CS6670: Computer Vision

Noah Snavely

Lecture 15: Single-view modeling

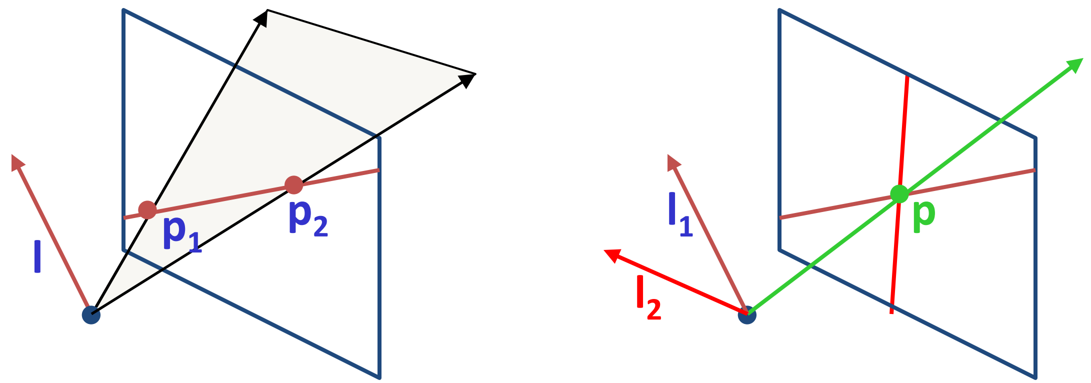
Announcements

- Partners for project 2b
- Midterm handed out Friday, due Wednesday at the beginning of class

Point and line duality

– A line **l** is a homogeneous 3-vector

– It is ⊥ to every point (ray) **p** on the line: **l p**=0



What is the line **l** spanned by rays $\mathbf{p_1}$ and $\mathbf{p_2}$?

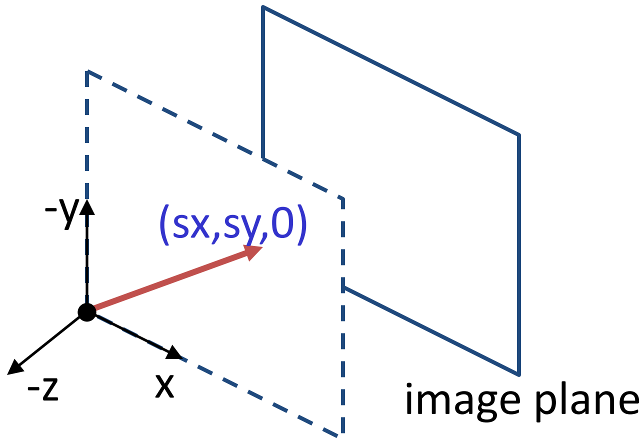
- **l** is ⊥ to $\mathbf{p_1}$ and $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- **l** can be interpreted as a *plane normal*

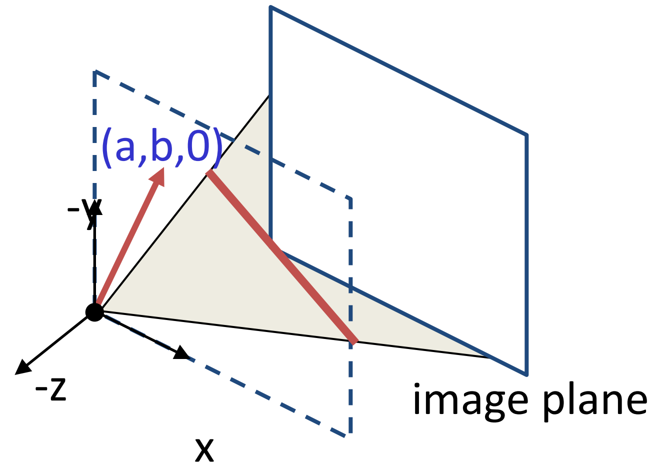
What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

- **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

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*)

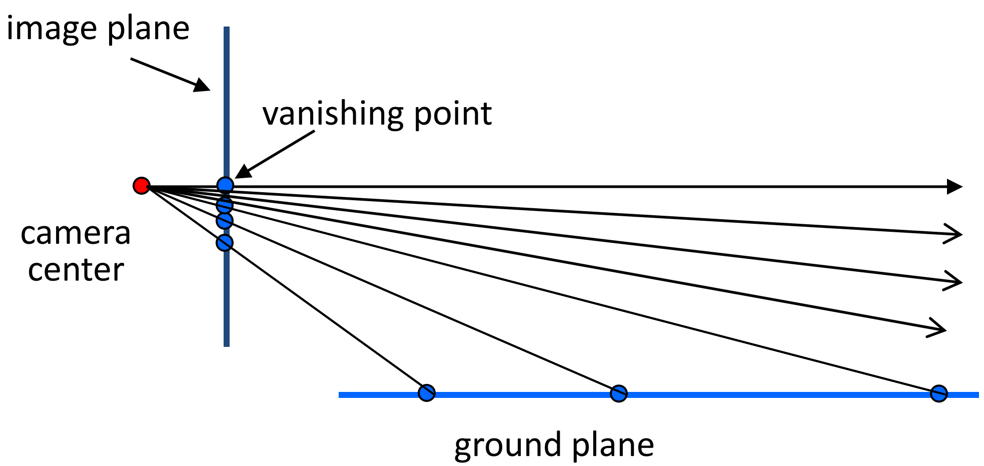
3D projective geometry

- 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

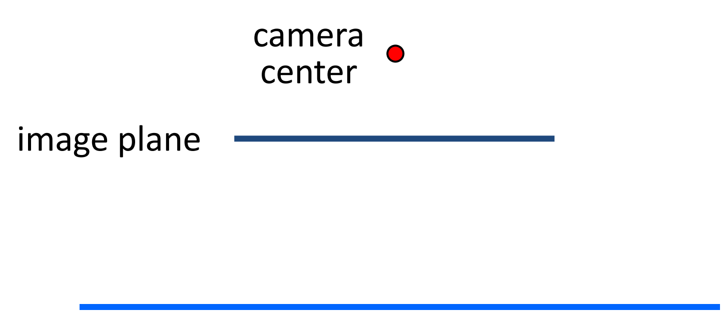
3D to 2D: perspective projection

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

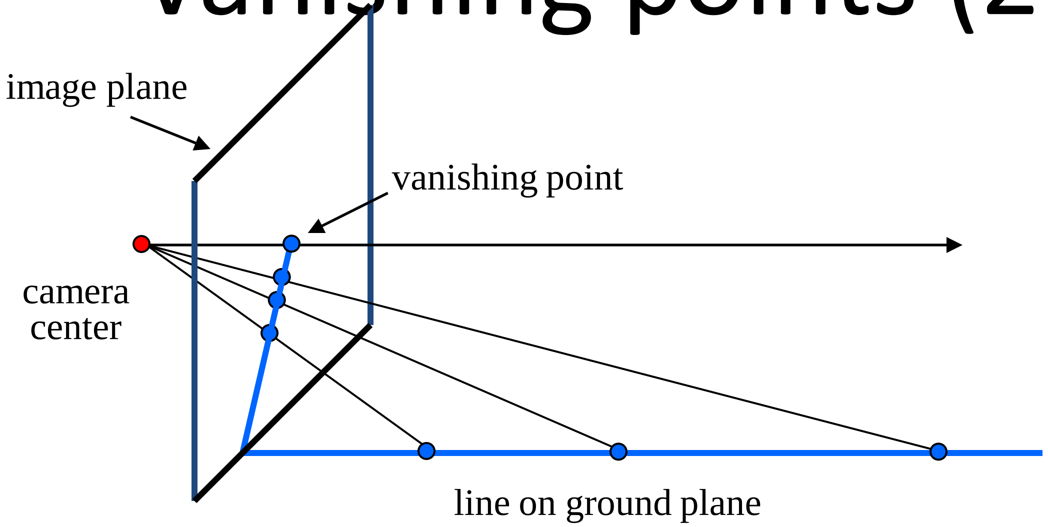
Vanishing points (1D)



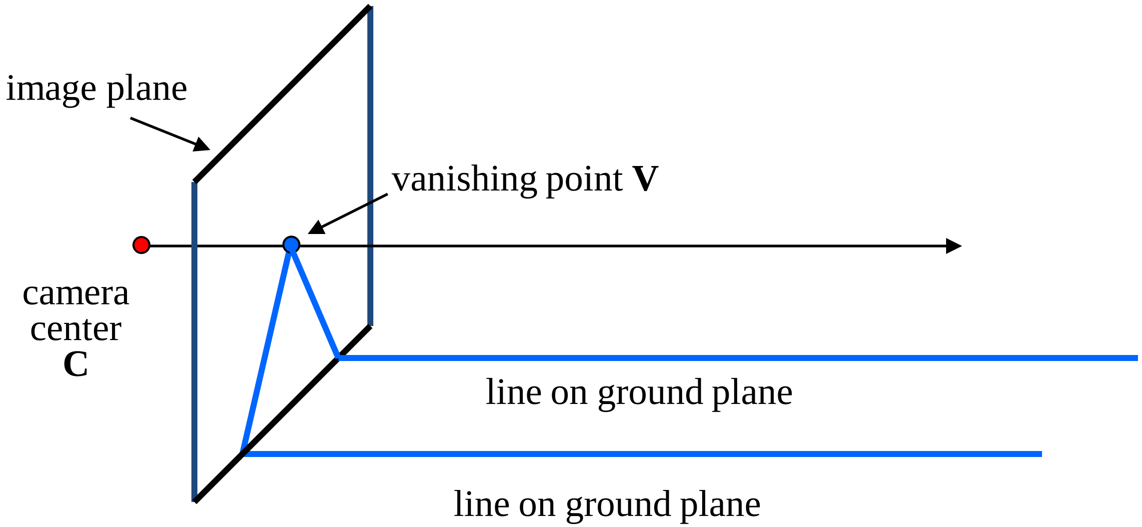
- Vanishing point
  - projection of a point at infinity
  - can often (but not always) project to a finite point in the image



Vanishing points (2D)

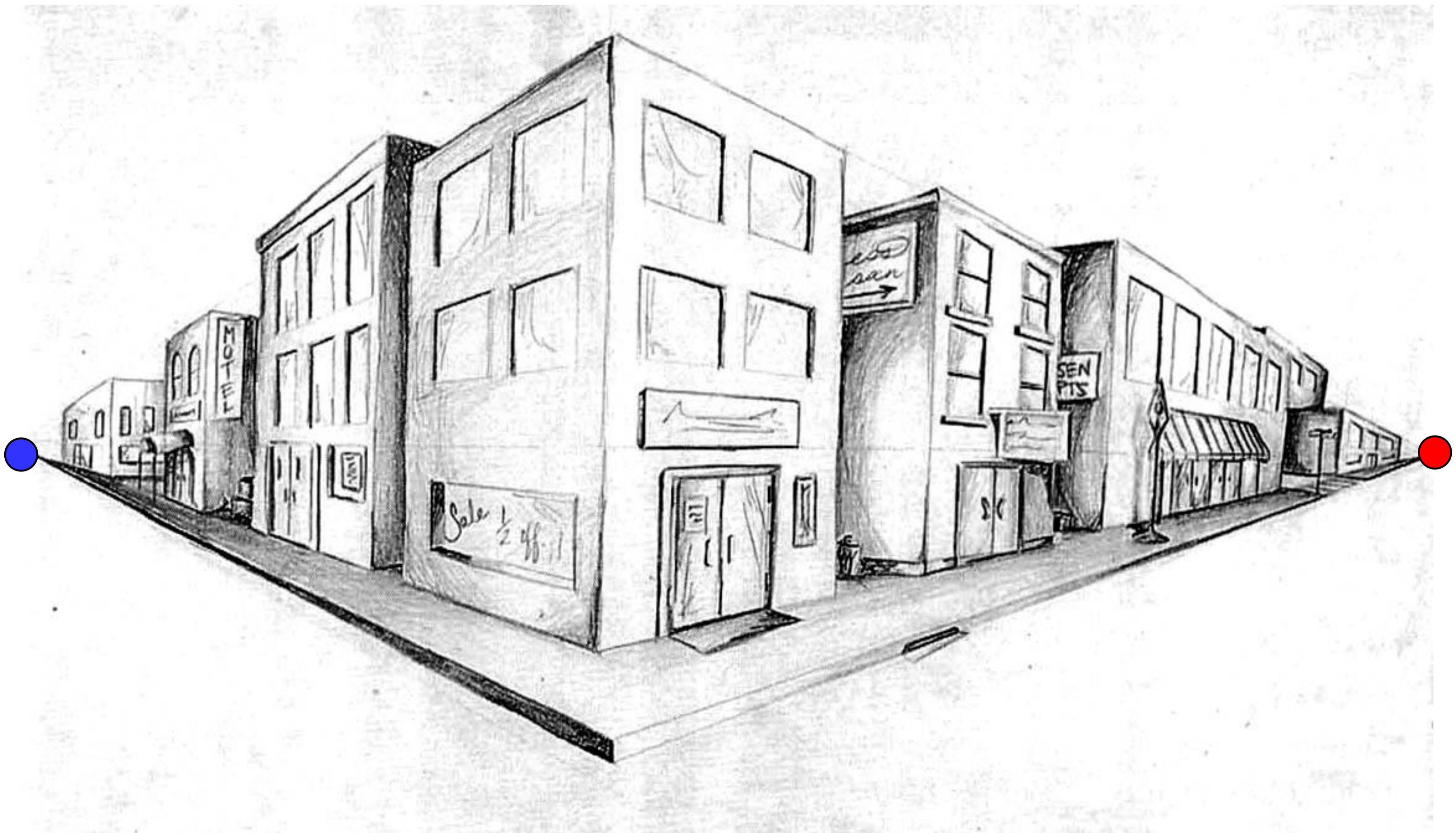


Vanishing points



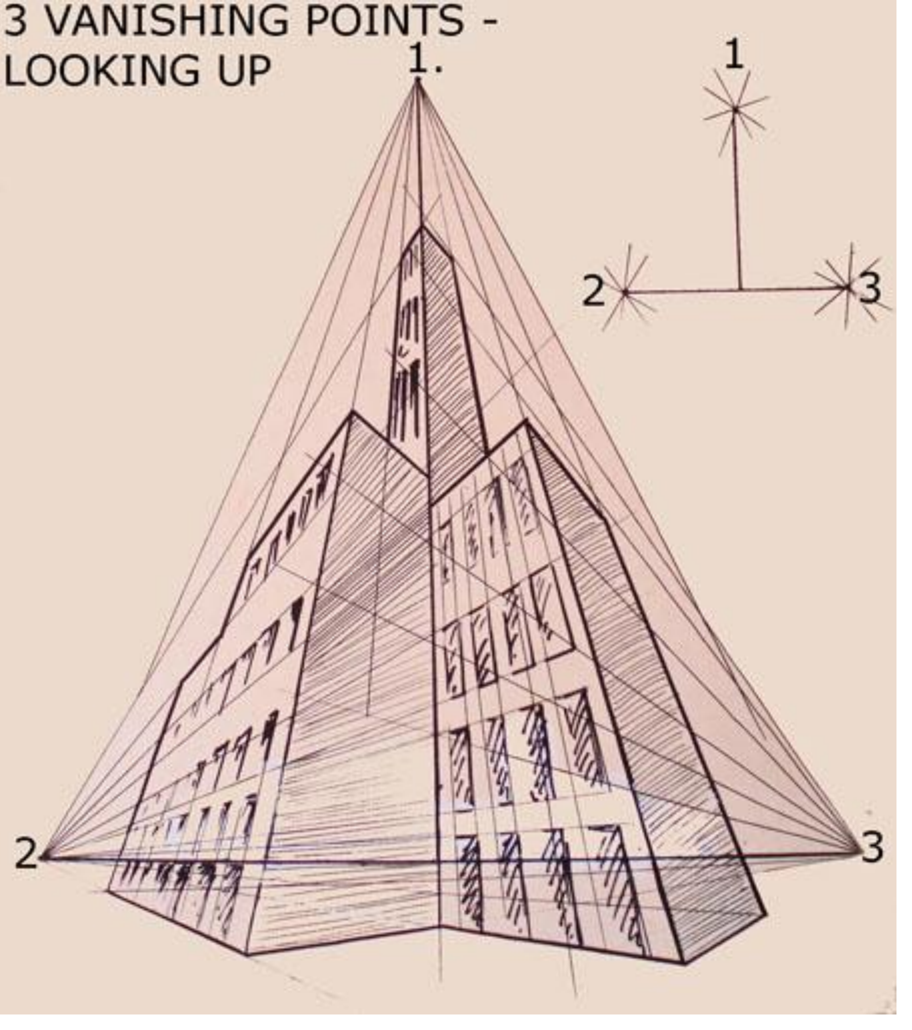
- Properties
  - Any two parallel lines (in 3D) have the same vanishing point **v**
  - The ray from **C** through **v** is parallel to the lines
  - An image may have more than one vanishing point
    - in fact, every image point is a potential vanishing point

Two point perspective

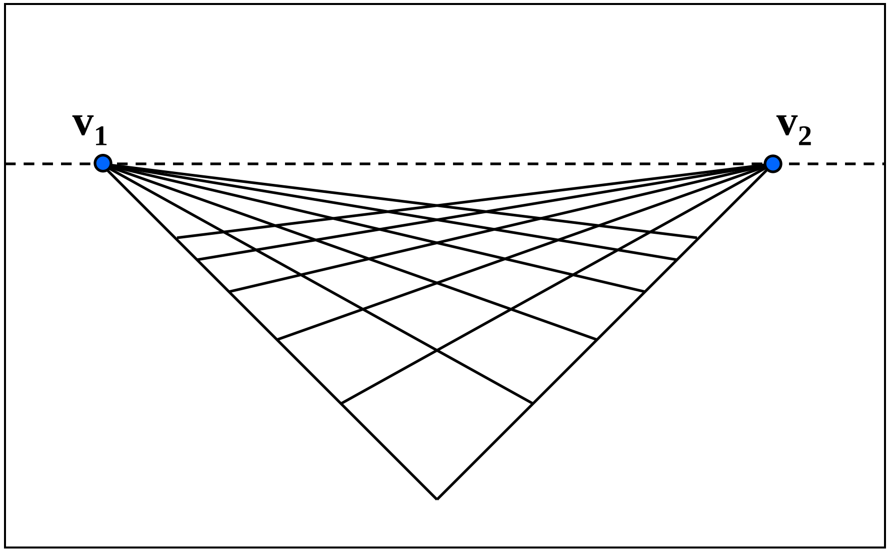
v_y


v_x

Three point perspective

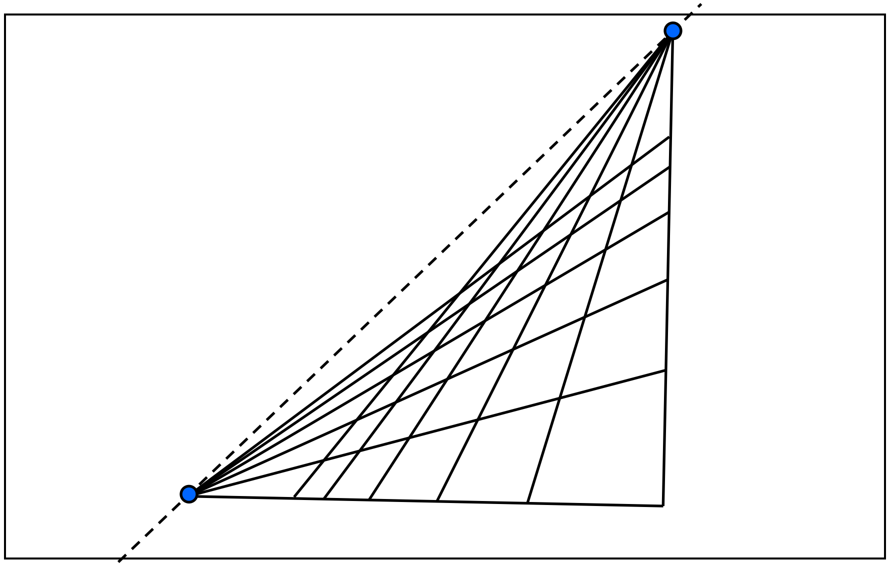


Vanishing lines



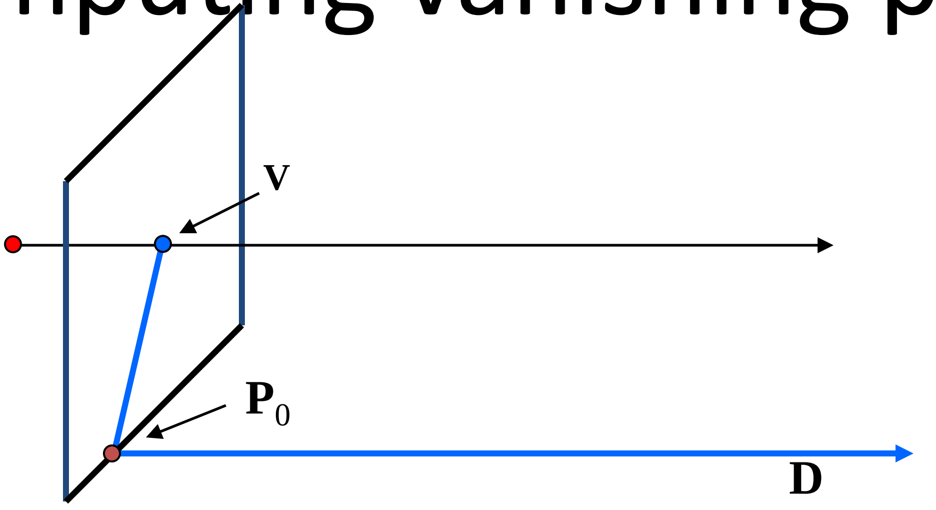
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes (can) define different vanishing lines

Vanishing lines



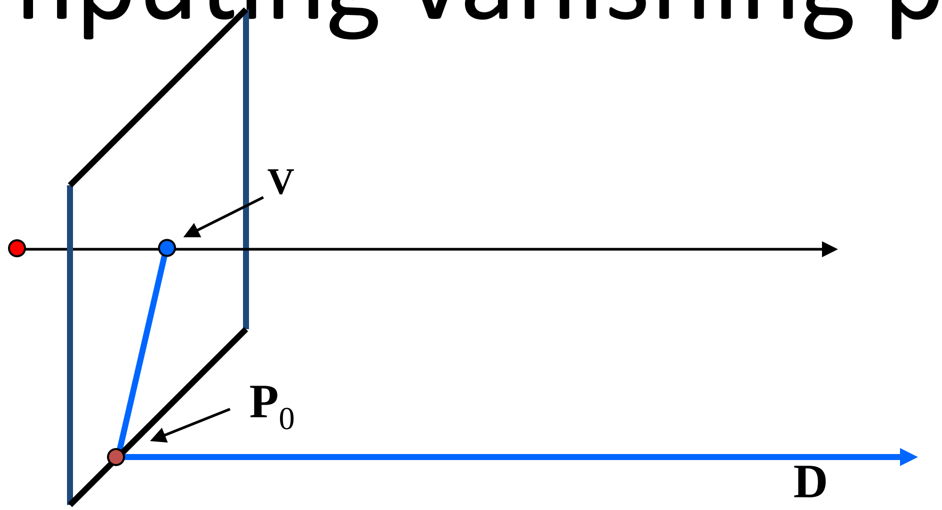
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes (can) define different vanishing lines

Computing vanishing points



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

Computing vanishing points

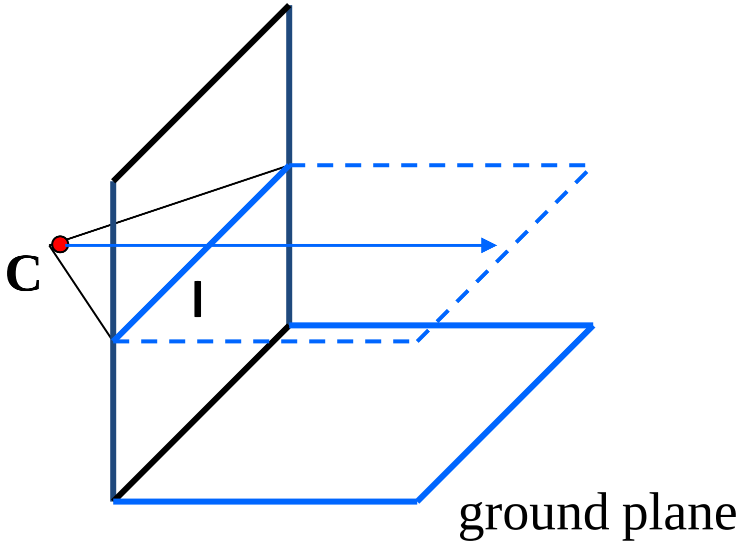


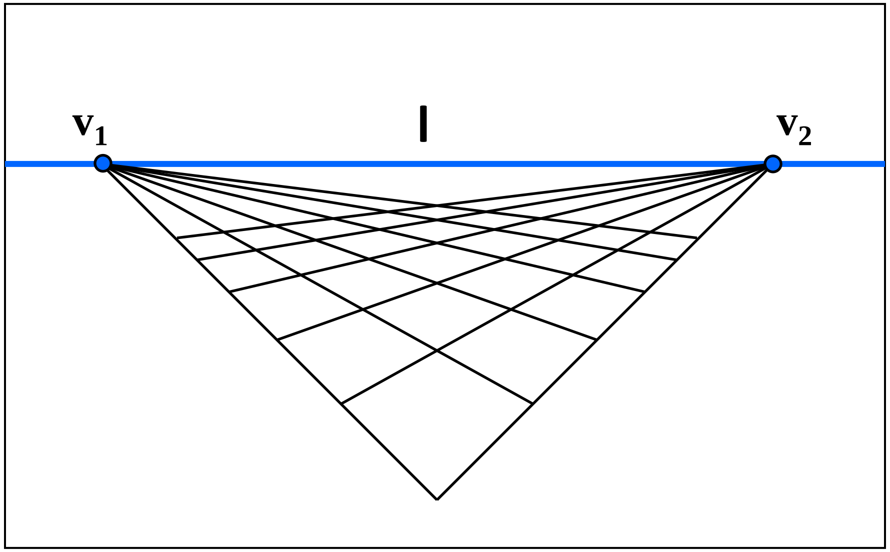
$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

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

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

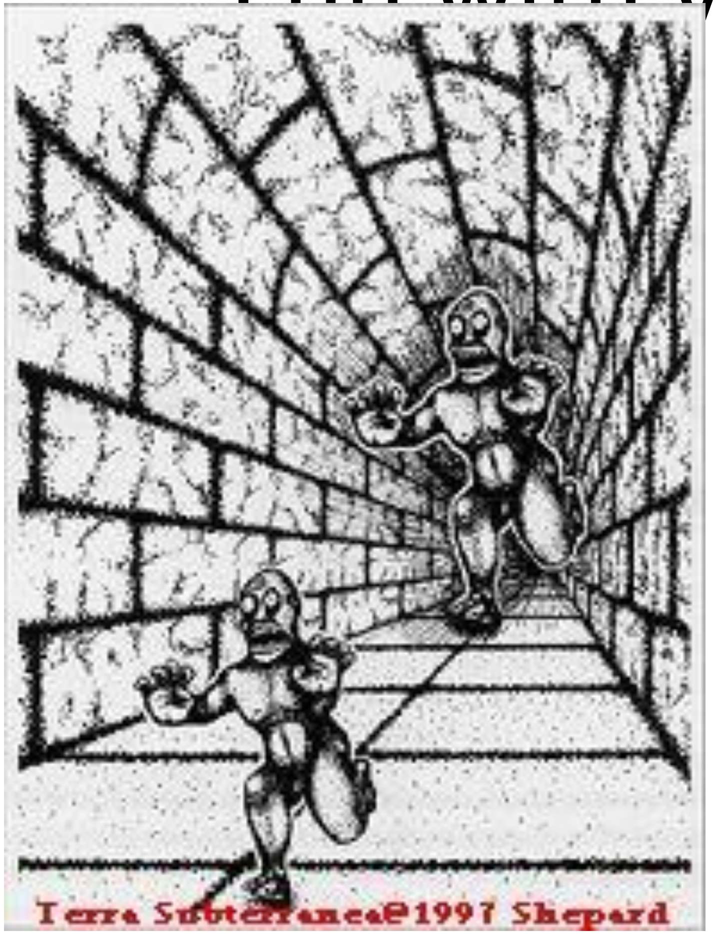
Computing vanishing lines

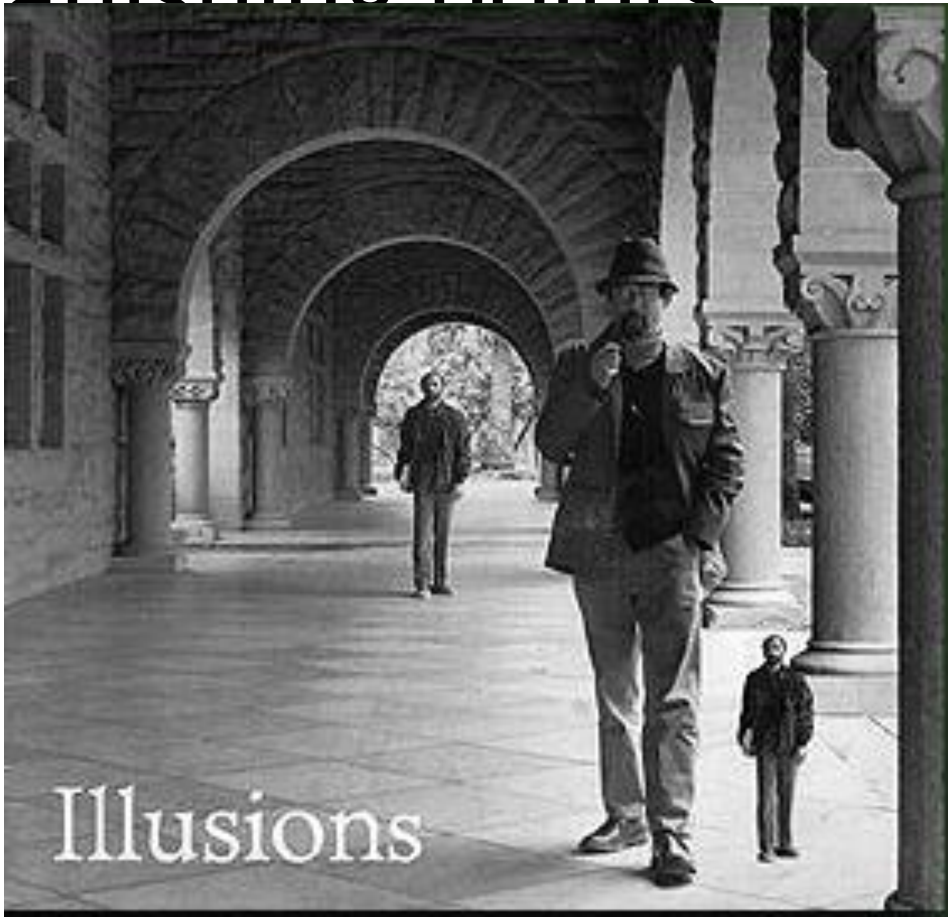




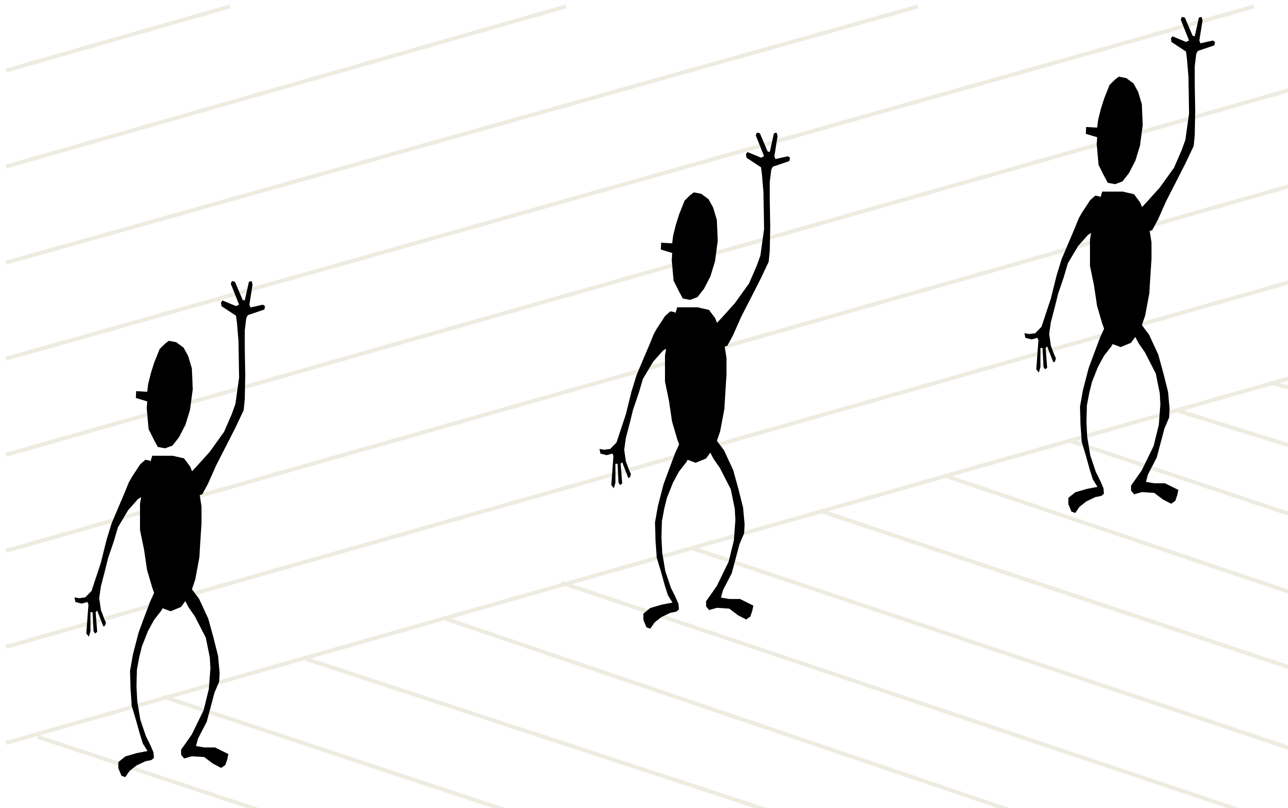
- Properties
  - **l** is intersection of horizontal plane through **C** with image plane
  - Compute **l** from two sets of parallel lines on ground plane
  - All points at same height as **C** project to **l**
    - points higher than C project above l
  - Provides way of comparing height of objects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

Fun with vanishing points

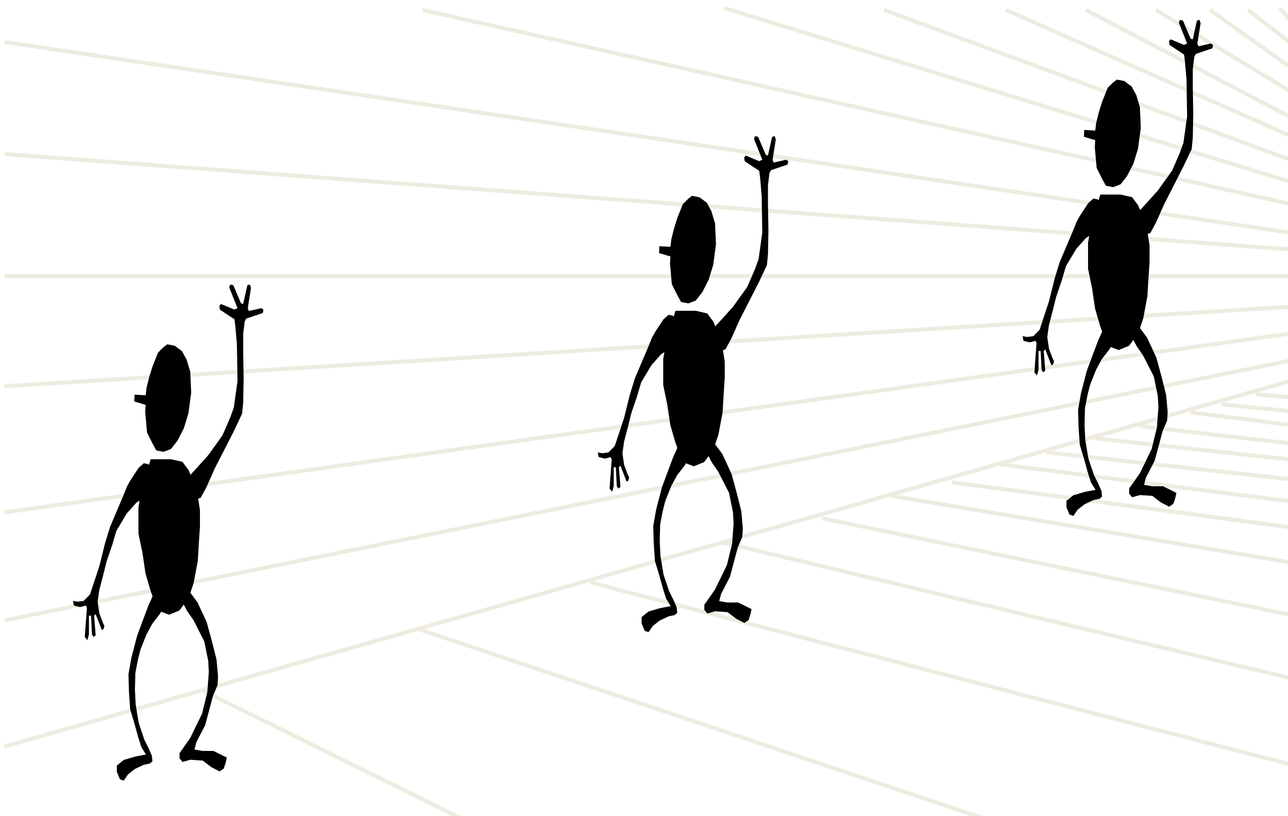




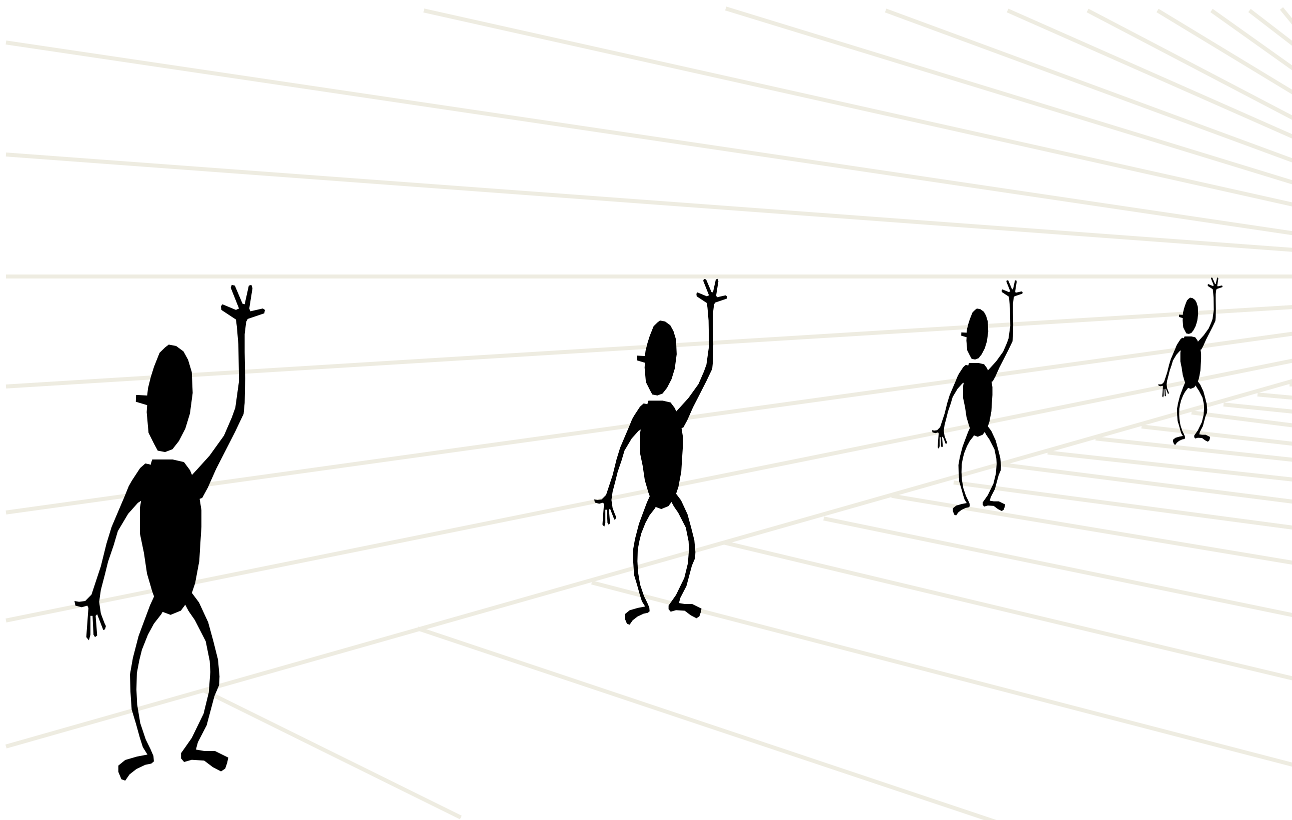
Perspective cues



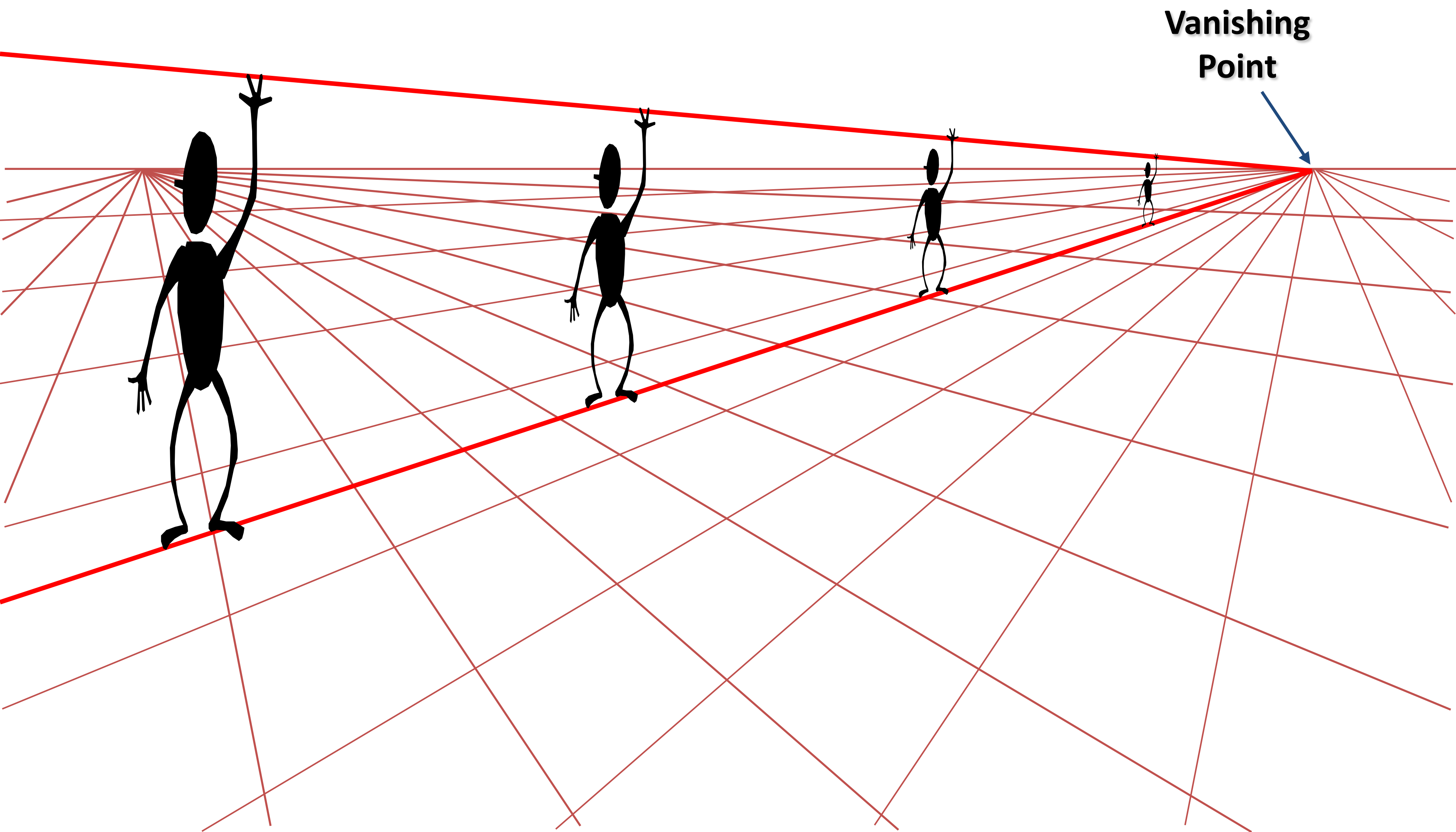
Perspective cues



Perspective cues



Comparing heights



Measuring height

How high is the camera?