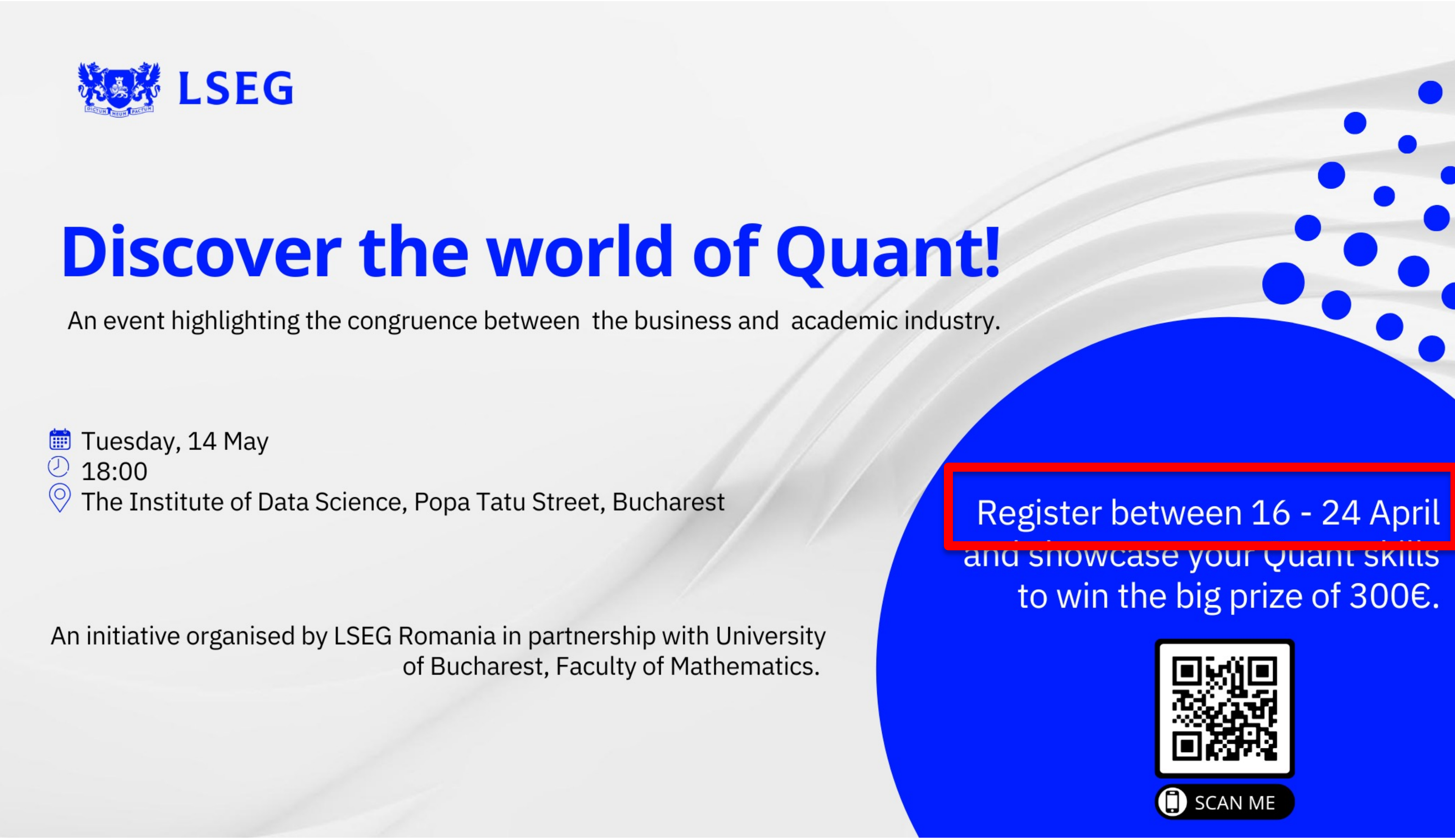
Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2023-2024

LSEG Challenge



LSEG Challenge

Administrative

Next week, 30th of April we don’t do the lecture (I’m away)

Project 1 – fake test added

- evaluation
  - code
    - evaluate_submission.py
    - fake_test_gt
  - fake_test
    - 1_01.jpg
    - 1_02.jpg
    - 1_03.jpg
    - 1_04.jpg
    - 1_05.jpg
    - 1_06.jpg
    - 1_07.jpg
    - 1_08.jpg
    - 1_09.jpg
    - 1_10.jpg
    - 1_11.jpg
    - 1_12.jpg
    - 1_13.jpg
    - 1_14.jpg
    - 1_15.jpg
    - 1_16.jpg
    - 1_17.jpg
    - 1_18.jpg
    - 1_19.jpg
    - 1_20.jpg
    - 1_21.jpg

Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection, template matching

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo object instance recognition

4. **Object Recognition: high – level vision**

   Object classification, object detection, part based models, bovw models

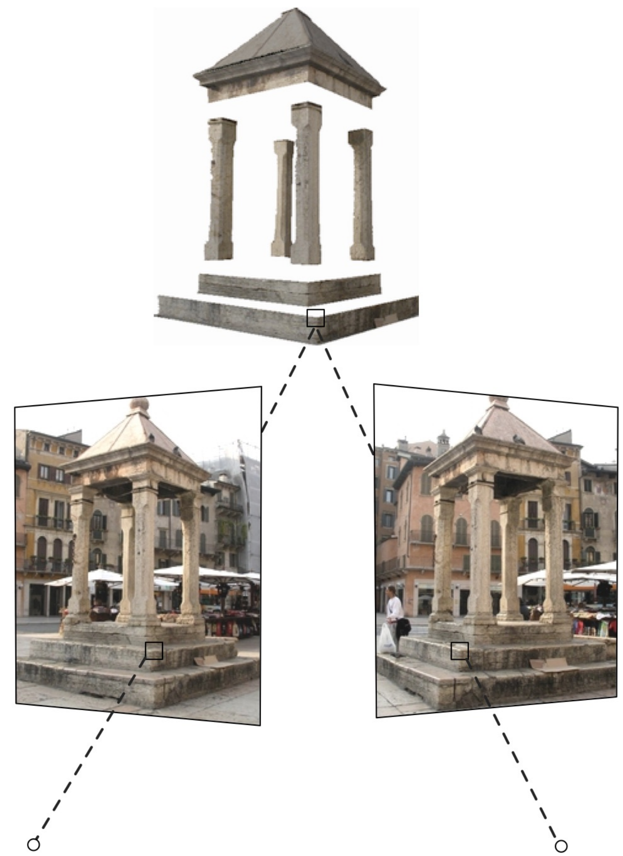
5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

Goal: Recovery of 3D structure

- When certain assumptions hold, we can recover structure from a single view

- In general, we need *multi-view geometry*



- But first, we need to understand the geometry of a single camera…

Image formation

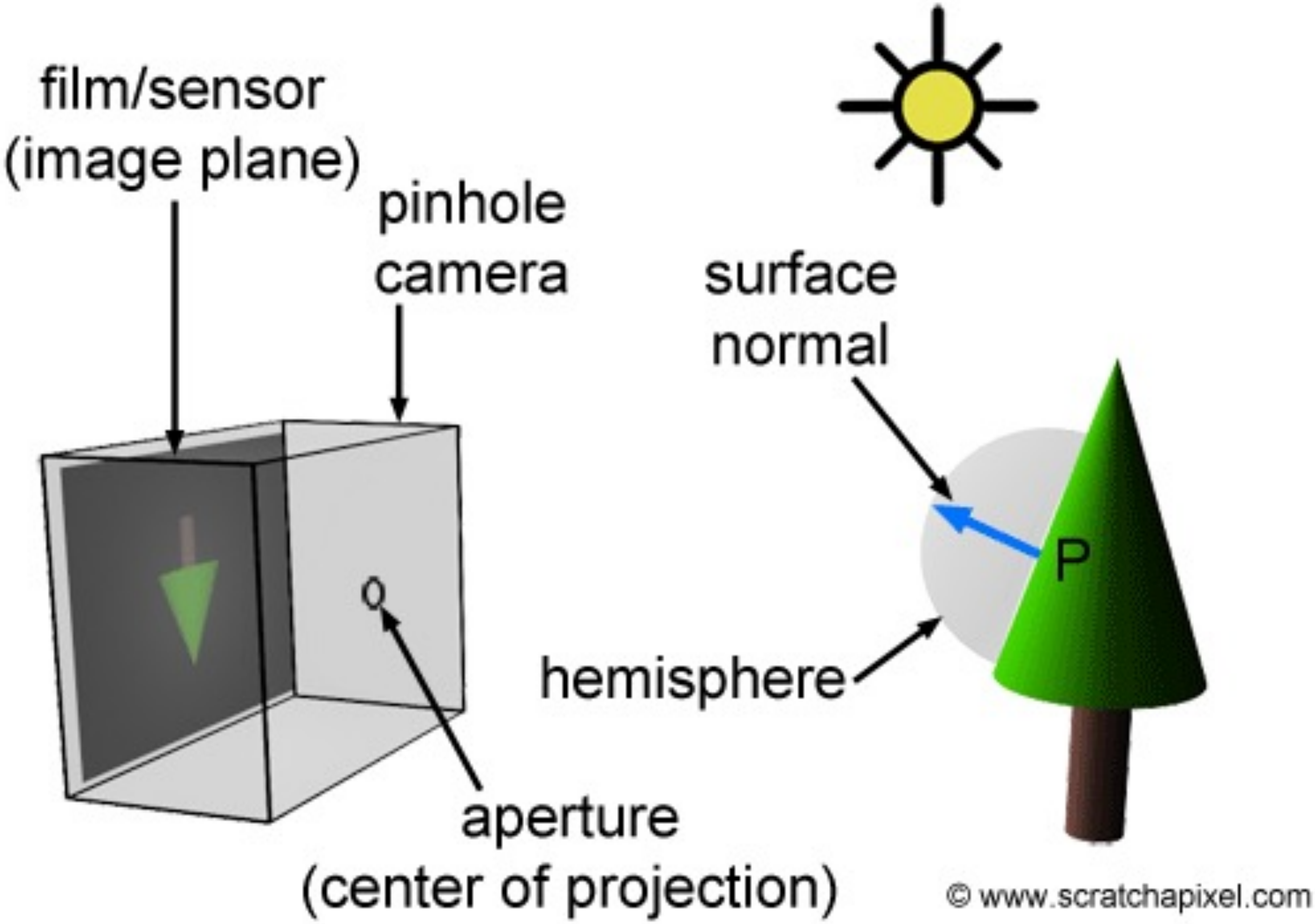
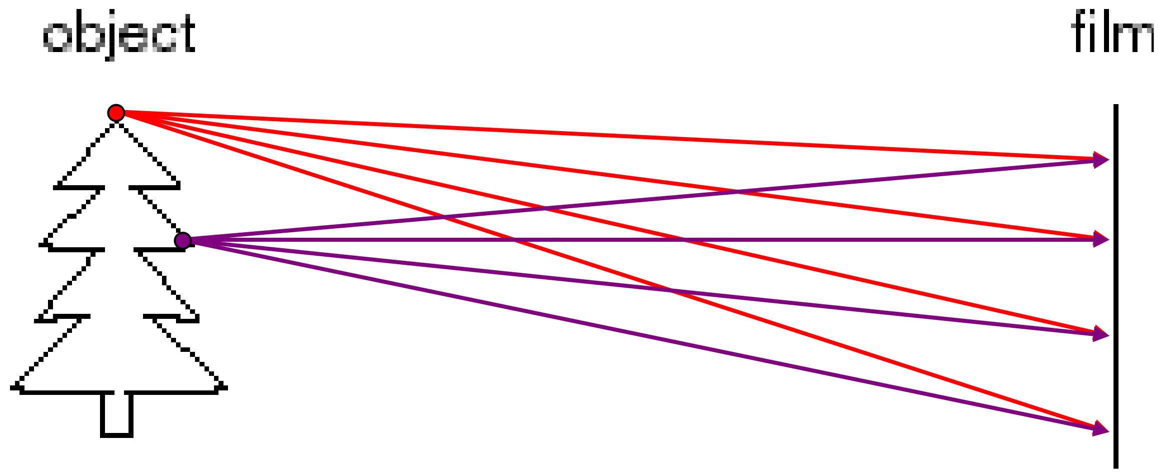


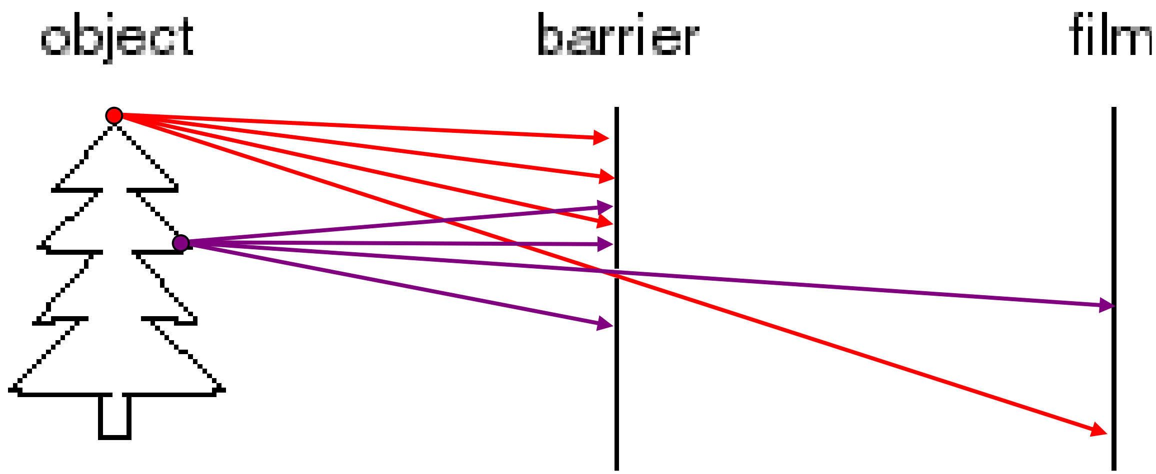
Image formation



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

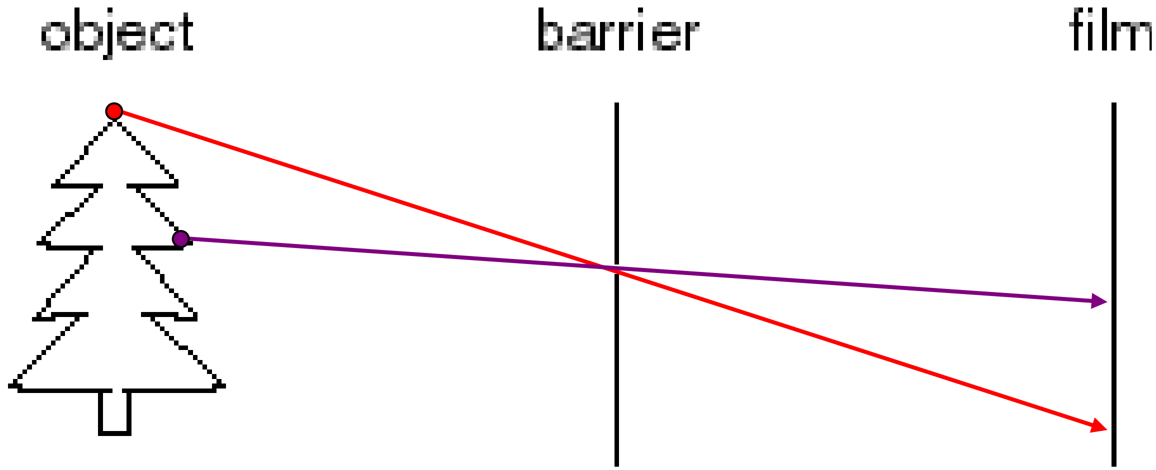


Idea 2: add a barrier to block off most of the rays

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

Slide credit Steve Seitz

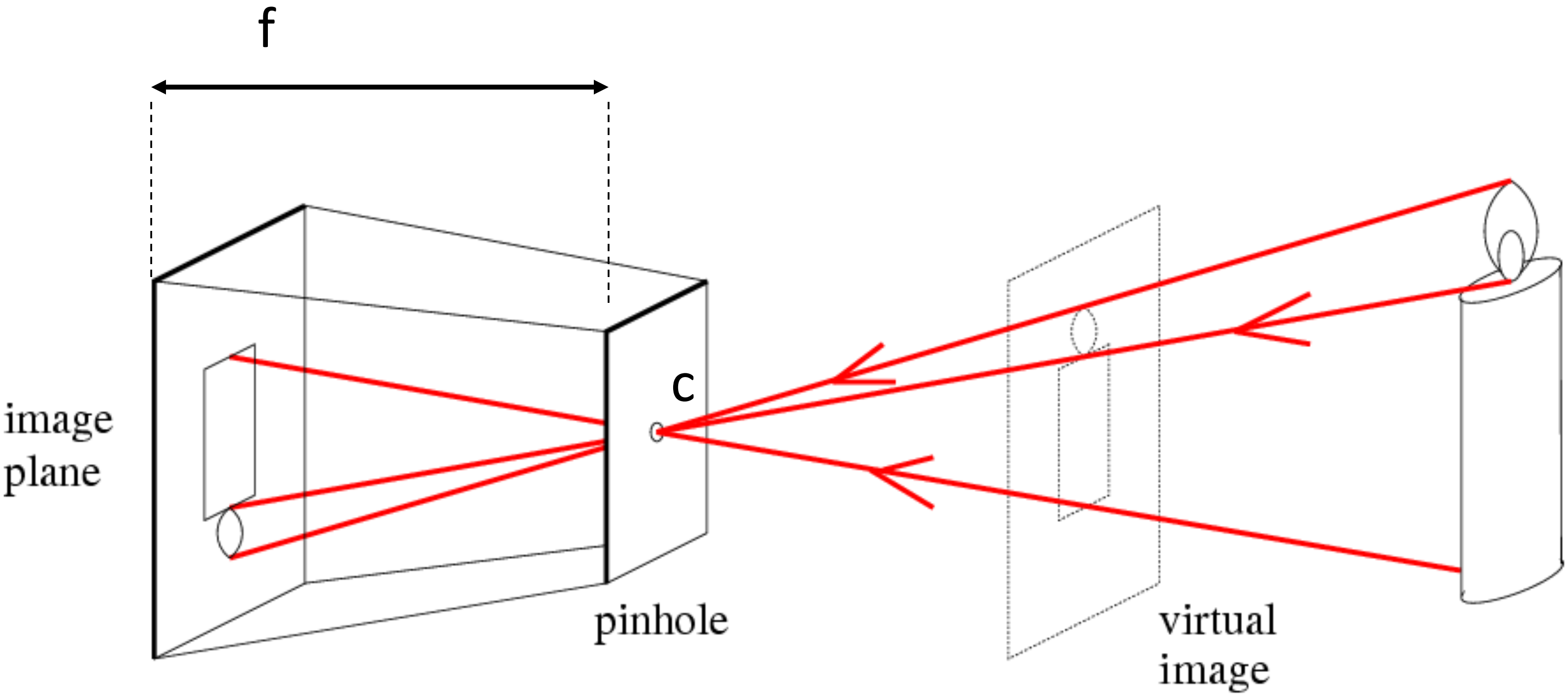
Pinhole camera



- Captures **pencil of rays** – all rays through a single point: **aperture, center of projection, optical center, focal point, camera center**
- The image is formed on the **image plane**

Slide credit Steve Seitz

Pinhole camera

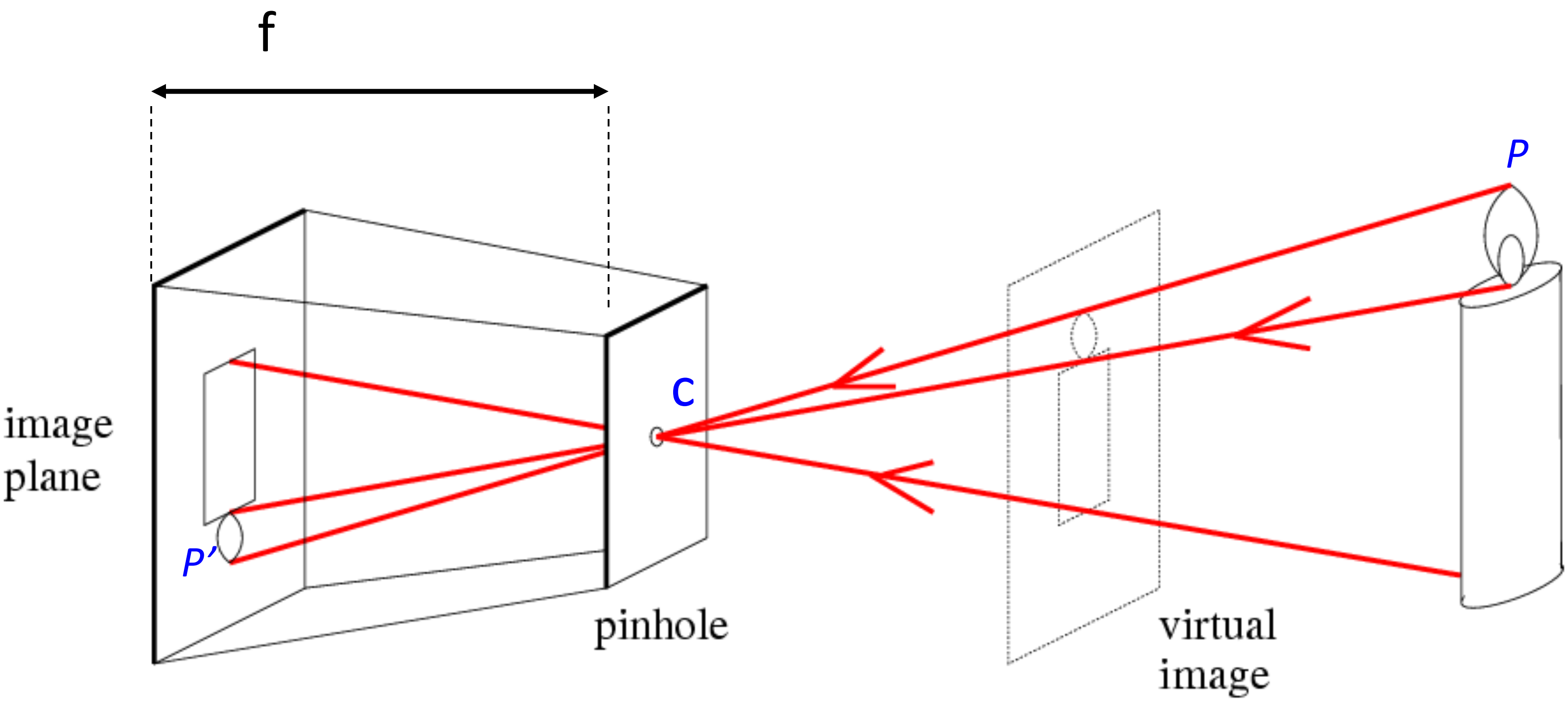


f = focal length
C = center of the camera

Note: instead of dealing with an image that is upside down, most of the time we will pretend that the image plane is in front of the camera center.

Slide credit D. Forsyth

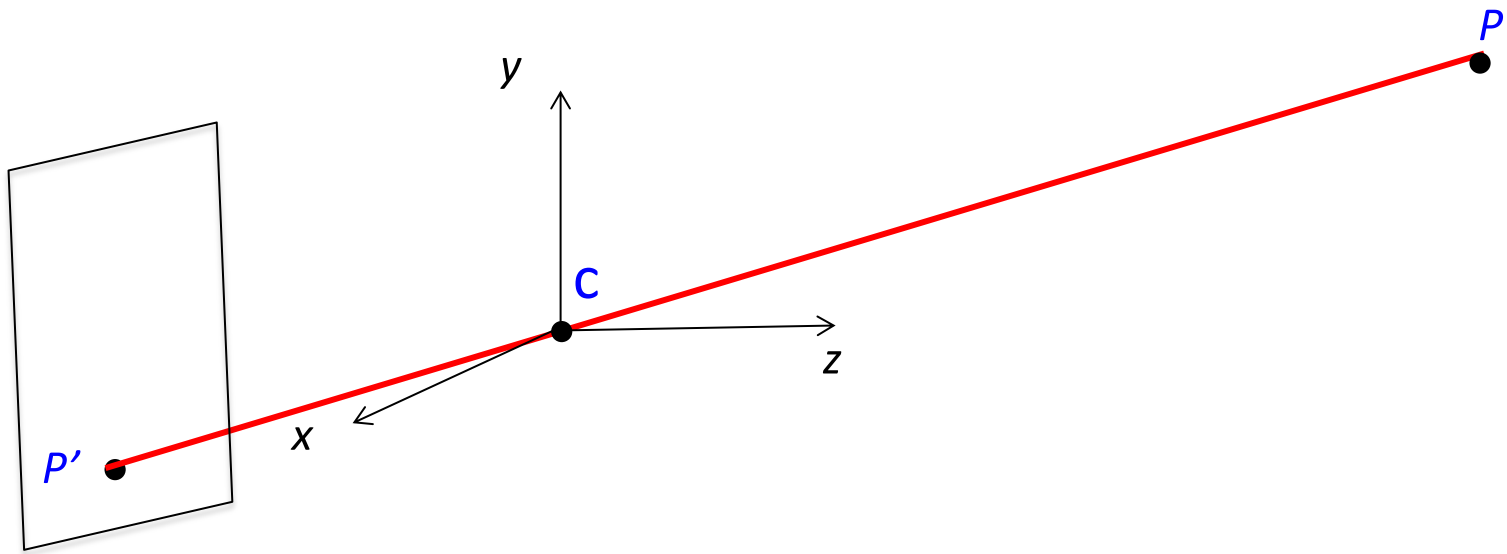
Pinhole projection model



- To compute the projection *P'* of a scene point *P*, form the **visual ray** connecting *P* to the **camera center** *C* and find where it intersects the **image plane**

Slide credit D. Forsyth

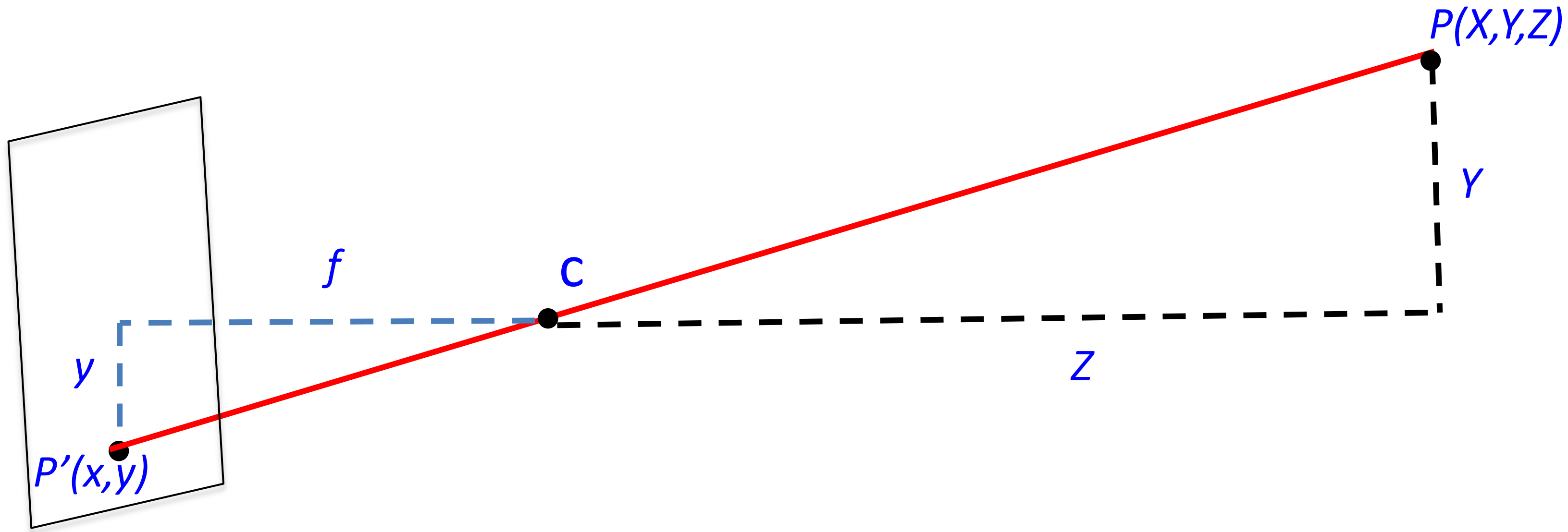
Pinhole projection model



- The coordinate system
  - The optical center (***C***) is at the origin
  - The image plane is parallel to xy-plane and perpendicular to the z-axis, which is the *optical axis* (gives the depth in 3D)

Pinhole projection model

Under the pinhole camera model, a point in space with coordinates P = (X, Y, Z)^T is mapped to the point on the image plane where a line joining the point P to the centre C of projection meets the image plane.



- Projection equations – derived using similar triangles

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

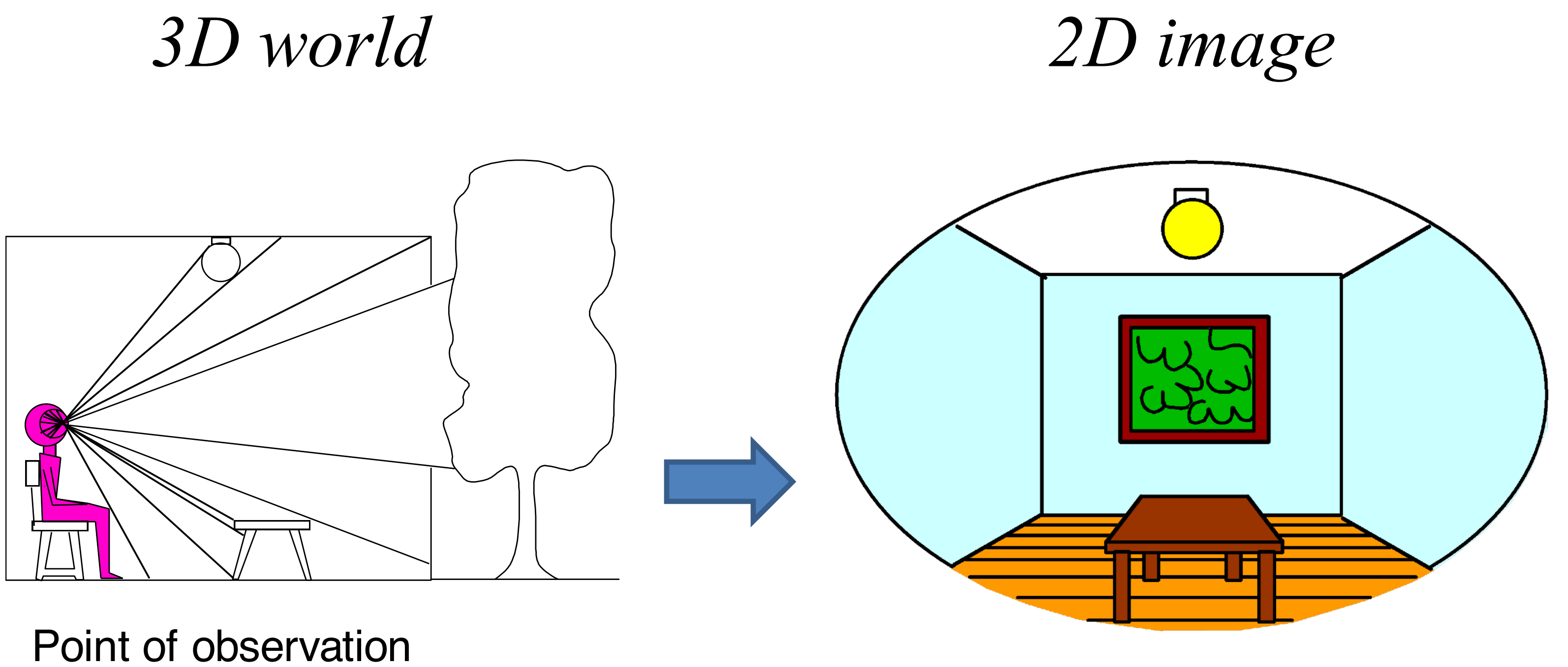
$$(X, Y, Z) \rightarrow (f\frac{X}{Z}, f\frac{Y}{Z})$$

3D world coordinates

2D image coordinates

$$P(X, Y, Z) \rightarrow P'(f\frac{X}{Z}, f\frac{Y}{Z}, f)$$

Dimensionality reduction: from 3D to 2D



Slide credit D. Forsyth

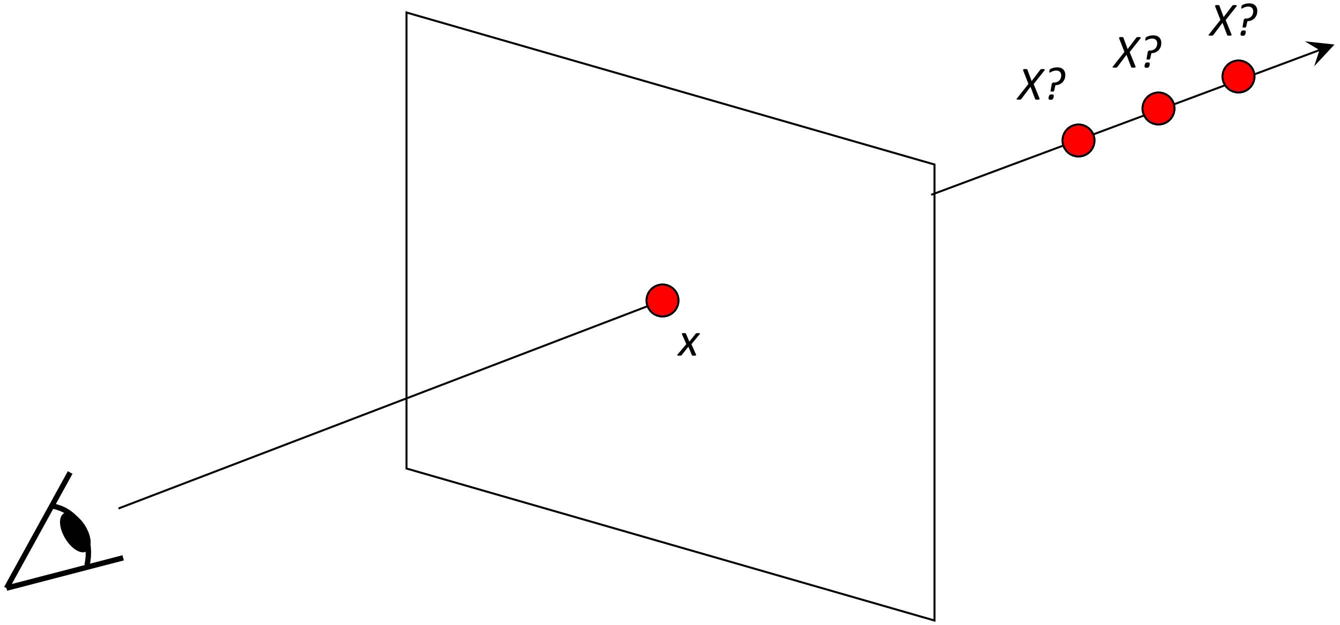
Figures © Stephen E. Palmer, 2002

Projection can be tricky…

Projection can be tricky…

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

Single-view ambiguity



A point in the 2D image corresponds to a ray in the 3D world. By projecting from 3D to 2D we loose depth.

Single-view ambiguity

Single-view ambiguity

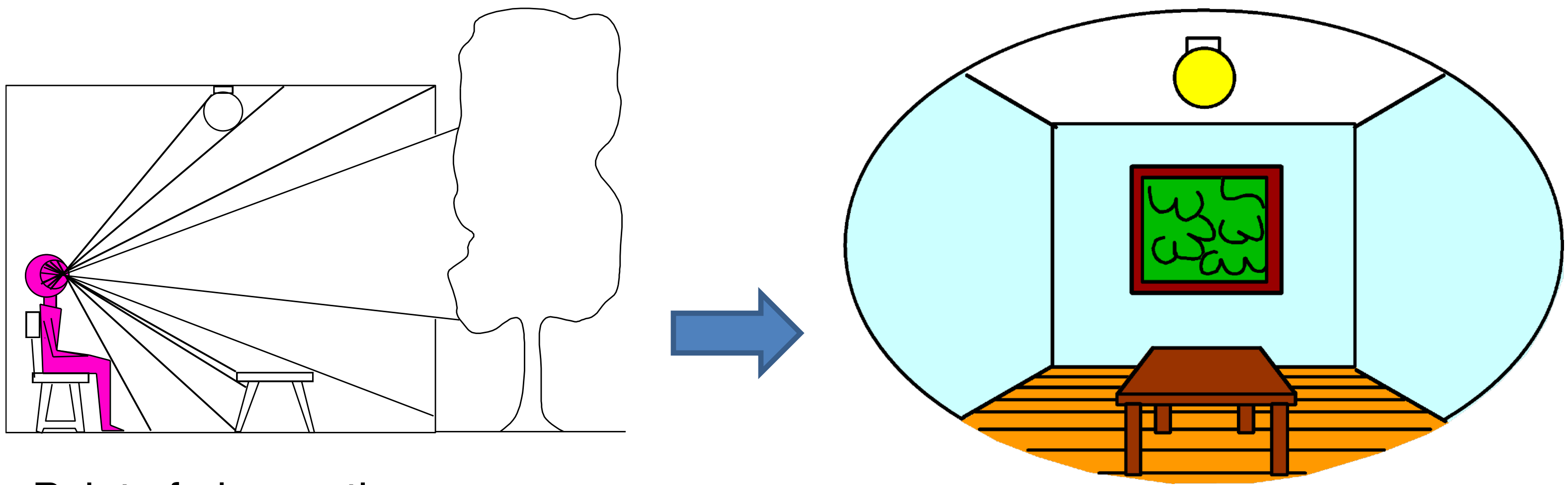
[Rashad Alakbarov shadow sculptures]

Single-view ambiguity

[Rashad Alakbarov shadow sculptures]

Dimensionality reduction: from 3D to 2D

3D world *2D image*



Point of observation

What properties of the world are preserved?

- Straight lines, incidence

What properties are not preserved?

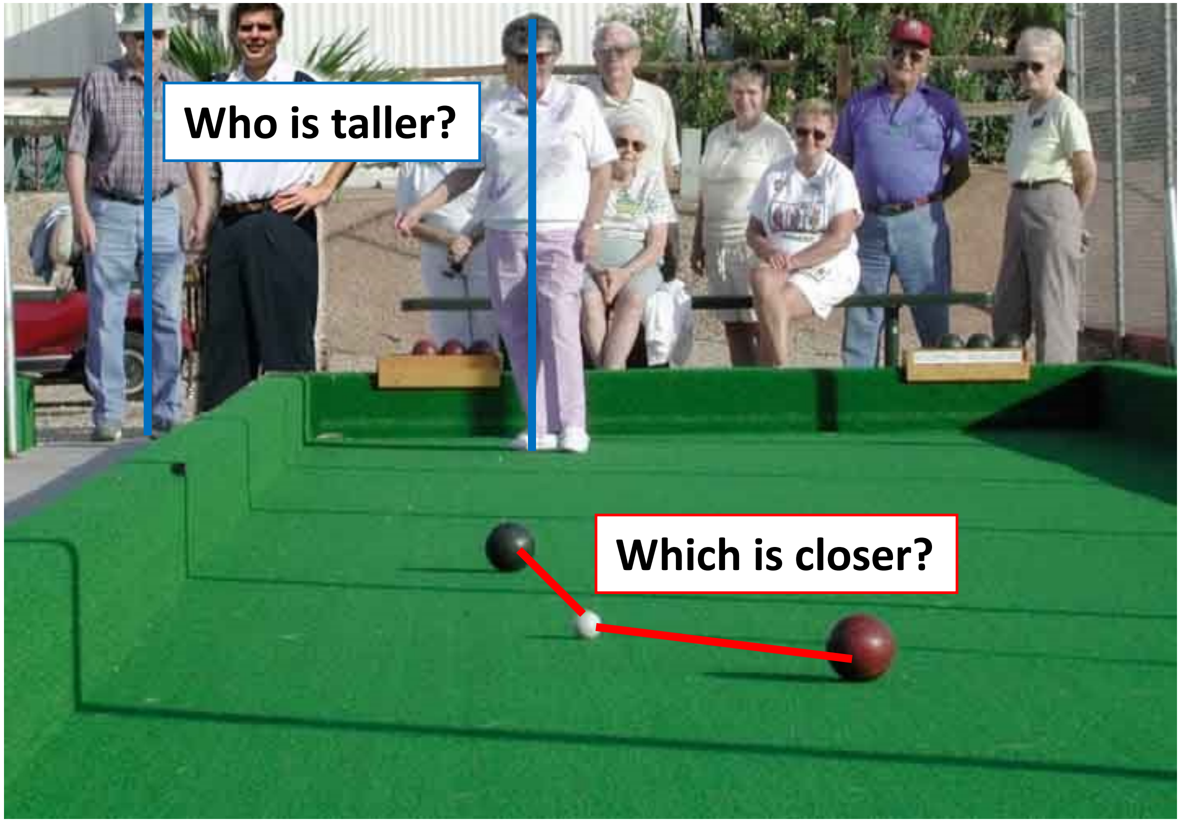
- Angles, lengths

Slide credit D. Forsyth

Projective Geometry

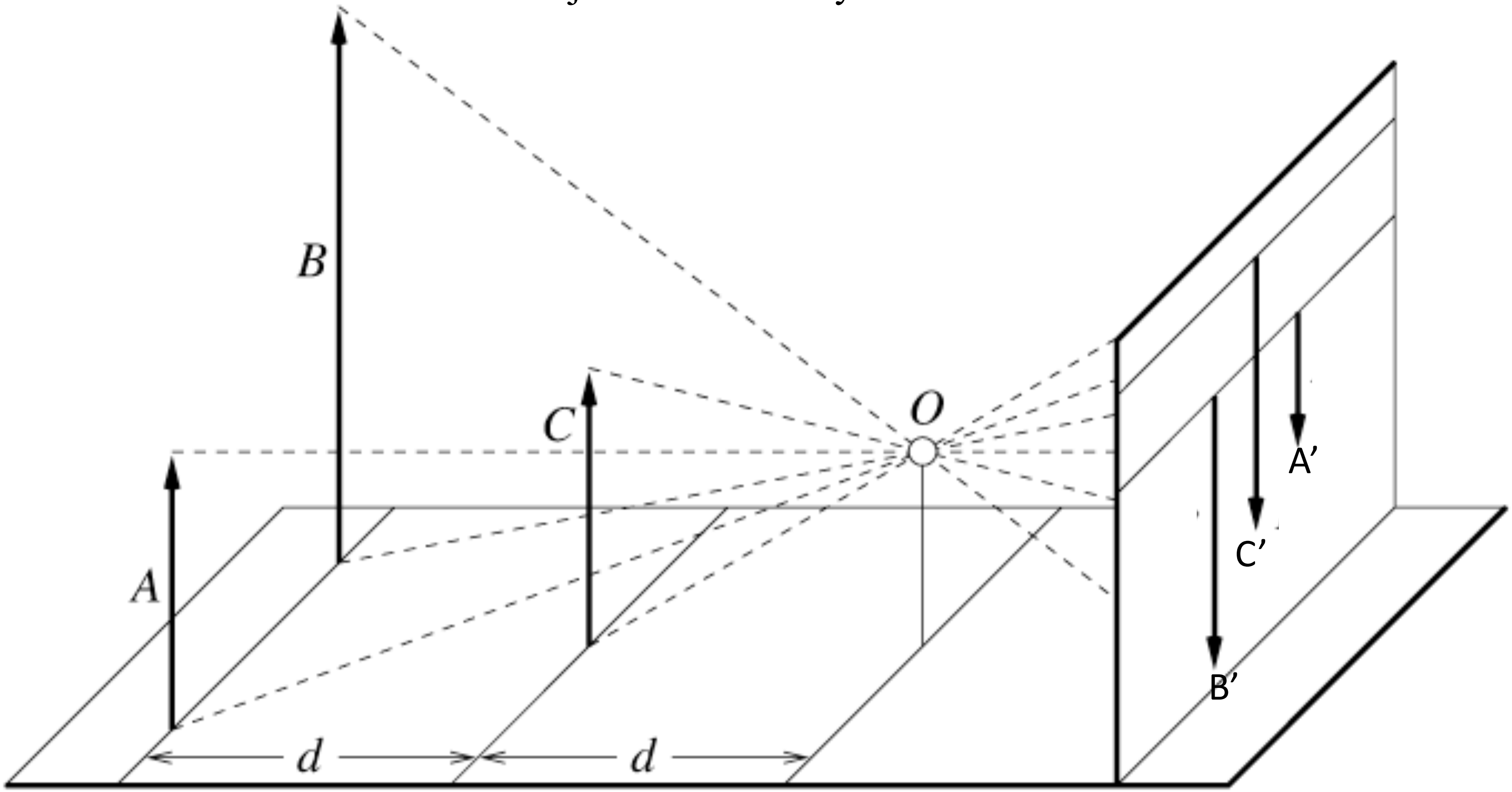
What is lost?

- Length



Length is not preserved

The camera projection scales the apparent height of objects by the inverse of the depth. This results in the well-known effect that objects farther away from the camera look smaller.

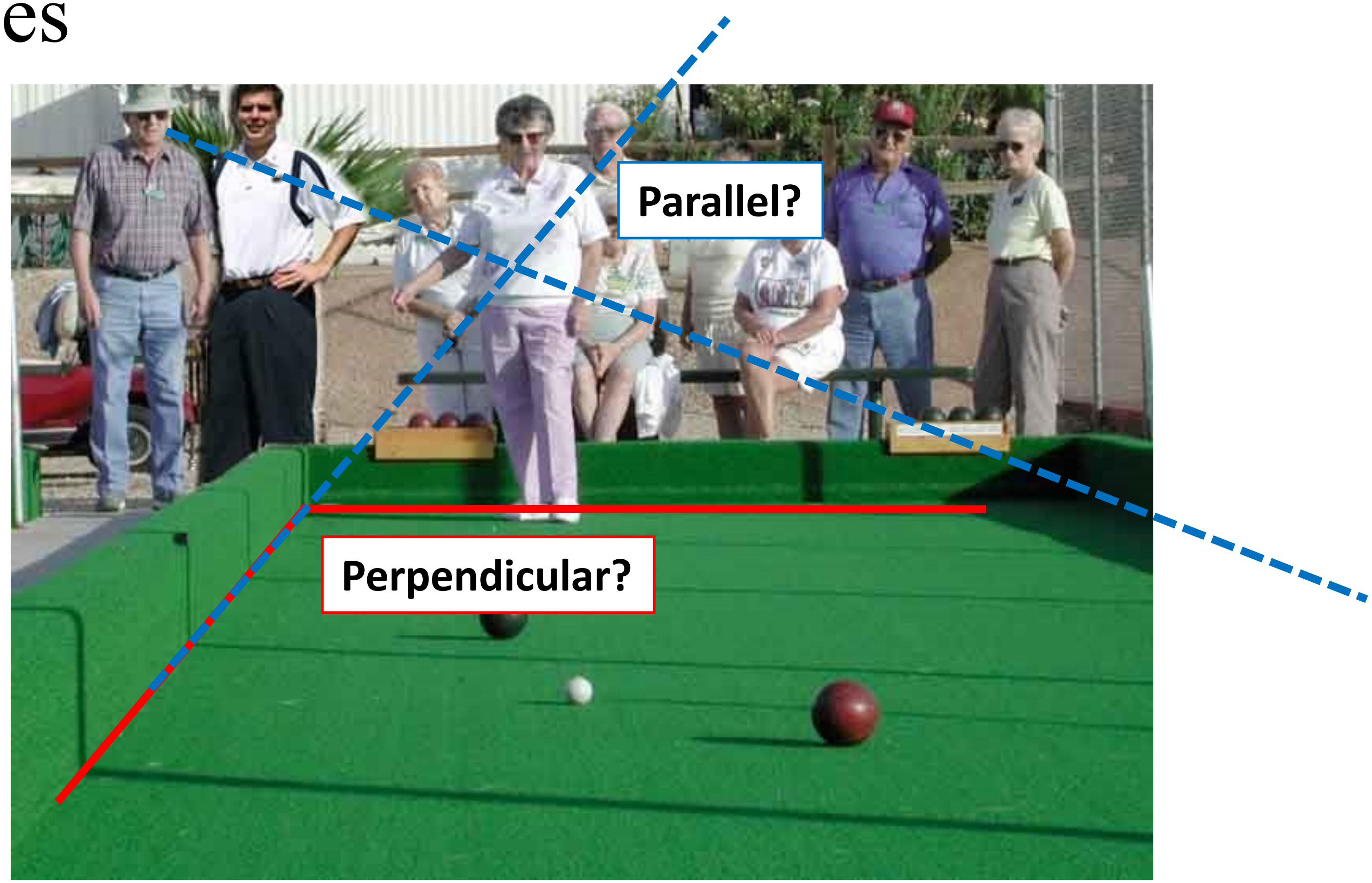


Slide credit D. Forsyth

Projective Geometry

What is lost?

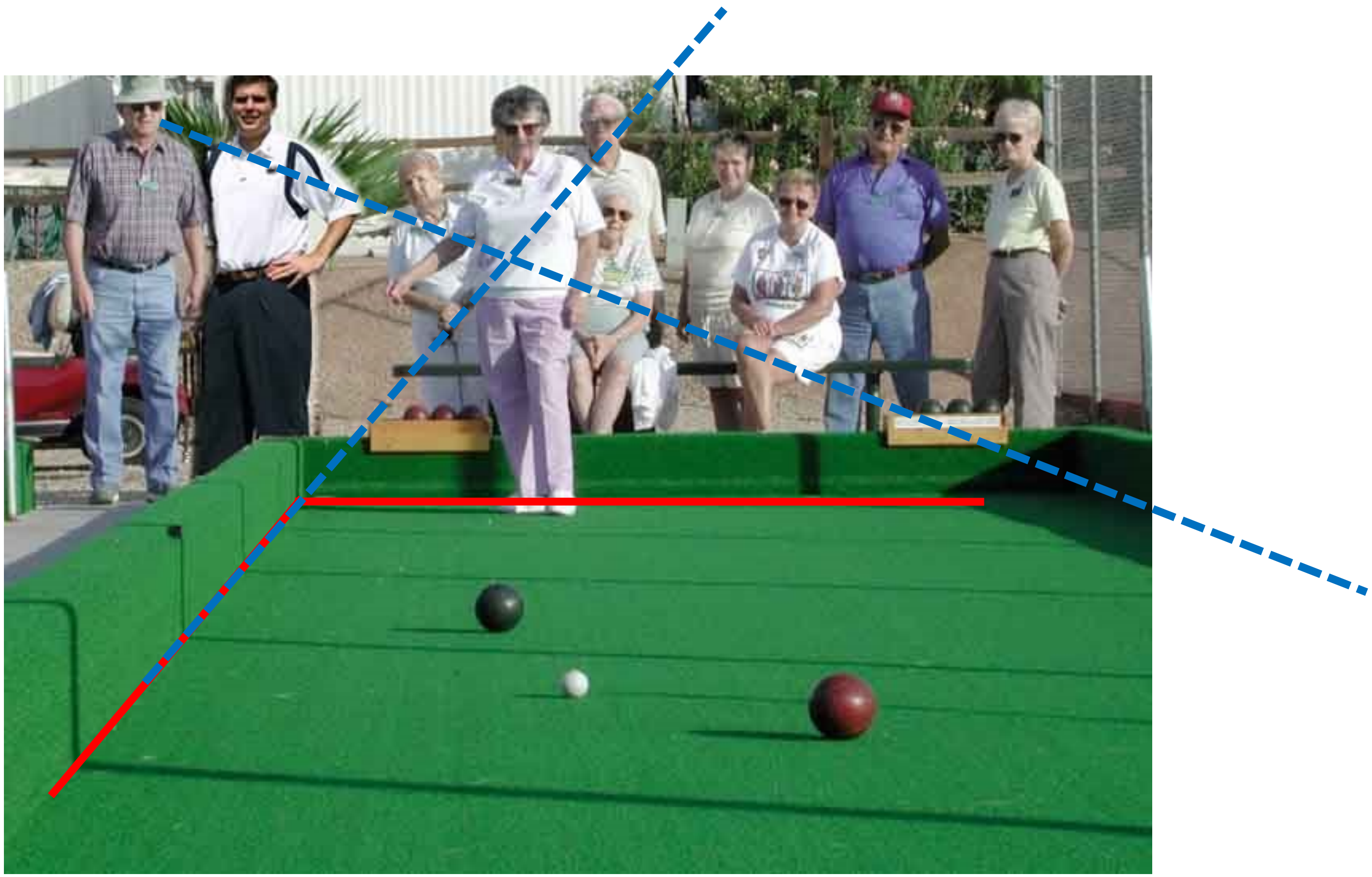
- Length
- Angles



Projective Geometry

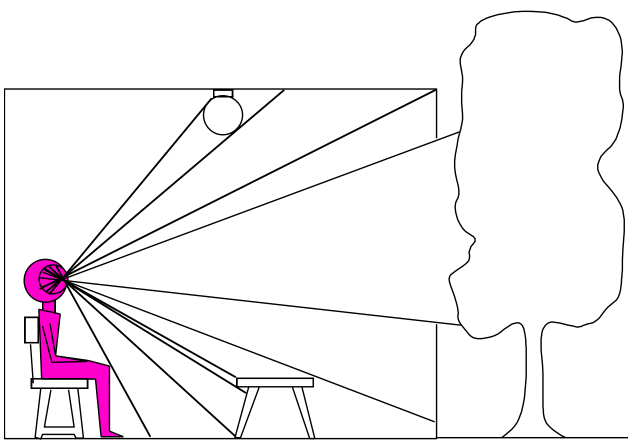
What is preserved?

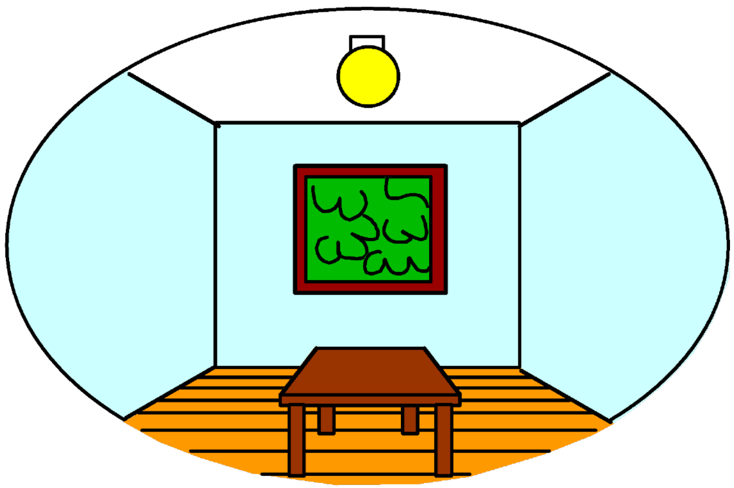
- Straight lines are still straight



Fronto-parallel planes

- What happens to the projection of a pattern on a plane parallel to the image plane?
  - All points on that plane are at a fixed *depth* Z
  - The pattern gets scaled by a factor of f / Z, but angles and ratios of lengths/areas are preserved

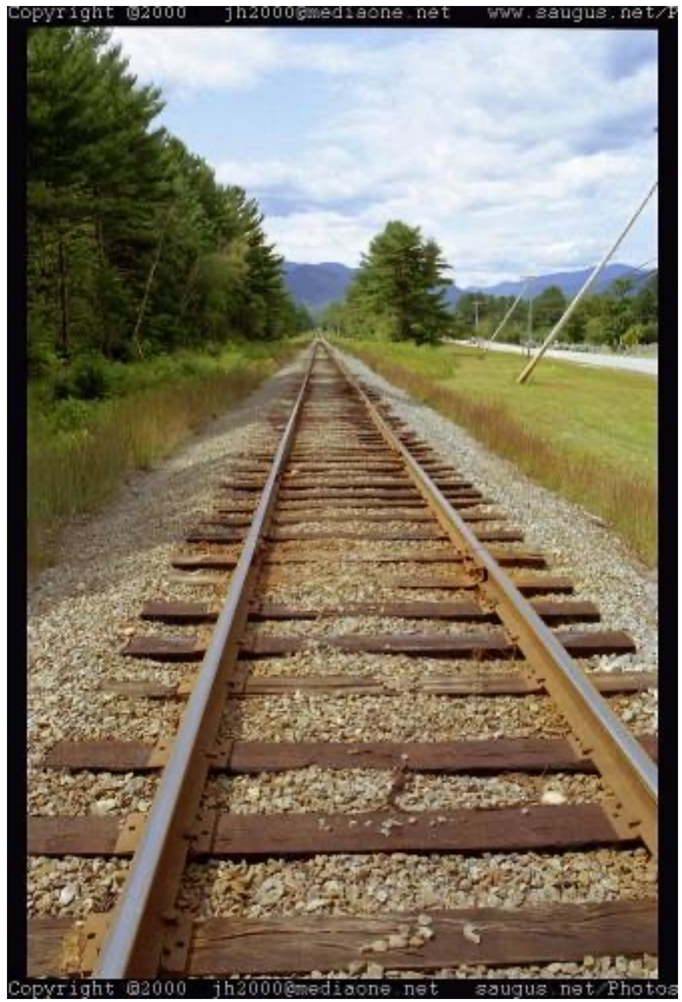




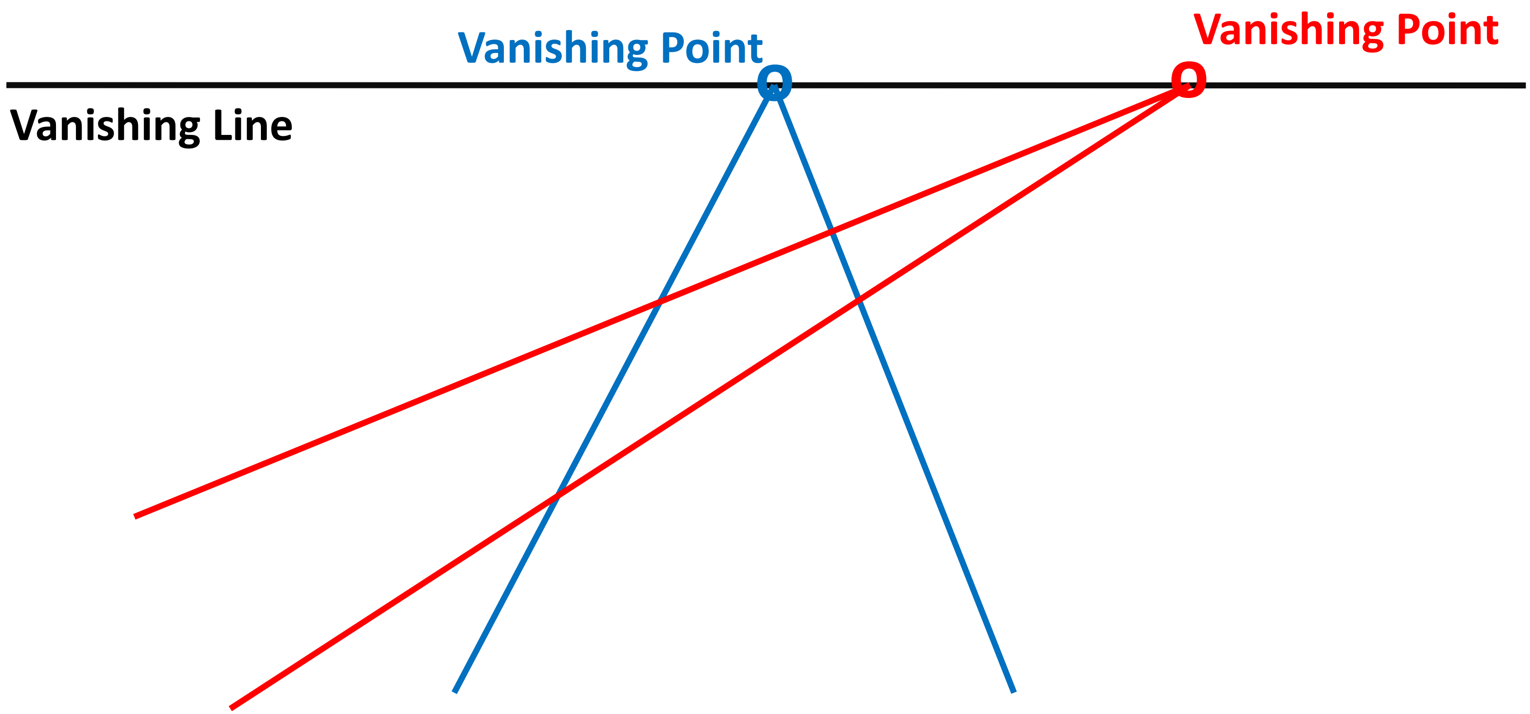
$$(X, Y, Z) \rightarrow \left(f\frac{X}{Z}, f\frac{Y}{Z}\right) = \left(X\frac{f}{Z}, Y\frac{f}{Z}\right)$$

Vanishing points and lines

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

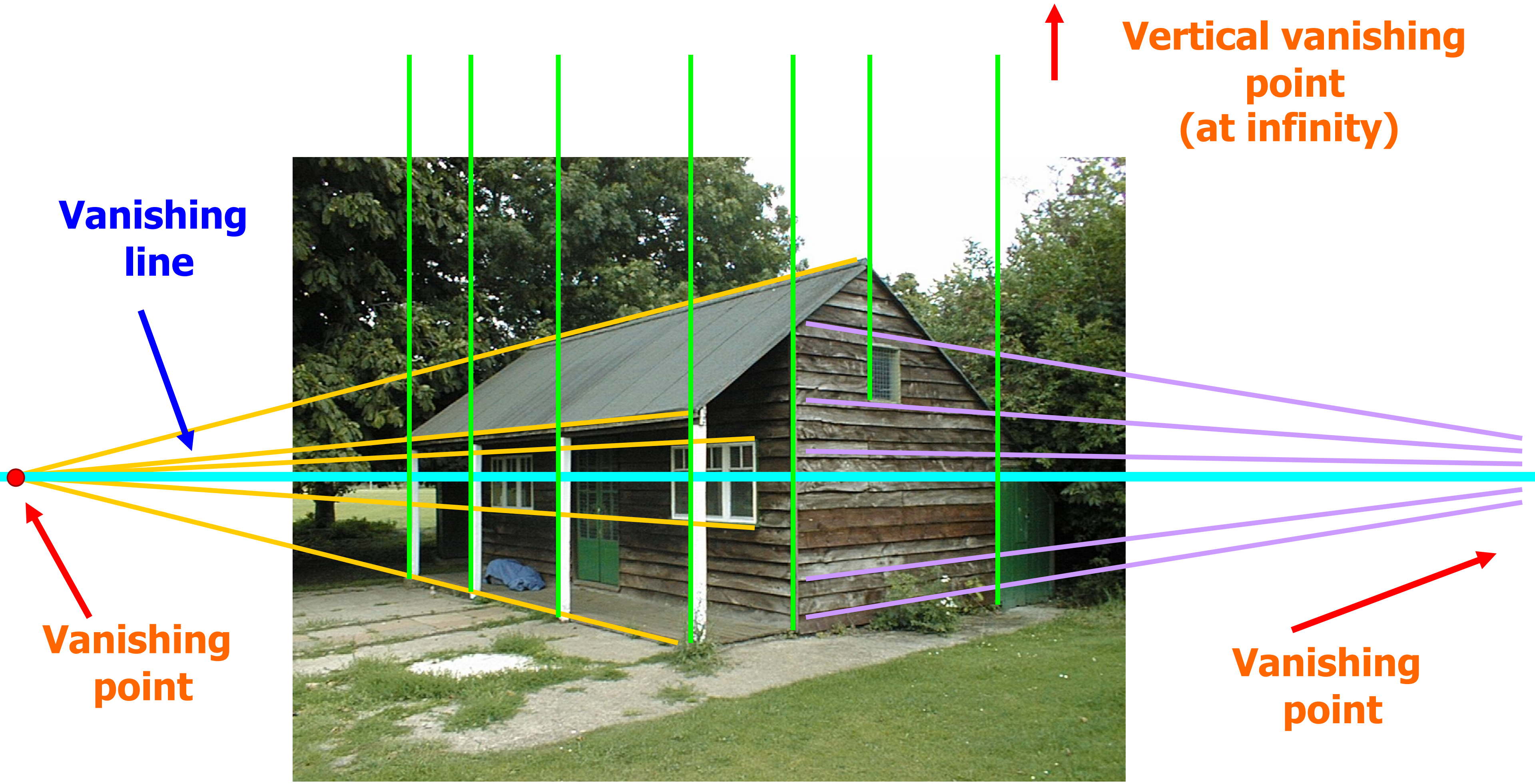


Vanishing points and lines



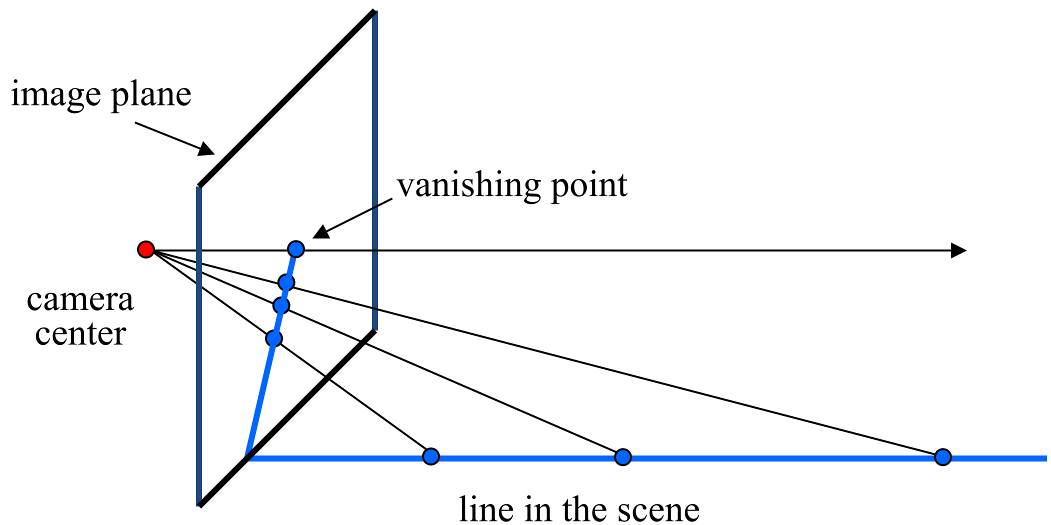
- The projections of parallel 3D lines intersect at a **vanishing point**
- The projection of parallel 3D planes intersect at a **vanishing line**
- If a set of parallel 3D lines are also parallel to a particular plane, their vanishing point will lie on the vanishing line of the plane
- Not all lines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

Vanishing points and lines



Slide credit A. Efros,
Photo from Criminisi

Constructing the vanishing point of a line



If we have another line in the scene parallel to the first one, it will have the same vanishing point

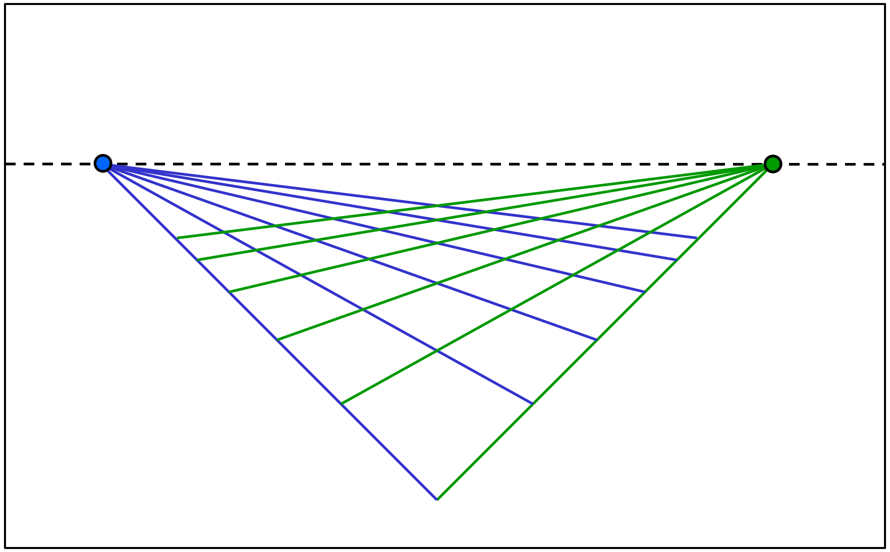
Slide by Steve Seitz

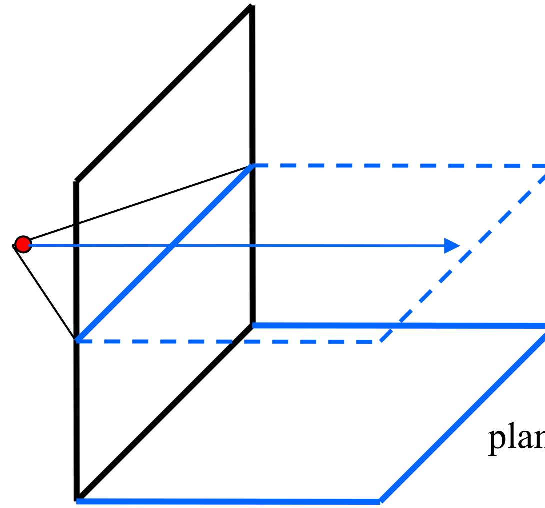
Vanishing lines of planes

How do we construct the vanishing line of a plane?

Image source

Vanishing lines of planes





- *Horizon*: vanishing line of the ground plane
  - All points at the same height as the camera project to the horizon
  - Points higher (resp. lower) than the camera project above (resp. below) the horizon
  - Provides way of comparing height of objects

Slide credit Steve Seitz

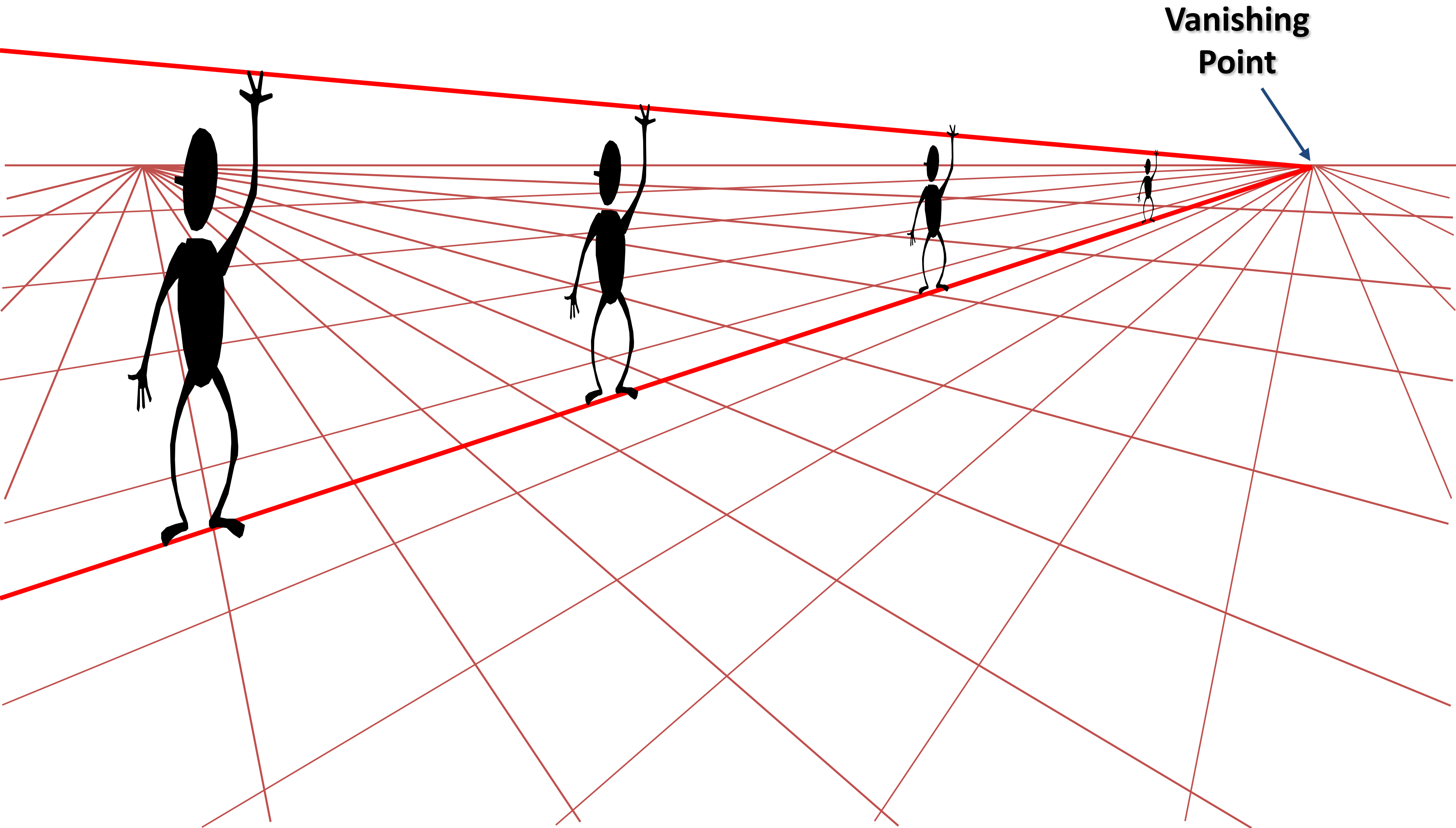
Vanishing lines of planes

Is the parachutist above or below the camera?

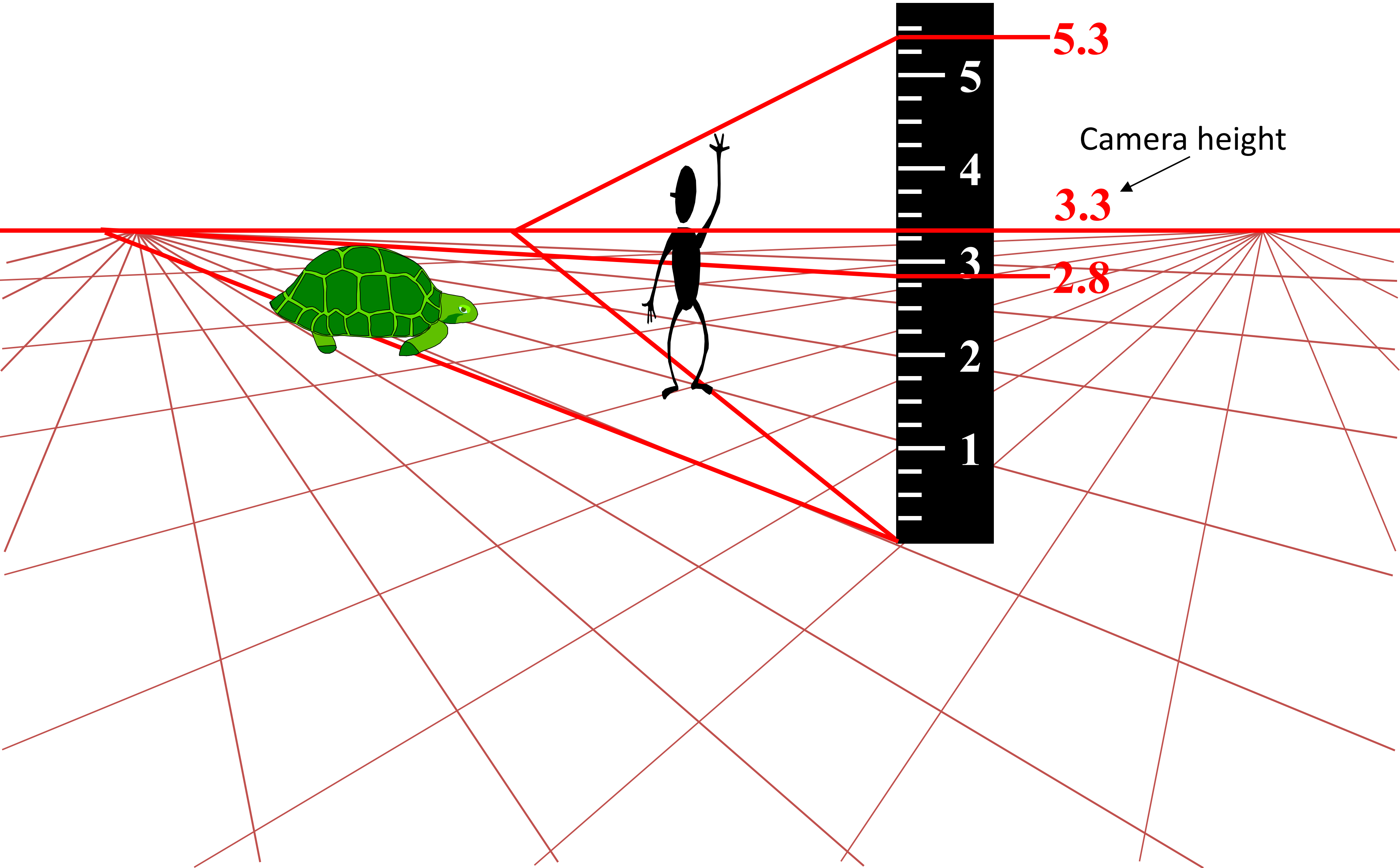
A: The parachutist is below the horizon, so is below the camera

Slide credit Steve Seitz

Comparing heights

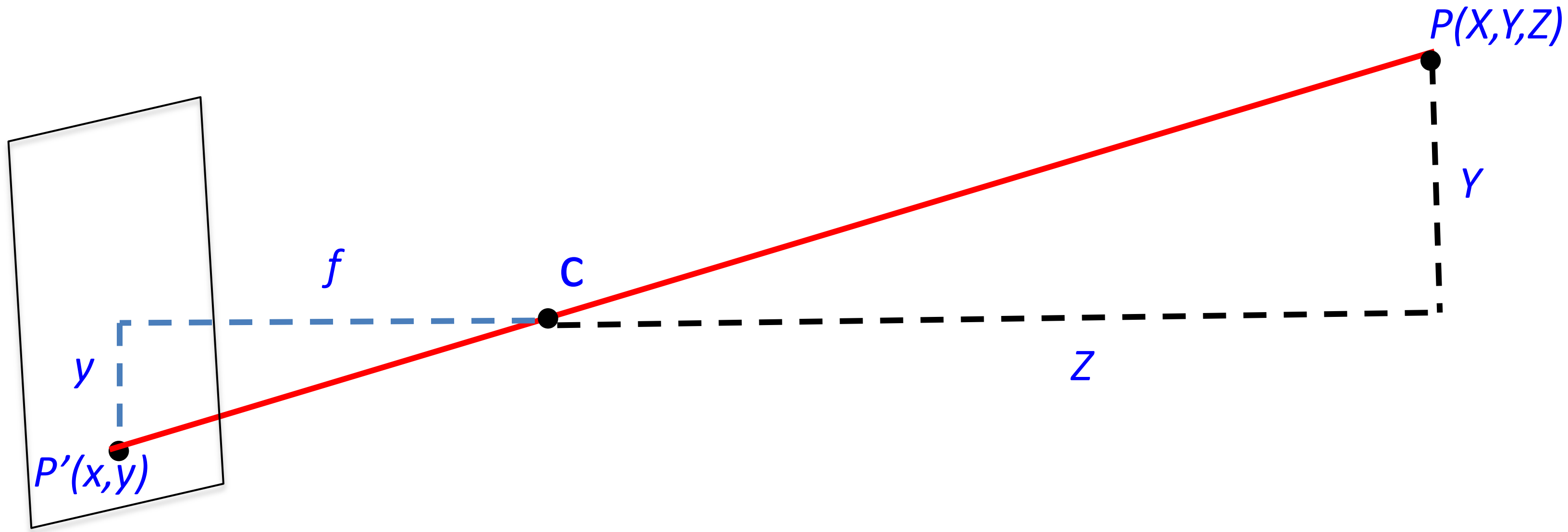


Measuring height



Pinhole projection model

Under the pinhole camera model, a point in space with coordinates $P = (X, Y, Z)^T$ is mapped to the point on the image plane where a line joining the point P to the centre C of projection meets the image plane.



- Projection equations – derived using similar triangles

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

$$P(X,Y,Z) \rightarrow P'(f\frac{X}{Z}, f\frac{Y}{Z}, f)$$

$$(X,Y,Z) \rightarrow (f\frac{X}{Z}, f\frac{Y}{Z})$$

3D world coordinates

2D image coordinates

Homogeneous coordinates

- Is this a linear transformation? $(X,Y,Z) \rightarrow (f\frac{X}{Z}, f\frac{Y}{Z})$
  - no - division by z is nonlinear

Trick to make it linear: add one more coordinate, go from Cartesian to homogenous coordinates

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

Cartesian image coordinates — homogeneous image coordinates

$$(X,Y,Z) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Cartesian Scene coordinates — homogeneous scene coordinates

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} \Rightarrow \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix}$$

Homogeneous Coordinates

Cartesian Coordinates

Slide credit: Steve Seitz

Homogeneous coordinates

Conversion

Converting from *Cartesian* 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 to *Cartesian* coordinates

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

Cartesian coordinates

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \Rightarrow (X / W, Y / W, Z / W)$$

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 a 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

$$p_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

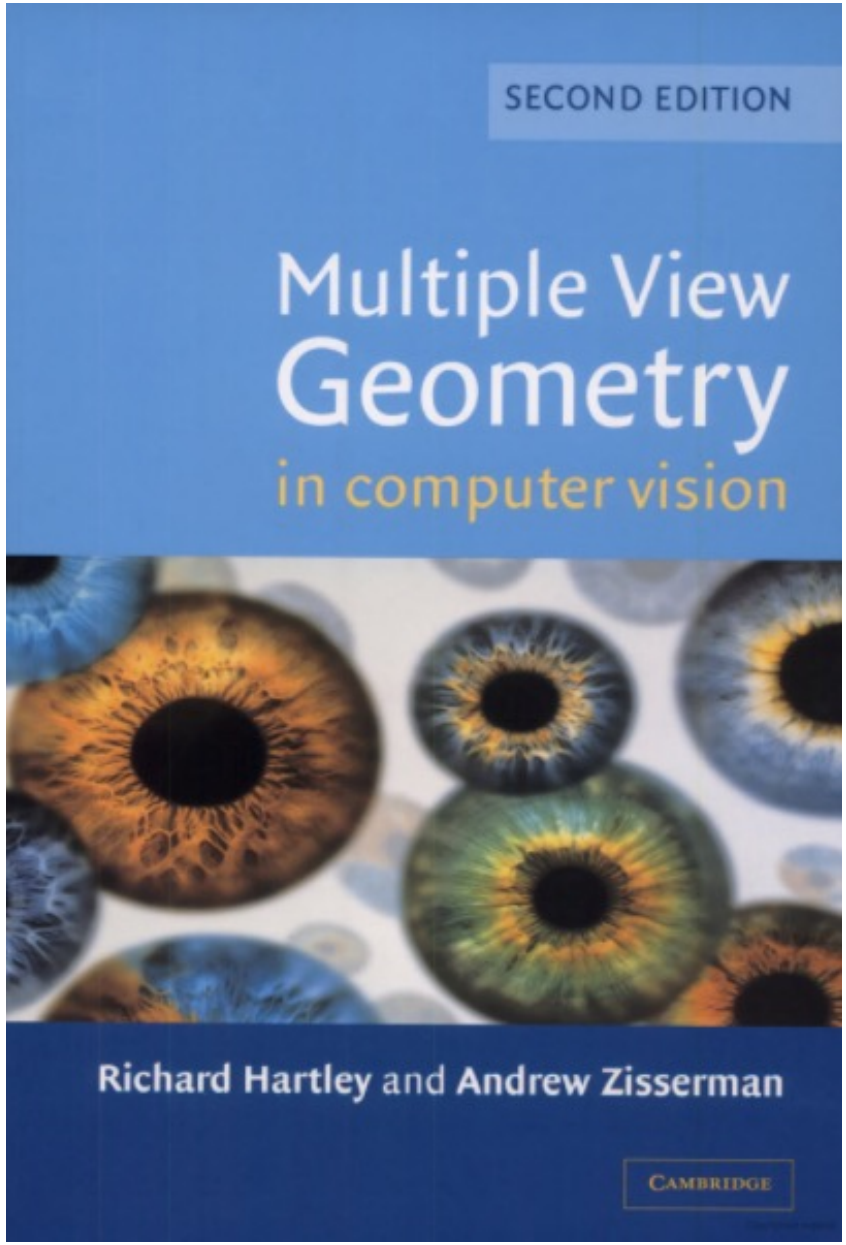
- Line given by cross product of two points

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

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

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

Useful reference

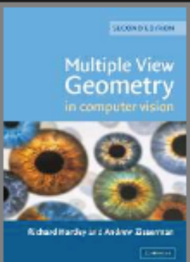


Useful reference

Not Secure | robots.ox.ac.uk/~vgg/hzbook/index.html

Cookies

This website uses Google Analytics to help us improve the website content. This requires the use of standard Google Analytics cookies, as well as a cookie to record your res
to this confirmation request. If this is OK with you, please click 'Accept cookies', otherwise you will see this notice on every page. For more information, please click here.



Multiple View Geometry in Computer Vision Second Edition

Richard Hartley and Andrew Zisserman,
Cambridge University Press, March 2004.

Sample chapters

- Contents pdf
- Introduction pdf
- Epipolar Geometry and the Fundamental Matrix pdf
- The Trifocal Tensor pdf
- Bibliography pdf

Figures

- Download figures in pdf, png, bmp or postscript format.

Clarifications

- Image rectification (chapter 11) pdf
- Number of linearly independent trilinear relations (chapter 17) pdf mathematica

Also Available

See the First Edition's page for sample chapters, downloadable figures, corrections and errata perta
to the first edition.

Buy Online

Amazon: Second edition.
CUP: Second edition.

Basic geometry in homogeneous coordinates

- Line given by cross product of two points

$$p_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad p_j = \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} \quad line_{ij} = p_i \times p_j = \begin{vmatrix} e_1 & e_2 & e_3 \\ x_i & y_i & 1 \\ x_j & y_j & 1 \end{vmatrix}$$

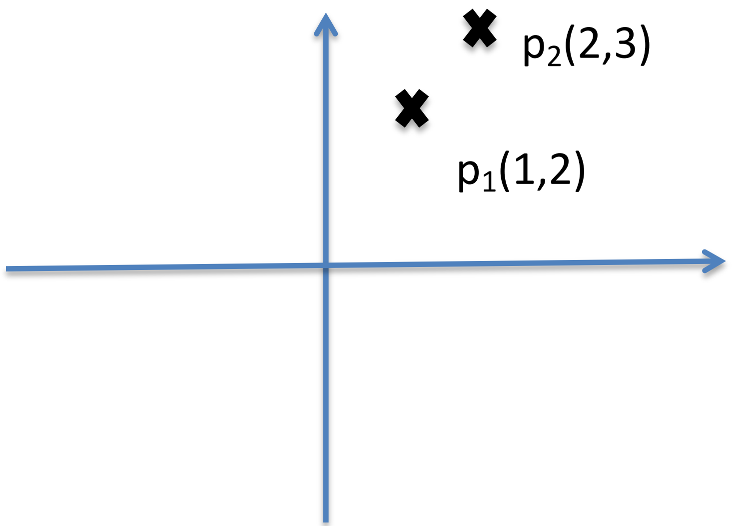
$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Basic geometry in homogeneous coordinates

- Line given by cross product of two points

$$p_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad p_2 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \qquad line_{12} = p_1 \times p_2 = \begin{vmatrix} e_1 & e_2 & e_3 \\ 1 & 2 & 1 \\ 2 & 3 & 1 \end{vmatrix}$$

$$line_{12} = 2e_1 + 3e_3 + 2e_2 - 4e_3 - 3e_1 - e_2 = -e_1 + e_2 - e_3 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

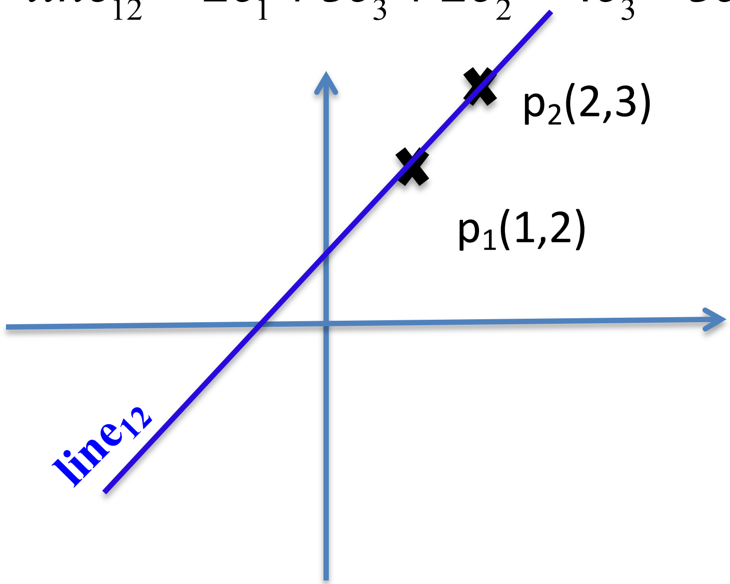


Basic geometry in homogeneous coordinates

- Line given by cross product of two points

$$p_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad p_2 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \qquad line_{12} = p_1 \times p_2 = \begin{vmatrix} e_1 & e_2 & e_3 \\ 1 & 2 & 1 \\ 2 & 3 & 1 \end{vmatrix}$$

$$line_{12} = 2e_1 + 3e_3 + 2e_2 - 4e_3 - 3e_1 - e_2 = -e_1 + e_2 - e_3 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$



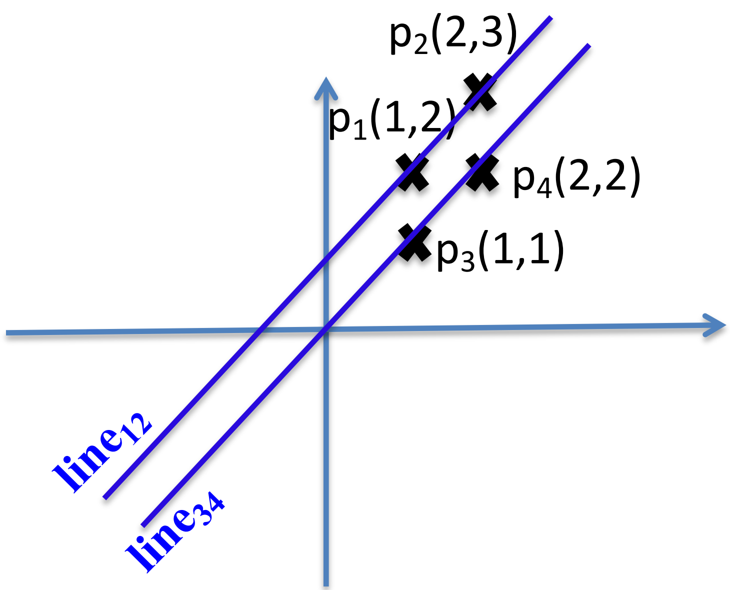
$$line_{12} : ax + by + c = 0, a = -1, b = 1, c = -1$$

$$line_{12} : -x + y - 1 = 0$$

Basic geometry in homogeneous coordinates

- Line given by cross product of two points

$$p_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad p_2 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \quad p_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad p_4 = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \quad line_{34} = p_3 \times p_4 = \begin{vmatrix} e_1 & e_2 & e_3 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{vmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$



$$line_{12} : -x + y - 1 = 0$$

$$line_{34} : x - y = 0$$

Basic geometry in homogeneous coordinates

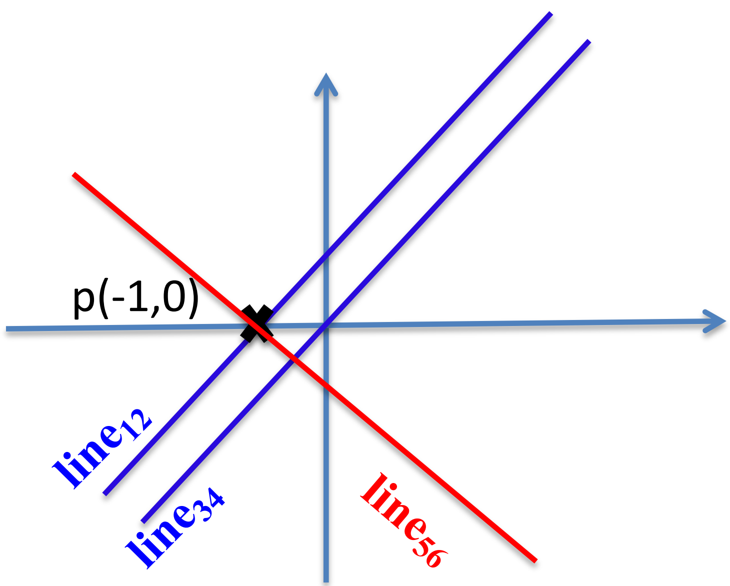
- Intersection of two lines given by cross product of the lines

$$line_{12} : -x + y - 1 = 0$$

$$line_{34} : x - y = 0$$

$$line_{56} : x + y + 1 = 0$$

$$p = line_{12} \times line_{56} = \begin{vmatrix} e_1 & e_2 & e_3 \\ -1 & 1 & -1 \\ 1 & 1 & 1 \end{vmatrix} = \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} = -2 \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



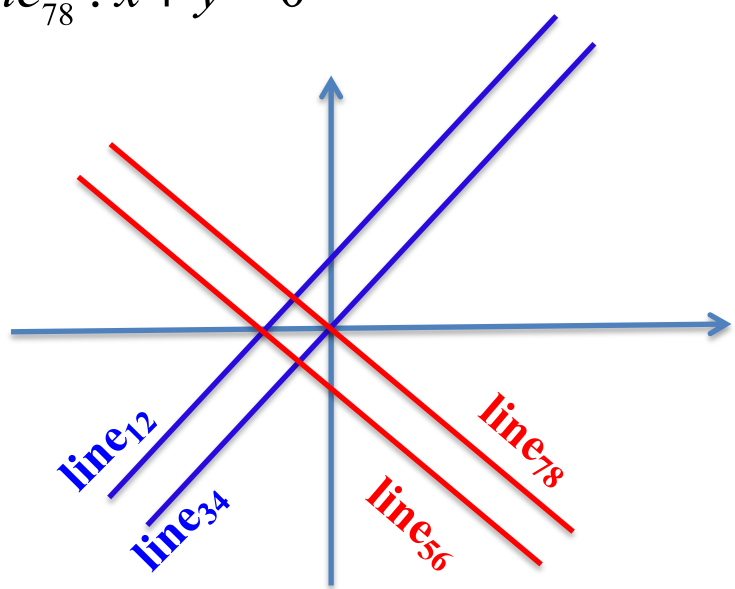
Another problem solved by homogeneous coordinates

Intersection of parallel lines

$$line_{12}: -x + y - 1 = 0$$
$$line_{34}: x - y = 0$$
$$line_{56}: x + y + 1 = 0$$
$$line_{78}: x + y = 0$$

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

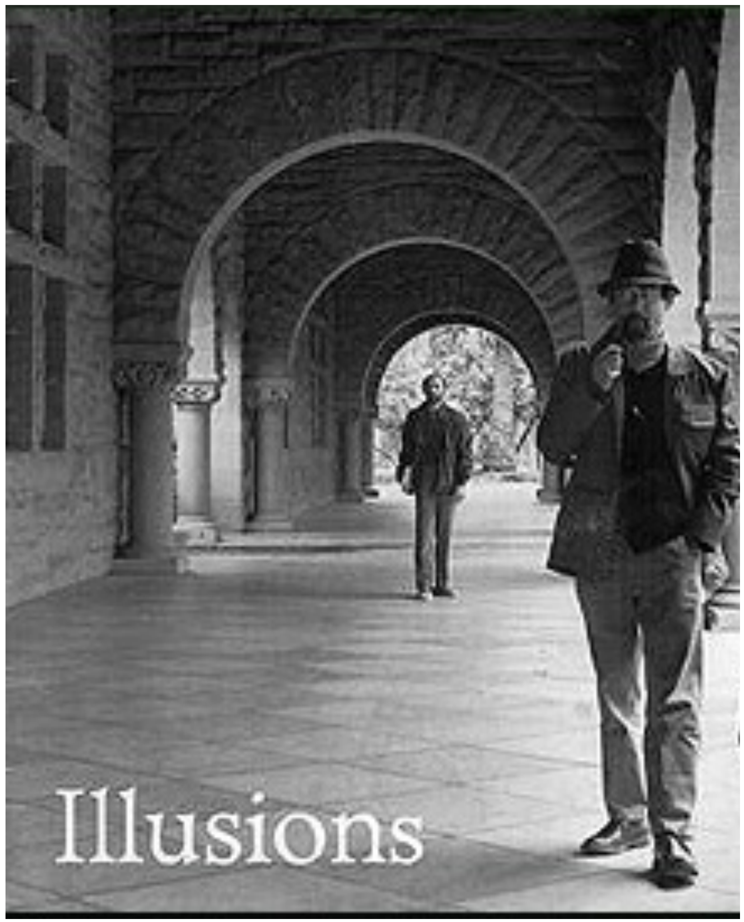
Cartesian: (Inf, Inf)
Homogeneous: (-1, 1, 0)



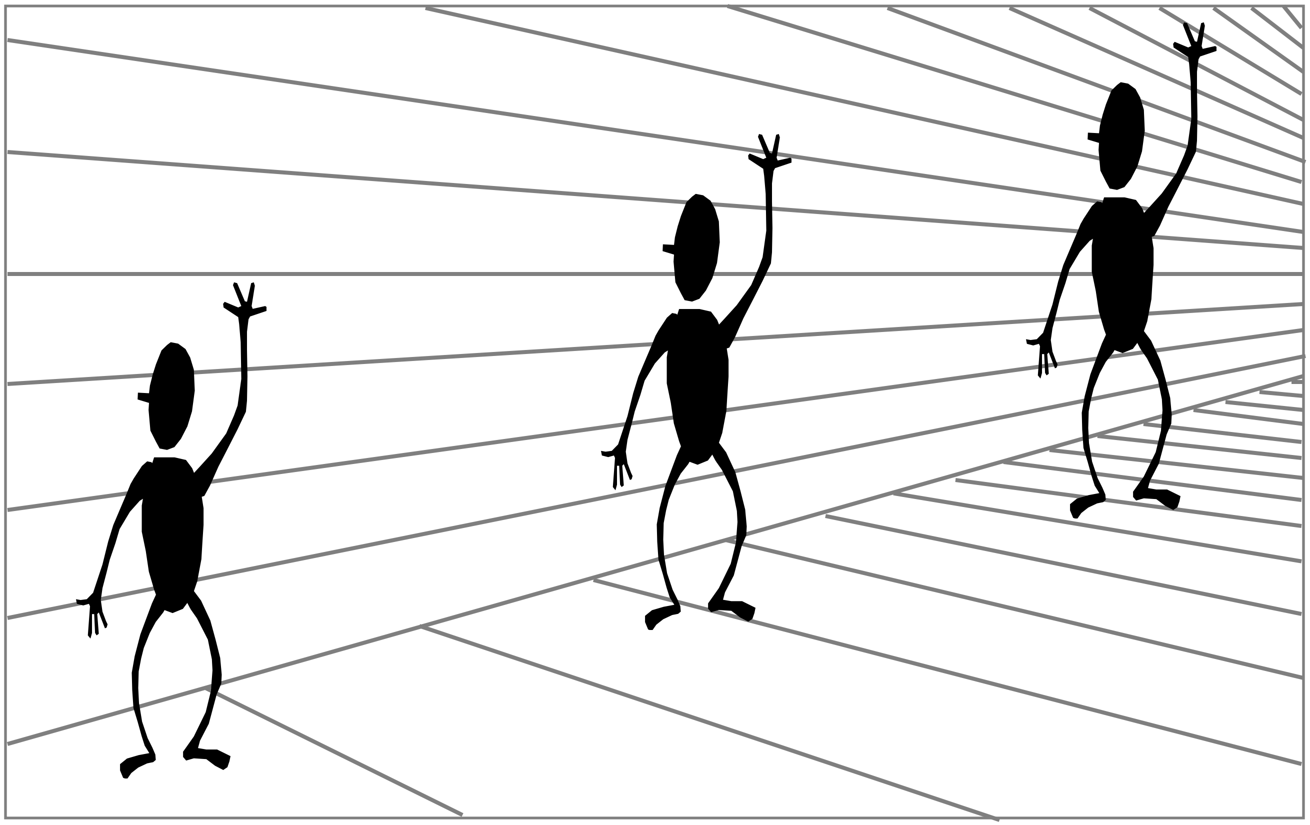
$$p = line_{12} \times line_{34} = \begin{vmatrix} e_1 & e_2 & e_3 \\ -1 & 1 & -1 \\ 1 & -1 & 0 \end{vmatrix} = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$$

$$p = line_{56} \times line_{78} = \begin{vmatrix} e_1 & e_2 & e_3 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

How can we measure the size of 3D objects from an image?

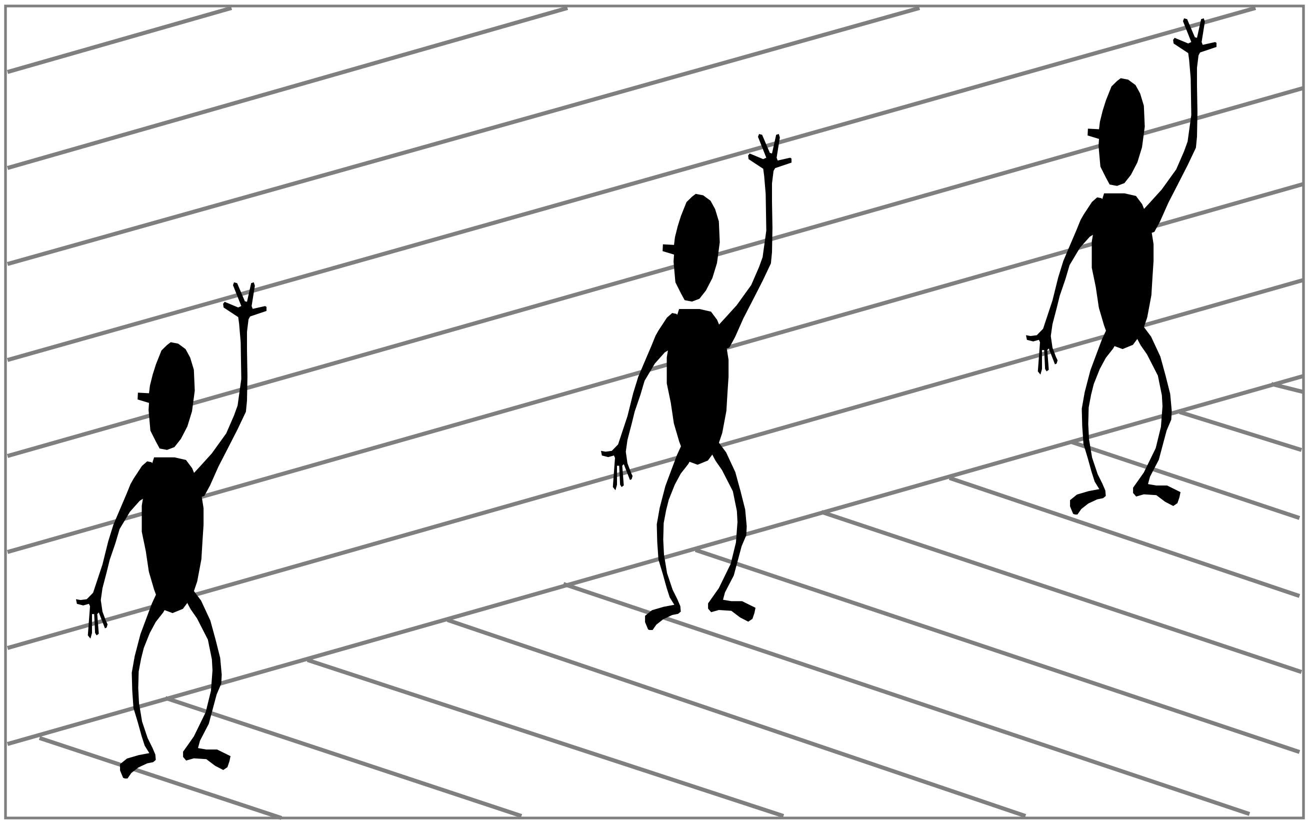


Perspective cues



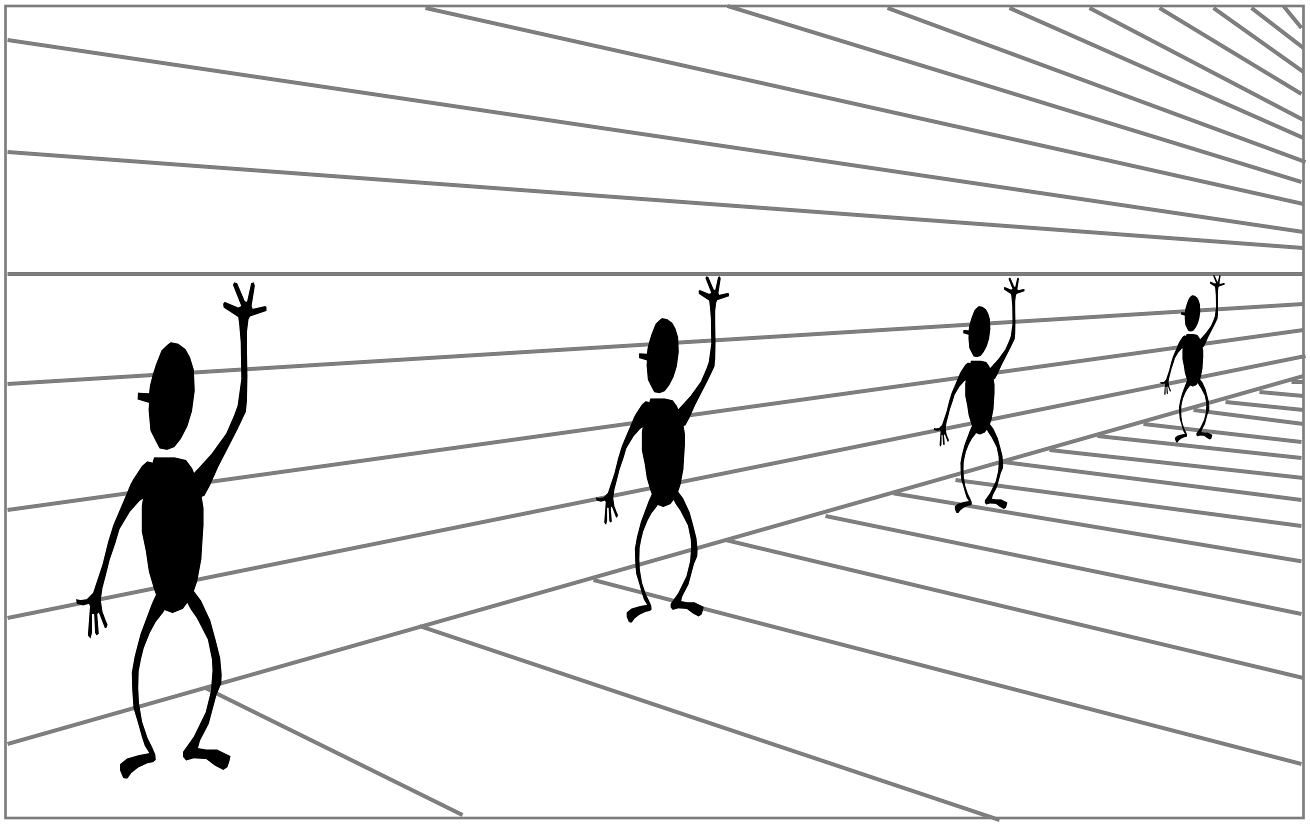
Slide credit: Steve Seitz

Perspective cues



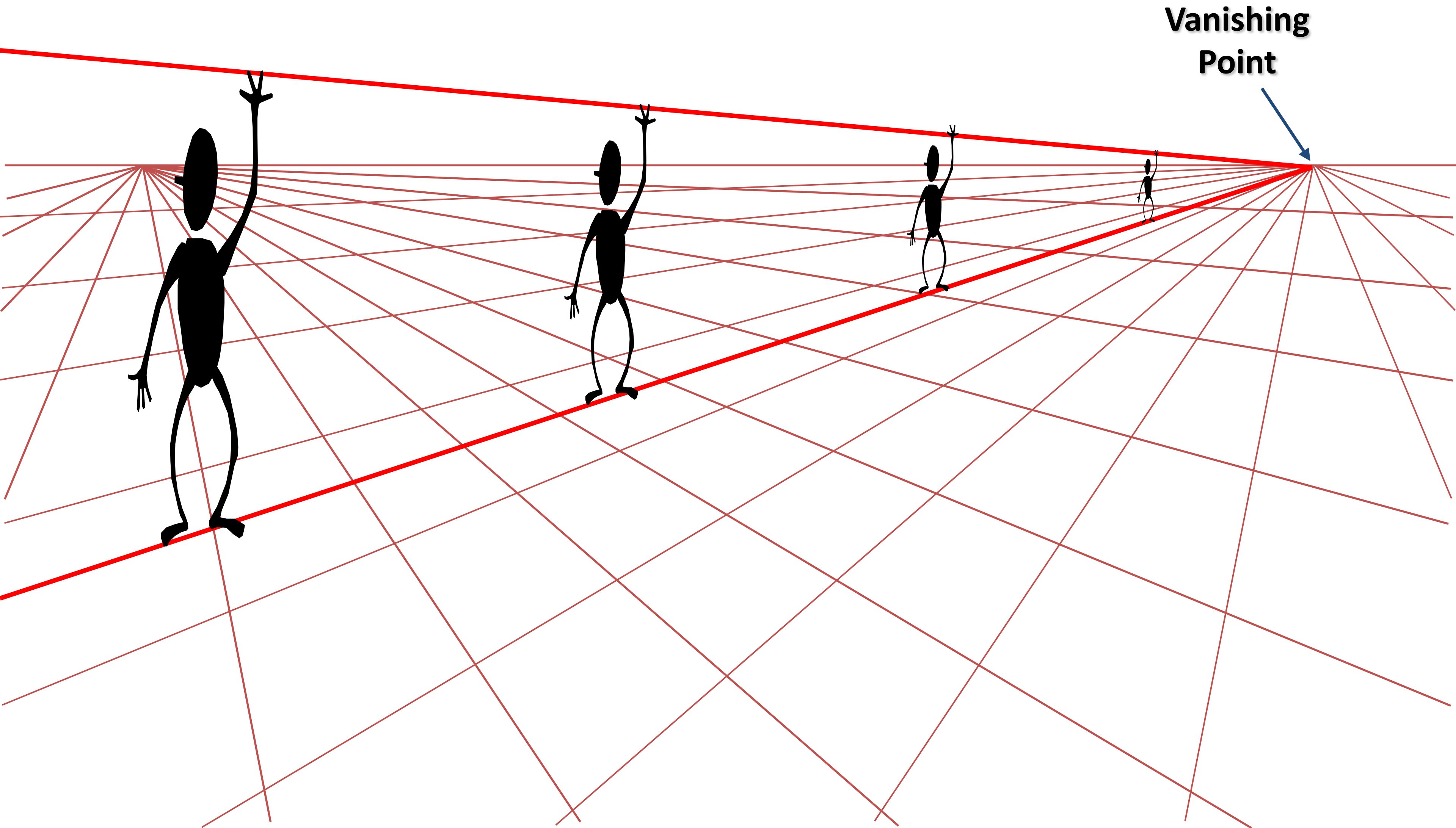
Slide credit: Steve Seitz

Perspective cues

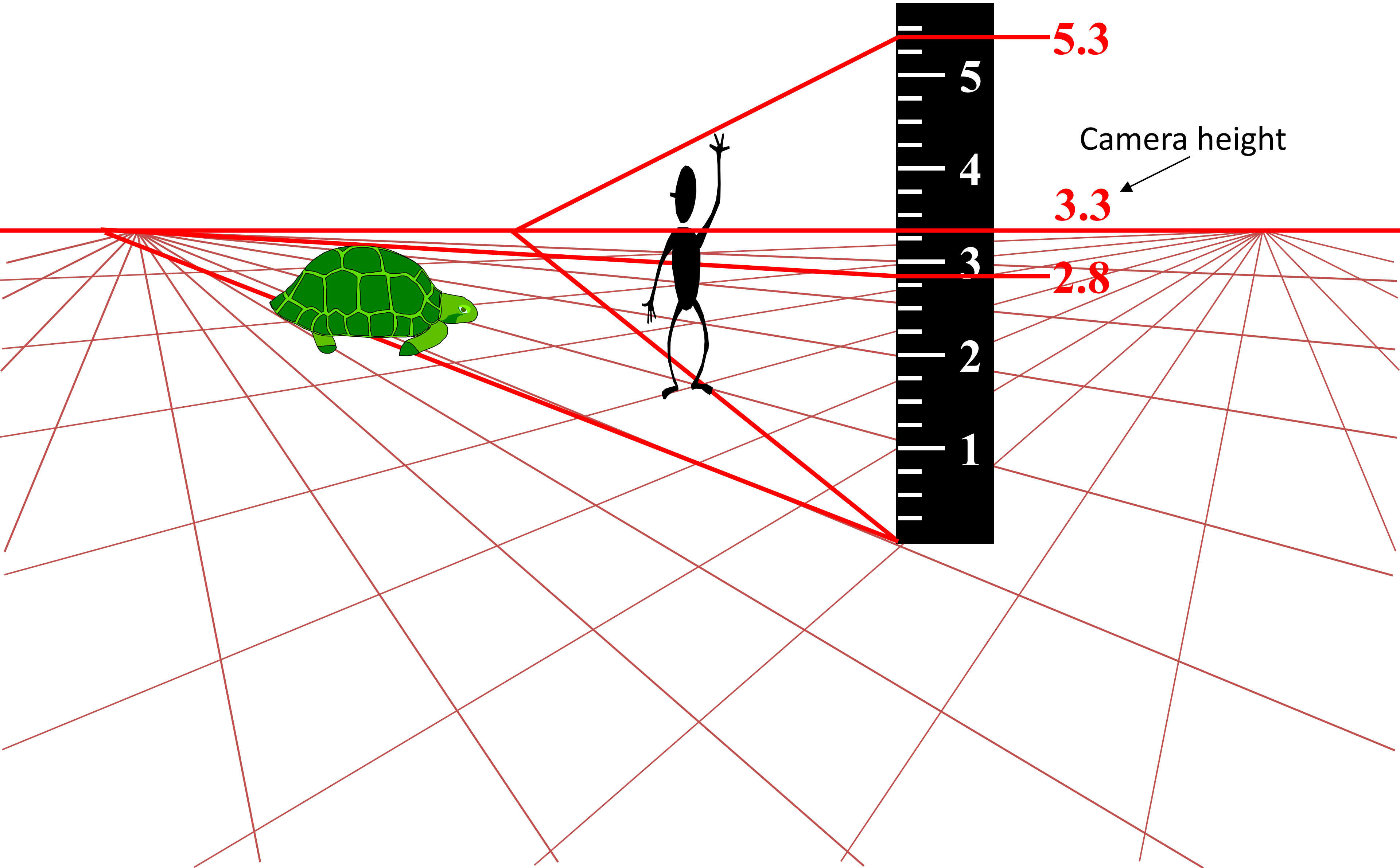


Slide credit: Steve Seitz

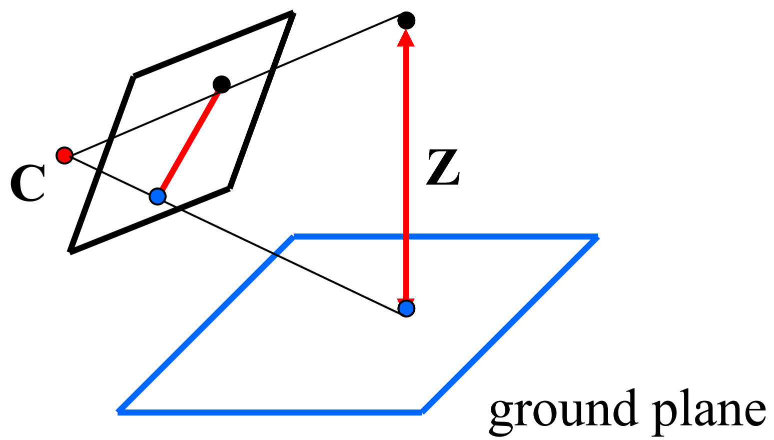
Comparing heights



Measuring height



Measuring height without a ruler

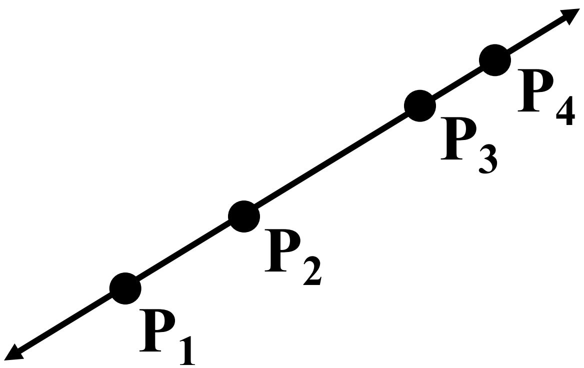


Compute Z from image measurements

The cross ratio

- A Projective Invariant
  - Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|}$$

$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

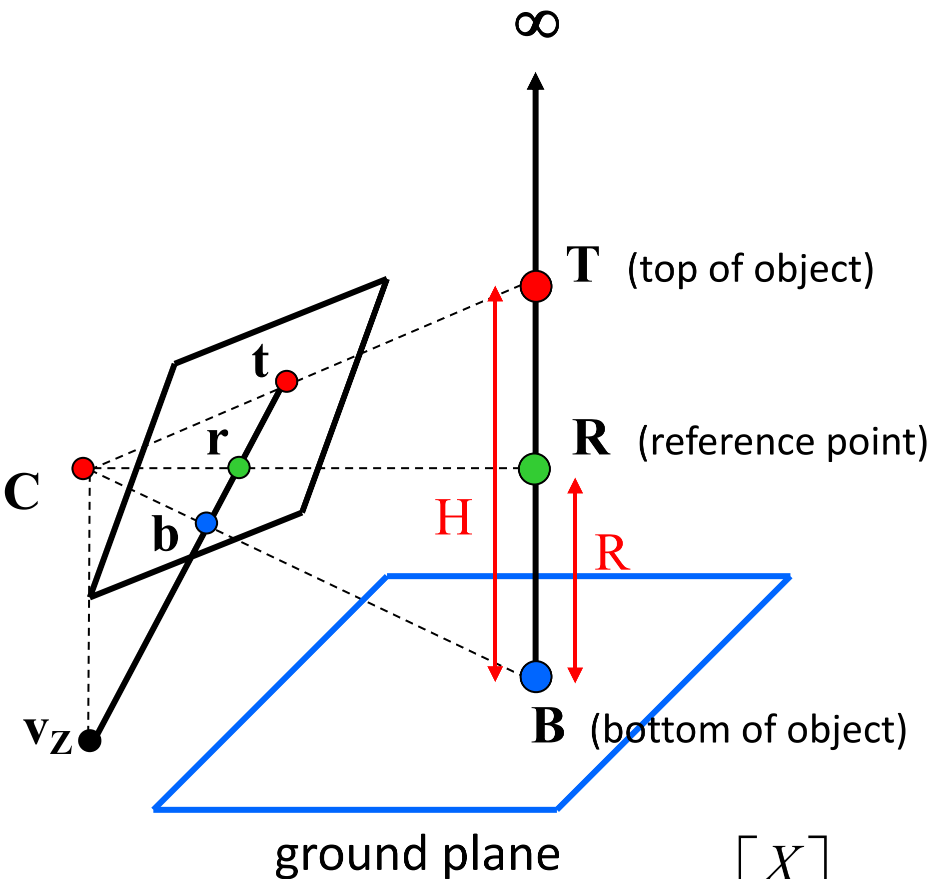
Can permute the point ordering

$$\frac{\|\mathbf{P}_1 - \mathbf{P}_3\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_3\|}$$

- 4! = 24 different orders (but only 6 distinct values)

This is the fundamental invariant of projective geometry

Measuring height



$$\frac{\|\mathbf{B}-\mathbf{T}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{B}-\mathbf{R}\|\,\|\infty-\mathbf{T}\|} = \frac{H}{R}$$

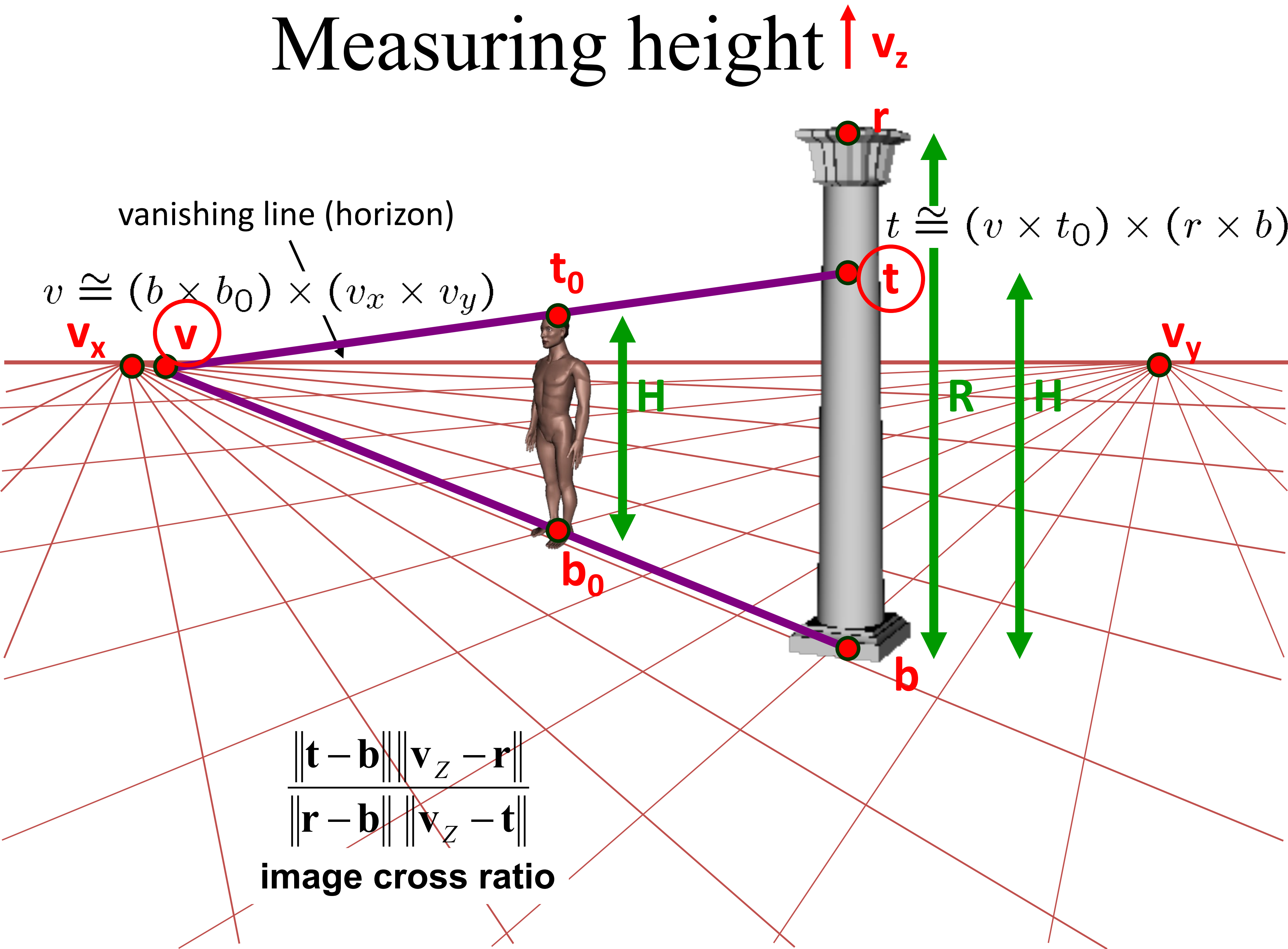
scene cross ratio

$$\frac{\|\mathbf{b}-\mathbf{t}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{b}-\mathbf{r}\|\,\|\mathbf{v}_Z-\mathbf{t}\|} = \frac{H}{R}$$

image cross ratio

scene points represented as $\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ image points as $\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

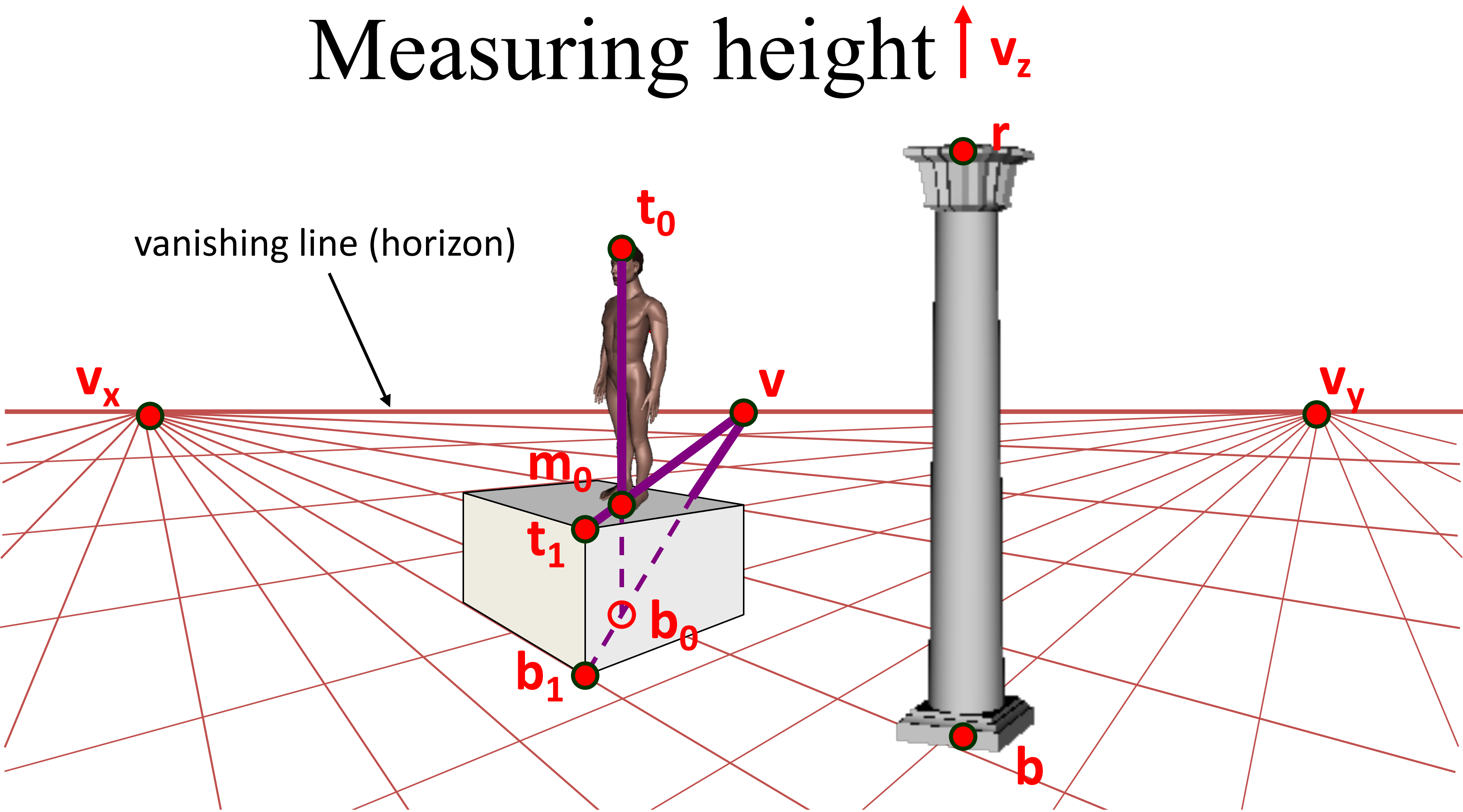
Measuring height



$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}$$

image cross ratio

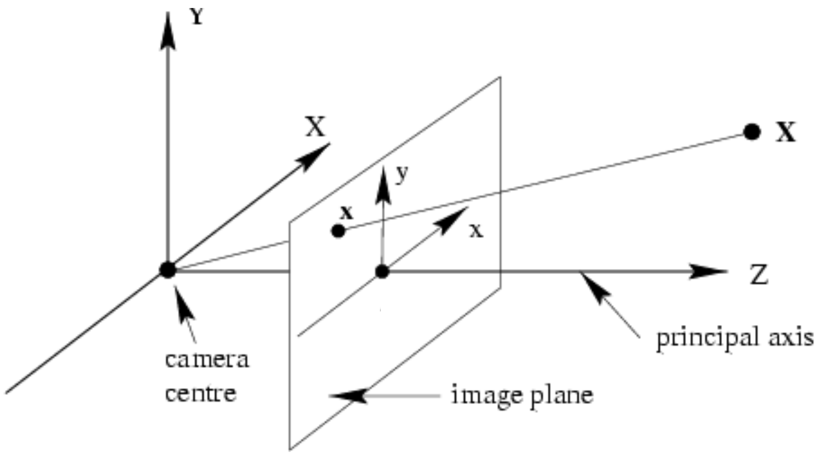
Measuring height

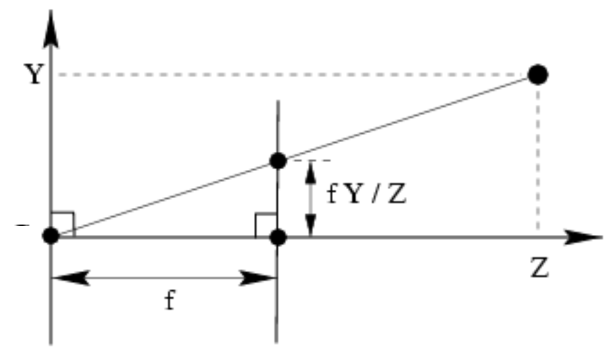


What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box, height of box is known
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

Pinhole camera model





$$(X, Y, Z) \mapsto (fX/Z, fY/Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

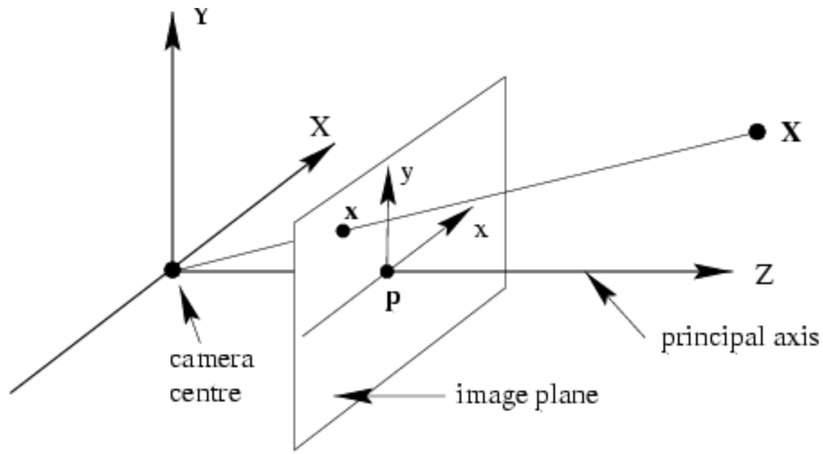
$$\mathbf{x} = \mathrm{P}\mathbf{X}$$

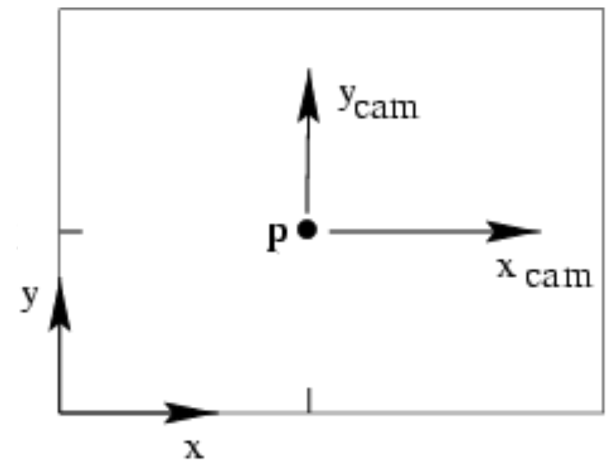
x = homogenous coordinates in the image plane

X = homogenous coordinates in the 3D scene

P = 3x4 homogeneous camera projection matrix

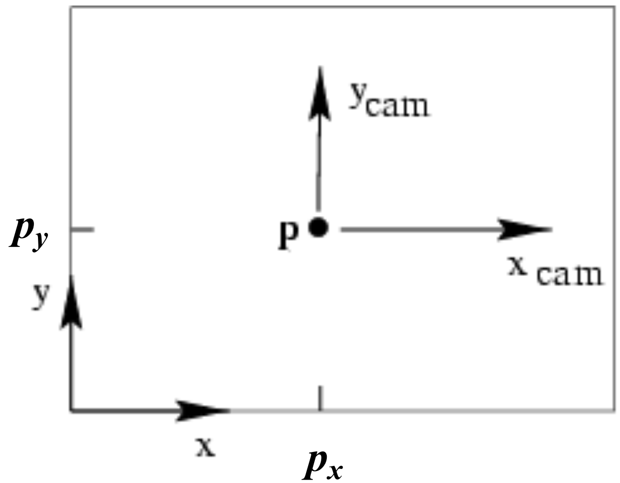
Principal point





- **Principal point (p):** point where principal axis intersects the image plane
- Normalized coordinate system: origin of the image is at the principal point
- Image coordinate system: origin is in the corner

Principal point offset

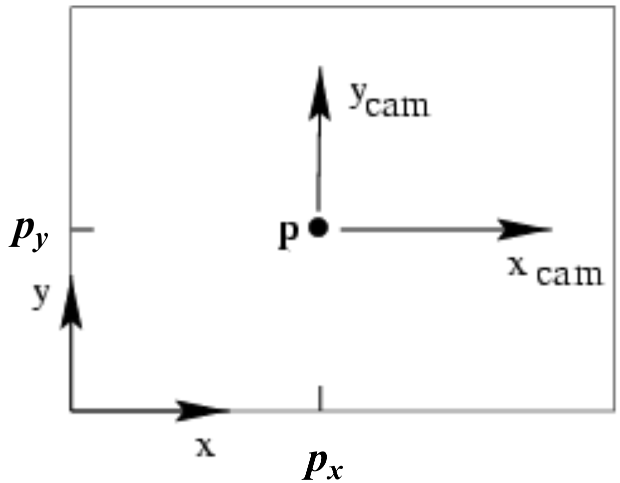


We want the principal point to map to (p_x, p_y) instead of (0,0)

$$(X, Y, Z) \mapsto (fX/Z \boxed{+p_x}, fY/Z \boxed{+p_y})$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \boxed{+Zp_x} \\ fY \boxed{+Zp_y} \\ Z \end{pmatrix} = \begin{bmatrix} f & & \boxed{p_x} & 0 \\ & f & \boxed{p_y} & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

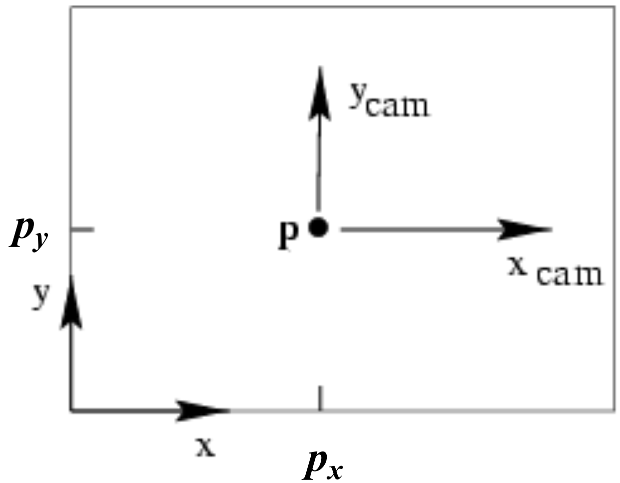
Principal point offset



principal point: (p_x, p_y)

$$\begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Principal point offset



principal point: (p_x, p_y)

$$\underbrace{\underset{\text{calibration matrix K}}{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}\underset{\text{projection matrix }[\mathrm{I}\mid 0]}{\begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix}}}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

P = 3x4 homogeneous camera projection matrix

$$\mathrm{P} = \mathrm{K}[\mathrm{I} \mid 0]$$

Camera coordinate frame

$$\mathrm{x} = \mathrm{PX} = \mathrm{K[I|0]} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K[I\,|\,0]X_{cam}$$

K - calibration matrix

$X_{cam} = (X, Y, Z, 1)^T$ = camera is assumed to be located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the Z-axis, and the point X_{cam} is expressed in this coordinate system. Such a coordinate system may be called the ***camera coordinate frame.***

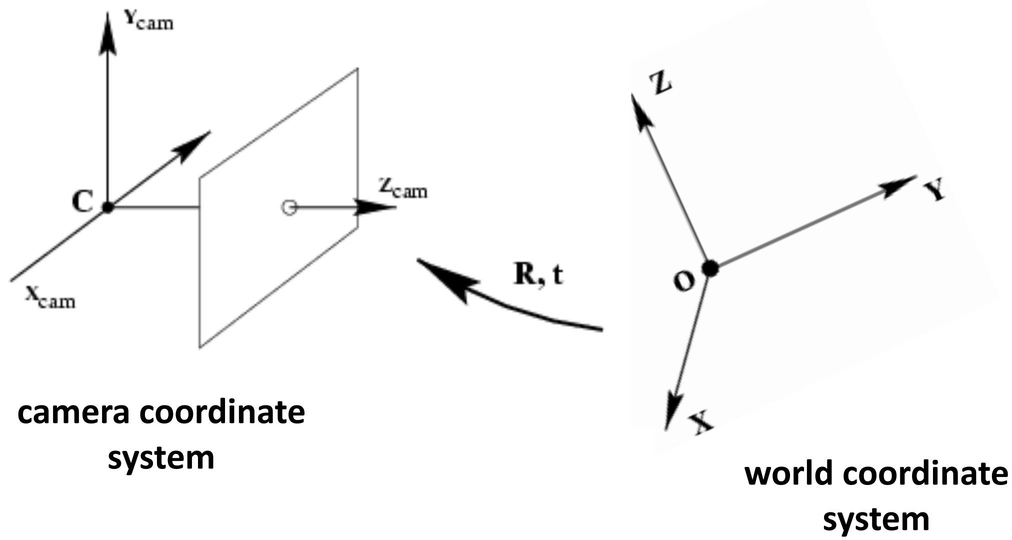
In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the ***world coordinate frame***. The two coordinate frames are related via a rotation and a translation.

World coordinate frame

$$\mathbf{x} = \mathrm{PX} = K[I\,|\,0]X_{cam}$$

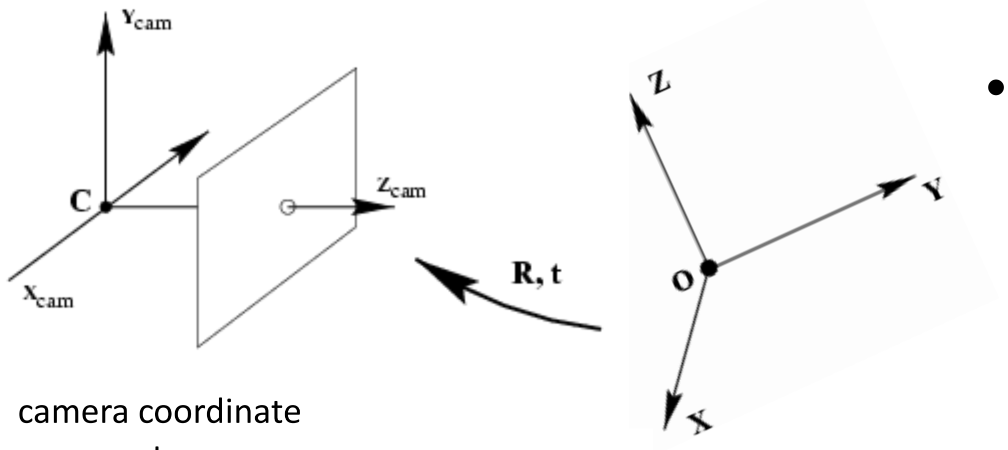
$X_{cam} = (X, Y, Z, 1)^T$ = expressed in the ***camera coordinate frame.***

In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the ***world coordinate frame***. The two coordinate frames are related via a rotation **R** and a translation **t**.



Camera calibration: figuring out transformation from *world* coordinate system to *image* coordinate system

Camera rotation and translation



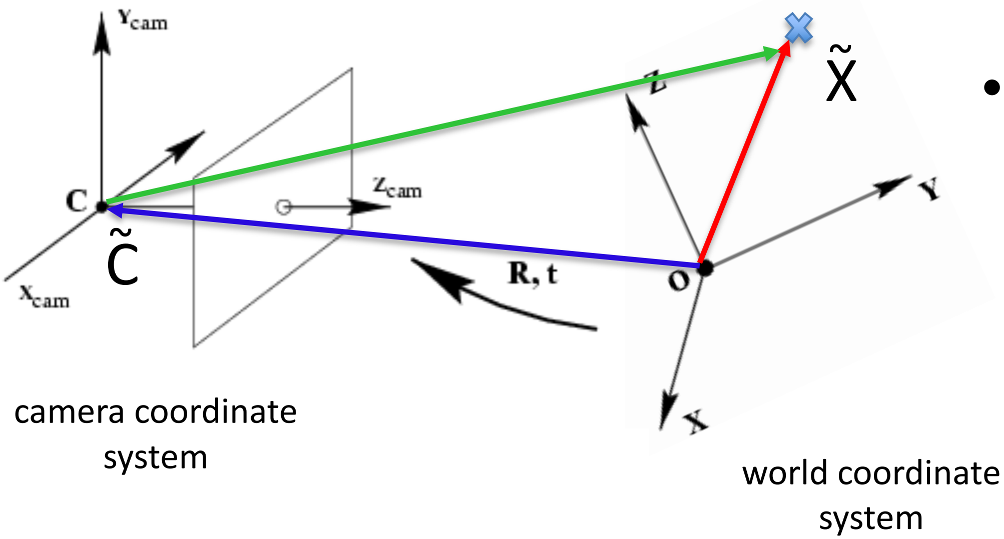
camera coordinate system

world coordinate system

- In general, the *camera* coordinate frame will be related to the *world* coordinate frame by a rotation and a translation

- Conversion from world to camera coordinate system (in non-homogeneous coordinates):

Camera rotation and translation



- In general, the *camera* coordinate frame will be related to the *world* coordinate frame by a rotation and a translation

- Conversion from world to camera coordinate system (in non-homogeneous coordinates):

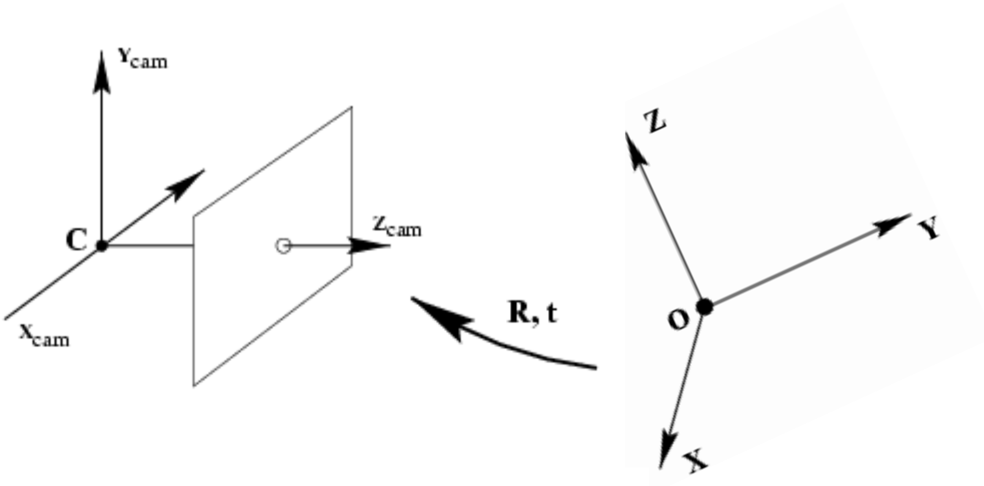
$$\widetilde{X}_{cam} = R\left(\widetilde{X} - \widetilde{C}\right)$$

coords. of point in camera frame

coords. of a point in world frame

coords. of camera center in world frame

Camera rotation and translation



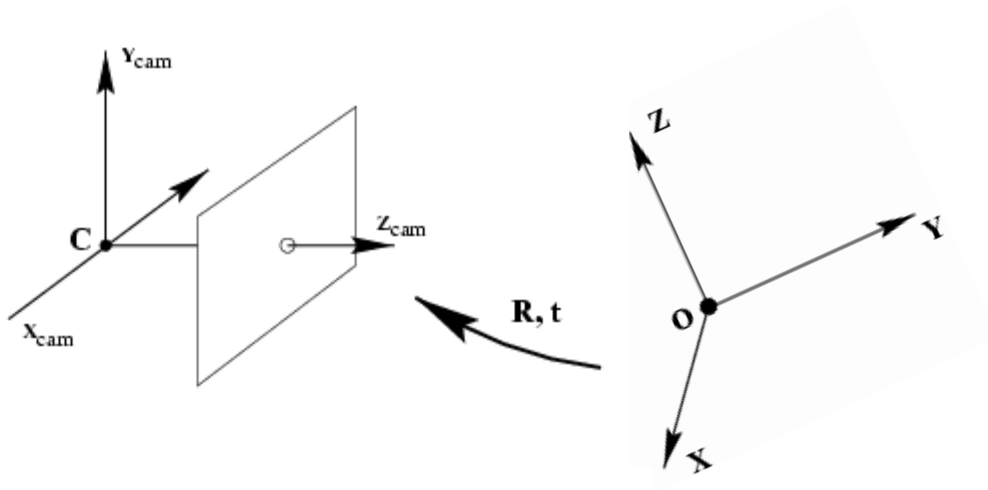
$$\widetilde{X}_{cam} = R\left(\widetilde{X} - \widetilde{C}\right)$$

$$\begin{pmatrix} \widetilde{X}_{cam} \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \widetilde{X} \\ 1 \end{pmatrix}$$

3D transformation matrix (4 x 4)

Homogeneous coordinates

Camera rotation and translation

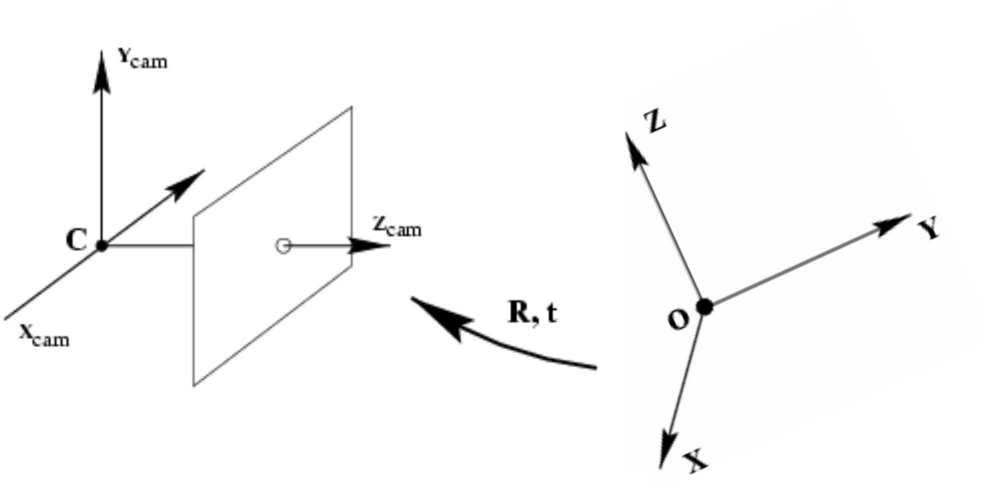


$$\widetilde{X}_{cam} = R\left(\widetilde{X} - \widetilde{C}\right)$$

$$X_{cam} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} X$$

3D transformation
matrix (4 x 4)

Camera rotation and translation



$$x = PX = K[I\,|\,0]X_{cam}$$

$$x = K[I\,|\,0]\begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

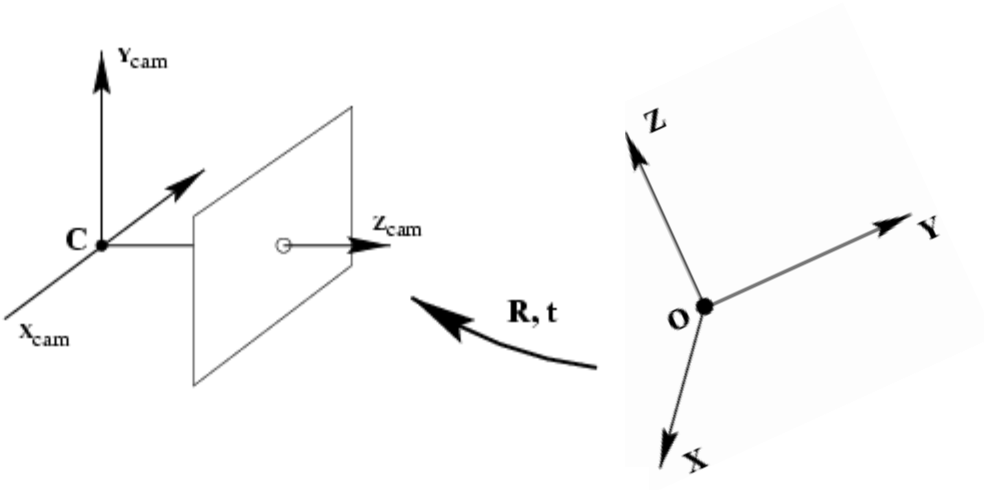
2D transformation matrix (3 x 3)

perspective projection matrix (3 x 4)

3D transformation matrix (4 x 4)

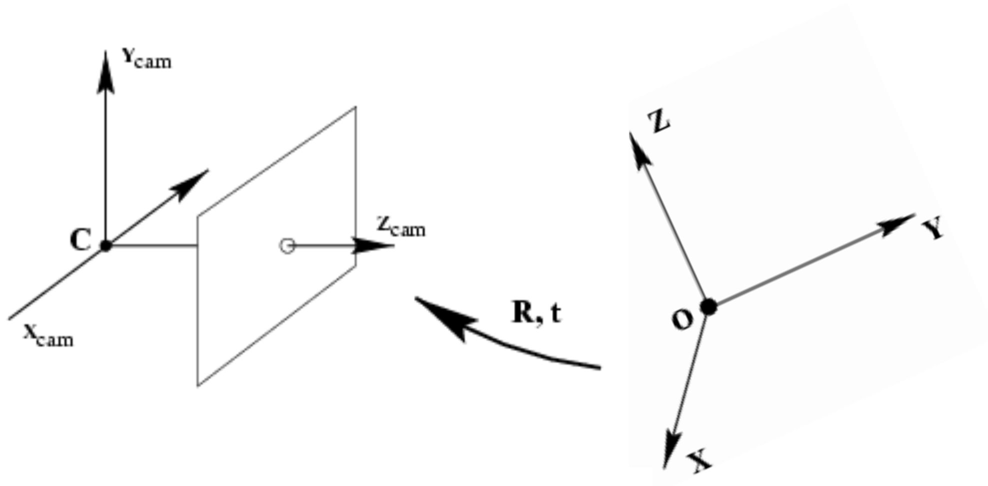
calibration matrix K

Camera rotation and translation



$$x = K\left[R \mid -R\tilde{C}\right]X$$

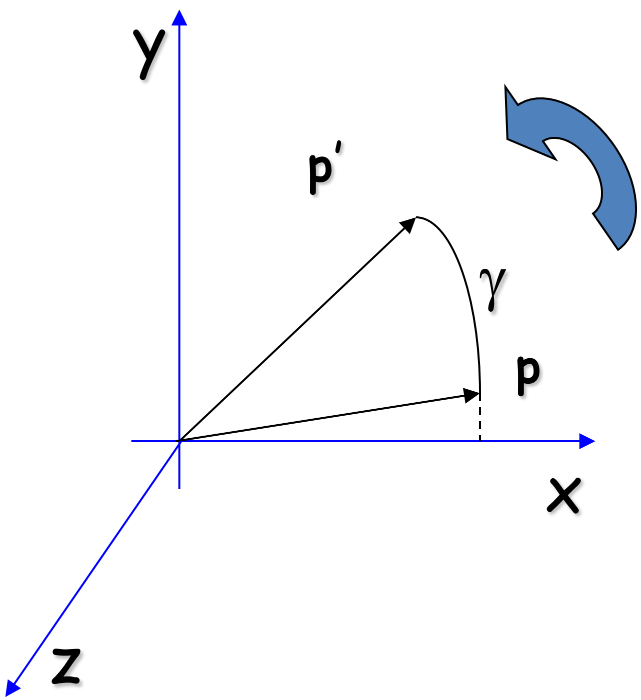
Camera rotation and translation



$$x = K[R \mid t]X \qquad t = -R\tilde{C}$$

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



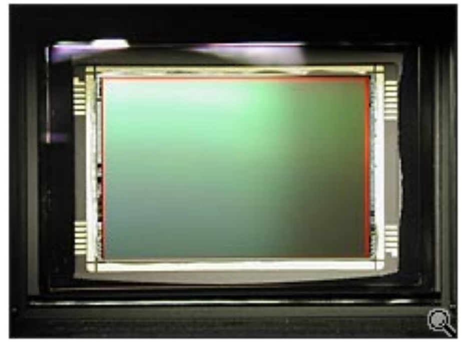
$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

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

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

Slide Credit: Saverese

Pixel coordinates for CCD cameras



Pixel size: $\dfrac{1}{m_x} \times \dfrac{1}{m_y}$

- m_x pixels per meter in horizontal direction, m_y pixels per meter in vertical direction

$$K = \underbrace{\begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}}_{\text{pixels/m}} \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\text{m}} = \underbrace{\begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}}_{\text{pixels}}$$

Skew parameter

$$K = \begin{bmatrix} \alpha_x & s & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

- s skew- parameter
- usually s = 0
- s ≠ 0 can be interpreted as a skewing of the pixel elements in the CCD array so that the x- and y-axes are not perpendicular. This is very unlikely to happen (might arise as a result of taking a image of an image)

7.4 Radial distortion

The assumption throughout these chapters has been that a linear model is an accurate model of the imaging process. Thus the world point, image point and optical centre are collinear, and world lines are imaged as lines and so on. For real (non-pinhole) lenses this assumption will not hold. The most important deviation is generally a radial distortion. In practice this error becomes more significant as the focal length (and price) of the lens decreases. See figure 7.4.

The cure for this distortion is to correct the image measurements to those that would have been obtained under a perfect linear camera action. The camera is then effectively again a linear device. This process is illustrated in figure 7.5. This correction must

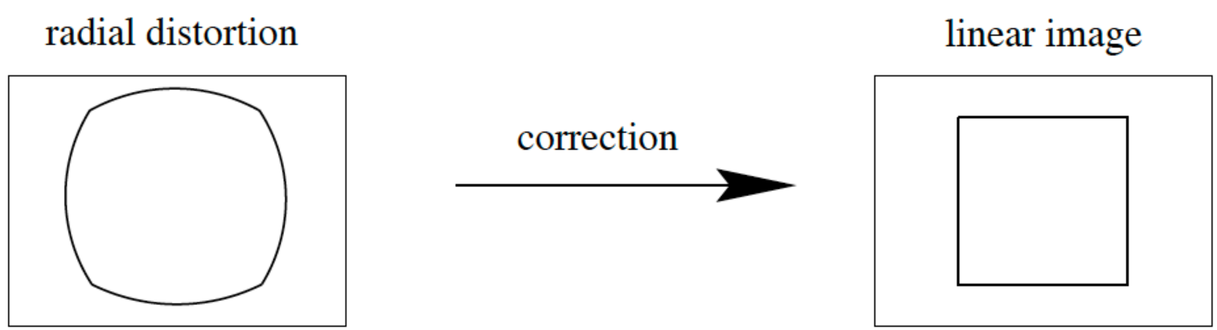


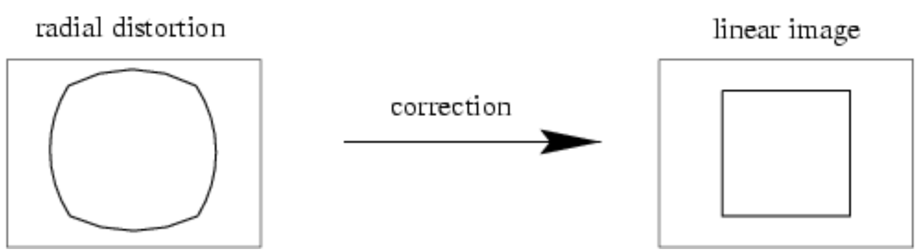
Fig. 7.5. *The image of a square with significant radial distortion is corrected to one that would have been obtained under a perfect linear lens.*

be carried out in the right place in the projection process. Lens distortion takes place during the initial projection of the world onto the image plane, according to (6.2–p154). Subsequently, the calibration matrix (7.6) reflects a choice of affine coordinates in the image, translating physical locations in the image plane to pixel coordinates.

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$



Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*
- Extrinsic parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}}\end{bmatrix}$
  - Rotation and translation relative to world coordinate system
  - What is the projection of the camera center?

coords. of camera center in world frame

$$\mathbf{PC} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{C}} \\ 1\end{bmatrix} = 0$$

The camera center is the *null space* of the projection matrix!

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

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

$$\Downarrow$$

$$\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

Degrees of freedom

$$\mathbf{x} = \mathbf{K}\left[\mathbf{R} \quad \mathbf{t}\right]\mathbf{X}$$

6 (3 for rotation and 3 for translation)

5

$$\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

5 + 6 = 11 degrees of freedom = 12 (matrix 3 x 4) – 1 (scale)

How to calibrate the camera?

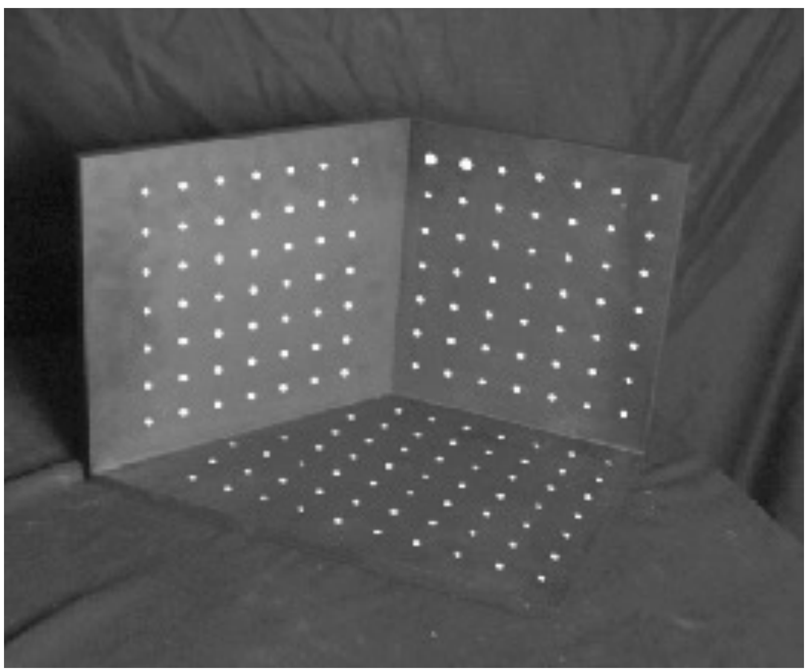
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

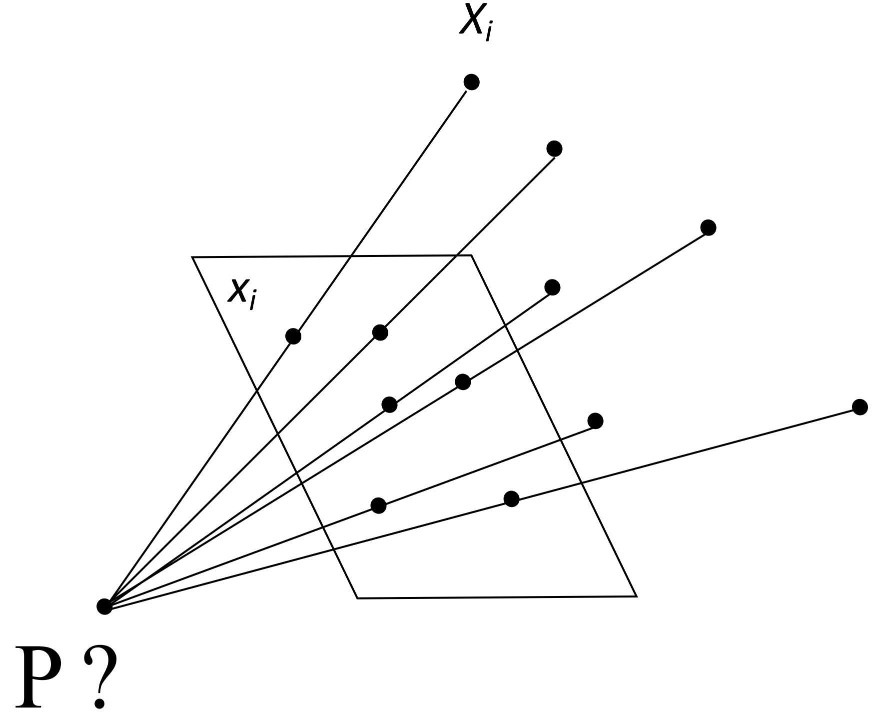
$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Calibrating the Camera

Method 1: Use an object (calibration grid) with known geometry

- Given n points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters

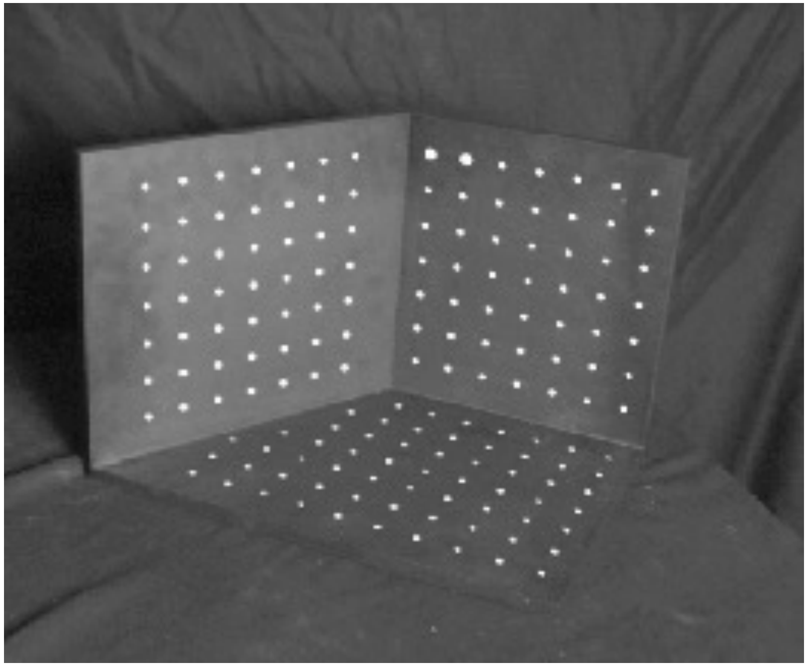




Calibrating the Camera

Method 1: Use an object (calibration grid) with known geometry

- Given n points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters
- Get least squares solution (or non-linear solution)



$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Linear method

- Solve using linear least squares

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- P has 11 degrees of freedom
- One 2D/3D correspondence gives us two linearly independent equations
  - 6 correspondences needed for a minimal solution

Linear method

- Solve using linear least squares

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(X_i, Y_i, Z_i) – 3D world coordinates

(u_i, v_i) – 2D image coordinates

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Ap=0 form

Linear method

- Solve using linear least squares

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Ap=0 form

- Homogeneous least squares: find p minimizing $||Ap||^2$
  - Solution given by the eigenvector of A^TA with smallest eigenvalue

Laboratory class 6

The goal of this lab is to introduce you to camera and scene geometry. Specifically we will estimate the camera projection matrix, which maps 3D world coordinates to image coordinates, as well as the fundamental matrix, which relates points in one scene to epipolar lines in another. The camera projection matrix and the fundamental matrix can each be estimated using point correspondences. To estimate the projection matrix (camera calibration), the input is corresponding 3D and 2D points. To estimate the fundamental matrix the input is corresponding 2D points across two images. We start by estimating the projection matrix and the fundamental matrix for a scene with ground truth correspondences. Then we'll move on to estimating the fundamental matrix using point correspondences from SIFT.

Tutorial on epipolar geometry is here: https://docs.opencv.org/master/da/de9/tutorial_py_epipolar_geometry.html or here: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_epipolar_geometry/py_epipolar_geometry.html

Data

We provide 2D and 3D ground truth point correspondences for the base image pair (pic_a.jpg and pic_b.jpg), as well as other images which will not have any ground truth dataset.

Image 1 of a scene	Image 2 of a scene
	

This laboratory consists of three parts: (1) estimating the projection matrix, (2) estimating the fundamental matrix, (3) estimating the fundamental matrix with unreliable SIFT matches using RANSAC.

Laboratory class 6

Laboratory class 6