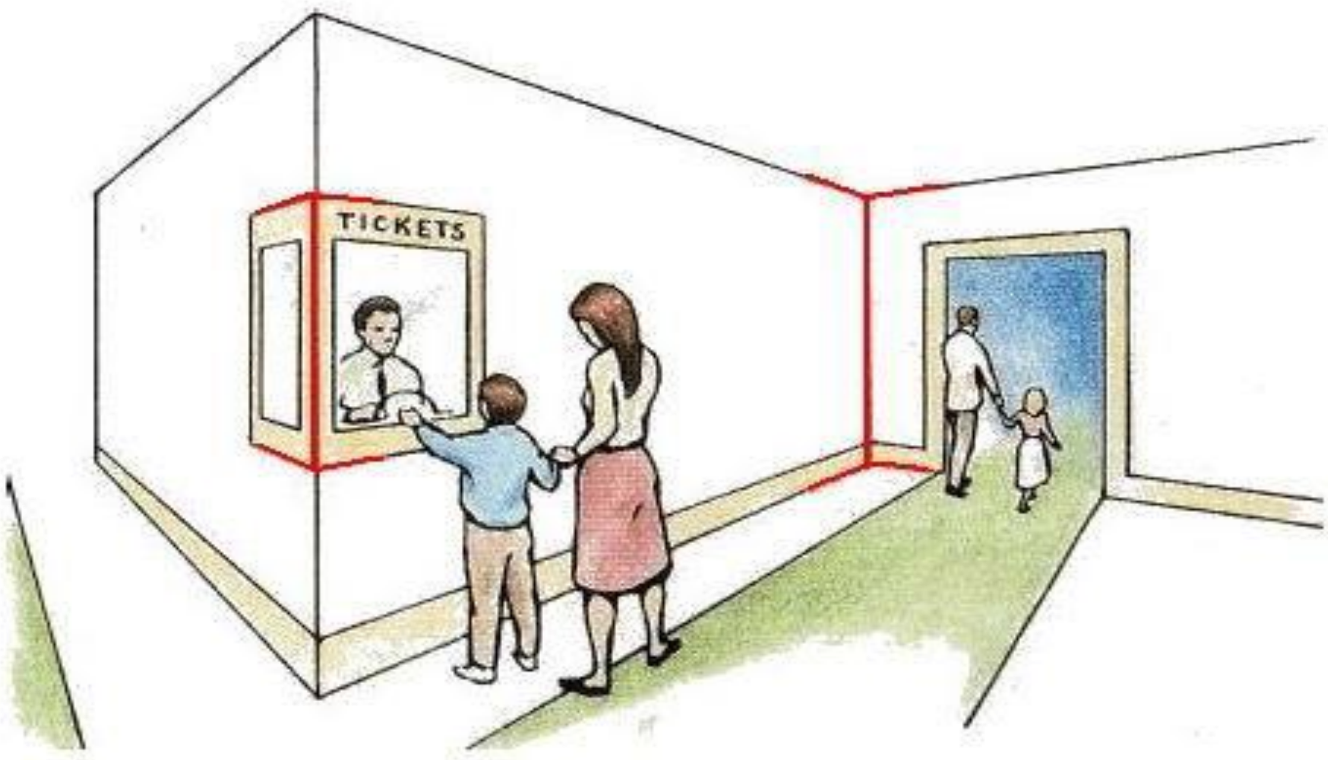
Projection can be tricky…

Slide source: Seitz

Projection can be tricky…

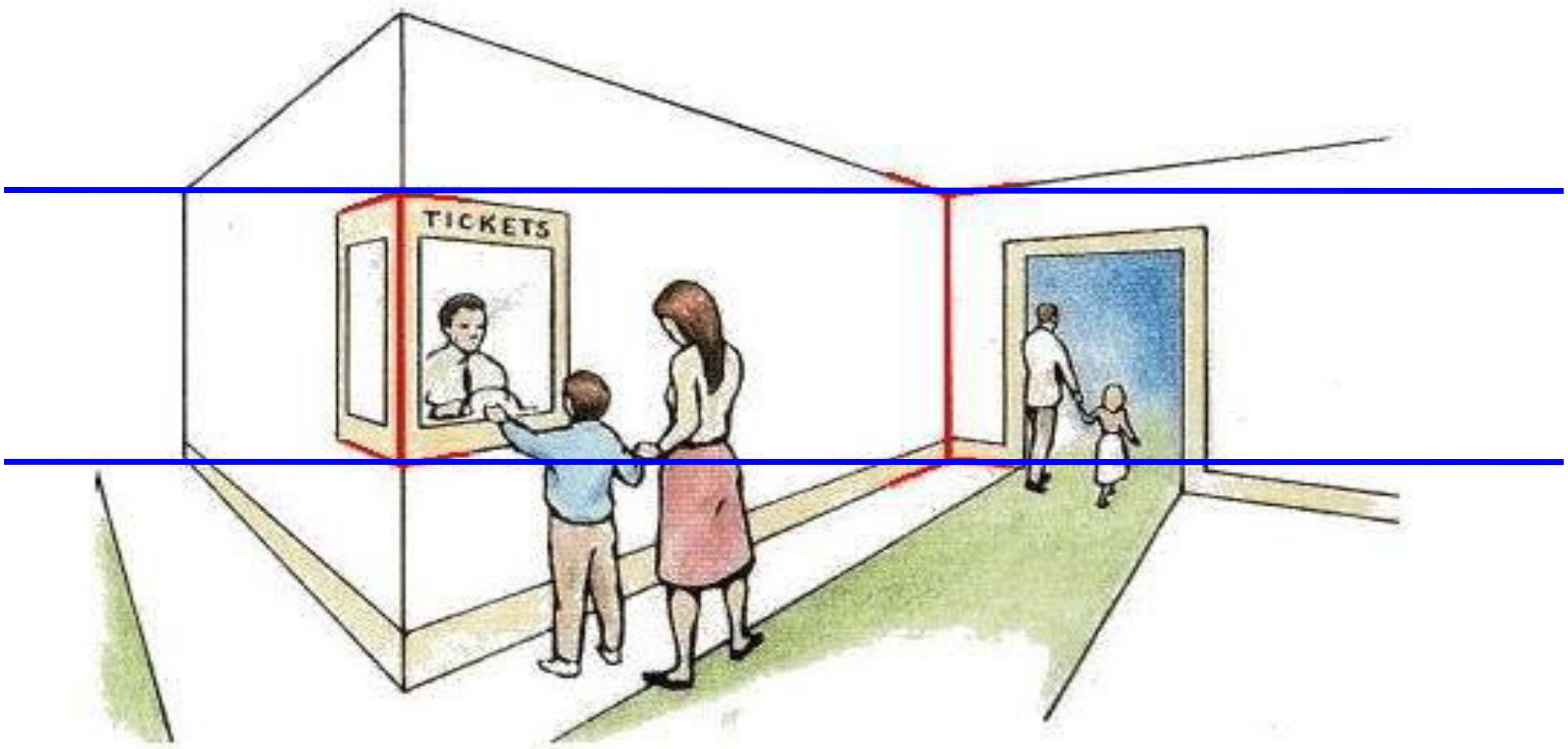
Making of 3D sidewalk art: http://www.youtube.com/watch?v=3SNYtd0Ayt0

Projection can be tricky…



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

Projection can be tricky…

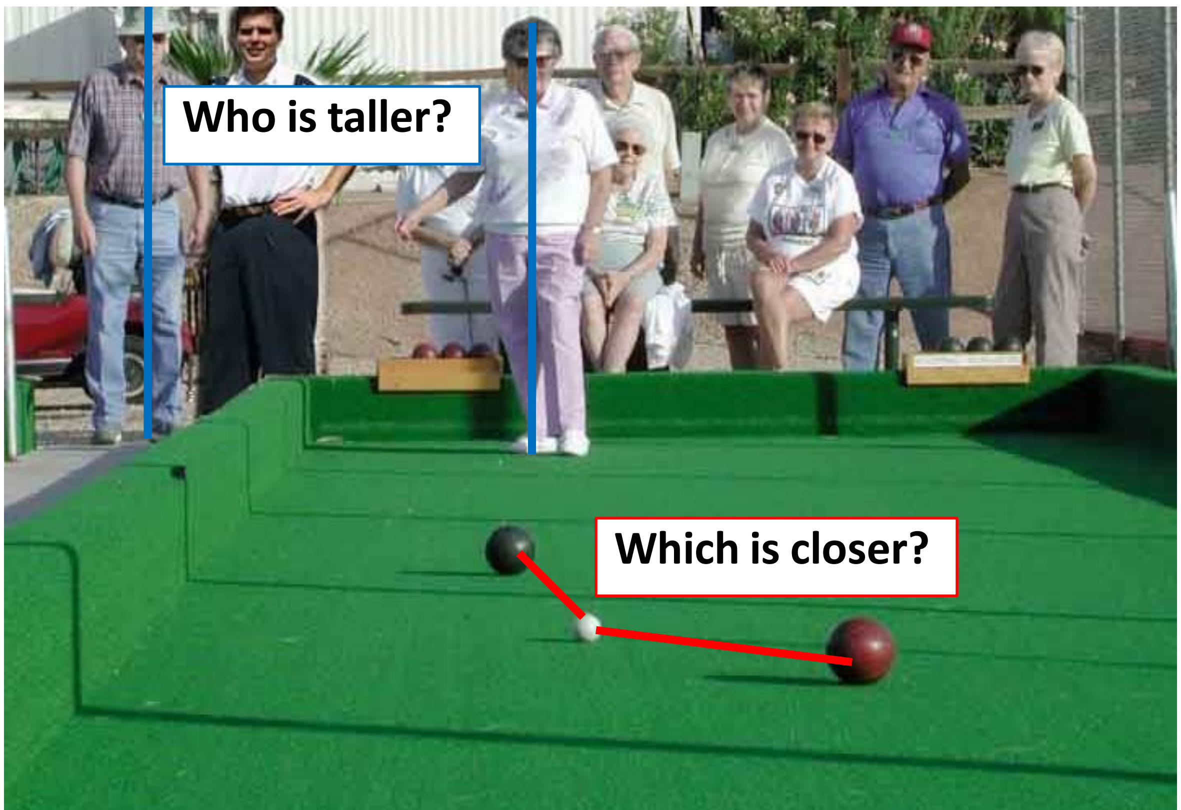


http://www.michaelbach.de/ot/sze_muelue/index.html

Projective Geometry

What is lost?

- Length



Length is not preserved

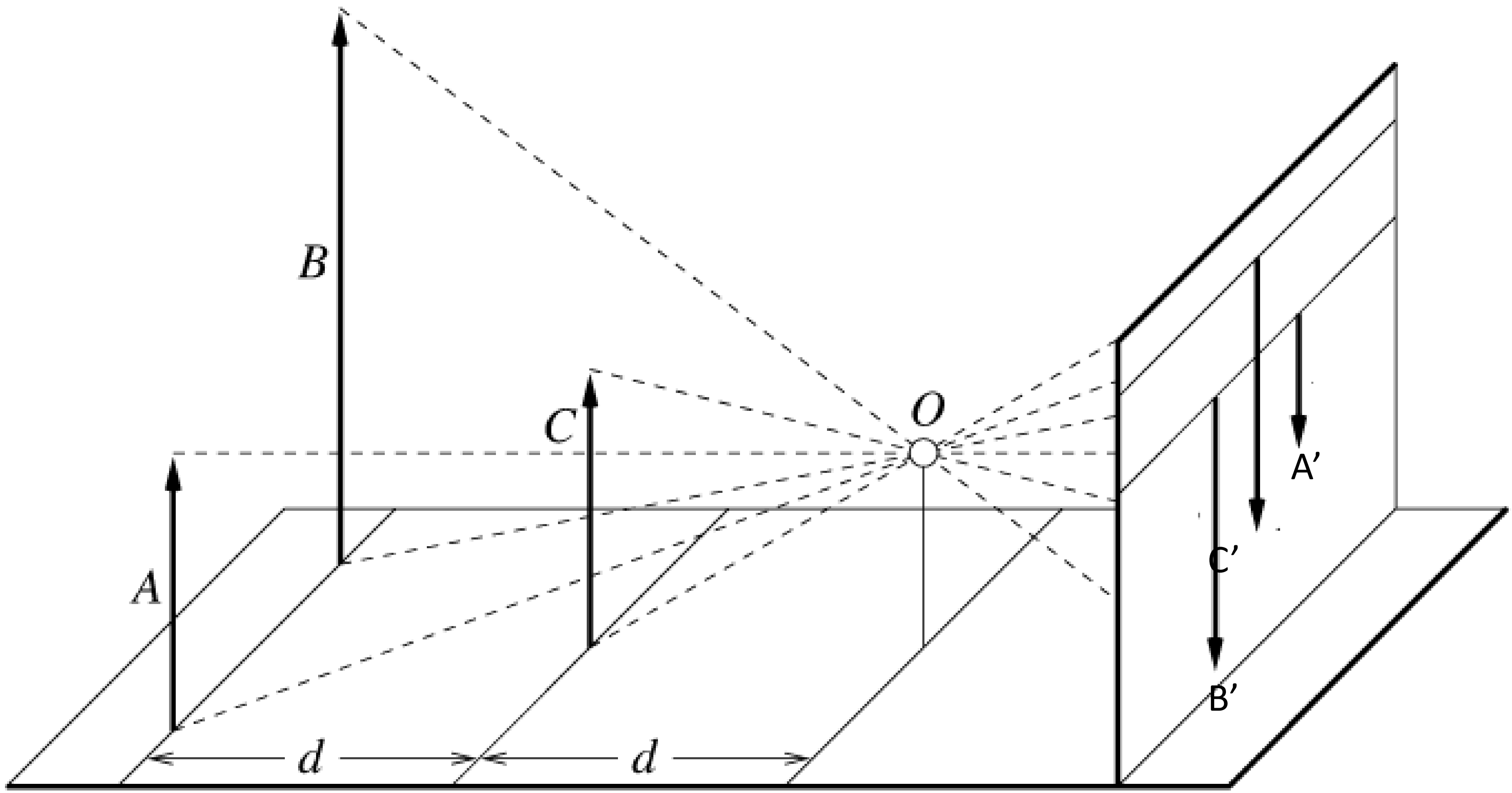
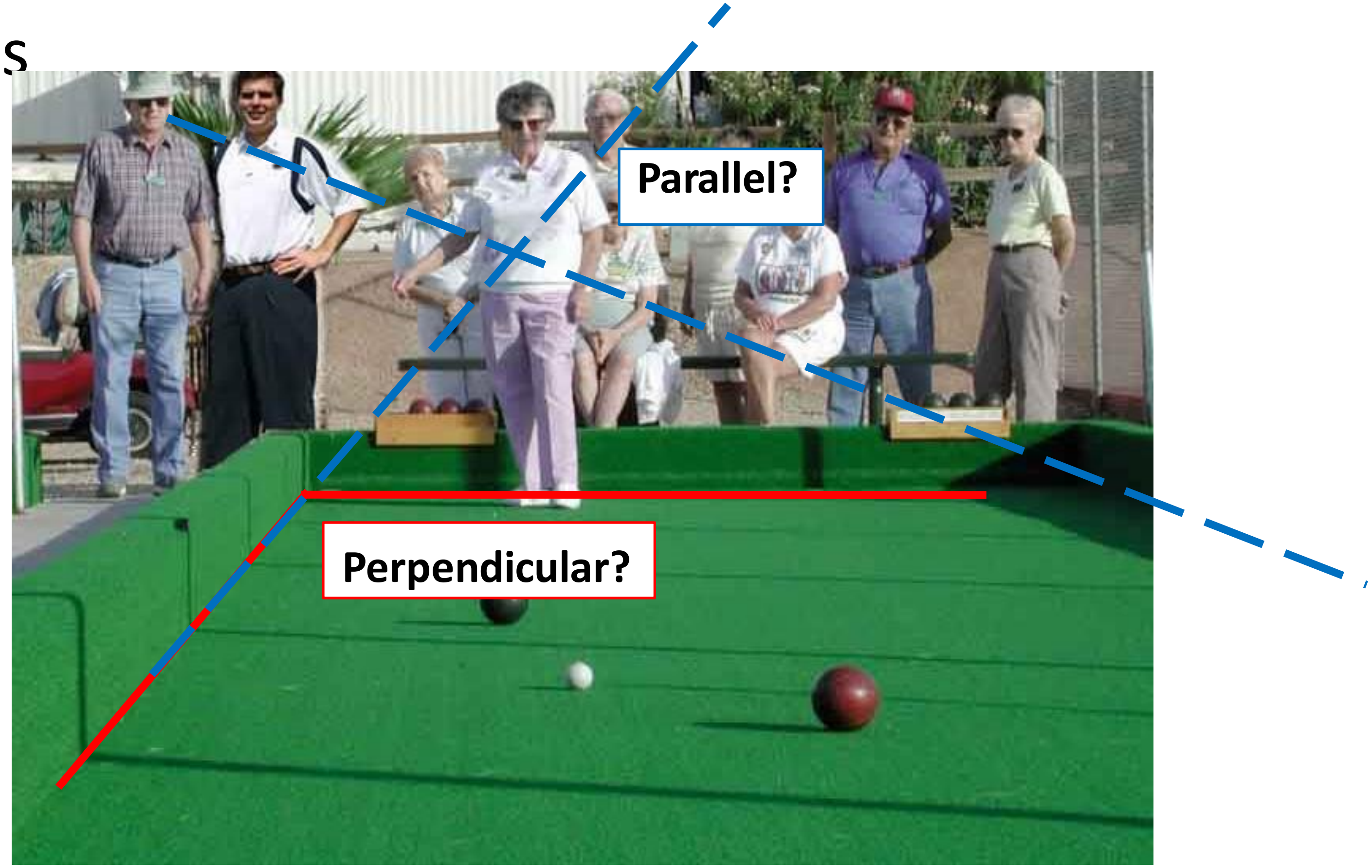


Figure by David Forsyth

Projective Geometry

What is lost?

- Length
- Angles



Projective Geometry

What is preserved?

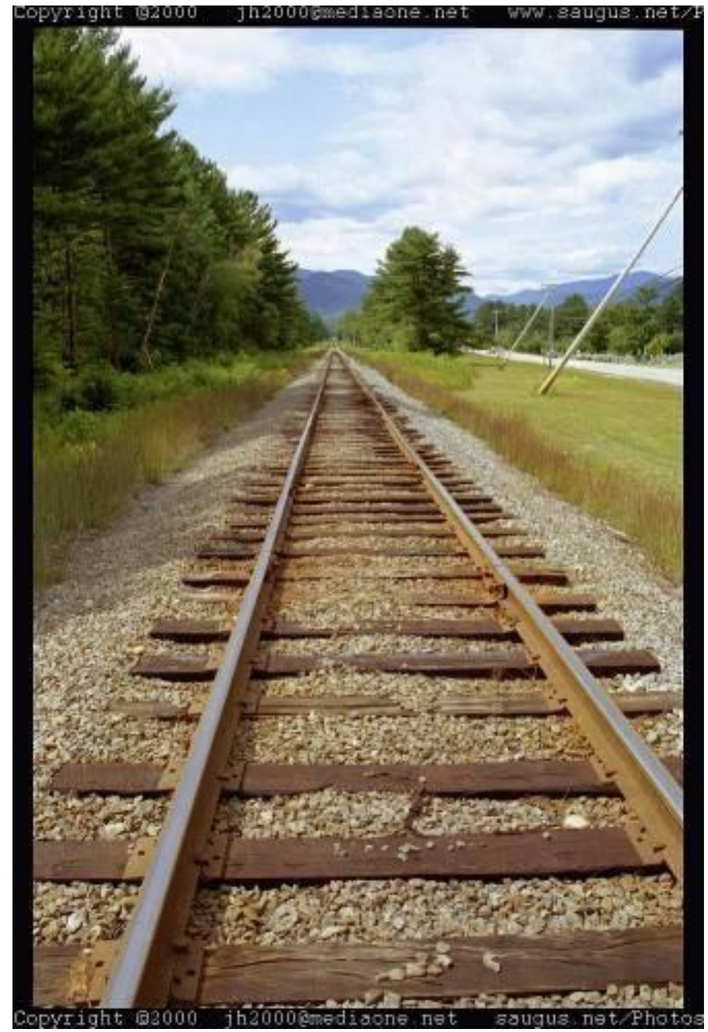
- Straight lines are still straight

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

Vanishing points and lines

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

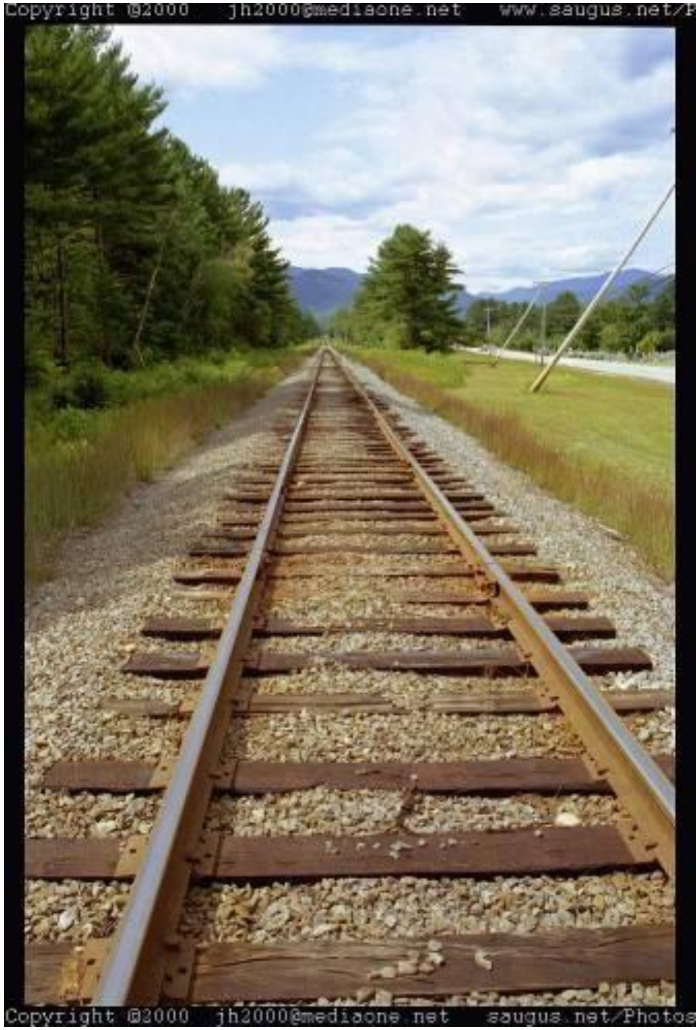


Vanishing points and lines

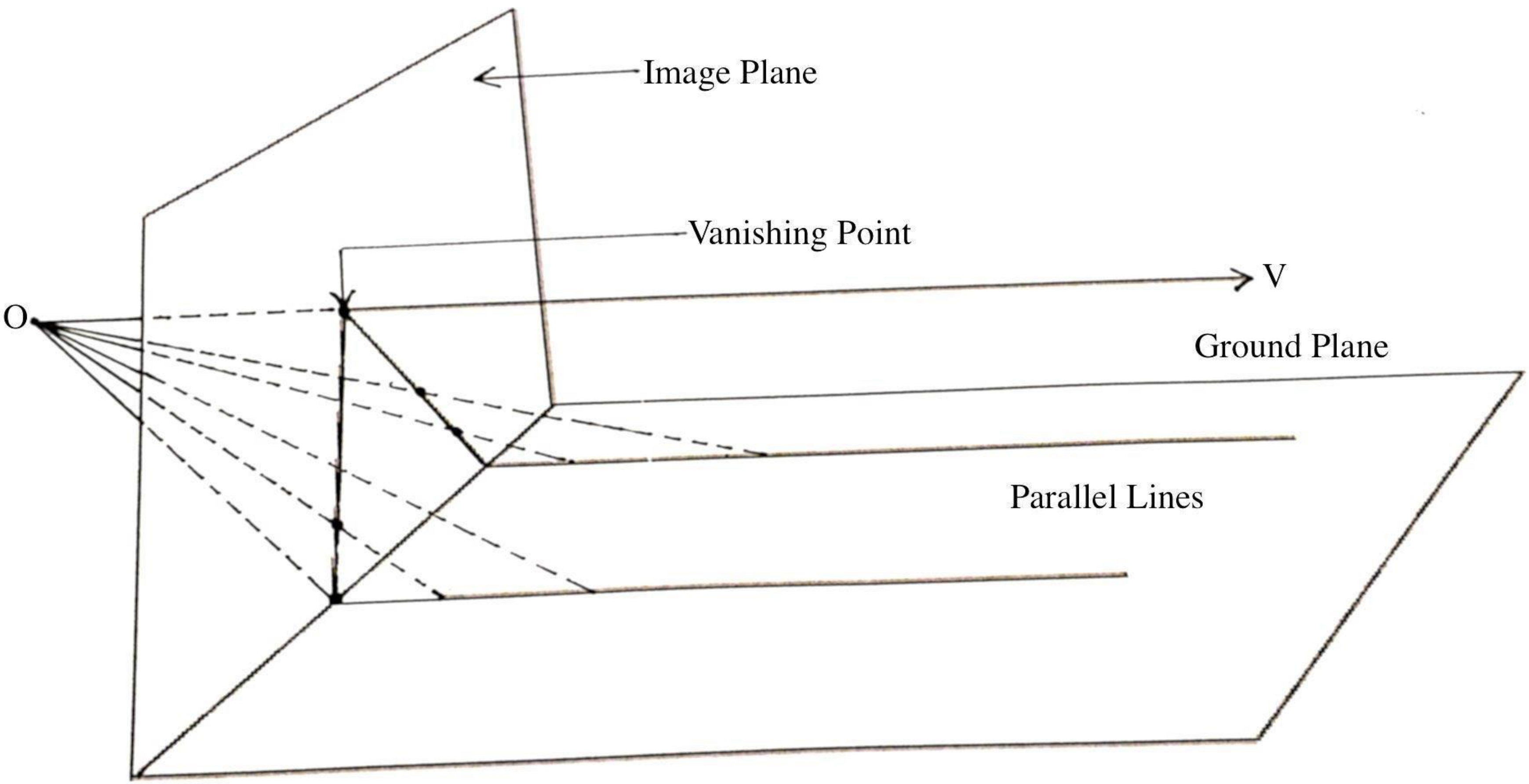
Parallel lines in the world intersect in the image at a “vanishing point”

•Each direction in space has its own vanishing point

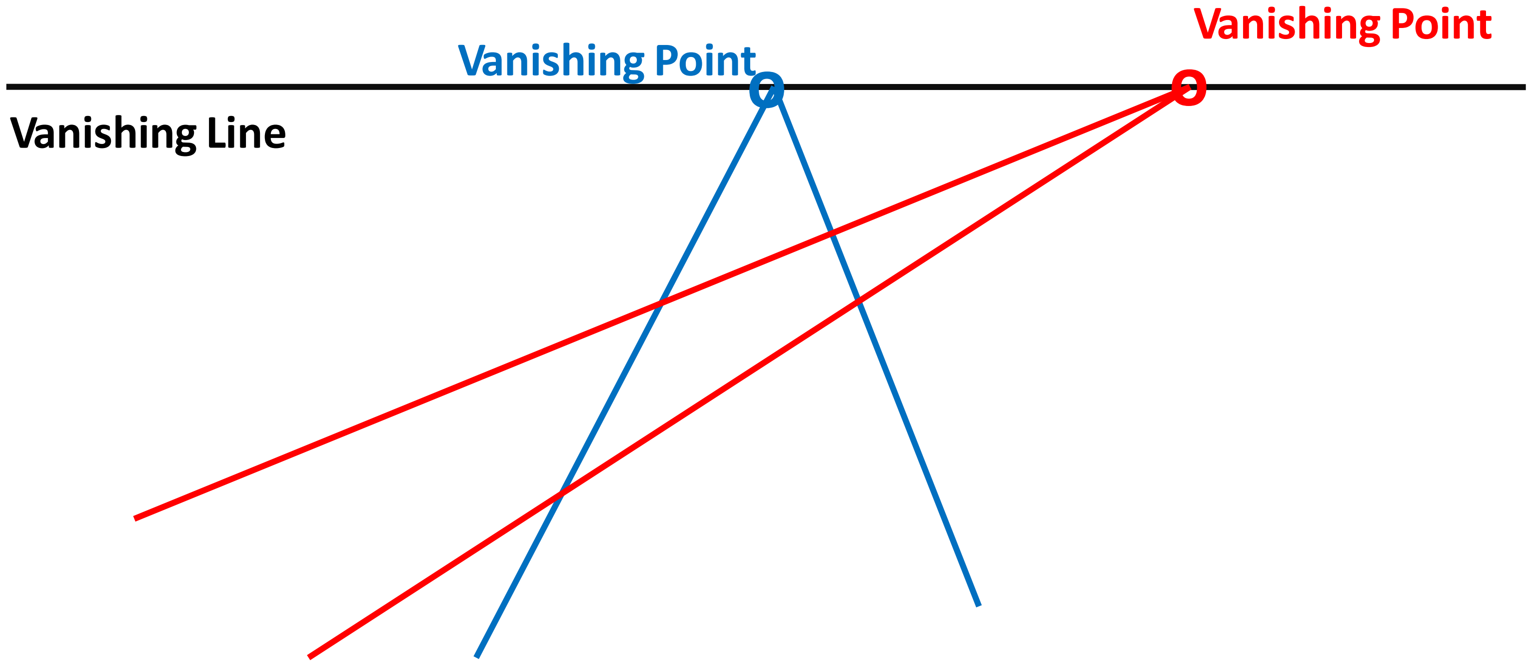
•But parallel lines to the image plane remain parallel



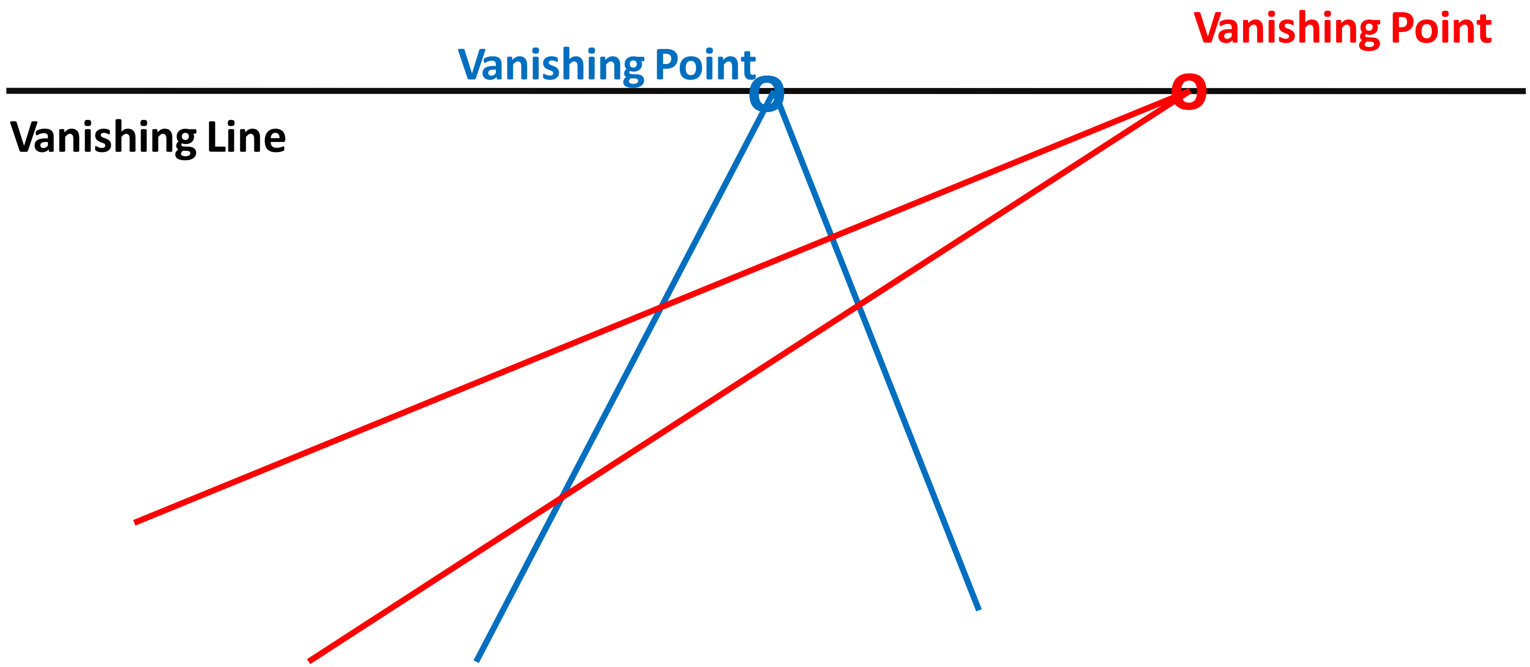
Vanishing points



Vanishing points and lines

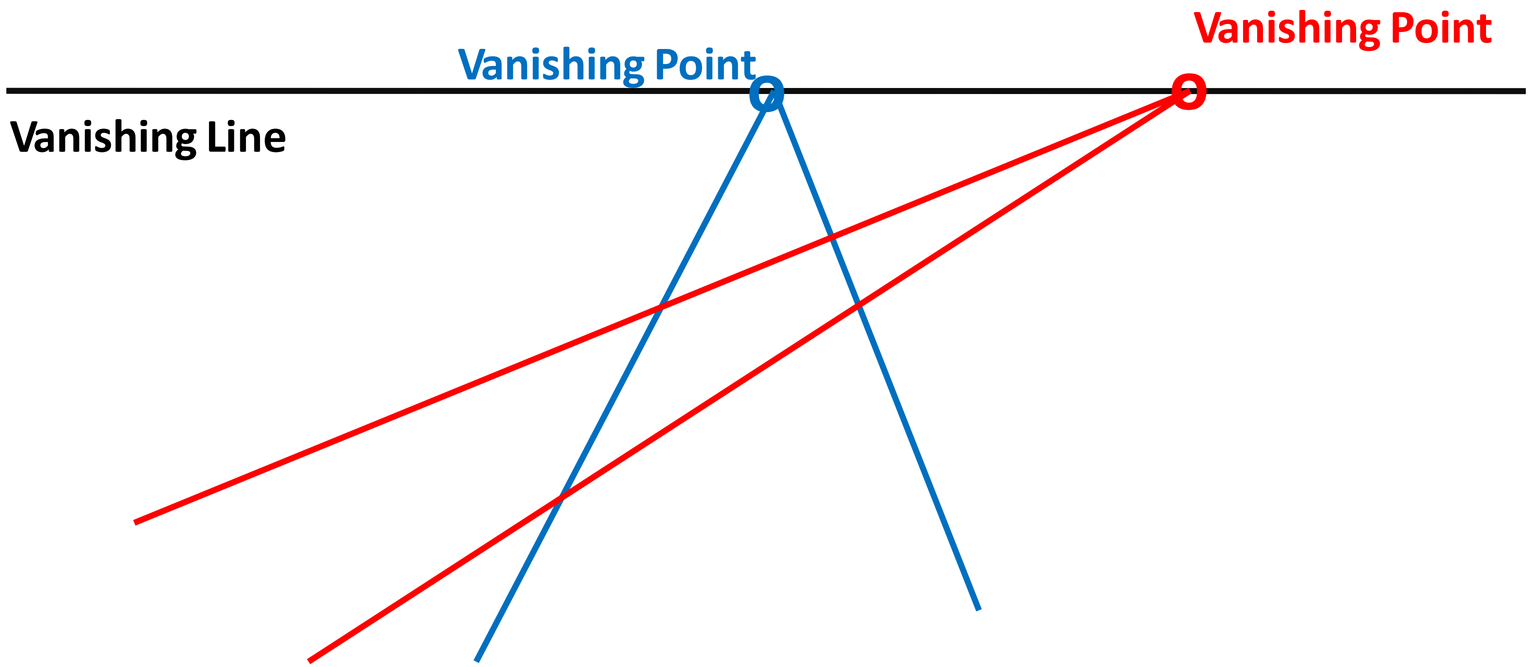


Vanishing points and lines



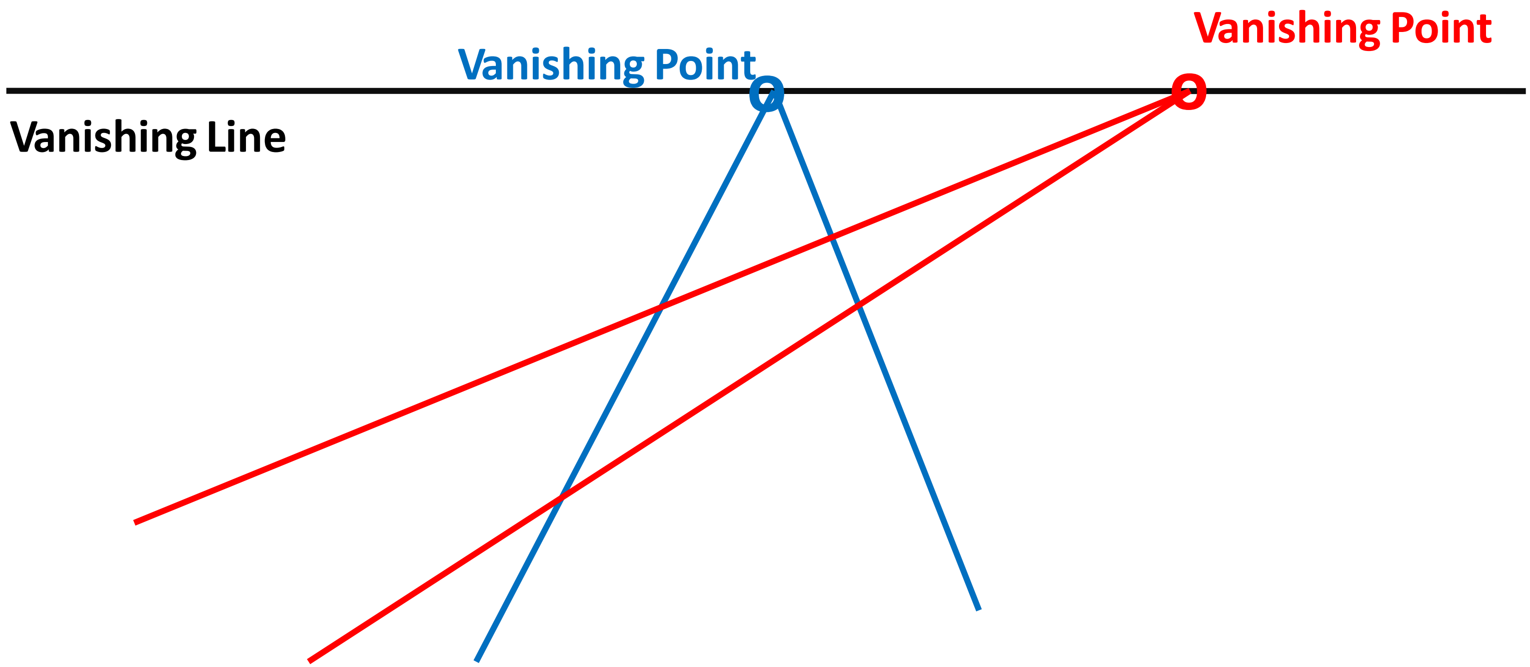
- The projections of parallel 3D lines intersect at a **vanishing point**
- The projection of parallel 3D planes intersect at a **vanishing line**

Vanishing points and lines



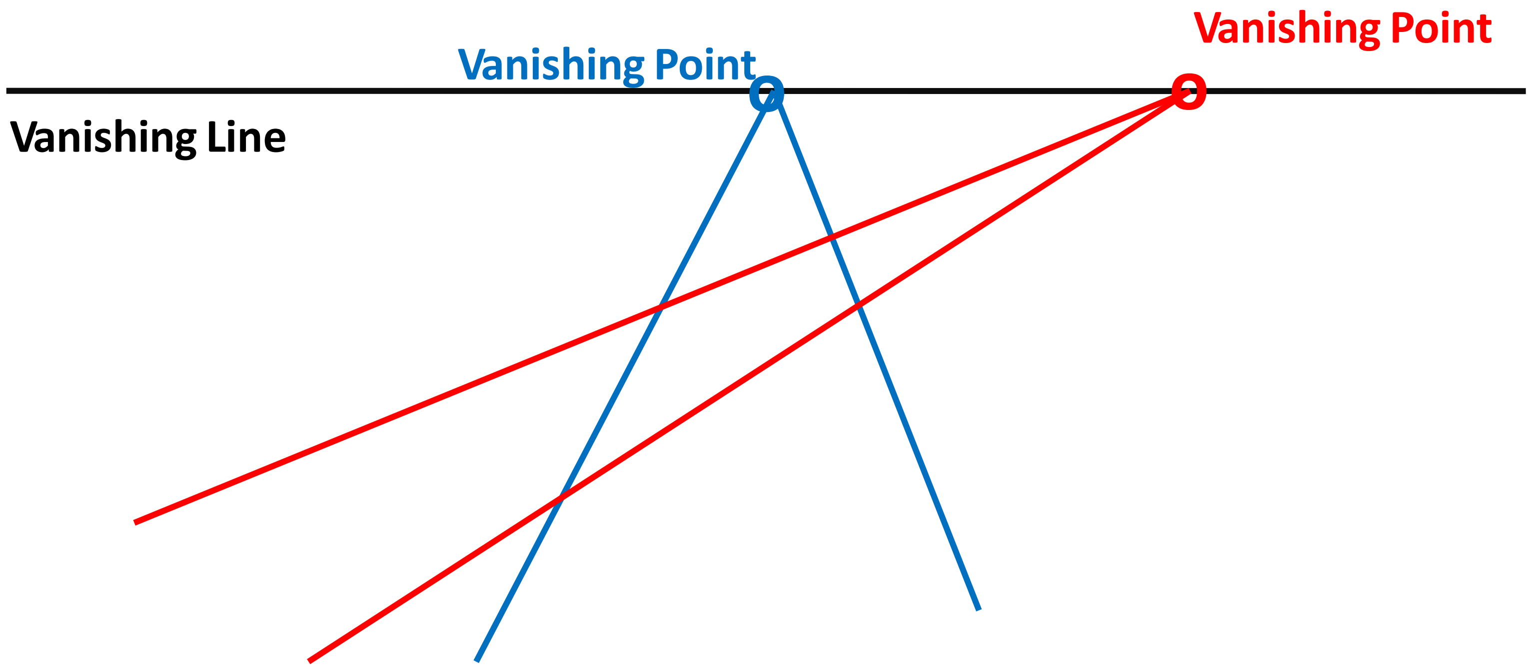
- The projections of parallel 3D lines intersect at a **vanishing point**
- The projection of parallel 3D planes intersect at a **vanishing line**
- If a set of parallel 3D lines are also parallel to a particular plane, their vanishing point will lie on the vanishing line of the plane

Vanishing points and lines



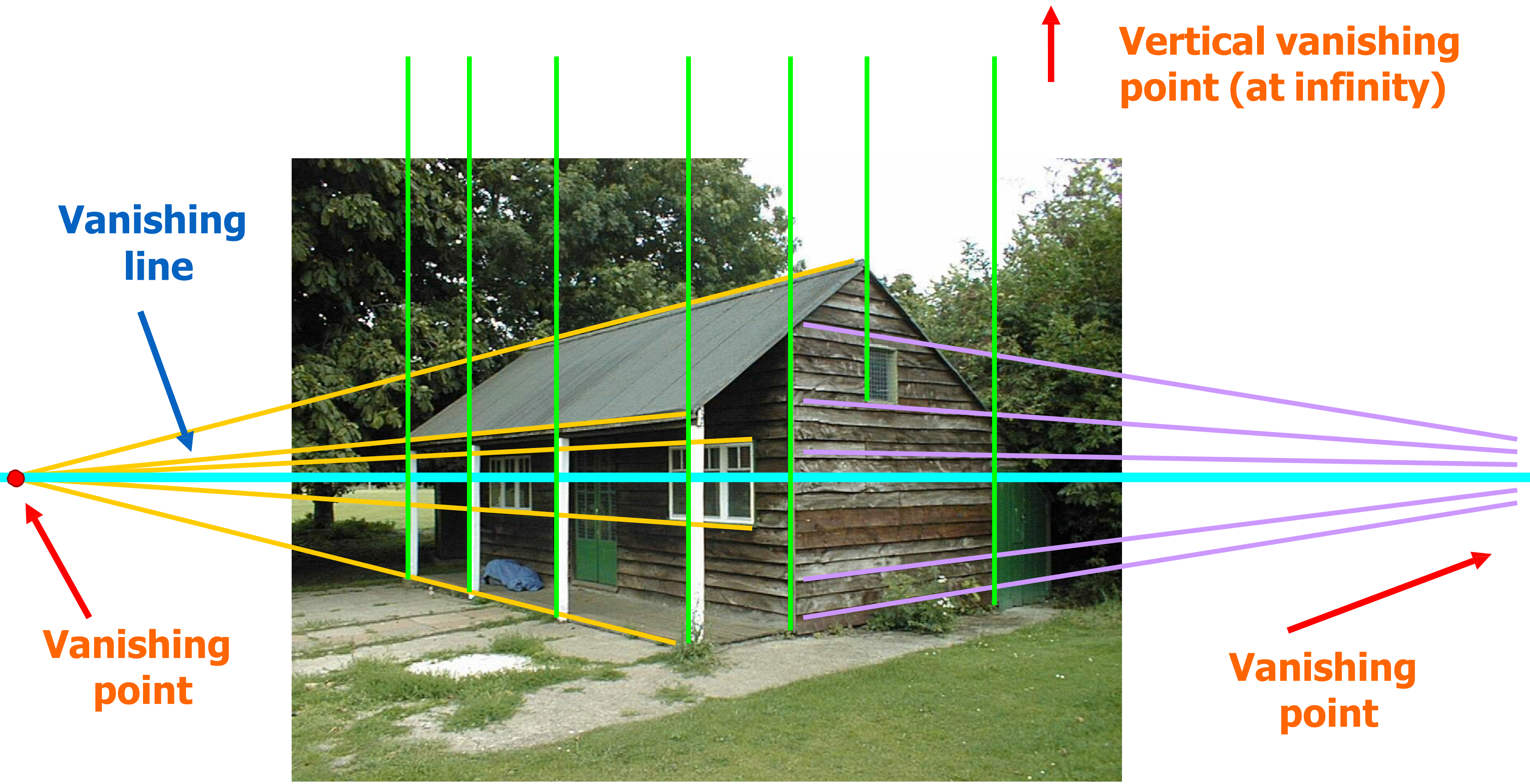
- The projections of parallel 3D lines intersect at a **vanishing point**
- The projection of parallel 3D planes intersect at a **vanishing line**
- If a set of parallel 3D lines are also parallel to a particular plane, their vanishing point will lie on the vanishing line of the plane
- Not all lines that intersect are parallel

Vanishing points and lines



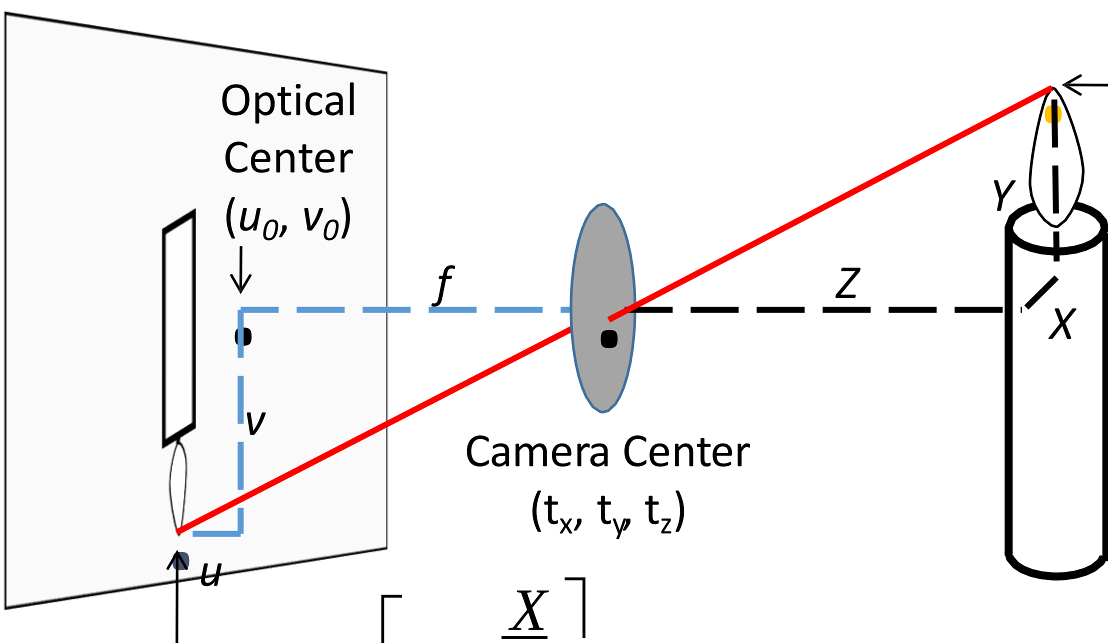
- The projections of parallel 3D lines intersect at a **vanishing point**
- The projection of parallel 3D planes intersect at a **vanishing line**
- If a set of parallel 3D lines are also parallel to a particular plane, their vanishing point will lie on the vanishing line of the plane
- Not all lines that intersect are parallel
- Vanishing point <-> 3D direction of a line

Vanishing points and lines



Slide from Efros, Photo from Criminisi

Projection: world coordinates → image coordinates



$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f\dfrac{X}{Z} \\ -f\dfrac{Y}{Z} \end{bmatrix}$$

Homogeneous coordinates

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

Homogeneous coordinates

Converting to homogeneous coordinates

$$(x, y) \Rightarrow \left[\begin{array}{c} x \\ y \\ 1 \end{array}\right]$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \left[\begin{array}{c} x \\ y \\ z \\ 1 \end{array}\right]$$

homogeneous scene coordinates

Converting from homogeneous coordinates

$$\left[\begin{array}{c} x \\ y \\ w \end{array}\right] \Rightarrow (x/w, y/w)$$

$$\left[\begin{array}{c} x \\ y \\ z \\ w \end{array}\right] \Rightarrow (x/w, y/w, z/w)$$

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

Cartesian Coordinates

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

Cartesian Coordinates

Point in Cartesian is ray in Homogeneous

Basic geometry in homogeneous coordinates

- Append 1 to pixel coordinate to get homogeneous coordinate

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Basic geometry in homogeneous coordinates

- Append 1 to pixel coordinate to get homogeneous coordinate

$$P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Line equation: $au + bv + c = 0$

$$line = [a\ b\ c]^\top$$

$$line^\top p = 0$$

Basic geometry in homogeneous coordinates

- Append 1 to pixel coordinate to get homogeneous coordinate

$$P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Line equation:

$$au + bv + c = 0$$
$$line = [a\ b\ c]^\top$$

$$line^\top p = 0$$

- Line given by cross product of two points

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

Basic geometry in homogeneous coordinates

- Append 1 to pixel coordinate to get homogeneous coordinate

$$P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Line equation: $au + bv + c = 0$

$$line = [a\ b\ c]^\top$$

$$line^\top p = 0$$

- 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

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

Basic geometry in homogeneous coordinates

- Append 1 to pixel coordinate to get homogeneous coordinate

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Line equation: $au + bv + c = 0$

$$line = [a\ b\ c]^\top \qquad line^\top p = 0$$

- 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

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

- Three points lies on the same line

$$p_k^\top (p_i \times p_j) = 0$$

Basic geometry in homogeneous coordinates

- Append 1 to pixel coordinate to get homogeneous coordinate
$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Line equation: $au + bv + c = 0$
$$line = [a\ b\ c]^\top$$
$$line^\top p = 0$$

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

- Three points lies on the same line
$$p_k^\top (p_i \times p_j) = 0$$

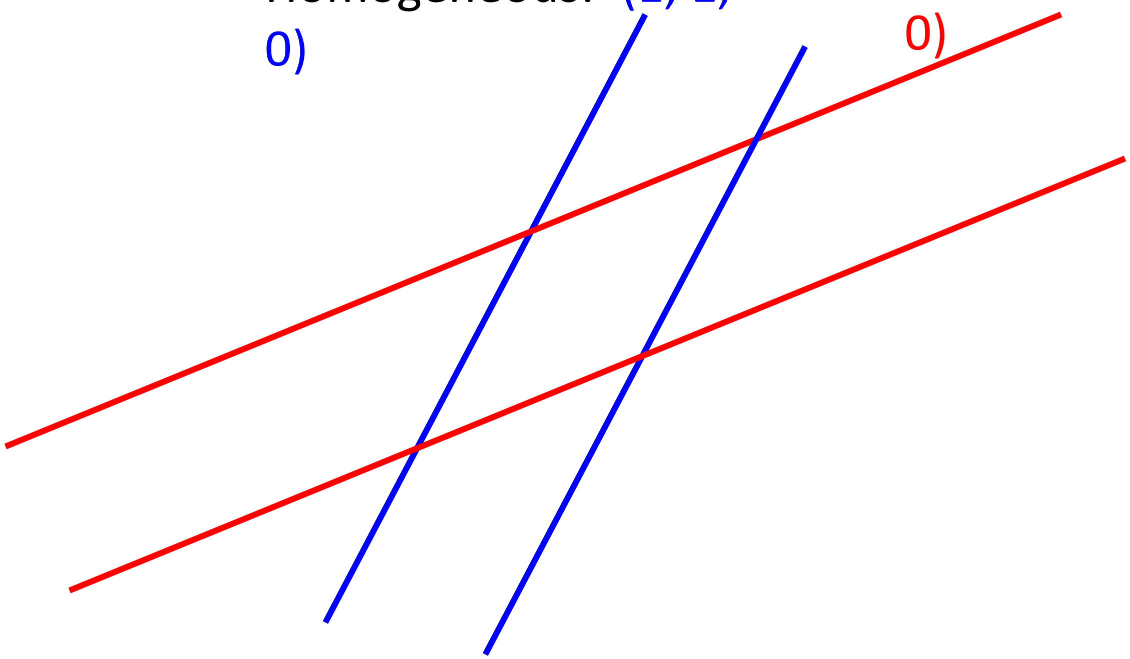
- Three lines intersect at the same point
$$line_k^\top (line_i \times line_j) = 0$$

Another problem solved by homogeneous coordinates

Intersection of parallel lines

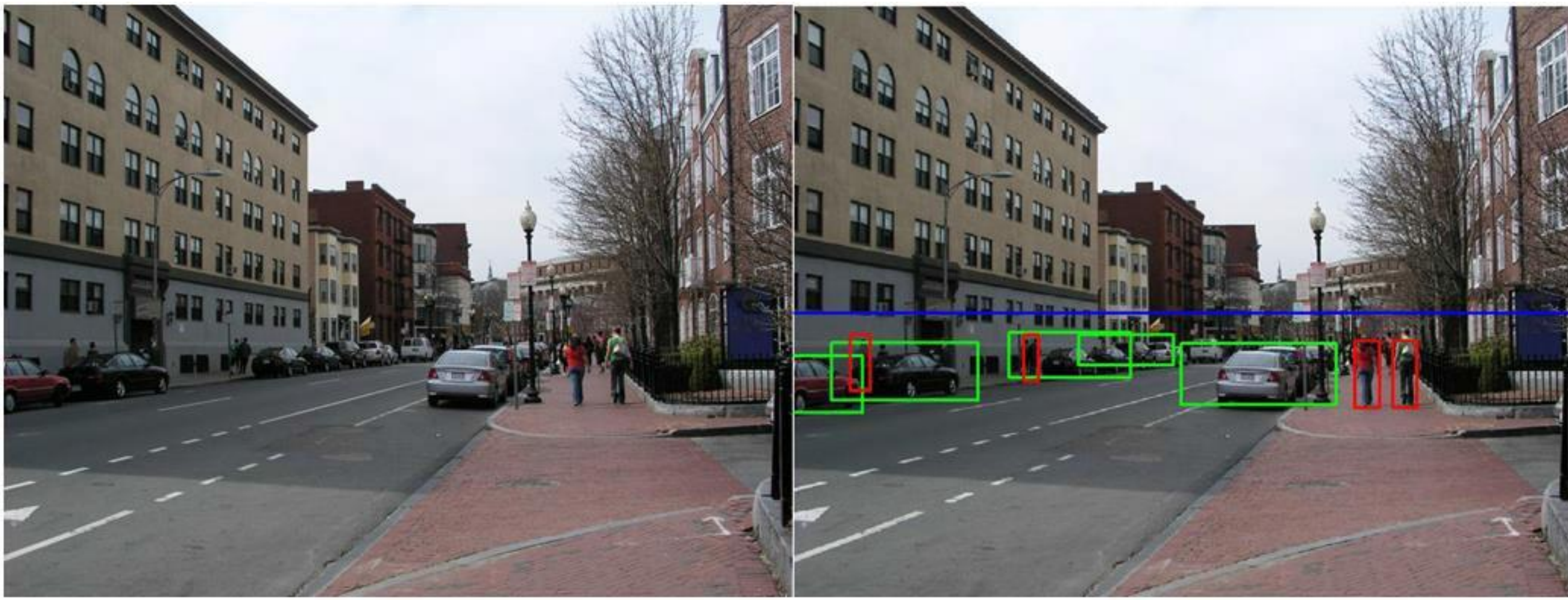
Cartesian: (Inf, Inf)
Homogeneous: (1, 1, 0)

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



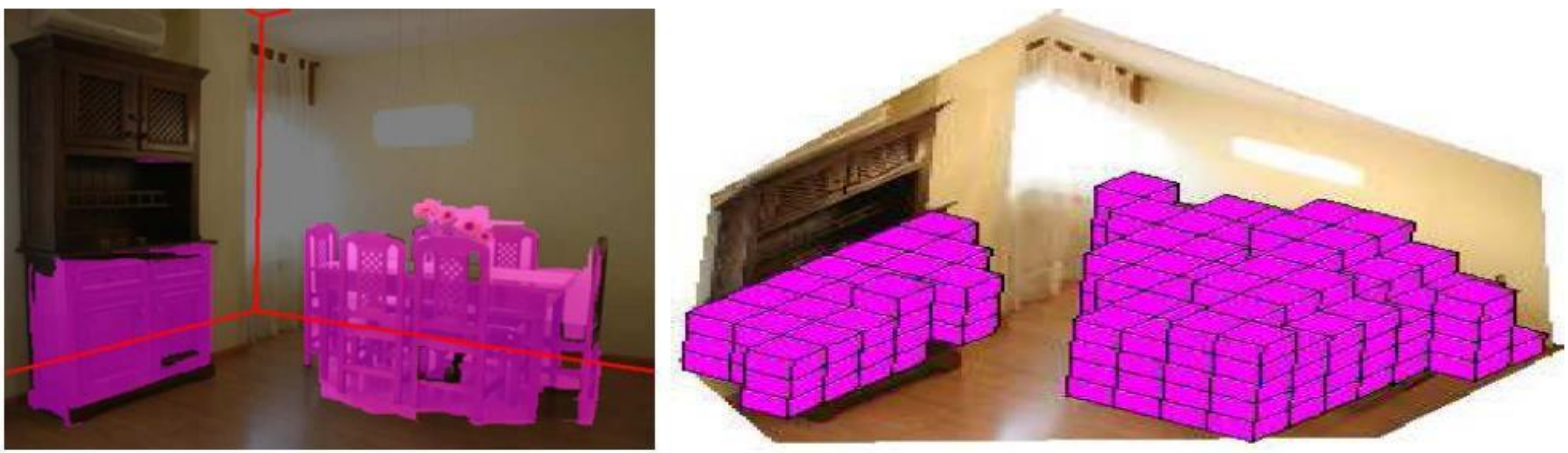
Applications

Object Recognition (CVPR 2006)



Applications

Getting spatial layout in indoor scenes (ICCV 2009)



Applications

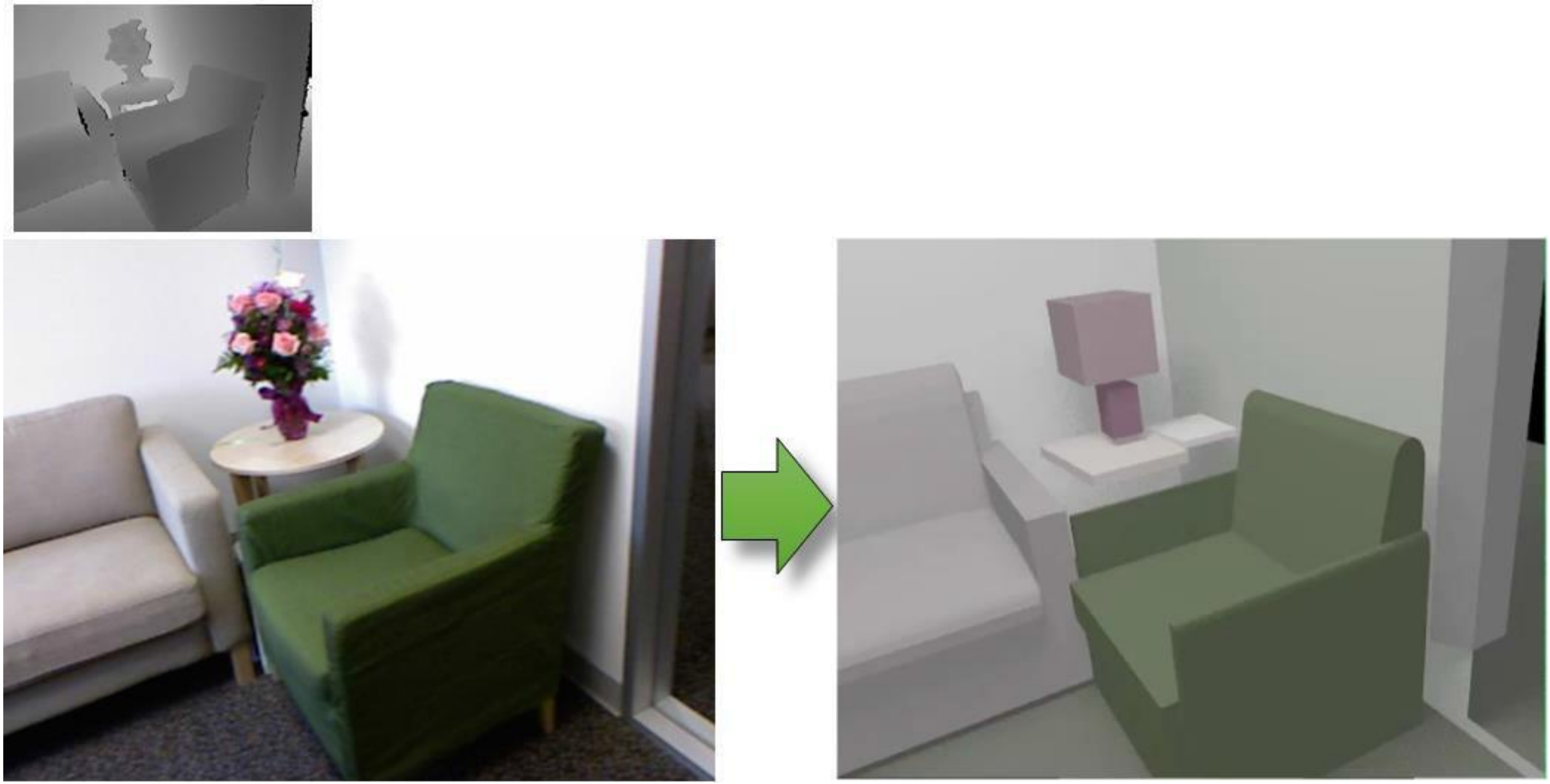
Inserting synthetic objects into images: http://vimeo.com/28962540

Applications

Inserting synthetic objects into images: http://vimeo.com/28962540

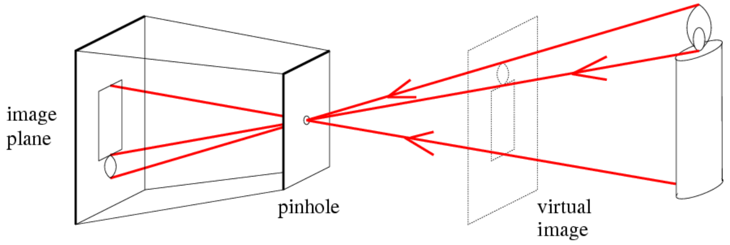
Applications

Creating detailed and complete 3D scene models from a single view



Things to remember

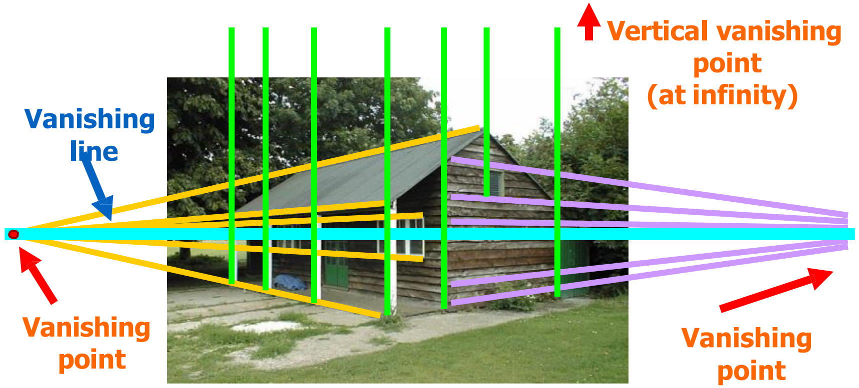
- Pinhole camera model



- Homogeneous coordinates

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

- Vanishing points and vanishing lines



Acknowledgements

- Thanks to the following researchers for making their teaching/research material online
  - Forsyth
  - Steve Seitz
  - Noah Snavely
  - J.B. Huang
  - Derek Hoiem
  - J. Hays
  - J. Johnson
  - R. Girshick
  - S. Lazebnik
  - K. Grauman
  - Antonio Torralba
  - Rob Fergus
  - Leibe
  - And many more ……….

Next class
Projection Matrix and Camera Calibration