Single-view metrology

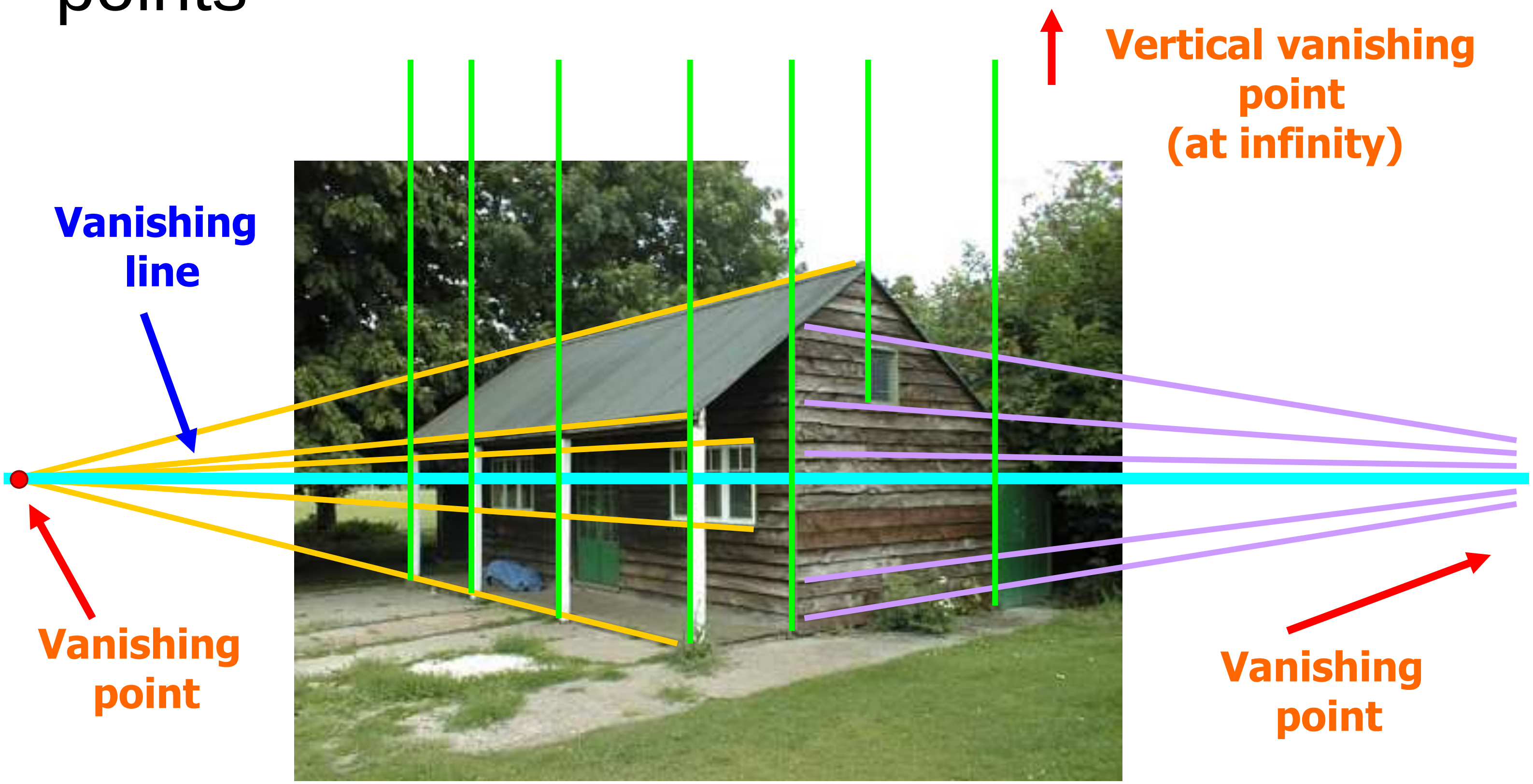
Magritte, *Personal Values*, 1952

Section 6.3.2, R. Szeliski

Many slides from S. Seitz, D. Hoiem

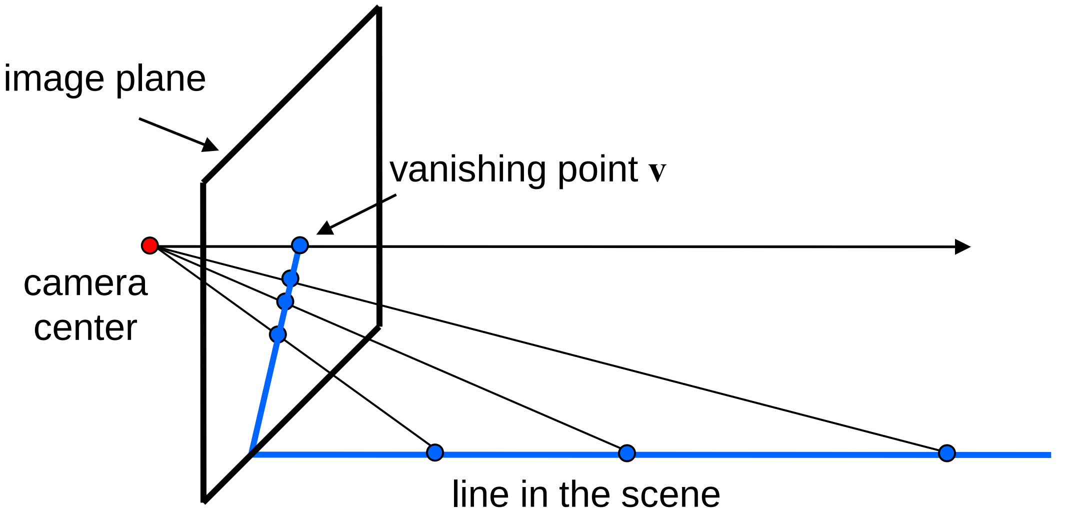
Camera calibration revisited

- What if world coordinates of reference 3D points are not known?
- We can use scene features such as vanishing points



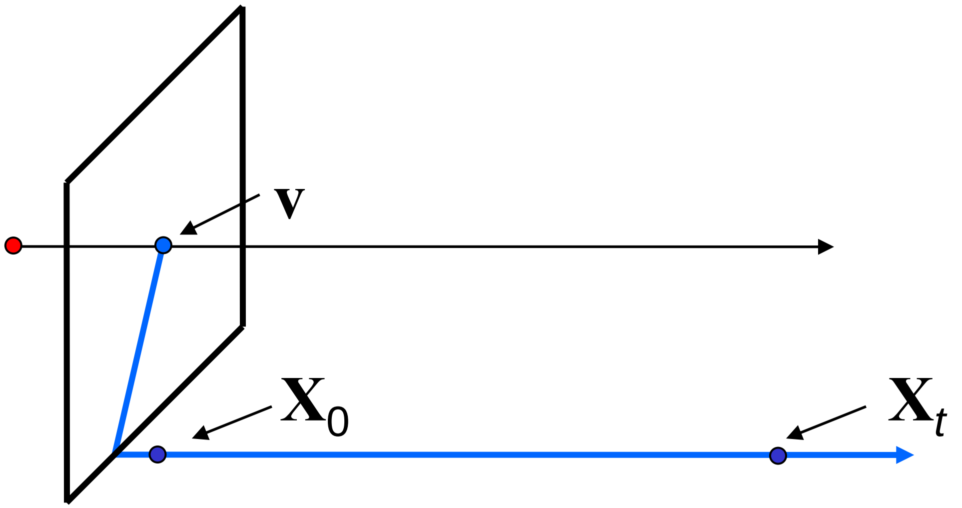
Slide from Efros, Photo from Criminisi

Recall: Vanishing points



- All lines having the same direction share the same vanishing point

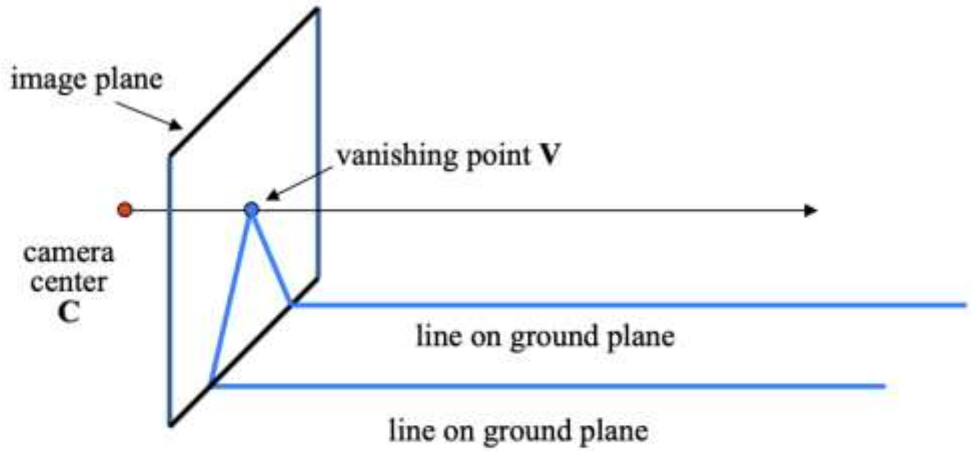
Computing vanishing points



$$\mathbf{X}_t = \begin{bmatrix} x_0 + td_1 \\ y_0 + td_2 \\ z_0 + td_3 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/t + d_1 \\ y_0/t + d_2 \\ z_0/t + d_3 \\ 1/t \end{bmatrix} \qquad \mathbf{X}_\infty = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ 0 \end{bmatrix}$$

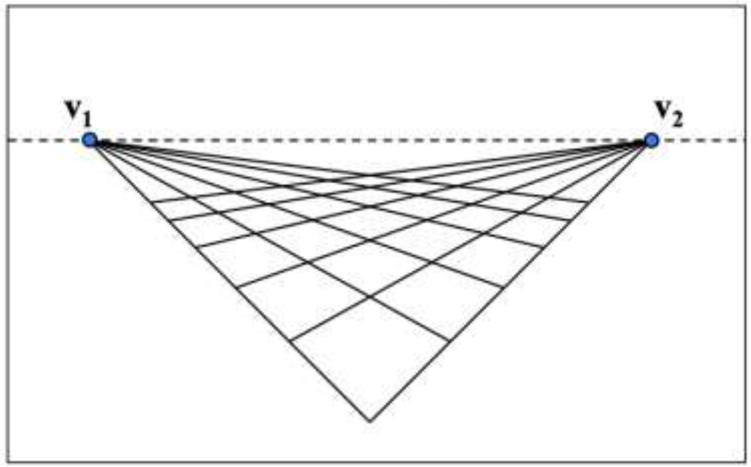
- $\mathbf{X}_\infty$ is a *point at infinity,* $\mathbf{v}$ is its projection: $\mathbf{v} = \mathbf{P}\mathbf{X}_\infty$
- The vanishing point depends only on *line direction*
- All lines having direction $\mathbf{d}$ intersect at $\mathbf{X}_\infty$

Vanishing points



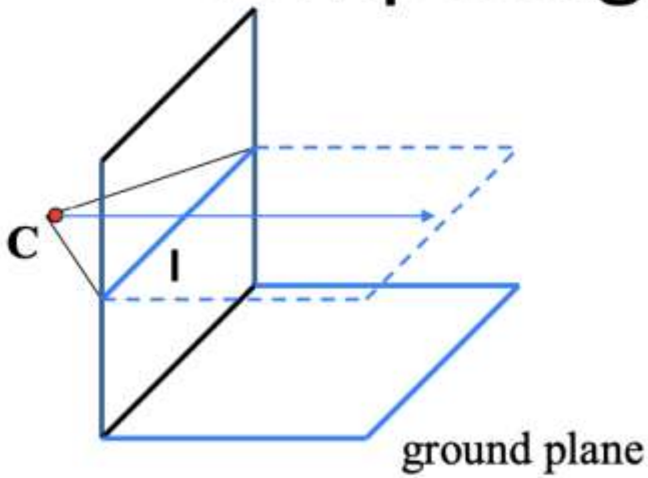
- Properties
  - Any two parallel lines (in 3D) have the same vanishing point **v**
  - The ray from **C** through **v** is parallel to the lines
  - An image may have more than one vanishing point
    - in fact, every image point is a potential vanishing point

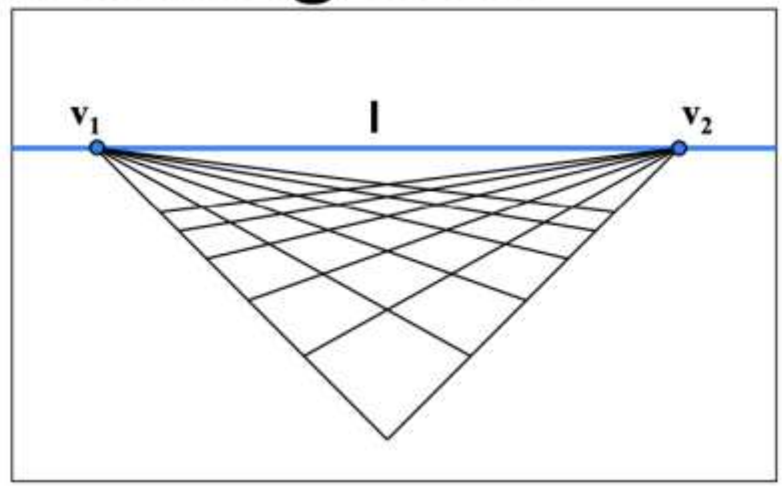
Vanishing lines



- 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

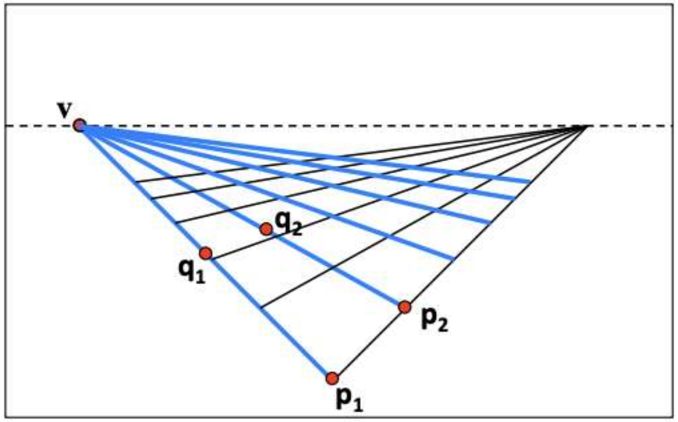
Computing vanishing lines





- Properties
  - **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

Computing vanishing points (from lines)



- Intersect p_1q_1 with p_2q_2

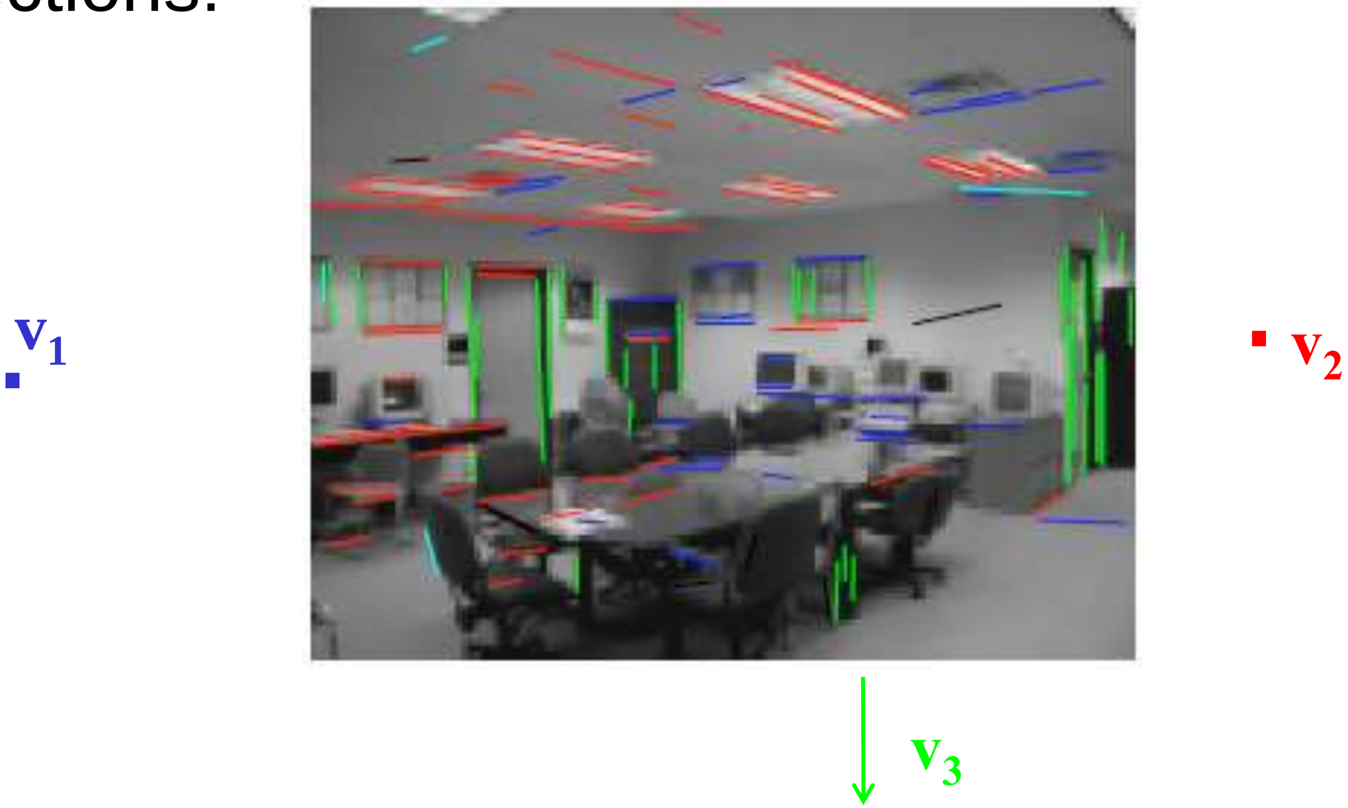
$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

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

Calibration from vanishing points

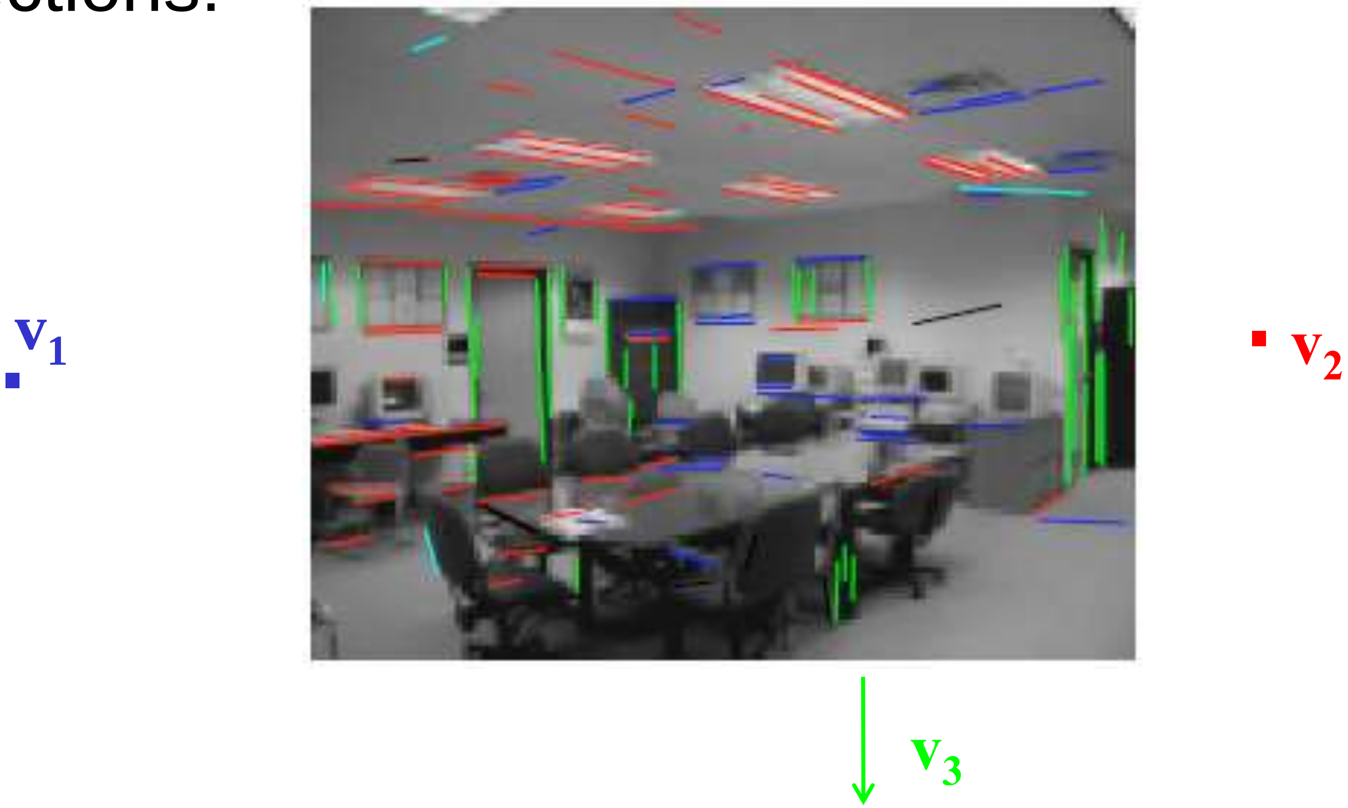
- Consider a scene with three orthogonal vanishing directions:



- Note: $\mathbf{v}_1$, $\mathbf{v}_2$ are *finite* vanishing points and $\mathbf{v}_3$ is an *infinite* vanishing point

Calibration from vanishing points

- Consider a scene with three orthogonal vanishing directions:



- We can align the world coordinate system with these directions

Calibration from vanishing points

$$\mathbf{P}X = \begin{bmatrix} \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix}$$

- $\mathbf{p}_1 = \mathbf{P}(1,0,0,0)^T$ – the vanishing point in the x direction
- Similarly, $\mathbf{p}_2$ and $\mathbf{p}_3$ are the vanishing points in the y and z directions
- $\mathbf{p}_4 = \mathbf{P}(0,0,0,1)^T$ – projection of the origin of the world coordinate system
- Problem: we can only know the four columns up to independent scale factors, additional constraints needed to solve for them

Calibration from vanishing points

- Let us align the world coordinate system with three orthogonal vanishing directions in the scene:

$$\mathbf{e_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e_2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e_3} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \lambda_i \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} = \mathbf{KRe}_i$$

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

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

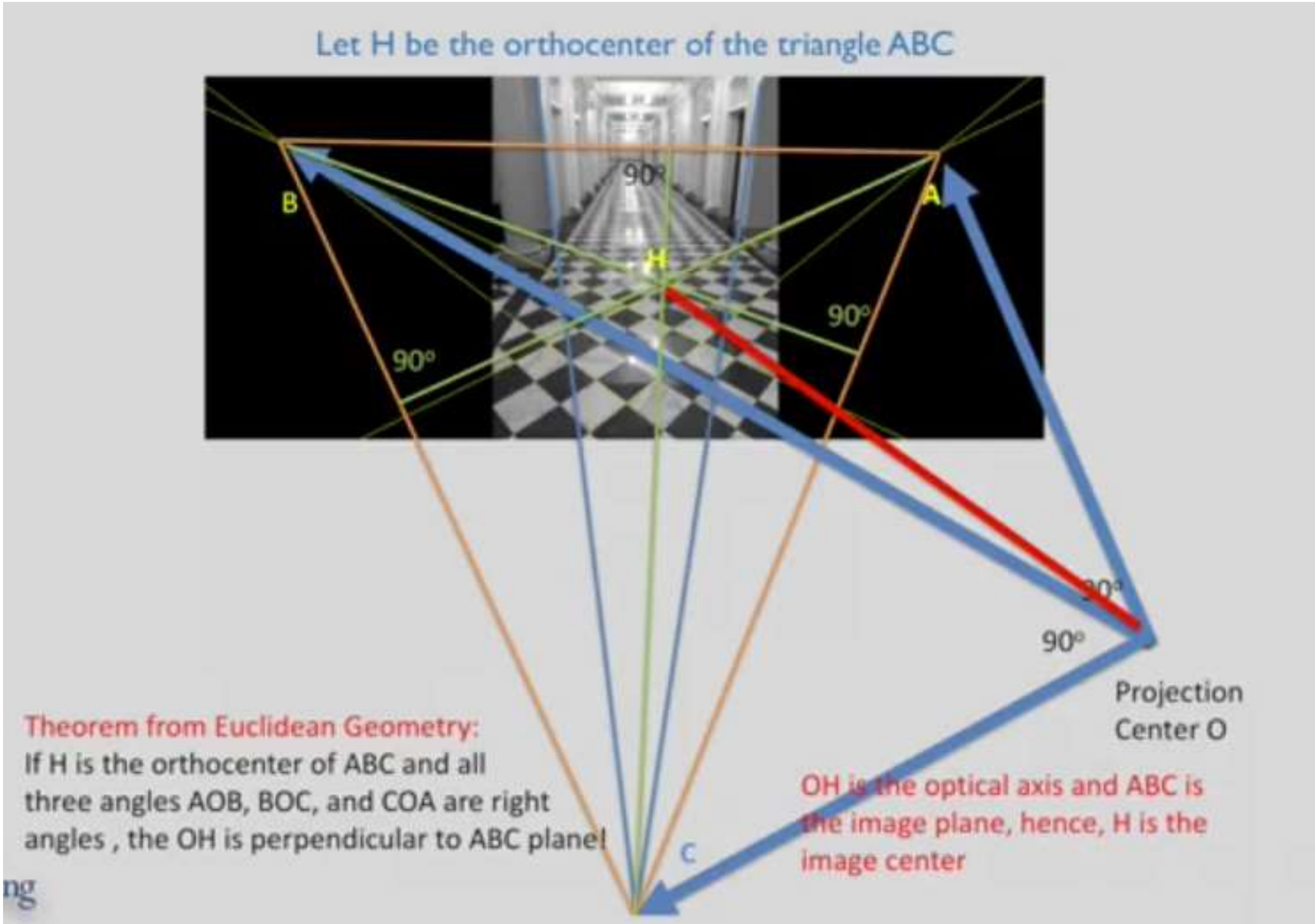
- Each pair of vanishing points gives us a constraint on the focal length and principal point

- zero skew, unit aspect ratio

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

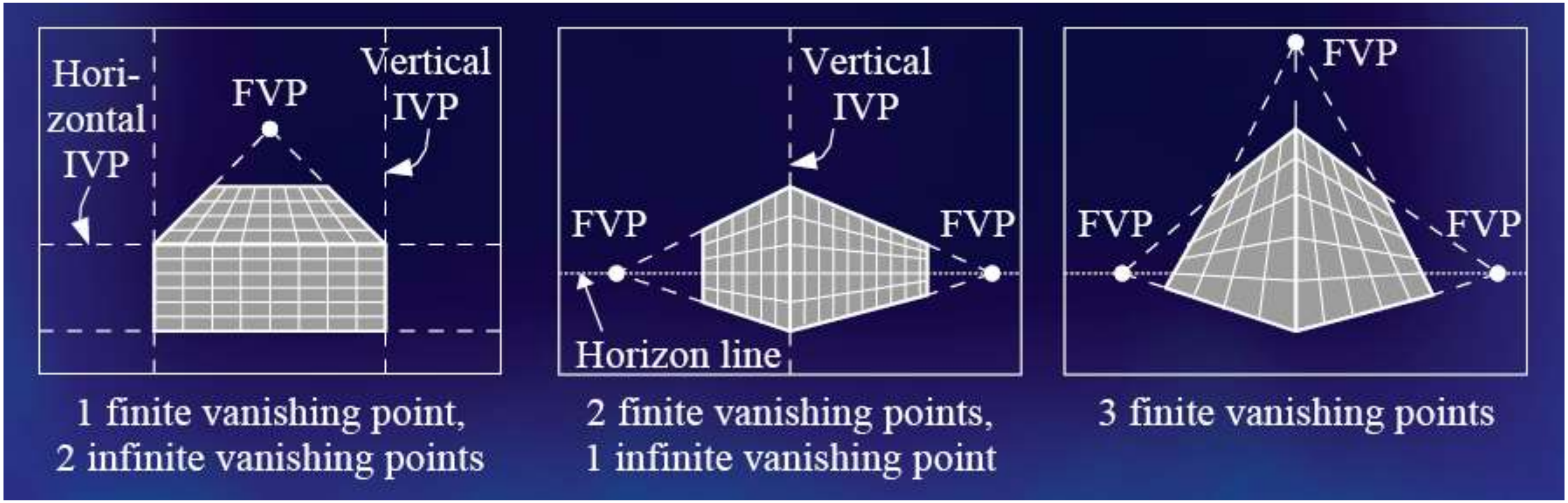
$$v_i^T K^{-T} K^{-1} v_j = 0$$
$$v_j^T K^{-T} K^{-1} v_k = 0$$
$$v_i^T K^{-T} K^{-1} v_k = 0$$

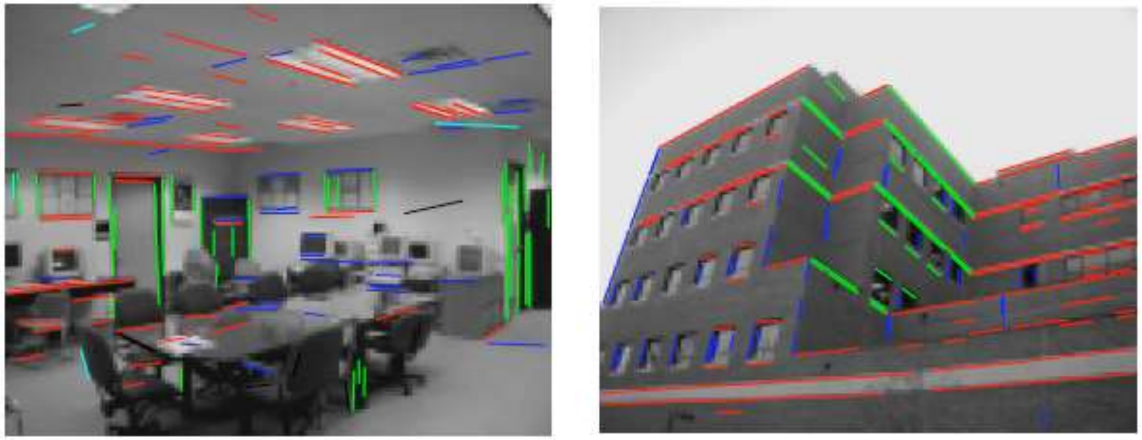
- 3 finite vanishing points: get f, u0, v0
- 2 finite and one infinite : u0,v0 as point on vf1 vf2 closest to image center, get f
- 2 infinite vanishing points : f cant be recovered u0, v0 is at the third vanishing point



https://www.coursera.org/learn/robotics-perception/lecture/jnaLs/how-to-compute-intrinsics-from-vanishing-points

Calibration from vanishing points





Cannot recover focal length, principal point is the third vanishing point

Can solve for focal length, principal point

Rotation from vanishing points

$$\lambda_i \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} = \mathbf{K}\mathbf{R}\mathbf{e}_i$$

$$\lambda_1 \mathbf{K}^{-1}\mathbf{v}_1 = \mathbf{R}\mathbf{e}_1 = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{r}_1$$

Thus, $\lambda_i \mathbf{K}^{-1}\mathbf{v}_i = \mathbf{r}_i$.

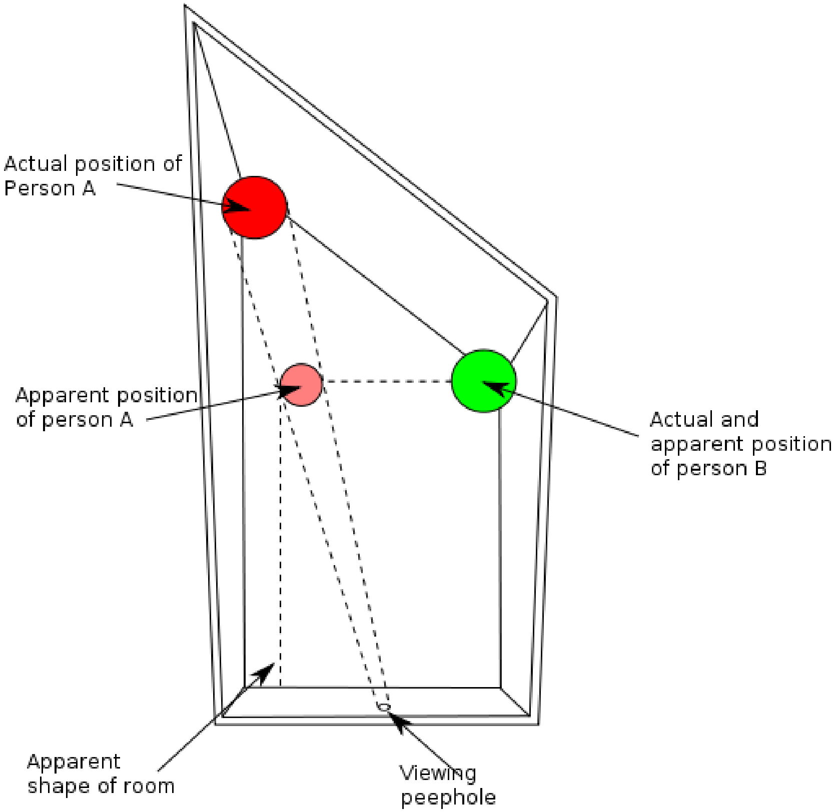
Get λ_i by using the constraint $\|\mathbf{r}_i\|^2=1$.

Calibration from vanishing points: Summary

- Solve for K (focal length, principal point) using three orthogonal vanishing points
- Get rotation directly from vanishing points once calibration matrix is known

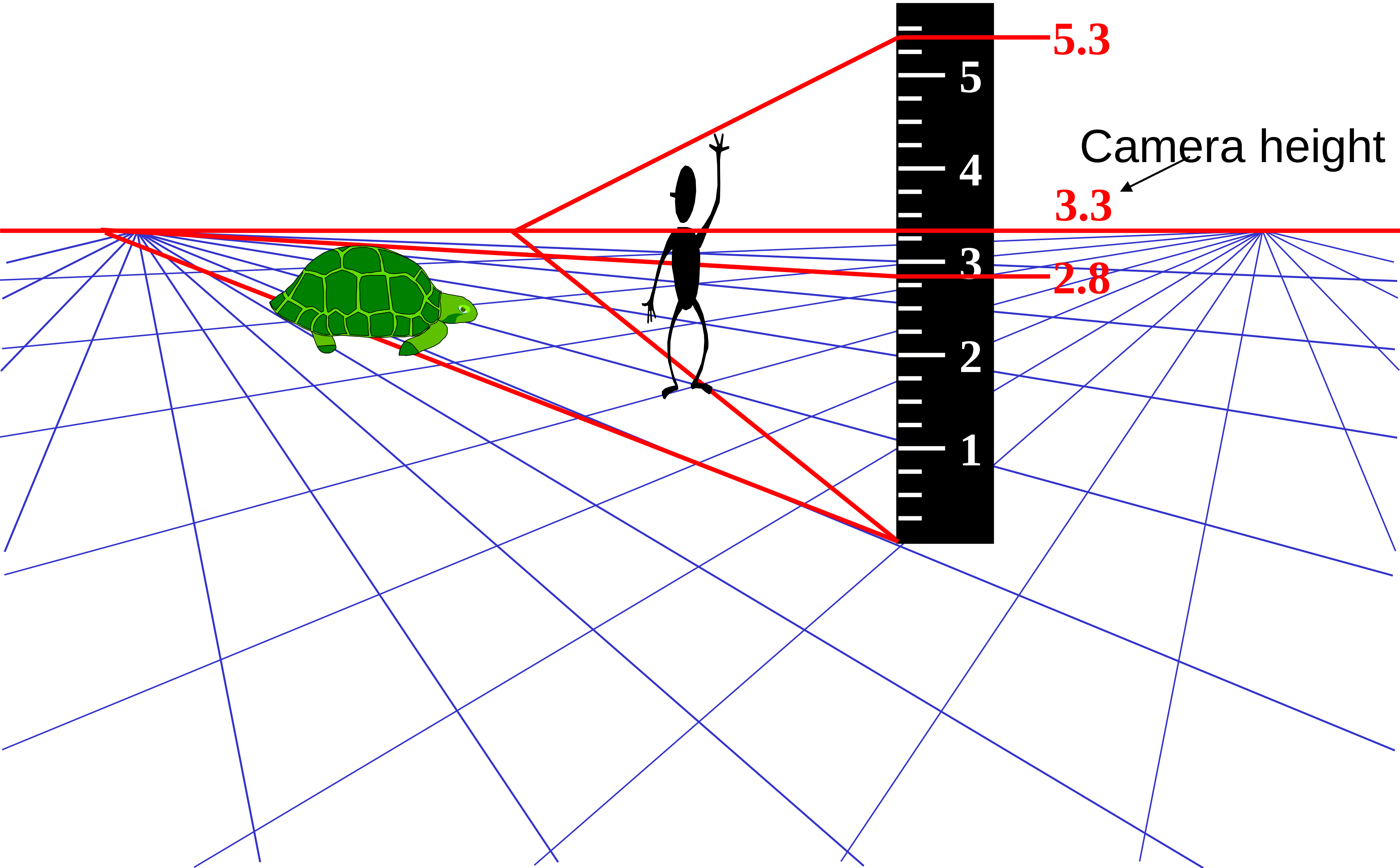
- Advantages
  - No need for calibration chart, 2D-3D correspondences
  - Could be completely automatic
- Disadvantages
  - Only applies to certain kinds of scenes
  - Inaccuracies in computation of vanishing points
  - Problems due to infinite vanishing points

Making measurements from a single image

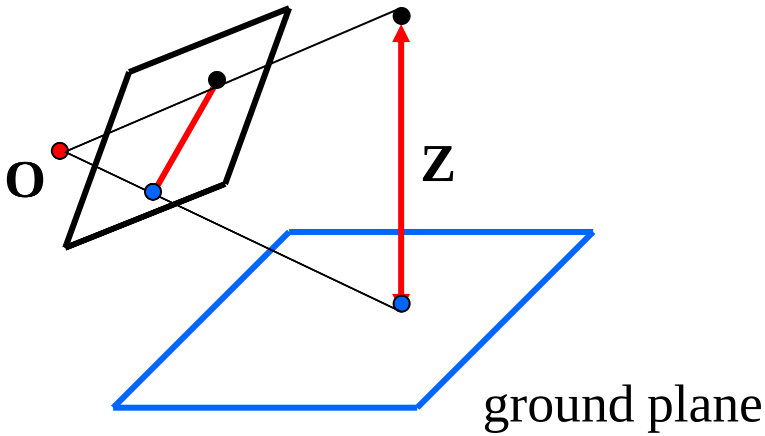


http://en.wikipedia.org/wiki/Ames_room

Measuring height



Measuring height without a ruler



Compute Z from image measurements

- Need more than vanishing points to do this

Projective invariant

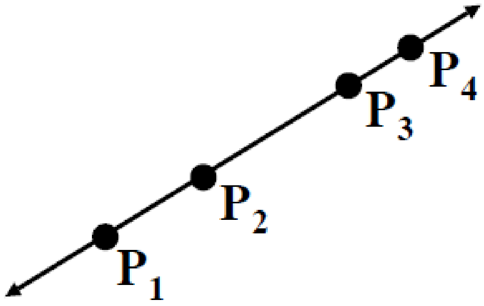
- We need to use a *projective invariant*: a quantity that does not change under projective transformations (including perspective projection)
  - What are some invariants for similarity, affine transformations?

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

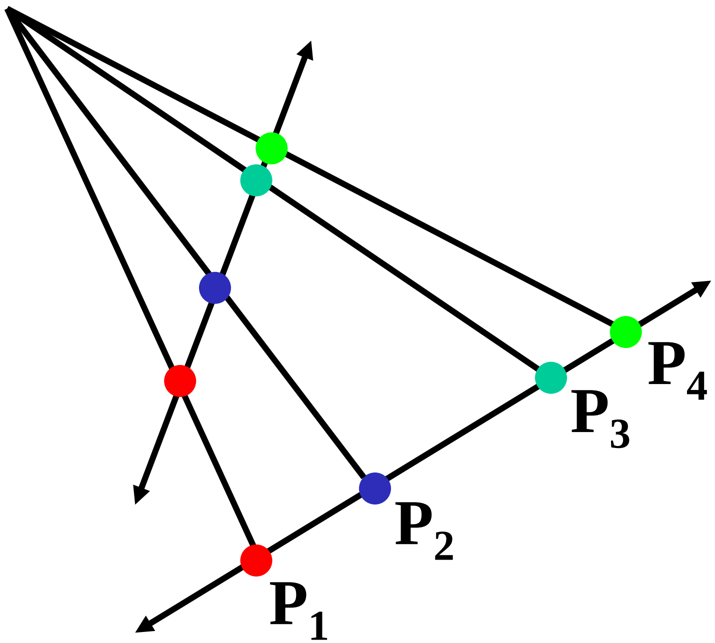
$$\frac{\|\mathbf{P}_1 - \mathbf{P}_3\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_3\|}$$

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

This is the fundamental invariant of projective geometry

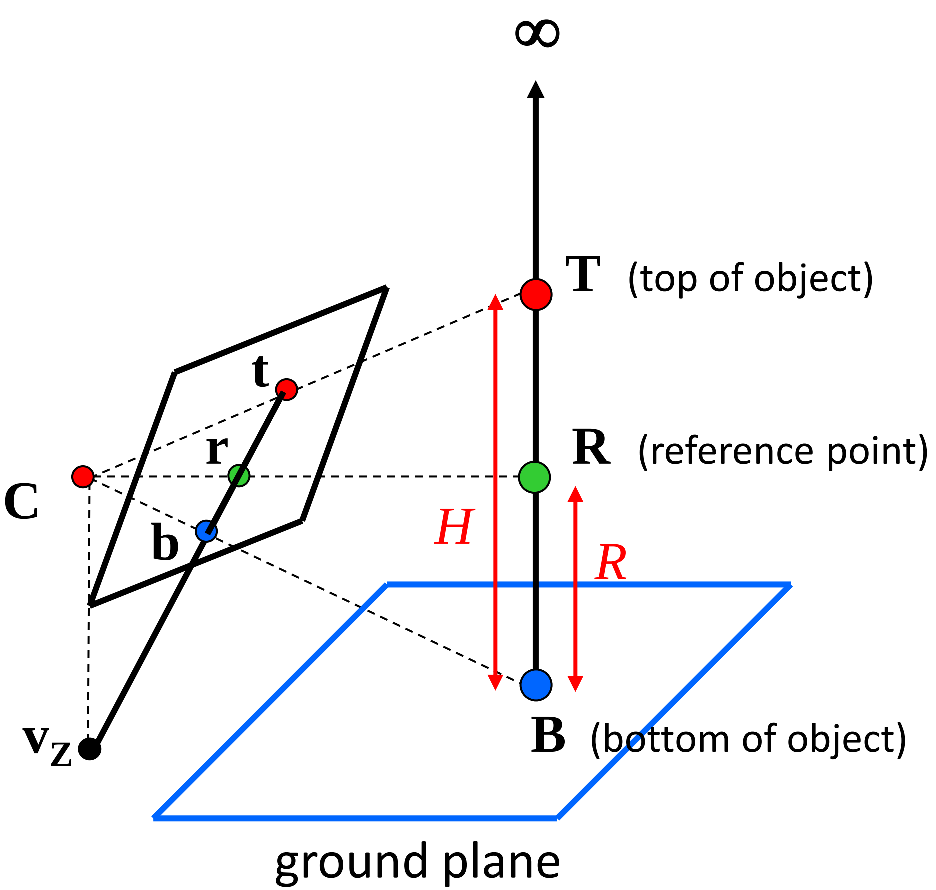
Projective invariant

- We need to use a *projective invariant*: a quantity that does not change under projective transformations (including perspective projection)
- The *cross-ratio* of four points:



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

Measuring height



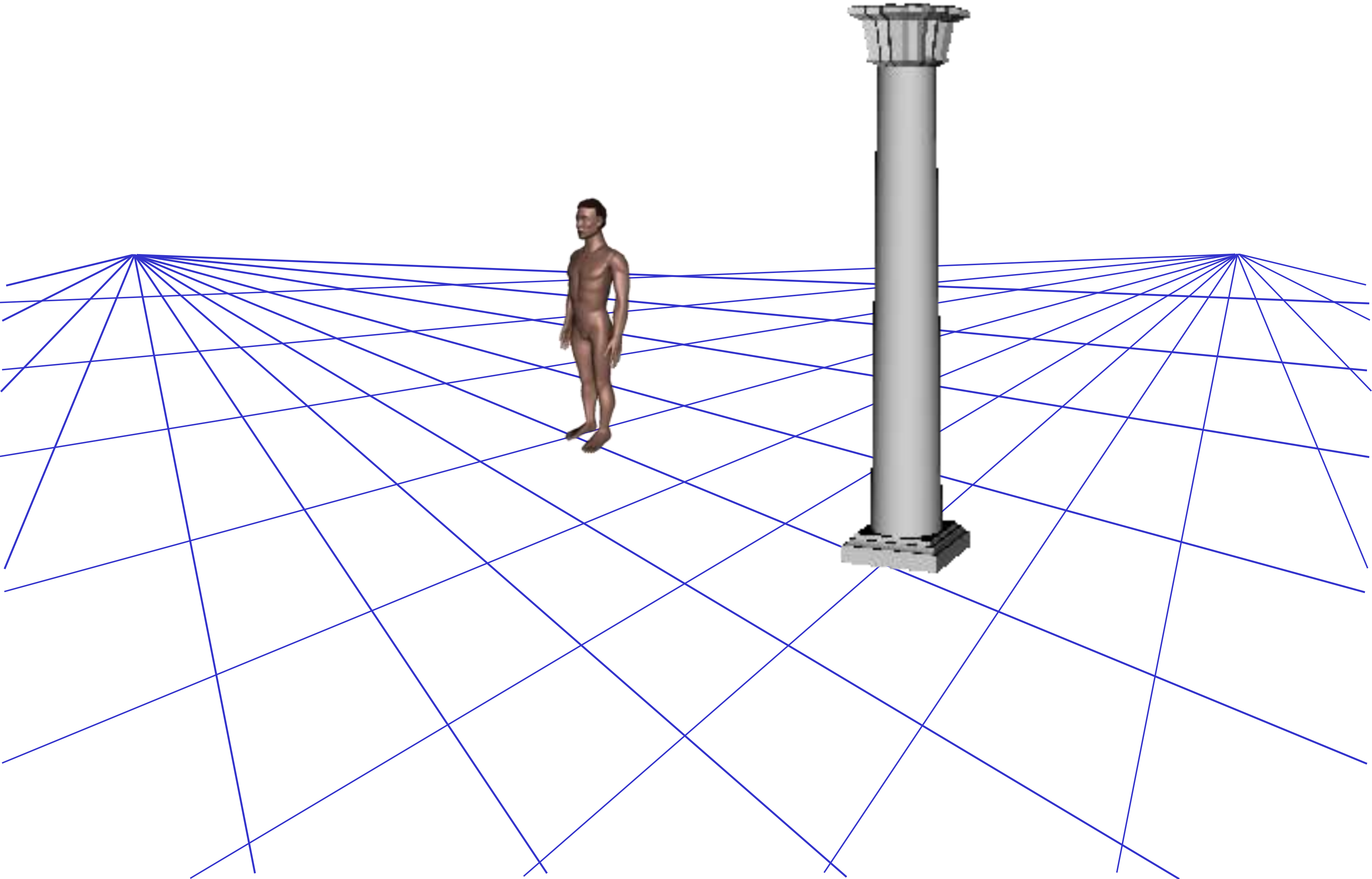
$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

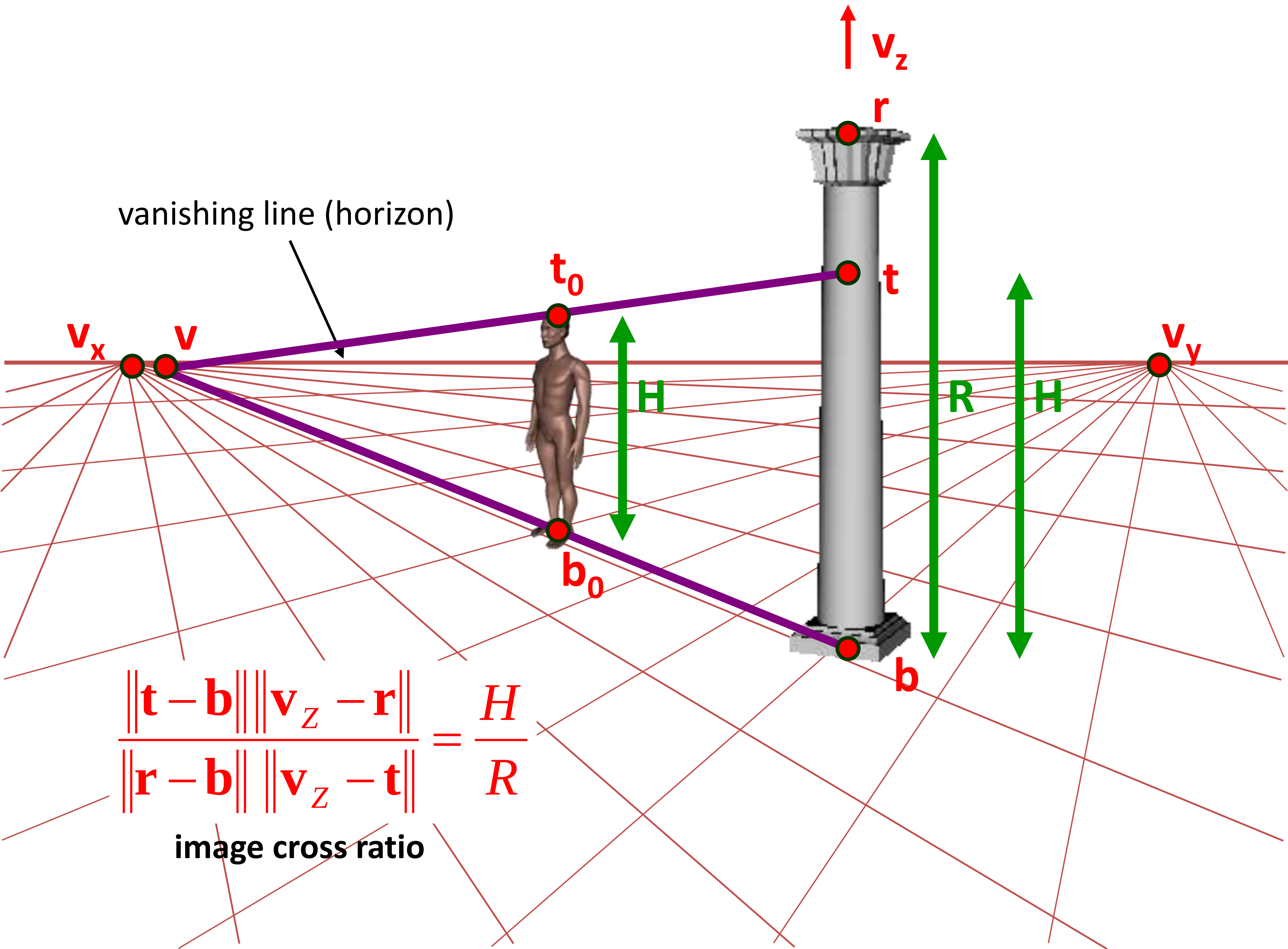
scene cross ratio

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

image cross ratio

Measuring height without a ruler





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

image cross ratio

2D lines in homogeneous coordinates

- Line equation: $ax + by + c = 0$

$$\mathbf{l}^T\mathbf{x} = 0 \quad \text{where} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Line passing through two points: $\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$

- Intersection of two lines: $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$

已知：二维平面的两点X（x_1，y_1），Y（x_2，y_2)，证明X，Y两点的齐次式叉乘为过XY的直线的系数.

证明： 叉乘的定义为已知向量$a = (a_1,a_2,a_3)$, $b=(b_1,b_2,b_3)$, a叉乘$b=(a_2b_3-a_3b_2, a_3b_1-a_1b_3, a_1b_2-a_2b_1)$

因为XY的齐次式为（$x_1,y_1,1$)和($x_2,y_2,1$)，代入叉乘的定义得（$y_1-y_2, x_2-x_1, x_1y_2-y_1x_2$)

定义直线的表达式为$y=kx + b$,将XY代入得：

$y_1 = kx_1 + b$

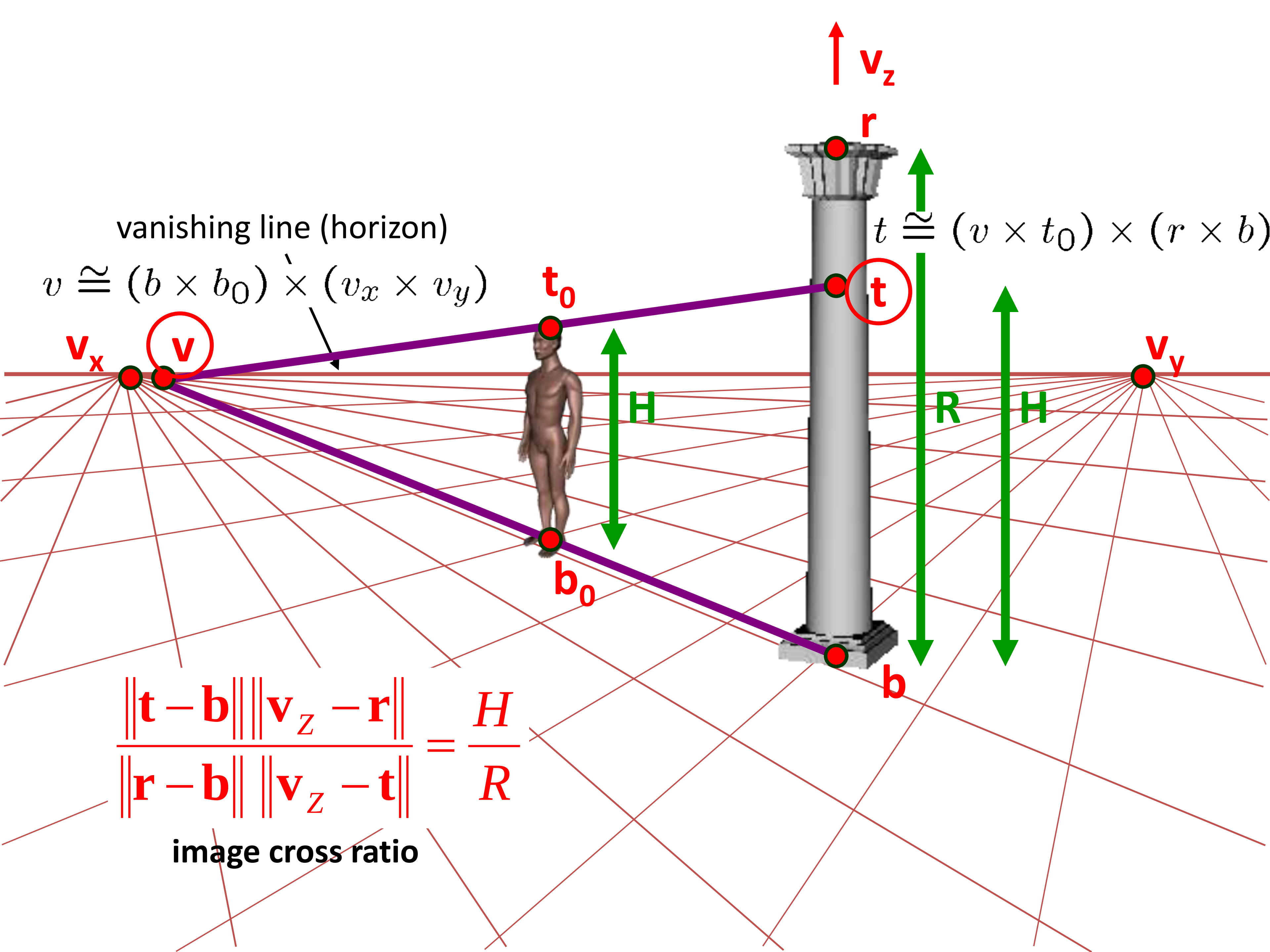
$y_2 = kx_2 + b$

化简后得：

$k = (y_2-y_1)/(x_2-x_1)$

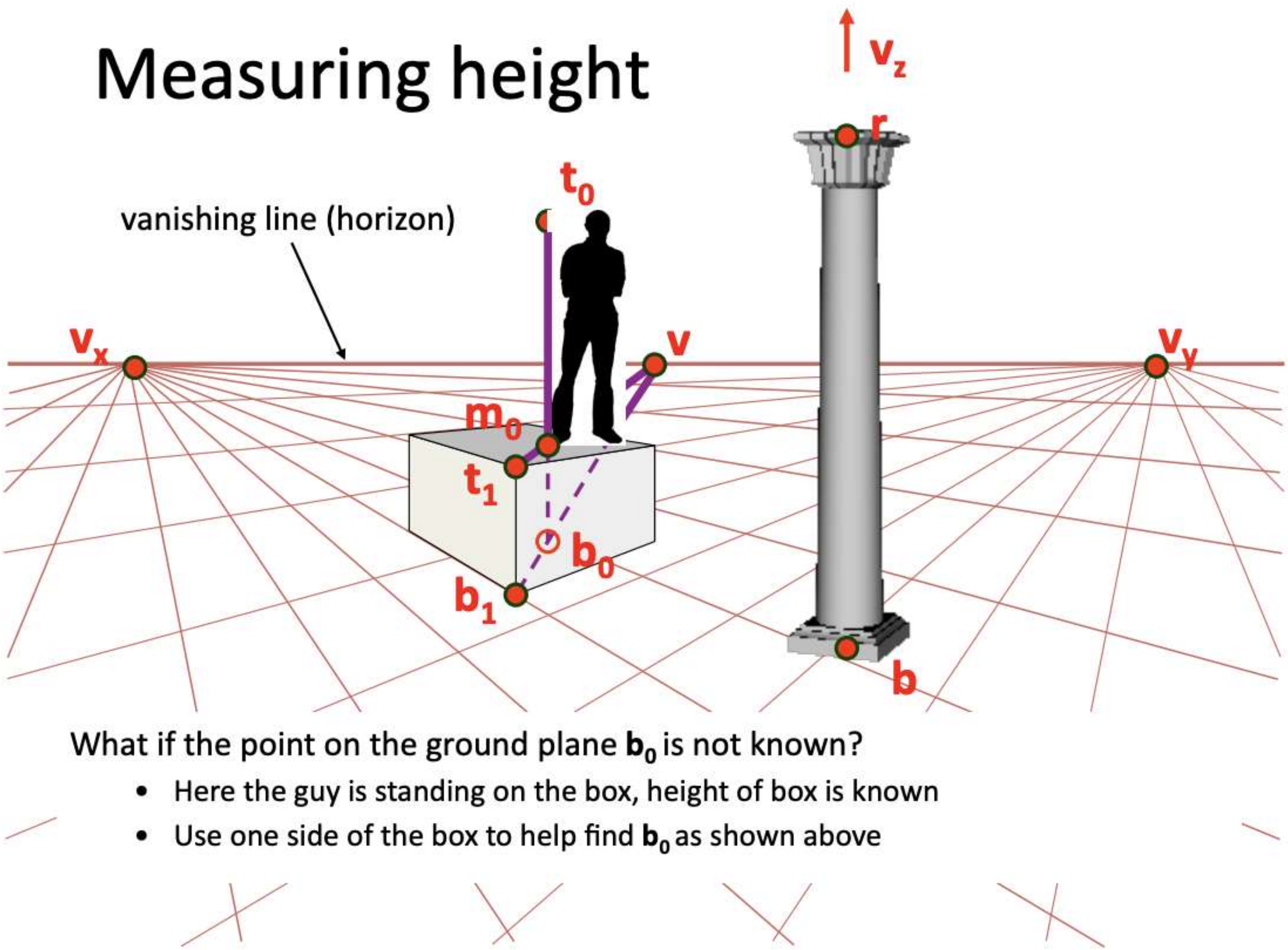
$b = y_1 - ((y_2-y_1)/(x_2-x_1)) * x_1$

将$y = kx + b$ 转化为 $ax + by + c = 0$的形式得 （a b c） = （-k， 1， -b） 化简后等于（$y_1-y_2, x_2-x_1, x_1y_2-y_1x_2$)



v_z
r
vanishing line (horizon)
$v \cong (b \times b_0) \times (v_x \times v_y)$
$t \cong (v \times t_0) \times (r \times b)$
t_0
t
v_x
v
v_y
H
R
H
b_0
b
$\frac{\|\mathbf{t}-\mathbf{b}\| \|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\| \|\mathbf{v}_Z-\mathbf{t}\|} = \frac{H}{R}$
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