CS5670: Computer Vision

Noah Snavely

Single-View Modeling

Congratulations to Noah!

It’s Baby Charlie!

Some Notes

- I (Abe) will take on most of the instructor duties moving forward
  - I will defer to Noah's precedent on course structure and policy whenever possible
- Bear with us on the transition
  - First time teaching a course, but frequently give talks (hopefully this will translate well…)
  - If I can't answer a question in class, I will try to answer it at the beginning of next class or send out a correction.
  - I may occasionally describe things differently from Noah---don't hesitate to ask for clarifications

Single-View Modeling

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

Announcements

- Midterm Due Today
- Assignment 3 (DEMO)

Roadmap ahead

- The next few lectures will finish up geometry
  - Next up is recognition / learning

- We already know about camera geometry & panoramas
- Coming up
  - Single-view modeling (today)
  - Two-view geometry
  - Multi-view geometry

Ames Room

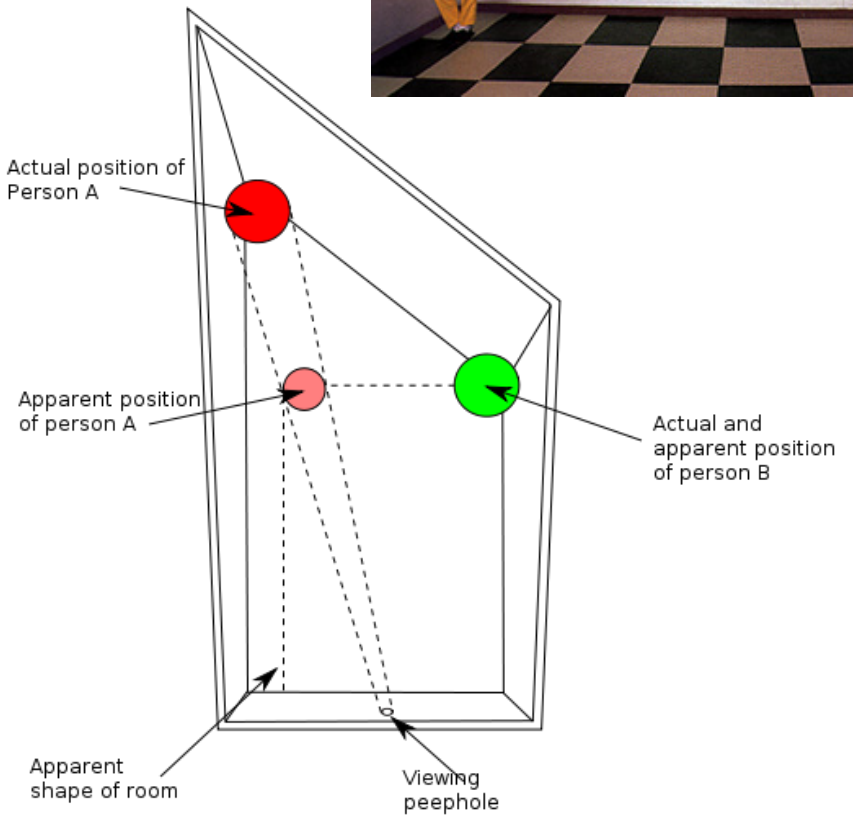
https://www.youtube.com/watch?v=bZGWRJyo0YU

Ames Room: What does this illustrate?

- Computer vision is under-constrained
- One image has multiple explanations
  - We are good at picking the most likely one

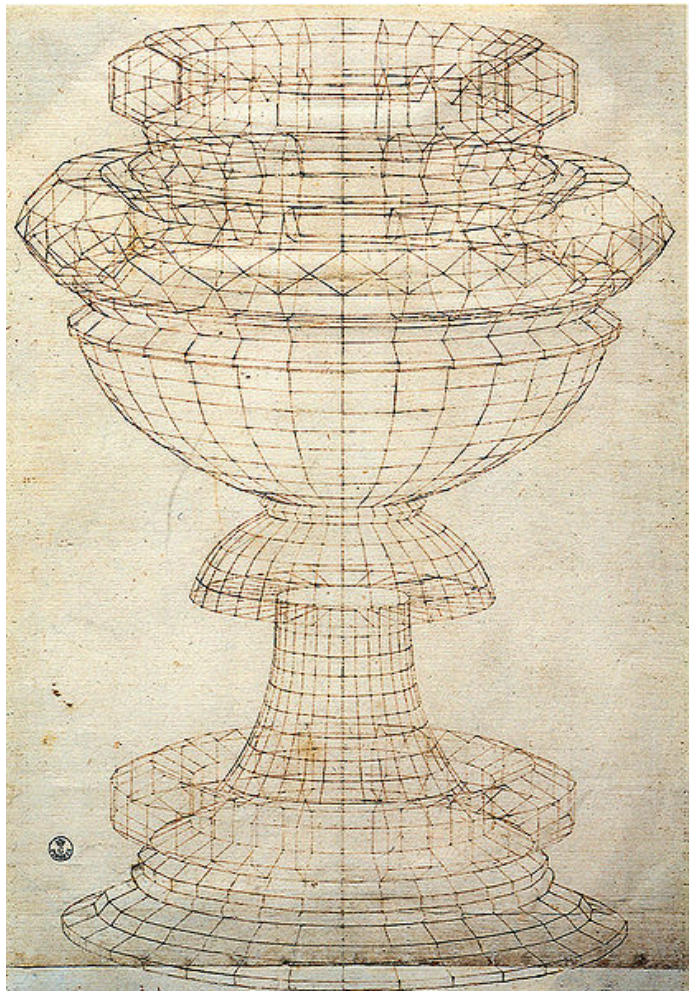
"Are children small, or just far away?"

-Magical Mysteries
by Thrilla Killa Klownz
SNL Digital Short



Projective geometry—what's it good for?

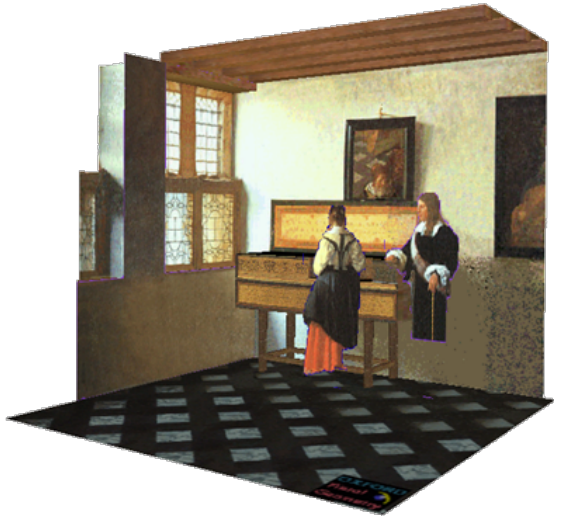
- Uses of projective geometry
  - Drawing
  - Measurements
  - Mathematics for projection
  - Undistorting images
  - Camera pose estimation
  - Object recognition



Paolo Uccello

Applications of projective geometry

Vermeer's *Music Lesson*



Reconstructions by Criminisi et al.

Making measurements in images

WARBY PARKER

Measure your pupillary distance (PD)

Your PD is the distance between your pupils. To measure it, follow the instructions below — once you submit your photo, our team of experts will determine your PD and email you once we've applied it to your order.

1

Wearing glasses?
Take 'em off before you get started.

2

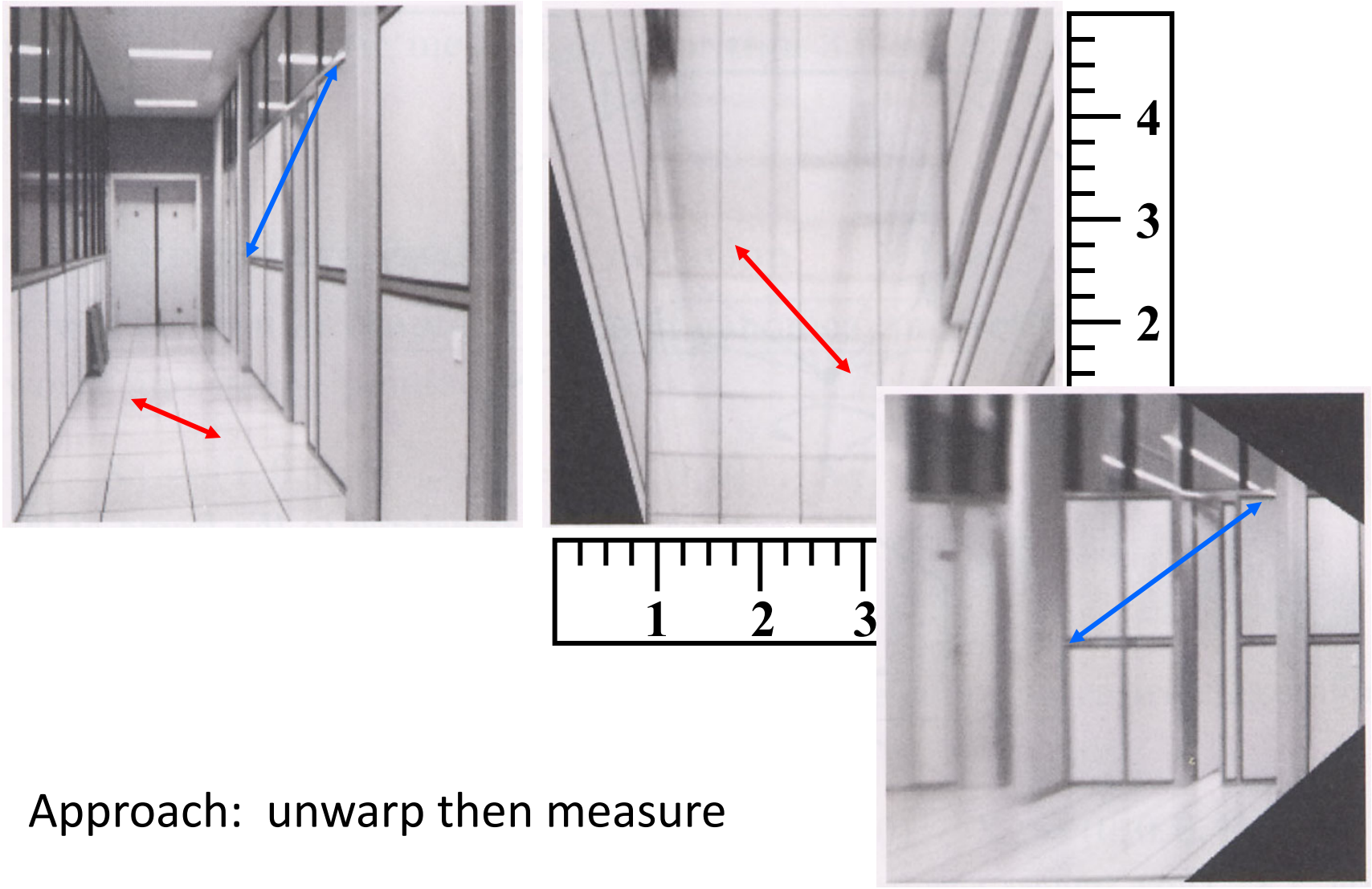
Hold up any card with a magnetic strip (we use this for scale).

3

Look straight ahead and snap a photo.

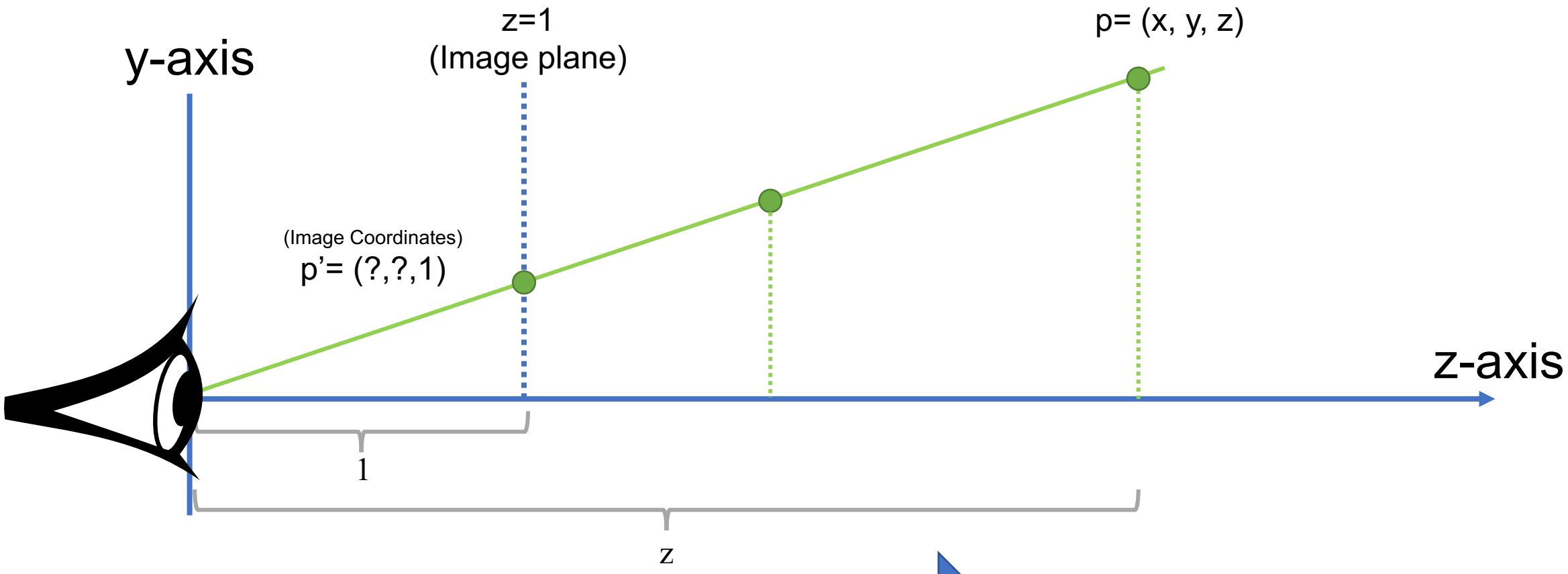
https://www.warbyparker.com/pd/instructions

Measurements on planes



Approach: unwarp then measure

Homogeneous Coordinates & Projection (Review?)



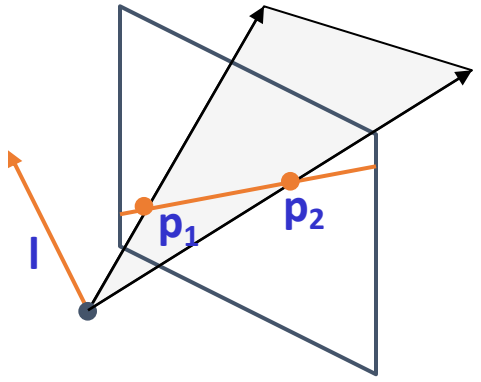
p'= (?,?,1)

Homogenization =
Apply Similar Triangles Rule

p'= (x/z, y/z, 1)

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: $\mathbf{l} \cdot \mathbf{p} = 0$



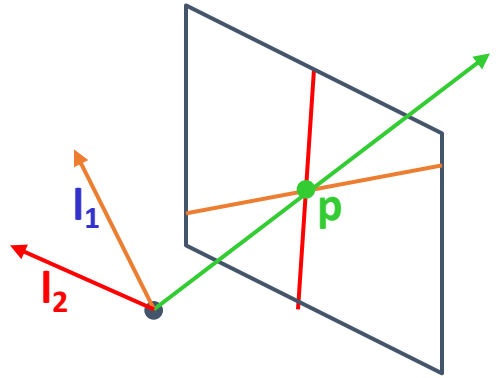
What is the line $\mathbf{l}$ spanned by rays $\mathbf{p}_1$ and $\mathbf{p}_2$?

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

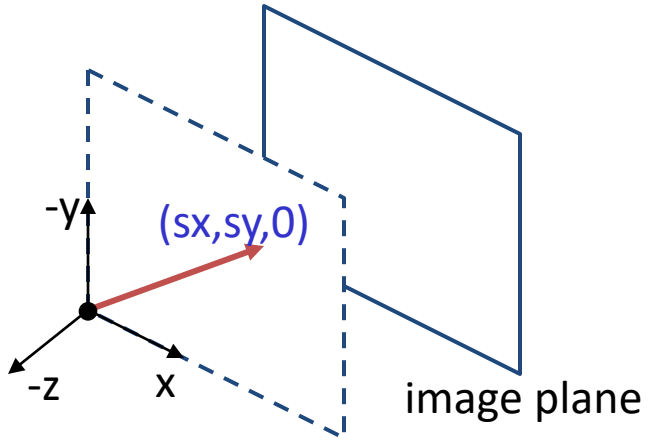
What is the intersection of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$?

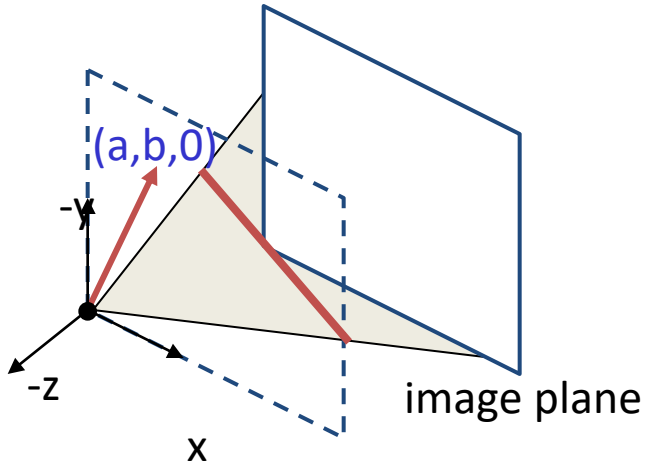
- $\mathbf{p}$ is $\perp$ to $\mathbf{l}_1$ and $\mathbf{l}_2$ $\Rightarrow$ $\mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$

Points and lines are *dual* in projective space



Ideal points and lines





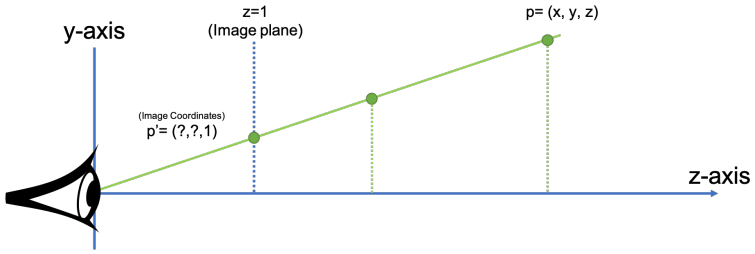
- Ideal point ("point at infinity")
  - $p \cong (x, y, 0)$ – parallel to image plane
  - It has infinite image coordinates

Ideal line

- $l \cong (a, b, 0)$ – parallel to image plane
- Corresponds to a line in the image (finite coordinates)
  - goes through image origin (*principal point*)

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
    - Three points define a plane, three planes define a point



When z-axis is no longer the optical axis of our imaging system (eye or camera), we need to add a new homogeneous coordinate

3D to 2D: perspective projection

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

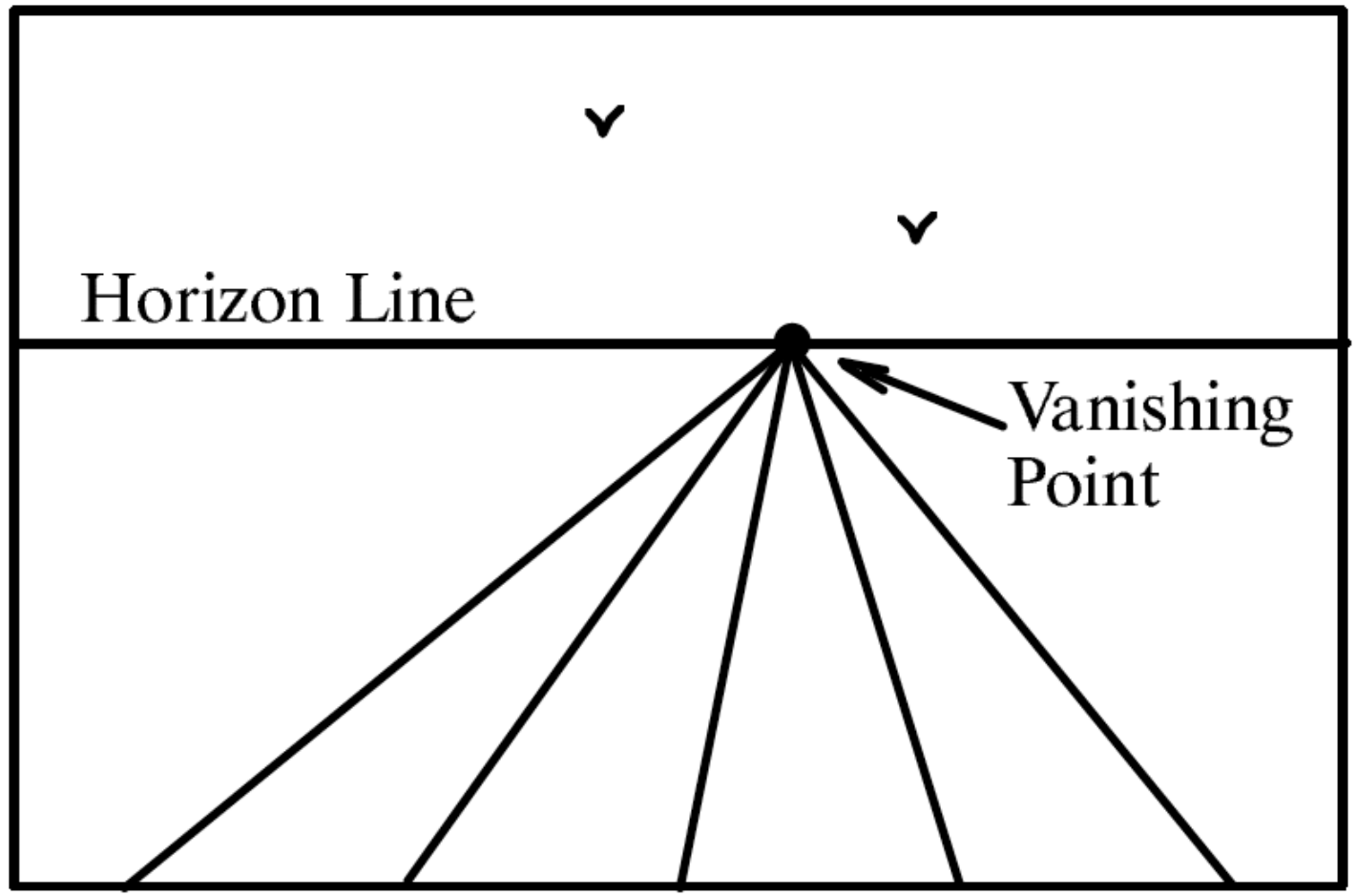
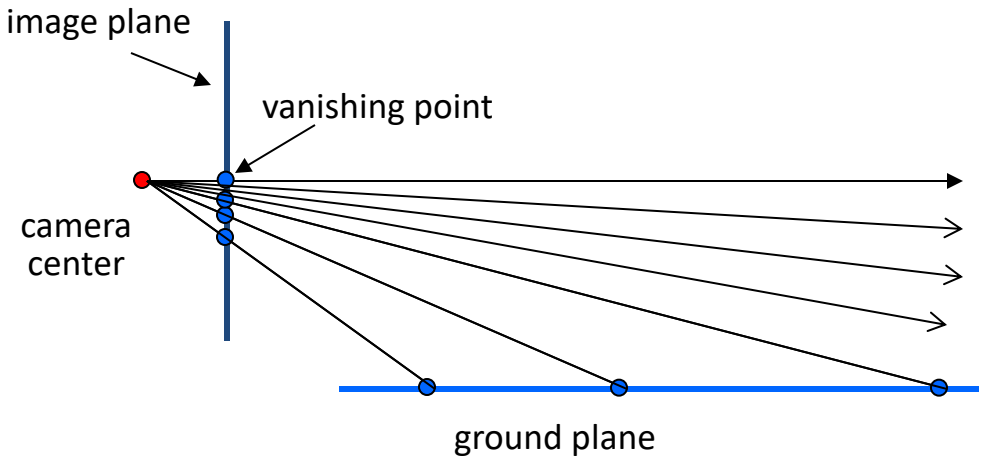
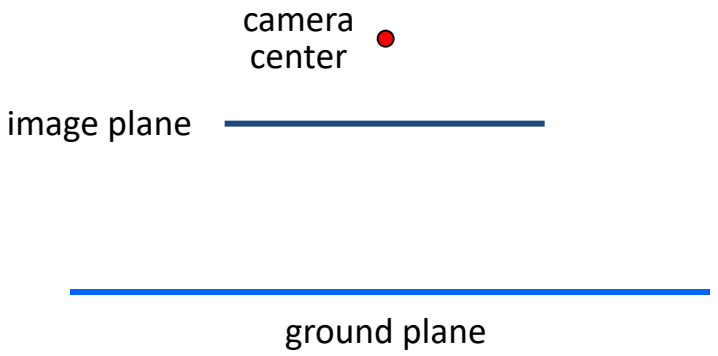


Figure 23.4
A perspective view of a set of parallel lines in the plane. All of the lines converge to a single vanishing point.

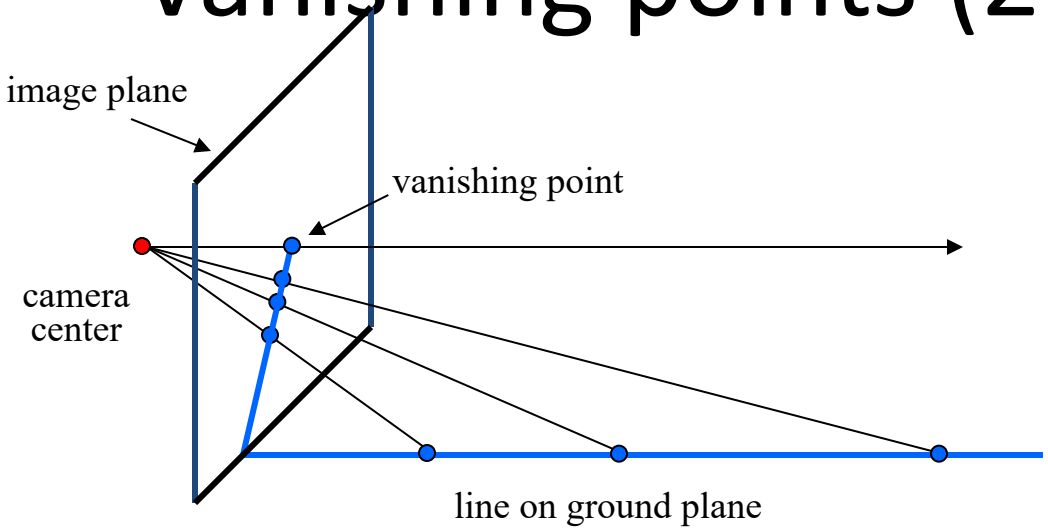
Vanishing points (1D)



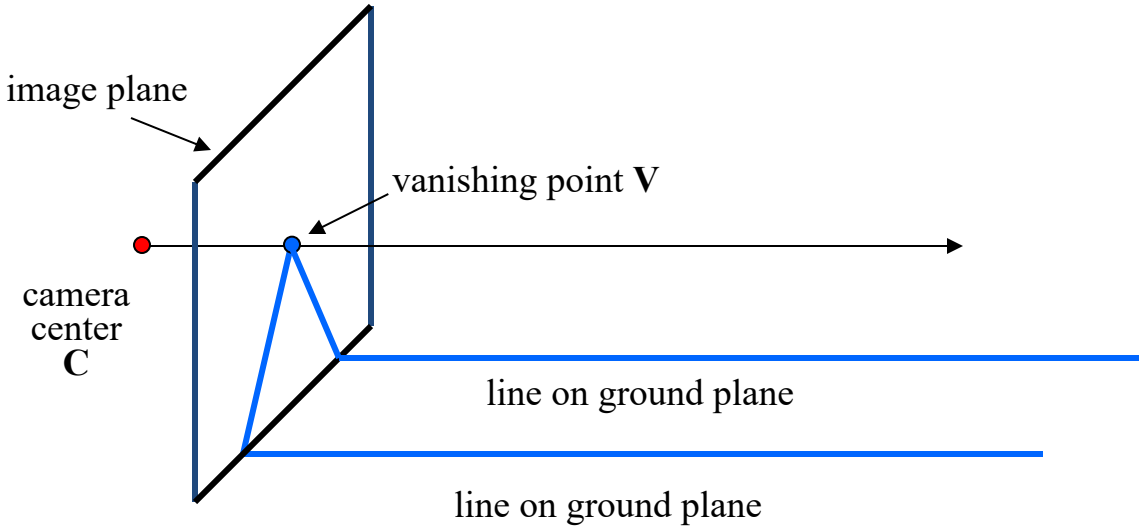
- Vanishing point
  - projection of a point at infinity
  - can often (but not always) project to a finite point in the image



Vanishing points (2D)



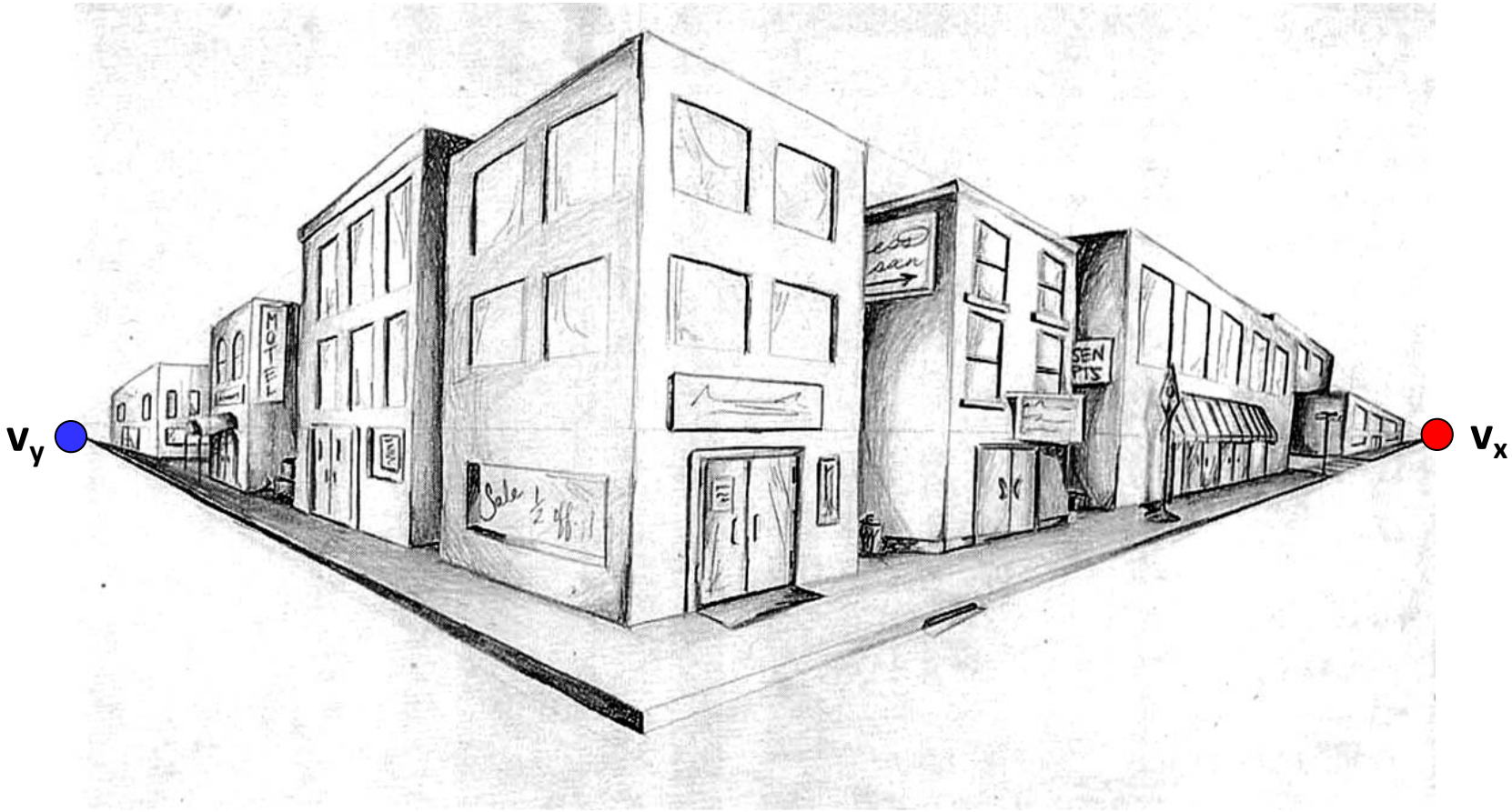
Vanishing points



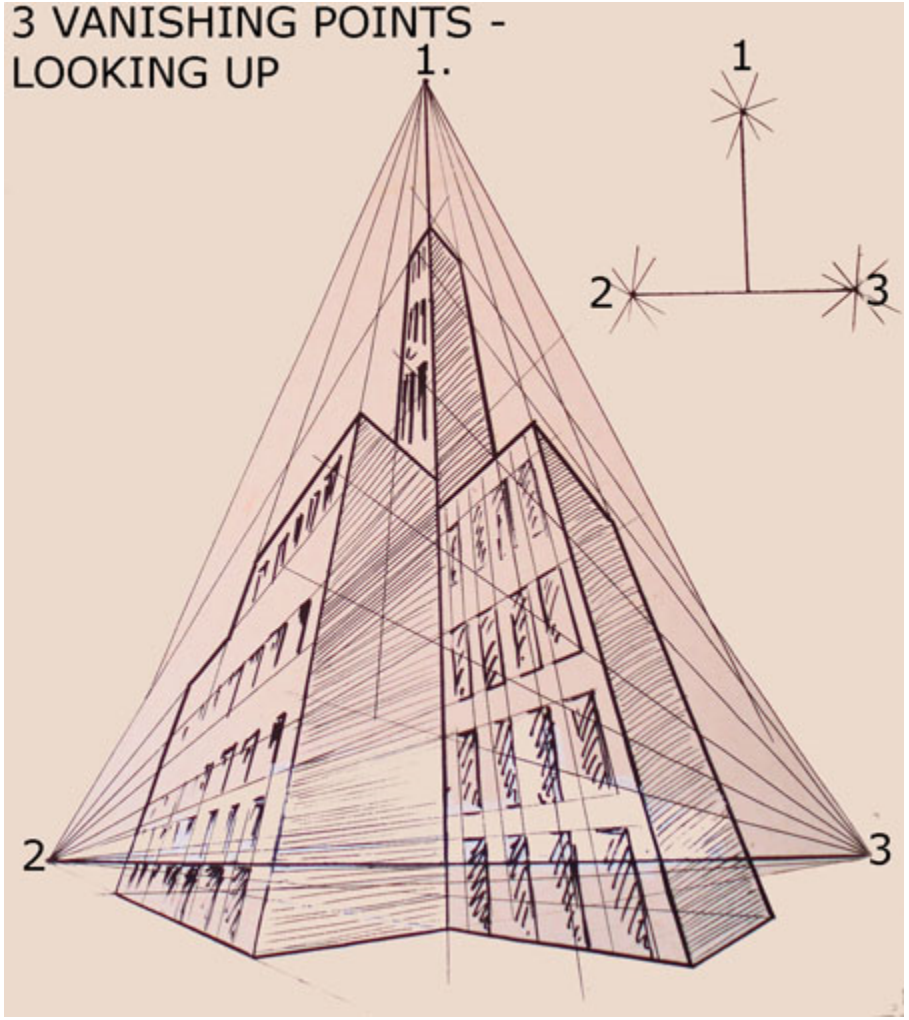
- Properties
  - Any two parallel lines (in 3D) have the same vanishing point **v**
  - The ray from **C** through **v** is parallel to the lines
  - An image may have more than one vanishing point
    - in fact, every image point is a potential vanishing point

One-point perspective

Two-point perspective

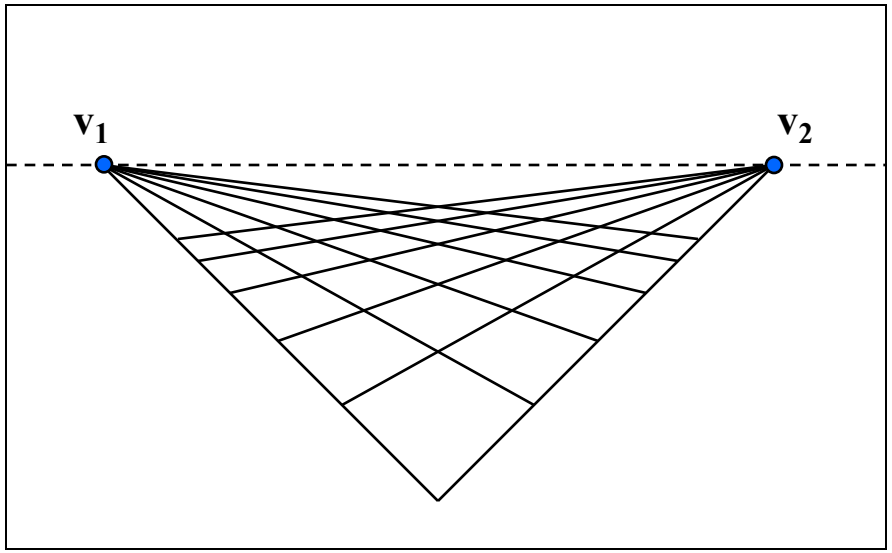


Three-point perspective



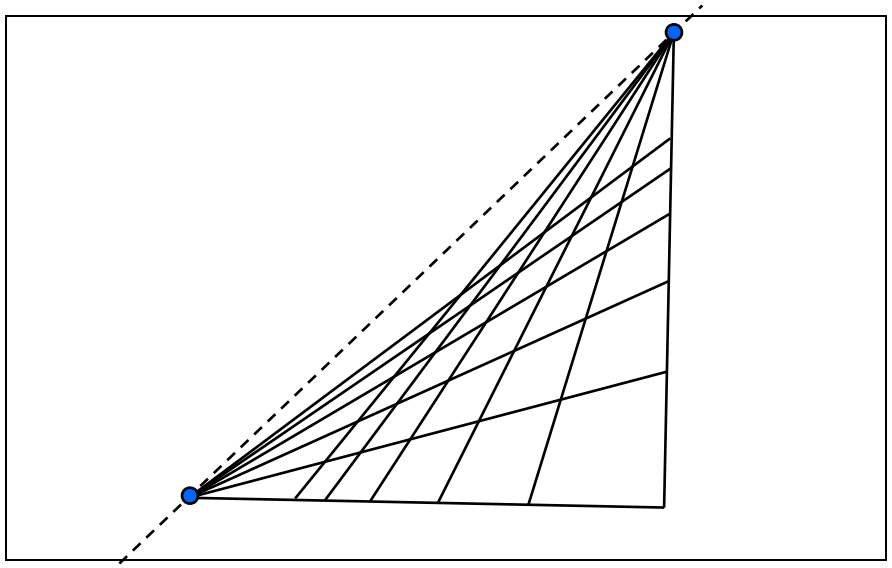
Questions?

Vanishing lines



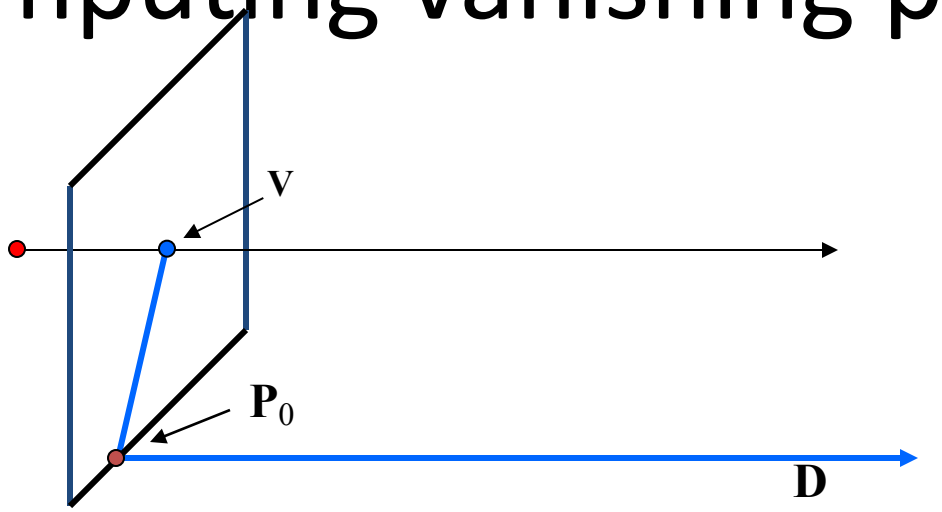
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes (can) define different vanishing lines

Vanishing lines



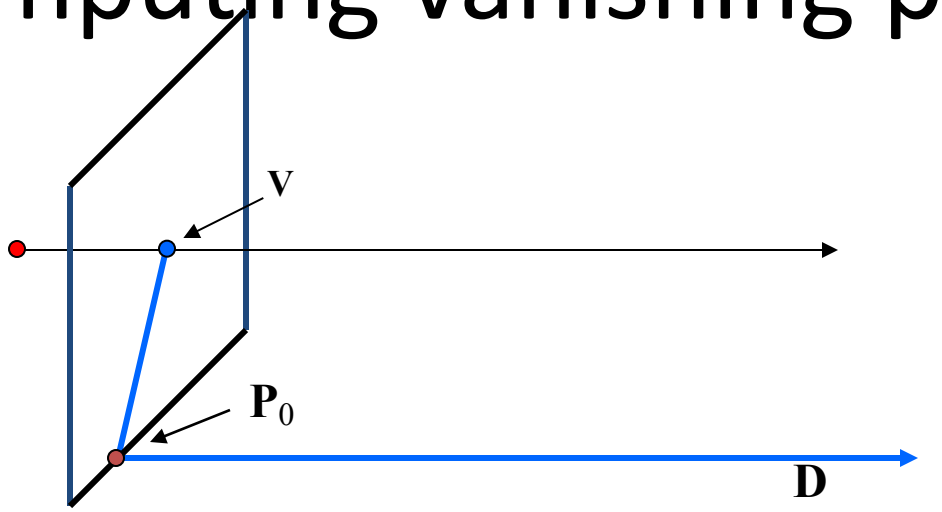
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes (can) define different vanishing lines

Computing vanishing points



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

Computing vanishing points



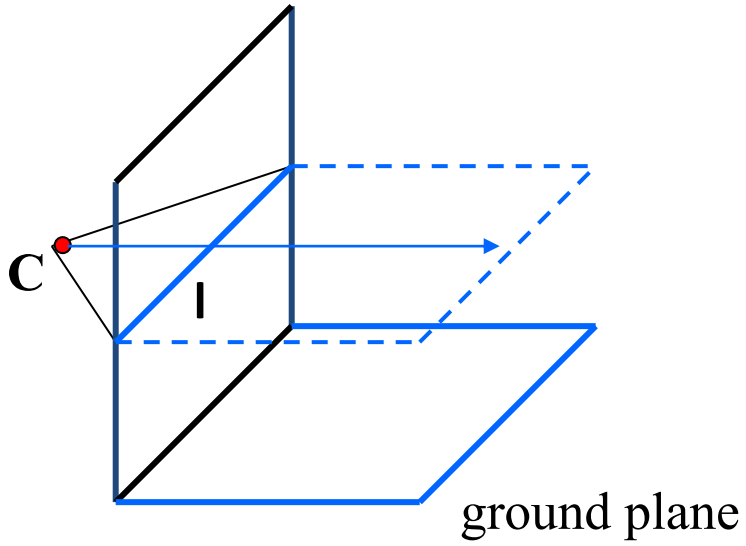
$$\mathbf{P}_t = \mathbf{P}_0 + t\mathbf{D}$$

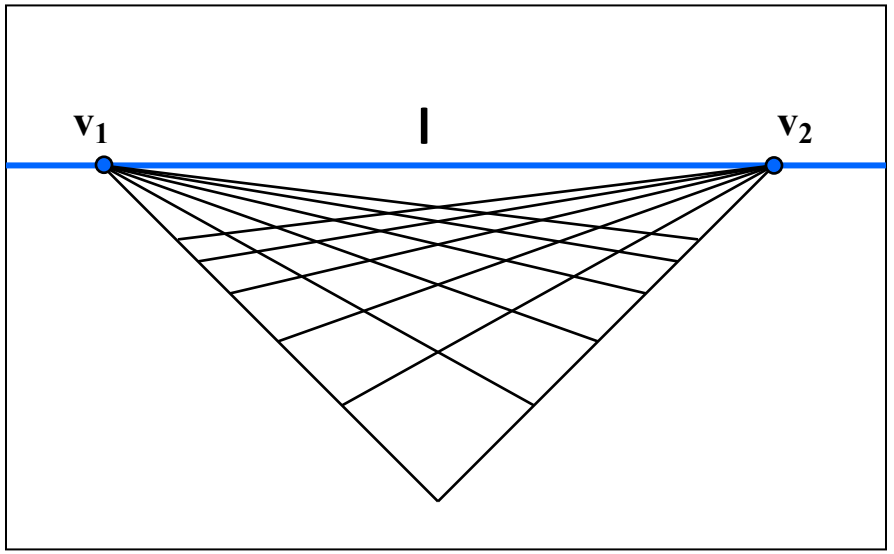
$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X/t + D_X \\ P_Y/t + D_Y \\ P_Z/t + D_Z \\ 1/t \end{bmatrix}$$

Takeaway:
Every directions maps to a vanishing point. Every vanishing point maps to a direction

- Properties $\mathbf{v} = \Pi\mathbf{P}_\infty$
  - $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
  - Depends 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

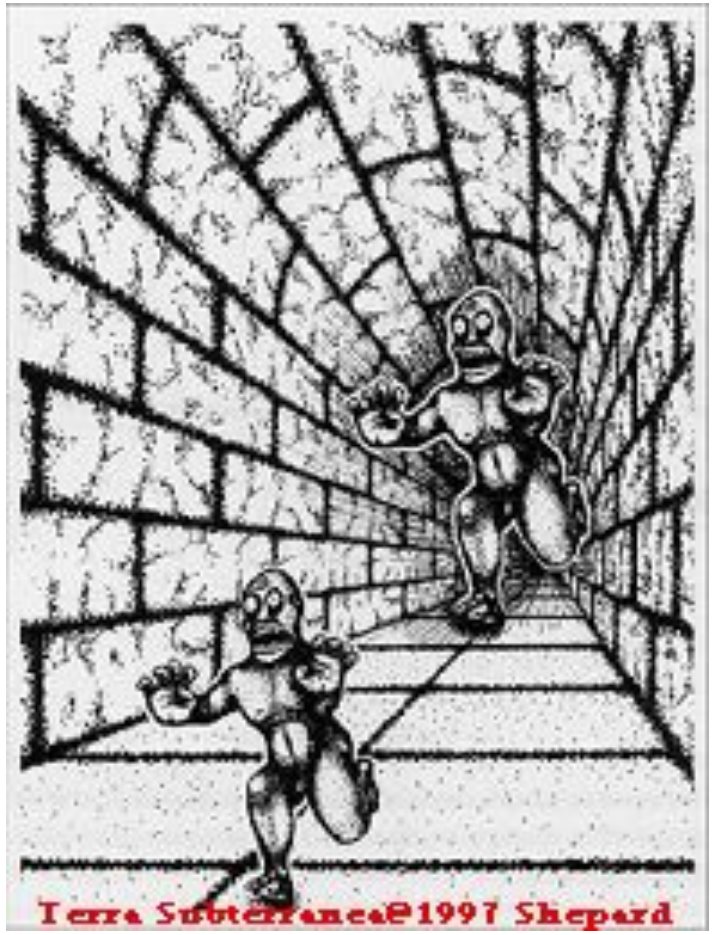


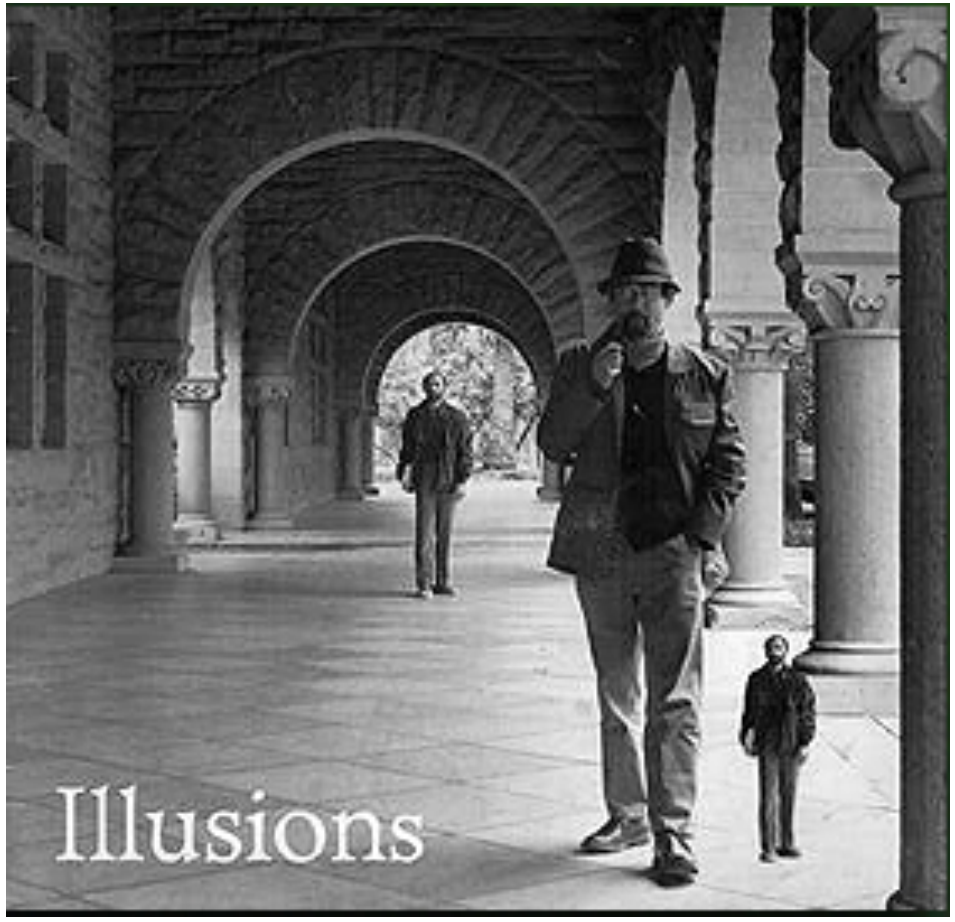


- Properties
  - **l** is intersection of horizontal plane through **C** with image plane
  - Compute **l** from two sets of parallel lines on ground plane
  - All points at same height as **C** project to **l**
    - points higher than C project above l
  - Provides way of comparing height of objects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

Riddle me this…

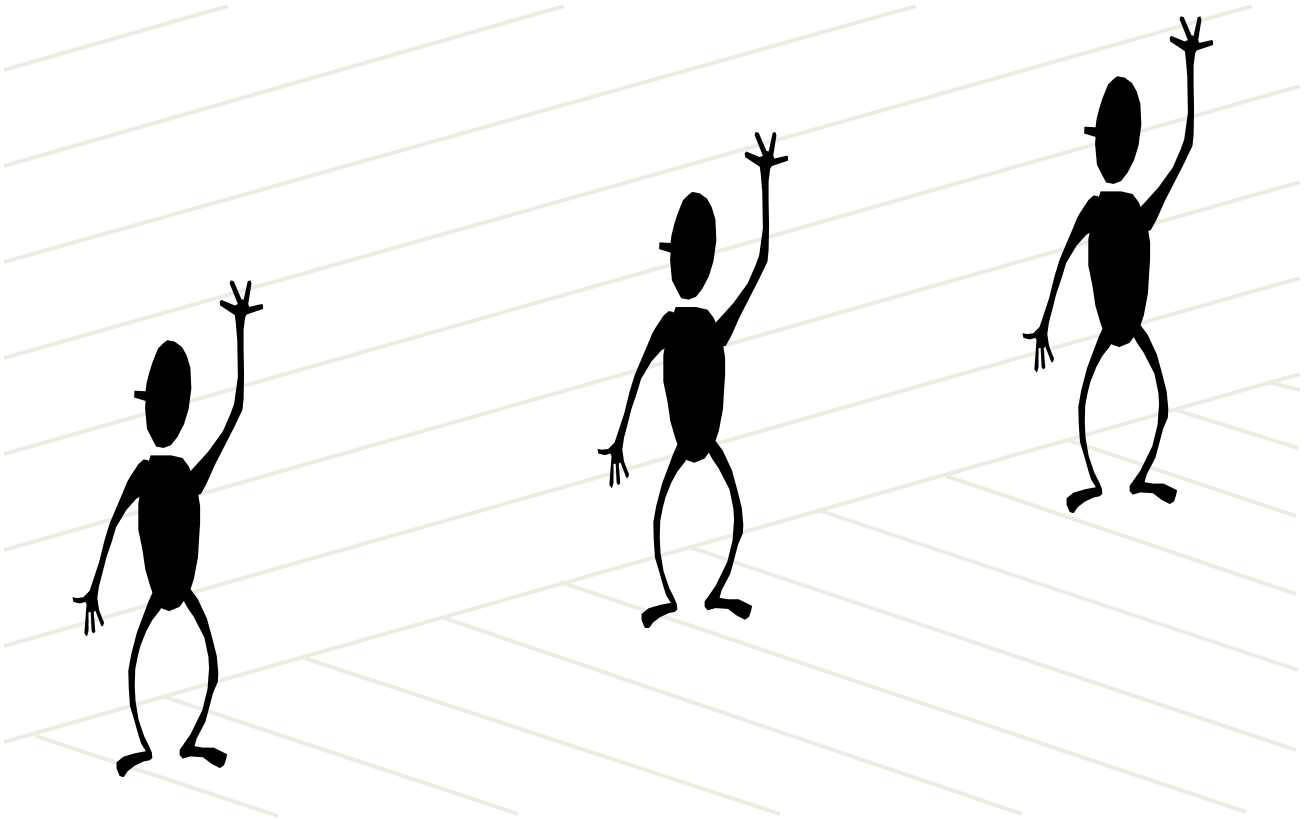
Fun with vanishing points



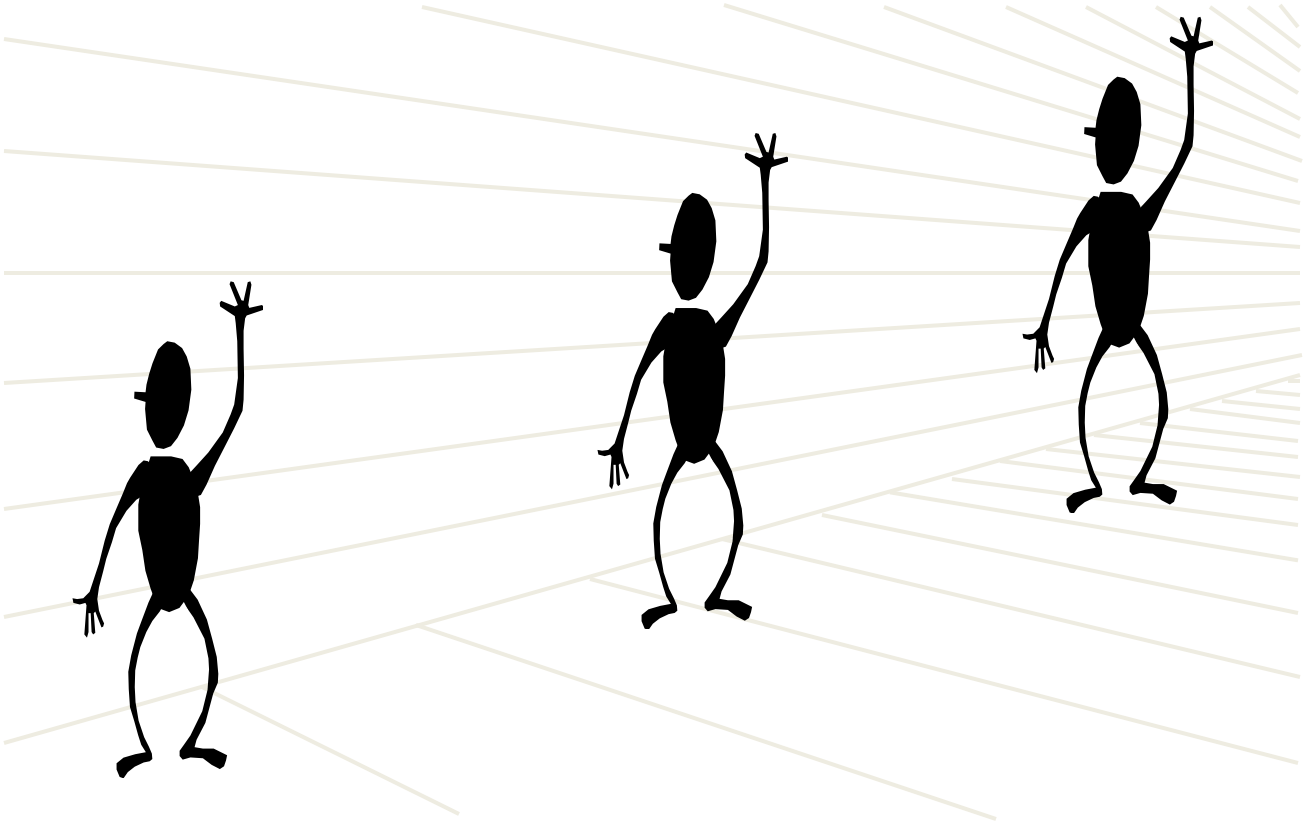


Lots of fun with vanishing points

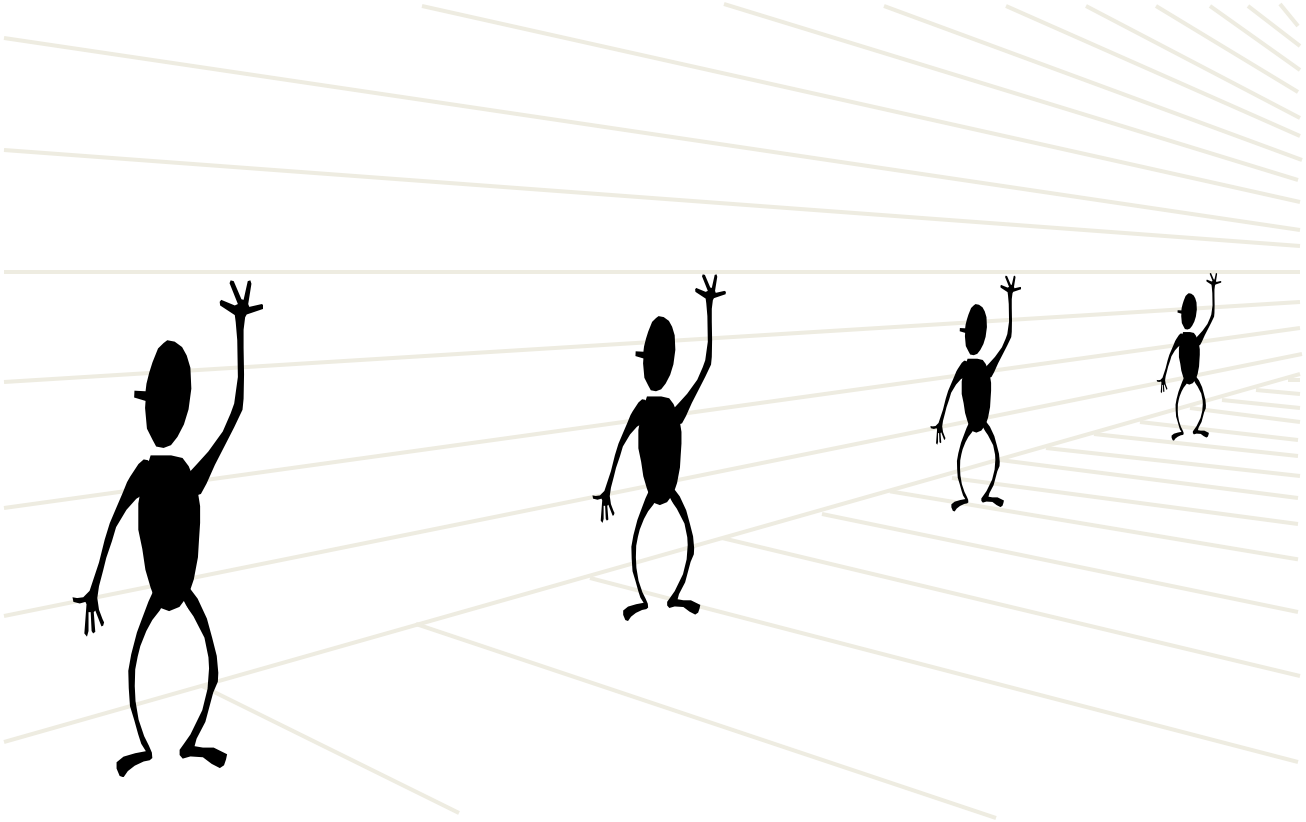
Perspective cues



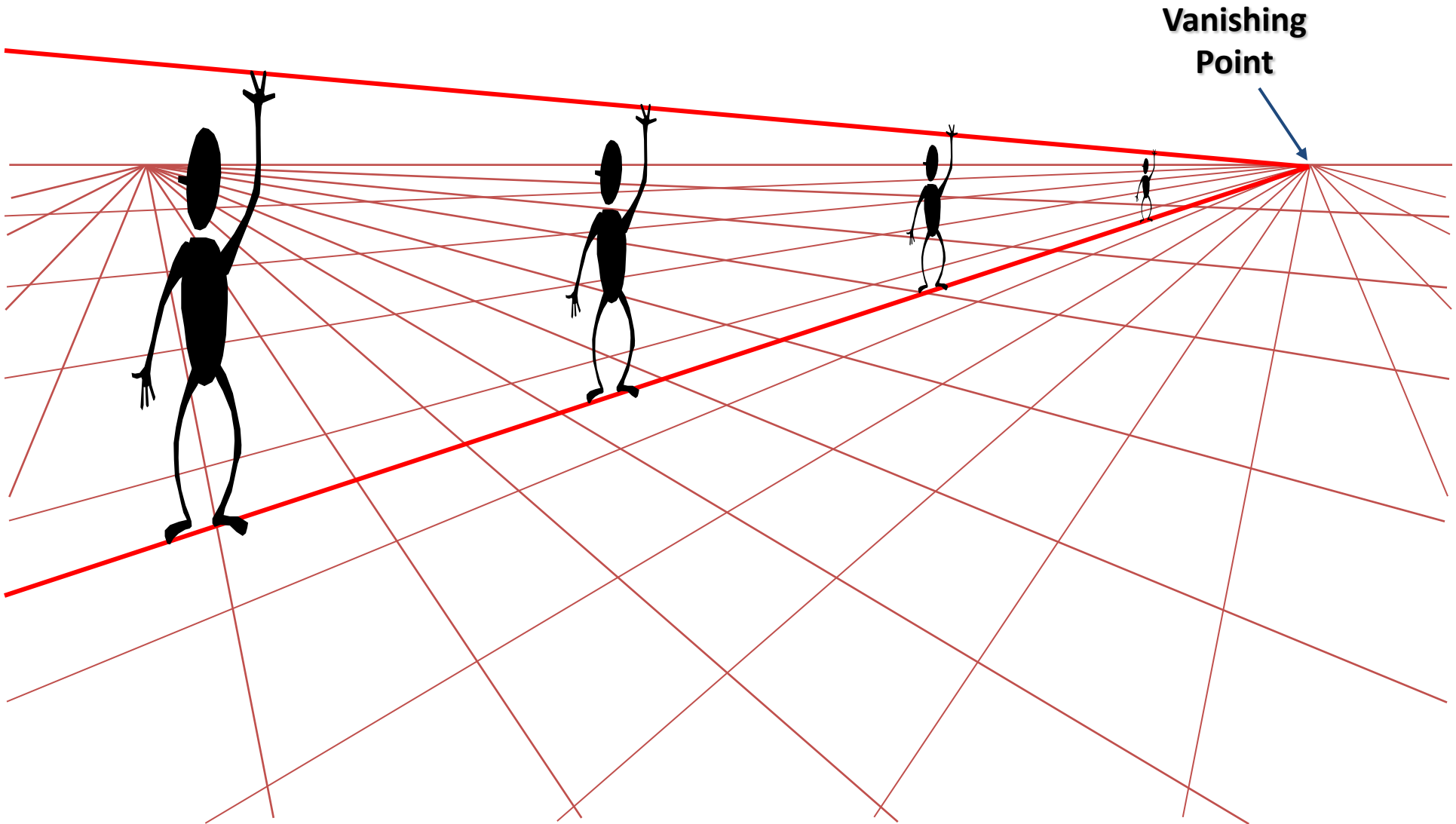
Perspective cues



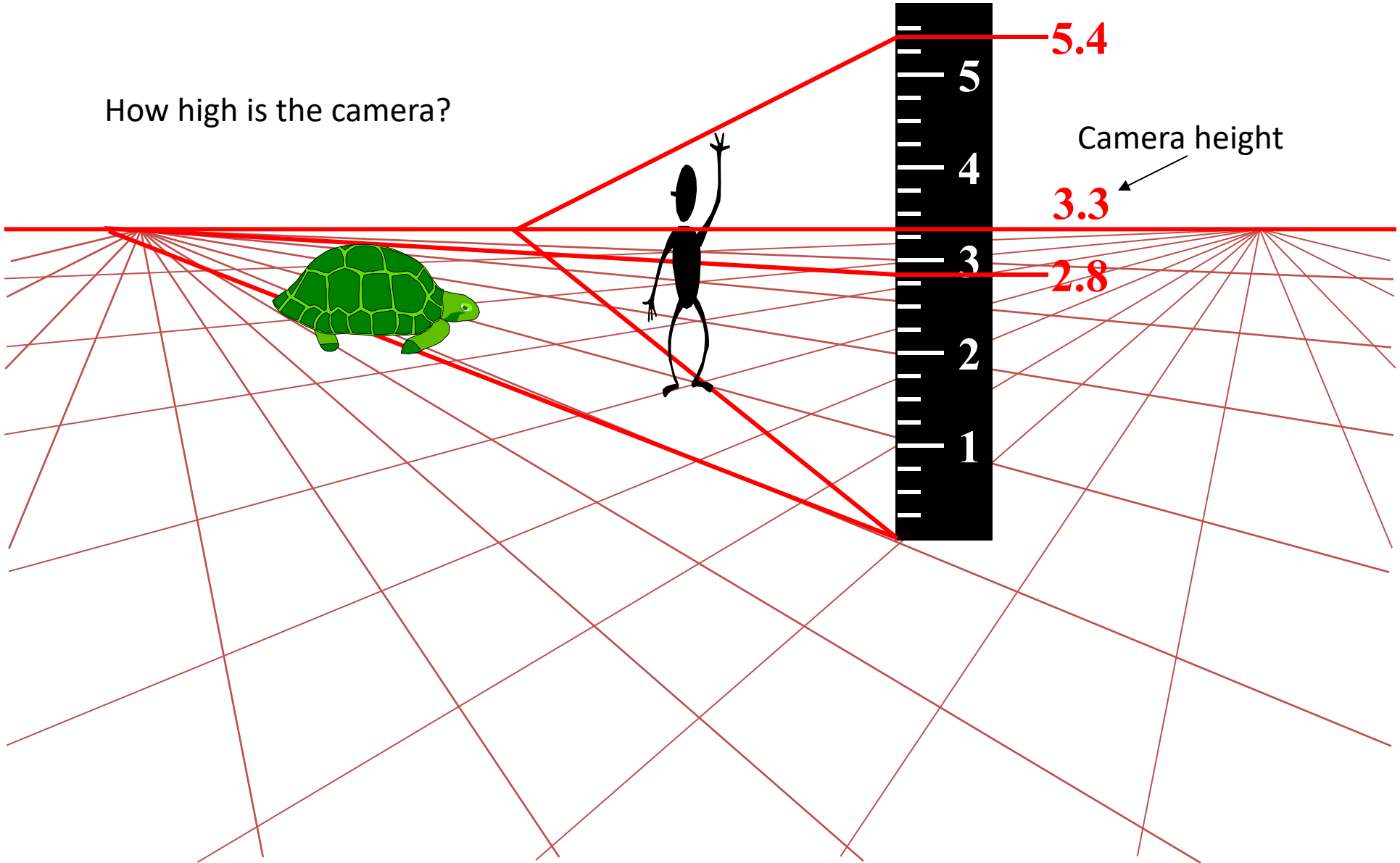
Perspective cues



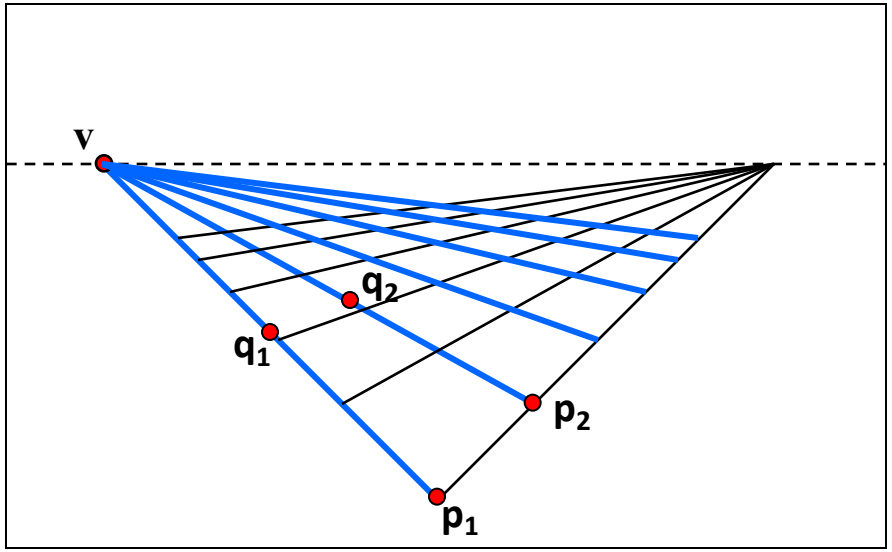
Comparing heights



Measuring height



Computing vanishing points (from lines)



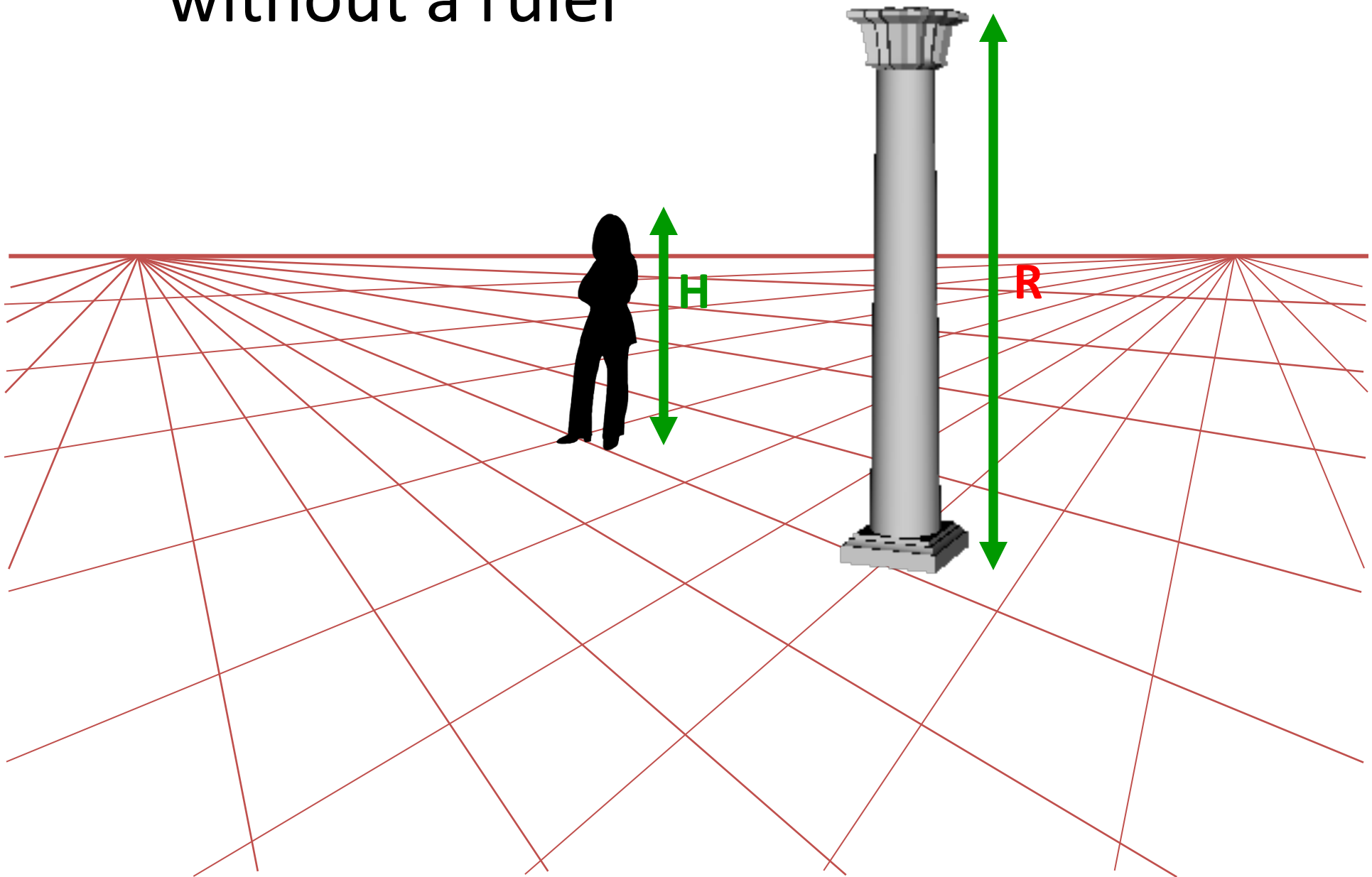
- Intersect p_1q_1 with p_2q_2

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

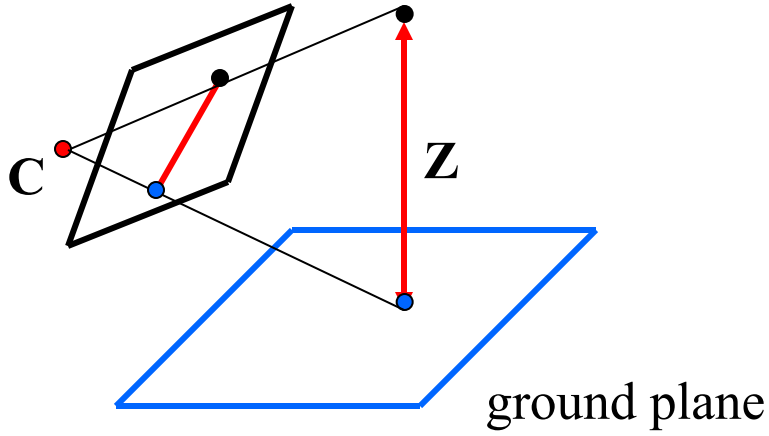
Least squares version

- 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



Measuring height without a ruler



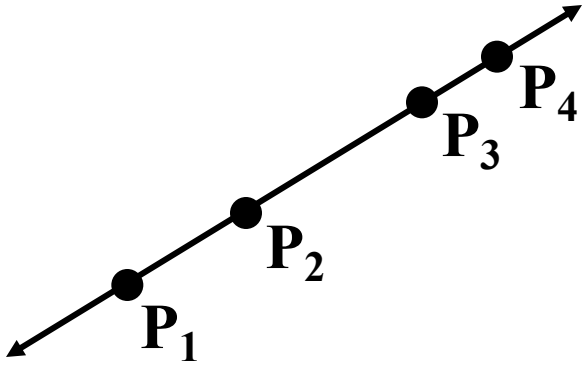
Compute Z from image measurements

- 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



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

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

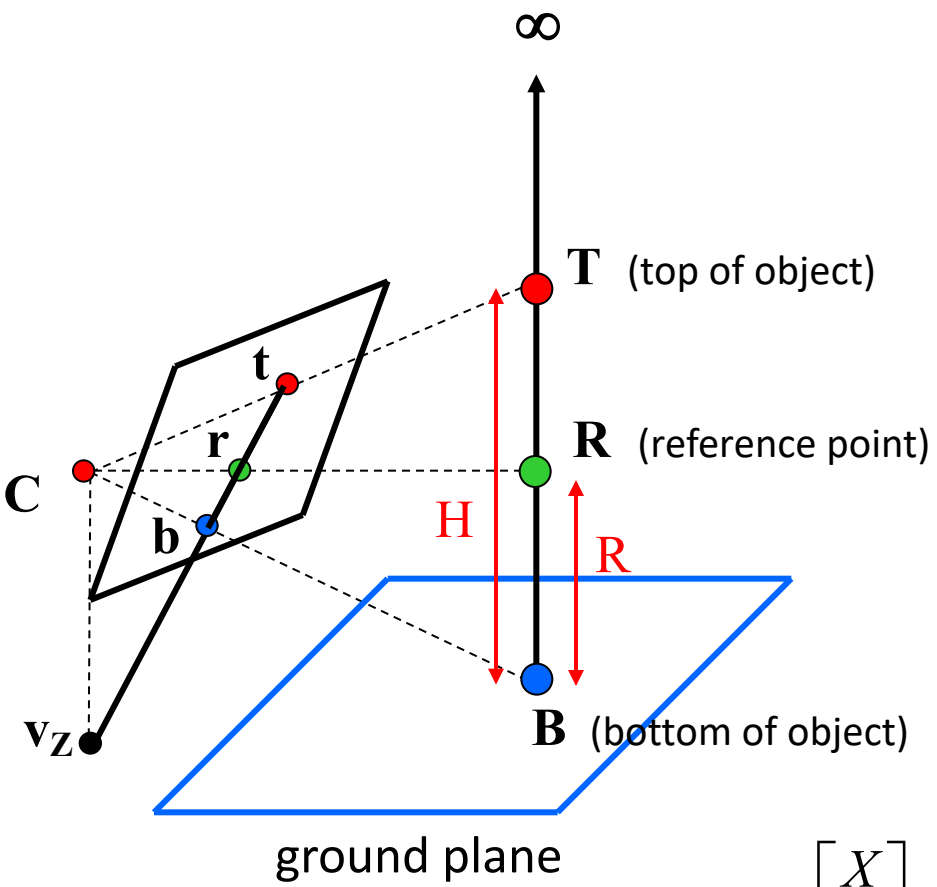
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

Measuring height



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

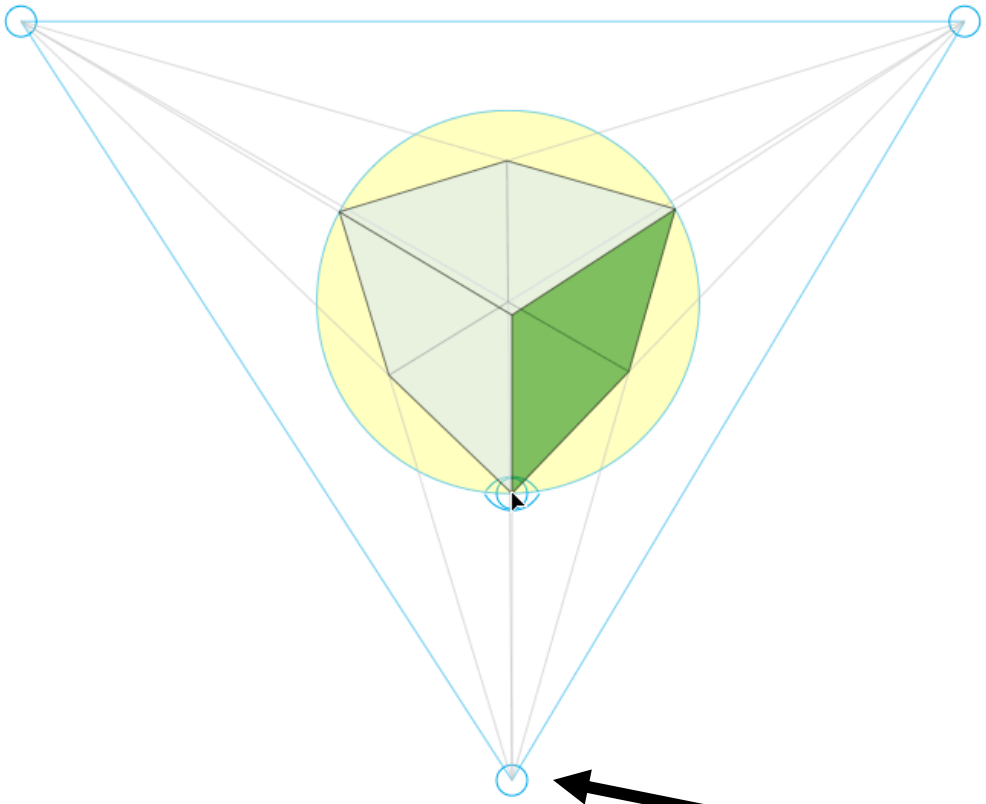
scene cross ratio

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

image cross ratio

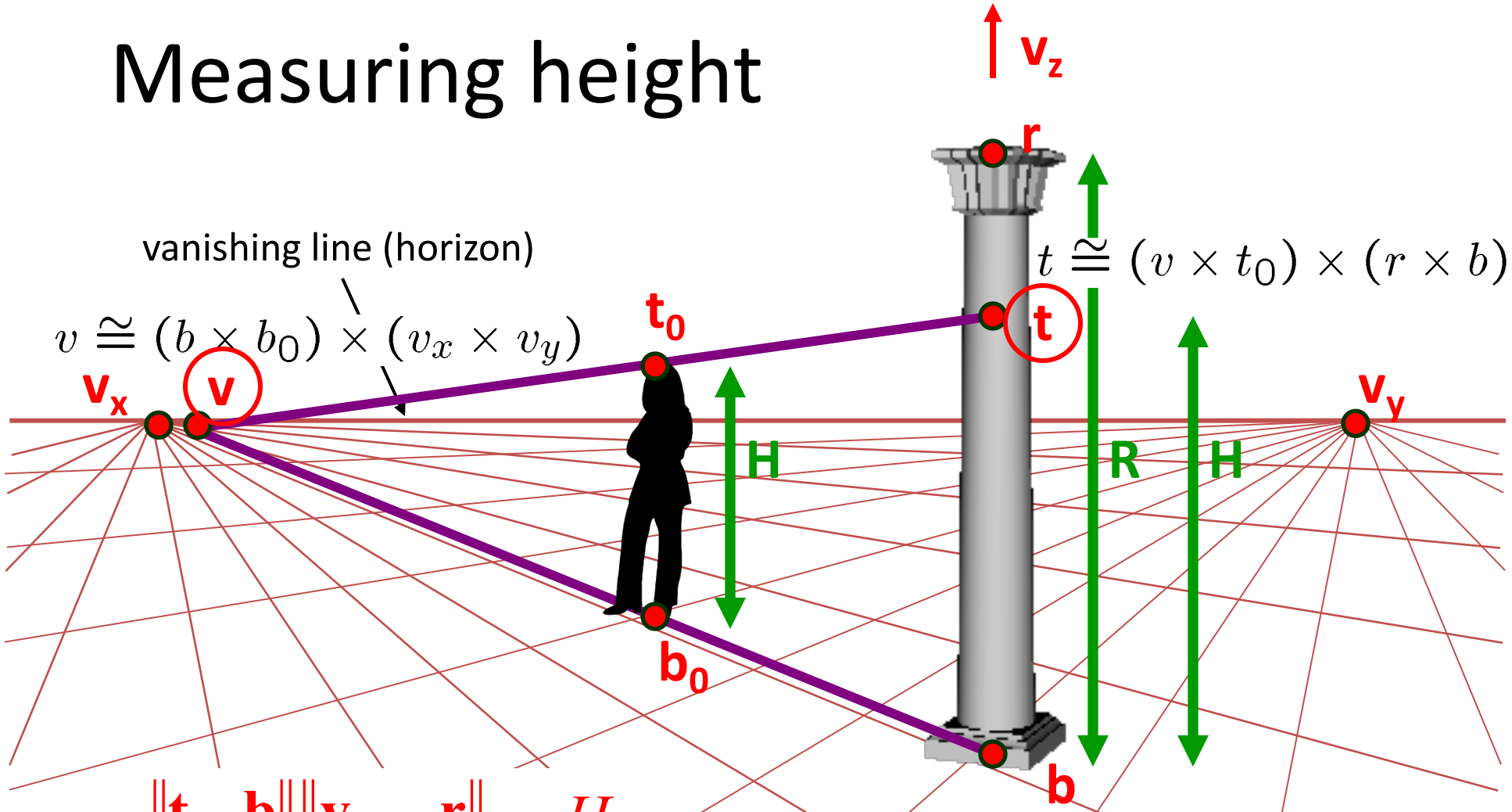
scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Finding the vertical (z) vanishing point



Find intersection of projections of vertical lines

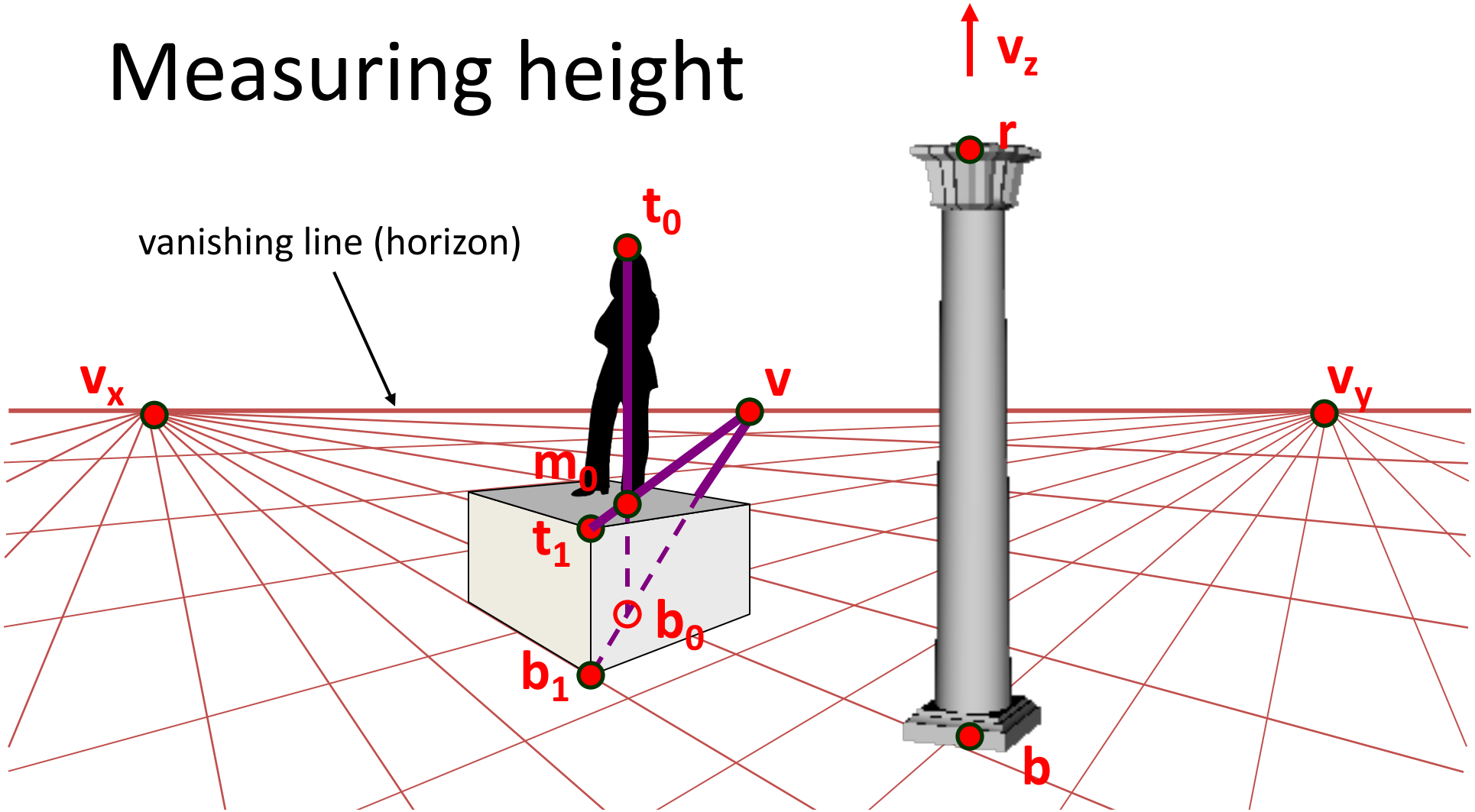
Measuring height



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

image cross ratio

Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the person 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

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

Related problem: camera calibration

- Goal: estimate the camera parameters
  - Version 1: solve for 3x4 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, principal point, pixel size)
    - extrinsics (rotation angles, translation)
    - radial distortion

Vanishing points and projection matrix

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

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

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

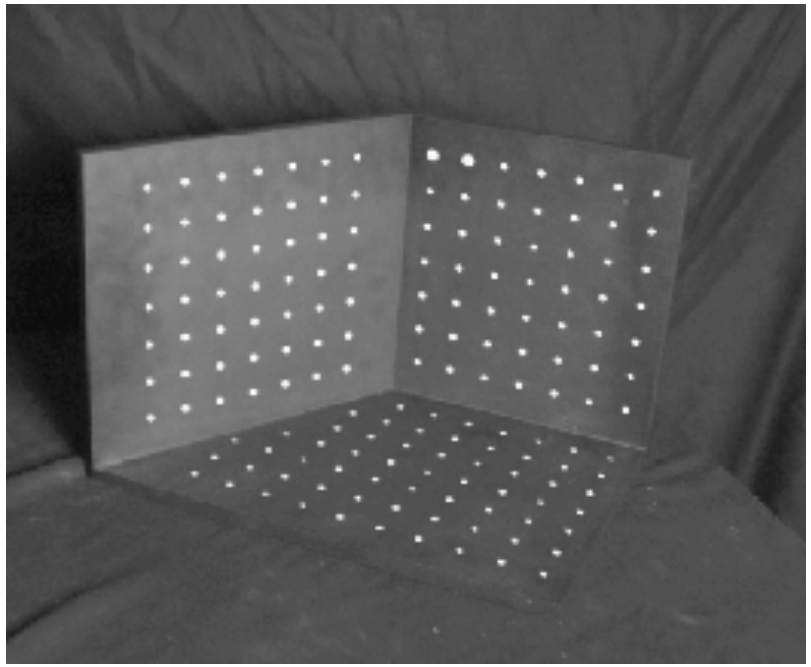
Not So Fast! We only know **v**'s up to a scale factor

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

- Can fully specify by providing 3 additional reference points with known coordinates

Calibration using a reference object

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



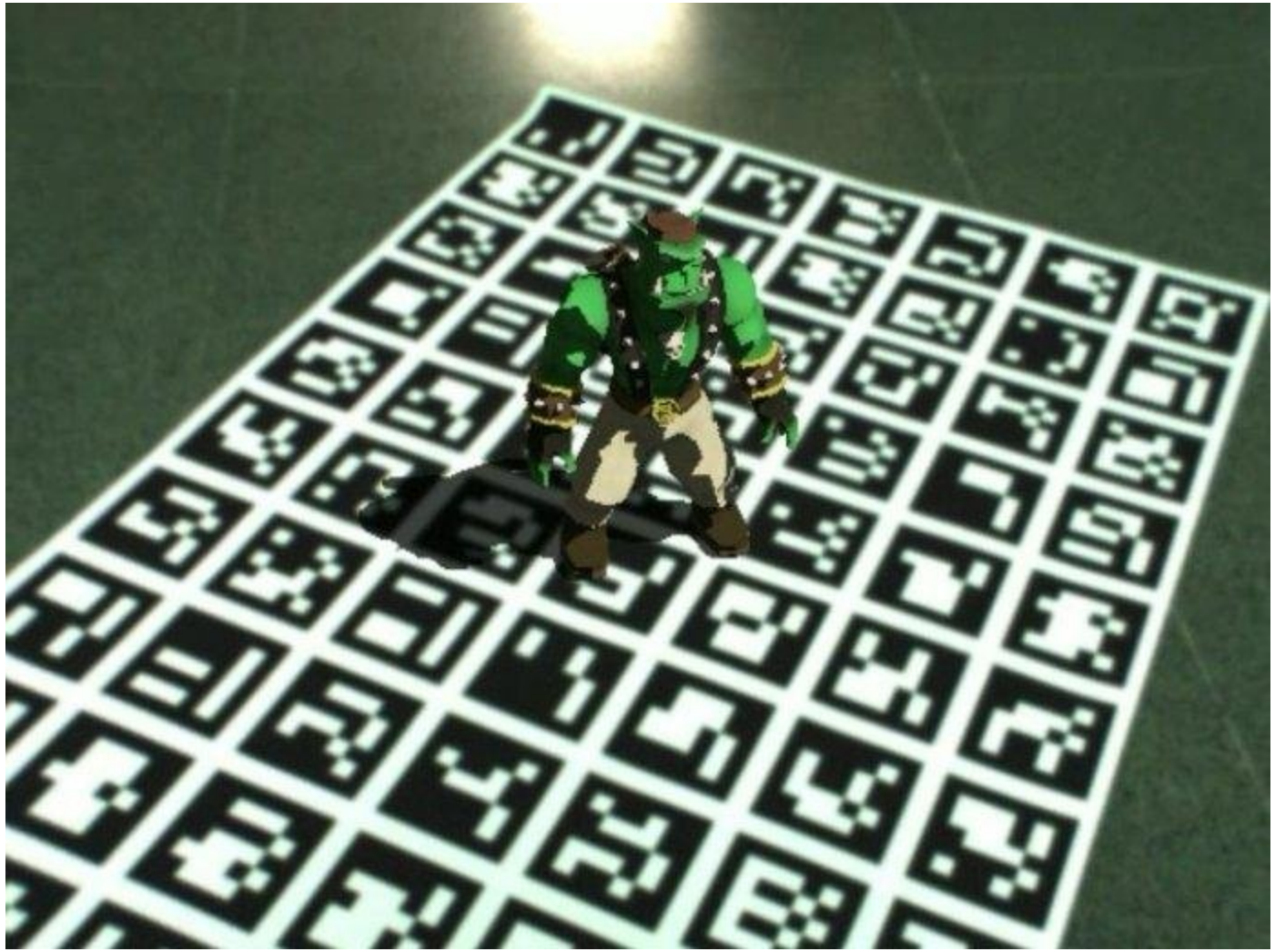
Issues

- 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

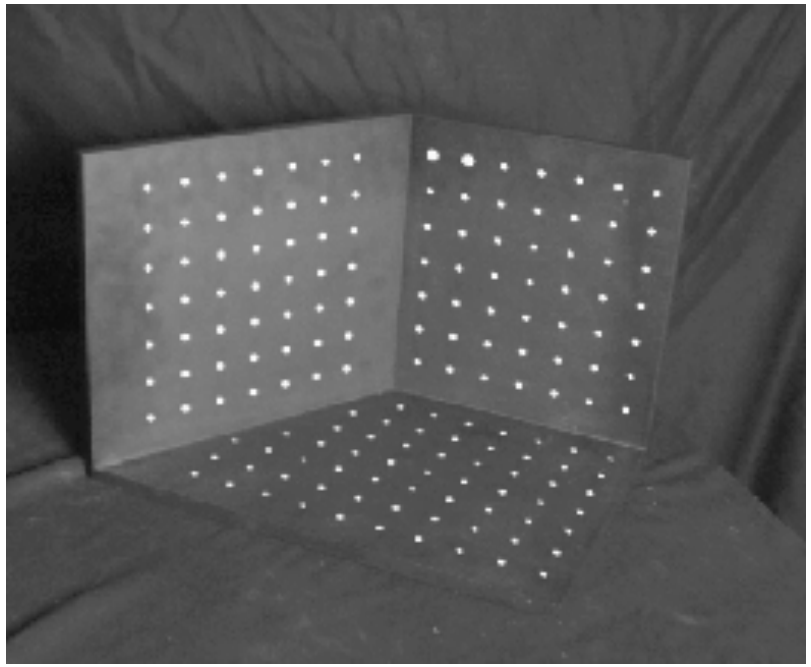
AR codes



ArUco

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

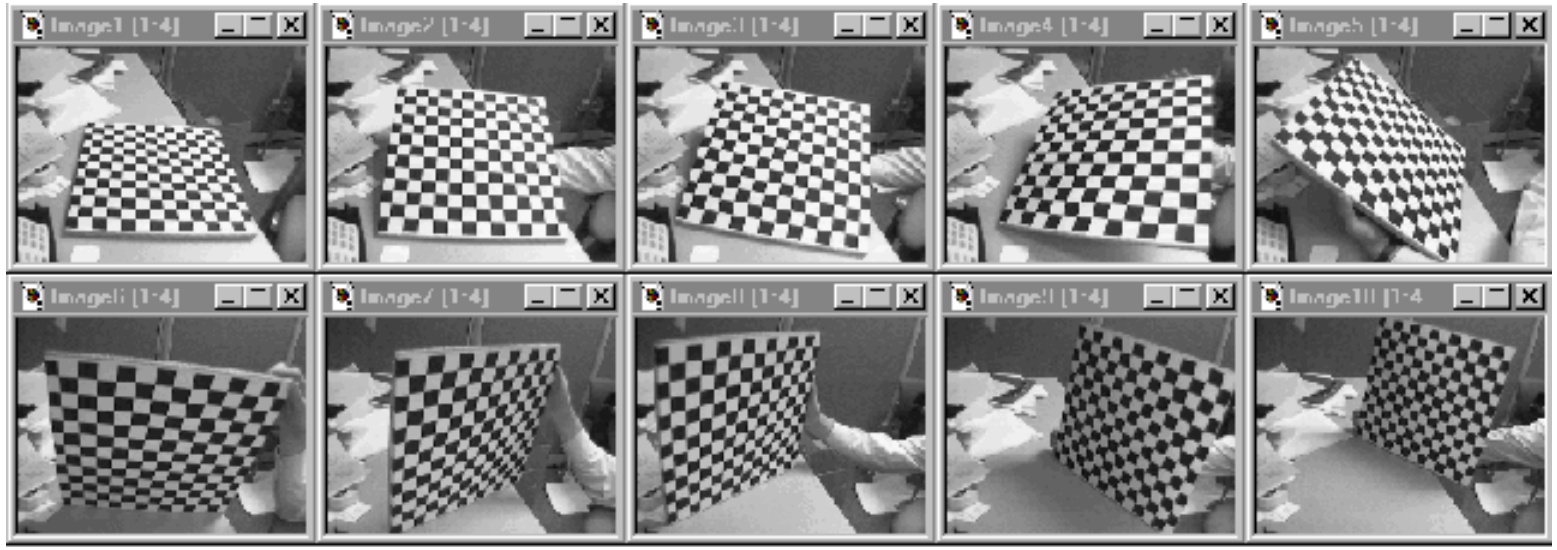
Direct linear calibration

- 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 *f*)
  - 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

Alternative: multi-plane calibration



Images courtesy Jean-Yves Bouguet

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/

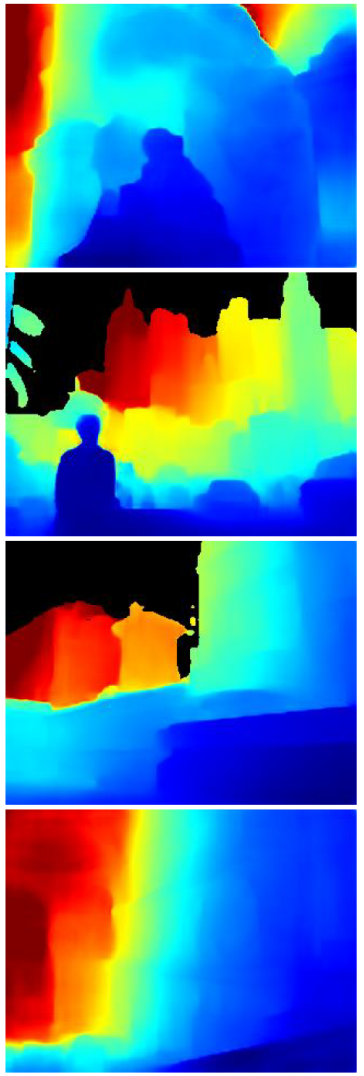
Some Related Techniques

- Image-Based Modeling and Photo Editing
  - Mok et al., SIGGRAPH 2001
  - http://graphics.csail.mit.edu/graphics/ibedit/
- Single View Modeling of Free-Form Scenes
  - Zhang et al., CVPR 2001
  - http://grail.cs.washington.edu/projects/svm/index.htm
- Tour Into The Picture
  - Anjyo et al., SIGGRAPH 1997
  - http://www.mizuno.org/gl/tip/

Single-image depth prediction

Image

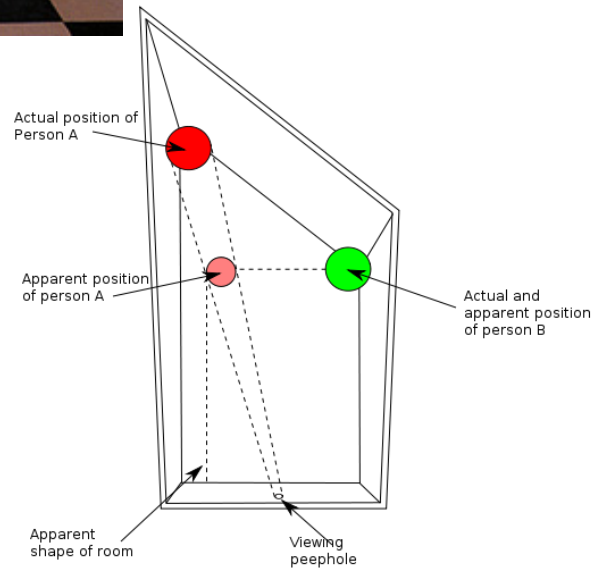
Deep learning



Depth map

Using Classic Vision to Understand Modern Learning

- Vision is under-constrained
  - Many explanations for the same image
  - Some prior assumptions always necessary
- Deep learning is a tool for developing data-driven priors
- Understanding the geometry of image formation guides this process
  - Designing learnable representations
  - Designing deep architectures
  - Curating data



Questions?