Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2020-2021

Course structure

1. **Features and filters: low-level vision**
   Linear filters, color, texture, edge detection

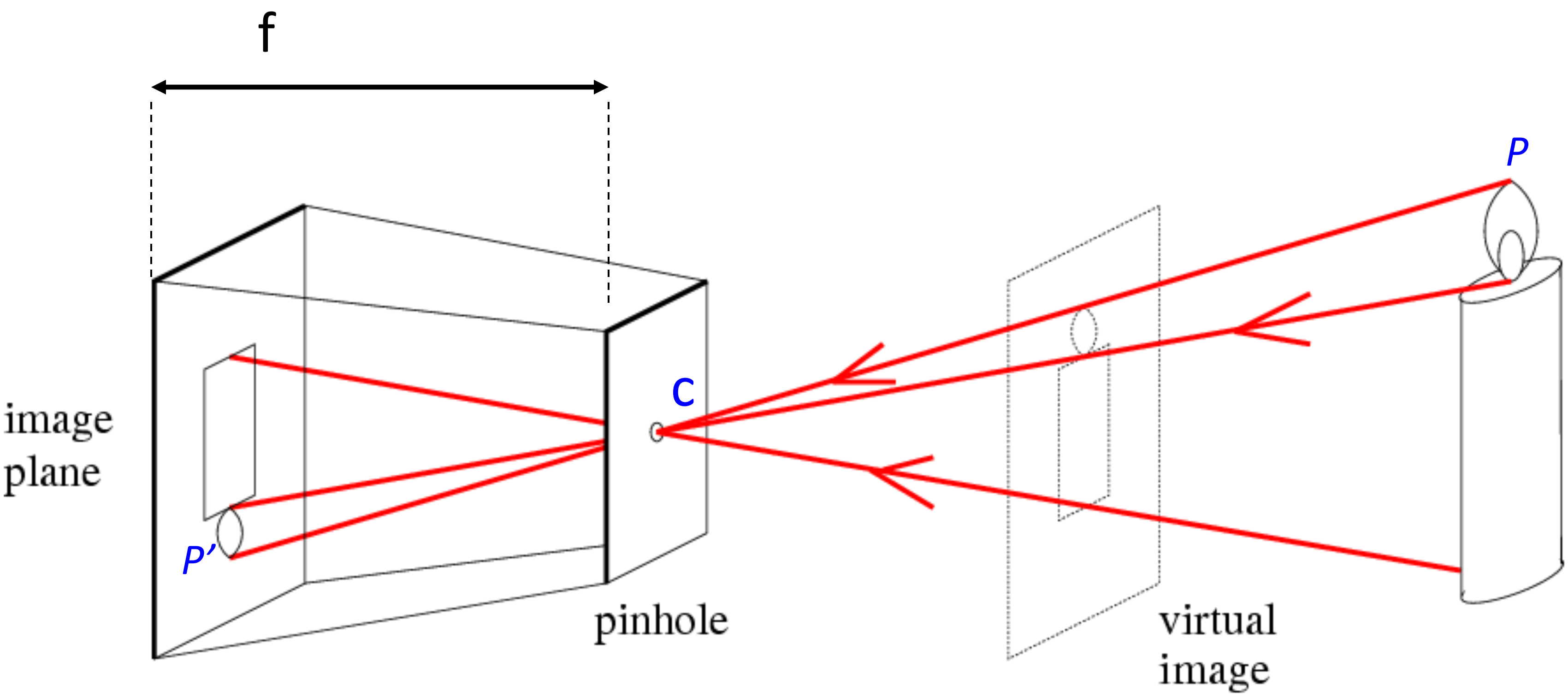
2. **Grouping and fitting: mid-level vision**
   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

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

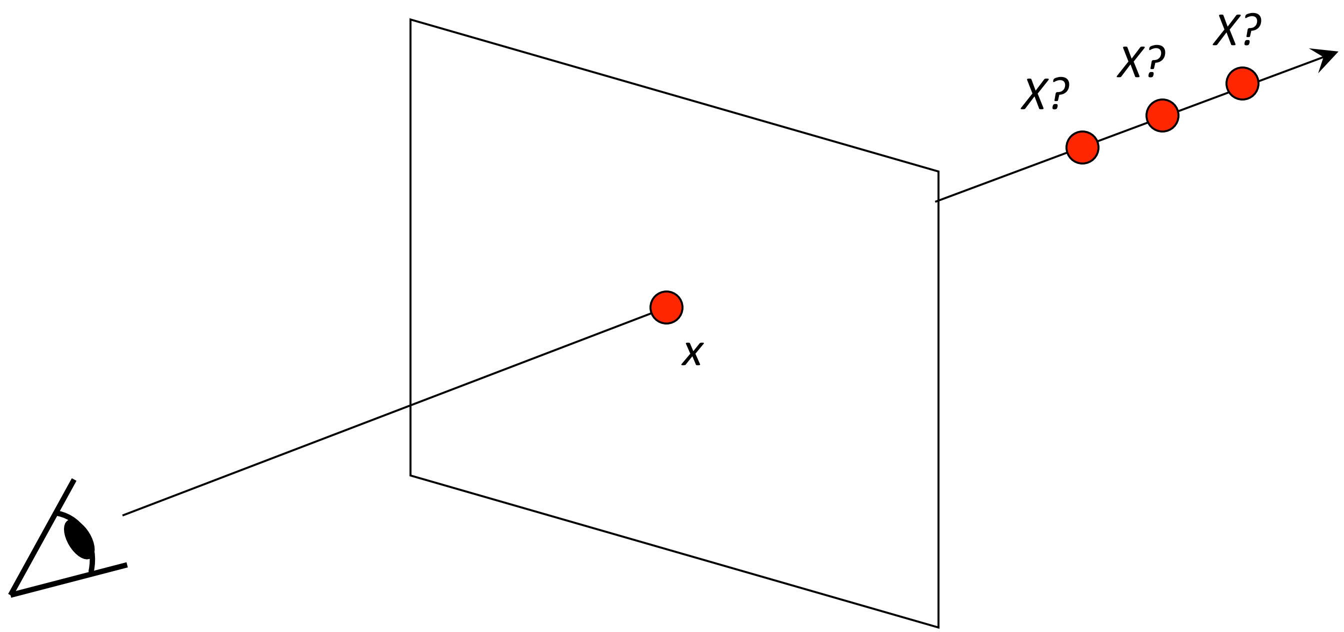
Recap: Pinhole projection model



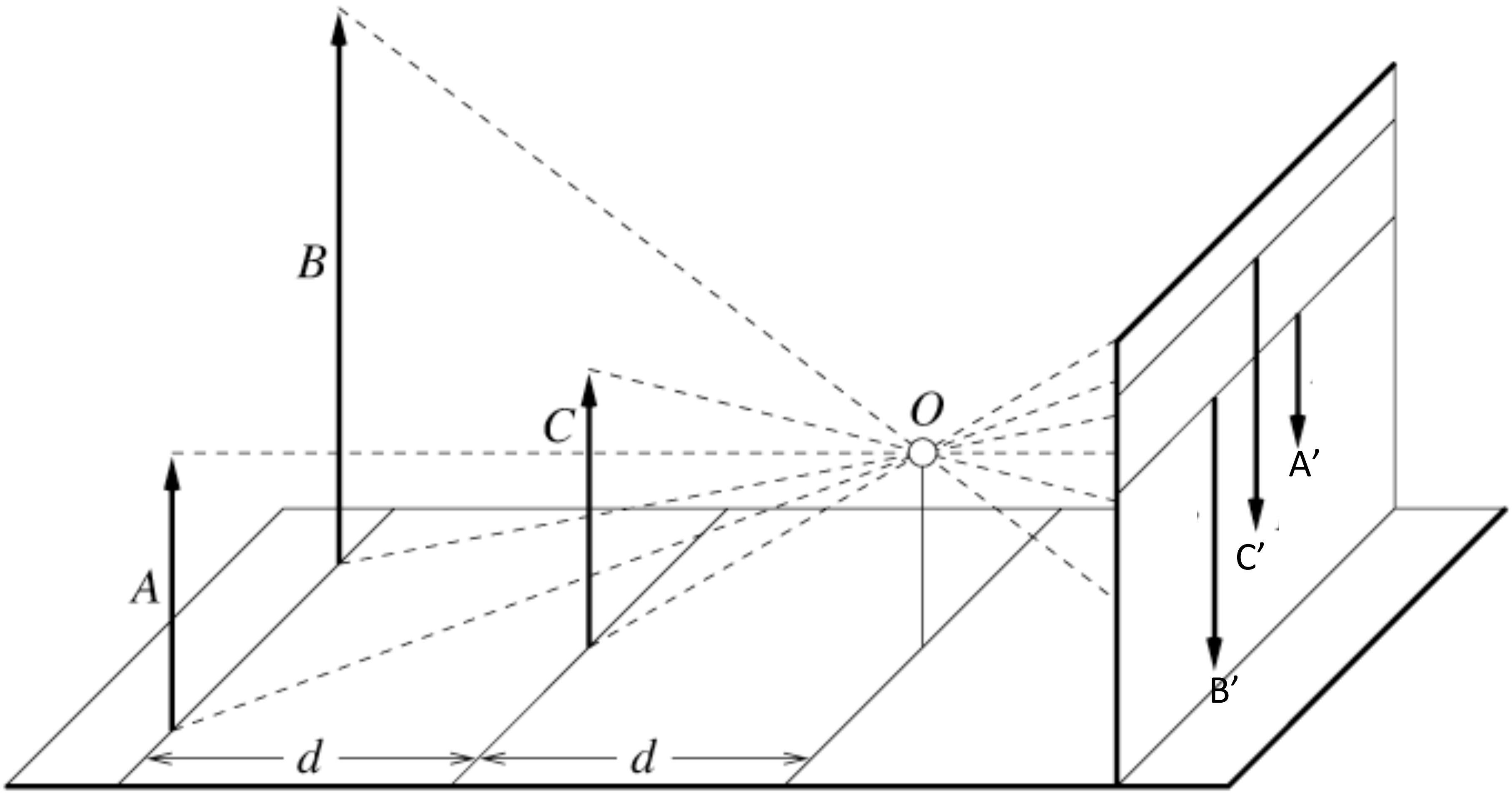
- To compute the projection *P'* of a scene point *P*, form the **visual ray** connecting *P* to the **camera center** *C* and find where it intersects the **image plane**

Slide credit D. Forsyth

Recap: Single-view ambiguity



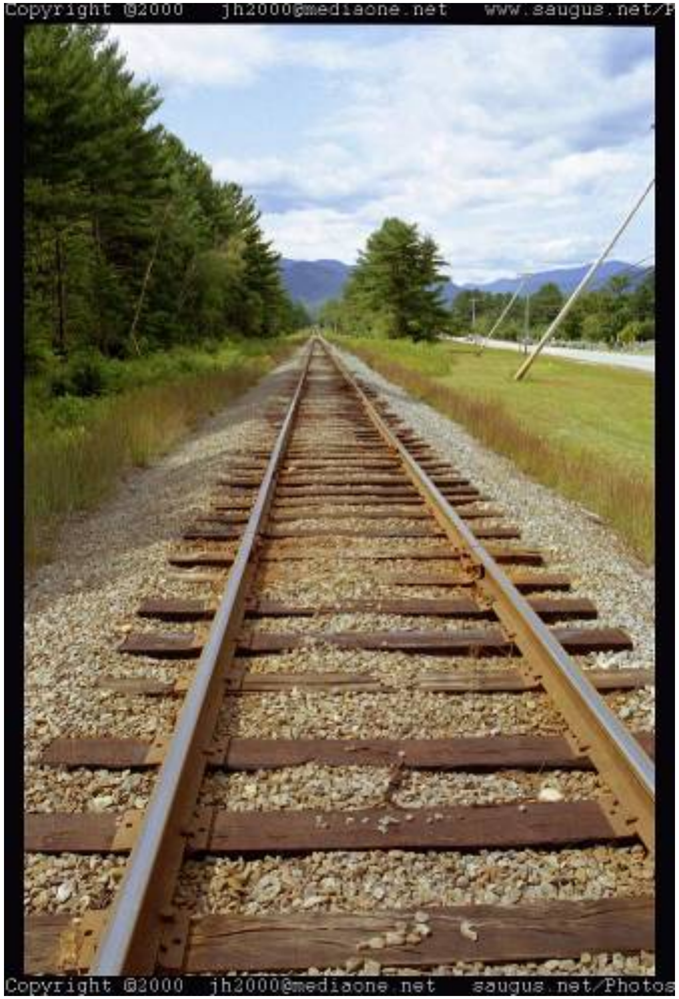
Recap: Length is not preserved



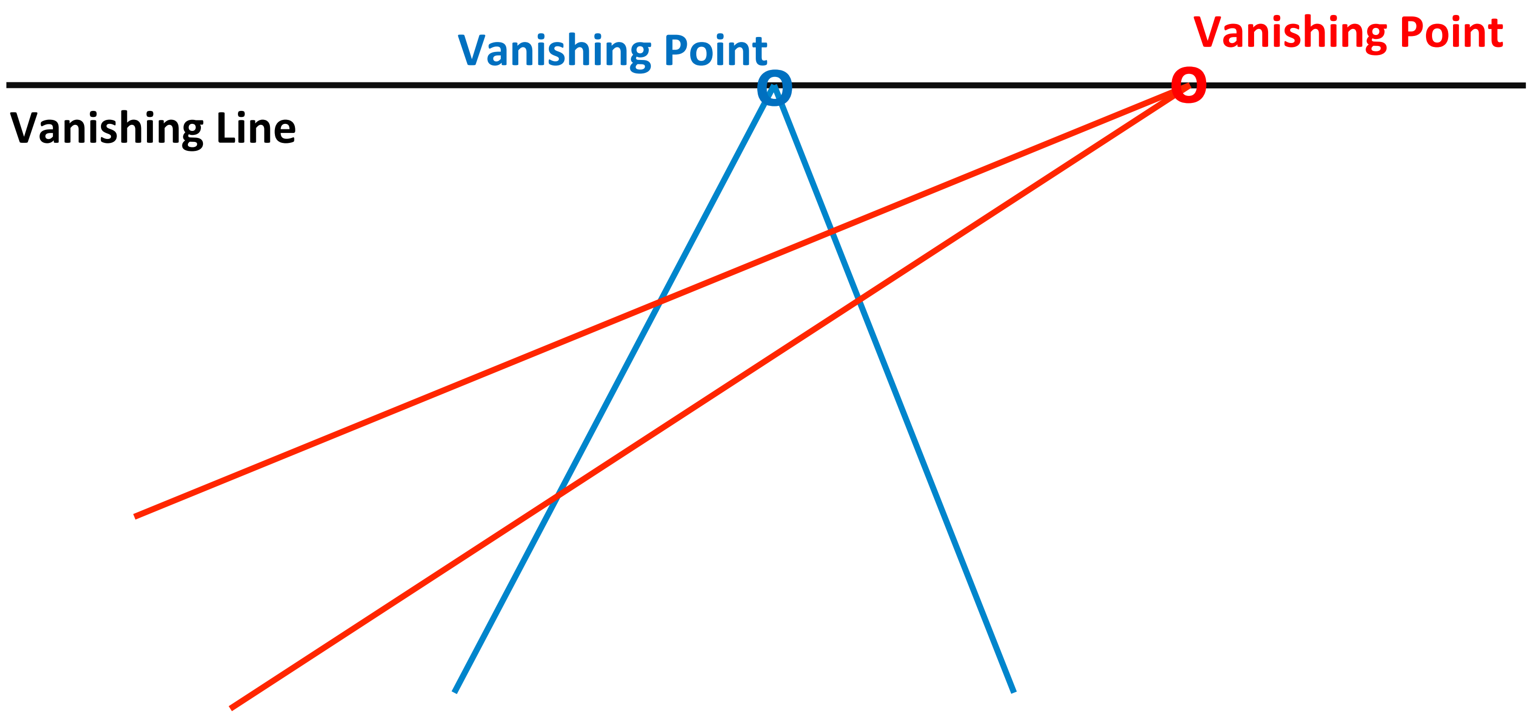
Slide credit D. Forsyth

Vanishing points and lines

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

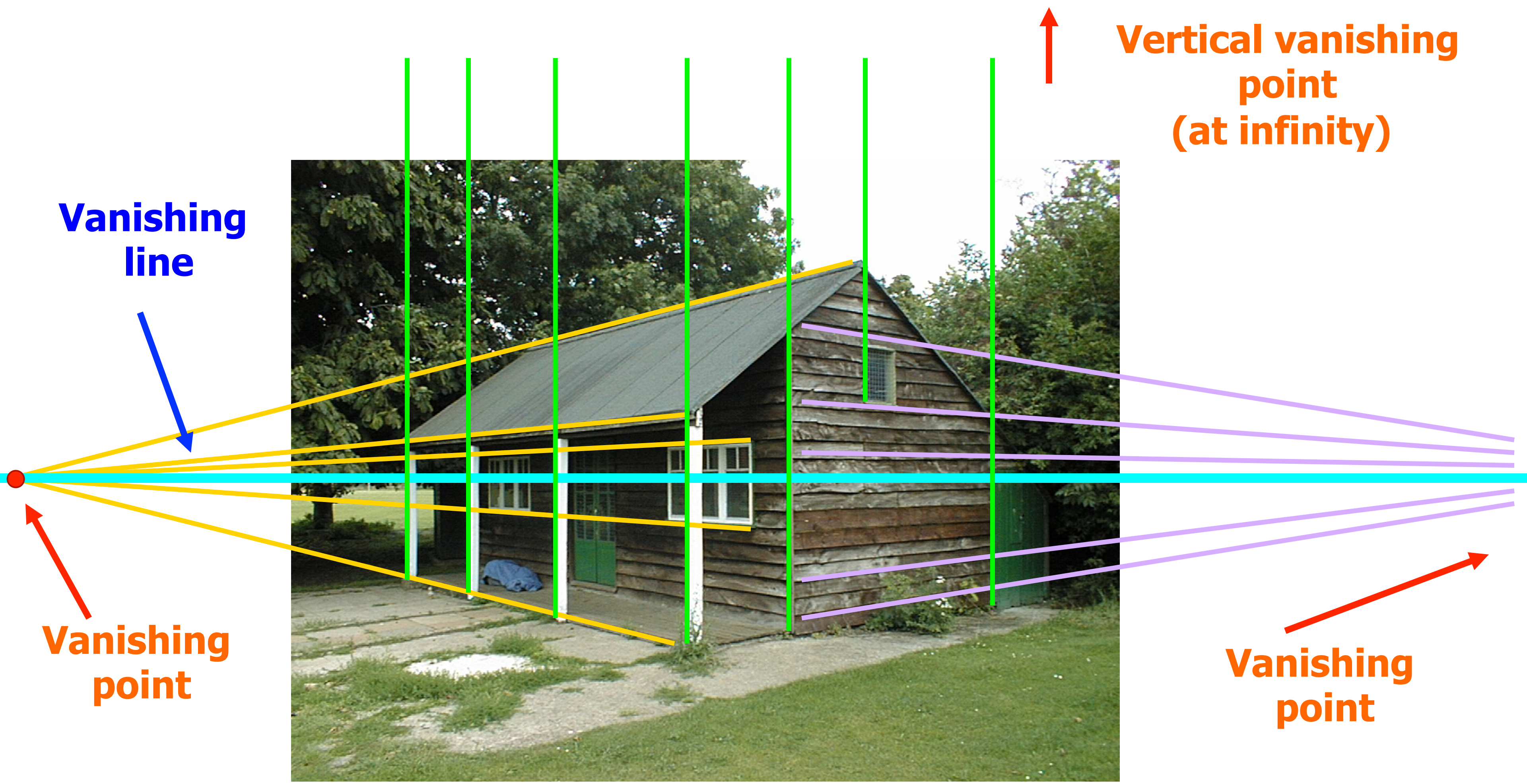


Vanishing points and lines



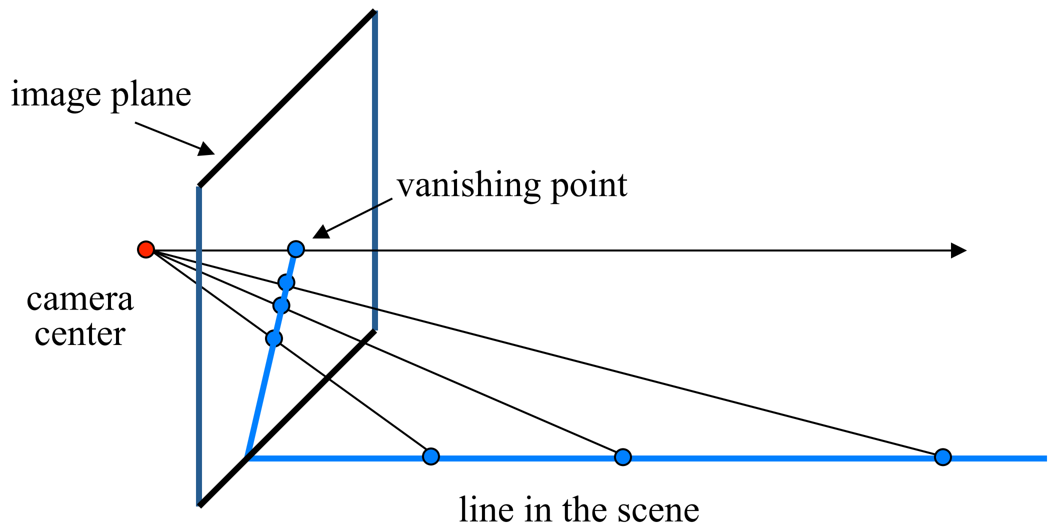
- The projections of parallel 3D lines intersect at a **vanishing point**
- The projection of parallel 3D planes intersect at a **vanishing line**
- If a set of parallel 3D lines are also parallel to a particular plane, their vanishing point will lie on the vanishing line of the plane
- Not all lines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

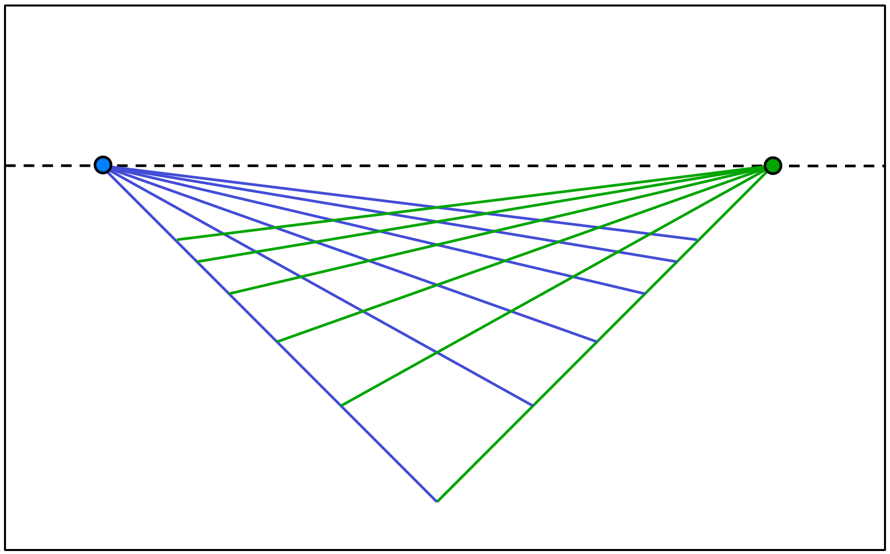
Slide by Steve Seitz

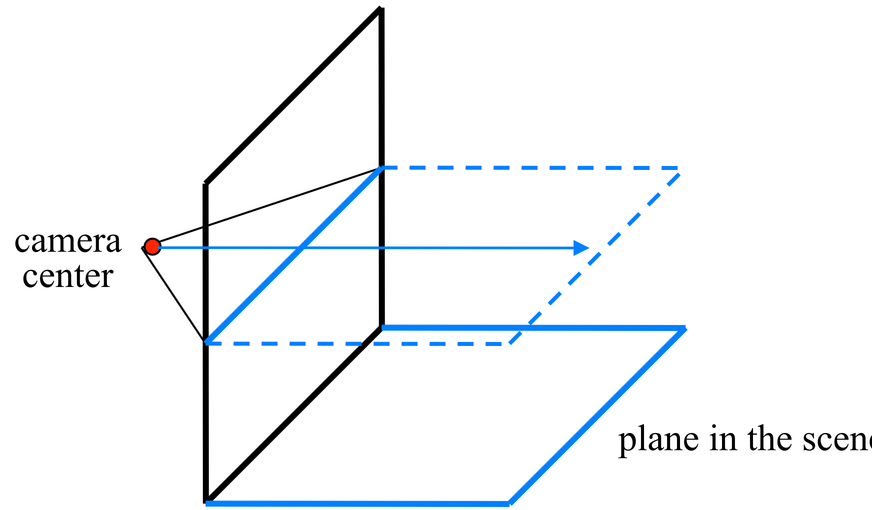
Vanishing lines of planes

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

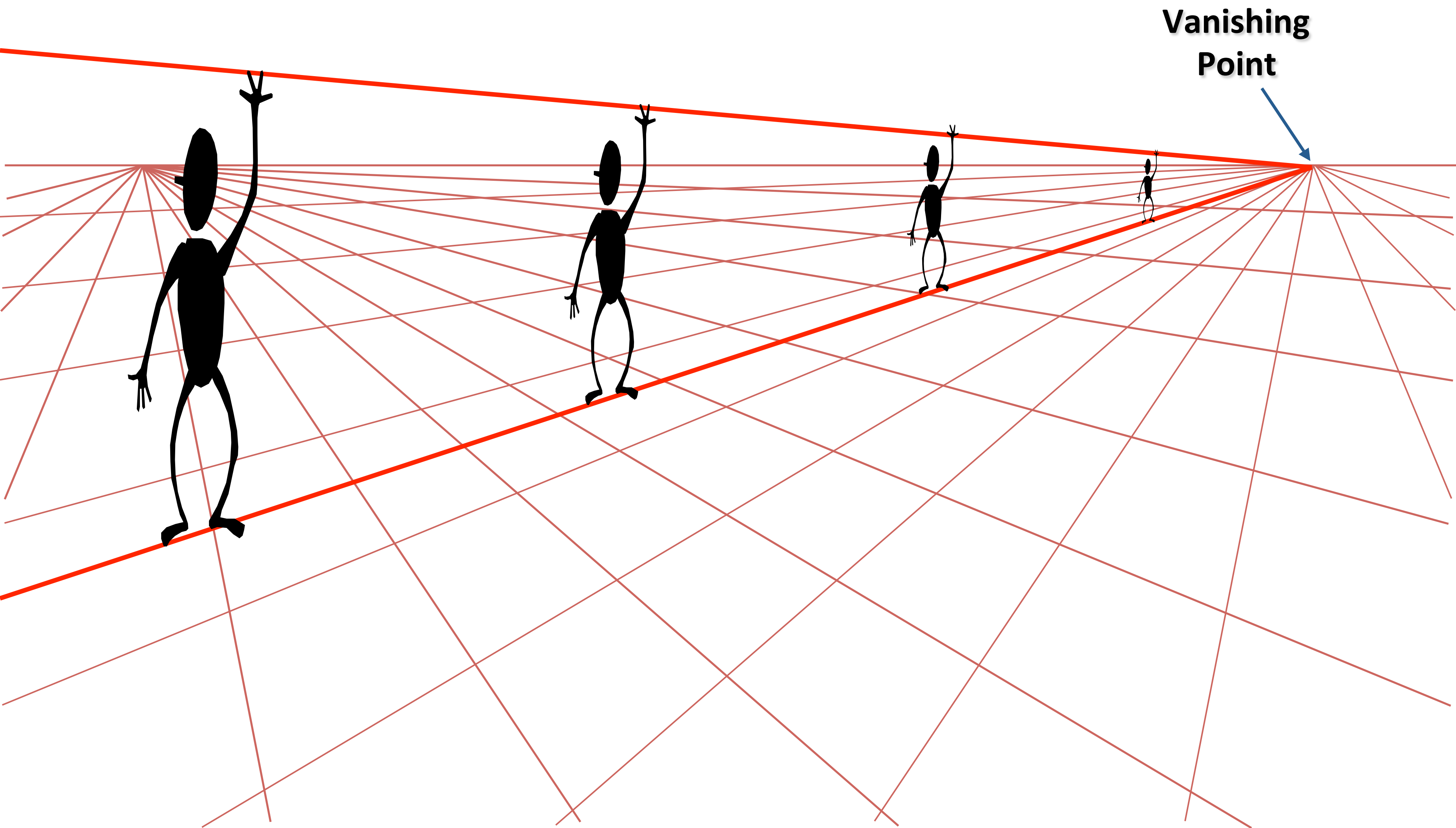
Vanishing lines of planes

Is the parachutist above or below the camera?

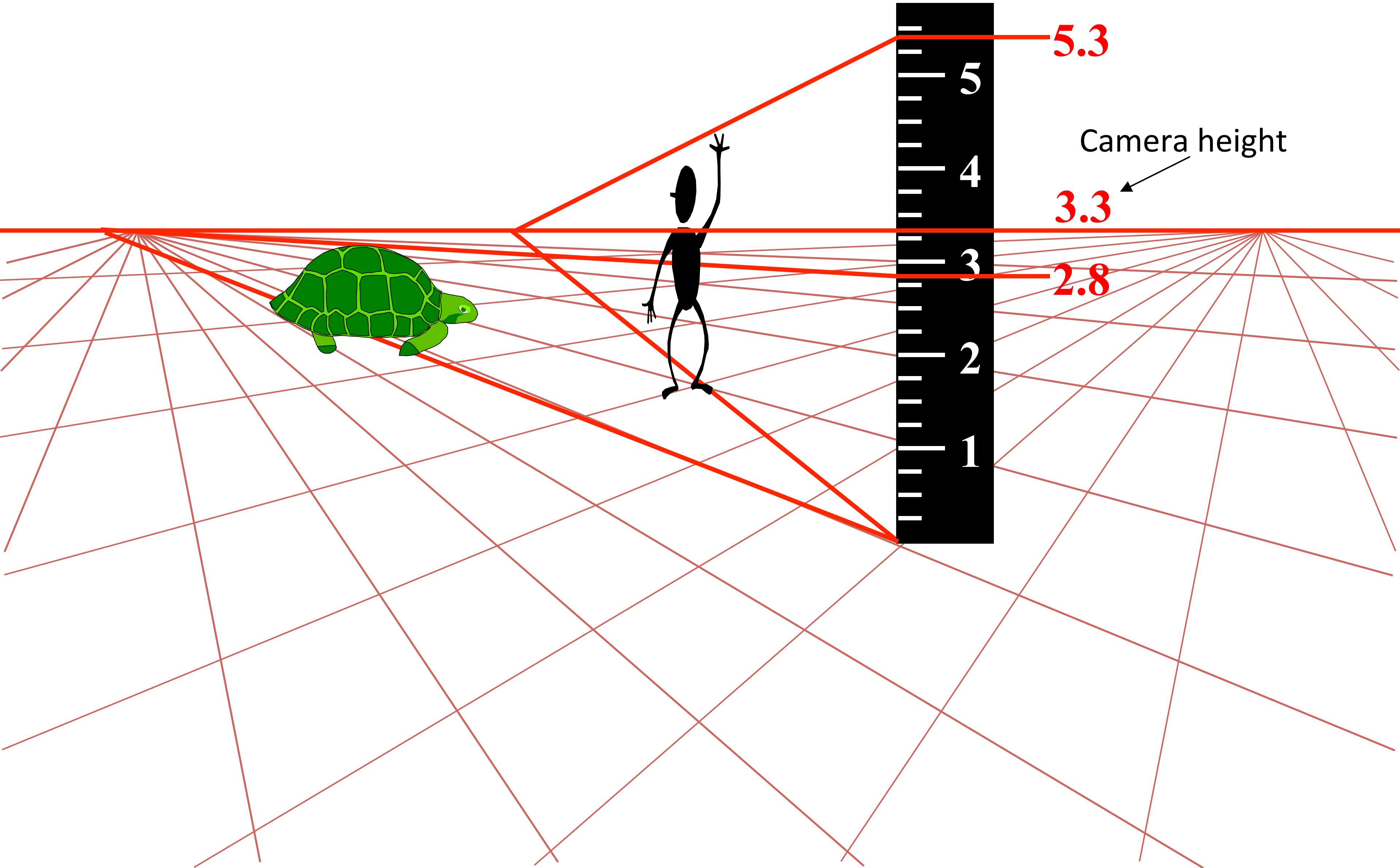
A: The parachutist is below the horizon, so is below the camera

Slide credit Steve Seitz

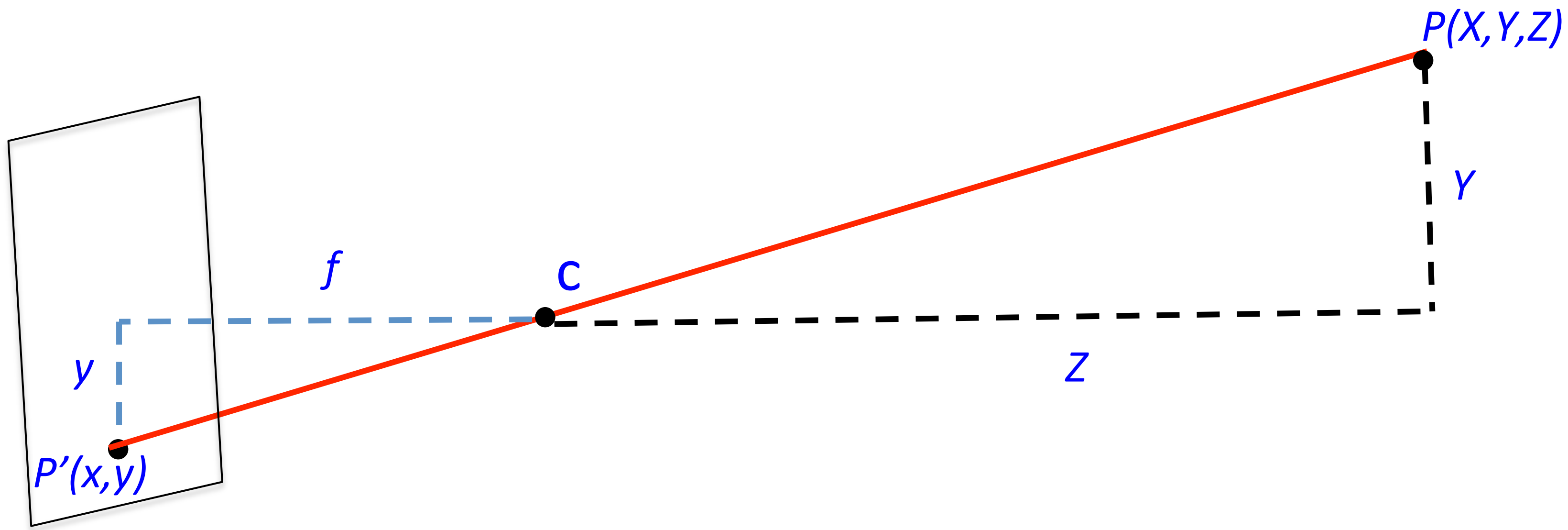
Comparing heights



Measuring height



Pinhole projection model



- 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: 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

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

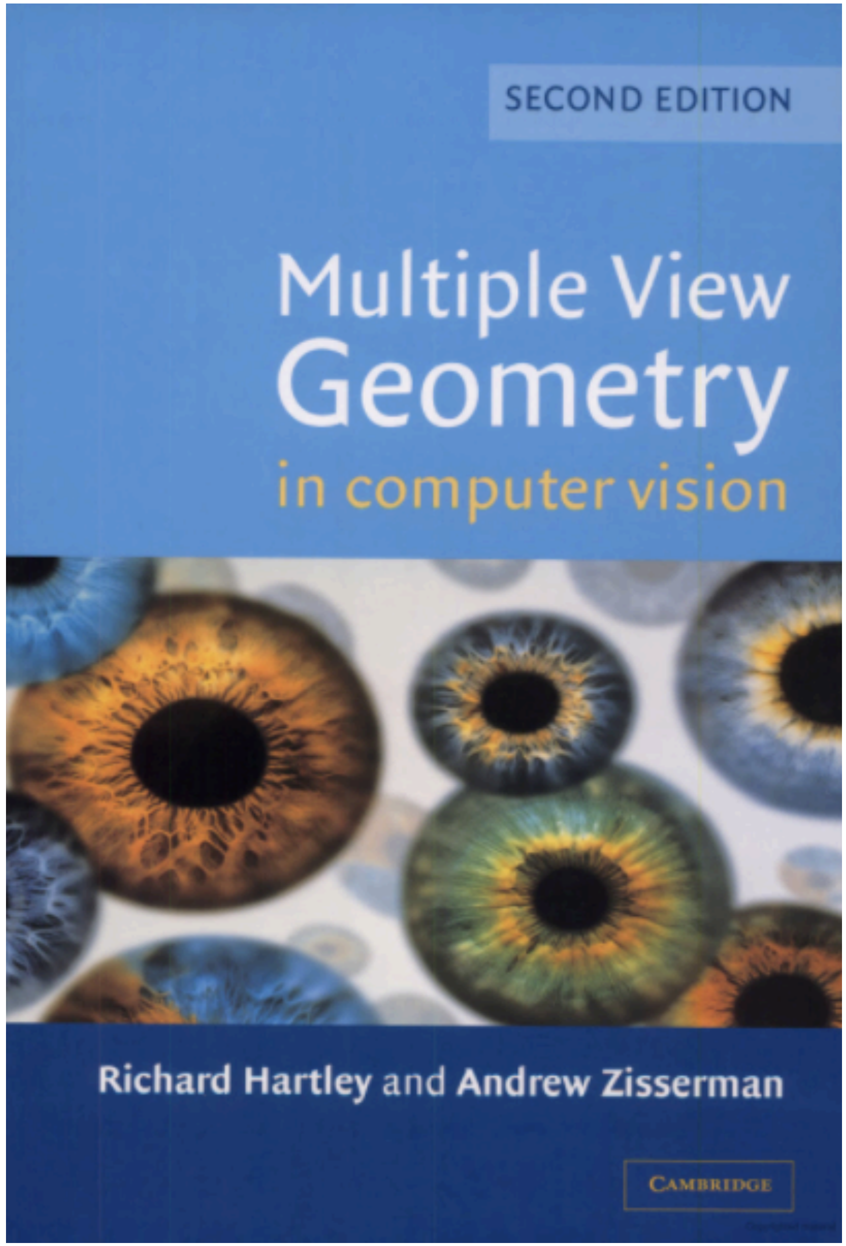
- Line given by cross product of two points

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

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

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

Useful reference

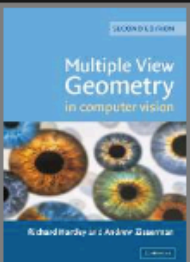


Useful reference

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

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

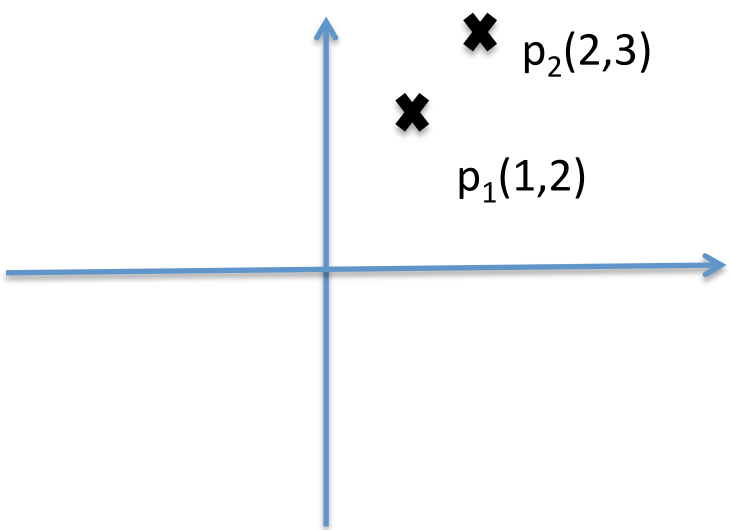
$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Basic geometry in homogeneous coordinates

- Line given by cross product of two points

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

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

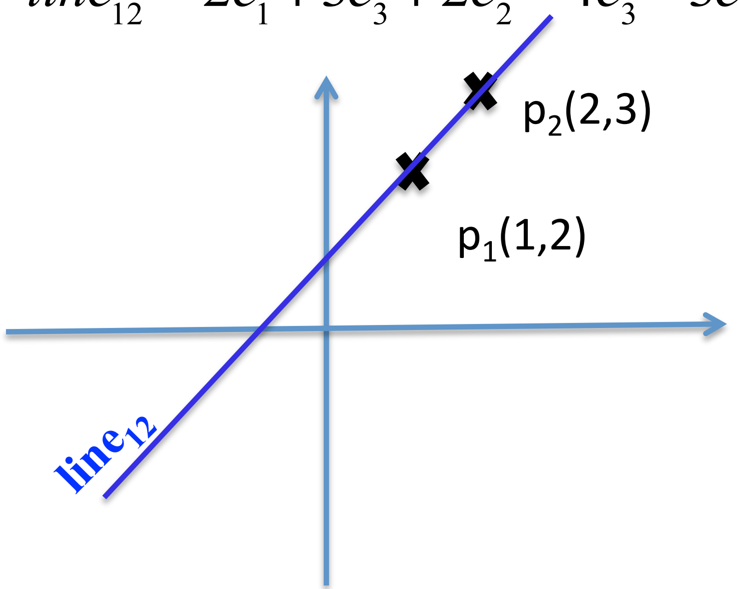


Basic geometry in homogeneous coordinates

- Line given by cross product of two points

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

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



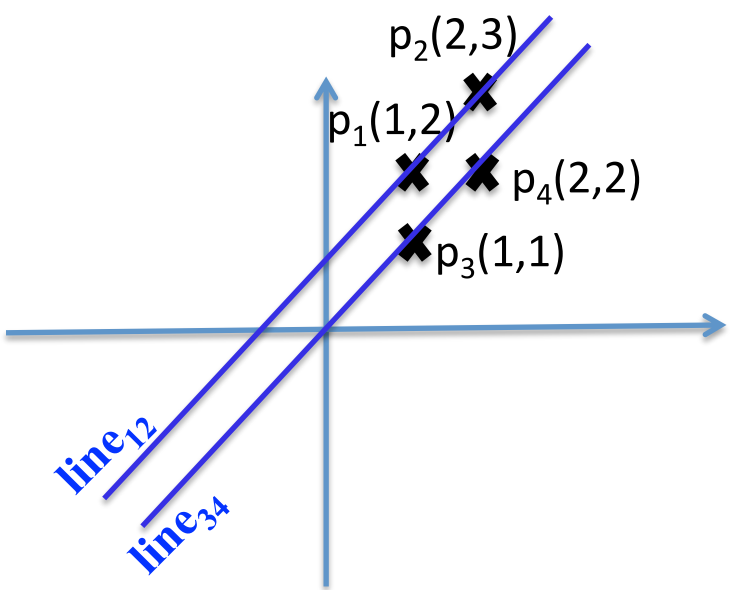
$$line_{12} : ax + by + c = 0, a = -1, b = 1, c = -1$$

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

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

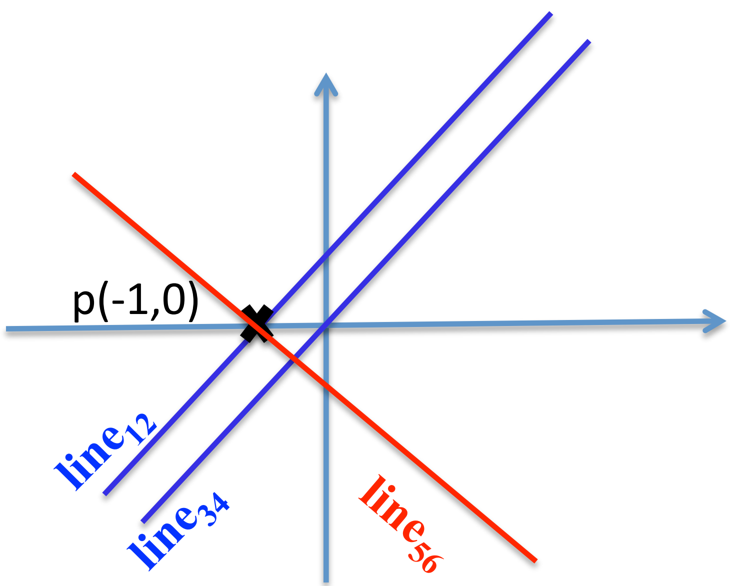
- Intersection of two lines given by cross product of the lines

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

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

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

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



Another problem solved by homogeneous coordinates

Intersection of parallel lines

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

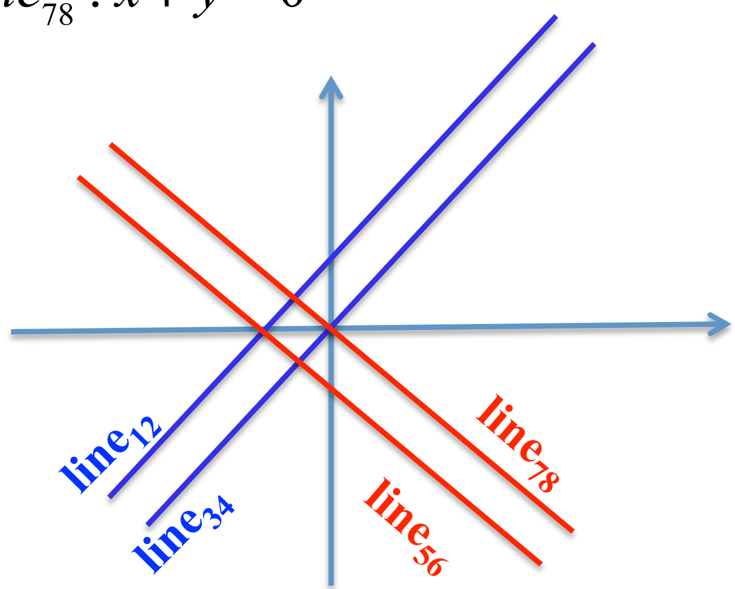
$line_{34} : x - y = 0$

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

$line_{78} : x + y = 0$

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

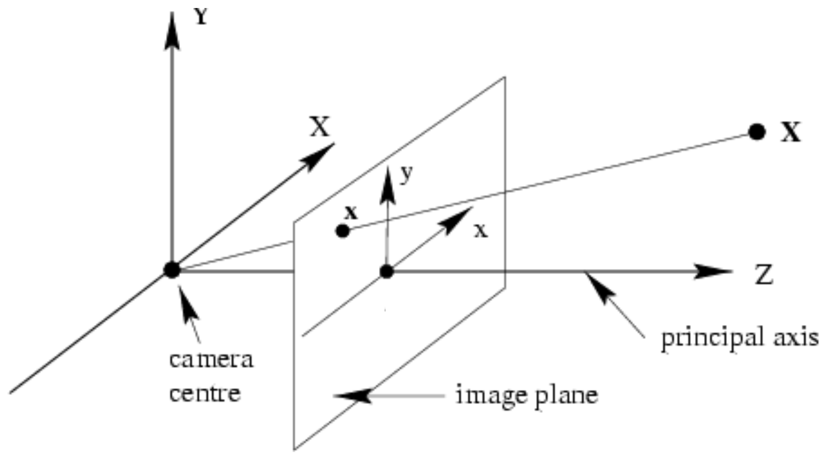
Cartesian: (Inf, Inf)
Homogeneous: (-1, 1, 0)

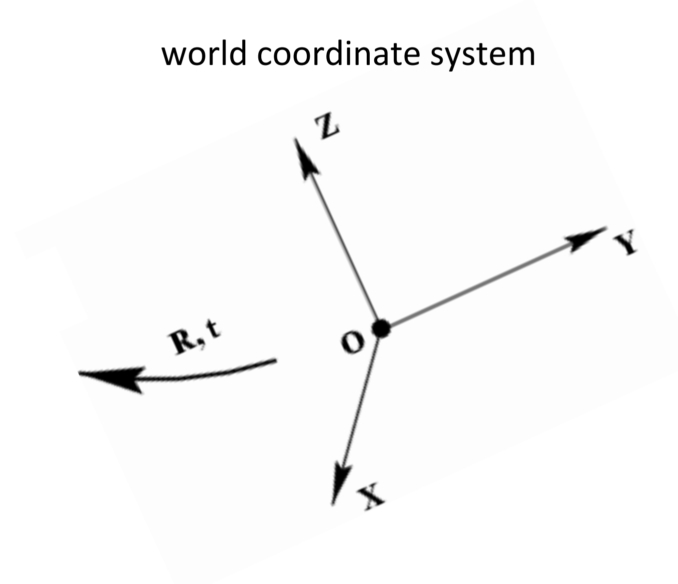


$$p = line_{12} \times line_{34} = \begin{vmatrix} e_1 & e_2 & e_3 \\ -1 & 1 & -1 \\ 1 & -1 & 0 \end{vmatrix} = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$$

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

Camera calibration

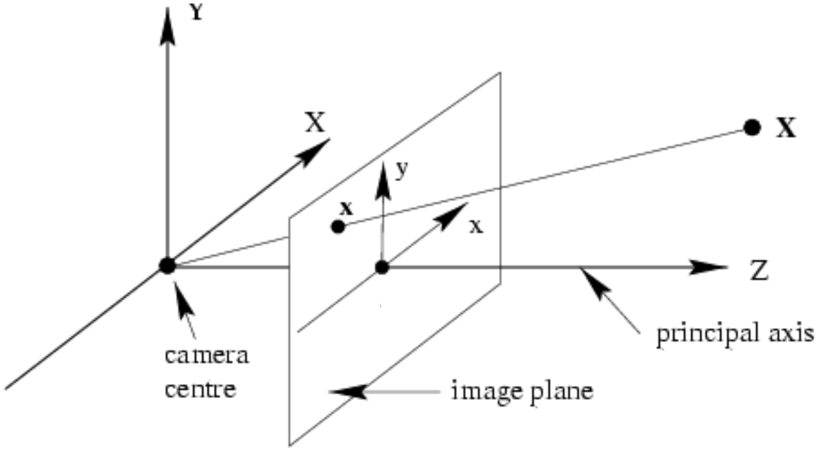


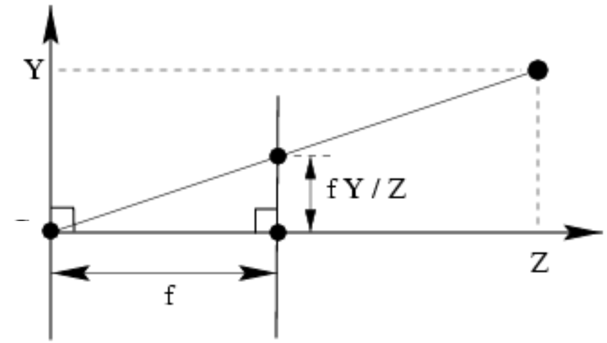


- **Normalized (camera) coordinate system:** camera center is at the origin, the *principal axis* is the z-axis, x and y axes of the image plane are parallel to x and y axes of the world

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

Pinhole camera model





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

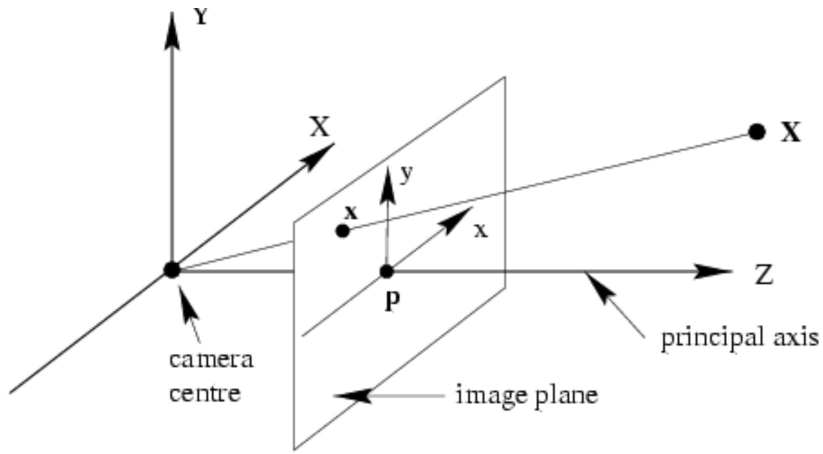
$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

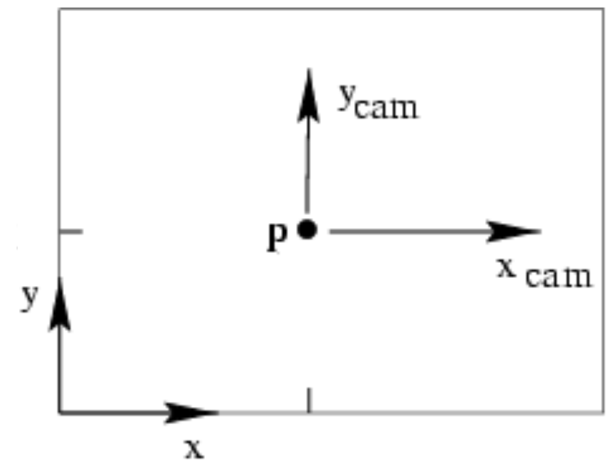
$$\mathbf{x} = \mathrm{P}\mathbf{X}$$

x = homogenous coordinates in the image plane

X = homogenous coordinates in the 3D scene

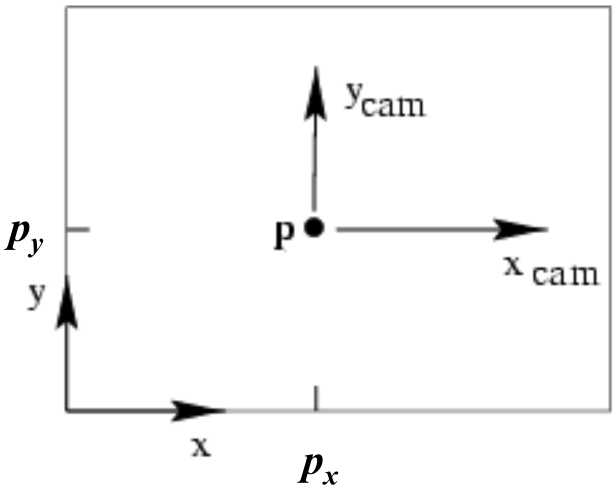
Principal point





- **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

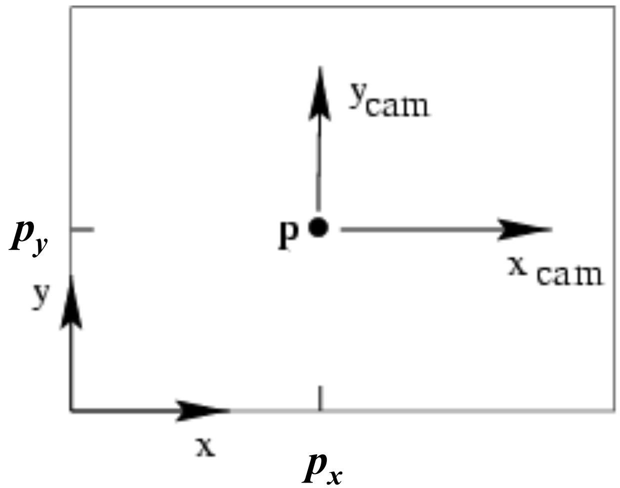


We want the principal point to map to (p_x, p_y) instead of (0,0)

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

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

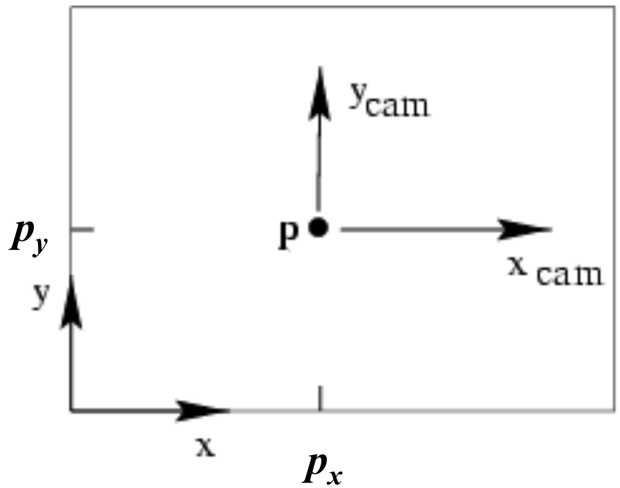
Principal point offset



principal point: (p_x, p_y)

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

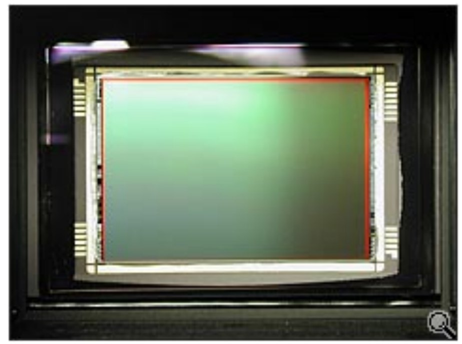
Principal point offset



principal point: (p_x, p_y)

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

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

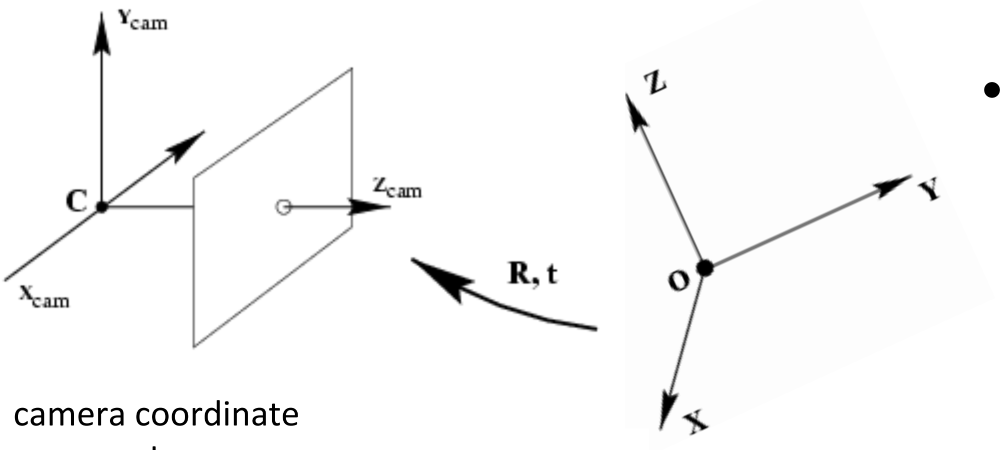
$$K = \underbrace{\begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}}_{\text{pixels/m}} \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\text{m}} = \underbrace{\begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}}_{\text{pixels}}$$

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)

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):

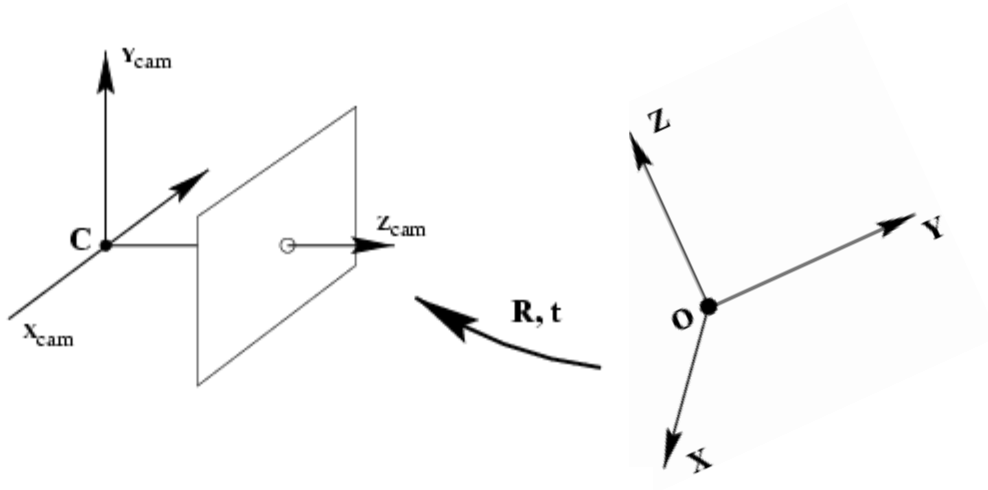
$$\widetilde{X}_{cam} = R\left(\widetilde{X} - \widetilde{C}\right)$$

coords. of point in camera frame

coords. of a point in world frame

coords. of camera center in world frame

Camera rotation and translation



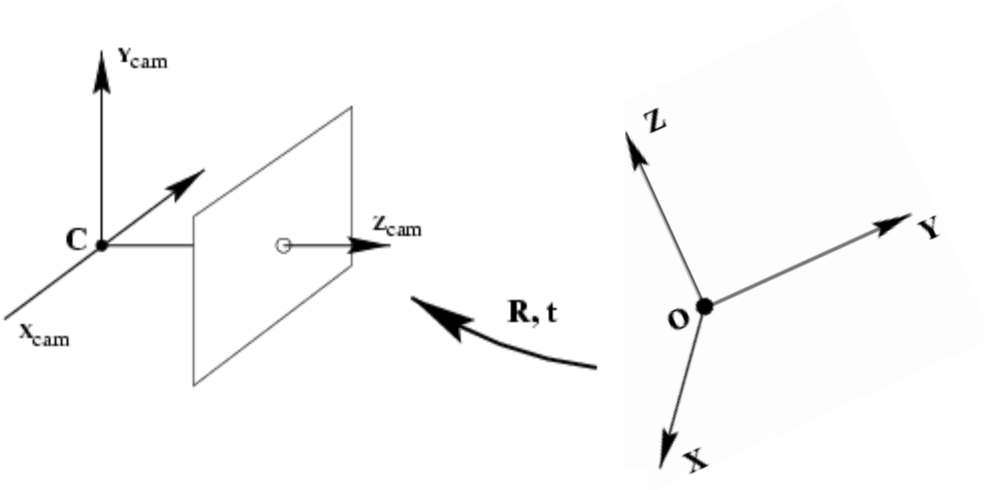
$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

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

3D transformation matrix (4 x 4)

Homogeneous coordinates

Camera rotation and translation



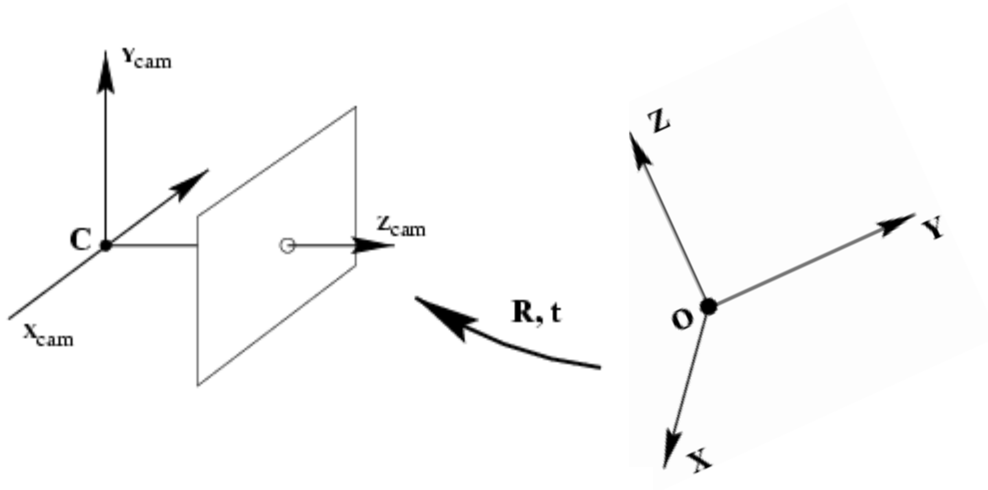
$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

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

3D transformation matrix (4 x 4)

Homogeneous coordinates

Camera rotation and translation



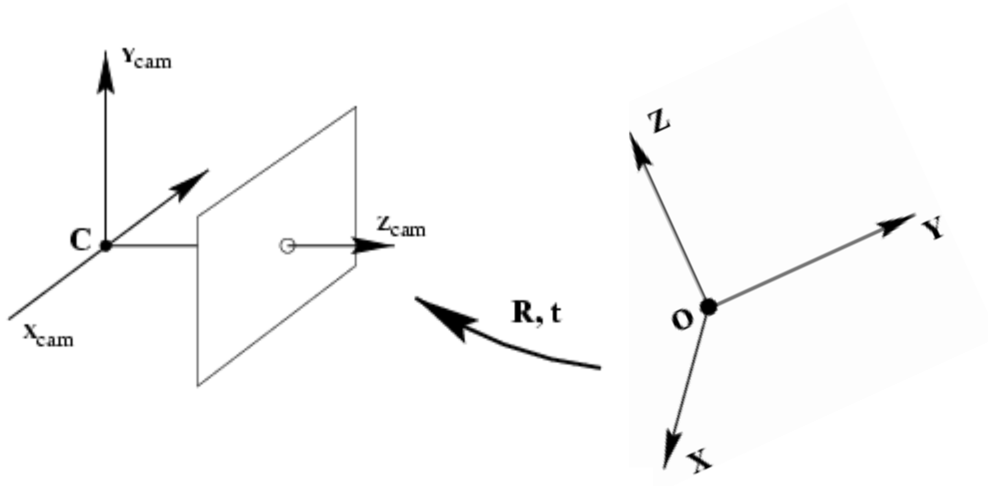
$$x = K[I \mid 0]\begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

2D transformation matrix (3 x 3)

perspective projection matrix (3 x 4)

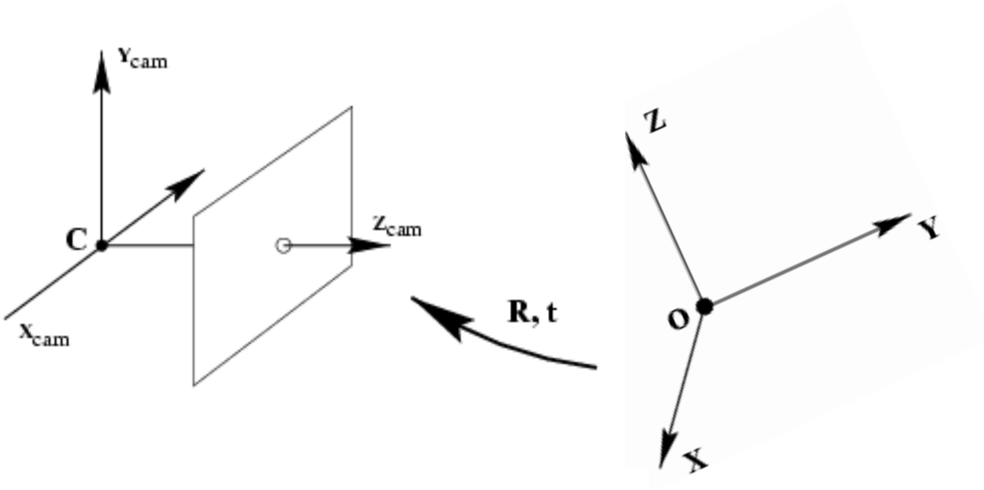
3D transformation matrix (4 x 4)

Camera rotation and translation



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

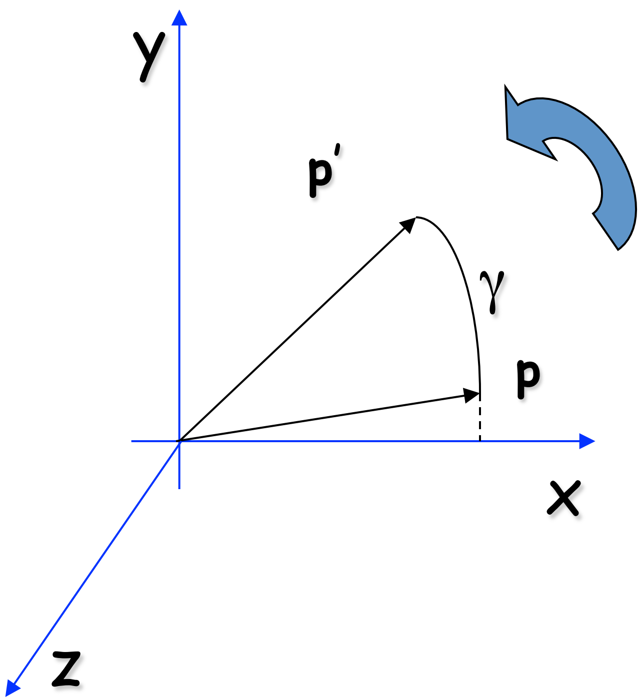
Camera rotation and translation



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

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



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

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

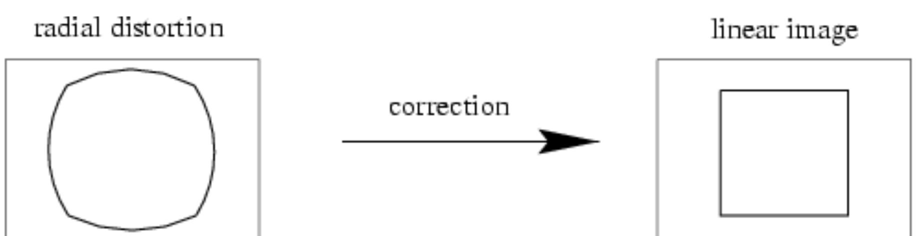
$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Slide Credit: Saverese

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 $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$

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

coords. of camera center in world frame

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

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

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

$$\overset{5}{\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}} \overset{6}{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}$$

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

How to calibrate the camera?

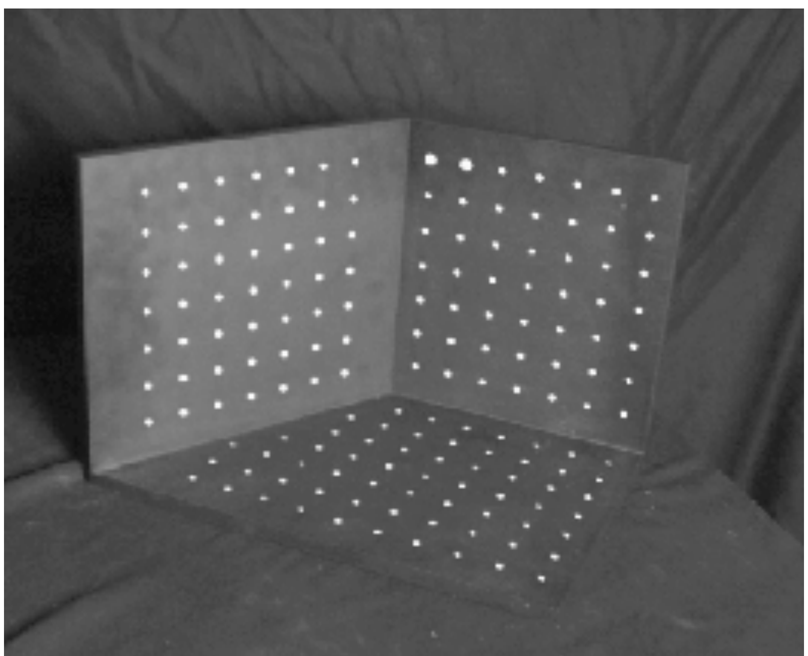
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

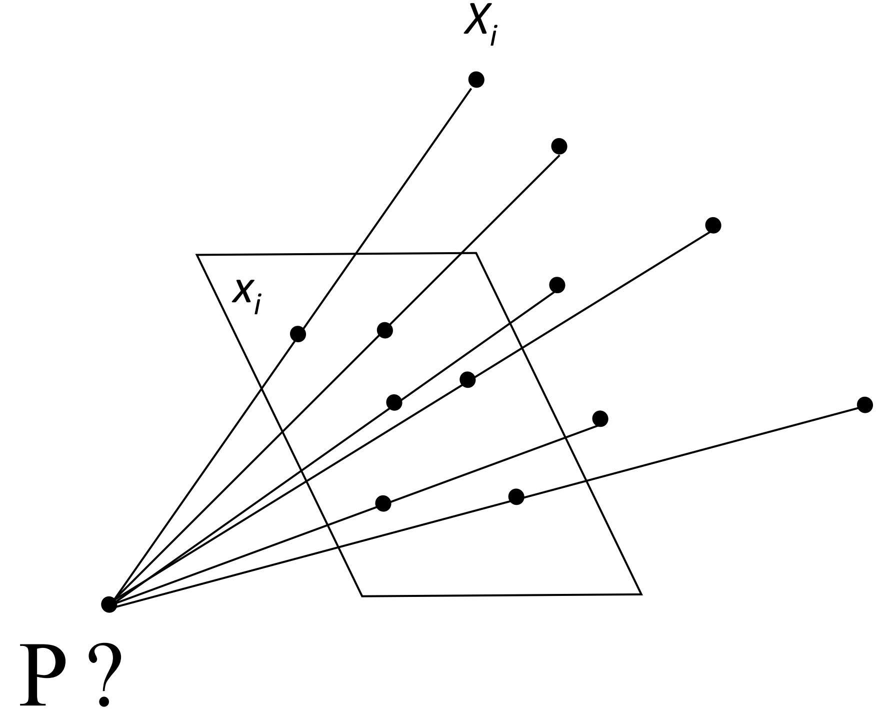
$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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

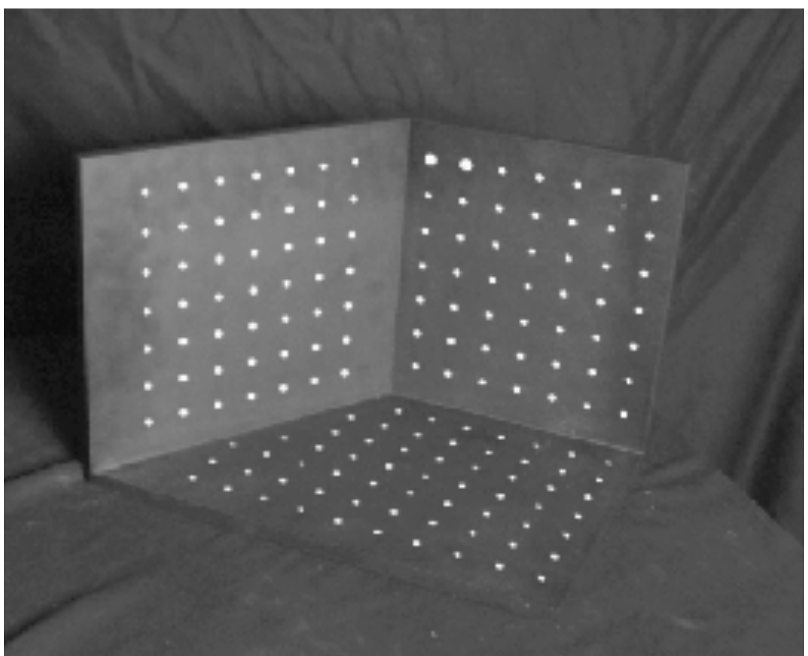




Calibrating the Camera

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

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

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

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

Calibration with linear method

- Advantages
  - Easy to formulate and solve
  - Provides initialization for non-linear methods

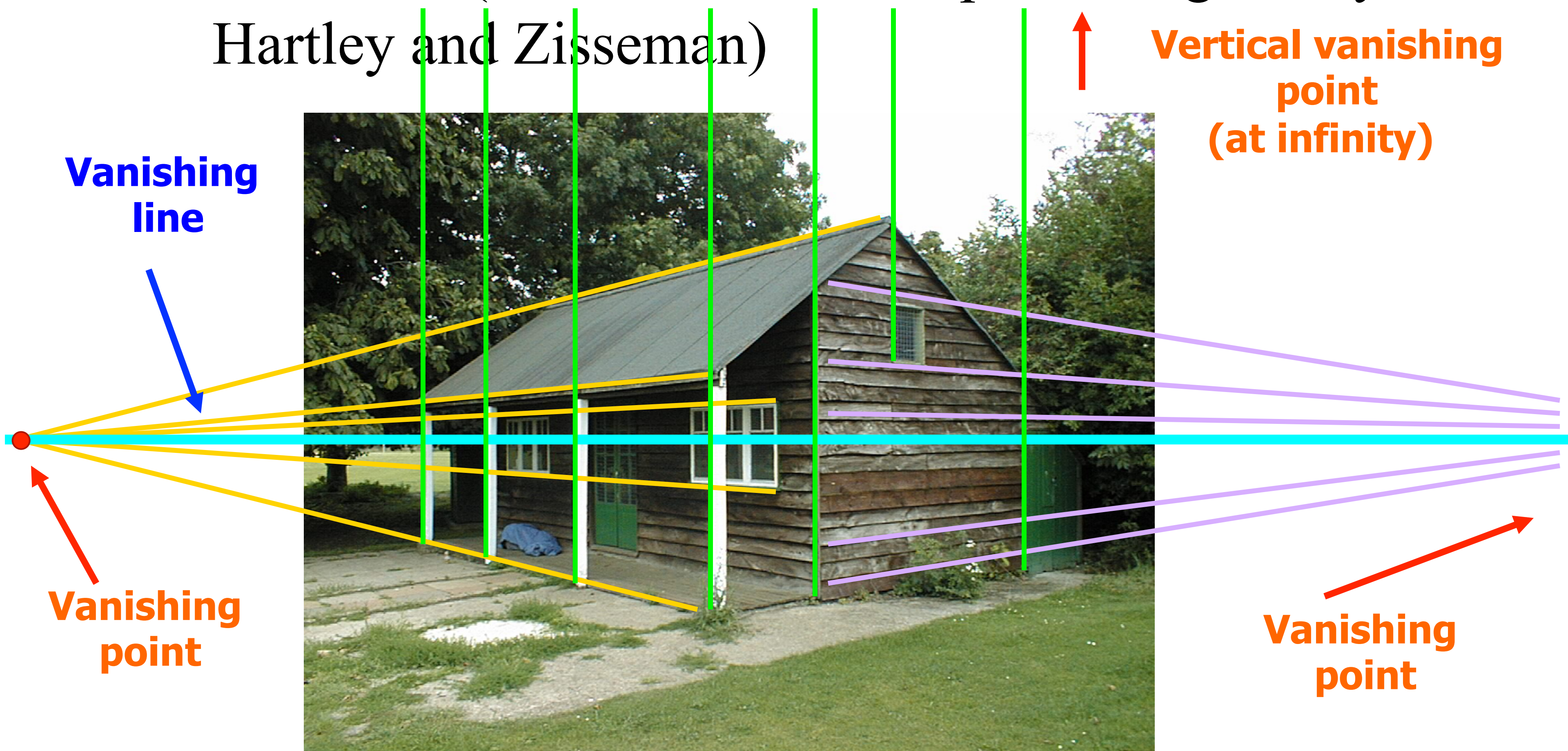
- Disadvantages
  - Doesn’t directly give you camera parameters
  - Doesn’t model radial distortion
  - Can’t impose constraints, such as known focal length
  - Doesn’t minimize projection error

- Non-linear methods are preferred
  - Define error as difference between projected points and measured points
  - Minimize error using Newton’s method or other non-linear optimization

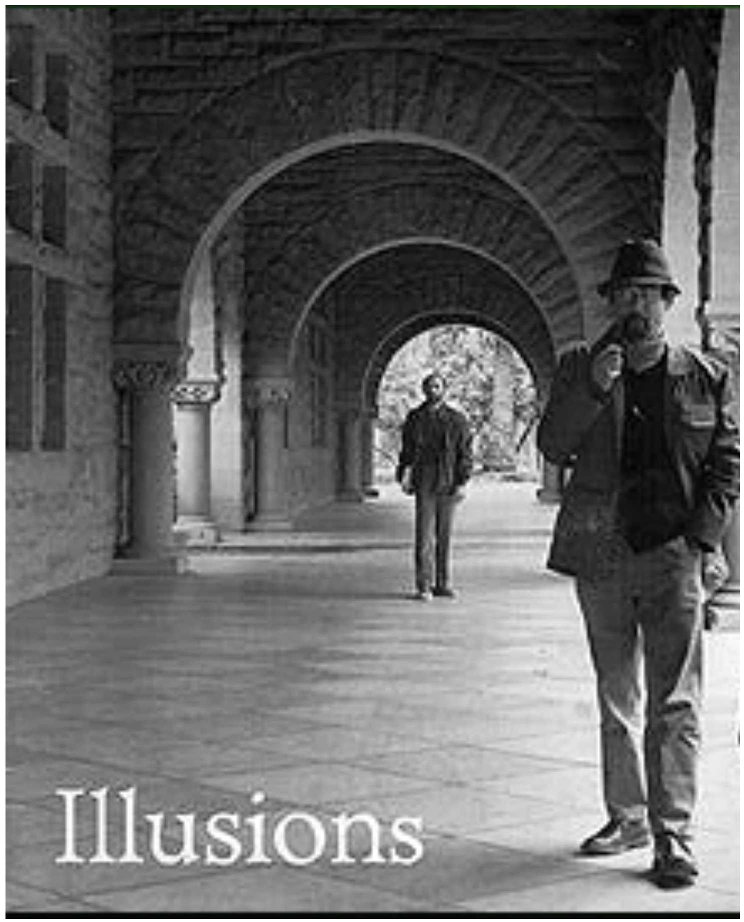
Calibrating the Camera

Method 2: Use vanishing points

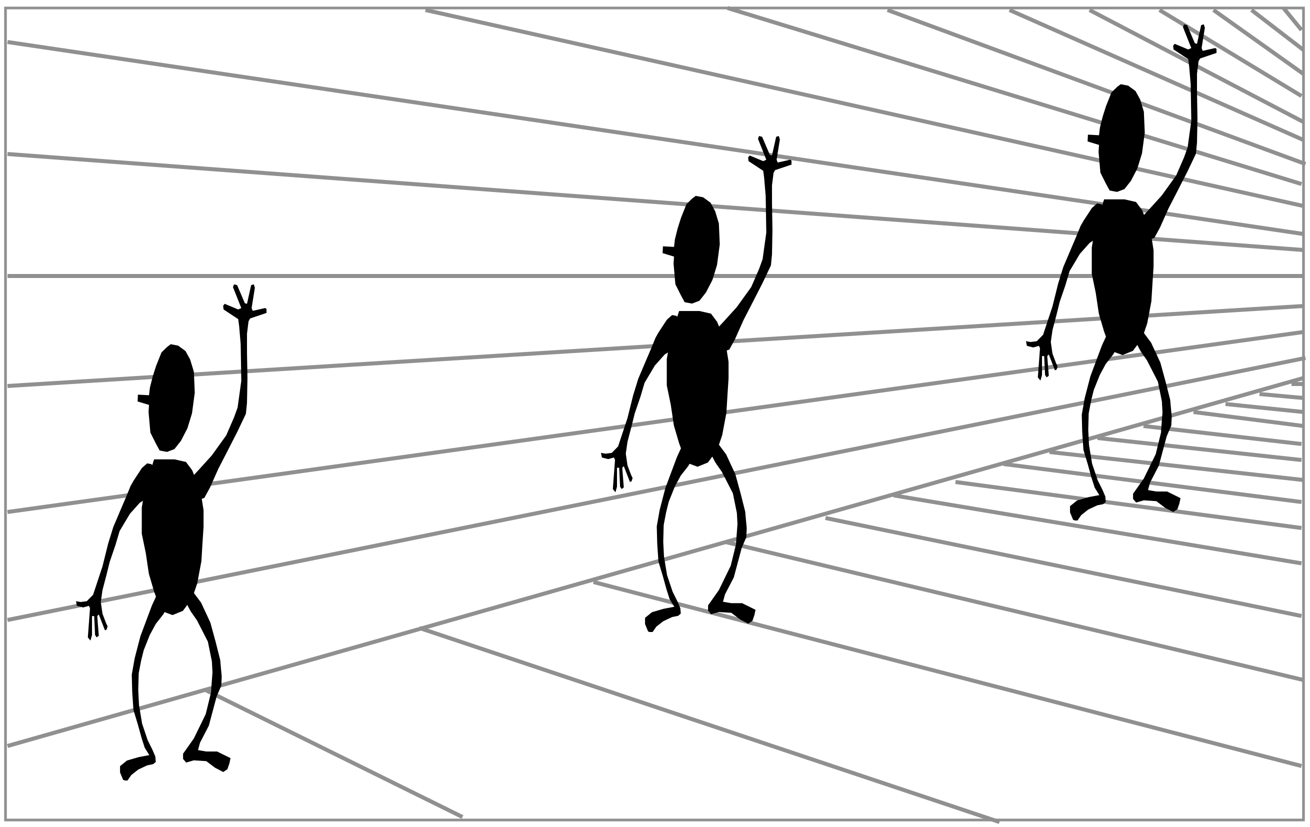
– Find vanishing points corresponding to orthogonal directions (see the book: Multiple view gemetry, Hartley and Zisseman)



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

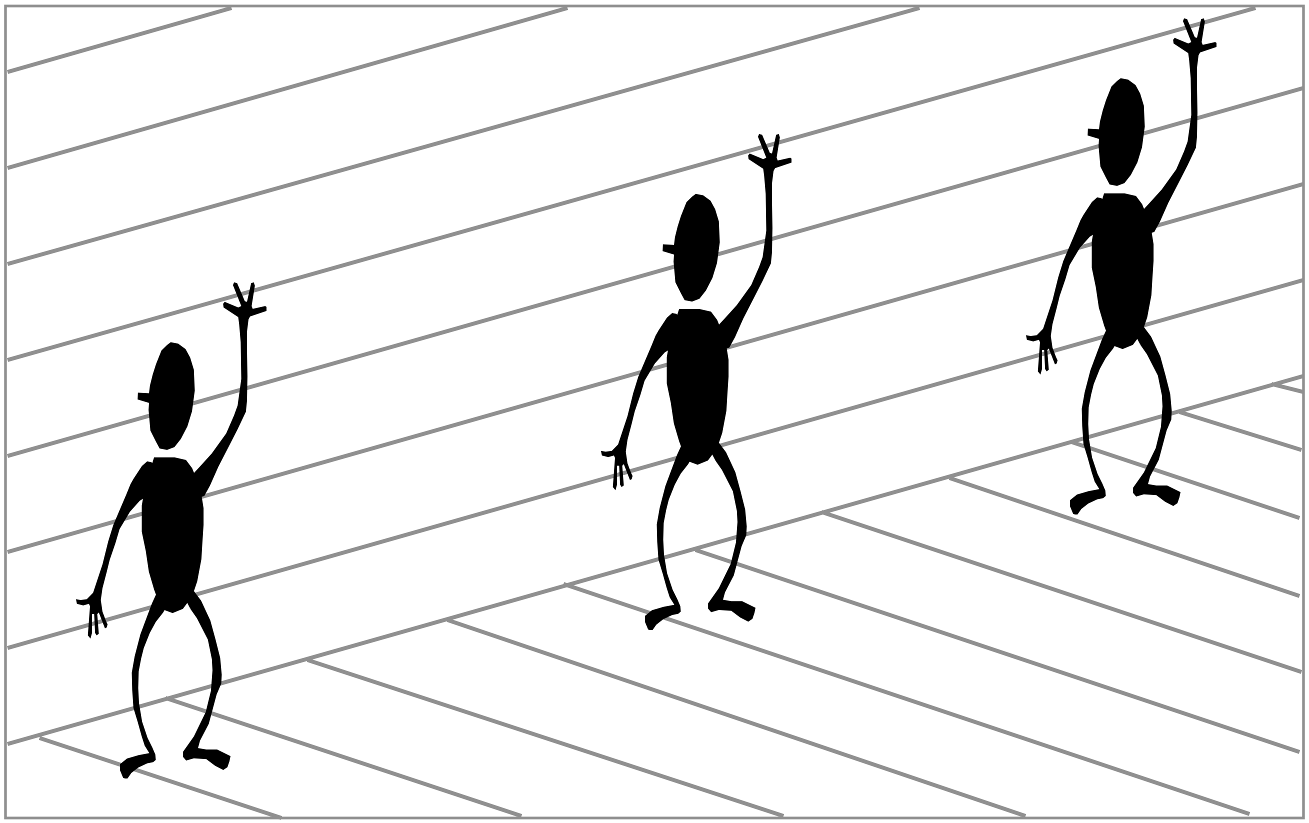


Perspective cues



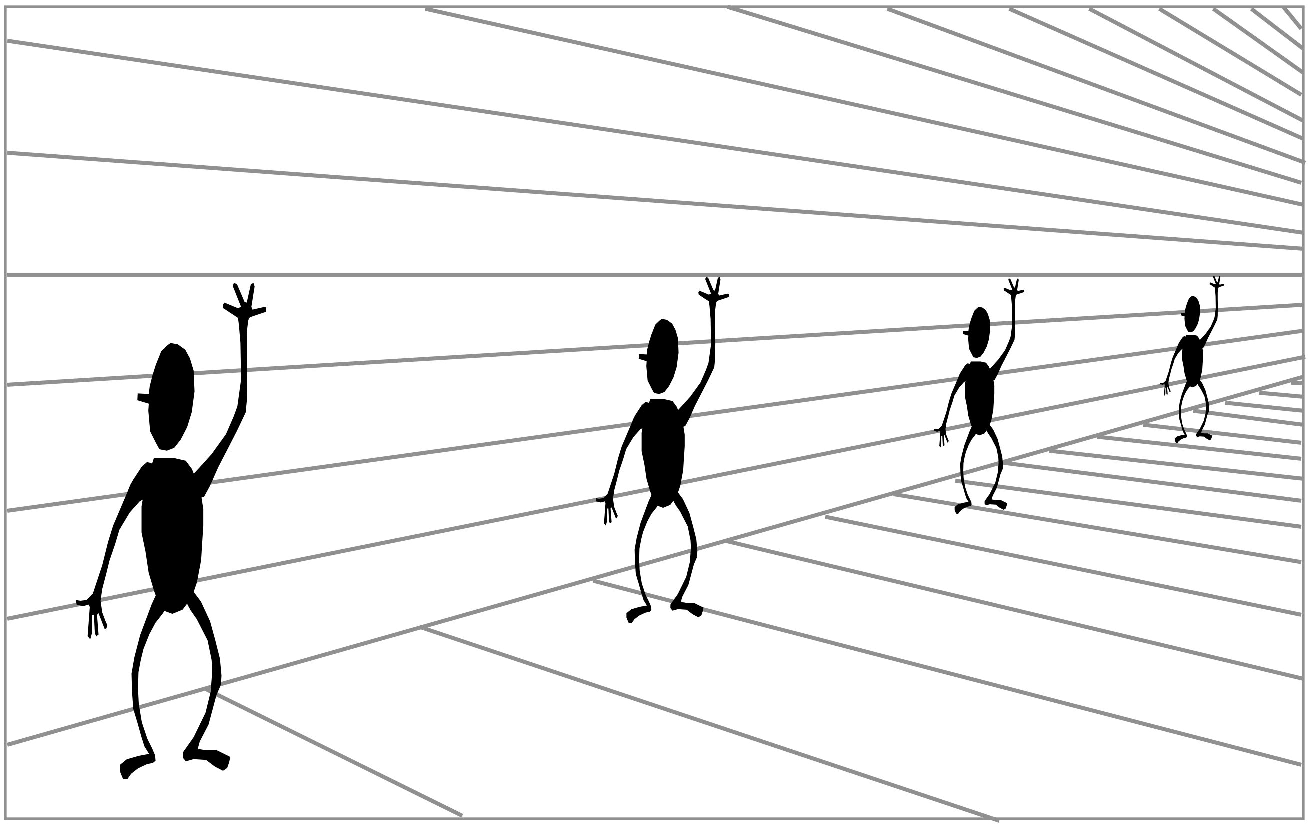
Slide credit: Steve Seitz

Perspective cues



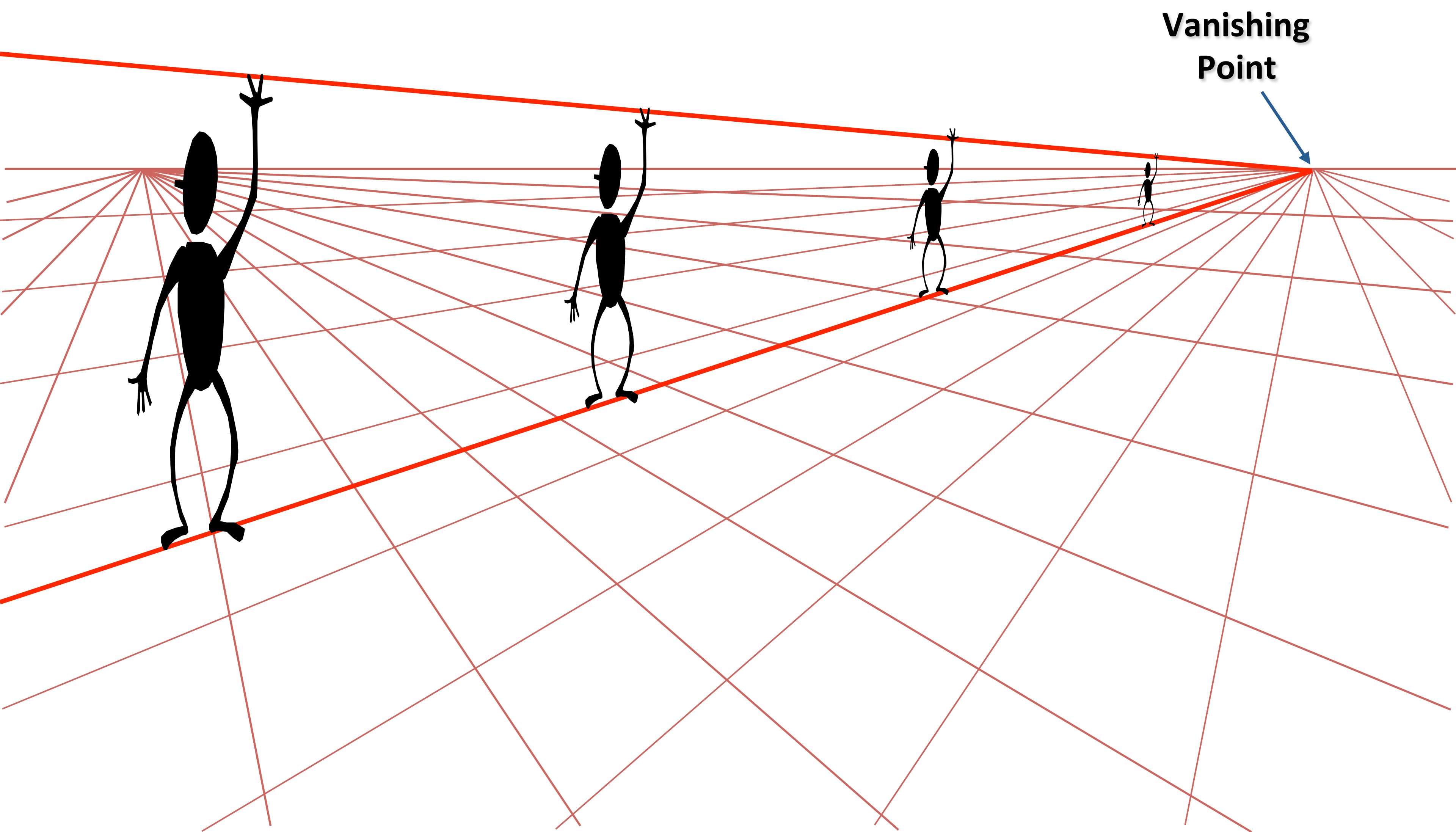
Slide credit: Steve Seitz

Perspective cues

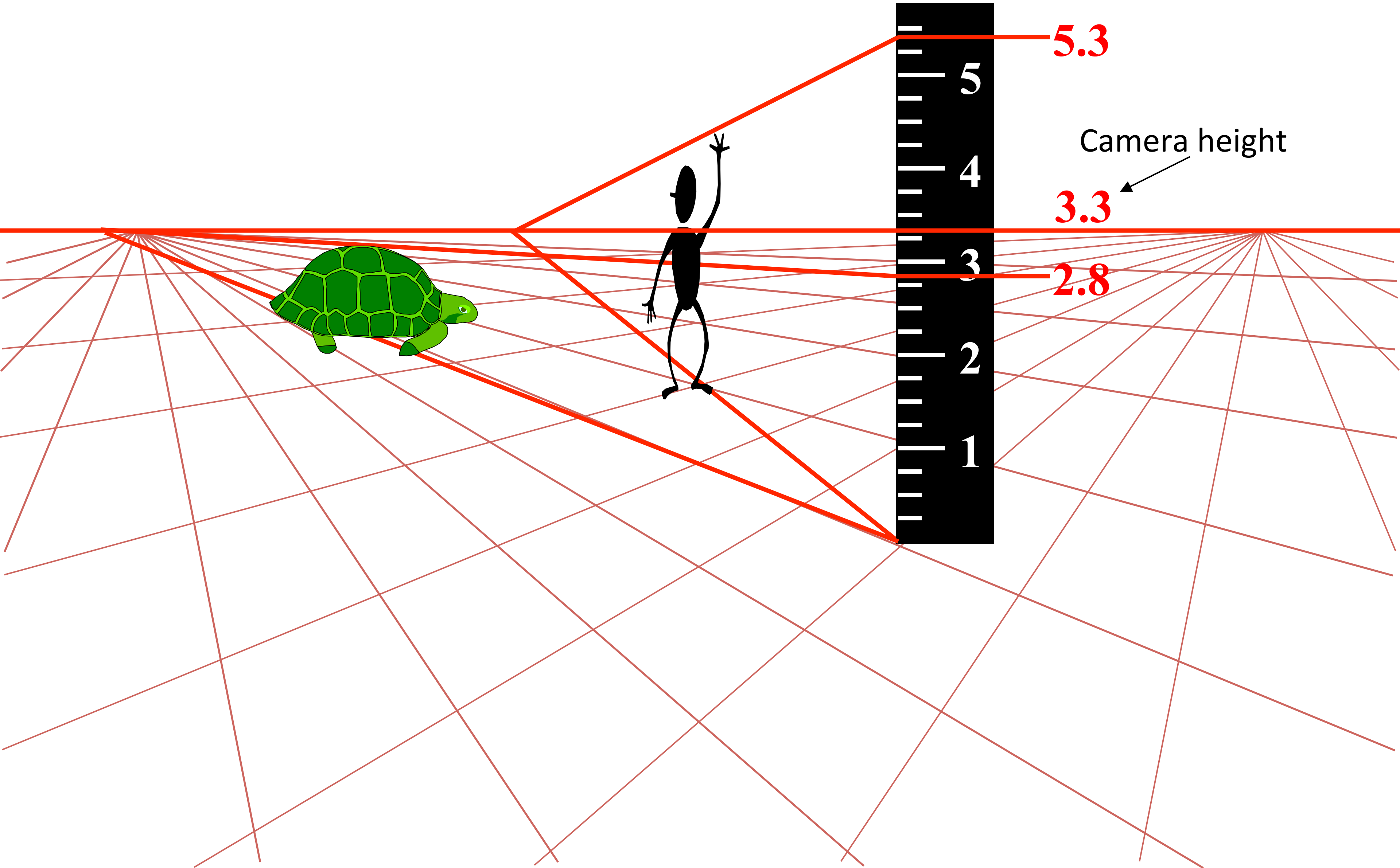


Slide credit: Steve Seitz

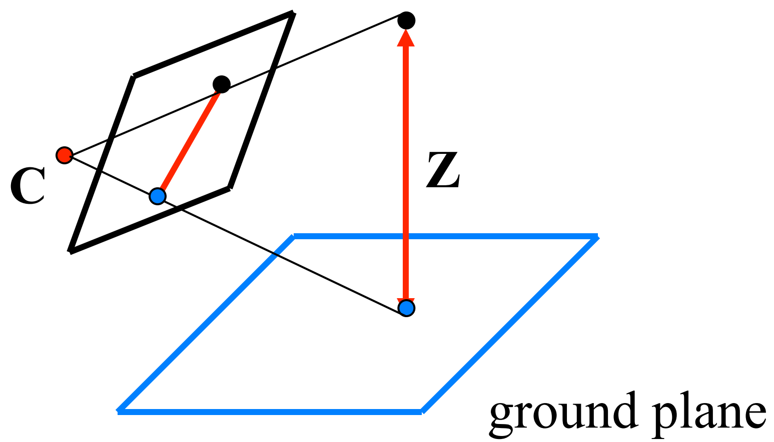
Comparing heights



Measuring height



Measuring height without a ruler

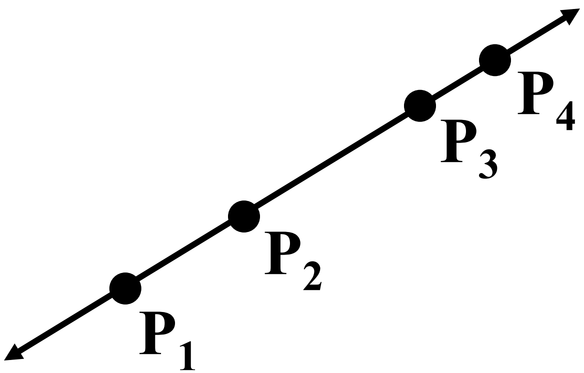


Compute Z from image measurements

The cross ratio

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

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|} \qquad \mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

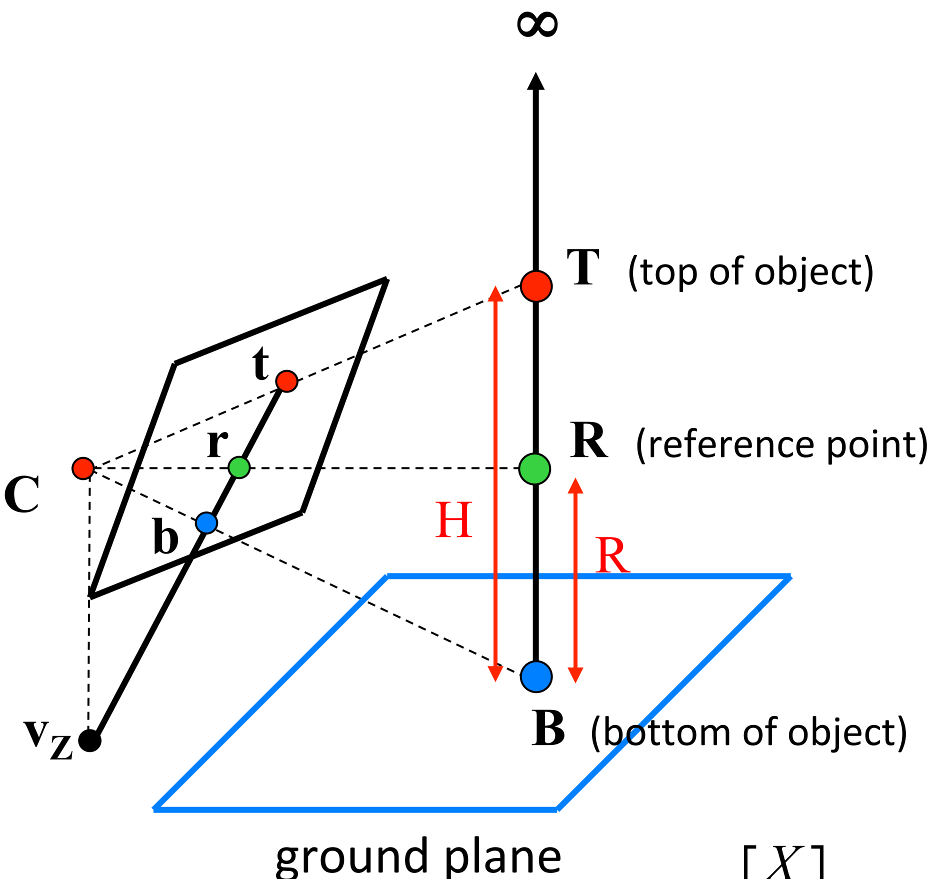
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

Measuring height



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

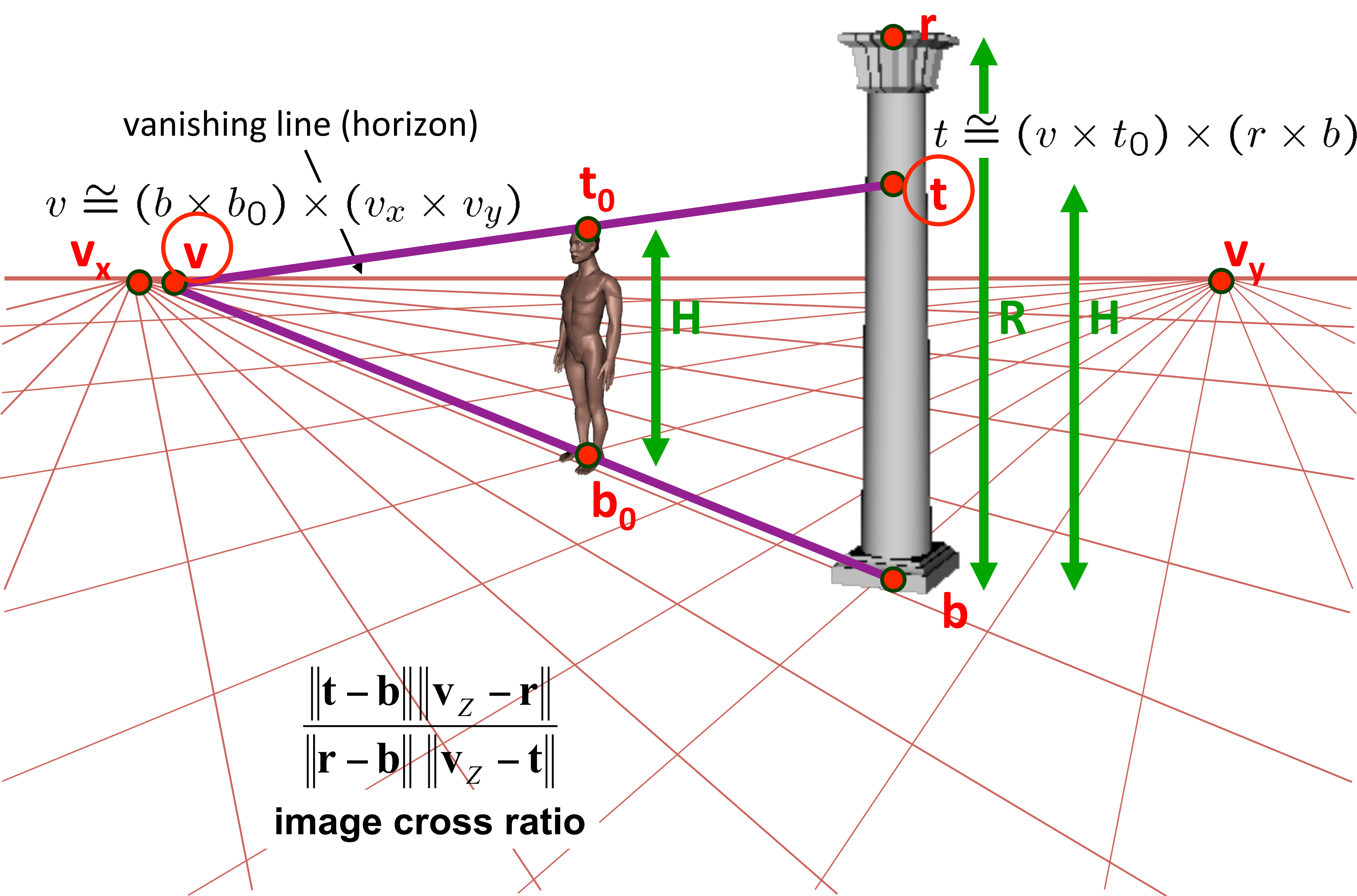
scene cross ratio

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

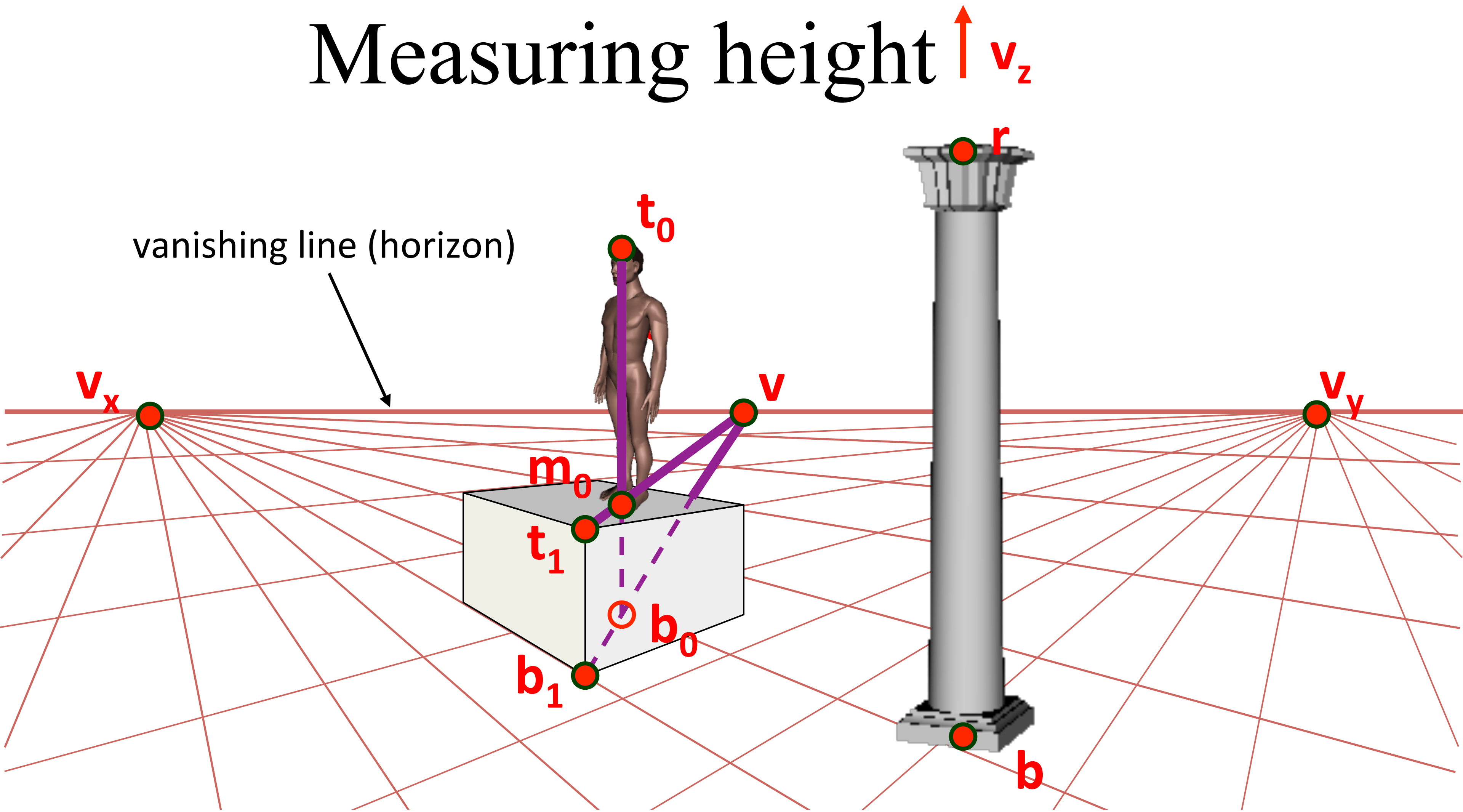
image cross ratio

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



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