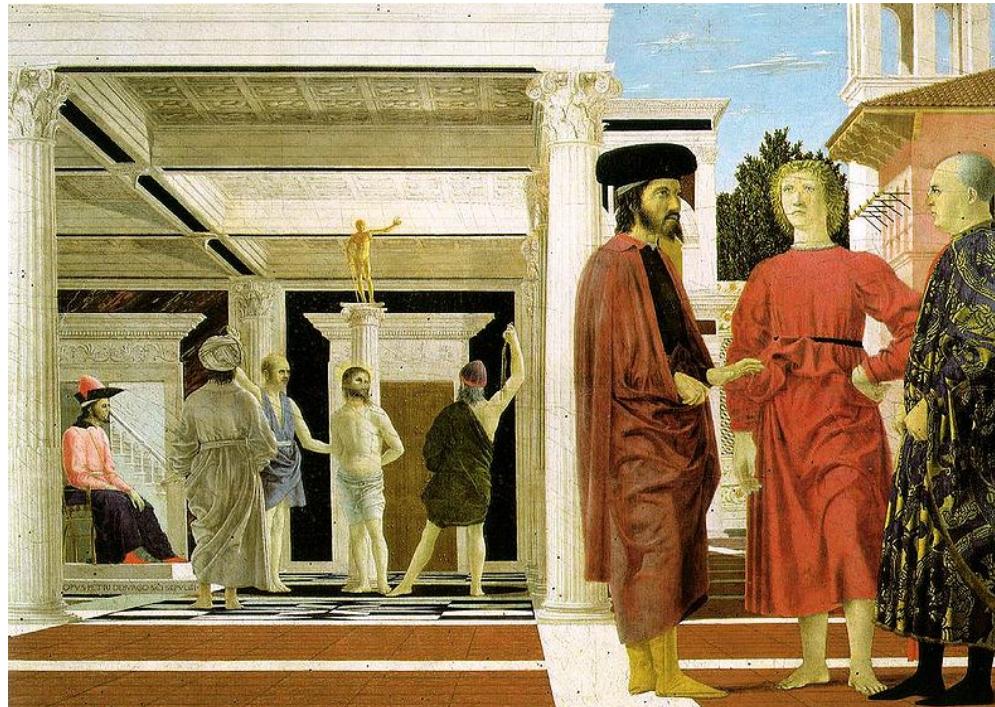


CS4670/5670: Computer Vision

Kavita Bala

Lec 19: Single-view modeling 2



NEW YORK TORONTO TELLURIDE
2012



"WE INSPIRED PORN AND TOLLEY'S STERLING DOCUMENTARY."

DISNEY ENTERTAINMENT AND MEDIA INC. HAS BEEN APPROVED BY THE SECURITIES AND EXCHANGE COMMISSION TO FILE

“预防以人为主”的新理念

Tim's Vermeer

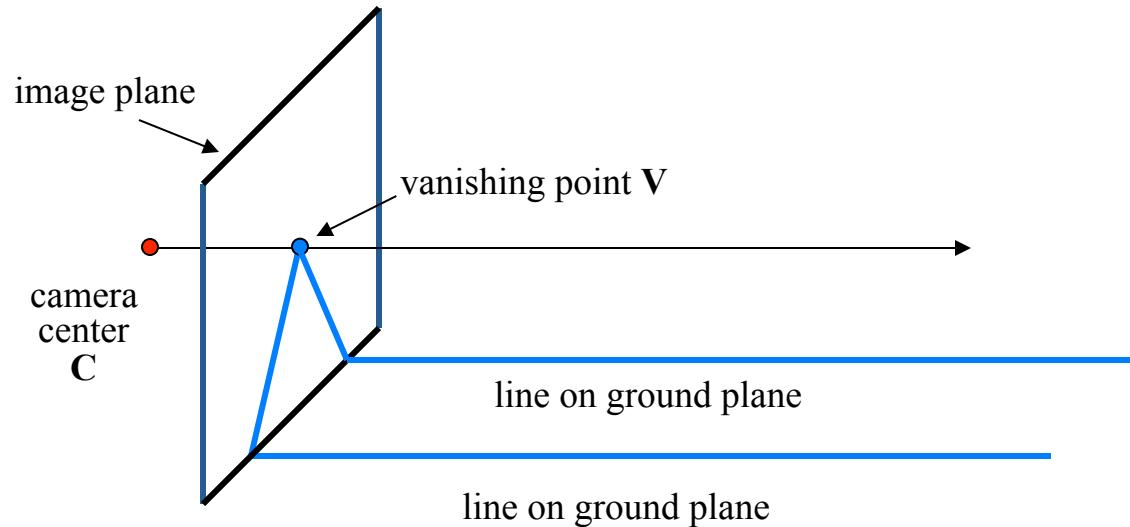
A Penn & Teller Film



Today

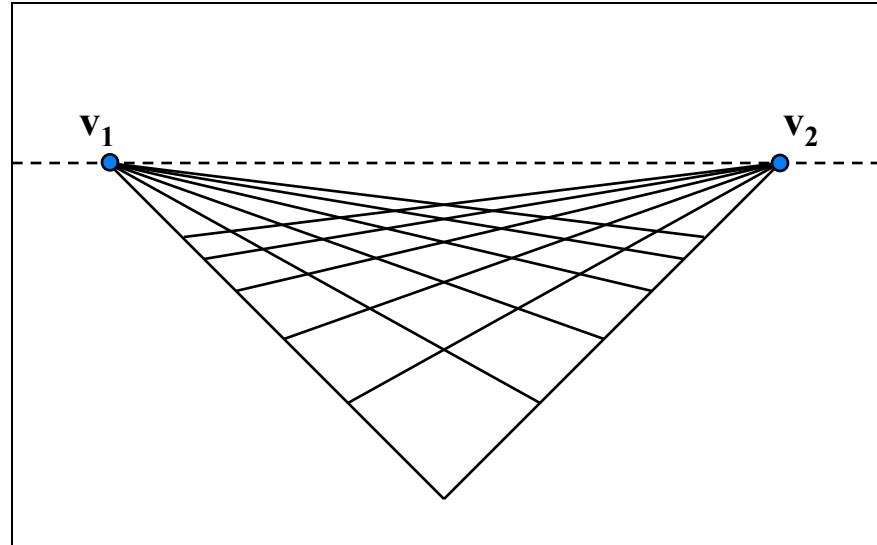
- Vanishing points in images are useful
 - Recover size
 - Camera calibration

Vanishing points



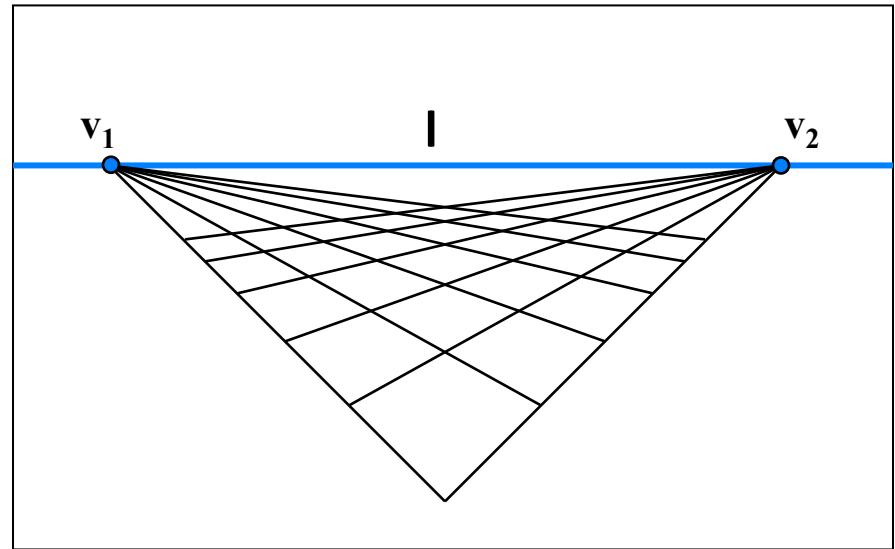
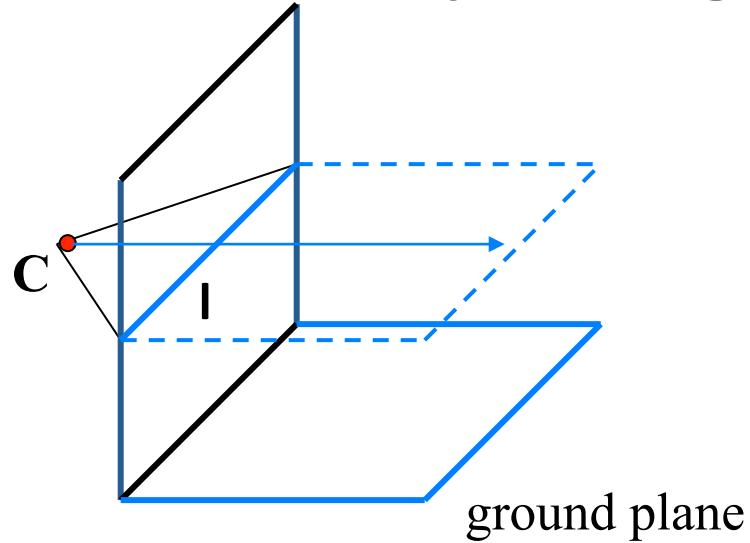
- Properties
 - Any two parallel lines (in 3D) have the same vanishing point v
 - The ray from C through v is parallel to the lines
 - An image may have more than one vanishing point
 - in fact, every image point is a potential vanishing point

Vanishing lines



- Multiple Vanishing Points
 - Any set of parallel lines on the plane define a vanishing point
 - The union of all of these vanishing points is the *horizon line*
 - also called *vanishing line*
 - Note that different planes (can) define different vanishing lines

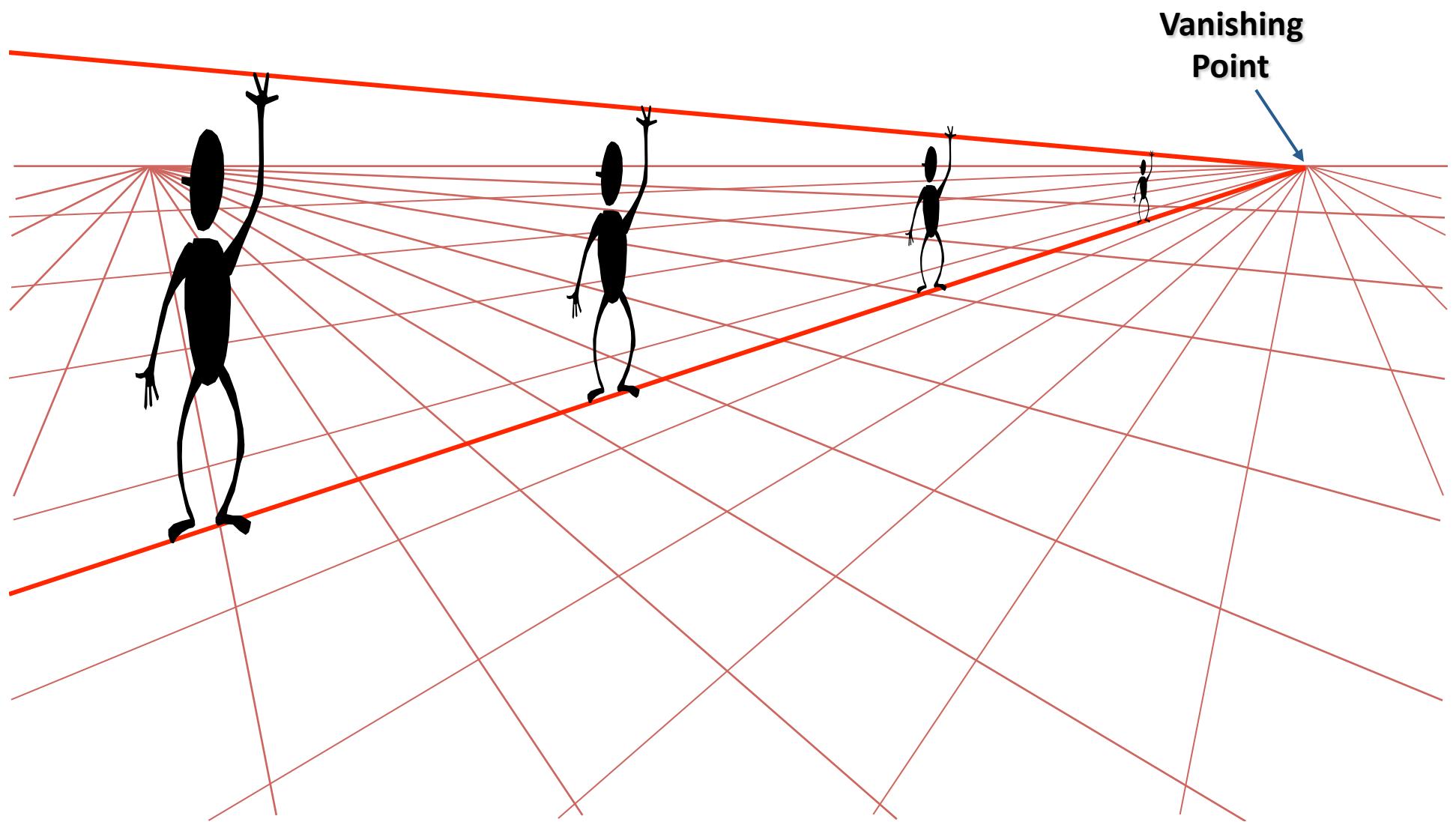
Computing vanishing lines



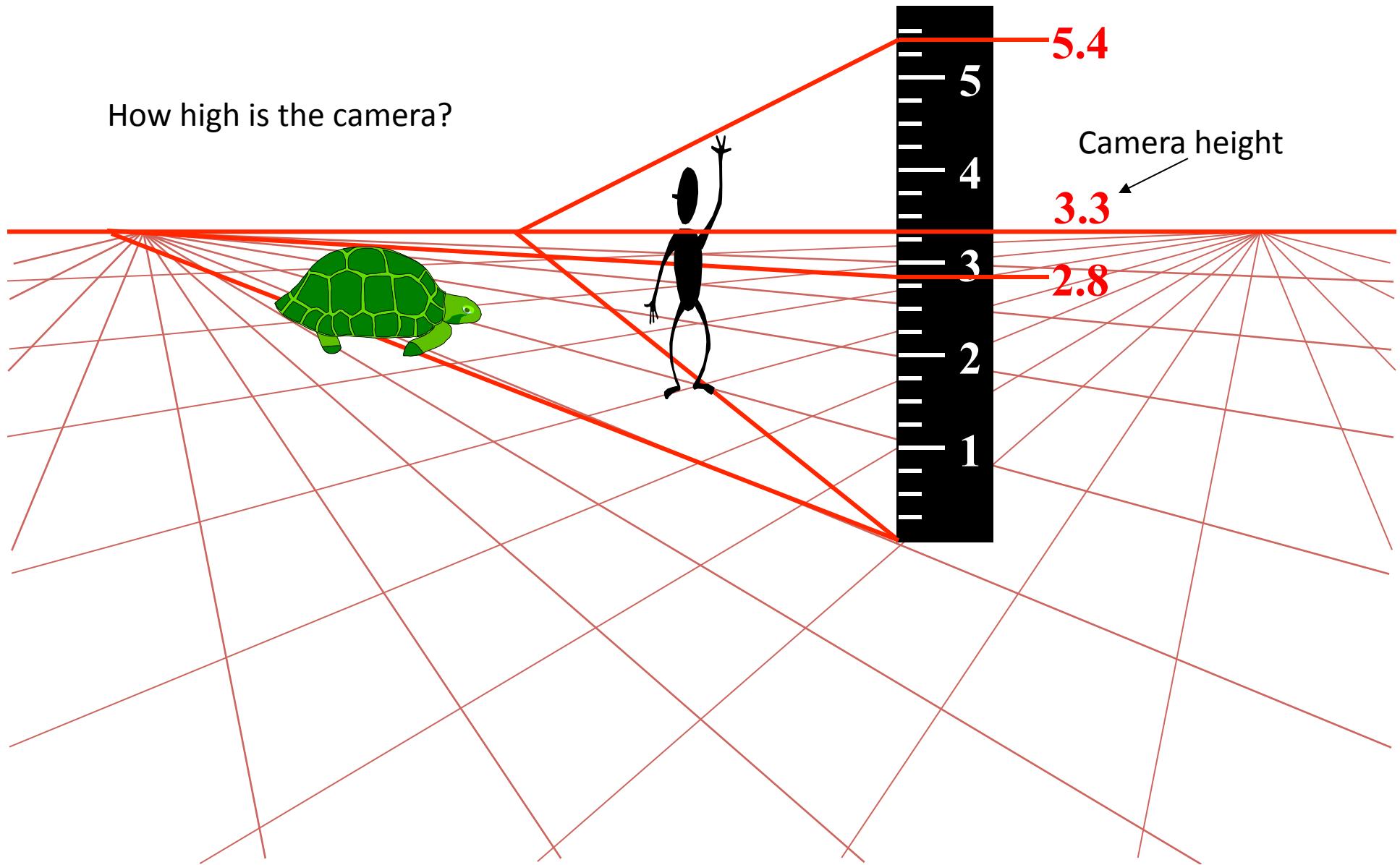
- ## Properties

- I is intersection of horizontal plane through C with image plane
- Compute I from two sets of parallel lines on ground plane
- All points at same height as C project to I
 - points higher than C project above I
- Provides way of comparing height of objects in the scene

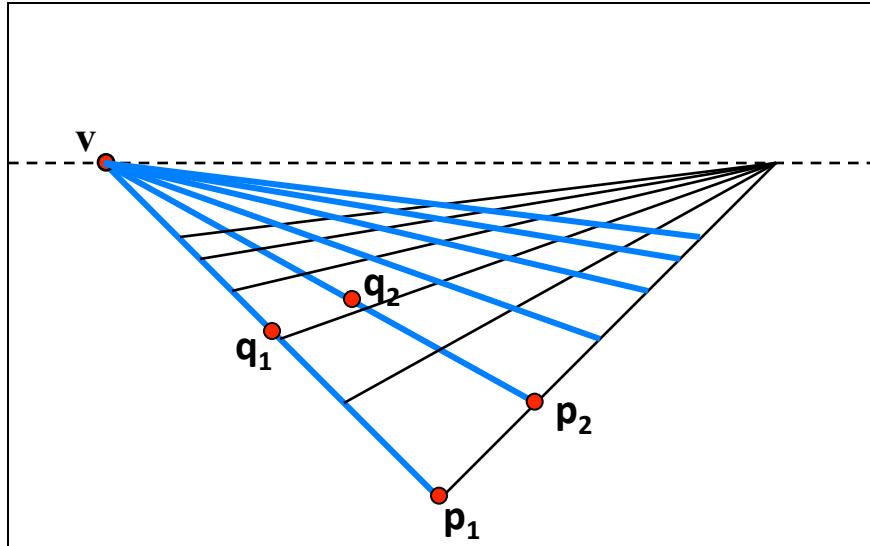
Comparing heights



Measuring height



Computing vanishing points (from lines)

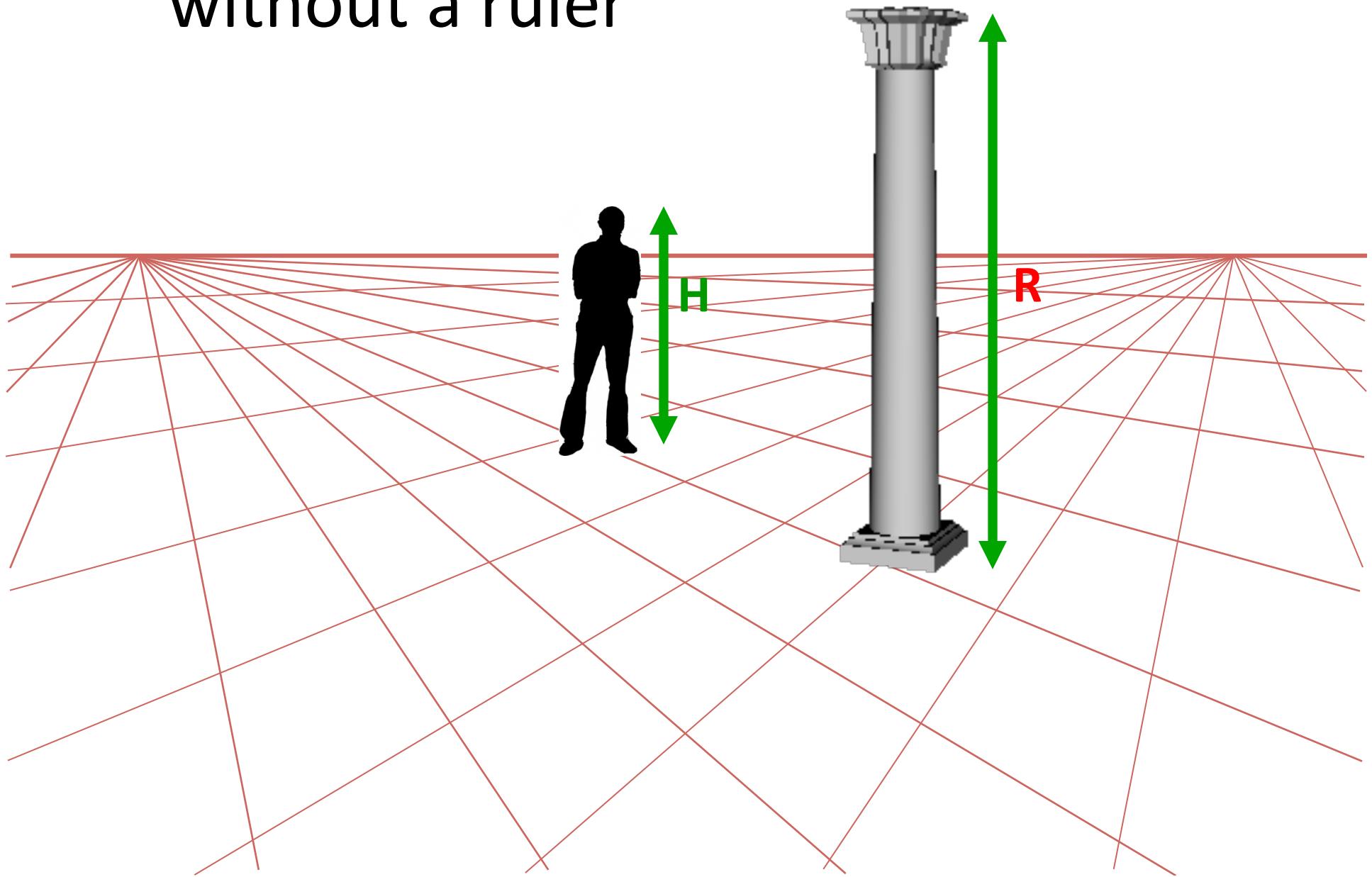


- Intersect p_1q_1 with p_2q_2
 $v = (p_1 \times q_1) \times (p_2 \times q_2)$

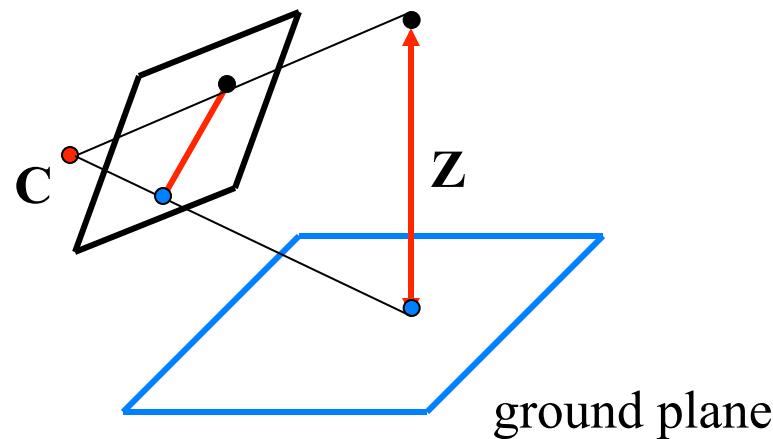
Least squares version

- Better to use more than two lines and compute the “closest” point of intersection
- See notes by [Bob Collins](#) for one good way of doing this:
 - <http://www-2.cs.cmu.edu/~ph/869/www/notes/vanishing.txt>

Measuring height without a ruler



Measuring height without a ruler



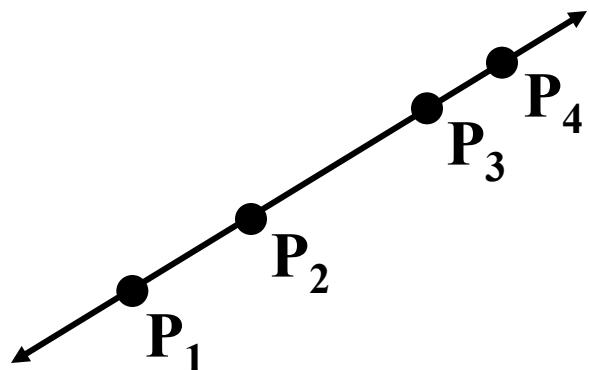
Compute Z from image measurements

Actually get a scaled version of z

The cross ratio

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

The *cross-ratio* of 4 collinear points



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

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

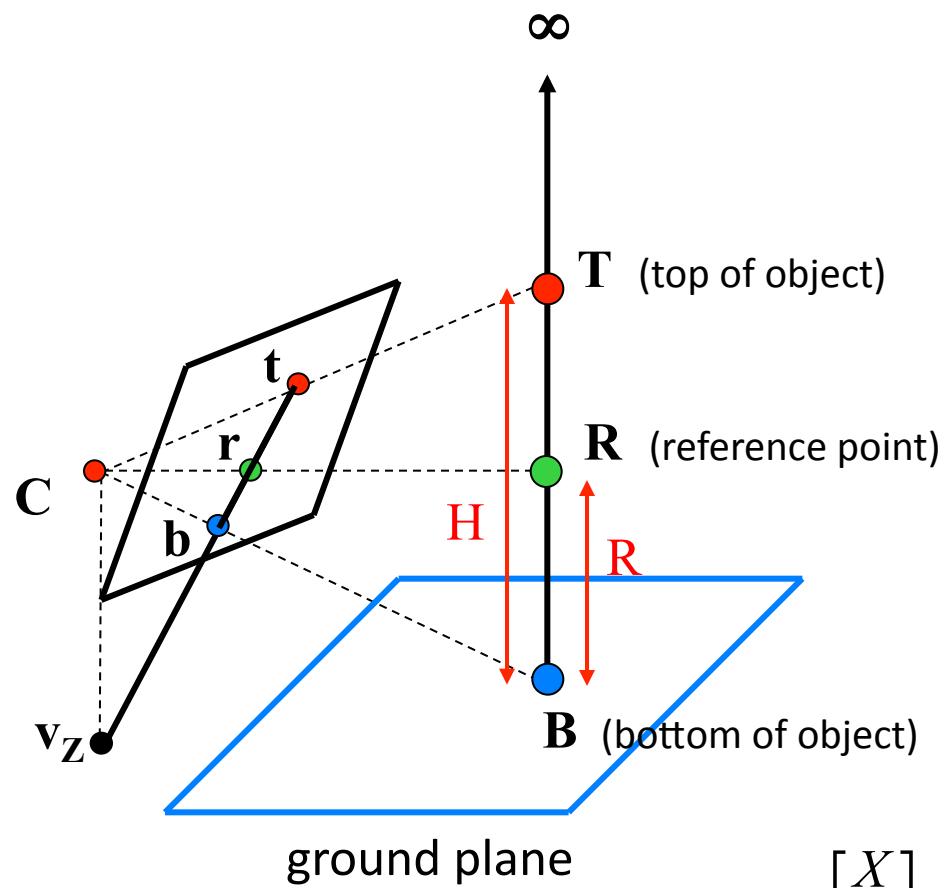
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

Measuring height



$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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

scene cross ratio

$$\frac{\|t - b\| \|v_z - r\|}{\|r - b\| \|v_z - t\|} = \frac{H}{R}$$

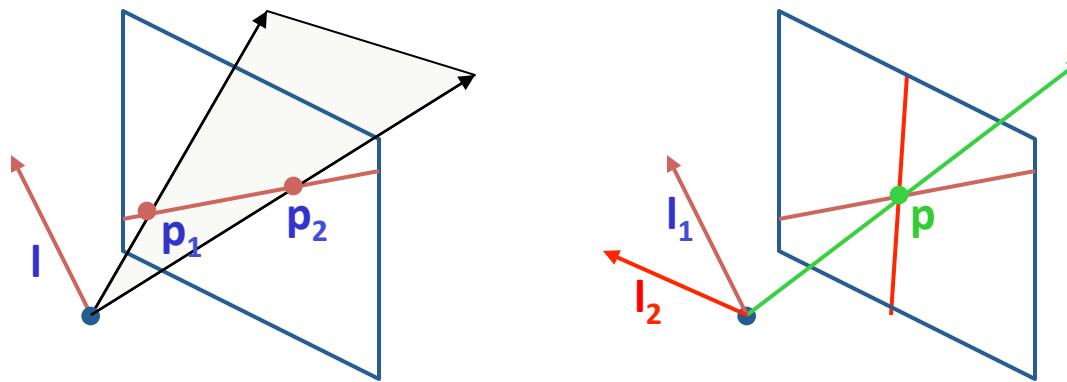
image cross ratio

image points as

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Point and line duality

- A line \mathbf{l} is a homogeneous 3-vector



What is the line \mathbf{l} spanned by rays \mathbf{p}_1 and \mathbf{p}_2 ?

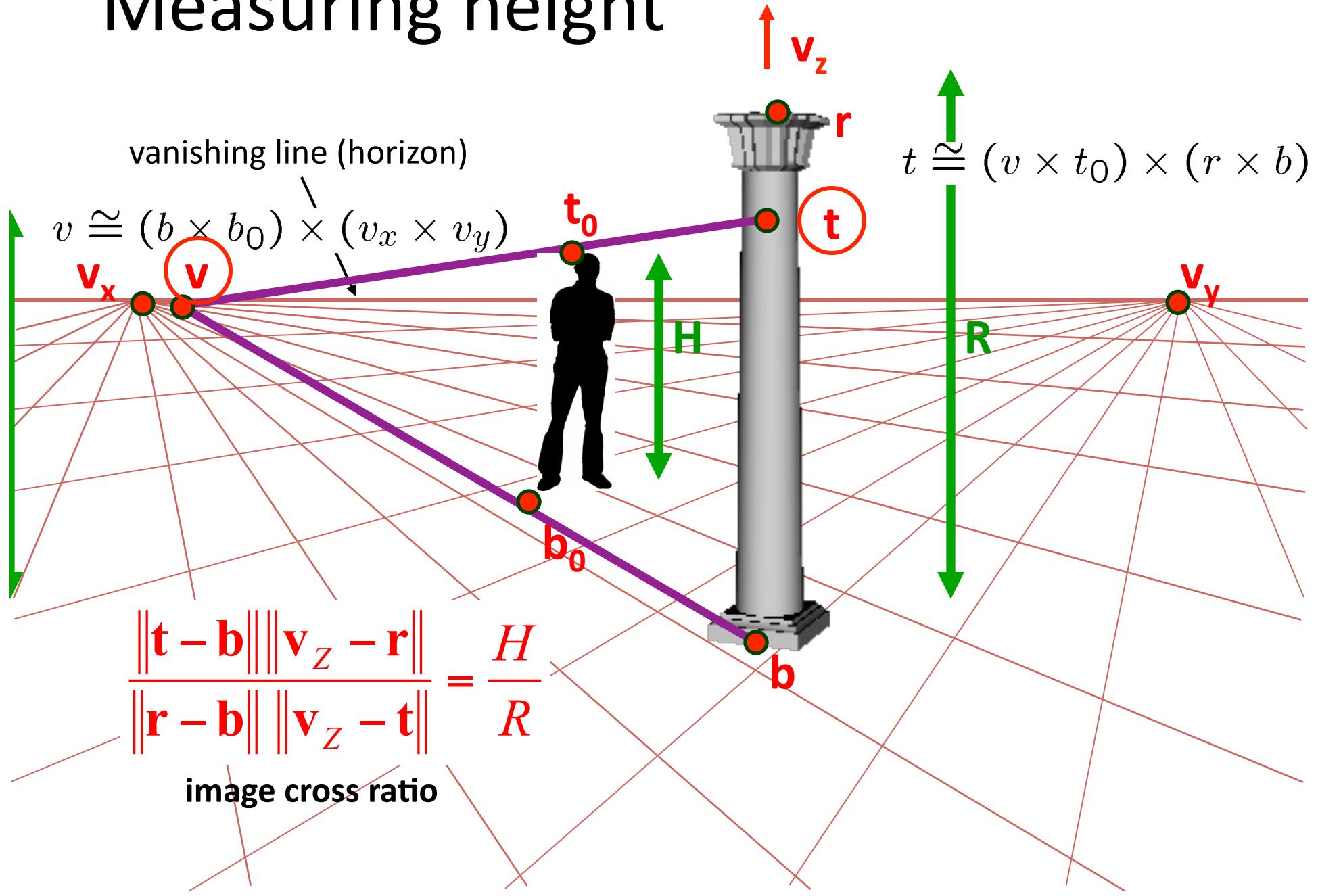
- \mathbf{l} is \perp to \mathbf{p}_1 and $\mathbf{p}_2 \Rightarrow \mathbf{l} = \mathbf{p}_1 \times \mathbf{p}_2$
- \mathbf{l} can be interpreted as a *plane normal*

What is the intersection of two lines \mathbf{l}_1 and \mathbf{l}_2 ?

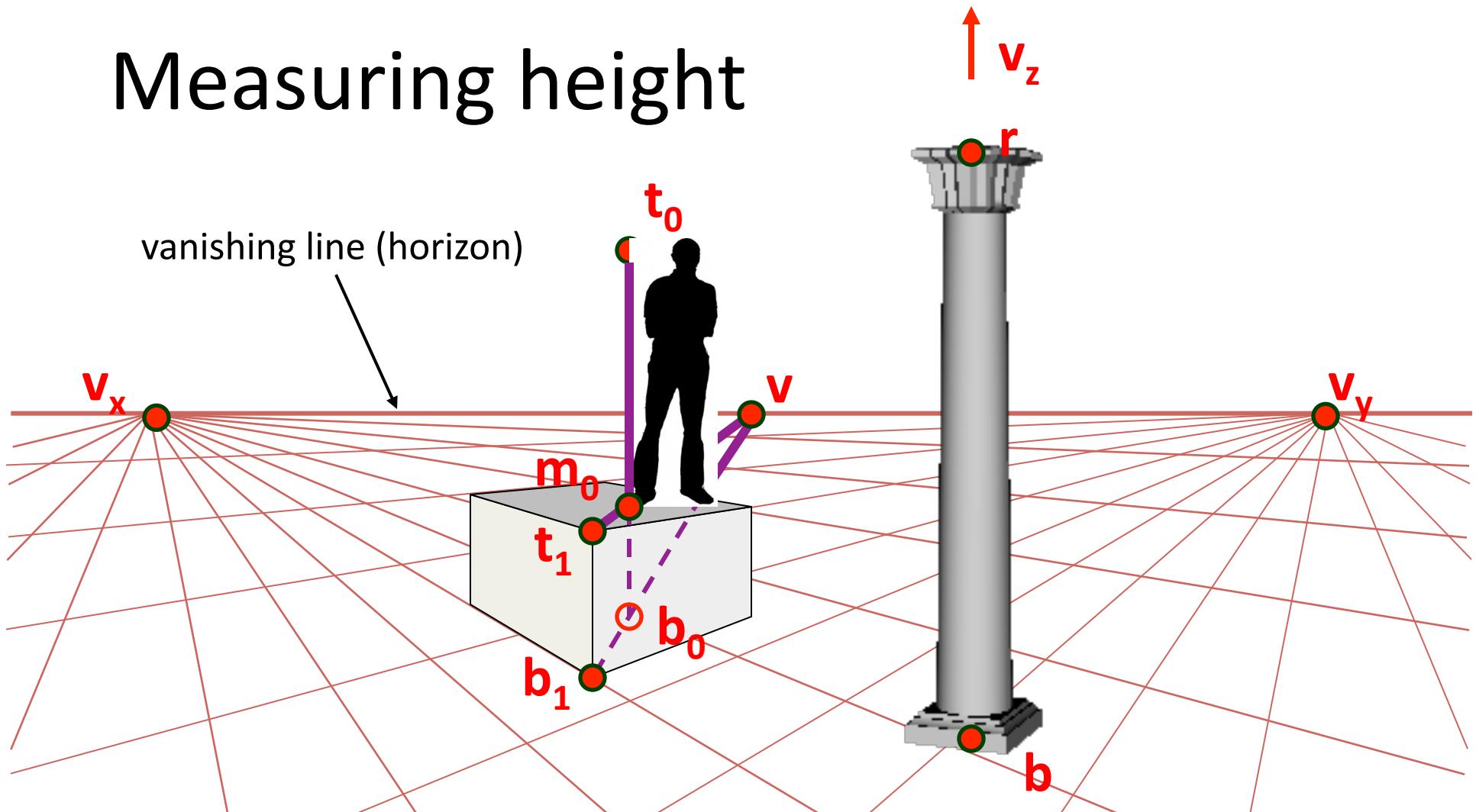
- \mathbf{p} is \perp to \mathbf{l}_1 and $\mathbf{l}_2 \Rightarrow \mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$

Points and lines are *dual* in projective space

Measuring height



Measuring height



What if the point on the ground plane 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 b_0 as shown above

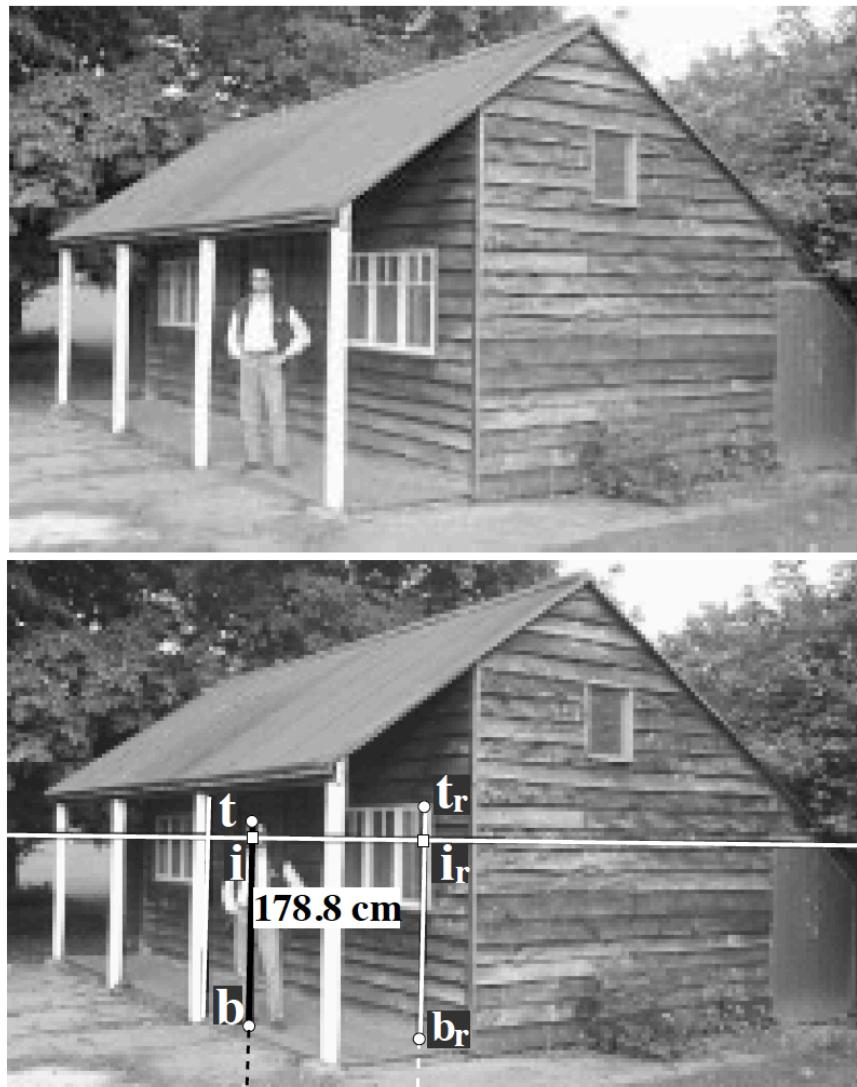


Figure 3: Measuring the height of a person: (top) original image; (bottom) the height of the person is computed from the image as 178.8cm (the true height is 180cm, but note that the person is leaning down a bit on his right foot). The vanishing line is shown in white and the reference height is the segment (t_r, b_r) . The vertical vanishing point is not shown since it lies well below the image. t is the top of the head and b is the base of the feet of the person while i is the intersection with the vanishing line.

3D Modeling from a photograph



St. Jerome in his Study, H. Steenwick

3D Modeling from a photograph



3D Modeling from a photograph



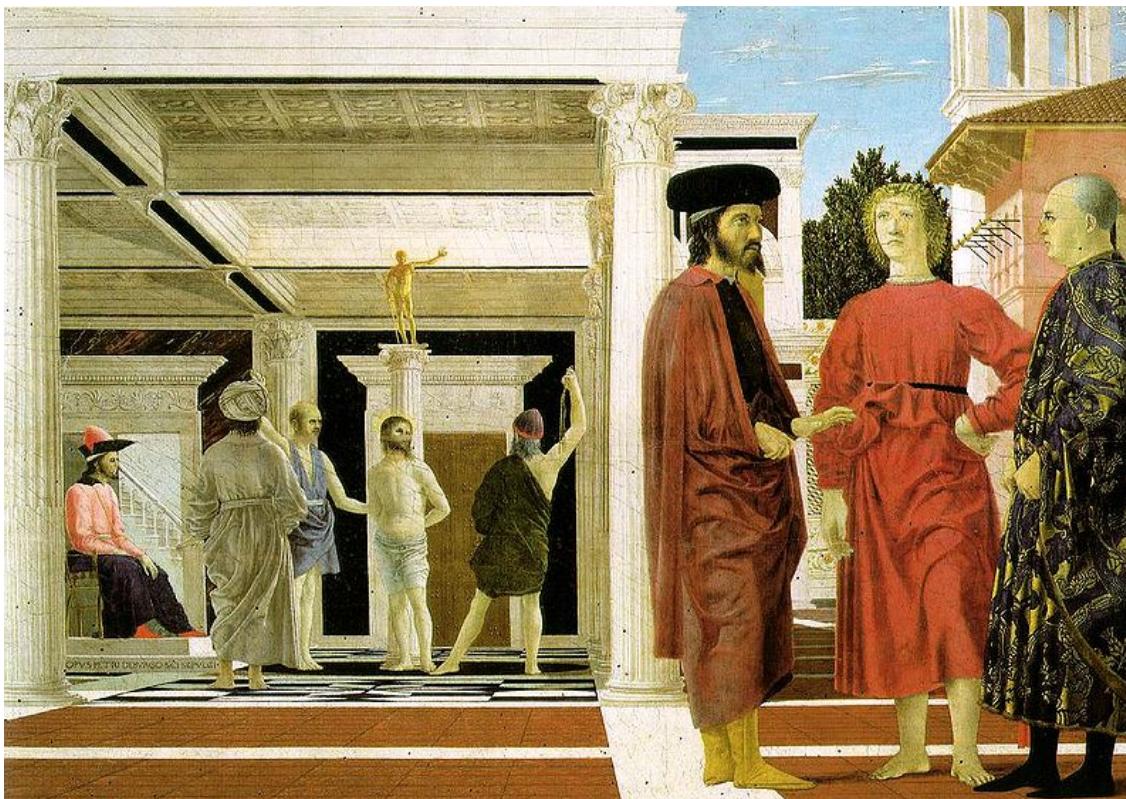
Flagellation, Piero della Francesca

3D Modeling from a photograph



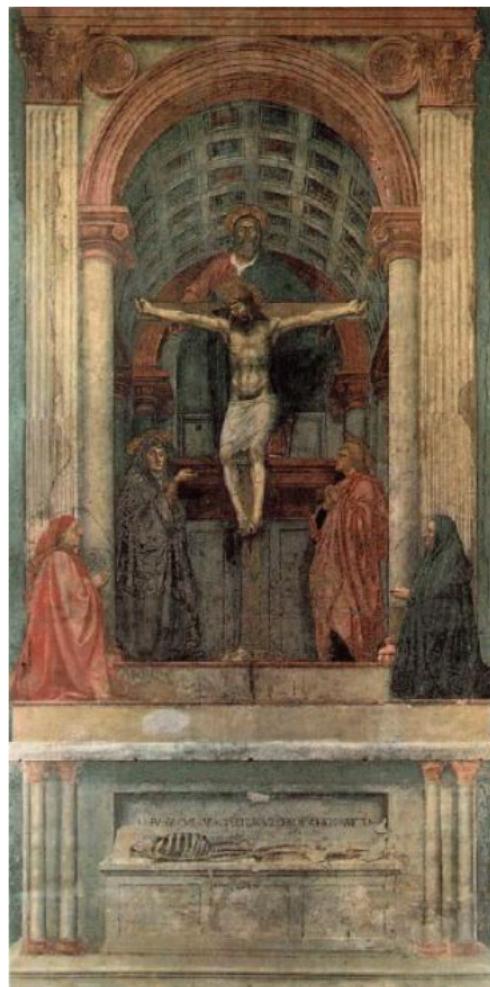
video by Antonio Criminisi

3D Modeling from a photograph



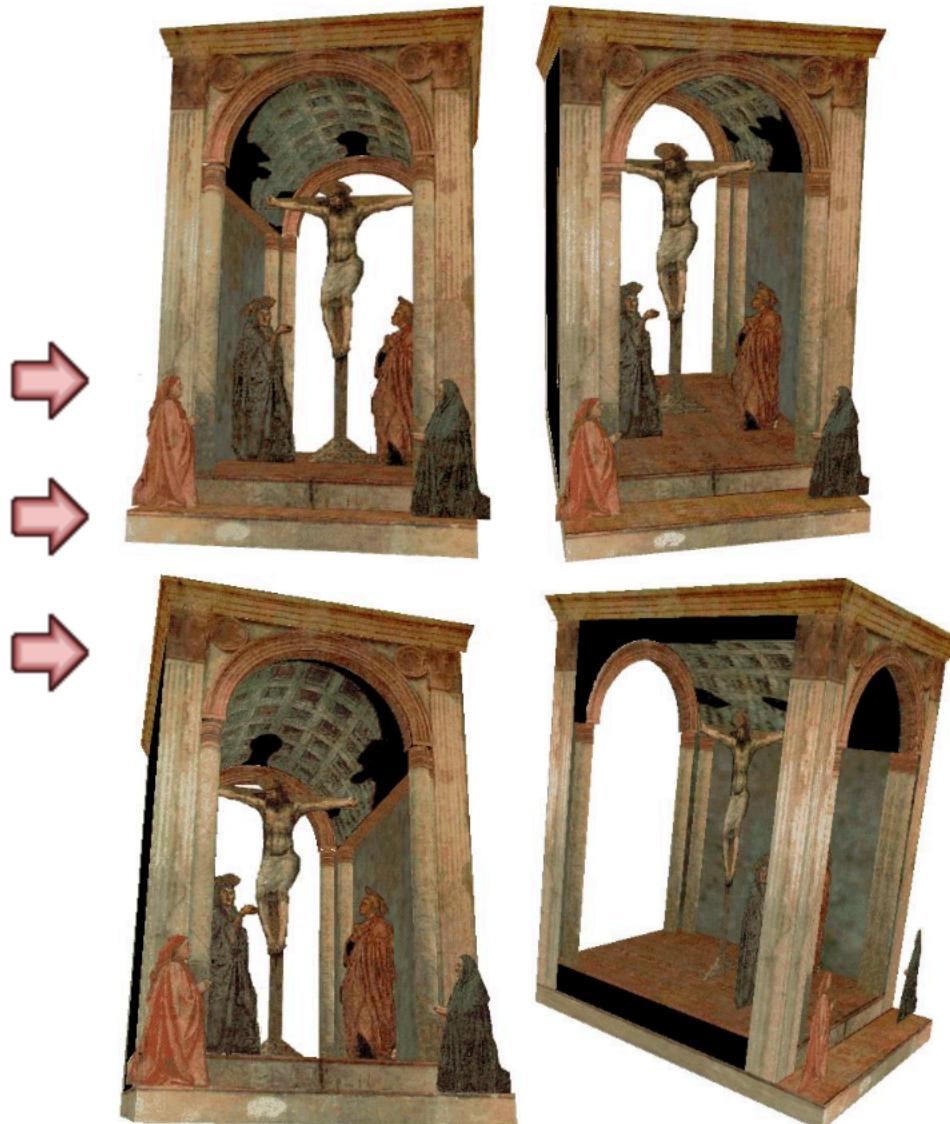
The following example shows the reconstruction of a chapel depicted in one of the earliest and most famous Renaissance frescoes: **La Trinita' (The Trinity)** (1427) by Masaccio (1401-1428).

original fresco



**La Trinita' (1427)
by Masaccio**

images of the reconstructed 3D model



Some Related Techniques

- Image-Based Modeling and Photo Editing
 - [Mok et al., SIGGRAPH 2001](#)
- Single View Modeling of Free-Form Scenes
 - [Zhang et al., CVPR 2001](#)
- Tour Into The Picture
 - [Anjyo et al., SIGGRAPH 1997](#)

Camera calibration

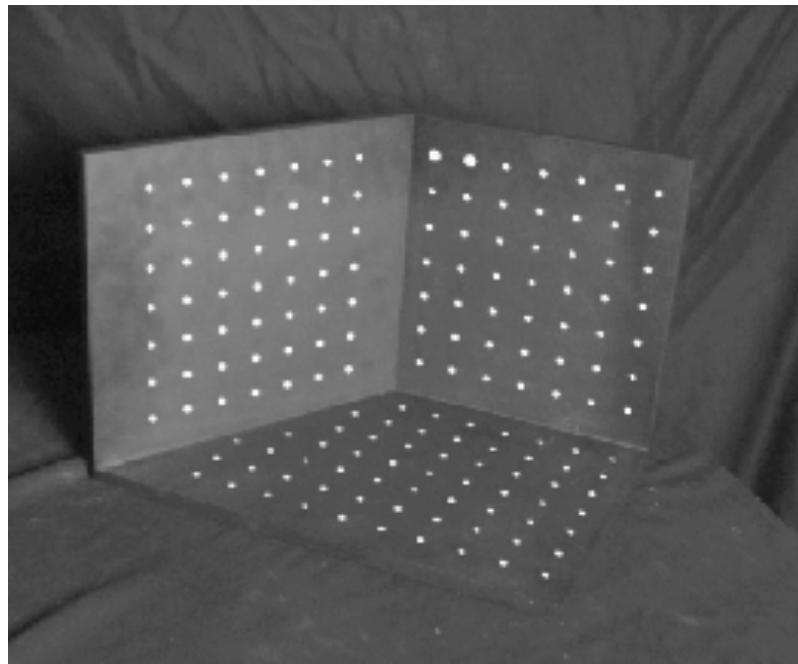
- Goal: estimate the camera parameters
 - Version 1: solve for projection matrix

$$\mathbf{X} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \boldsymbol{\Pi} \mathbf{X}$$

- Version 2: solve for camera parameters separately
 - intrinsics (focal length, principle point, pixel size)
 - extrinsics (rotation angles, translation)
 - radial distortion

Calibration using a reference object

- Place a known object in the scene
 - identify correspondence between image 2D and scene 3D
 - compute mapping from scene to image

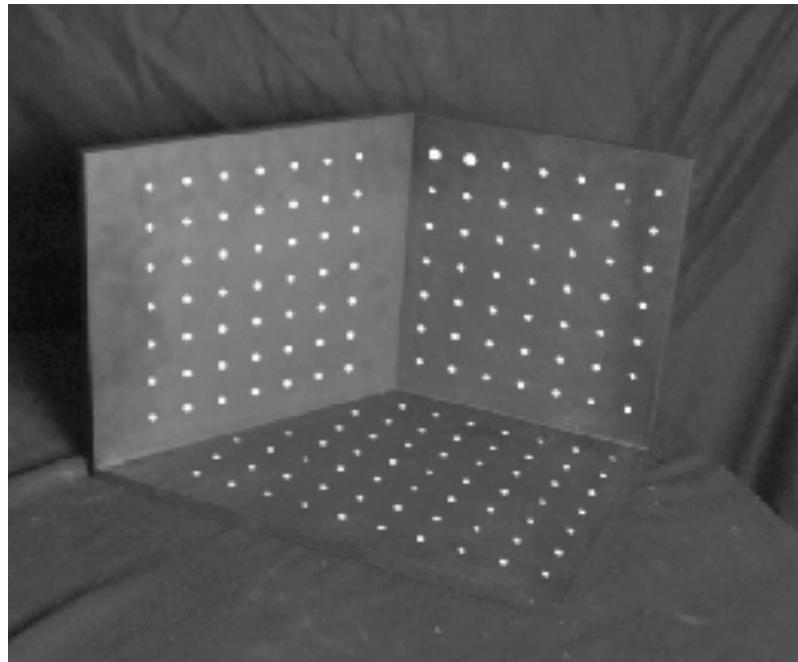


Issues

- must know geometry very accurately
- must know 3D->2D correspondence

Estimating the projection matrix

- Place a known object in the scene
 - identify correspondence between image and scene
 - compute mapping from scene to image



$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Direct linear calibration

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}$$

$$v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}$$

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Direct linear calibration

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} = \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Can solve for m_{ij} by linear least squares

- use eigenvector trick that we used for homographies. $A x = 0$

$$\begin{array}{c} \mathbf{A} \\ 2n \times 9 \end{array} \quad \begin{array}{c} \mathbf{h} \\ 9 \end{array} \quad \begin{array}{c} \mathbf{0} \\ 2n \end{array}$$

Defines a least squares problem: minimize $\|Ah - 0\|^2$

- Since \mathbf{h} is only defined up to scale, solve for unit vector $\hat{\mathbf{h}}$
- Solution: $\hat{\mathbf{h}} = \text{eigenvector of } \mathbf{A}^T \mathbf{A} \text{ with smallest eigenvalue}$
- Works with 4 or more points

Direct linear calibration

- Advantage:
 - Very simple to formulate and solve
- Disadvantages:
 - Doesn't tell you the camera parameters
 - Doesn't model radial distortion
 - Hard to impose constraints (e.g., known f)
 - Doesn't minimize the right error function

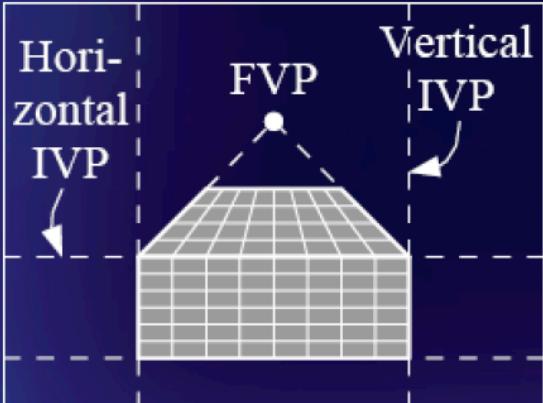
Nonlinear *methods* are preferred

- Define error function E between projected 3D points and image positions: nonlinear function of intrinsics, extrinsics, radial distortion
- Minimize E using nonlinear optimization techniques

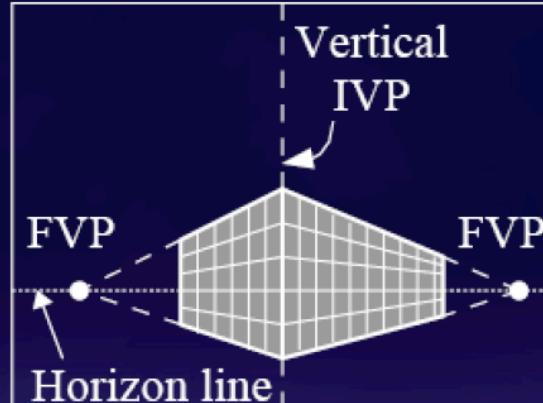
Summary

- Known correspondences
 - (u_i, v_i) and (X_i, Y_i, Z_i)
- Compute m_{ij} solving system of linear equations
 - May use this to initialize non linear error minimization problem to recover more accurate m_{ij}

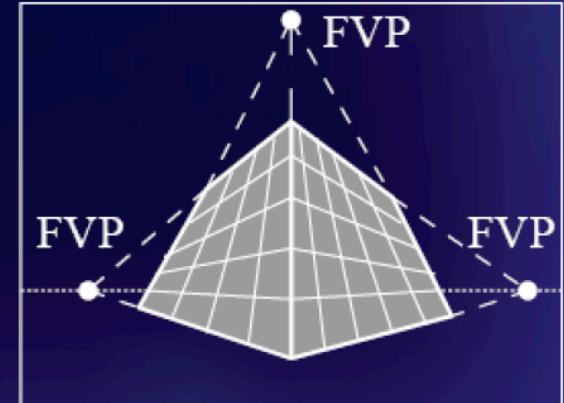
Calibration from vanishing points



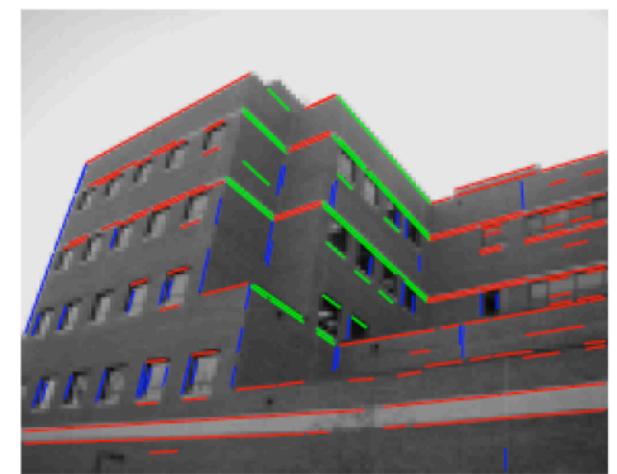
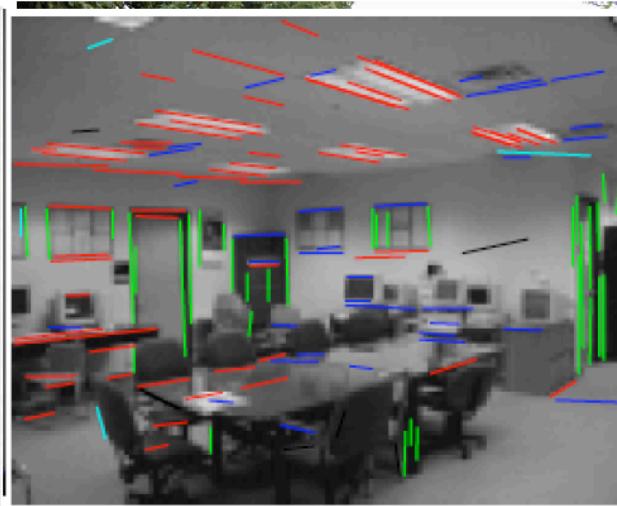
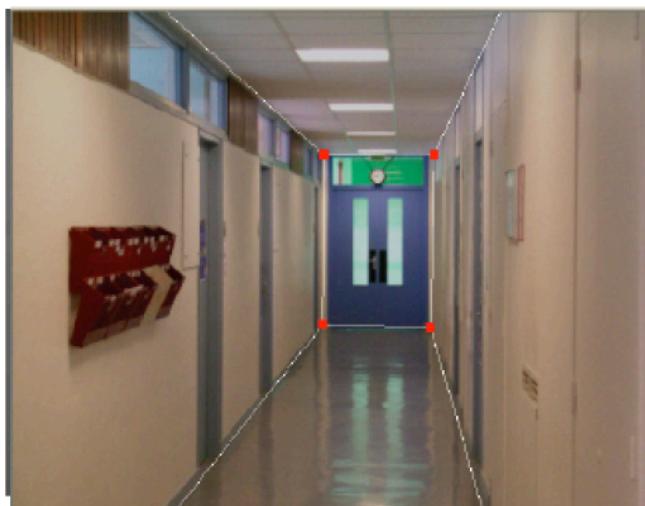
1 finite vanishing point,
2 infinite vanishing points



2 finite vanishing points,
1 infinite vanishing point



3 finite vanishing points



- From vanishing points corresponding to 3 orthogonal directions of world

$$e_i = [1, 0, 0]^T, e_j = [0, 1, 0]^T, e_k = [0 \quad 0 \quad 1]^T$$

$$\mathbf{v}_i = \bar{K}R\bar{e}_i, \mathbf{v}_j = \bar{K}R\bar{e}_j, \mathbf{v}_k = \bar{K}R\bar{e}_k$$

$$e_i^T e_j = 0$$

$$\mathbf{v}_i^T K^{-T} R R^T K^{-1} \mathbf{v}_j = \mathbf{v}_i^T K^{-T} K^{-1} \mathbf{v}_j = 0$$

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} K^{-1} = \begin{bmatrix} 1/f & 0 & -u_0/f \\ 0 & 1/f & -v_0/f \\ 0 & 0 & 1 \end{bmatrix}$$

$$v_i^T K^{-T} K^{-1} v_j = 0$$

$$v_j^T K^{-T} K^{-1} v_k = 0$$

$$v_i^T K^{-T} K^{-1} v_k = 0$$

- 3 finite vanishing points: get f , u_0 , v_0

Rotation from vanishing points

- R_{1c} 1st column vector of Rotation matrix

$$R = [R_{1c} \quad R_{2c} \quad R_{3c}]$$

$$\lambda v_i = K R e_i \quad e_i = [1, 0, 0]^T$$

$$R_{1c} = \lambda K^{-1} v_i$$

- λ from $\|R_{1c}\|_2 = 1$

Vanishing points and projection matrix

$$\Pi = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{bmatrix} = [\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4]$$

- Projection of x axis $= \mathbf{v}_x$ (X vanishing point)
- similarly, $\pi_2 = \mathbf{v}_Y$, $\pi_3 = \mathbf{v}_Z$
- $\pi_4 = \Pi[0 \ 0 \ 0 \ 1]^T$ = projection of world origin

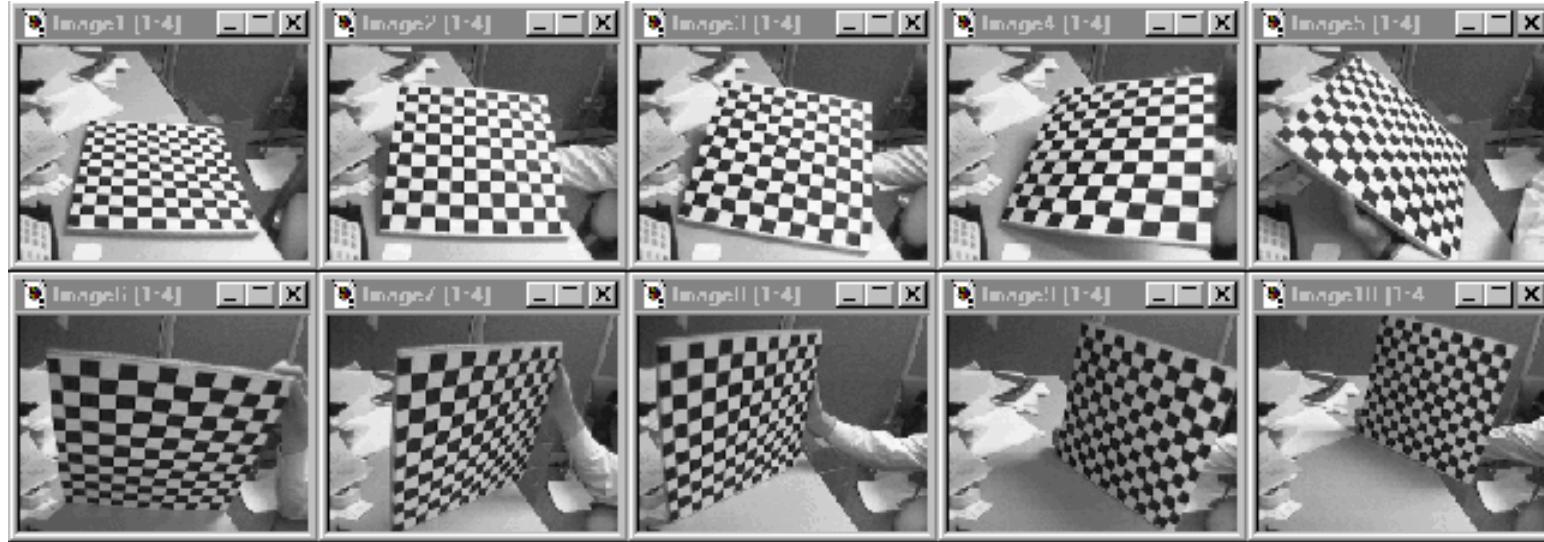
$$\Pi = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \mathbf{v}_Z \quad \mathbf{0}]$$

Not So Fast! We only know \mathbf{v} 's up to a scale factor

$$\Pi = [a\mathbf{v}_X \quad b\mathbf{v}_Y \quad c\mathbf{v}_Z \quad \mathbf{0}]$$

- Can fully specify by providing 3 reference points

Alternative: multi-plane calibration



Images courtesy Jean-Yves Bouguet, Intel Corp.

Advantage

- Only requires a plane
- Don't have to know positions/orientations
- Good code available online! (including in OpenCV)
 - Matlab version by Jean-Yves Bouget:
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
 - Zhengyou Zhang's web site: <http://research.microsoft.com/~zhang/Calib/>

Next time

Stereo