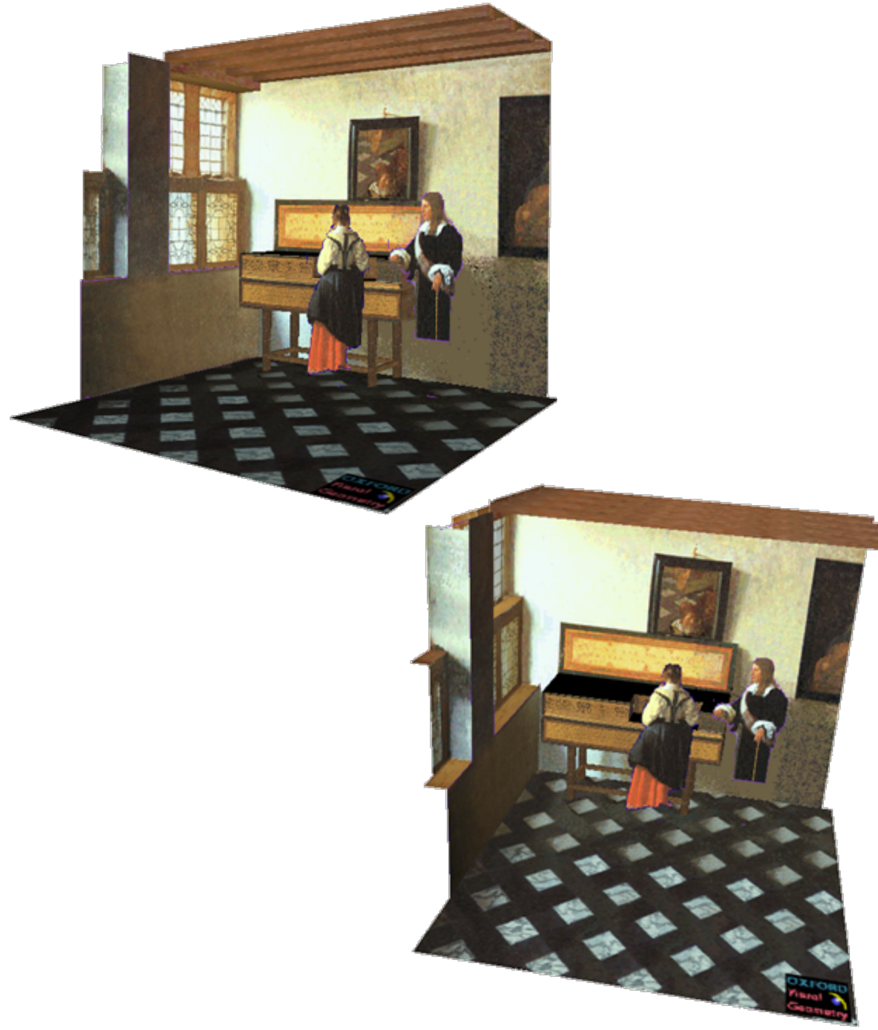
Today

Projective Geometry

Single View Modeling

Vermeer's *Music Lesson*



Reconstructions by Criminisi et al.

on to 3D…

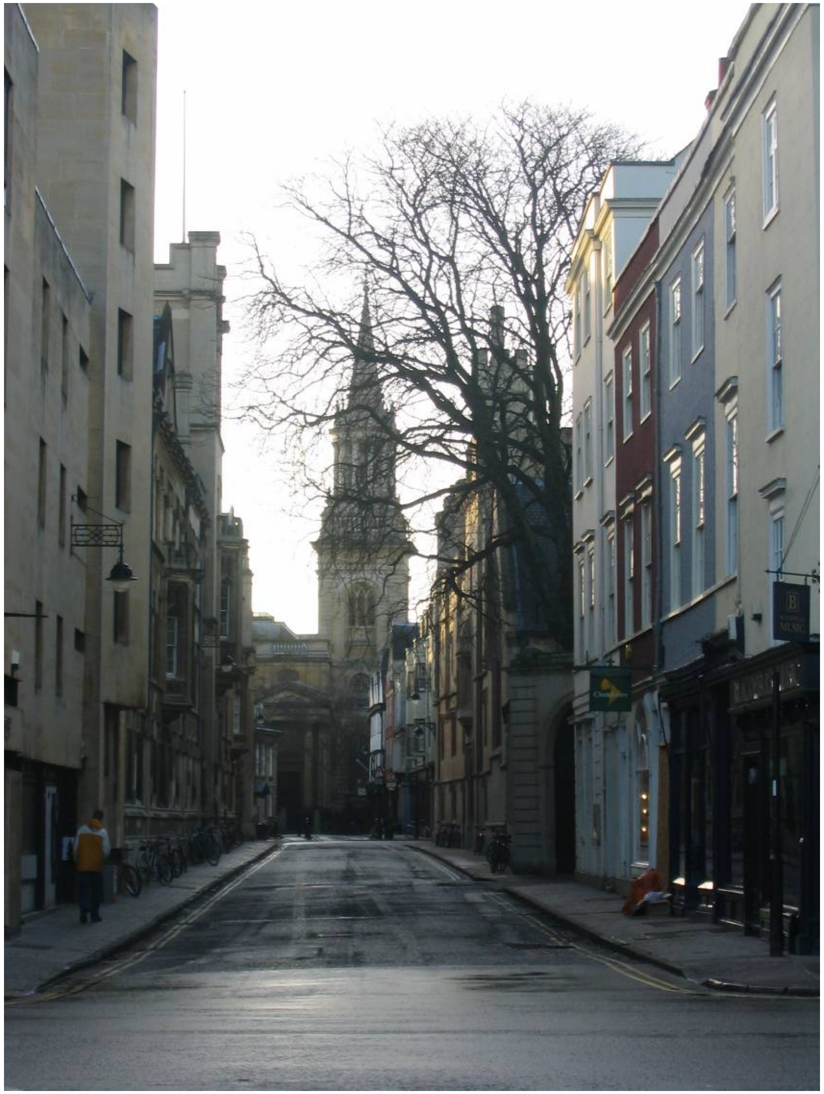
Enough of images!

We want more from the image

We want real 3D scene

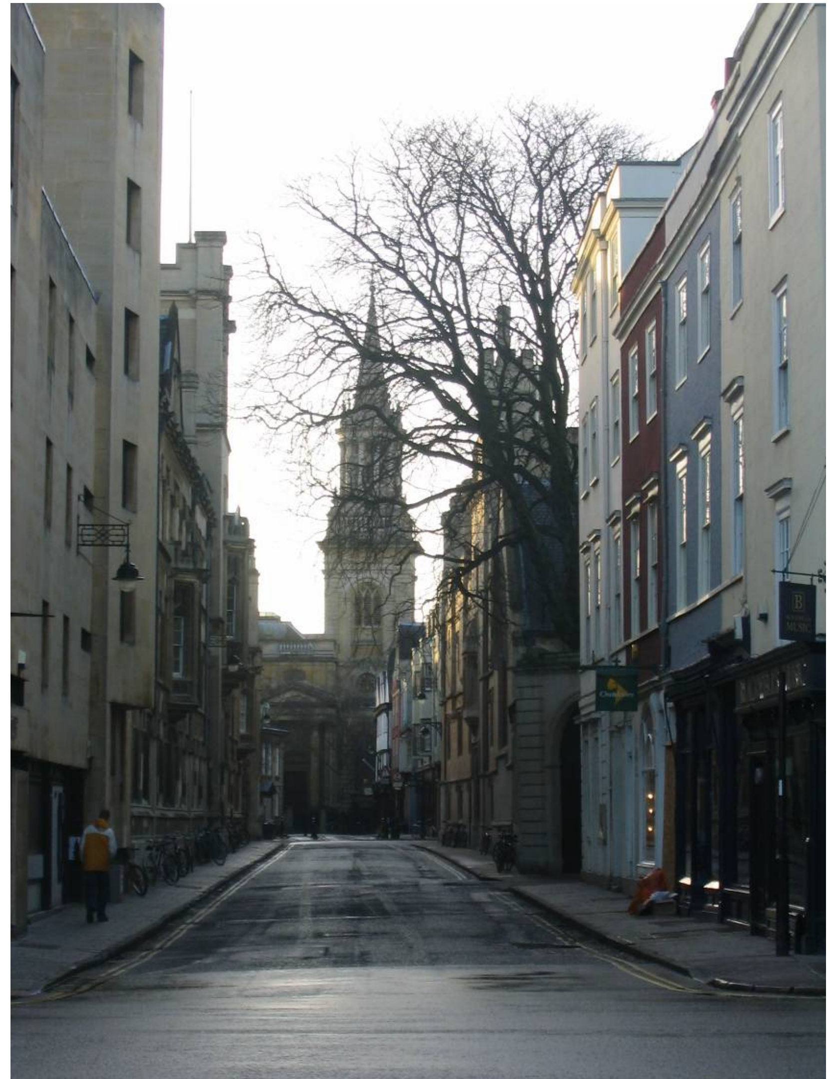
walk-throughs:

- Camera rotation
- Camera translation



So, what can we do here?

- Model the scene as a set of planes!

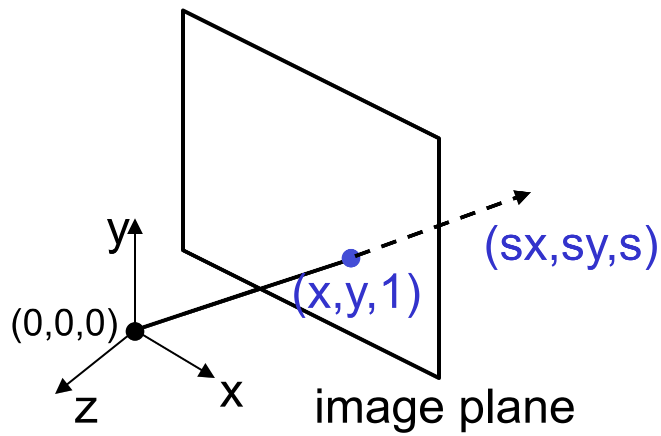


Another example

- http://mit.edu/jxiao/museum/

The projective plane

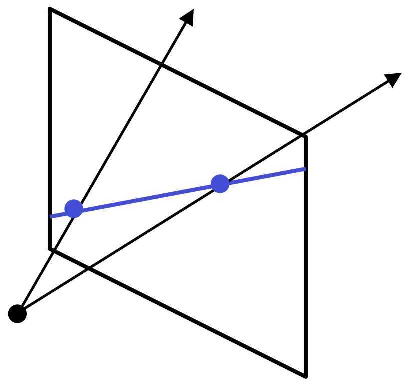
- Why do we need homogeneous coordinates?
  – represent points at infinity, homographies, perspective projection, multi-view relationships
- What is the geometric intuition?
  – a point in the image is a *ray* in projective space



- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  – all points on the ray are equivalent: (x, y, 1) ≅ (sx, sy, s)

Projective lines

- What does a line in the image correspond to in projective space?



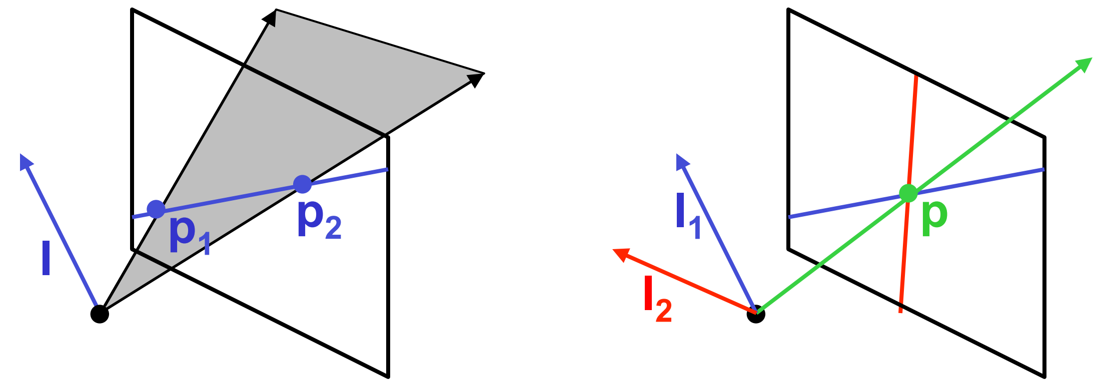
- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

$$\text{in vector notation:} \quad 0 = \underbrace{\begin{bmatrix} a & b & c \end{bmatrix}}_{\mathbf{l}} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{p}}$$

- A line is also represented as a homogeneous 3-vector **l**

Point and line duality

- A line $\mathbf{l}$ is a homogeneous 3-vector
- It is $\perp$ to every point (ray) $\mathbf{p}$ on the line: $\mathbf{l}\ \mathbf{p}=0$



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 $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- $\mathbf{l}$ is the plane normal

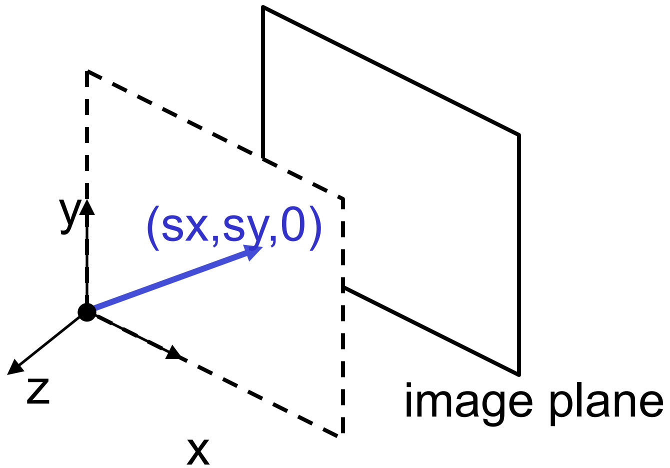
What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

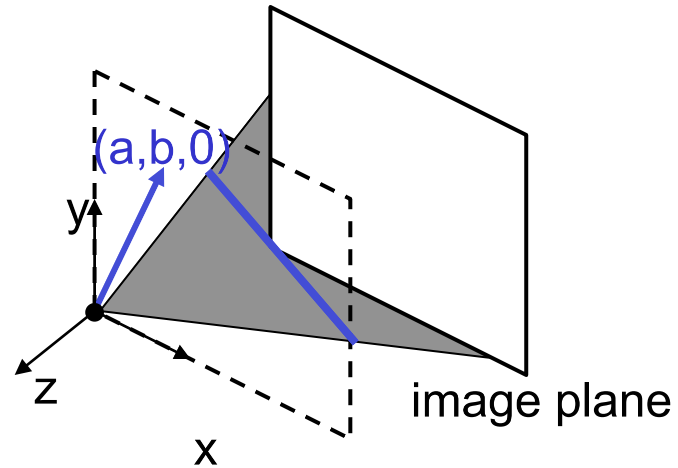
- $\mathbf{p}$ is $\perp$ to $\mathbf{l_1}$ and $\mathbf{l_2}$ $\Rightarrow$ $\mathbf{p} = \mathbf{l_1} \times \mathbf{l_2}$

Points and lines are *dual* in projective space

- can switch the meanings of points and lines to get another formula

Ideal points and lines





- Ideal point ("point at infinity")
  - $p \cong (x, y, 0)$ – parallel to image plane
  - It has infinite image coordinates

Ideal line

- $l \cong (a, b, 0)$ – parallel to image plane
- Corresponds to a line in the image (finite coordinates)

Homographies of points and lines

- Computed by 3x3 matrix multiplication
  - To transform a point: $\mathbf{p}' = \mathbf{Hp}$
  - To transform a line: $\mathbf{lp}=0 \rightarrow \mathbf{l}'\mathbf{p}'=0$
    - $0 = \mathbf{lp} = \mathbf{lH}^{-1}\mathbf{Hp} = \mathbf{lH}^{-1}\mathbf{p}' \Rightarrow \mathbf{l}' = \mathbf{lH}^{-1}$
    - lines are transformed by postmultiplication of $\mathbf{H}^{-1}$

3D projective geometry

- These concepts generalize naturally to 3D
  - Homogeneous coordinates
    - Projective 3D points have four coords: **P** = (X,Y,Z,W)
  - Duality
    - A plane **N** is also represented by a 4-vector
    - Points and planes are dual in 4D: **N P**=0
  - Projective transformations
    - Represented by 4x4 matrices **T**: **P' = TP, N' = N** $\mathbf{T}^{-1}$

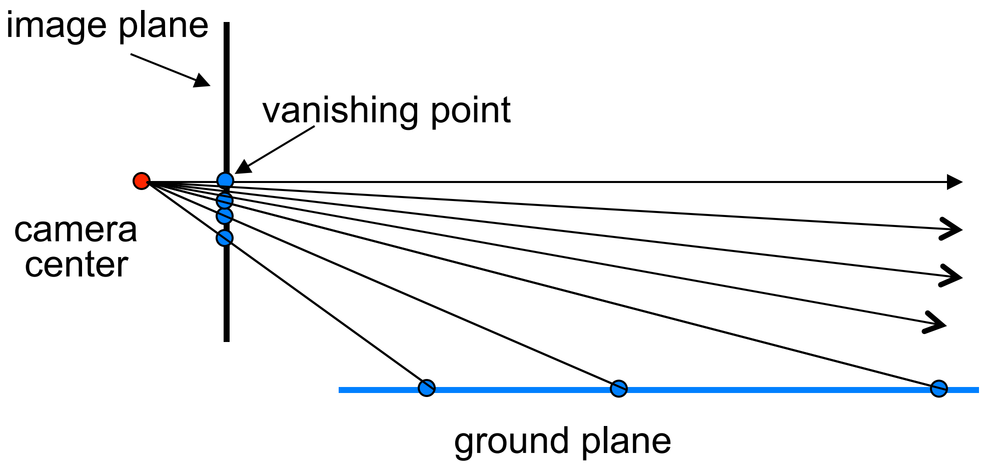
3D to 2D: “perspective” projection

- Matrix Projection: $\mathbf{p} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi P}$

What is *not* preserved under perspective projection?

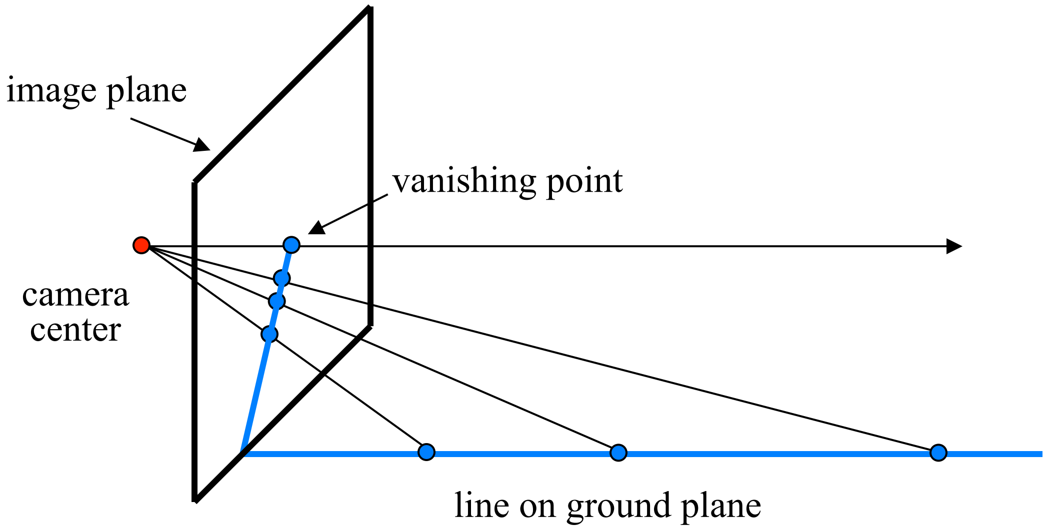
What IS preserved?

Vanishing points

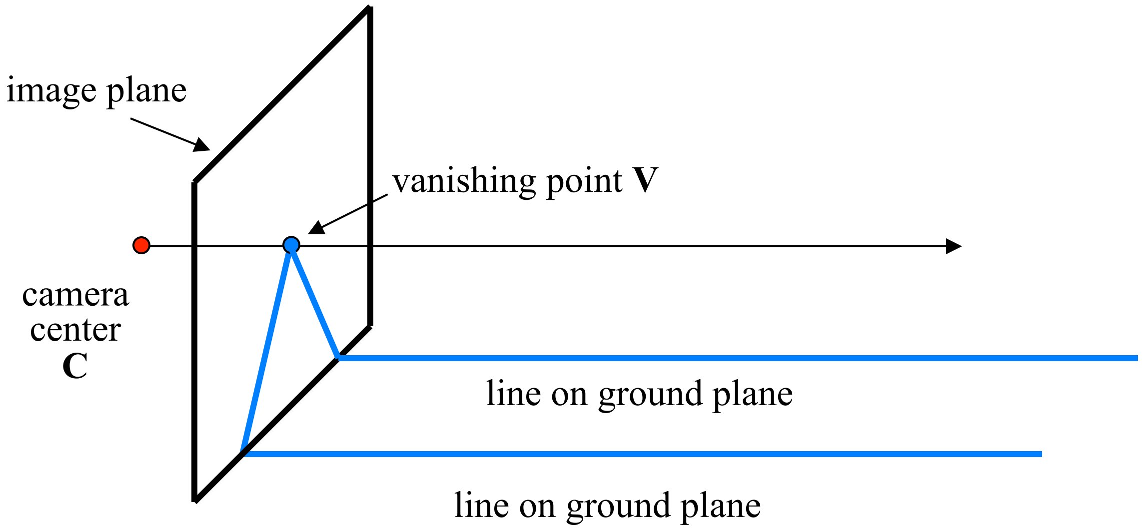


- Vanishing point
  - projection of a point at infinity

Vanishing points (2D)

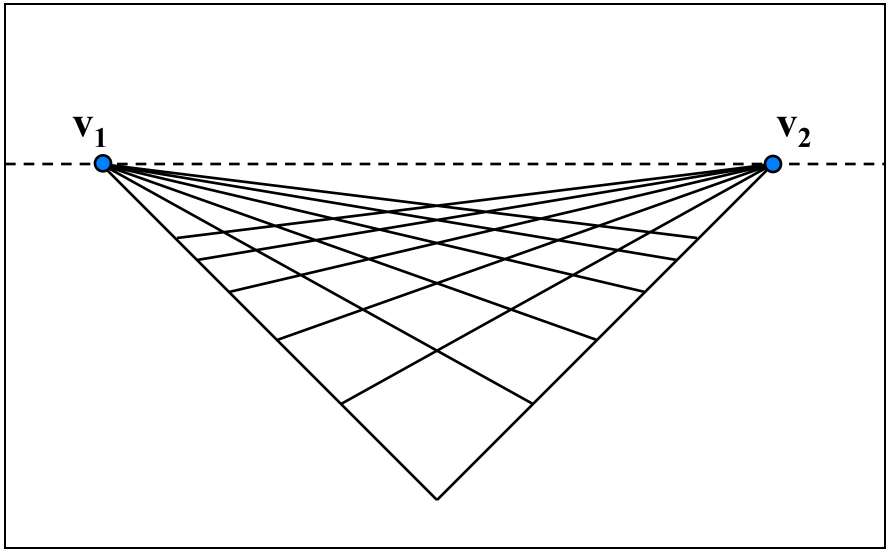


Vanishing points



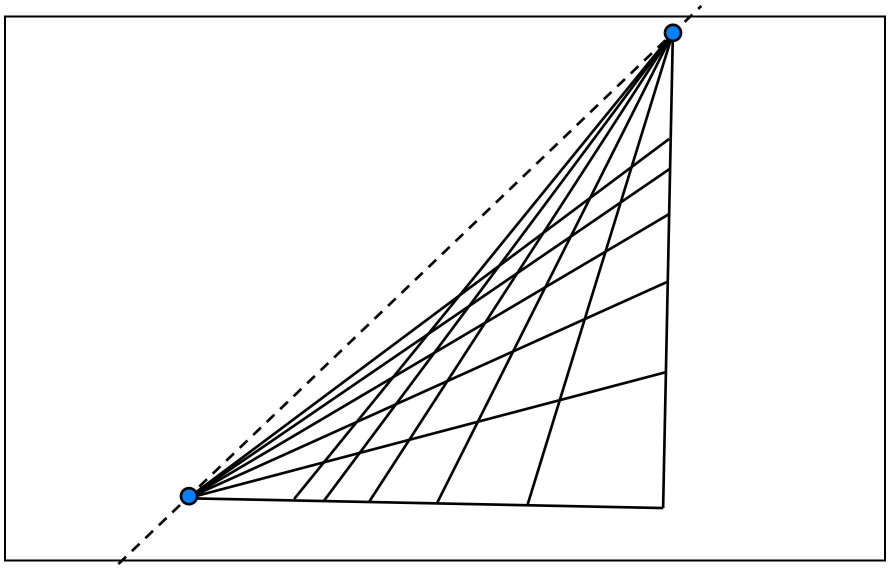
- Properties
  - Any two parallel lines have the same vanishing point **v**
  - The ray from **C** through **v** is parallel to the lines
  - An image may have more than one vanishing point
    - in fact every pixel is a potential vanishing point

Vanishing lines



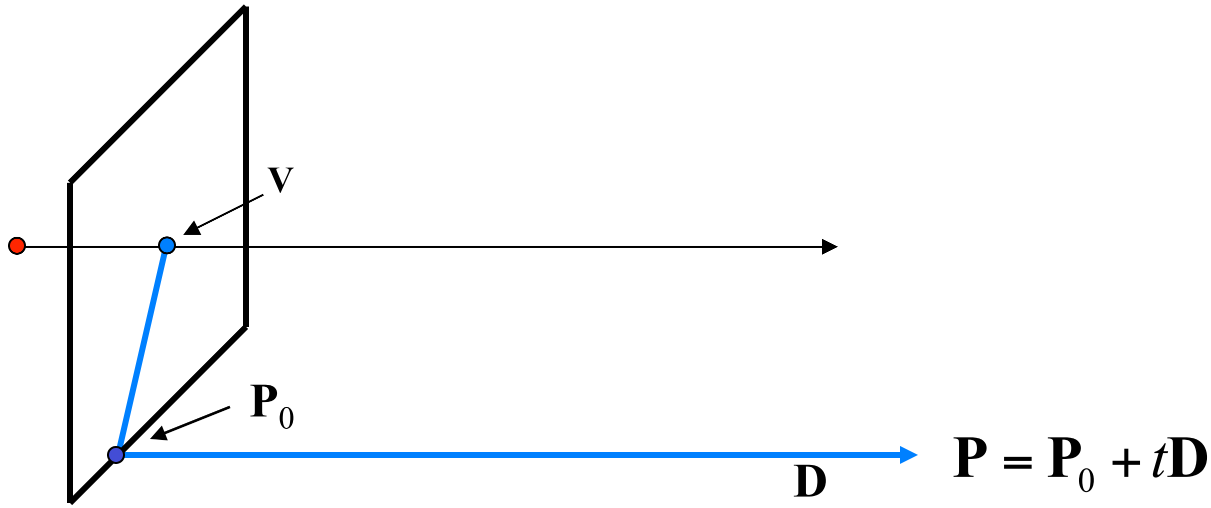
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes define different vanishing lines

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 define different vanishing lines

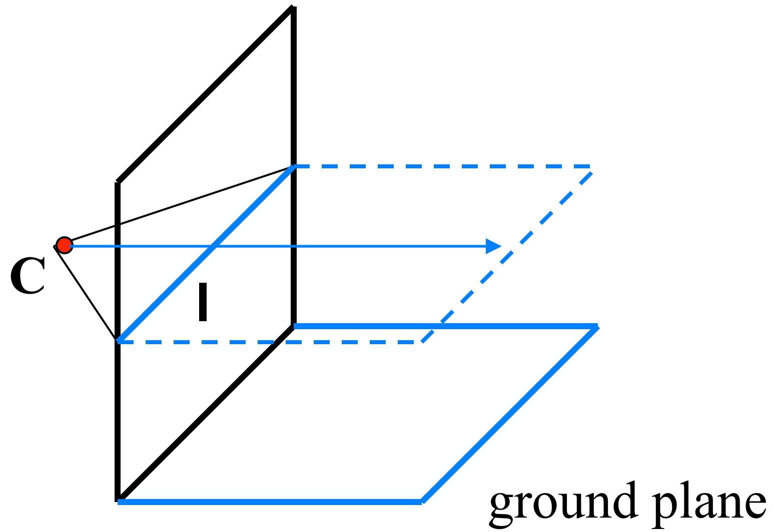
Computing vanishing points



$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

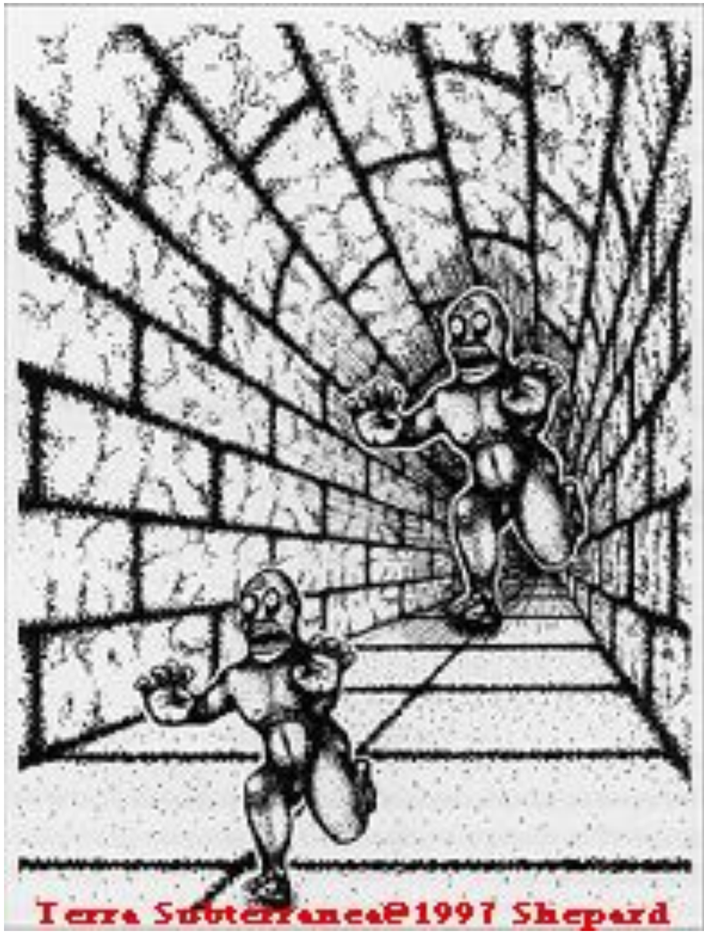
- Properties $\mathbf{v} = \mathbf{\Pi P}_\infty$
  - $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
  - They depend only on line *direction*
  - Parallel lines $\mathbf{P}_0 + t\mathbf{D}$, $\mathbf{P}_1 + t\mathbf{D}$ intersect at $\mathbf{P}_\infty$

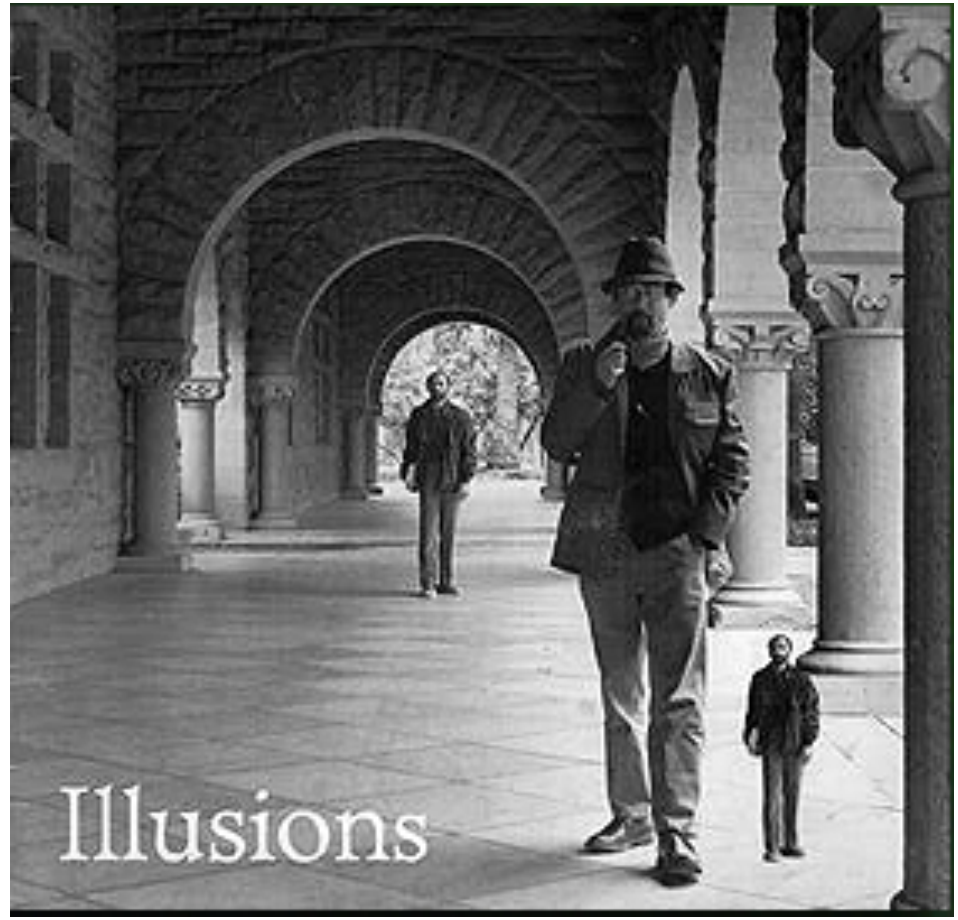
Computing vanishing lines



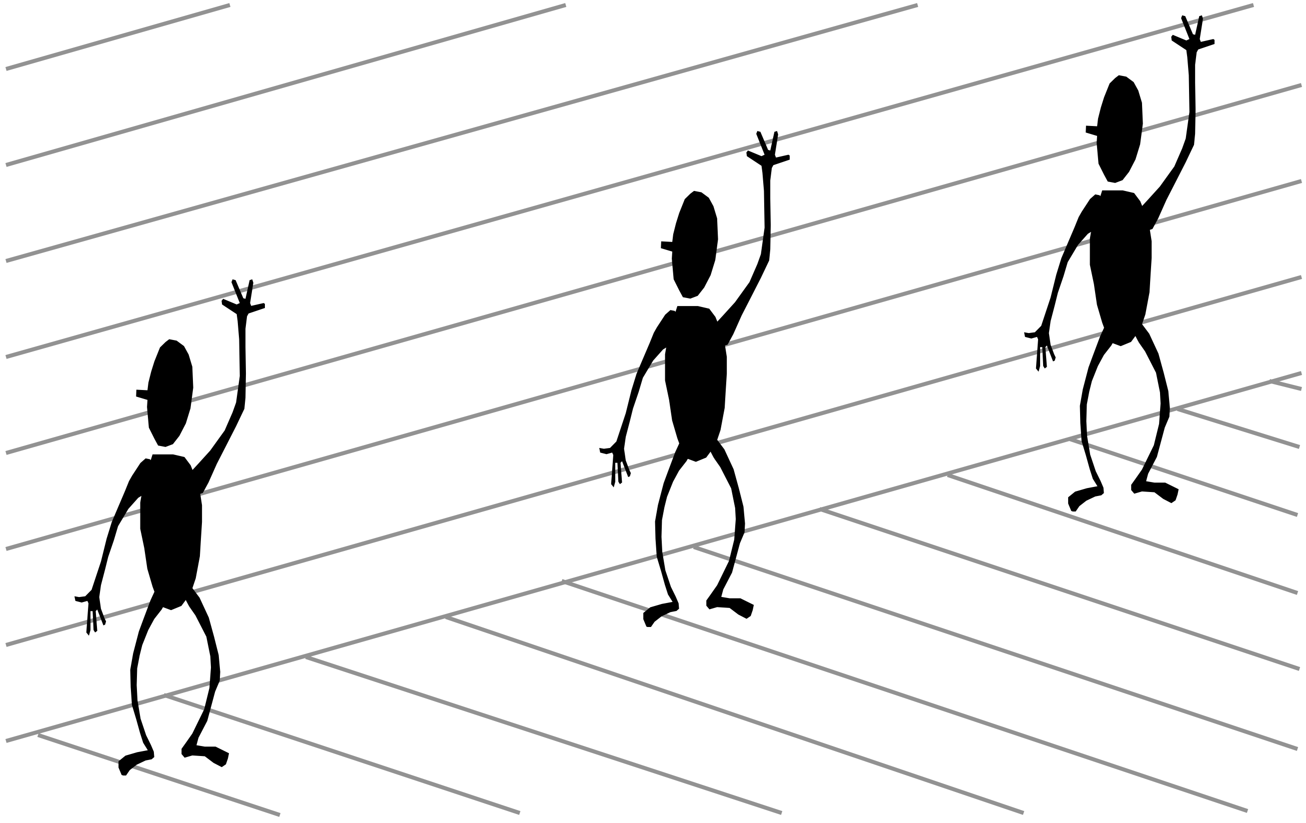
- Properties
  - **l** is intersection of horizontal plane through **C** with image plane
  - Compute **l** from two sets of parallel lines on ground plane
  - All points at same height as **C** project to **l**
    - points higher than C project above l
  - Provides way of comparing height of objects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

Fun with vanishing points

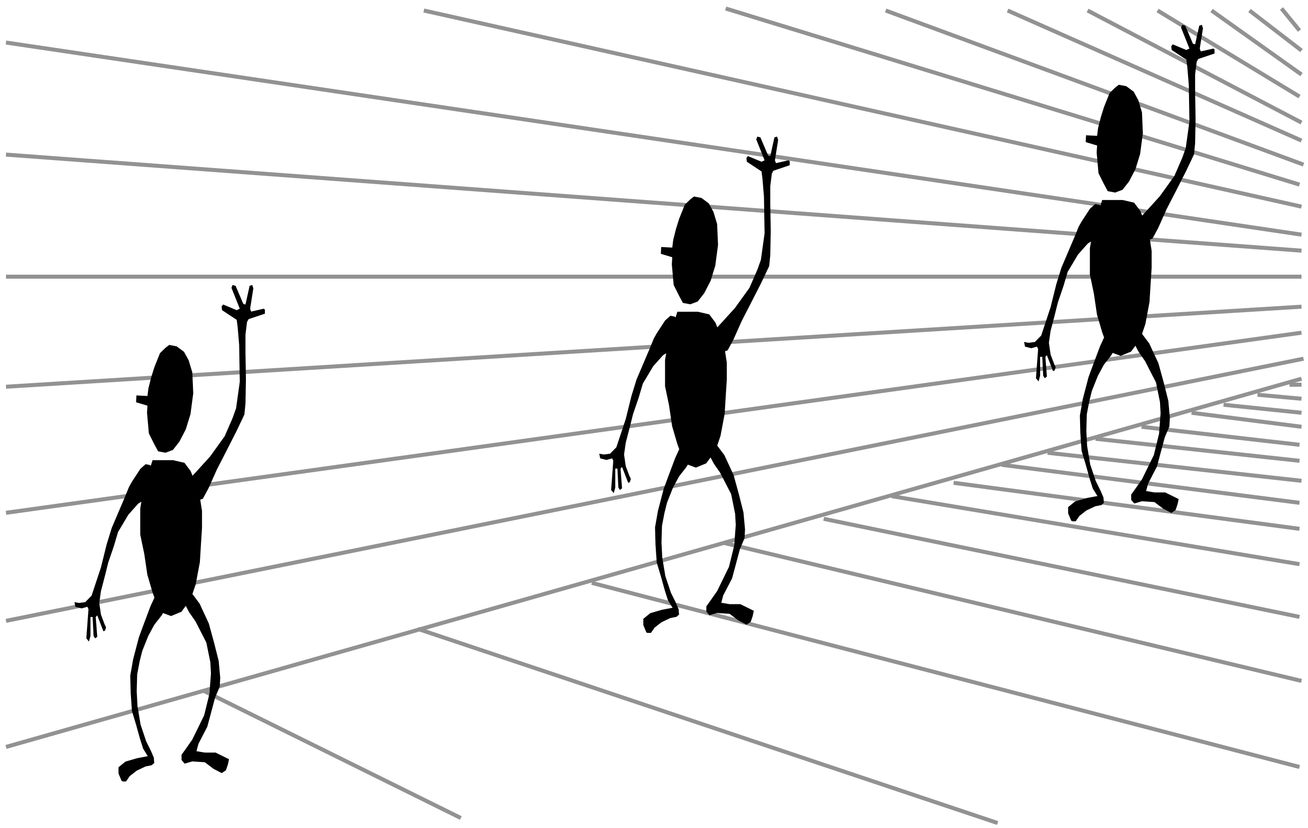




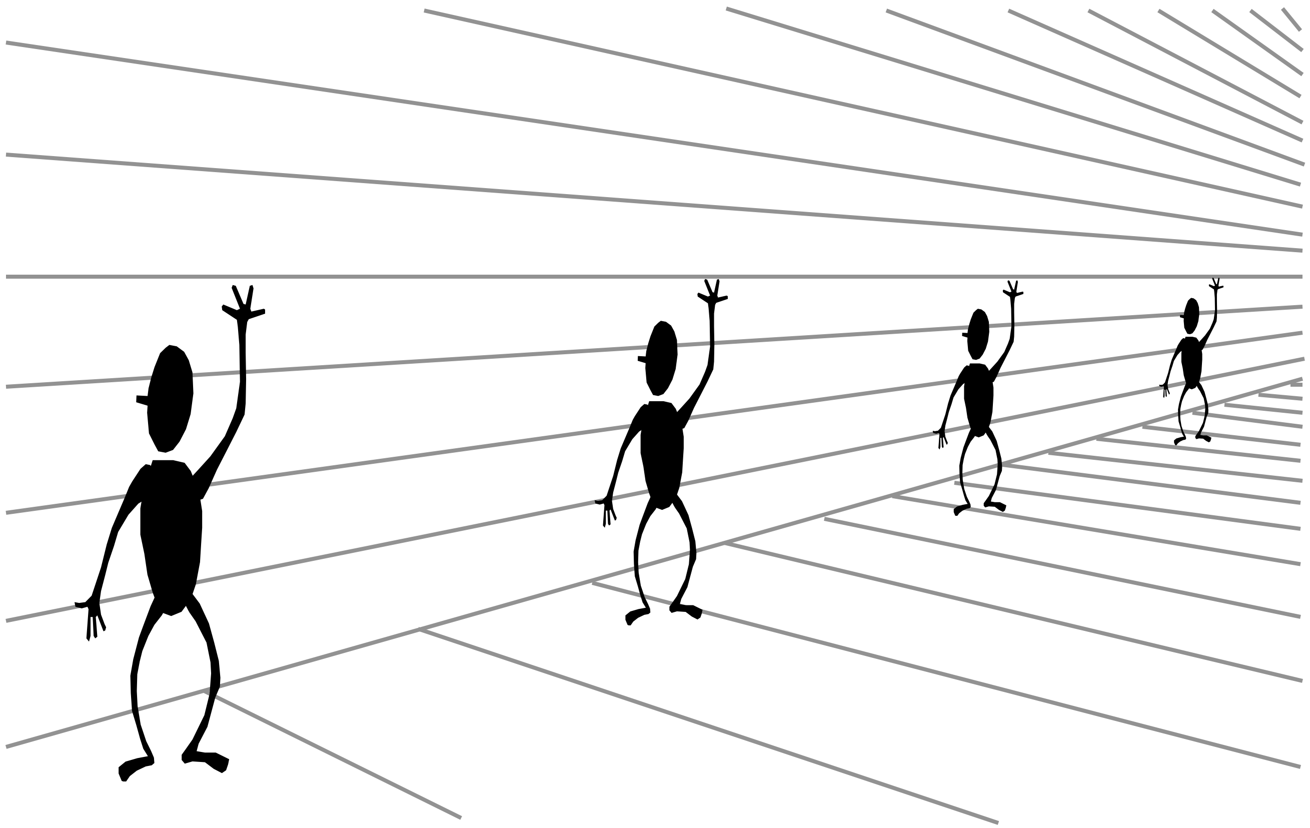
Perspective cues



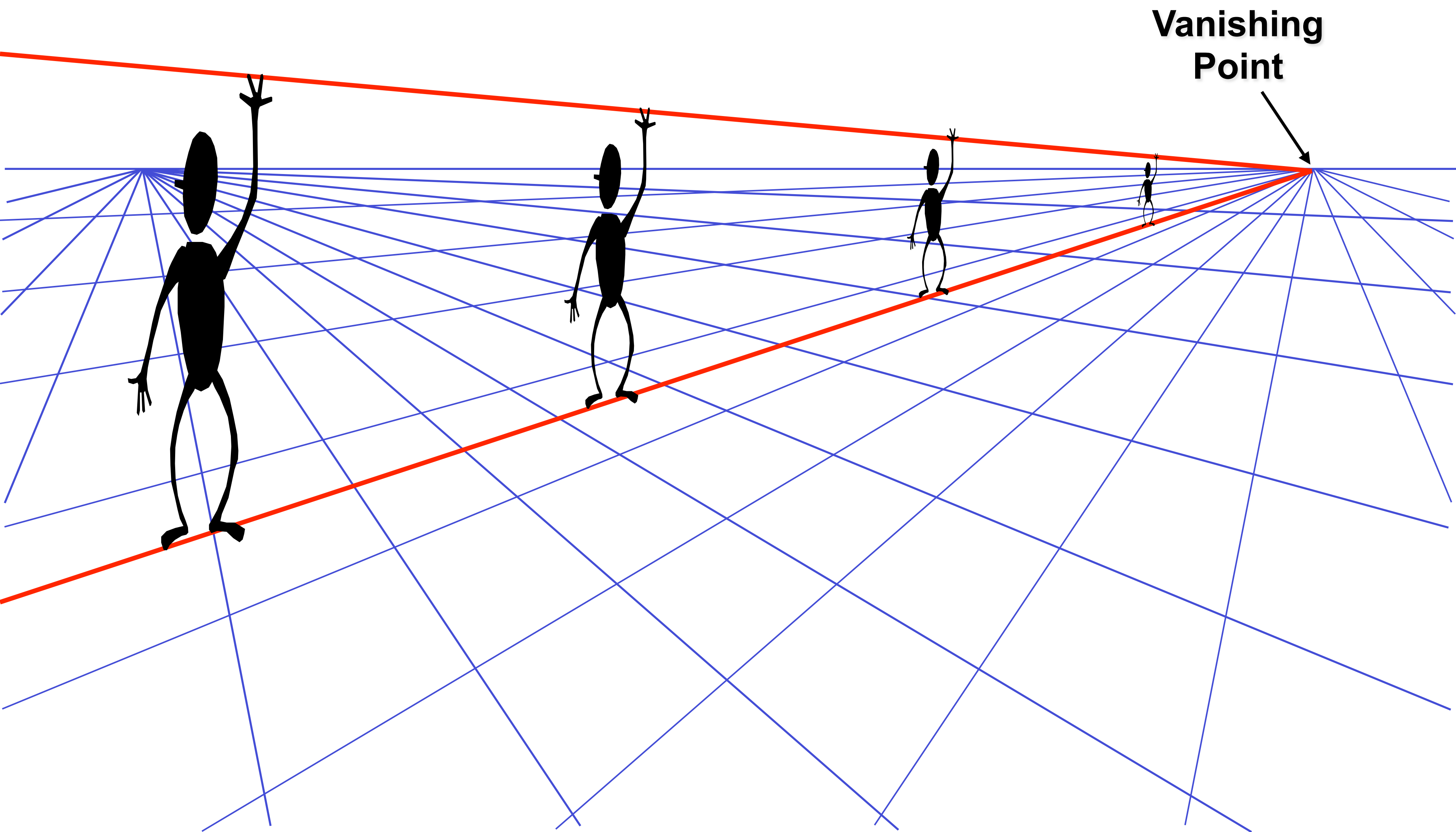
Perspective cues



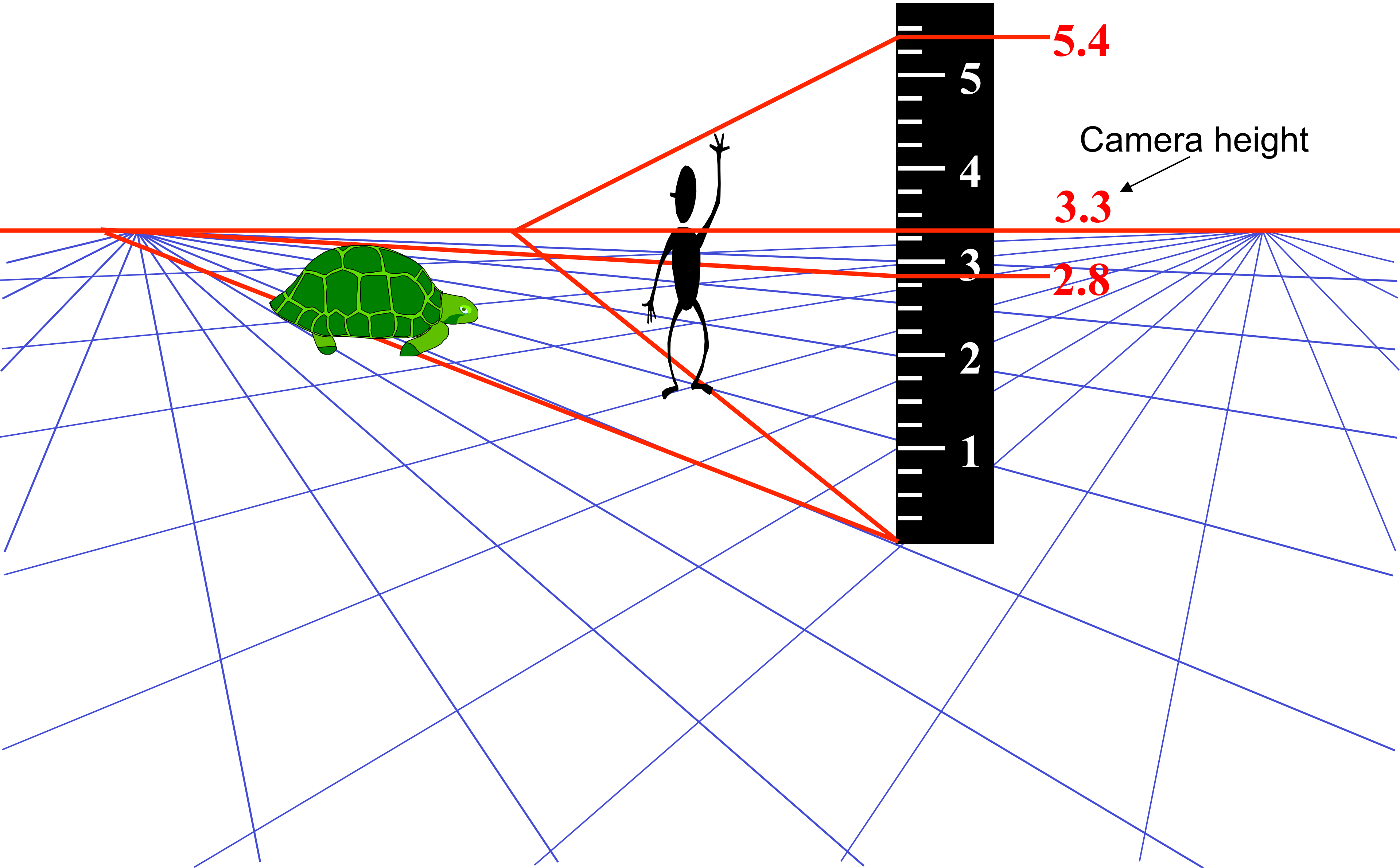
Perspective cues



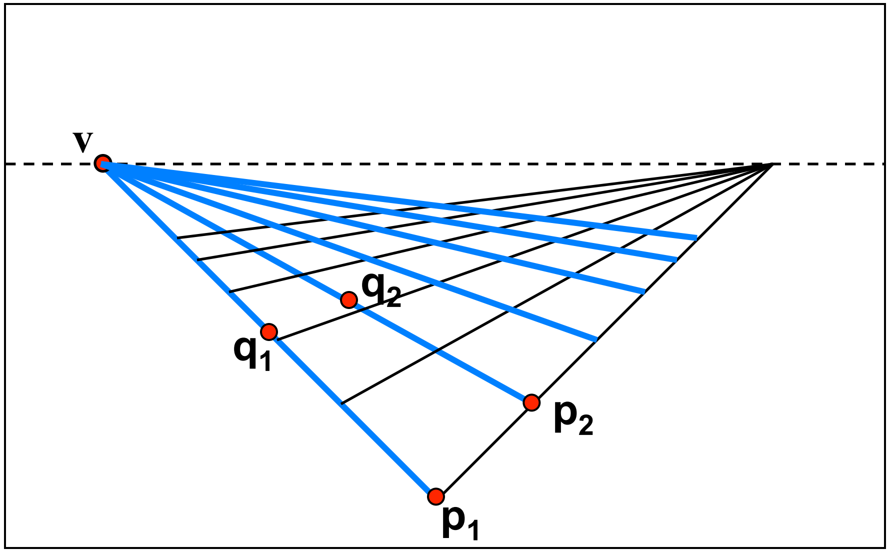
Comparing heights



Measuring height



Computing vanishing points (from lines)



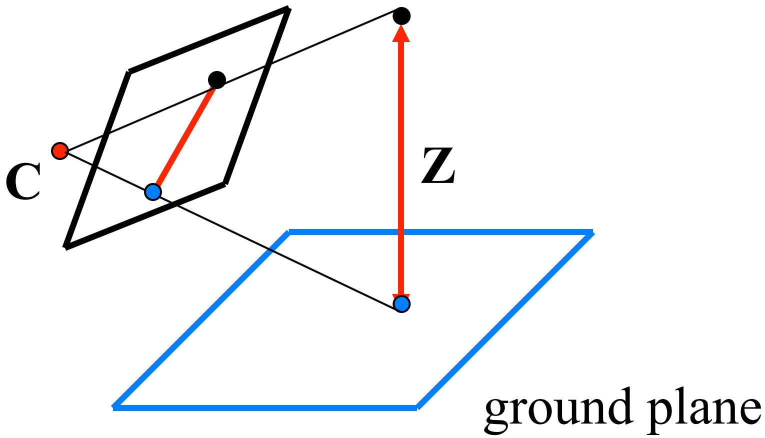
- Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

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



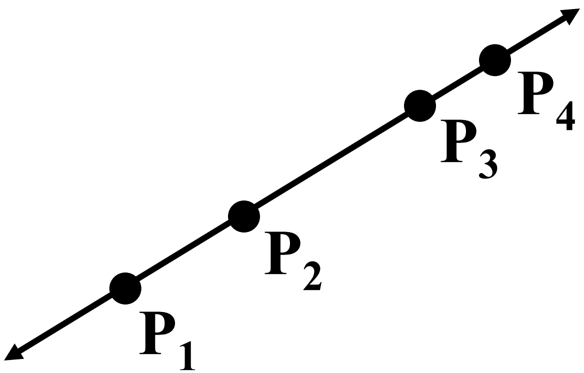
Compute Z from image measurements

- Need more than vanishing points to do this

The cross ratio

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

The cross-ratio of 4 collinear points



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

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

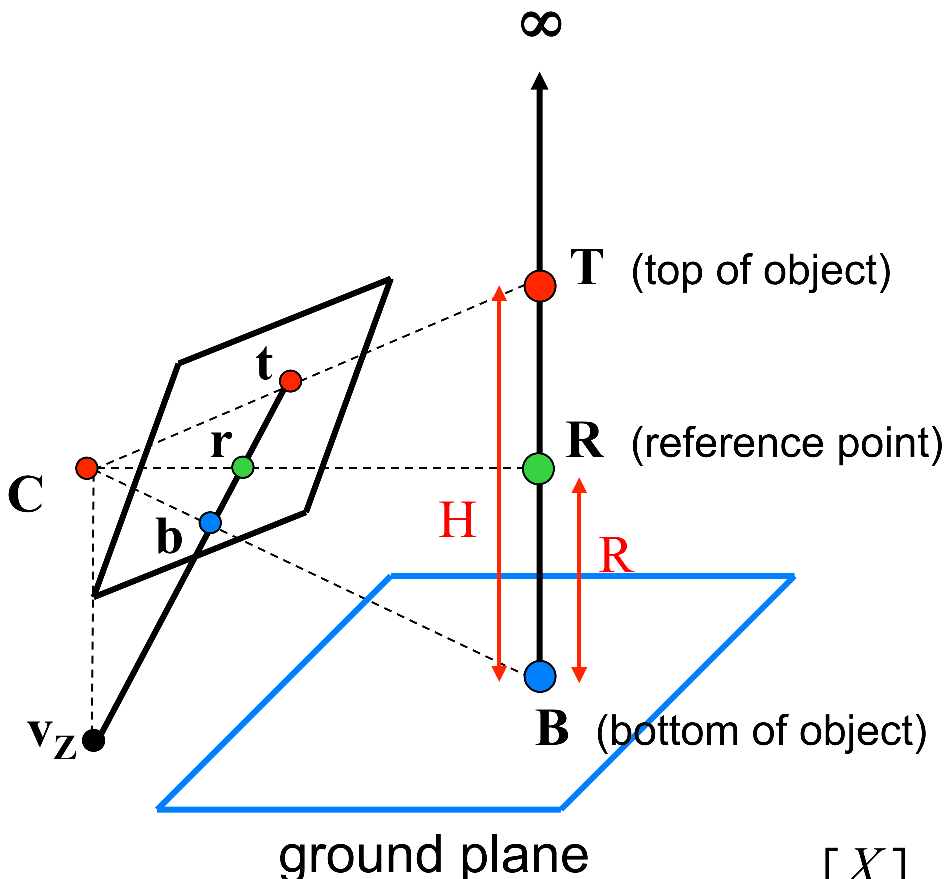
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

Measuring height



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

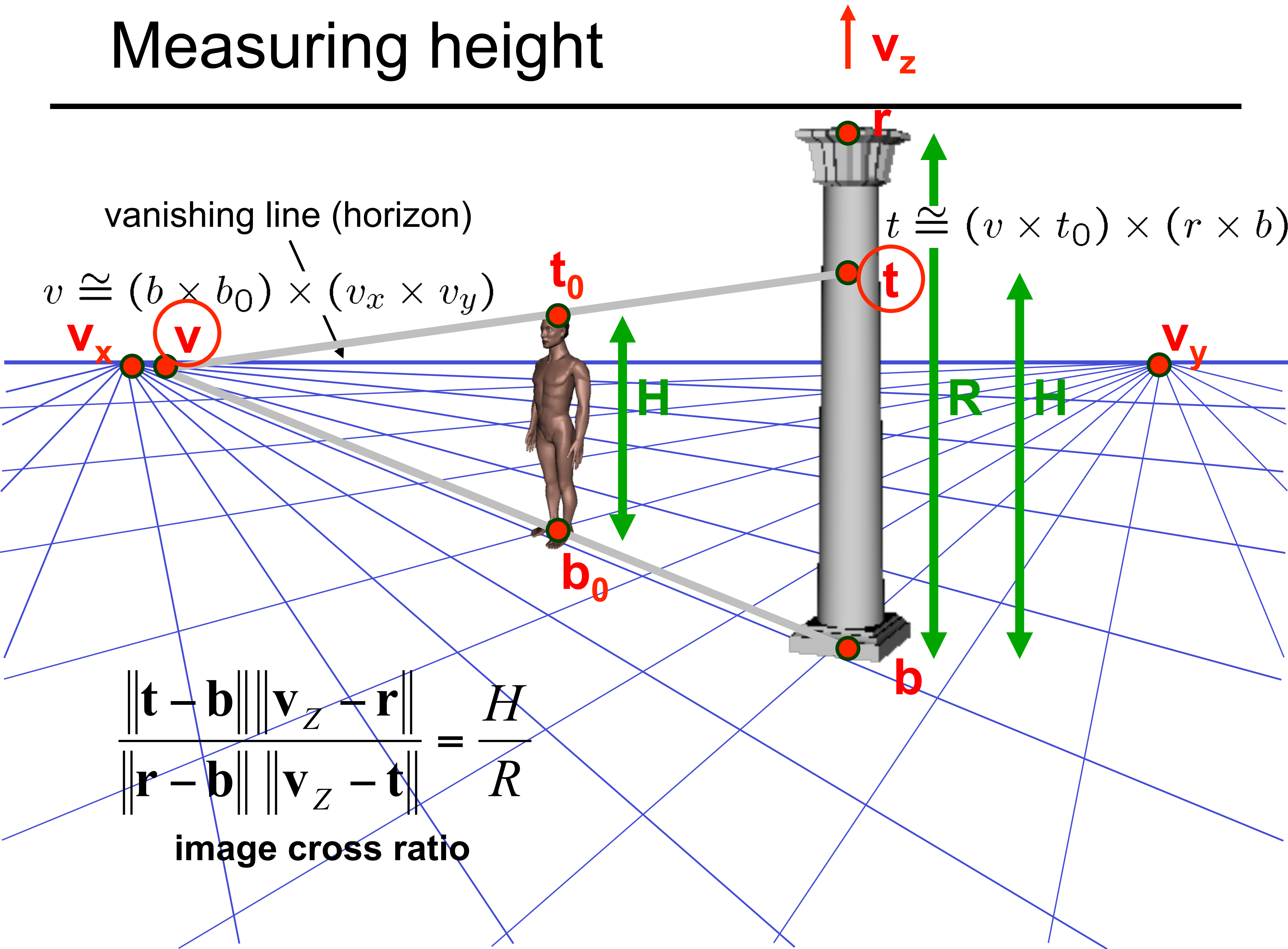
scene cross ratio

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

image cross ratio

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

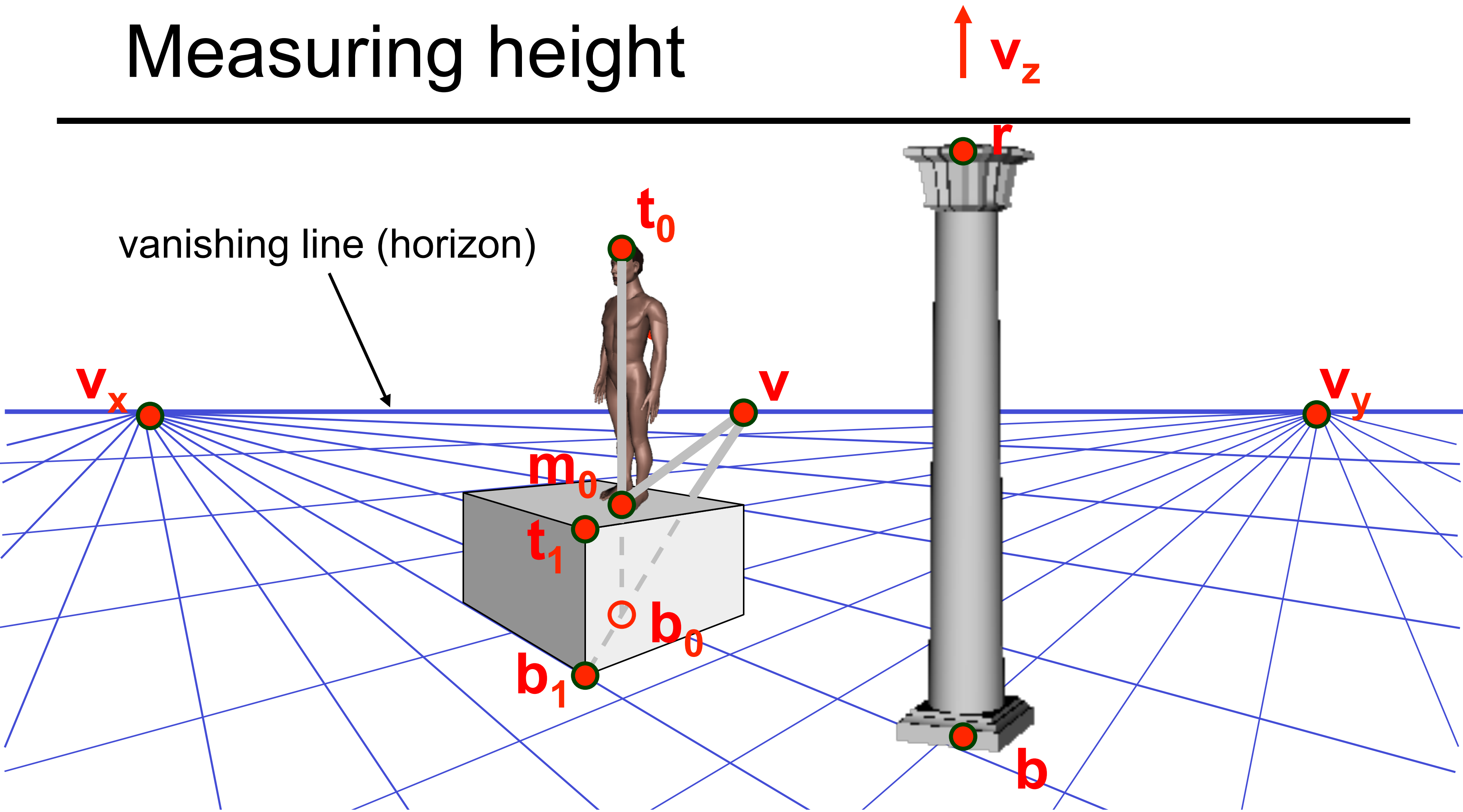
Measuring height



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

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

Computing (X,Y,Z) coordinates

- Okay, we know how to compute height (Z coords)
  - how can we compute X, Y?

Camera calibration

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

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

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

Vanishing points and projection matrix

$$\mathbf{\Pi} = \begin{array}{c} \begin{bmatrix} \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} \end{bmatrix} \\ \boldsymbol{\pi}_1 \quad \boldsymbol{\pi}_2 \quad \boldsymbol{\pi}_3 \quad \boldsymbol{\pi}_4 \end{array} = \begin{bmatrix} \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \boldsymbol{\pi}_3 & \boldsymbol{\pi}_4 \end{bmatrix}$$

- $\boldsymbol{\pi}_1 = \mathbf{\Pi}\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ = $\mathbf{v}_x$ (X vanishing point)
- similarly, $\boldsymbol{\pi}_2 = \mathbf{v}_Y$, $\boldsymbol{\pi}_3 = \mathbf{v}_Z$
- $\boldsymbol{\pi}_4 = \mathbf{\Pi}\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T = \text{projection of world origin}$

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{v}_X & \mathbf{v}_Y & \mathbf{v}_Z & \mathbf{o} \end{bmatrix}$$

Not So Fast! We only know **v**'s up to a scale factor

$$\mathbf{\Pi} = \begin{bmatrix} a\,\mathbf{v}_X & b\mathbf{v}_Y & c\mathbf{v}_Z & \mathbf{o} \end{bmatrix}$$

- Can fully specify by providing 3 reference points