Single-view metrology

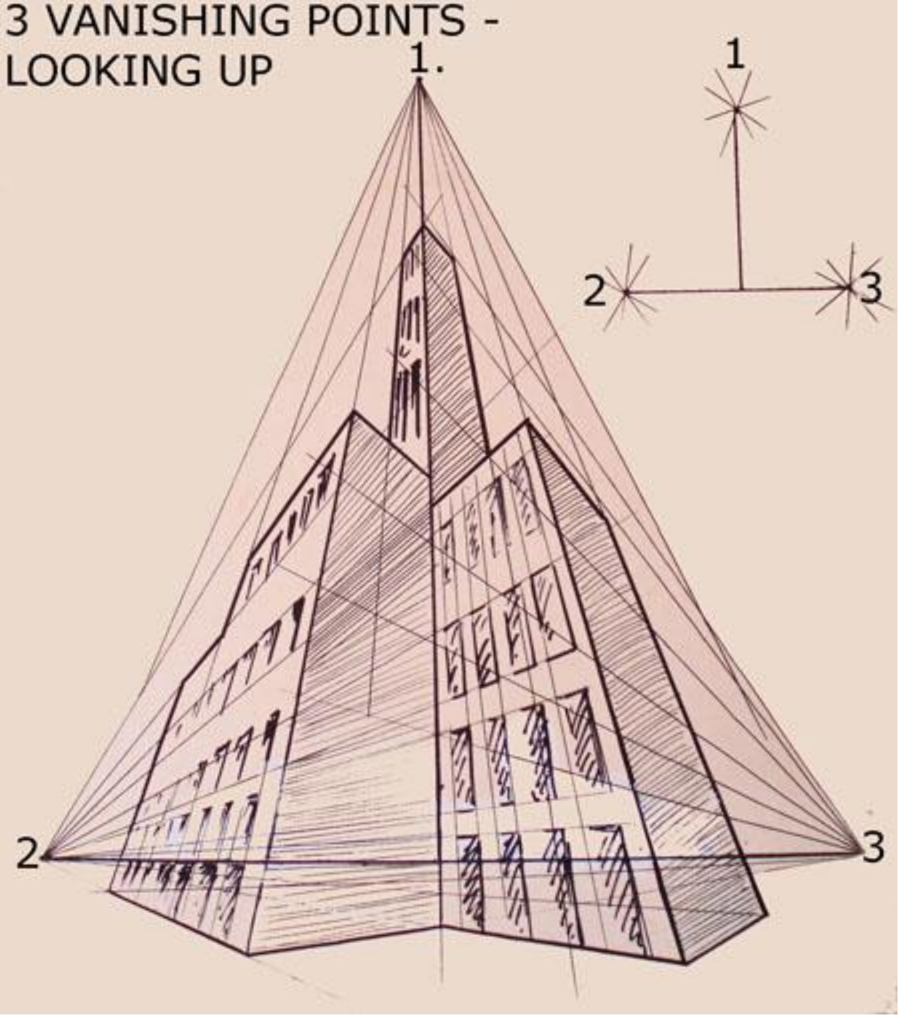
Lihi Zelnik-Manor, Computer Vision

Projective geometry

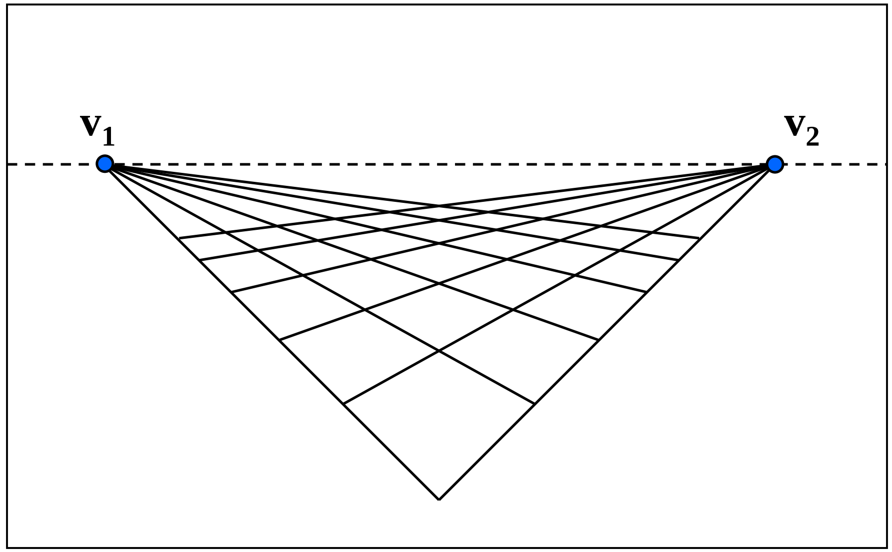
Ames Room

- 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

Three point perspective

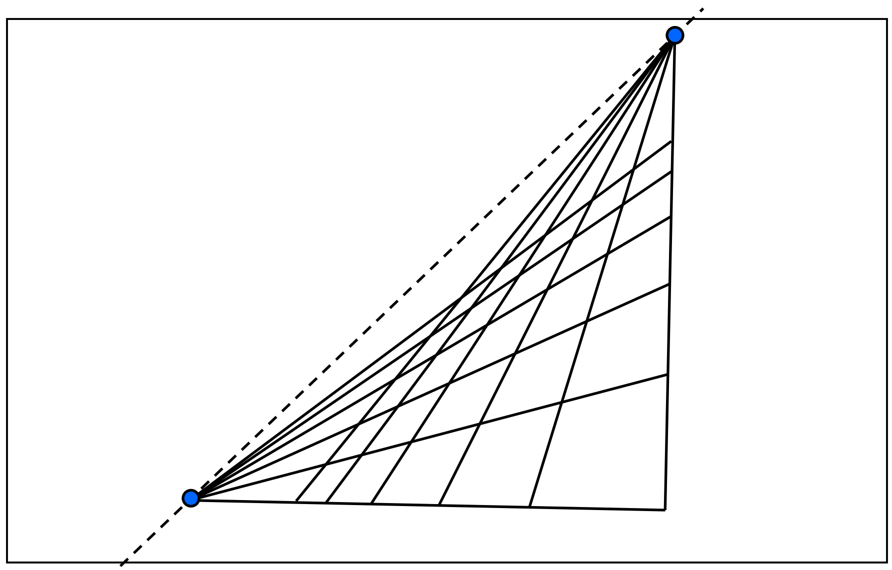


Vanishing lines



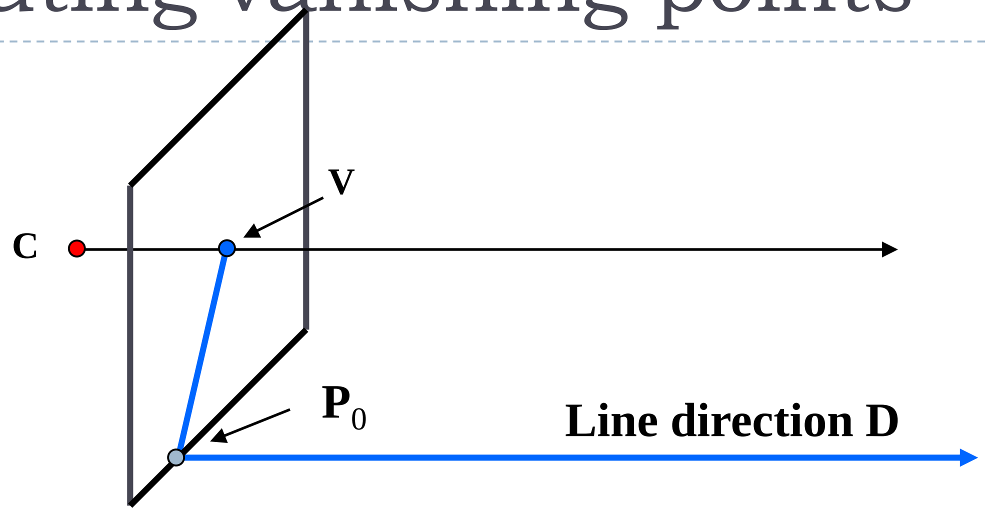
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes (can) define different vanishing lines

Vanishing lines



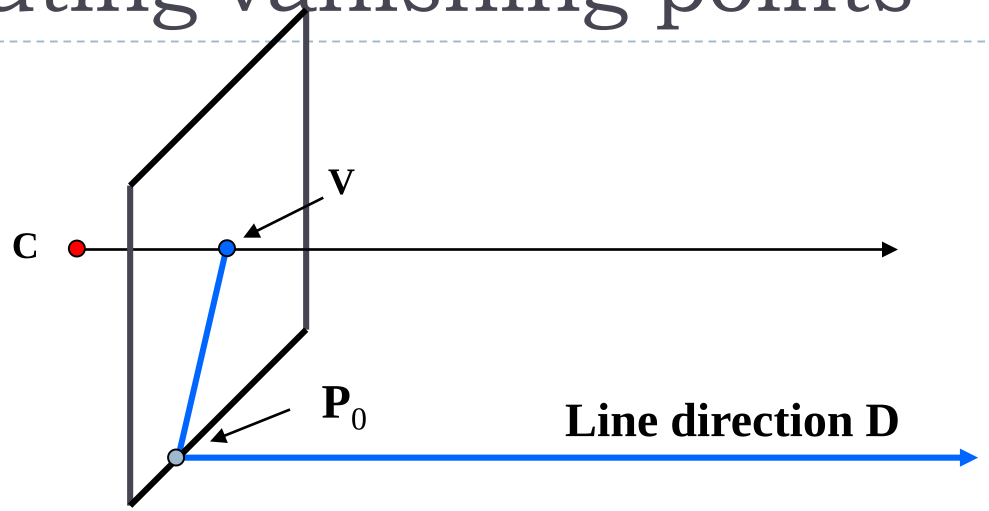
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes (can) define different vanishing lines

Computing vanishing points



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

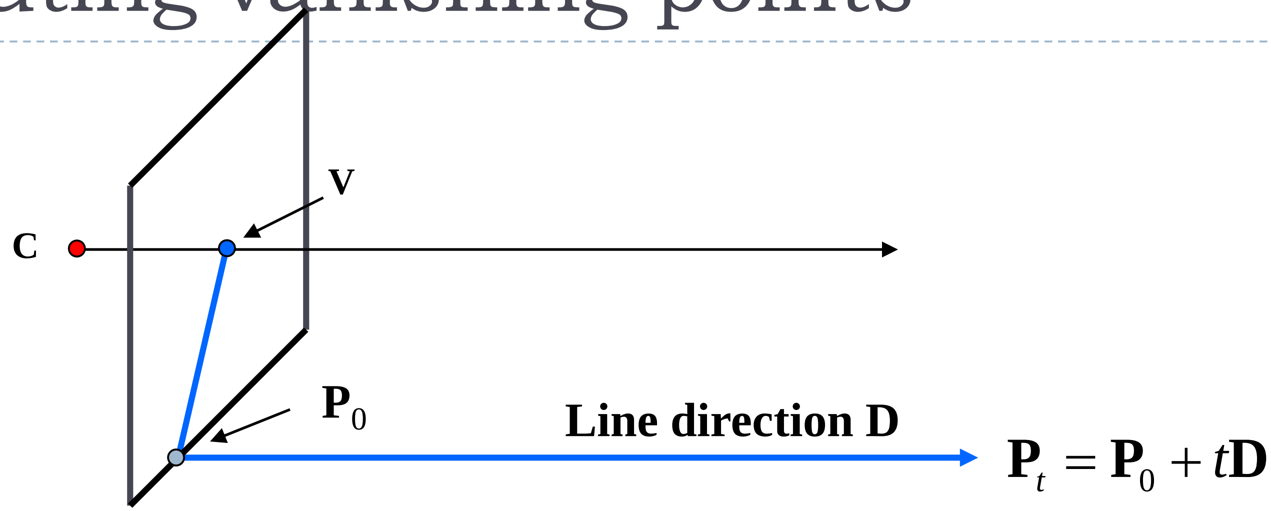
Computing vanishing points



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

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

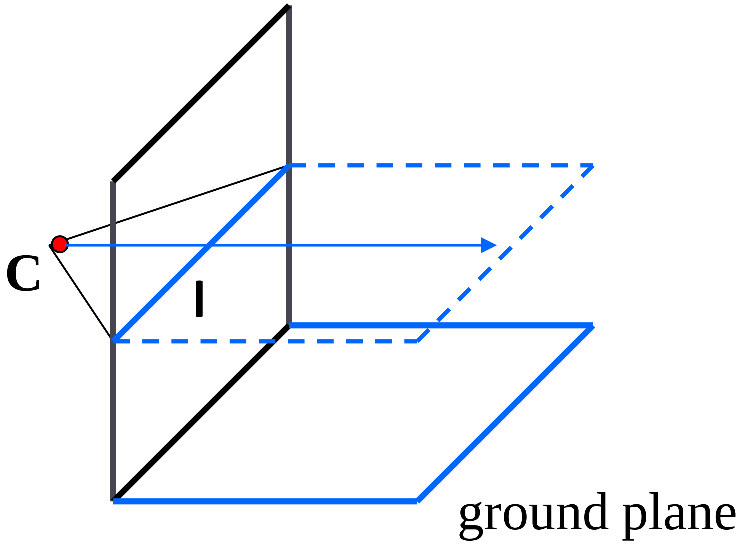
Computing vanishing points

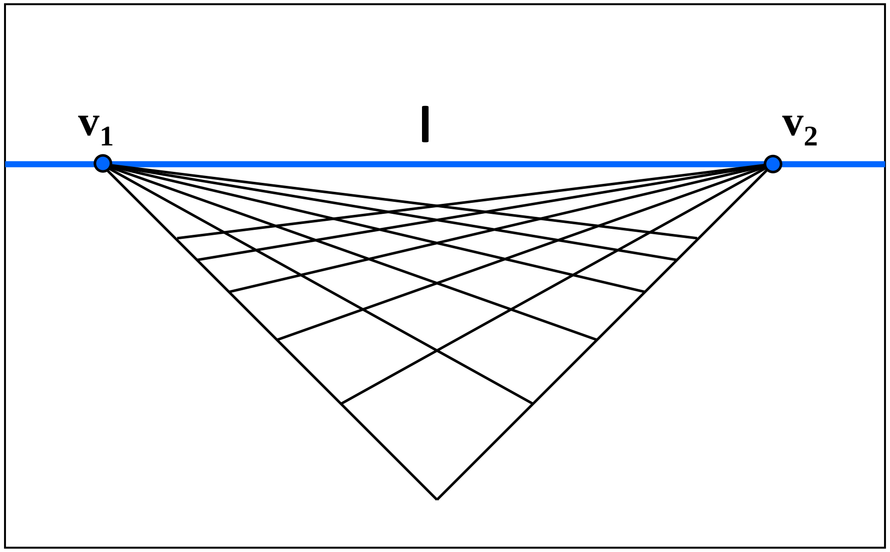


$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

- Properties of $\mathbf{v} = M\mathbf{P}_\infty$
  - $\mathbf{P}_\infty$ is a point at *infinity* where the parallel lines meet, $\mathbf{v}$ is its projection
  - Depends only on line *direction* $\mathbf{D}$
  - Parallel lines $\mathbf{P}_0 + t\mathbf{D}$, $\mathbf{P}_1 + t\mathbf{D}$ intersect at $\mathbf{P}_\infty$

Computing vanishing lines

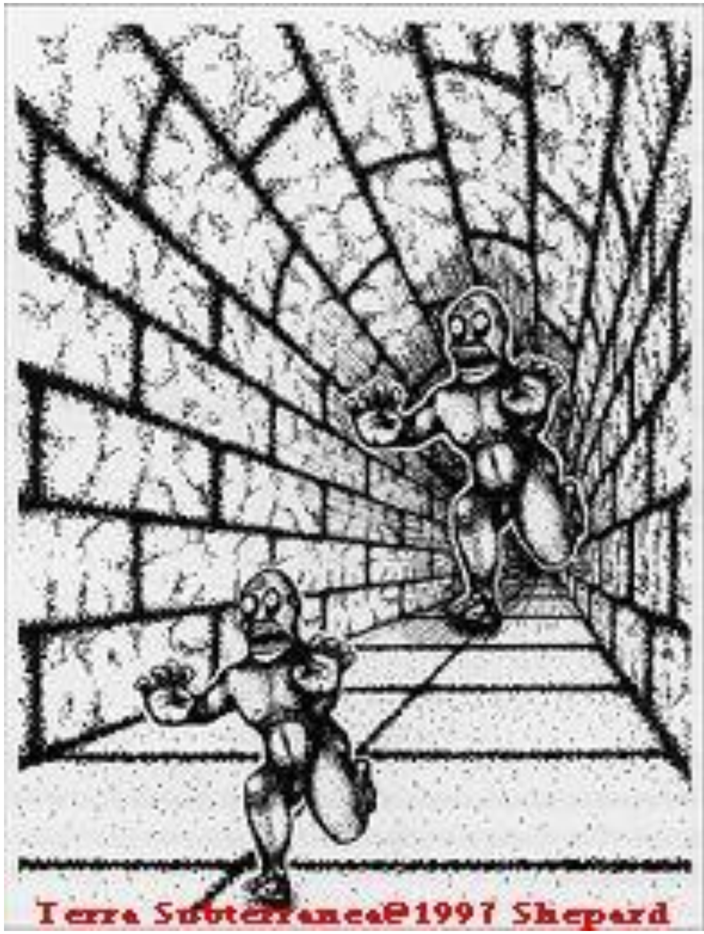


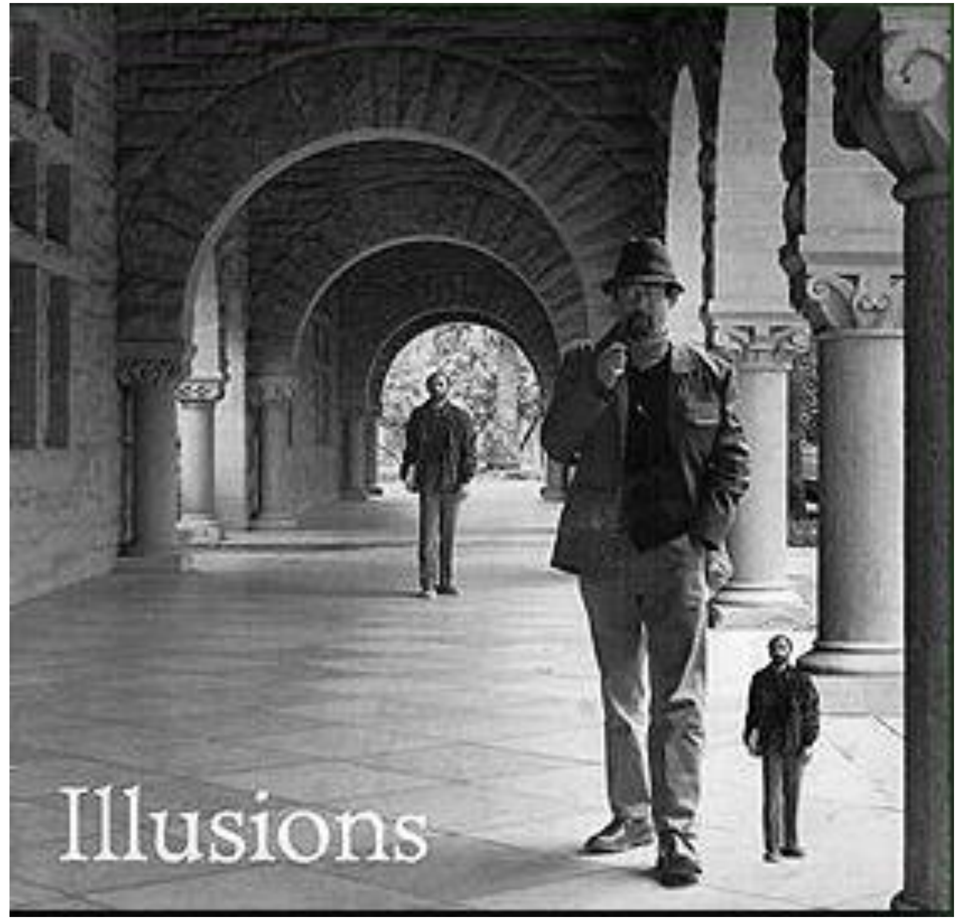


- Properties
  - **l** is intersection of horizontal plane through **C** with image plane
  - Compute **l** from two sets of parallel lines on ground plane (more on that later)
  - All points at same height as **C** project to **l**
    - points higher than C project above l
  - Provides way of comparing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

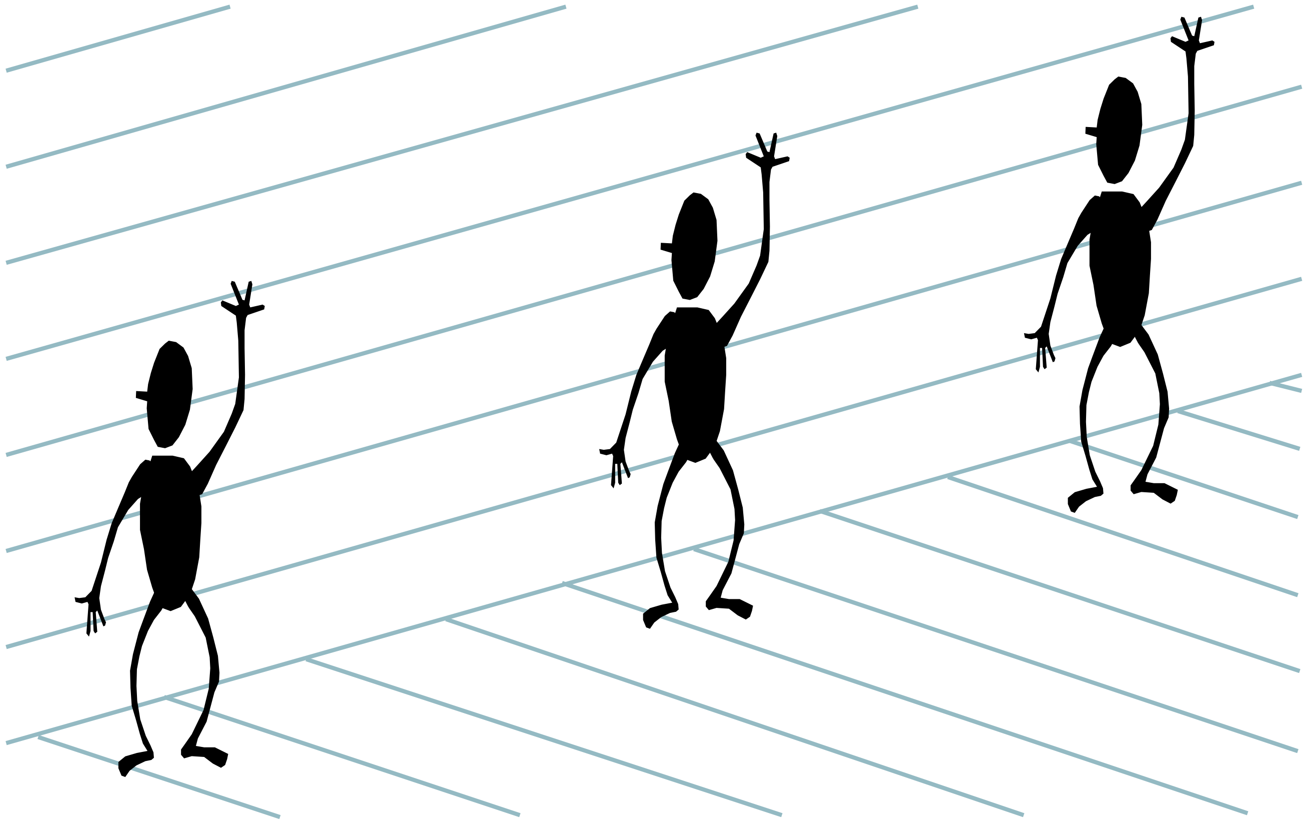
Which is higher – the camera or the man in the parachute?

Fun with vanishing points

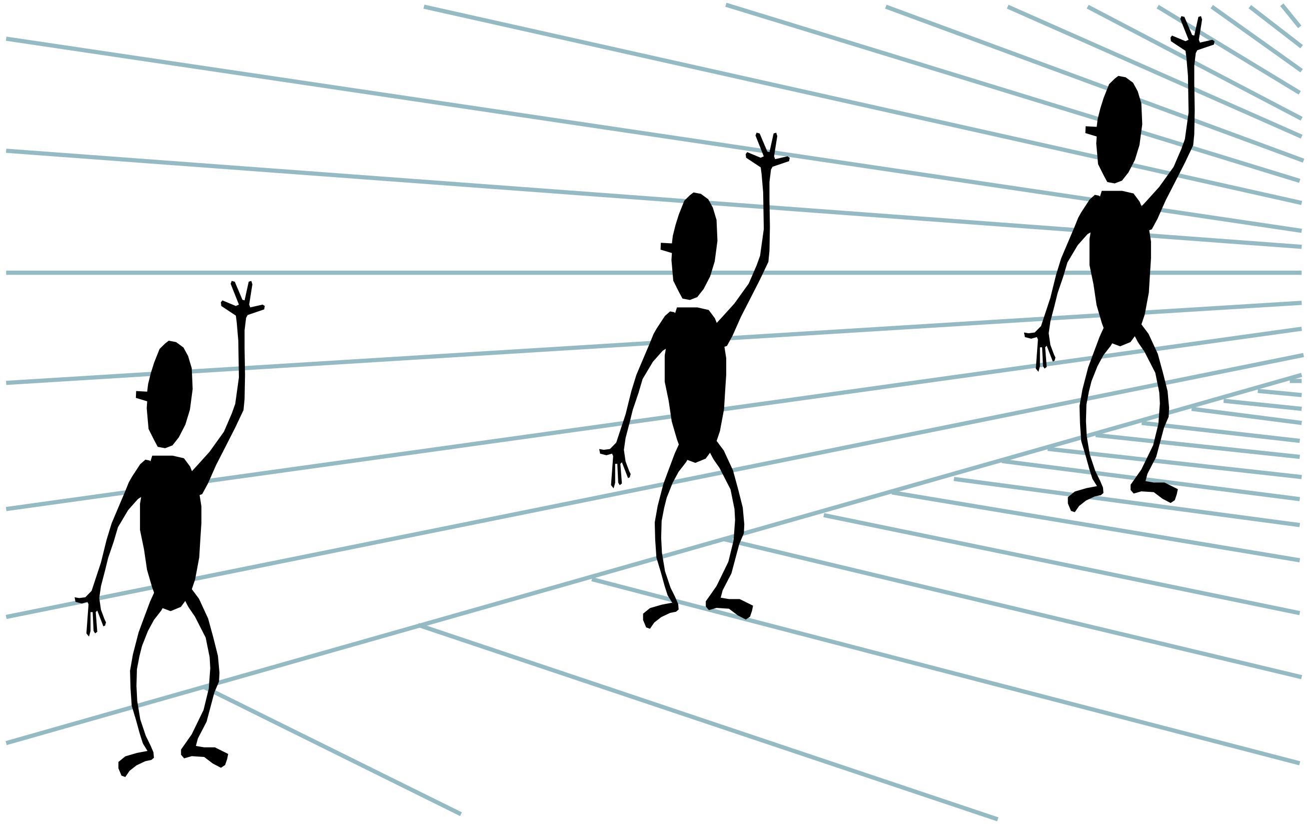




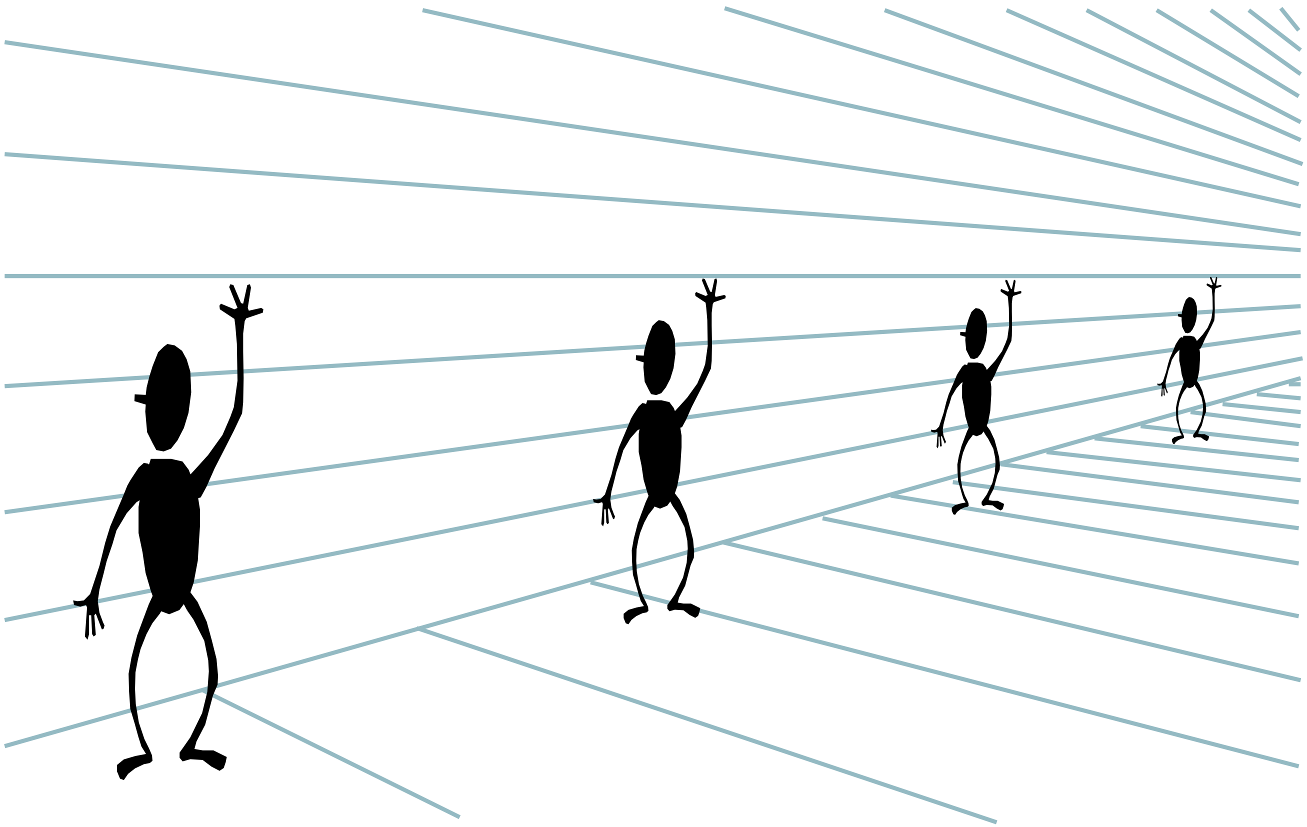
Perspective cues



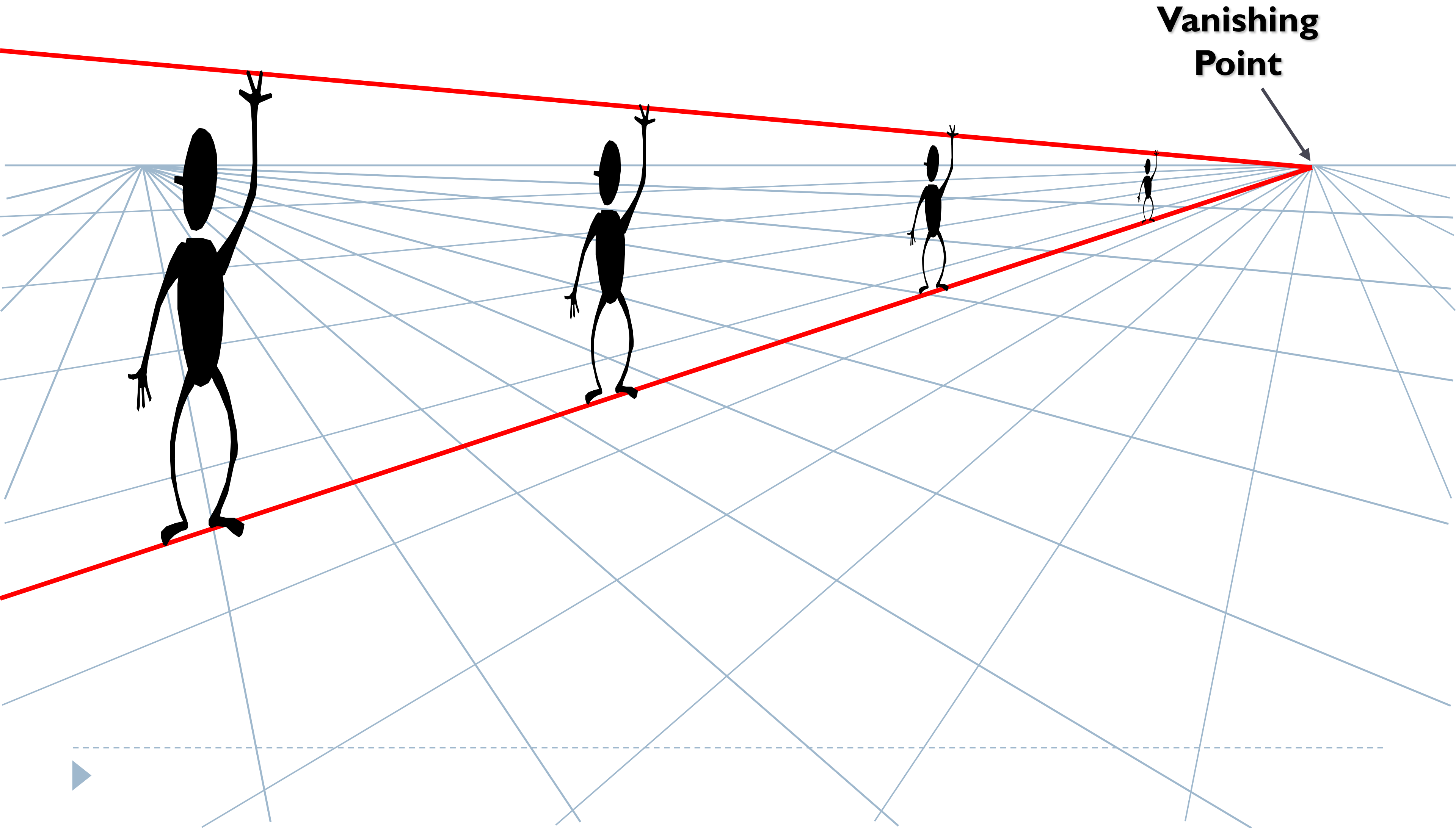
Perspective cues



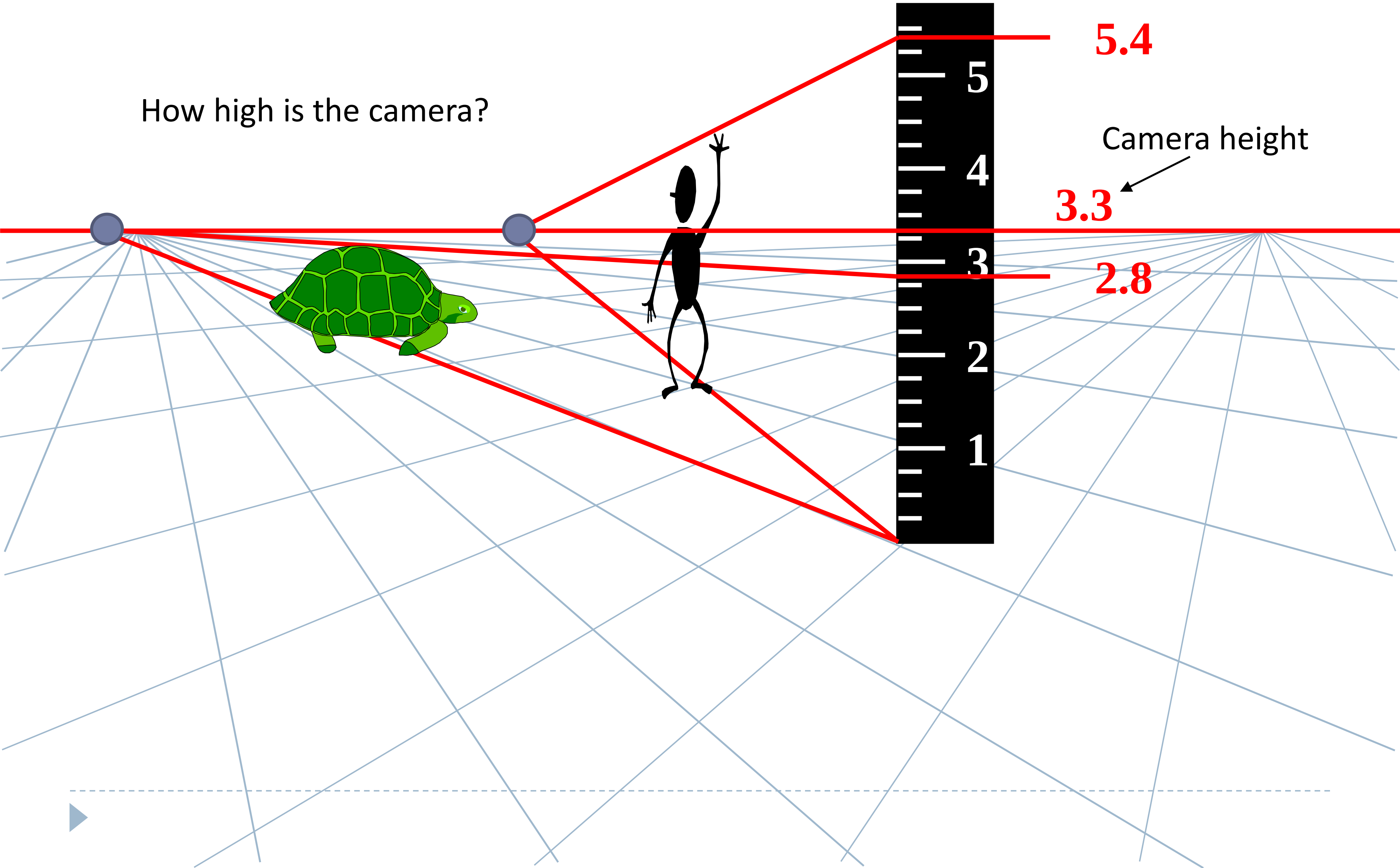
Perspective cues



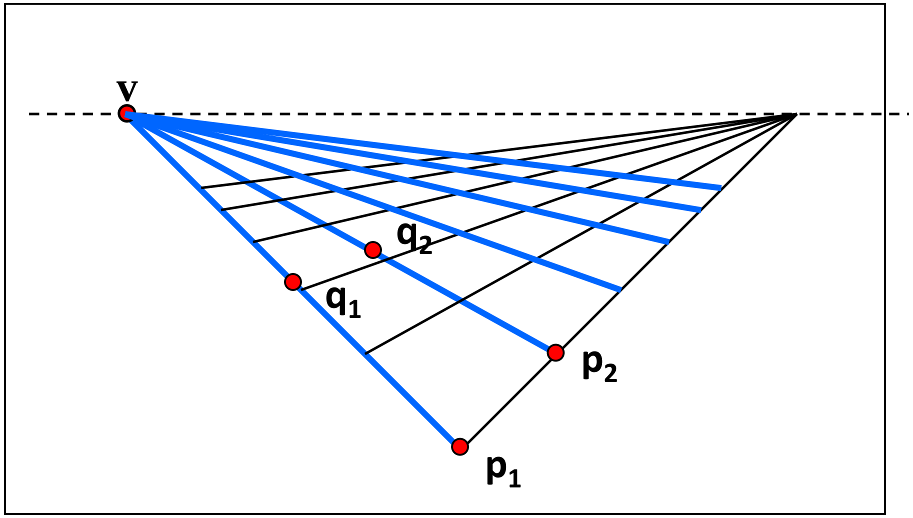
Comparing heights



Measuring height



Computing vanishing points (from lines)



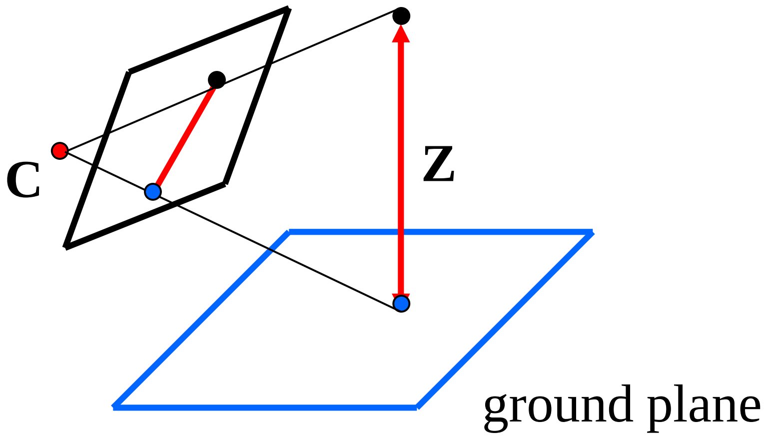
- Intersect p_1q_1 with p_2q_2

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

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

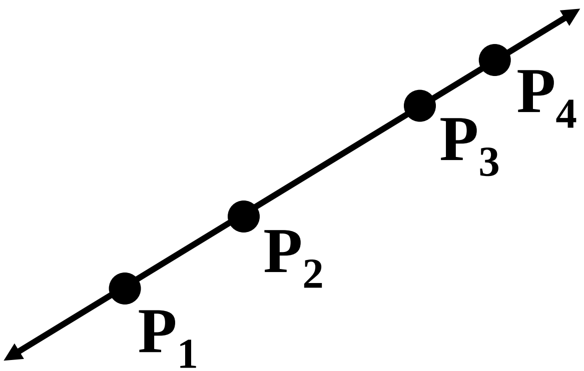


Compute Z from image measurements

- 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

$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|} \qquad \mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Can permute the point ordering

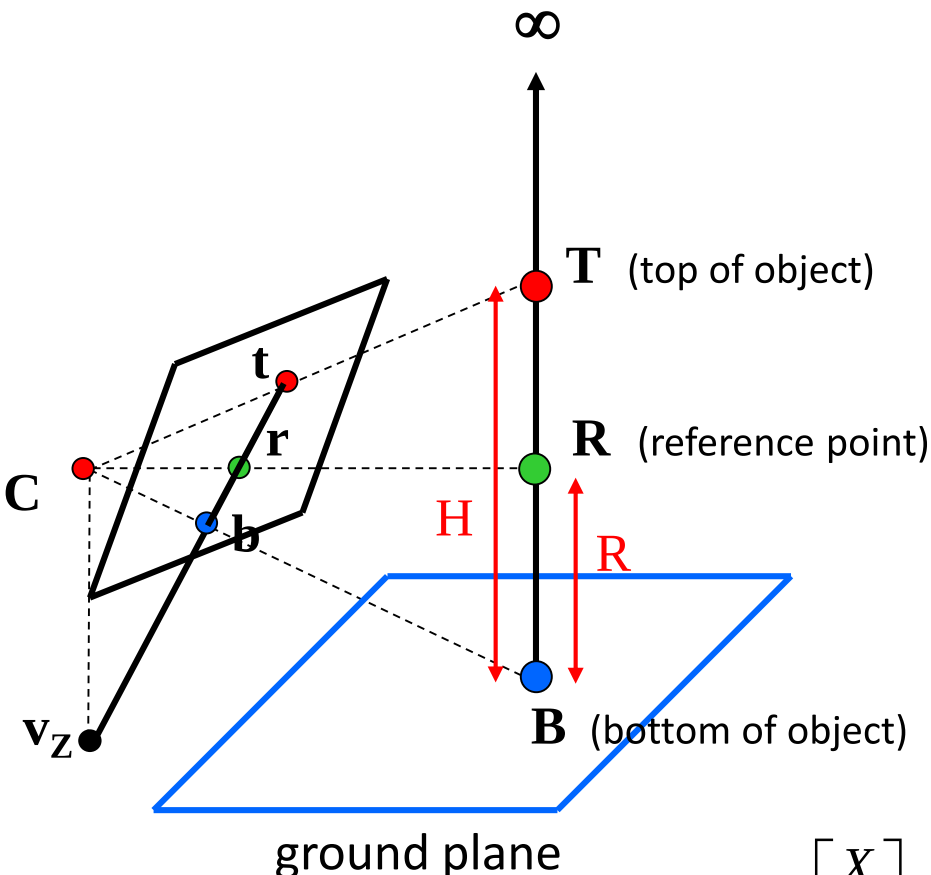
$$\frac{\|\mathbf{P}_1 - \mathbf{P}_3\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_3\|}$$

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

This is the fundamental invariant of projective geometry

Measuring height

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



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

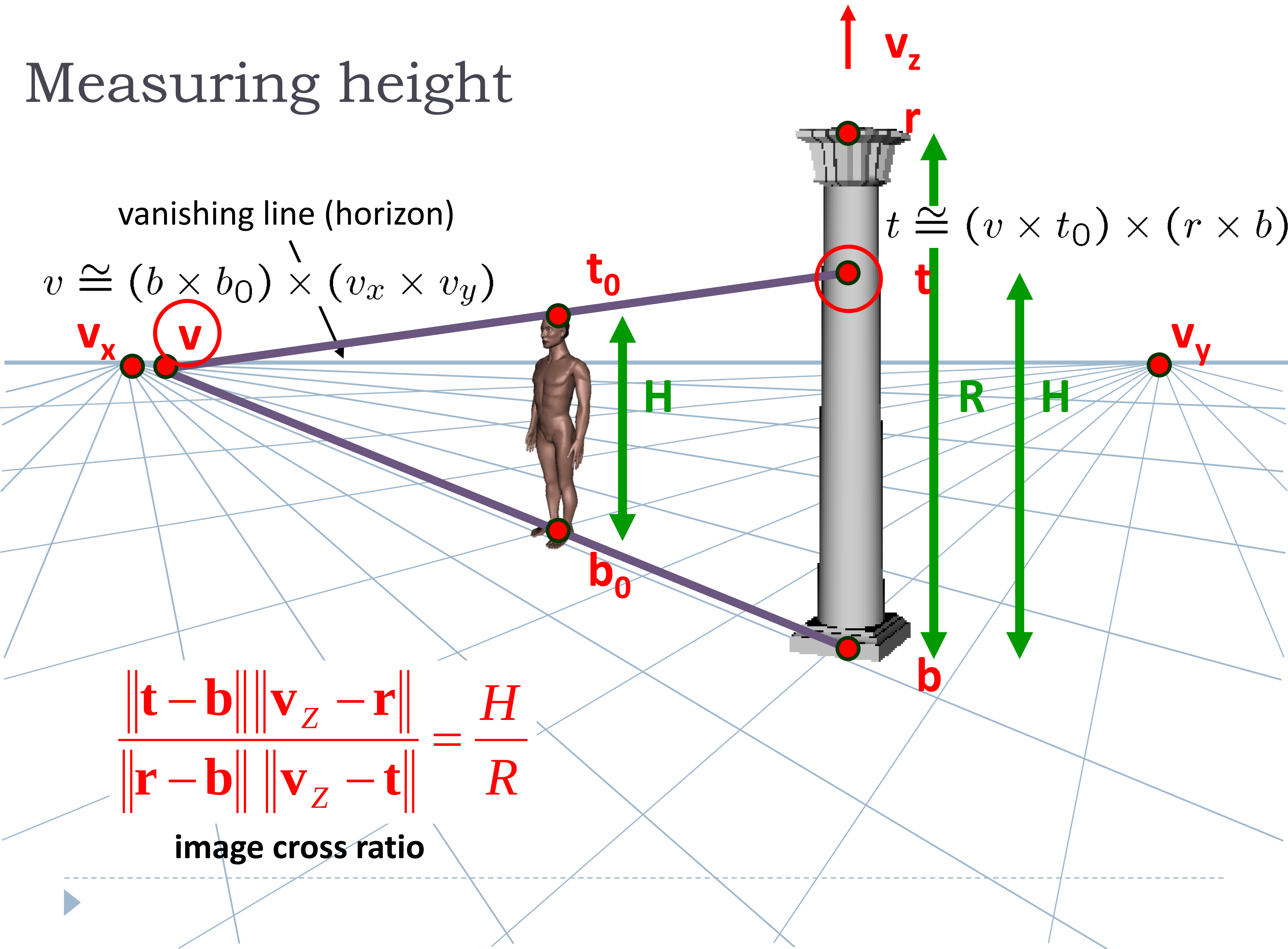
scene cross ratio

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

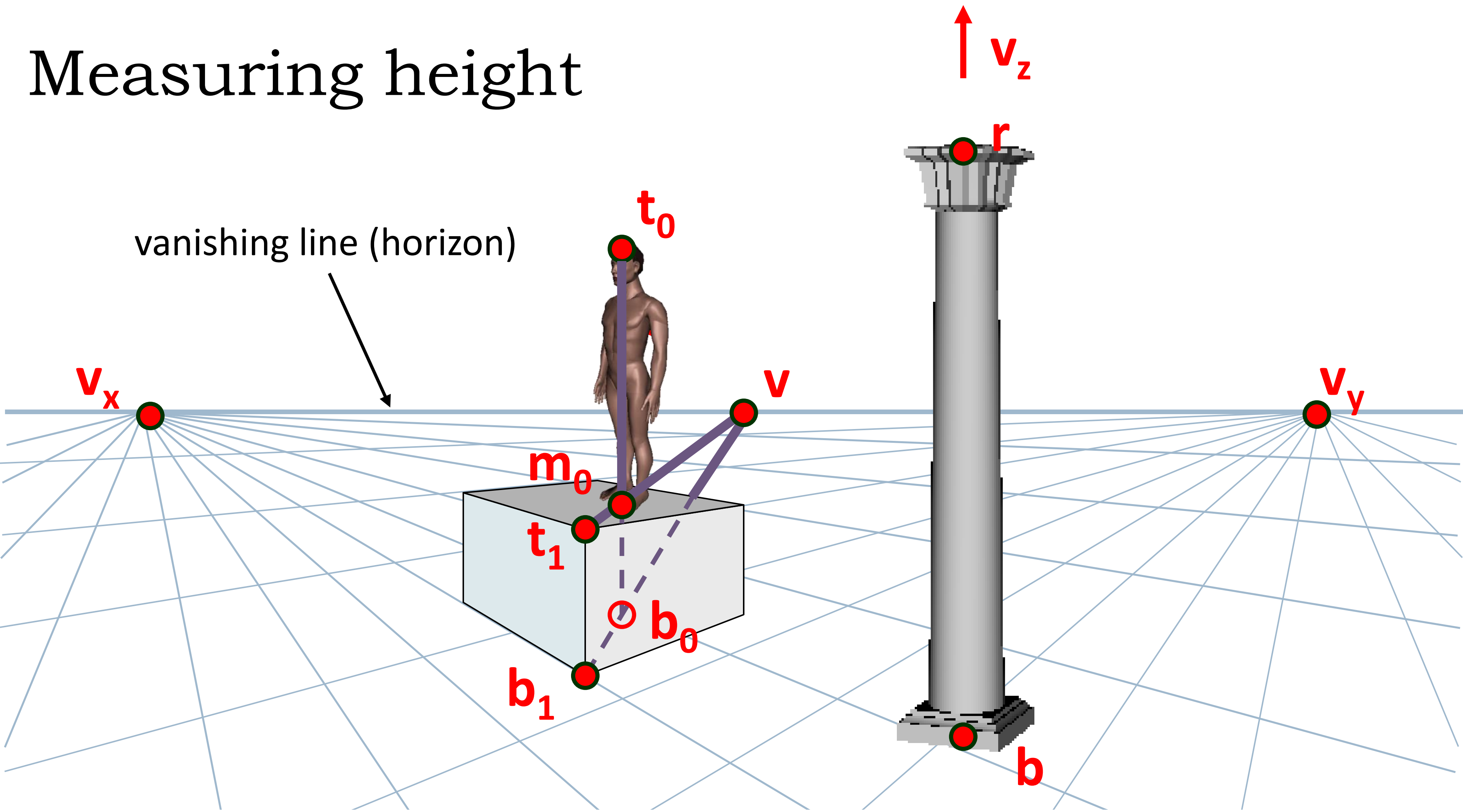
image cross ratio

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



Measuring height



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

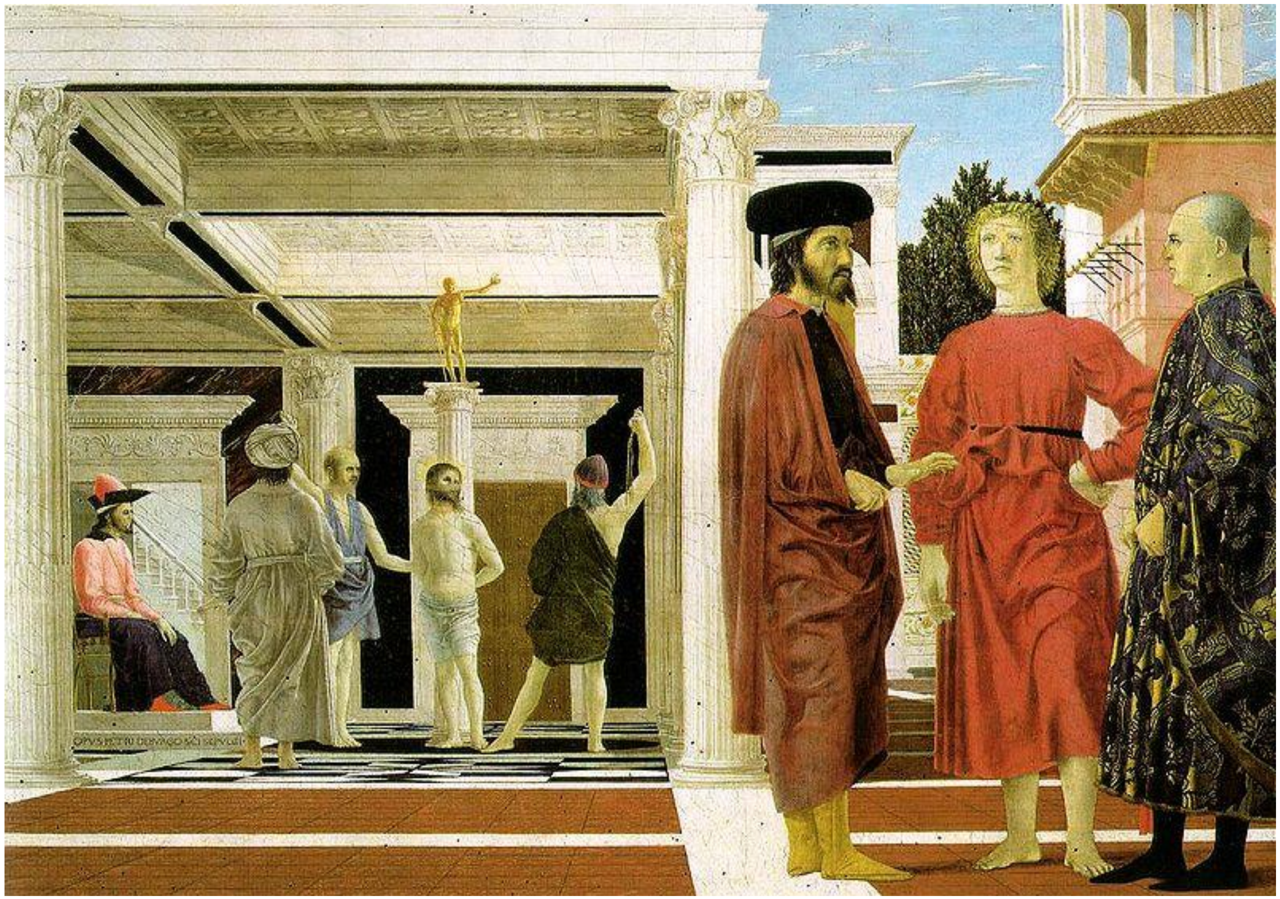
- Here the guy is standing on the box, height of box is known
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

3D Modeling from a photograph

St. Jerome in his Study, H. Steenwick

3D Modeling from a photograph

3D Modeling from a photograph

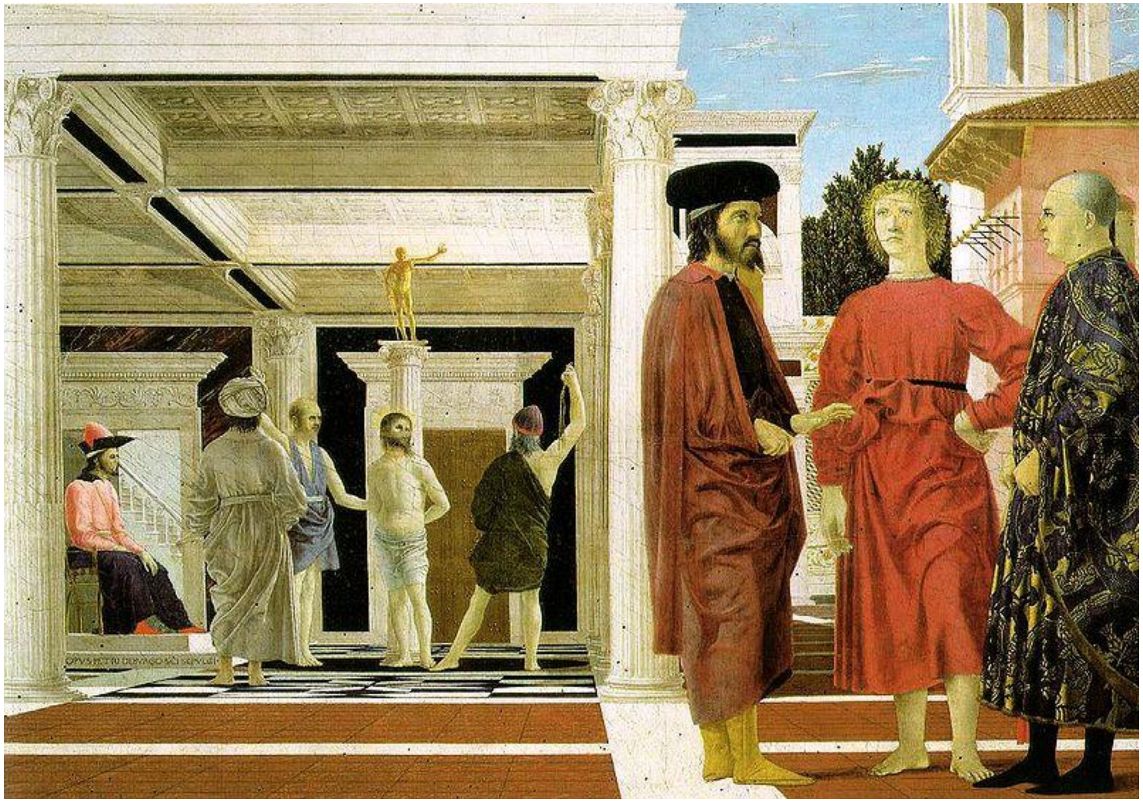


Flagellation, Piero della Francesca

3D Modeling from a photograph

video by Antonio Criminisi

3D Modeling from a photograph



3D Modeling from a photograph

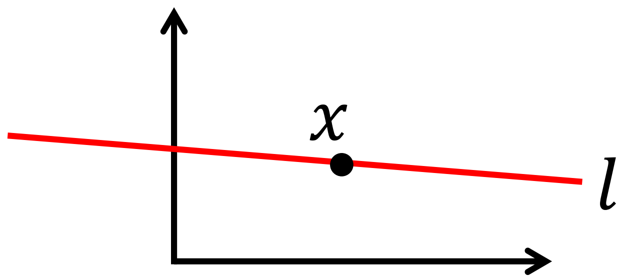
What can we do with a single image?

- Measure height ✓
- Camera calibration
- 3D reconstruction ?

Lines in a 2D plane

- The line equation

$$ax_1 + bx_2 + c = 0$$

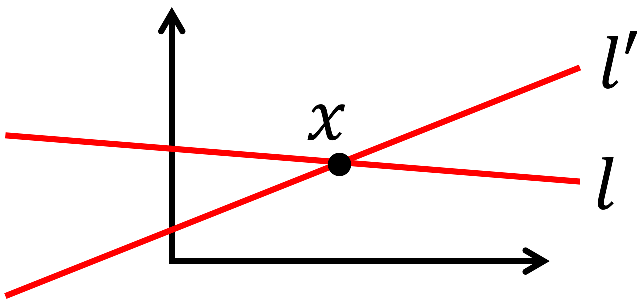


- Vector notation
  - Line $l = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
  - A point in homogeneous coordinates $x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$
- If the point lies on the line then $\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$

Lines in a 2D plane

- Intersecting lines

$$x = l \times l'$$



- Proof
  - $l \times l' \perp l \;\rightarrow\; (l \times l') \cdot l = 0 \;\rightarrow\; x \in l$
  - $l \times l' \perp l' \;\rightarrow\; (l \times l') \cdot l' = 0 \;\rightarrow\; x \in l'$

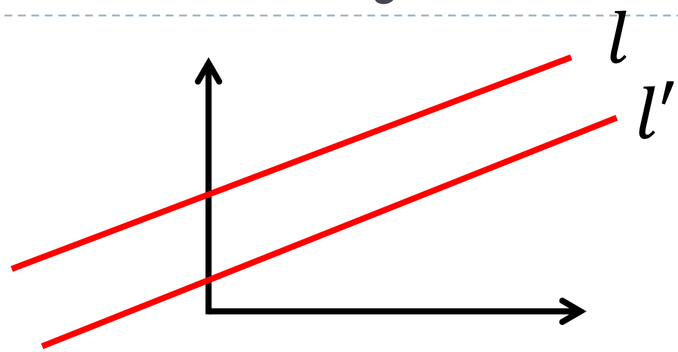
$$\Downarrow$$

- x is the intersection point

Ideal points = points at infinity

- A point $x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$

- At infinity $x_\infty = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$

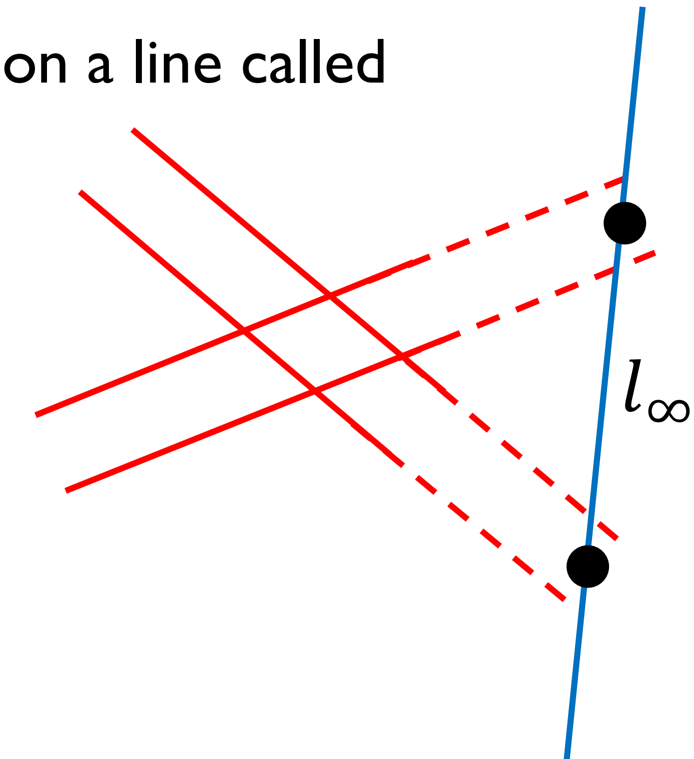


$$l = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad l' = \begin{bmatrix} a \\ b \\ c' \end{bmatrix}$$

- The intersection between two parallel lines in 2D is a point at infinity
  - $v = l \times l' = (c - c')\begin{bmatrix} b \\ -a \\ 0 \end{bmatrix}$

The line at infinity

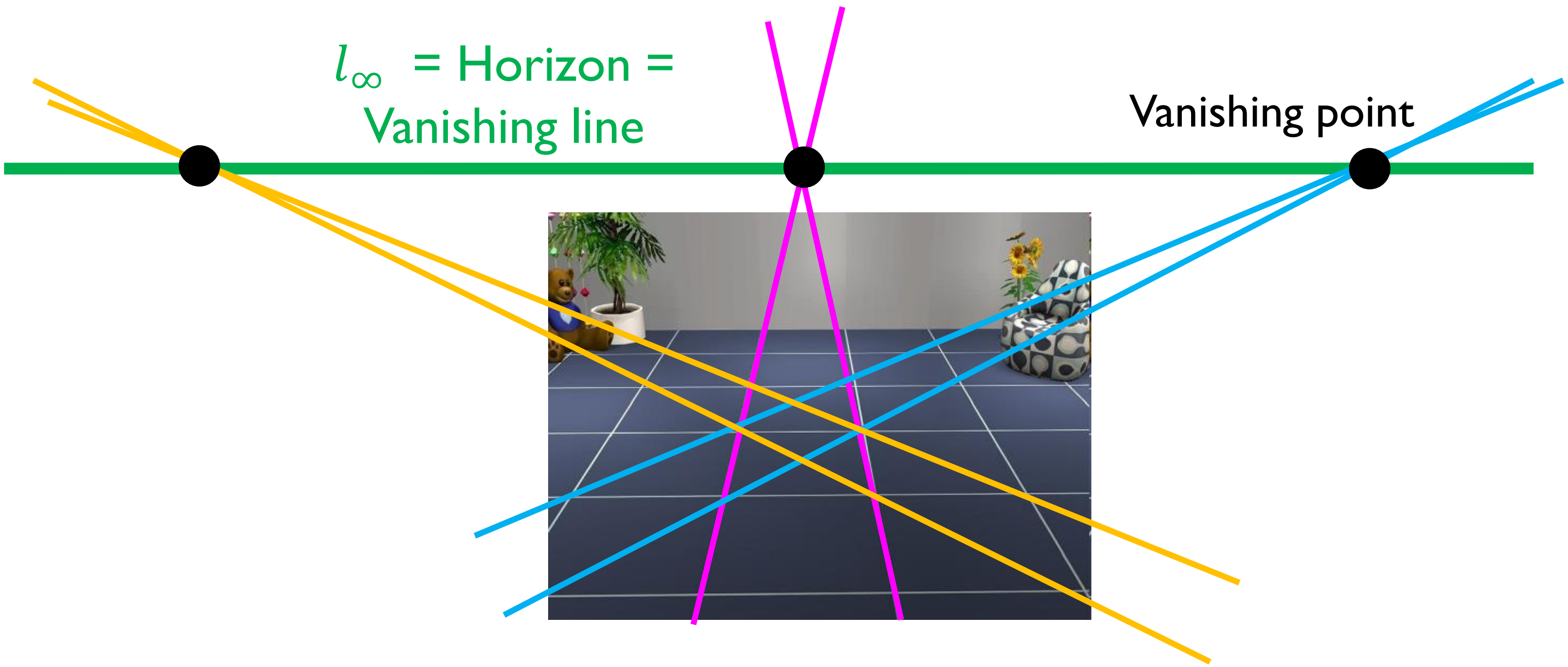
- A set of ideal points (at infinity) lies on a line called "the line at infinity"
- $l_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$



- Let's verify this via the line equation
  - $\begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$ (dot product is 0 → the point is on the line)

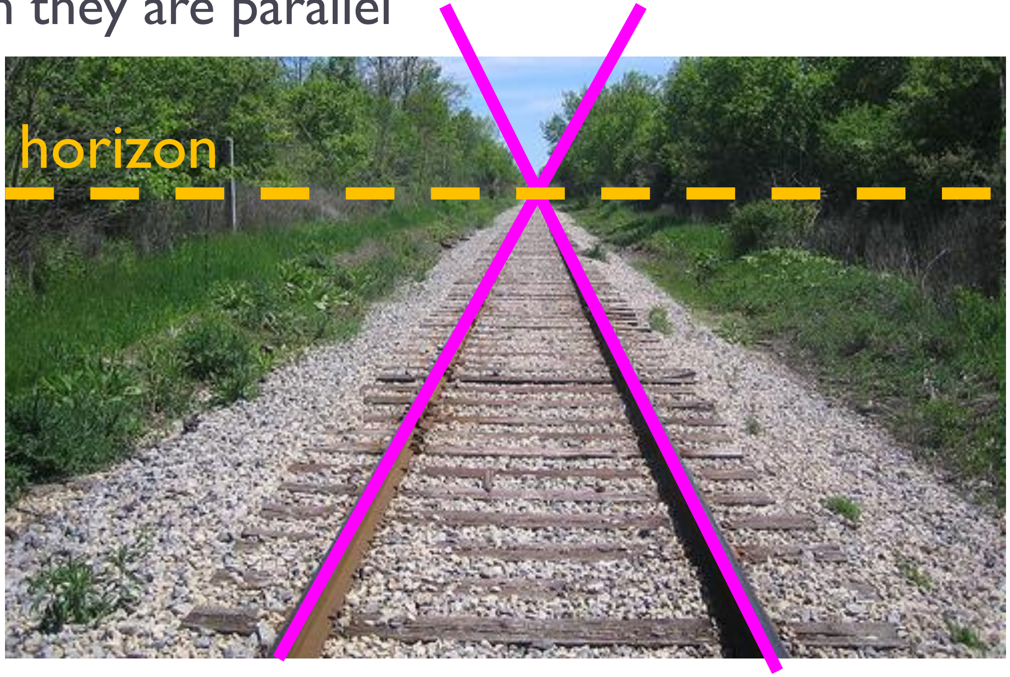
Parallel lines on a plane in 3D

- Vanishing point: The projection onto the image of the intersection of two parallel lines in 3D
- Vanishing line: Vanishing points of parallel lines that lie on the same plane, lie on the vanishing line of that plane.

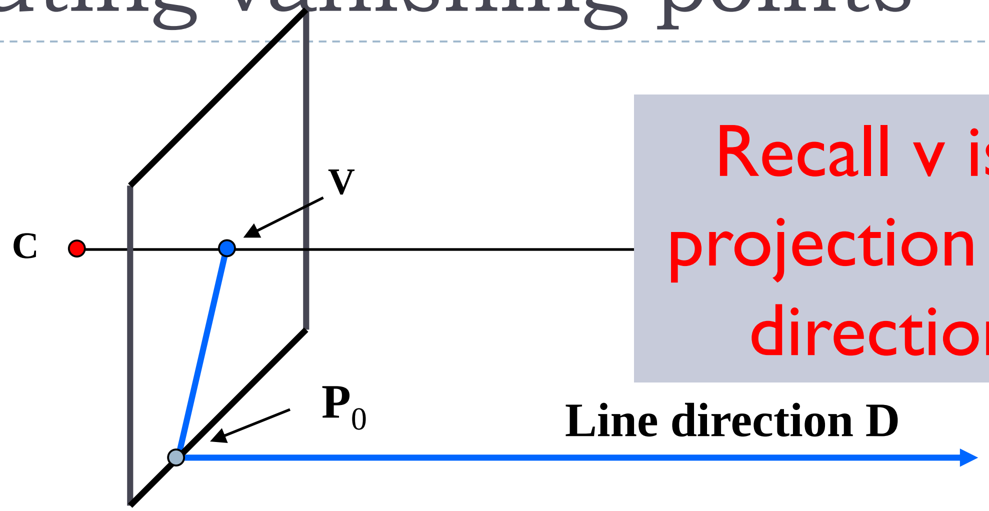


The horizon

- How can we tell if two lines in the image are parallel in the world?
  - Find the horizon line of the corresponding plane
  - Check if the two lines intersect at the horizon
  - If yes, then they are parallel



Computing vanishing points



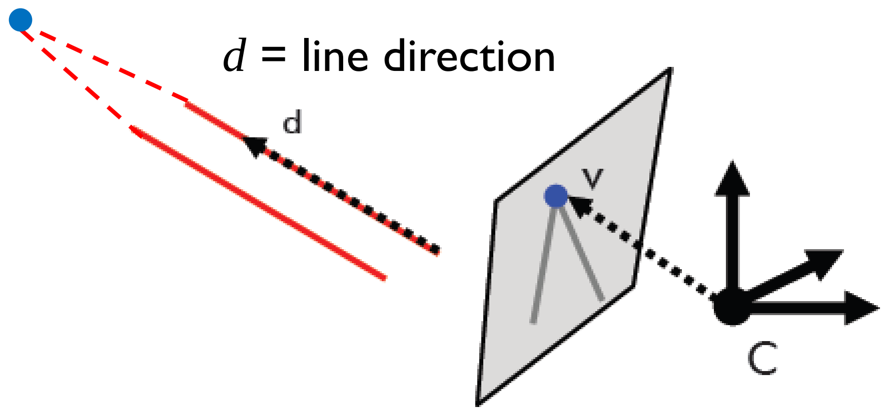
Recall v is the projection of the direction D

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

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

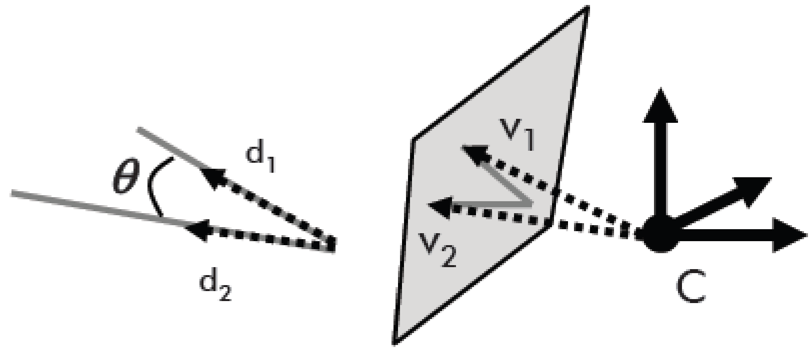
- Properties of $\mathbf{v} = M\mathbf{P}_\infty$
  - $\mathbf{P}_\infty$ is a point at *infinity*, $\mathbf{v}$ is its projection
  - Depends only on line *direction* $\mathbf{D}$
  - Parallel lines $\mathbf{P}_0 + t\mathbf{D}$, $\mathbf{P}_1 + t\mathbf{D}$ intersect at $\mathbf{P}_\infty$

Vanishing points



- Assume camera projection matrix is $M = K[I \quad 0]$
- Then the projection of the vanishing point is $v = Kd$

Angle between 2 scene lines



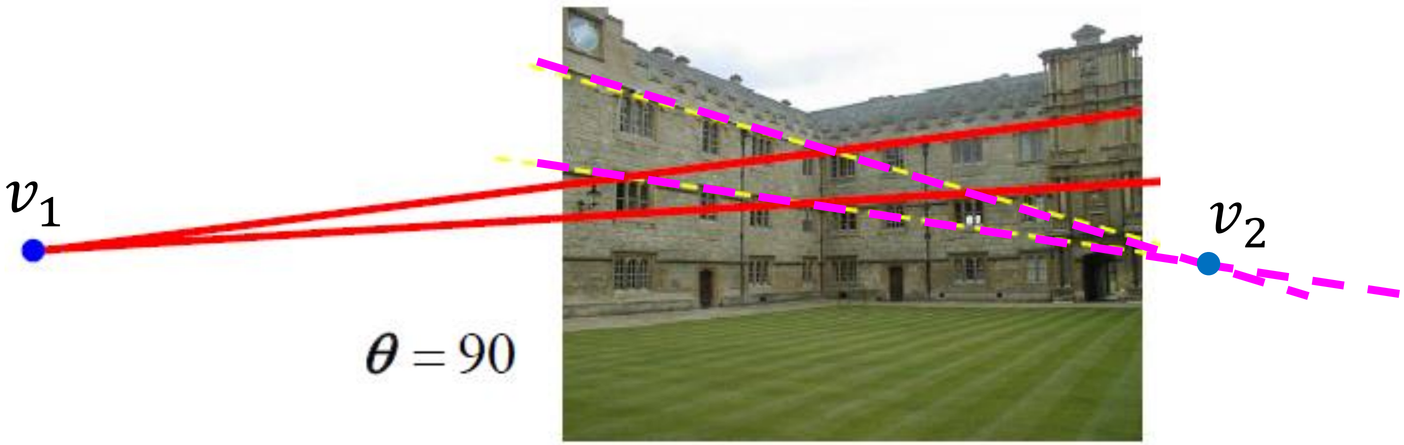
$$v = Kd$$

$$\cos\theta = \frac{{v_1}^T K^T K v_2}{\sqrt{{v_1}^T K^T K v_1}\sqrt{{v_2}^T K^T K v_2}}$$

If the lines are orthogonal then

$\theta = 90$ and ${v_1}^T K^T K v_2 = 0$ Let's use this!

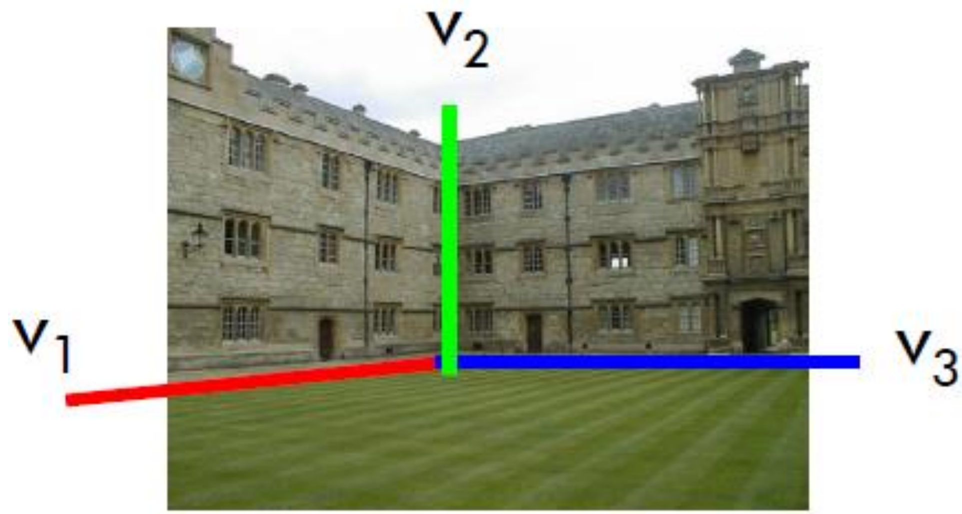
Angle between 2 scene lines



$$v_1{}^T K^T K v_2 = 0 \Rightarrow \text{constraint on } K$$

From two vanishing points!

Single view calibration



Mark 3 orthogonal lines, find 3 vanishing points, and solve for *K* using three constraints

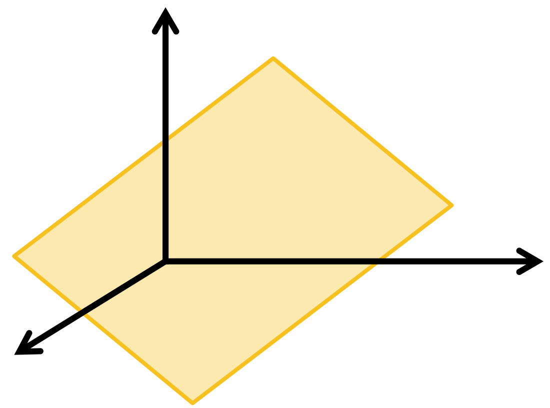
$$\begin{cases} {v_1}^T K^T K v_2 = 0 \\ {v_1}^T K^T K v_3 = 0 \\ {v_2}^T K^T K v_3 = 0 \end{cases}$$

What can we do with a single image?

- Measure height ✓
- Camera calibration ✓
- 3D reconstruction
  - Manhattan world

Points and planes in 3D

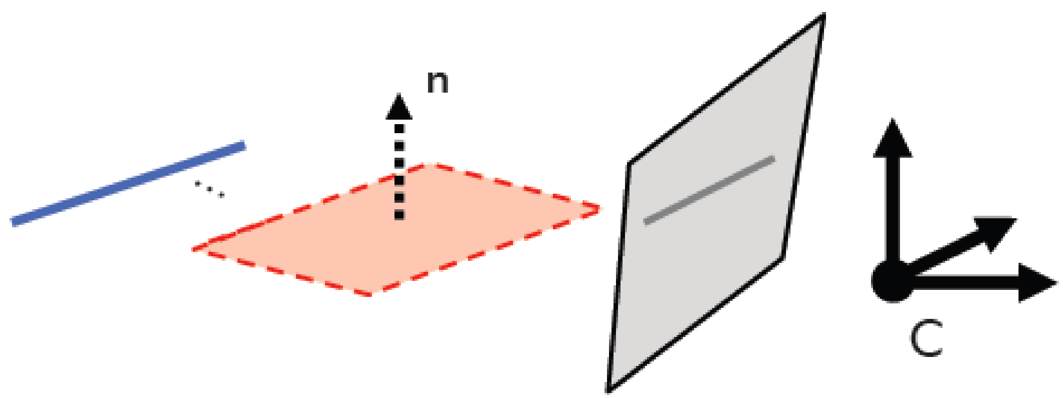
- A point $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$
- A plane $\pi = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$
- A point that lies on a plane $\mathrm{x} \cdot \pi = 0$



The vanishing line

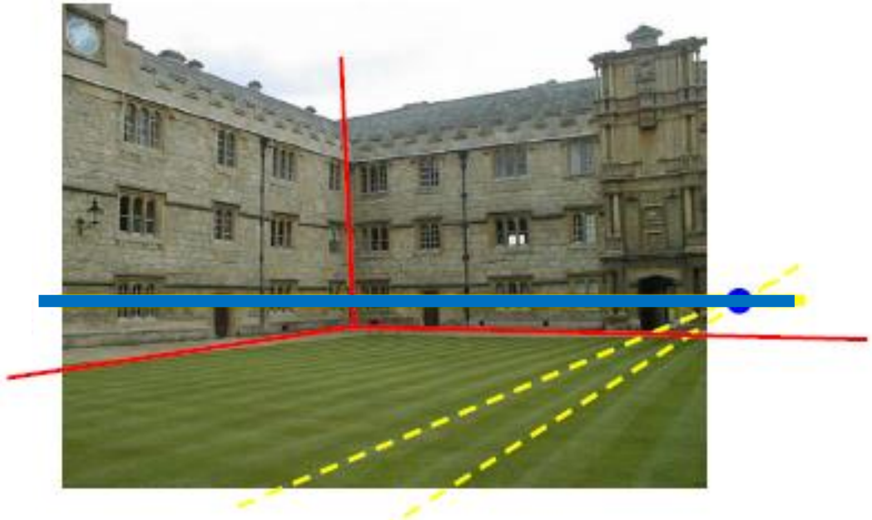
- Parallel planes intersect the plane at infinity in a common line – the vanishing line = horizon
- The normal to these planes can be computed from the horizon

$$n = K^T l_{horizon}$$ (*K* is the camera calibration matrix)



Reconstruct surface normals

$l_{horizon}$



- If K is known we can compute plane normals

$$n = K^T l_{horizon}$$

Application

- These transformations are used in single-view metrology

Criminisi & Zisserman, 99

Application

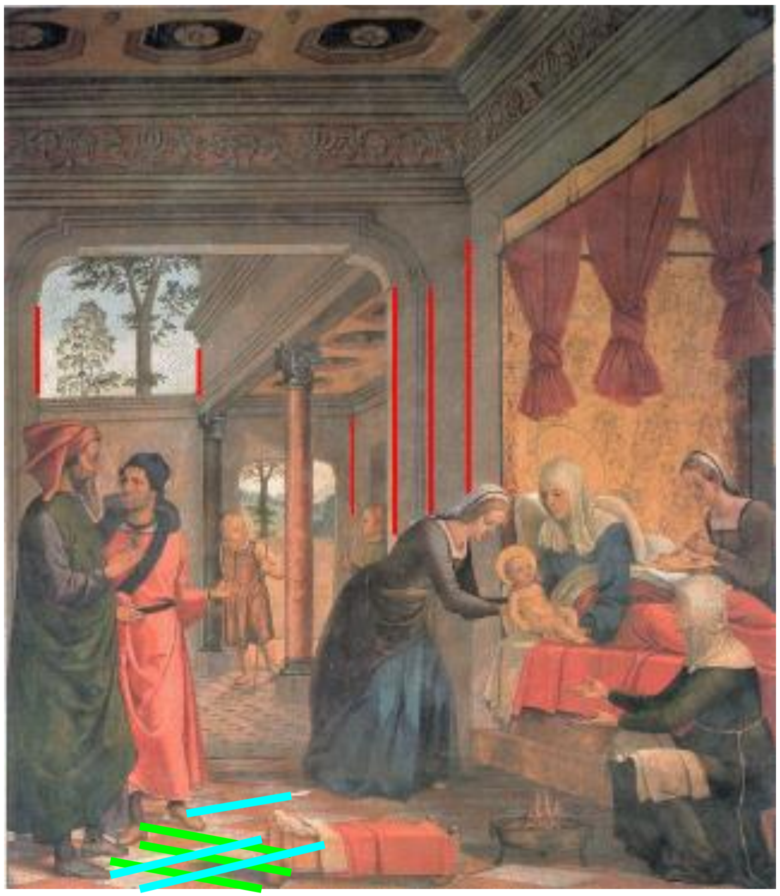
- These transformations are used in single-view metrology

Criminisi & Zisserman, 99

Application

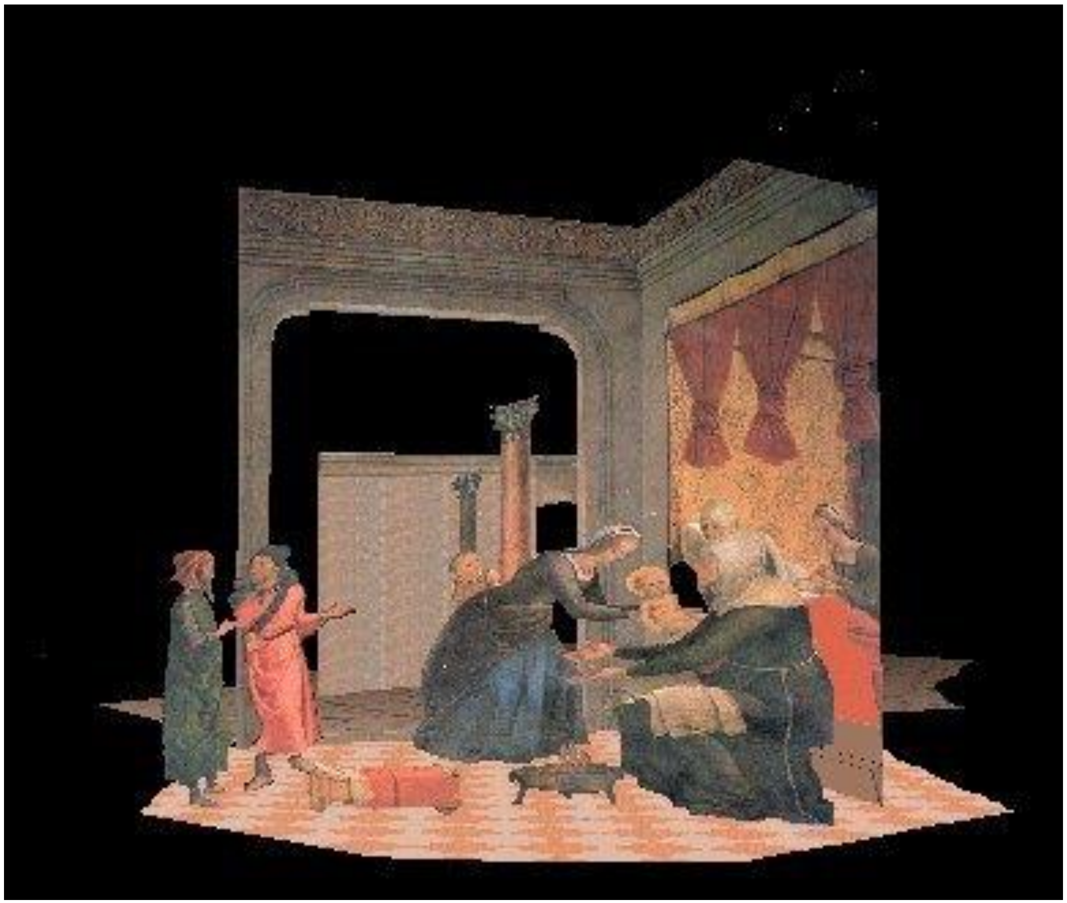
- These transformations are used in single-view metrology

Criminisi & Zisserman, 99



Application

- These transformations are used in single-view metrology



Criminisi & Zisserman, 99

Single view metrology

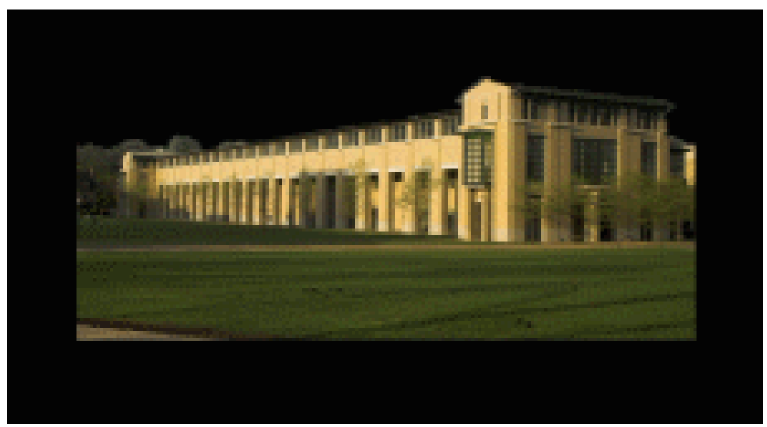
- Pros
  - Cool
  - Only a single image required

- Cons
  - Manually select vanishing points and lines
  - Planar surfaces
  - Occlusion boundaries
  - …

Other approaches

- Learn appearance-based models of surfaces at various orientations

Hoiem et al, 2005



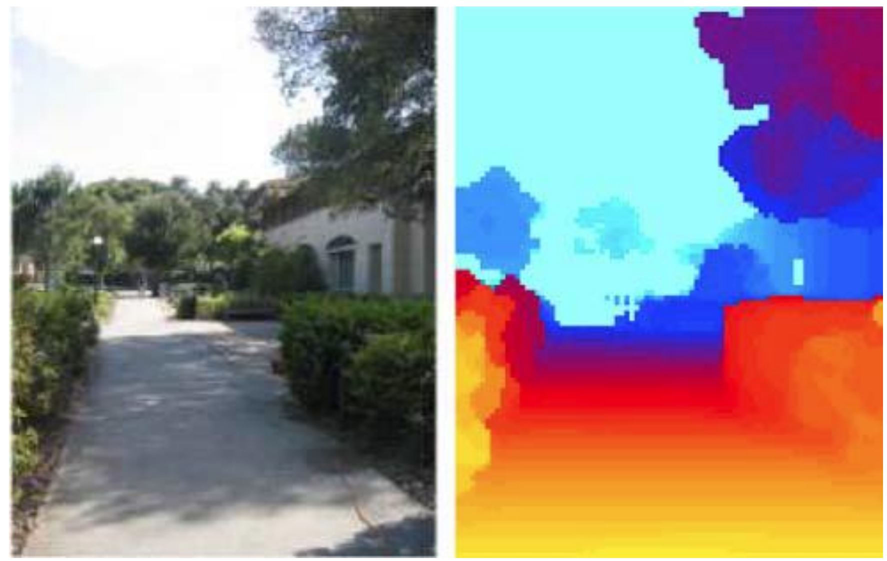
http://www.cs.uiuc.edu/homes/dhoiem/projects/software.html

Other approaches

- A learning-based approach to single-view metrology

Saxena, Sun, Ng, 2005

- Input = image + corresponding depth map
- Learn how to match image patches to corresponding depth patches



http://make3d.cs.cornell.edu/

Up-to-date applications

Inserting synthetic objects into images

http://vimeo.com/28962540

[Rendering synthetic objects into legacy photographs]()

Karsch et al

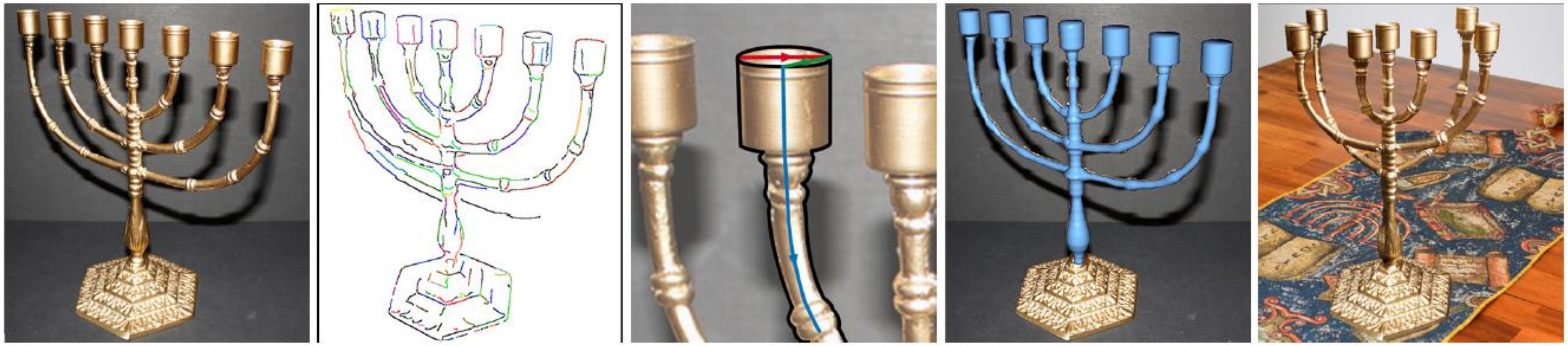
SIGGRAPH 2011

Image manipulation

- http://www.cse.iitb.ac.in/~jaseem/graphicsa23.pdf
- Interactive Images: Cuboid proxies for Smart Image Manipulation,
  Youyi Zheng, Xiang Chen, Ming-Ming Cheng, Kun Zhou, Shi-Min Hu and Niloy J. Mitra, SIGGRAPH 2012

3-Sweep: Extracting Editable Objects from a Single Photo

- Chen et al. SIGGRAPH 2013
  http://www.faculty.idc.ac.il/arik/site/3Sweep.asp
  https://vimeo.com/148236679



(a) (b) (c) (d) (e)

Figure 1: *3-Sweep Object Extraction. (a) Input image. (b) Extracted edges. (c) 3-Sweep modeling of one component of the object. (d) The full extracted 3D model. (e) Editing the model by rotating each arm in a different direction, and pasting onto a new background. The base of the object is transferred by alpha matting and compositing.*

Augmented reality glasses (not exactly single view)

- https://www.getameta.com/
- https://www.rideonvision.com/
- https://www.microsoft.com/microsoft-hololens/en-us

End – Single-view metrology

Now you know how it works

Projection of the line at infinity

- Perspective projection 3D→2D can be written as
  - $M = \begin{bmatrix} A & t \\ v & b \end{bmatrix}_{3\times 4}$
- Perspective projection of a line gives a line
  - $l' = Ml$
- Perspective projection of the line at infinity
  - $Ml_\infty = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ b \end{bmatrix}$ not a line at infinity!!
  - This is the horizon line