# Projective Geometry

Lecture slides by Steve Seitz (mostly)
Lecture presented by Rick Szeliski

# Final project ideas

Discussion by Steve Seitz and Rick Szeliski

…

# Projective geometry

Ames Room

## Readings

- Mundy, J.L. and Zisserman, A., Geometric Invariance in Computer Vision, Appendix: Projective Geometry for Machine Vision, MIT Press, Cambridge, MA, 1992, **(read 23.1 - 23.5, 23.10)**
  - available online: http://www.cs.cmu.edu/~ph/869/papers/zisser-mundy.pdf

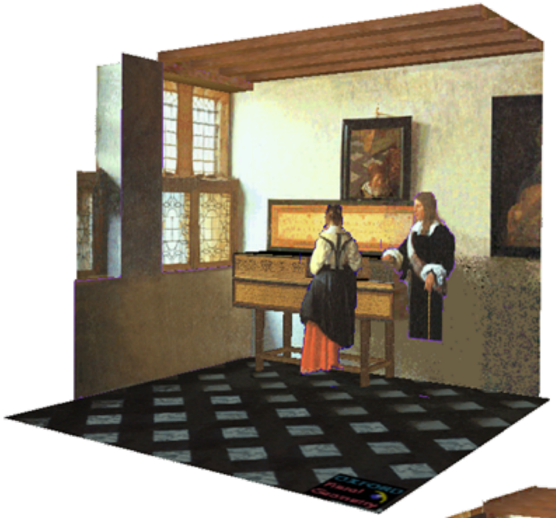
# Projective geometry—what's it good for?

## Uses of projective geometry

- Drawing
- Measurements
- Mathematics for projection
- Undistorting images
- Focus of expansion
- Camera pose estimation, match move
- Object recognition

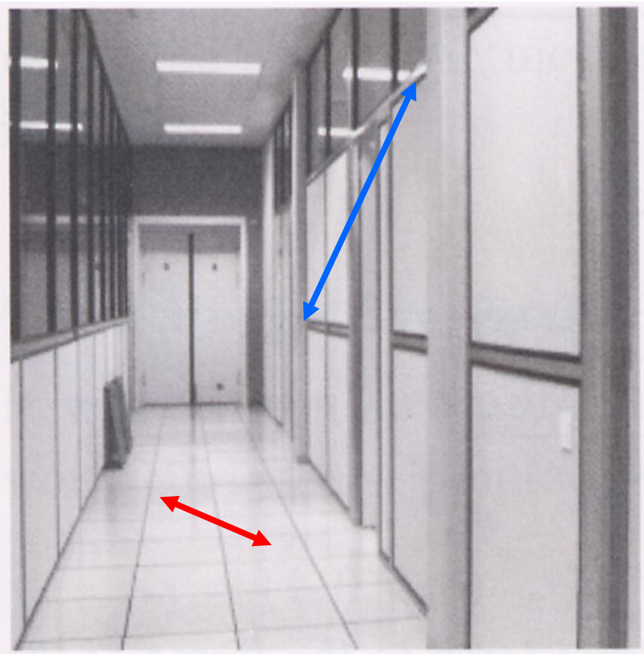
# Applications of projective geometry

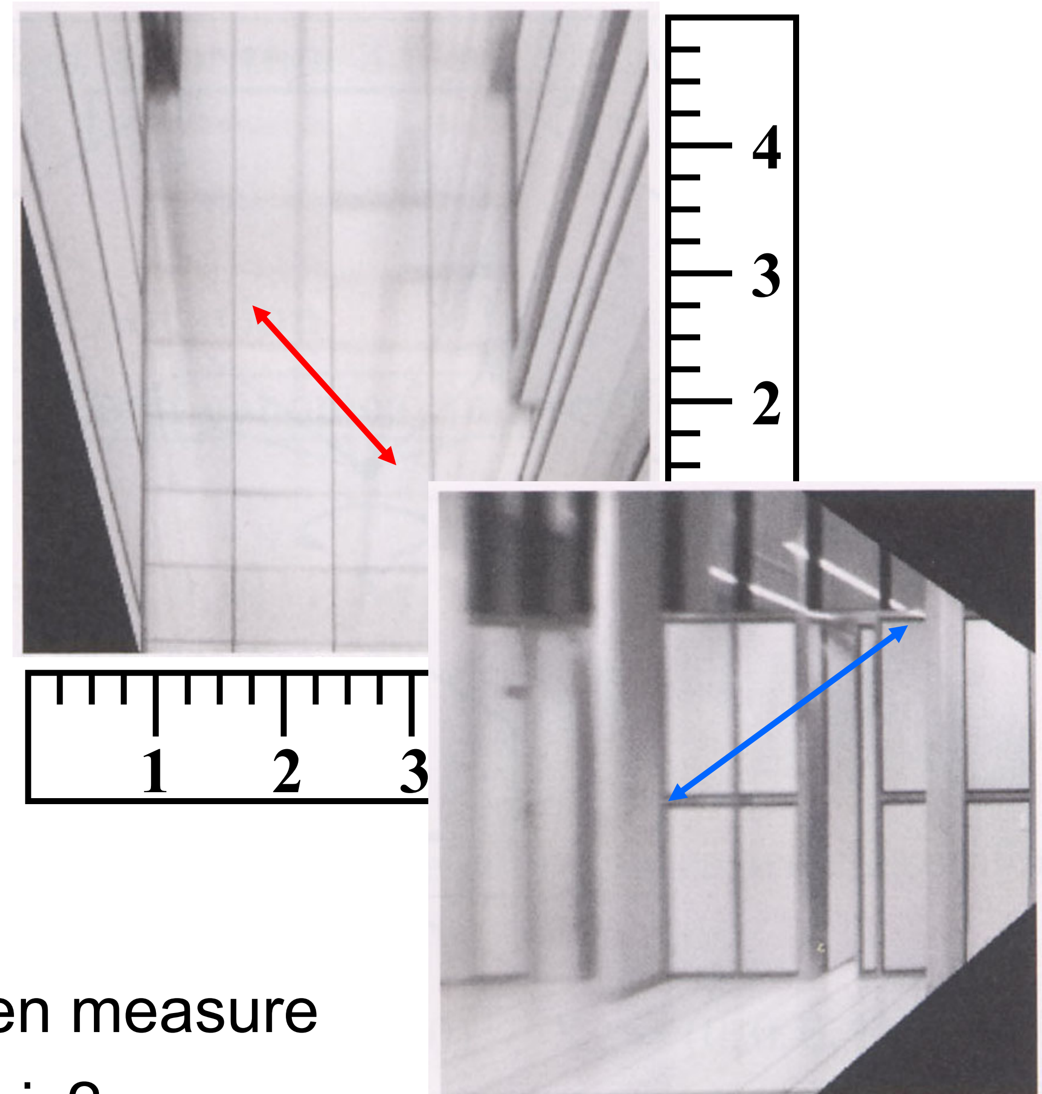
Vermeer's *Music Lesson*



Reconstructions by Criminisi et al.

# Measurements on planes

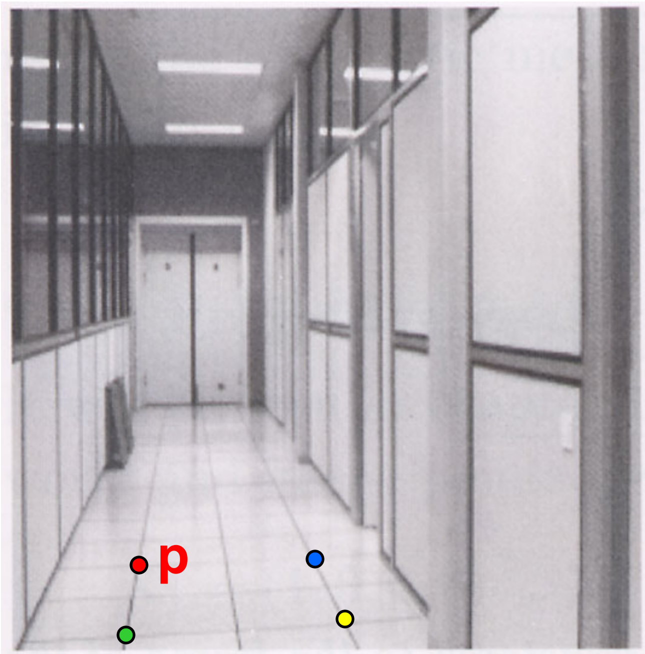


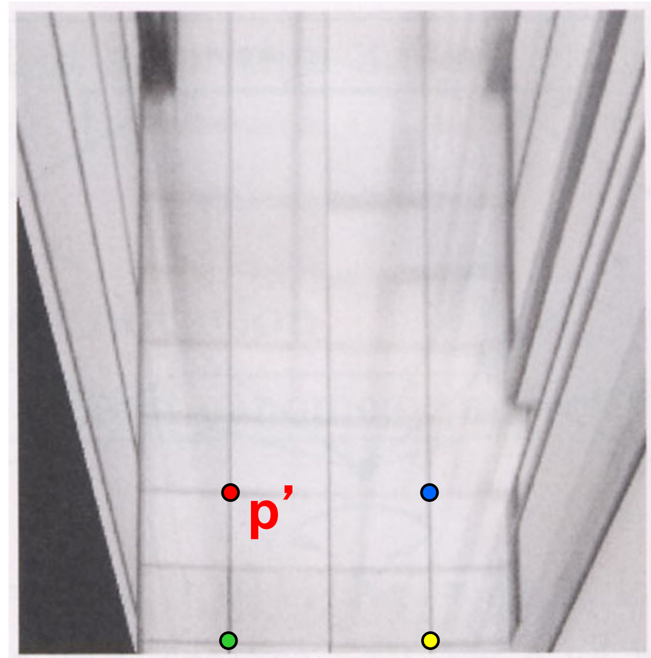



Approach:  unwarp then measure

What kind of warp is this?

# Image rectification




To unwarp (rectify) an image

- solve for homography **H** given **p** and **p'**
- solve equations of the form: w**p'** = **Hp**
  - linear in unknowns: w and coefficients of **H**
  - H is defined up to an arbitrary scale factor
  - how many points are necessary to solve for **H**?

work out on board

# Solving for homographies

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\begin{aligned} x_i' &= \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}} \\ y_i' &= \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}} \end{aligned}$$

$$\begin{aligned} x_i'(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{00}x_i + h_{01}y_i + h_{02} \\ y_i'(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{10}x_i + h_{11}y_i + h_{12} \end{aligned}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

# Solving for homographies

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\underset{2n \times 9}{\mathbf{A}} \qquad \underset{9}{\mathbf{h}} \qquad \underset{2n}{\mathbf{0}}$$

Defines a least squares problem: $\text{minimize } \|\mathbf{Ah} - \mathbf{0}\|^2$

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

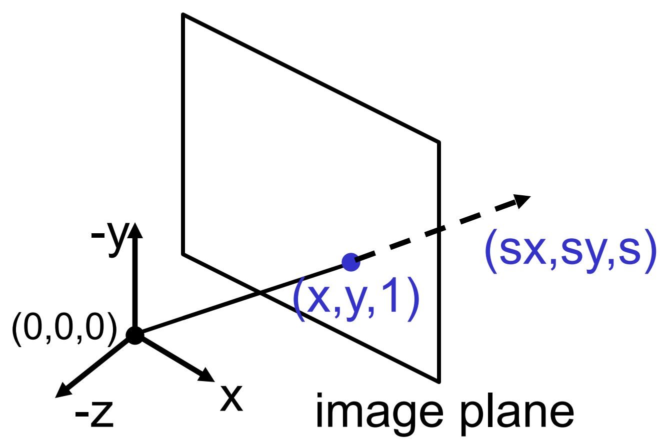
# The projective plane

Why do we need homogeneous coordinates?

- represent points at infinity, homographies, perspective projection, multi-view relationships

What is the geometric intuition?

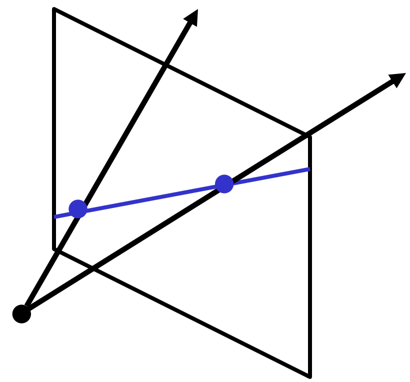
- a point in the image is a *ray* in projective space




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

# Projective lines

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



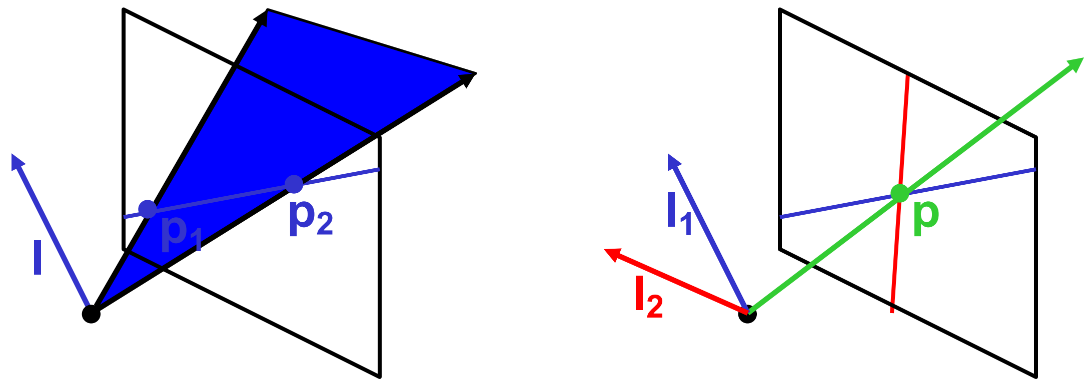
- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

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

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

# Point and line duality

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




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

- **l** is ⊥ to $\mathbf{p}_1$ and $\mathbf{p}_2$ $\Rightarrow$ $\mathbf{l} = \mathbf{p}_1 \times \mathbf{p}_2$
- **l** is the plane normal

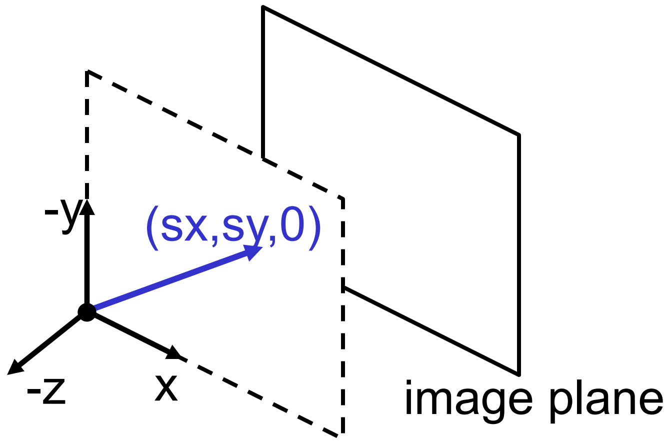
What is the intersection of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$ ?

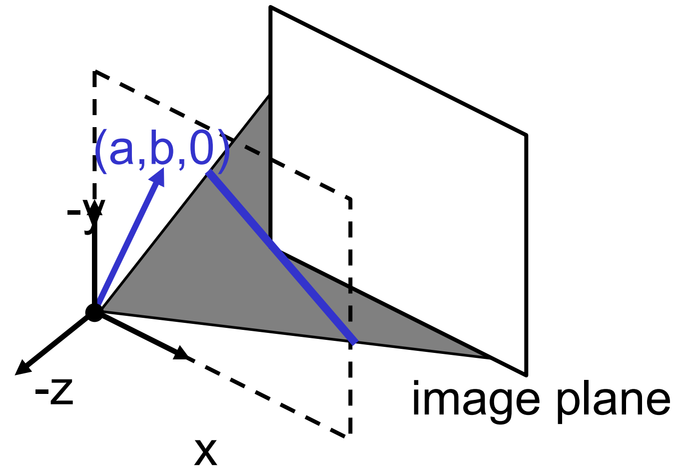
- **p** is ⊥ to $\mathbf{l}_1$ and $\mathbf{l}_2$ $\Rightarrow$ $\mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$

Points and lines are *dual* in projective space

- given any formula, 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)
  - goes through image origin (*principle point*)

# 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 3D: **N P**=0
- Projective transformations
  - Represented by 4x4 matrices **T**: $\mathbf{P}' = \mathbf{TP}, \quad \mathbf{N}' = \mathbf{N}\,\mathbf{T}^{-1}$

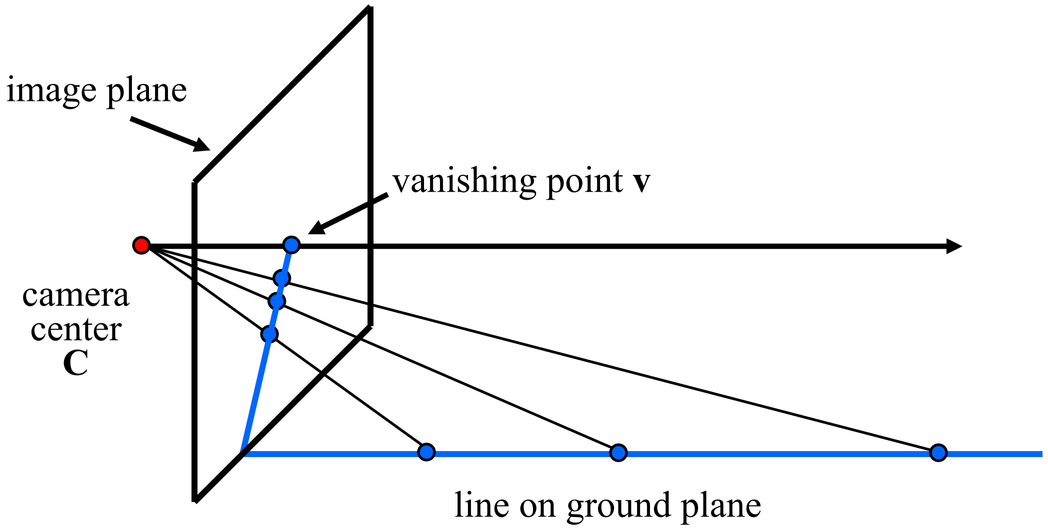
# 3D to 2D: "perspective" projection

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?

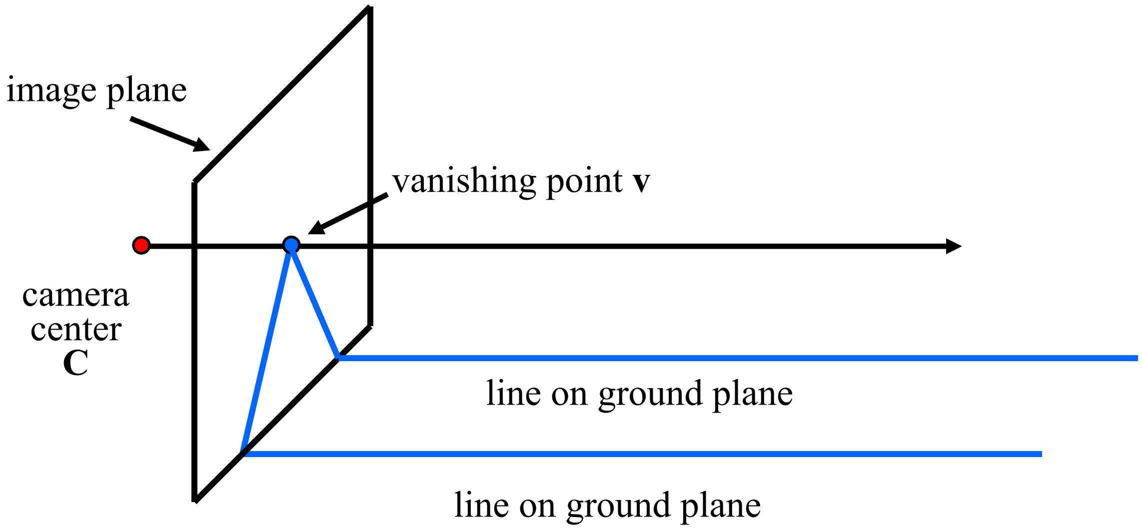
# Vanishing points




## Vanishing point

- projection of a point at infinity

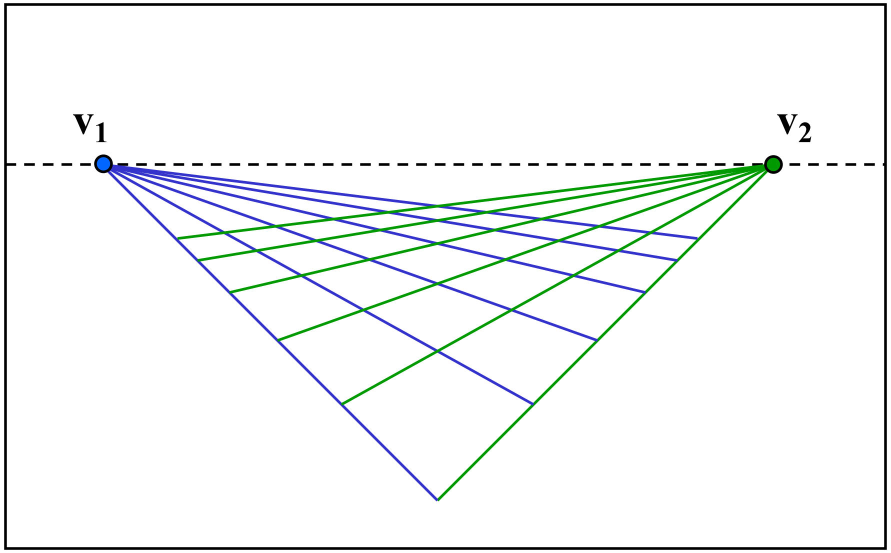
# Vanishing points




## 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

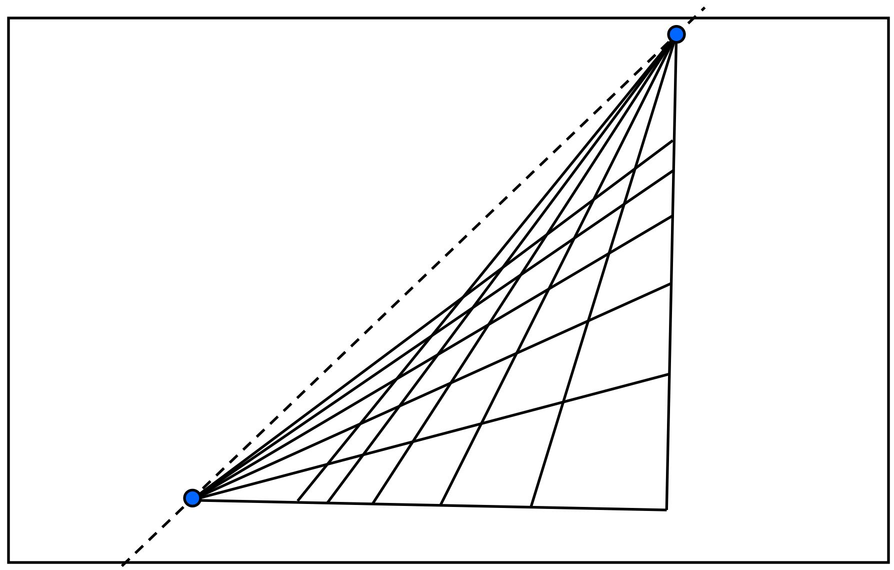
# Vanishing lines




## Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of vanishing points from lines on the same plane is the *vanishing line*
  - For the ground plane, this is called the *horizon*

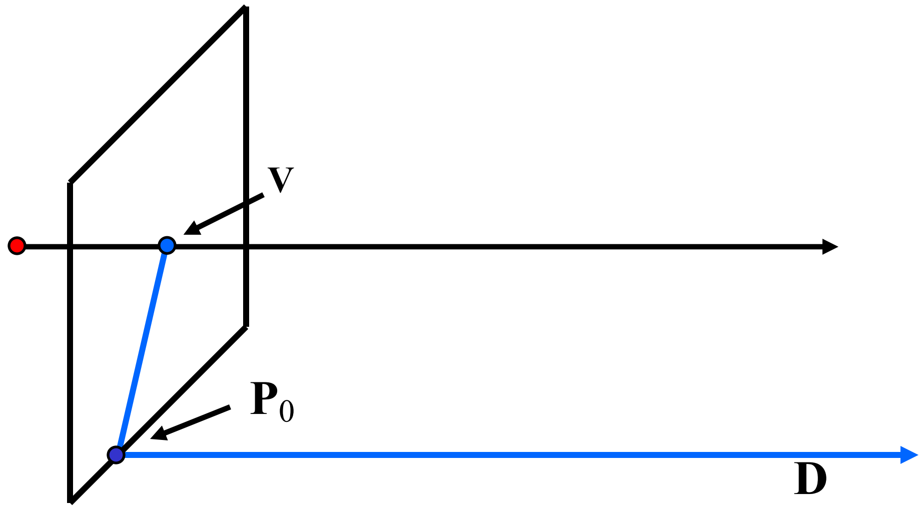
# Vanishing lines



## Multiple Vanishing Points

- Different planes define different vanishing lines

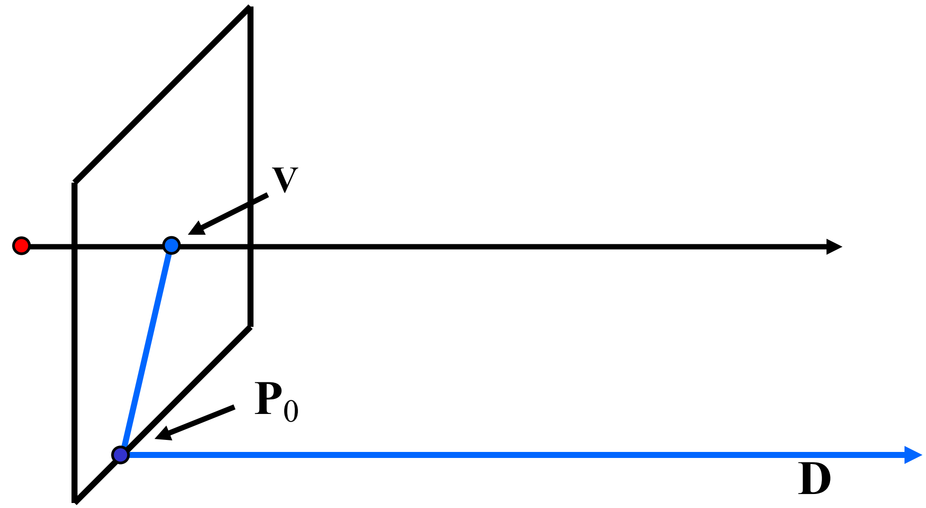
# Computing vanishing points




$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix}$$

# Computing vanishing points



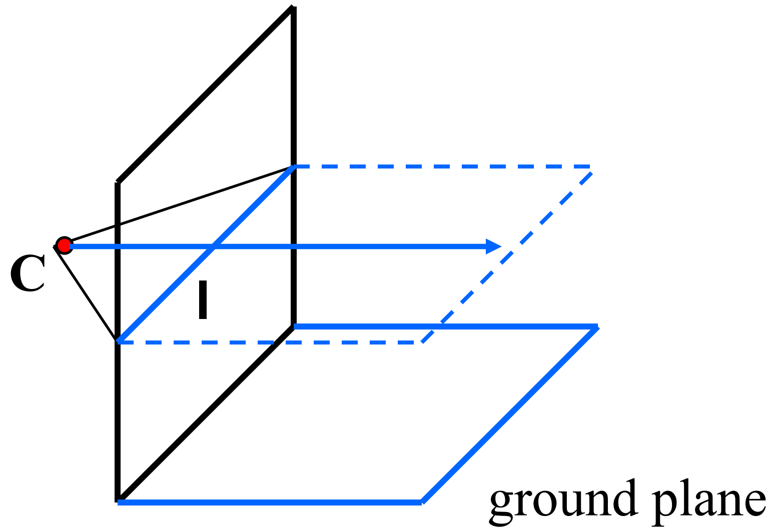

$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

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

Properties $\quad \mathbf{v} = \mathbf{\Pi}\mathbf{P}_\infty$

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

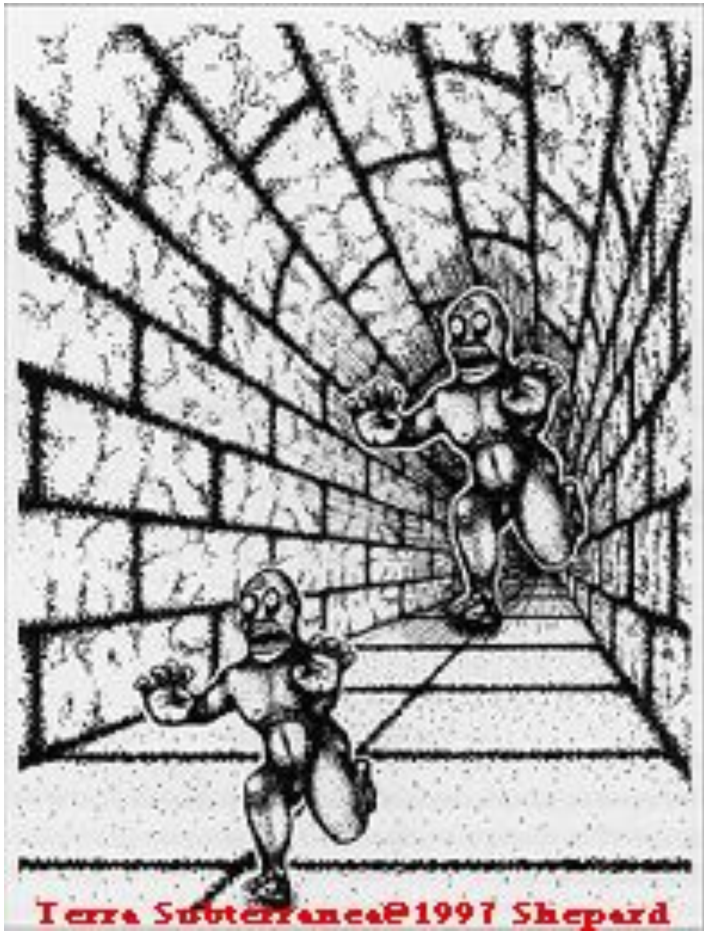
# Computing the horizon

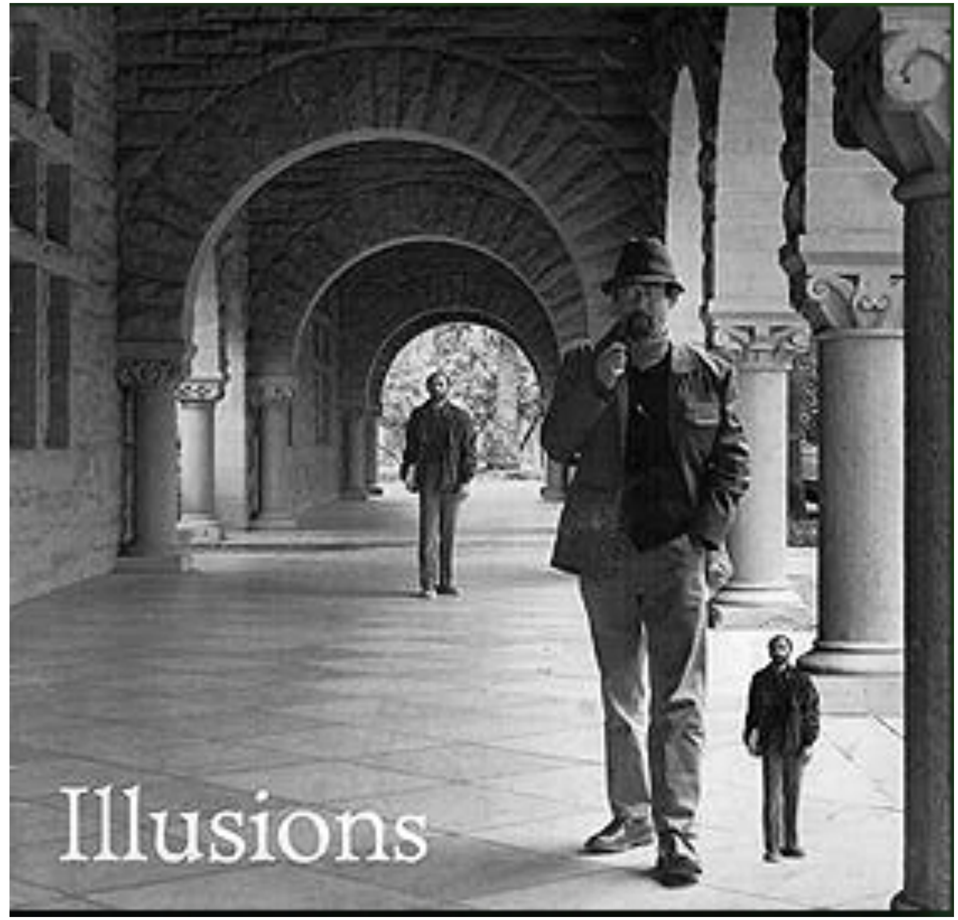



## Properties

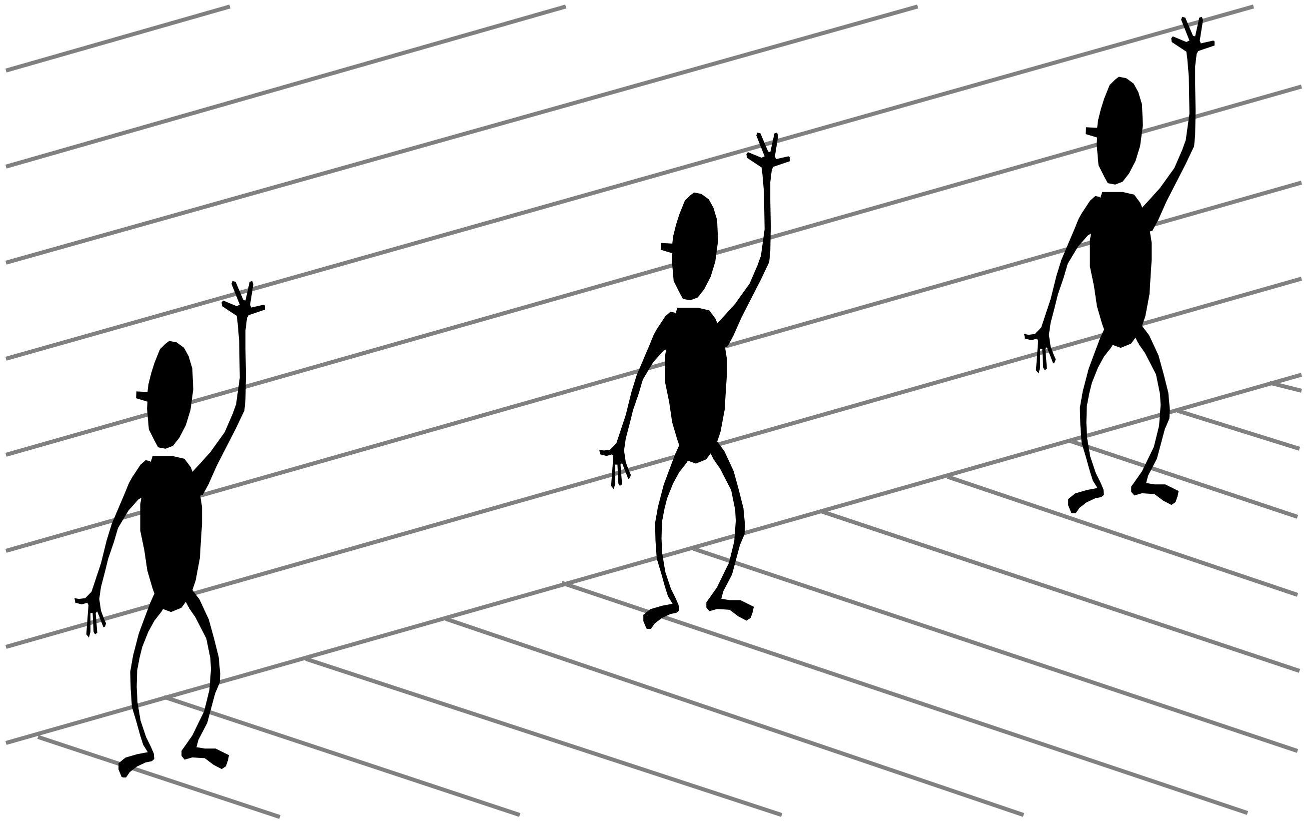
- **l** is intersection of horizontal plane through **C** with image plane
- Compute **l** from two sets of parallel lines on ground plane
- All points at same height as **C** project to **l**
  - points higher than **C** project above **l**
- Provides way of comparing height of objects in the scene

# Fun with vanishing points

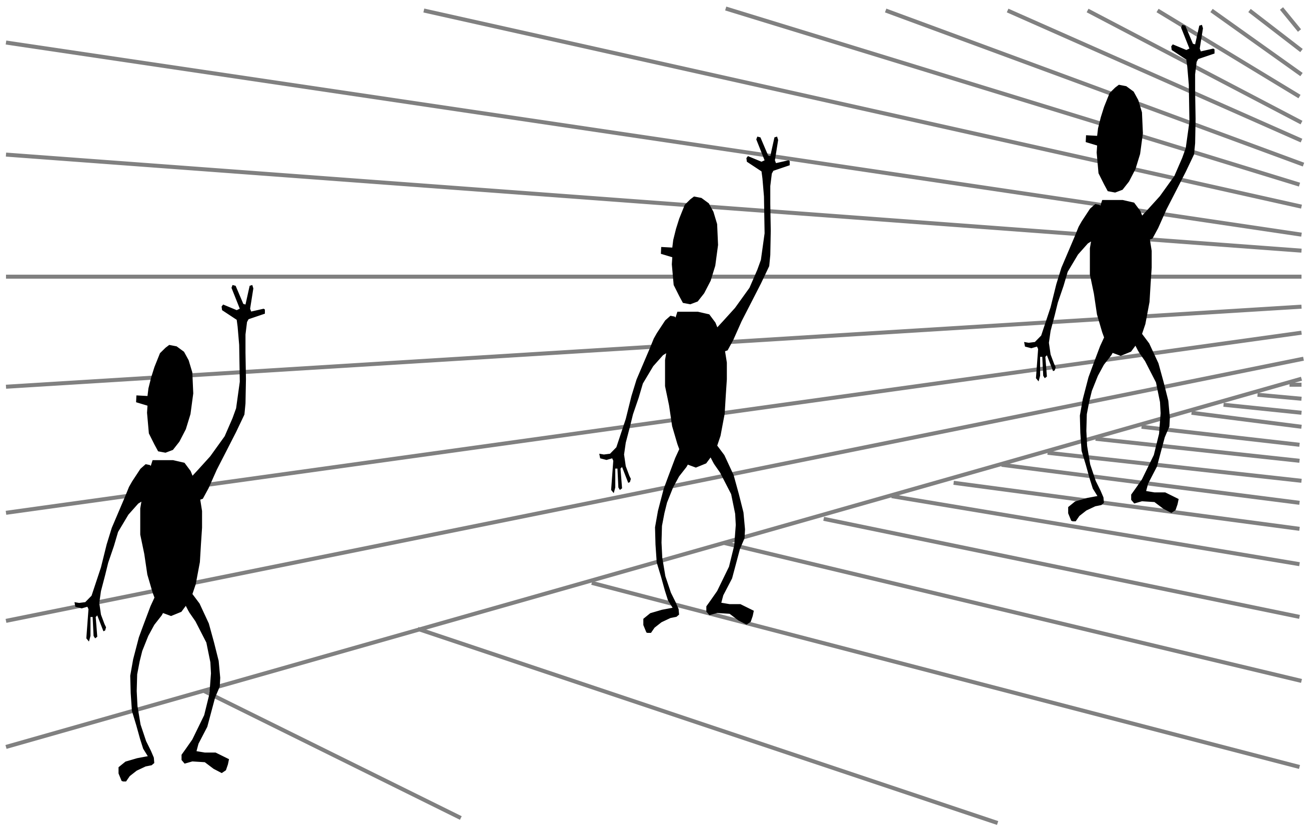


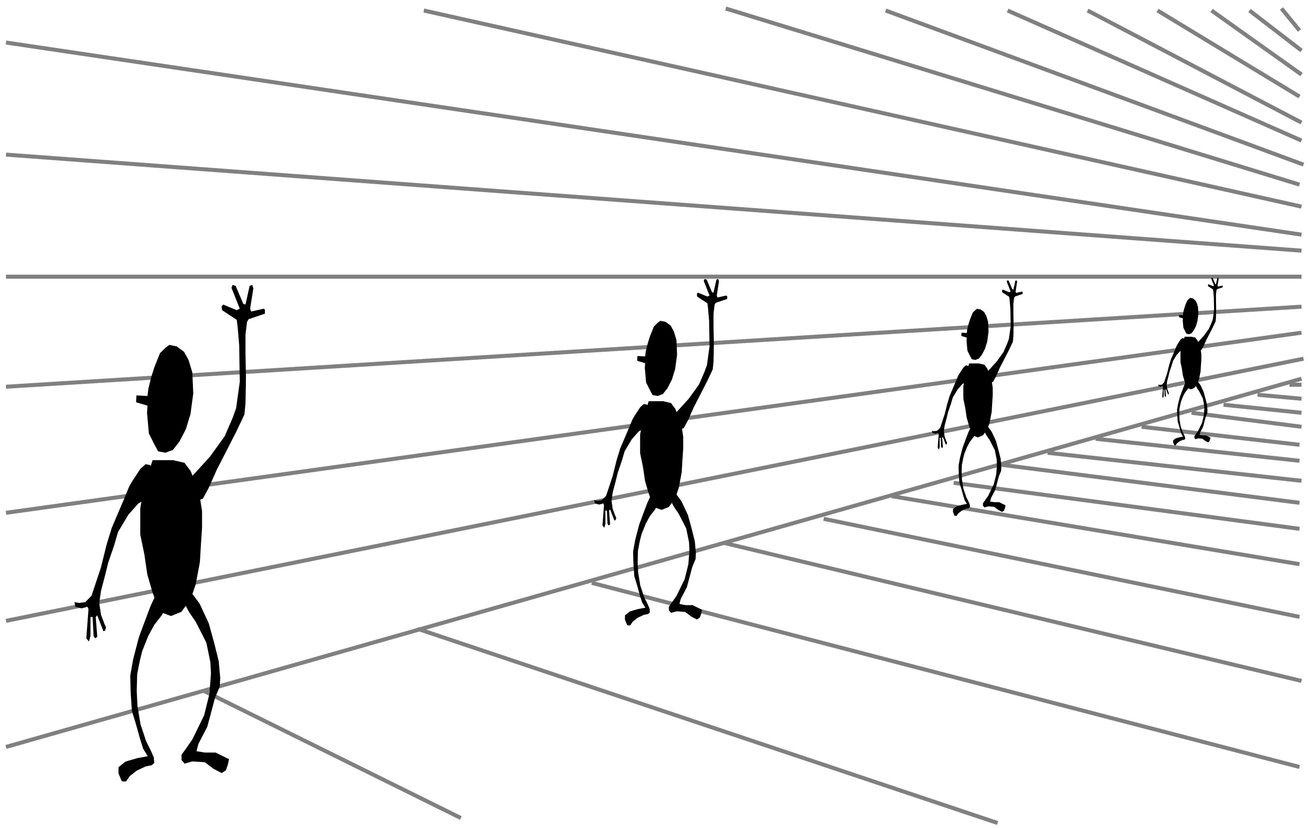
# Perspective cues



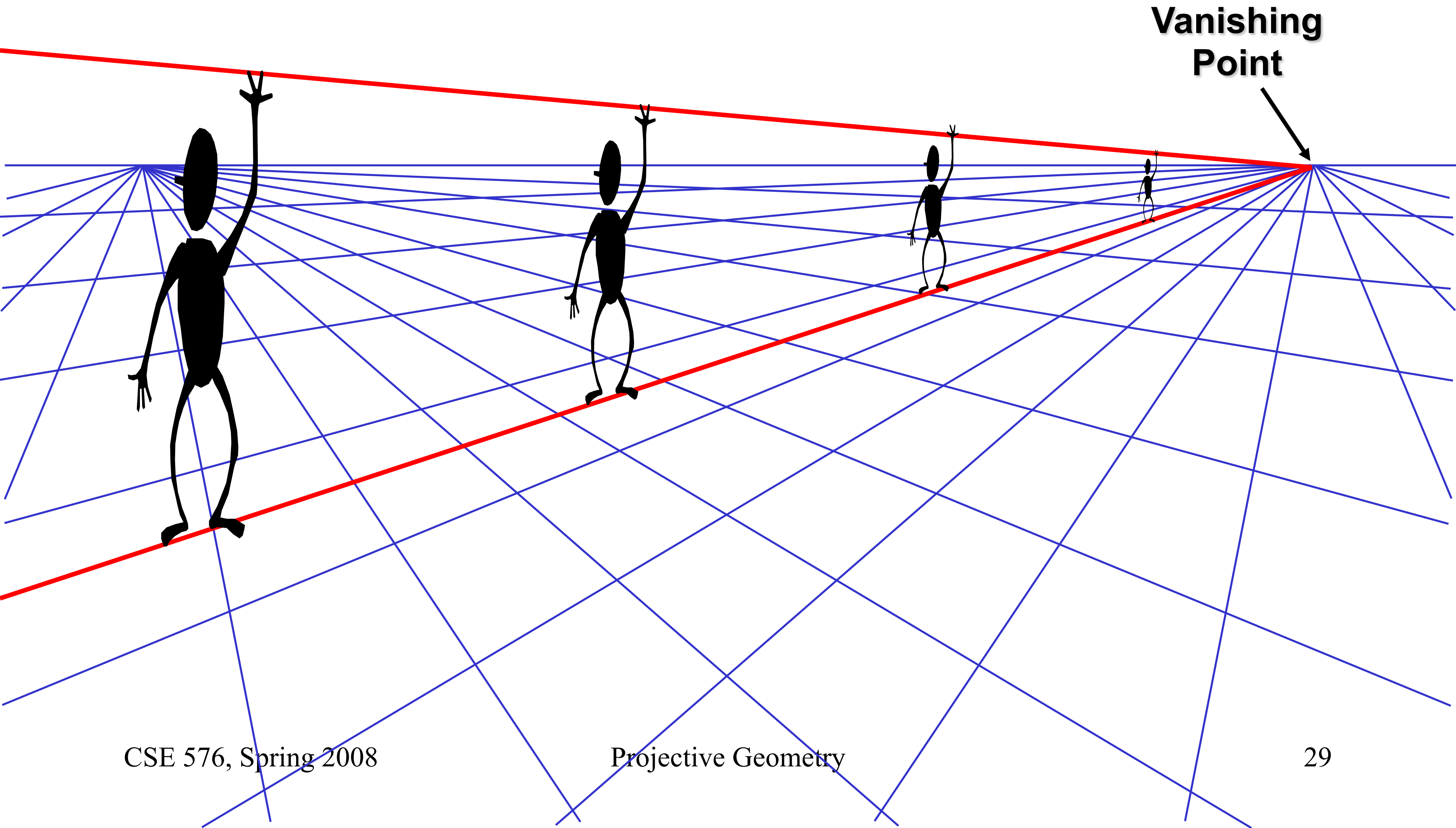
# Perspective cues



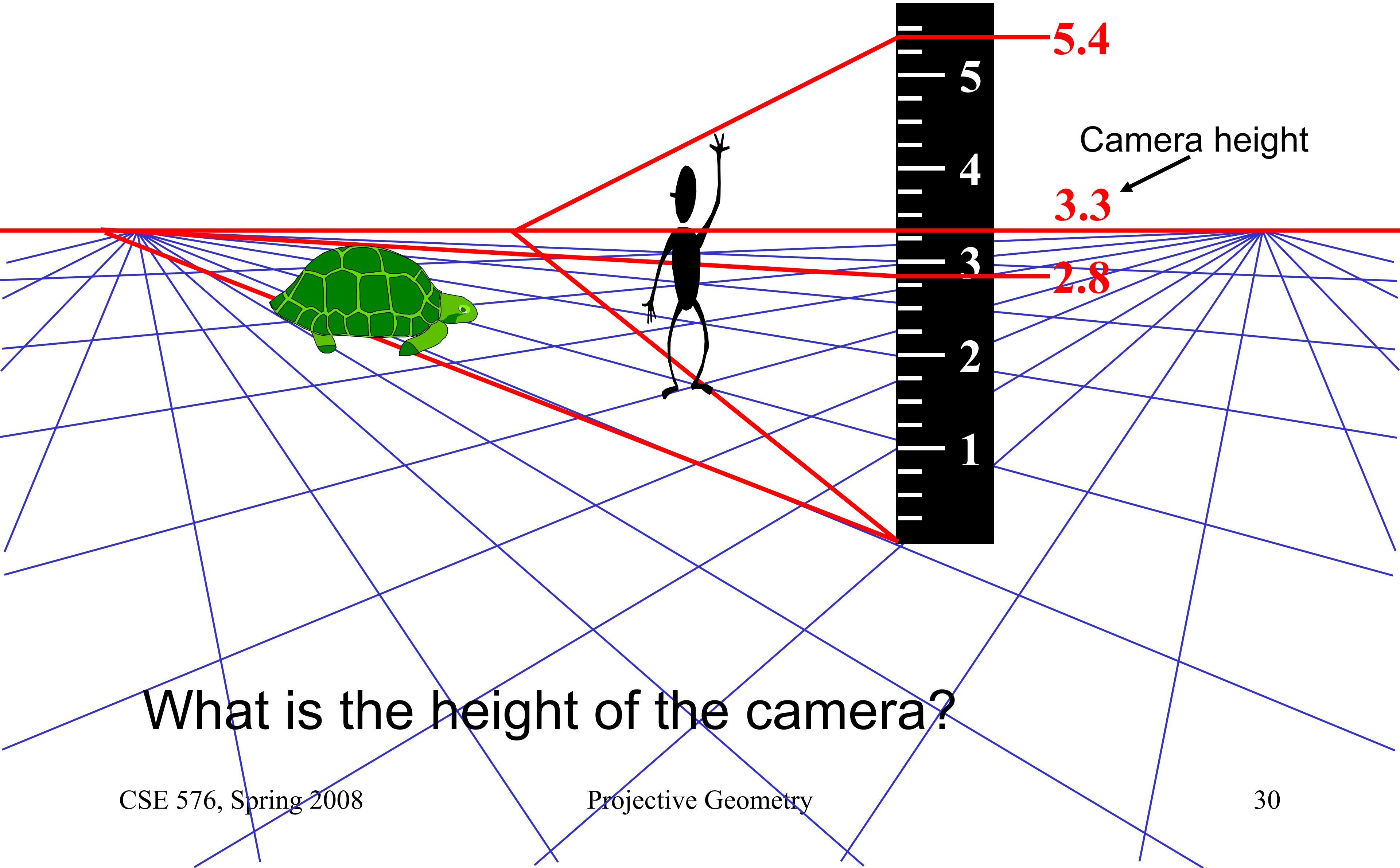
# Perspective cues



# Comparing heights

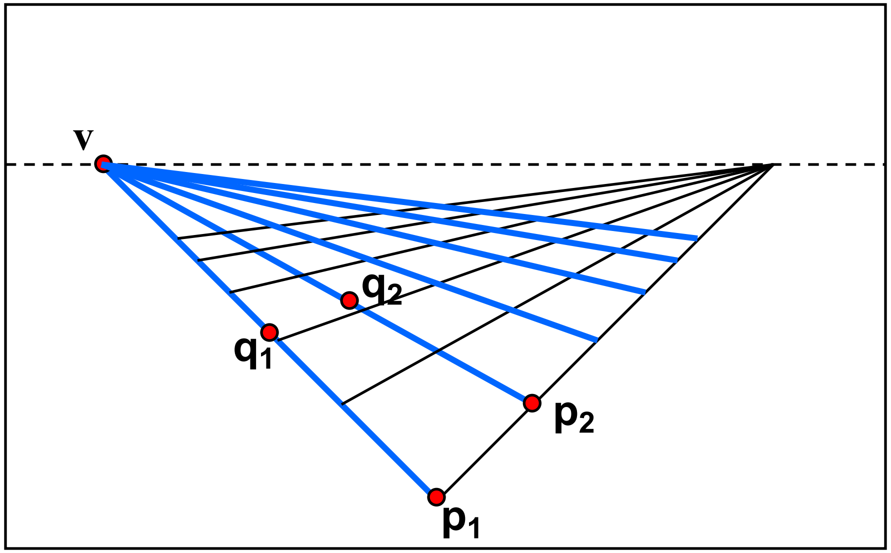


# Measuring height




What is the height of the camera?

# Computing vanishing points (from lines)



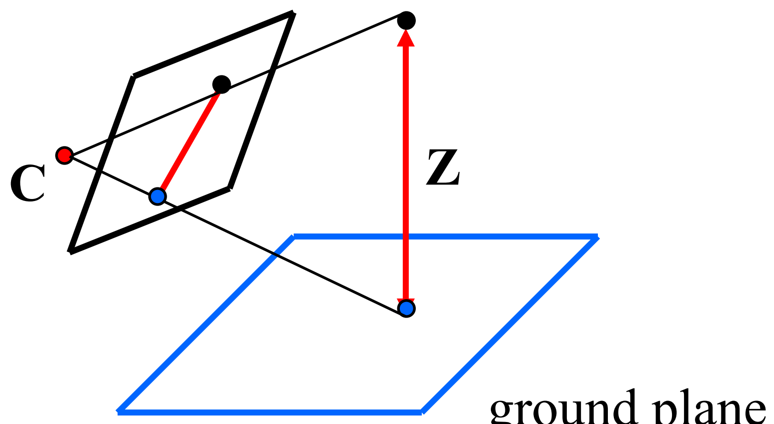

Intersect $p_1q_1$ with $p_2q_2$

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

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

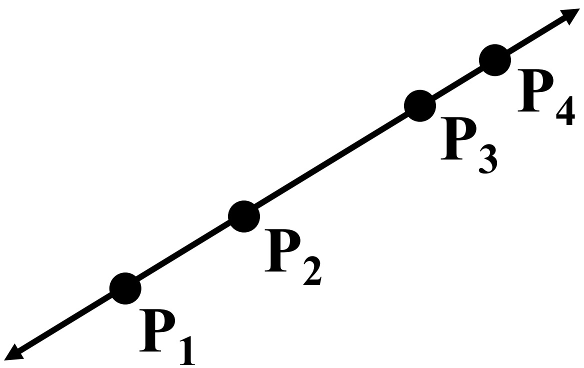
- Need more than vanishing points to do this

# The cross ratio

## A Projective Invariant

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

## The cross-ratio of 4 collinear points




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

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

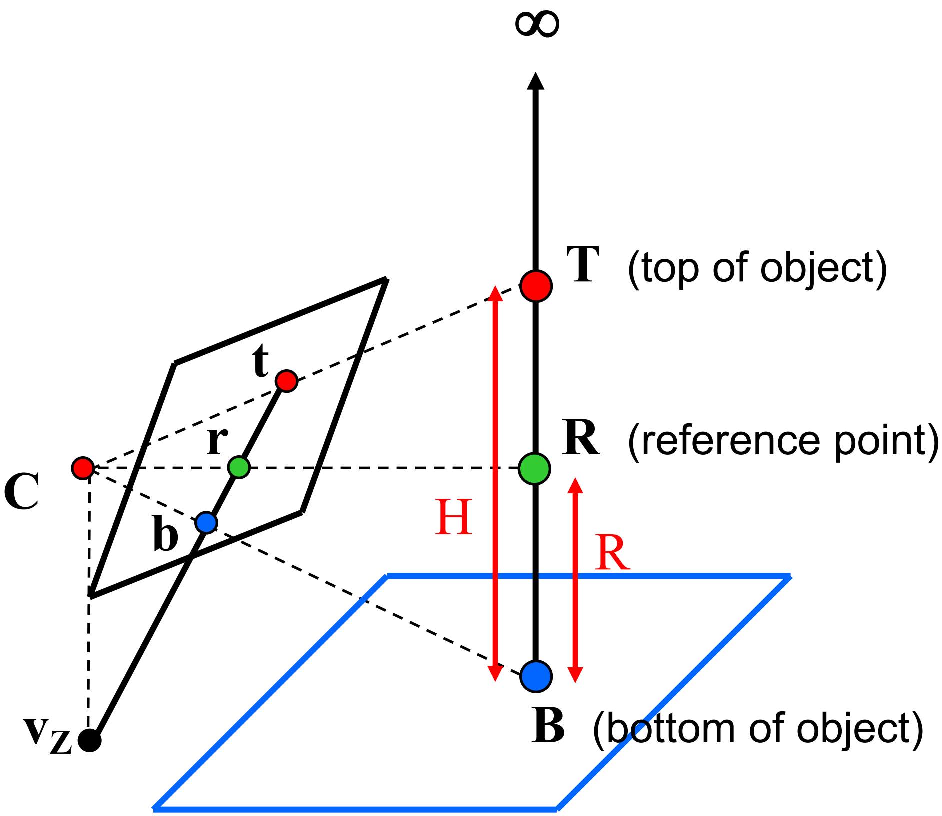
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

# Measuring height




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

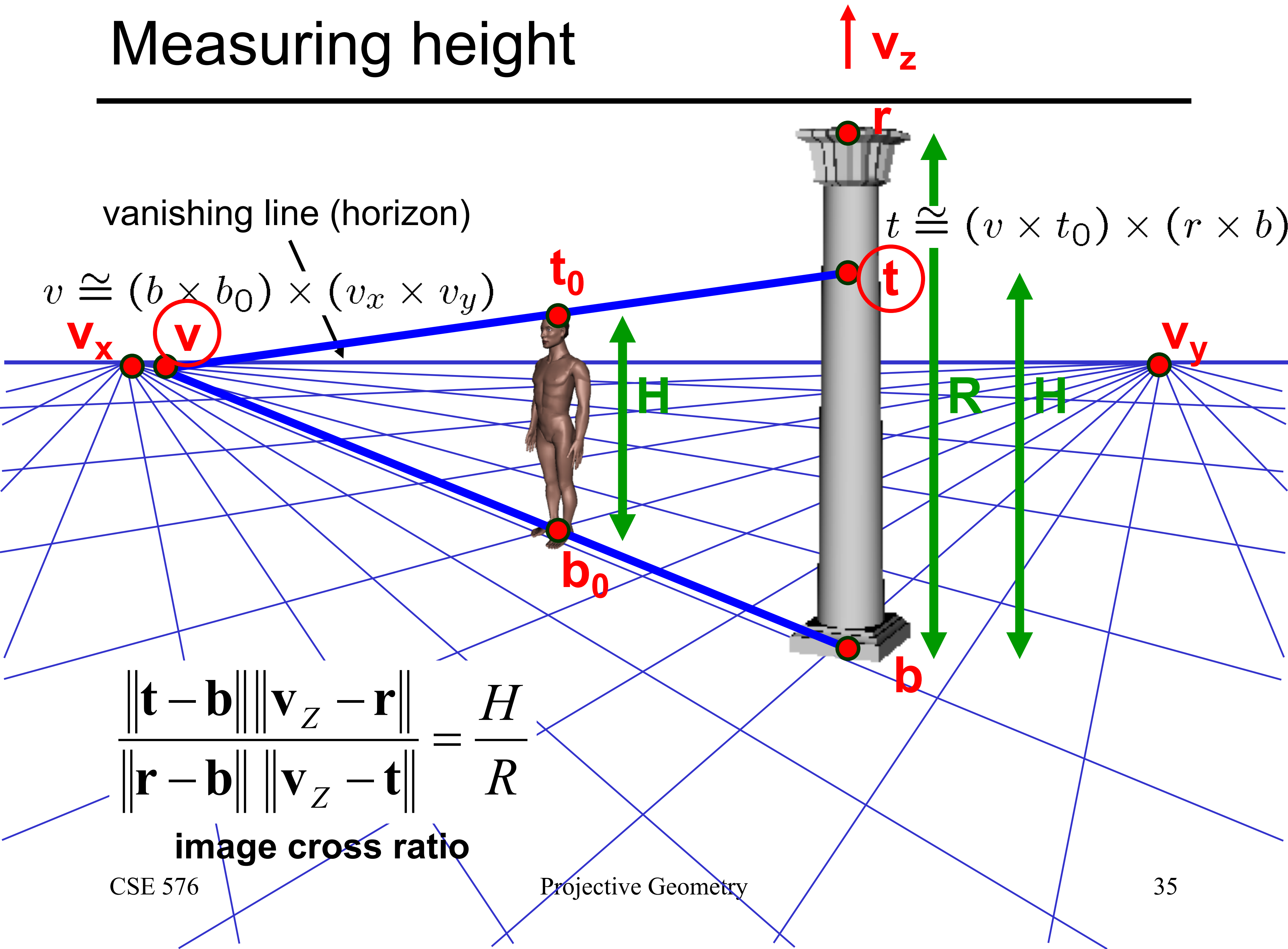
**scene cross ratio**

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

**image cross ratio**

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

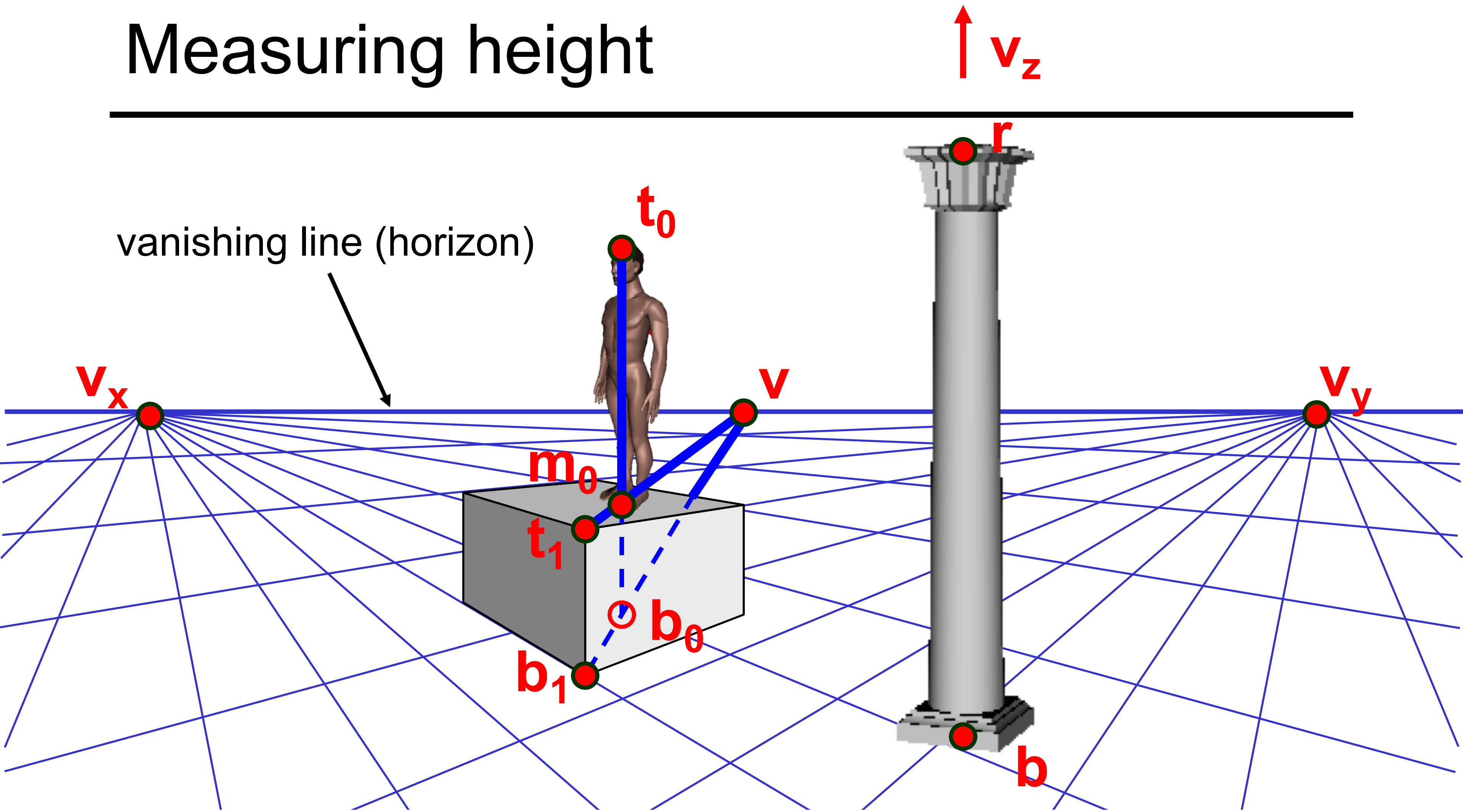
# Measuring height




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

**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

# 3D Modeling from a photograph

# 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

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

- $\boldsymbol{\pi}_1 = \mathbf{\Pi}\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ = $\mathbf{v}_x$ (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 and **o** up to a scale factor

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

- Need a bit more work to get these scale factors…

# Finding the scale factors…

Let's assume that the camera is reasonable

- Square pixels
- Image plane parallel to sensor plane
- Principal point in the center of the image

$$\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/f \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & t'_1 \\ r_{21} & r_{22} & r_{23} & t'_2 \\ r_{31}/f & r_{32}/f & r_{33}/f & t'_3/f \end{bmatrix}$$

$$\begin{bmatrix} t'_1 \\ t'_2 \\ t'_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

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

# Solving for f

$$\begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & o_1 \\ v_{x2} & v_{y2} & v_{z2} & o_2 \\ v_{x3} & v_{y3} & v_{z3} & o_3 \end{bmatrix} = \begin{bmatrix} r_{11}/a & r_{12}/b & r_{13}/c & t'_1/d \\ r_{21}/a & r_{22}/b & r_{23}/c & t'_2/d \\ r_{31}/af & r_{32}/bf & r_{33}/cf & t'_3/df \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{v}_X & \mathrm{v}_Y & \mathrm{v}_Z & \mathrm{o} \end{bmatrix}$$

$$\begin{bmatrix} \boxed{\begin{matrix} v_{x1} \\ v_{x2} \\ fv_{x3} \end{matrix}} & \boxed{\begin{matrix} v_{y1} \\ v_{y2} \\ fv_{y3} \end{matrix}} & \begin{matrix} v_{z1} \\ v_{z2} \\ fv_{z3} \end{matrix} & \begin{matrix} o_1 \\ o_2 \\ fo_3 \end{matrix} \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} r_{11}/a \\ r_{21}/a \\ r_{31}/a \end{matrix}} & \boxed{\begin{matrix} r_{12}/b \\ r_{22}/b \\ r_{32}/b \end{matrix}} & \begin{matrix} r_{13}/c \\ r_{23}/c \\ r_{33}/c \end{matrix} & \begin{matrix} t'_1/d \\ t'_2/d \\ t'_3/d \end{matrix} \end{bmatrix}$$

Orthogonal vectors

Orthogonal vectors

$$v_{x1}v_{y1} + v_{x2}v_{y2} + f^2 v_{x3}v_{y3} = 0 \quad \Rightarrow \quad f = \sqrt{\frac{v_{x1}v_{y1} + v_{x2}v_{y2}}{-v_{x3}v_{y3}}}$$

# Solving for a, b, and c

$$\begin{bmatrix} v_{x1}/f & v_{y1}/f & v_{z1}/f & o_1/f \\ v_{x2}/f & v_{y2}/f & v_{z2}/f & o_2/f \\ v_{x3} & v_{y3} & v_{z3} & o_3 \end{bmatrix} = \begin{bmatrix} r_{11}/a & r_{12}/b & r_{13}/c & t'_1/d \\ r_{21}/a & r_{22}/b & r_{23}/c & t'_2/d \\ r_{31}/a & r_{32}/b & r_{33}/c & t'_3/d \end{bmatrix}$$

Norm = 1/a

Norm = 1/a

## Solve for a, b, c

- Divide the first two rows by f, now that it is known
- Now just find the norms of the first three columns
- Once we know a, b, and c, that also determines R

## How about d?

- Need a reference point in the scene

# Solving for d

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t'_1/d \\ r_{21} & r_{22} & r_{23} & t'_2/d \\ r_{31} & r_{32} & r_{33} & t'_3/d \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ H \\ 1 \end{bmatrix}$$

Suppose we have one reference height H

- E.g., we known that (0, 0, H) gets mapped to (u, v)

$$u = \frac{r_{13}H + t'_1/d}{r_{33}H + t'_3/d} \qquad d = \frac{t'_1 - ut'_3}{ur_{33}H - r_{13}H}$$

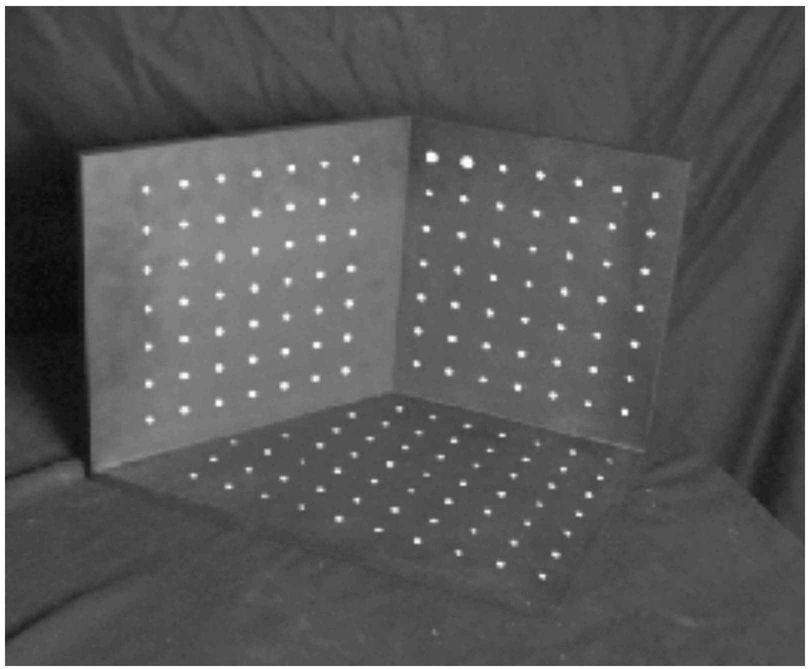
Finally, we can solve for t

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} t'_1 \\ t'_2 \\ t'_3 \end{bmatrix}$$

# Calibration using a reference object

Place a known object in the scene

- identify correspondence between image and scene
- compute mapping from scene to image



Issues

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

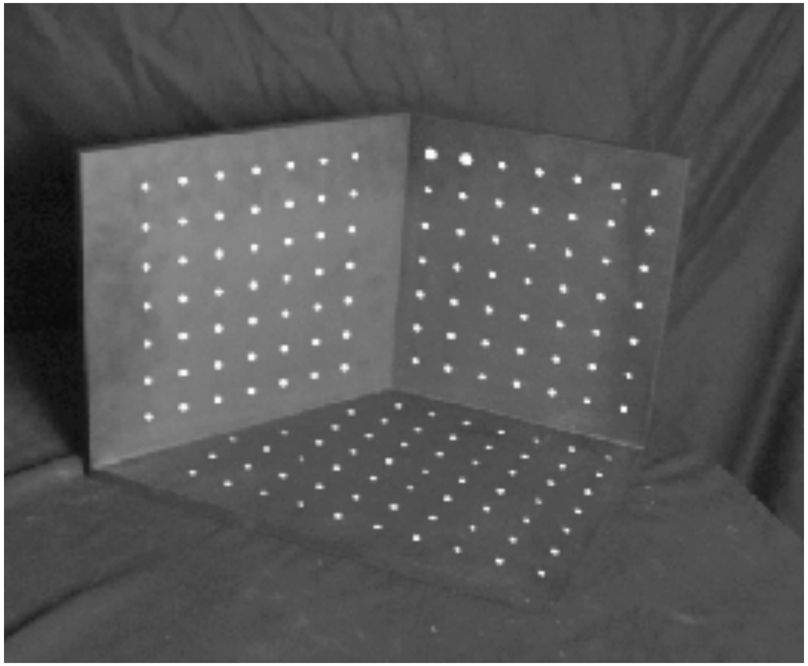
# Chromaglyphs

Courtesy of Bruce Culbertson, HP Labs
http://www.hpl.hp.com/personal/Bruce_Culbertson/ibr98/chromagl.htm

# 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

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

$$\begin{aligned} u_i &= \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \\ v_i &= \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \end{aligned}$$

$$u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}$$
$$v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}$$

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

# Direct linear calibration

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

Can solve for $m_{ij}$ by linear least squares

- use eigenvector trick that we used for homographies

# Direct linear calibration

Advantage:

- Very simple to formulate and solve

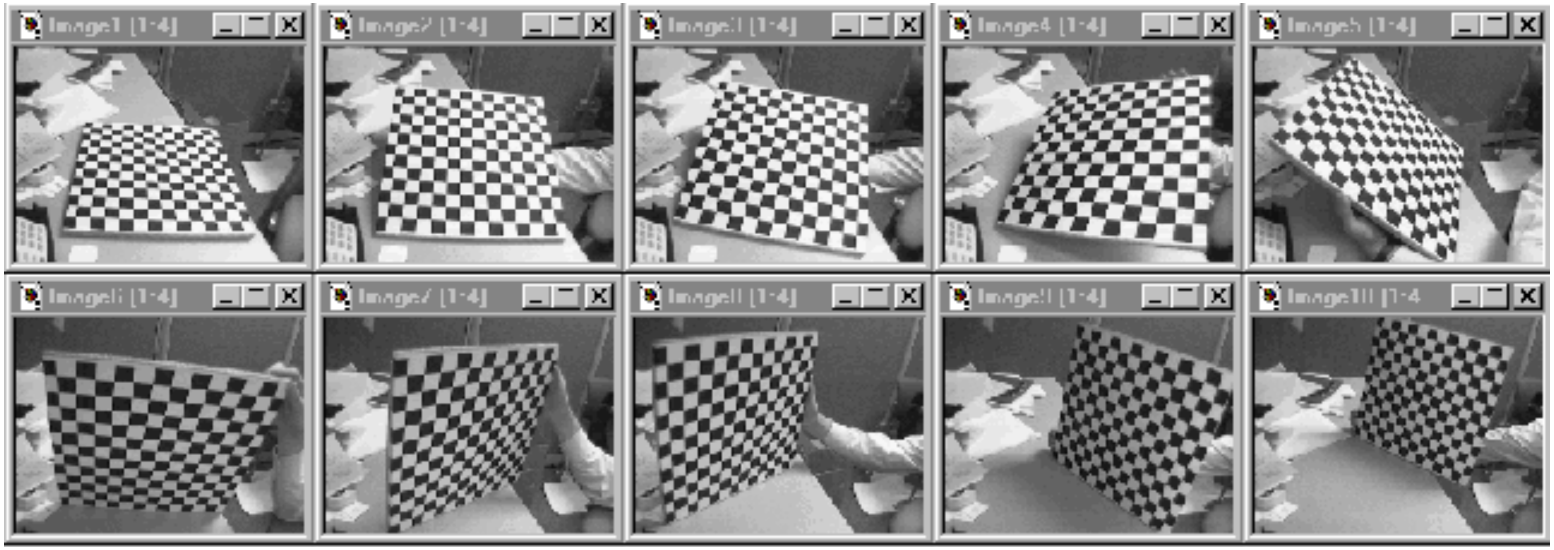
Disadvantages:

- Doesn't tell you the camera parameters
- Doesn't model radial distortion
- Hard to impose constraints (e.g., known focal length)
- Doesn't minimize the right error function

For these reasons, *nonlinear methods* are preferred

- Define error function E between projected 3D points and image positions
  - E is nonlinear function of intrinsics, extrinsics, radial distortion
- Minimize E using nonlinear optimization techniques
  - e.g., variants of Newton's method (e.g., Levenberg Marquart)

# 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!
  - Intel's OpenCV library: http://www.intel.com/research/mrl/research/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/

# Some Related Techniques

## Image-Based Modeling and Photo Editing

- Mok et al., SIGGRAPH 2001
- http://graphics.csail.mit.edu/ibedit/

## Single View Modeling of Free-Form Scenes

- Zhang et al., CVPR 2001
- http://grail.cs.washington.edu/projects/svm/

## Tour Into The Picture

- Anjyo et al., SIGGRAPH 1997
- http://koigakubo.hitachi.co.jp/little/DL_TipE.html