Ames Room

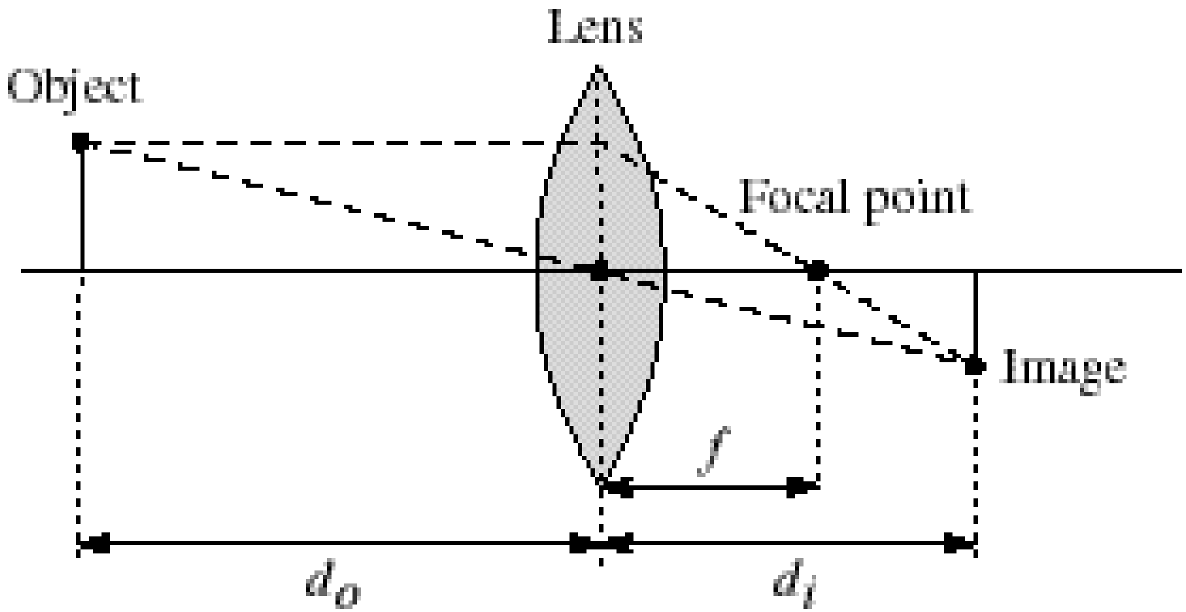
Projective Geometry

CSE 455, Winter 2010

January 27, 2010

Review from last time

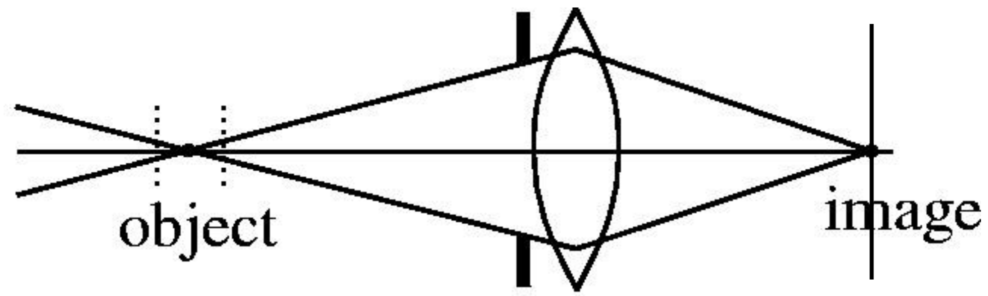
Lenses



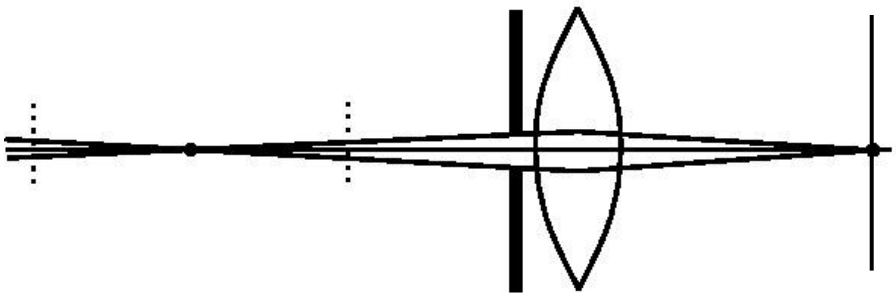
- A lens focuses parallel rays onto a single focal point
- Thin lens equation:

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$$

Depth of field



$f/32$

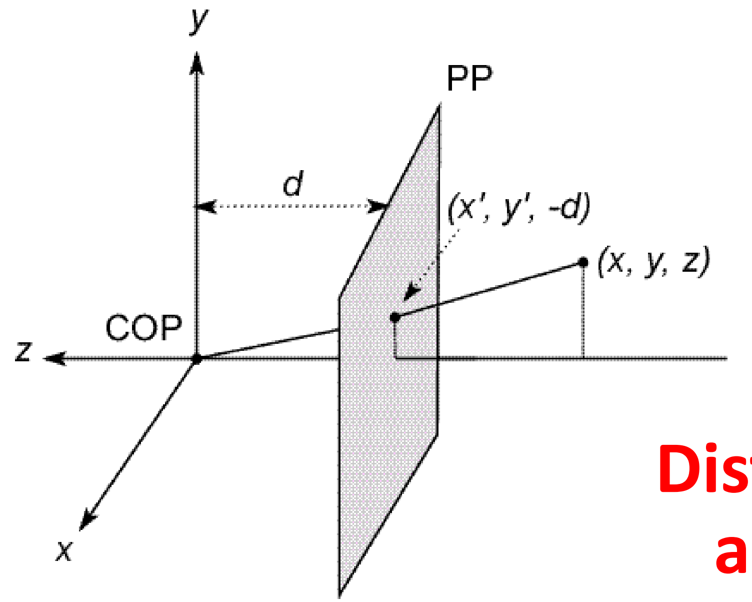


$f/5.6$

- Changing the aperture size affects depth of field
  - A smaller aperture increases the range in which the object is approximately in focus

Flower images from Wikipedia http://en.wikipedia.org/wiki/Depth_of_field

Modeling projection



Distant objects are smaller

- Projection equations

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z},\ -d)$$

- We get the projection by throwing out the last coordinate:

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$$

Projection

- Perspective Projection

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/\boldsymbol{f} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/\boldsymbol{f} \end{bmatrix} \Rightarrow (-\boldsymbol{f}\frac{x}{z}, \quad -\boldsymbol{f}\frac{y}{z})$$

divide by third coordinate

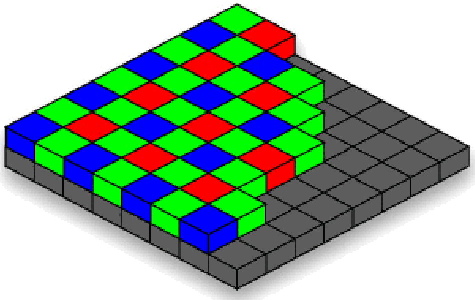
- Orthographic Project

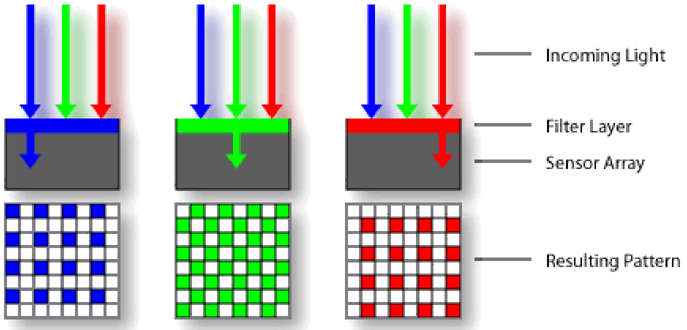
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

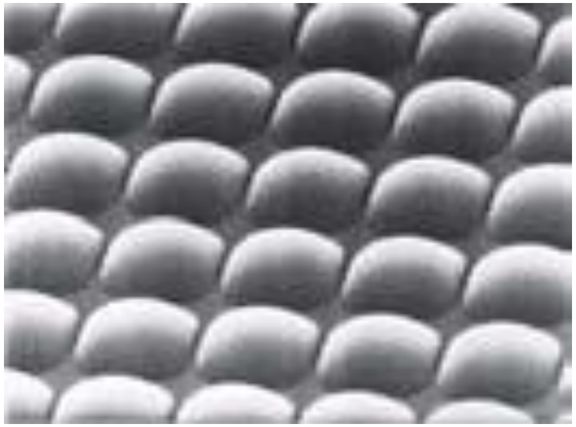
Distortions/Artifacts

Digital Cameras

- Basic process:
  - photons hit a detector
  - the detector becomes charged
  - the charge is read out as brightness
- Sensor types:
  - CCD (charge-coupled device)
  - CMOS







Today

- More on Cameras
- Projective Geometry

Issues with digital cameras

- Noise
  - big difference between consumer vs. SLR-style cameras
  - low light is where you most notice noise
- Compression
  - creates artifacts except in uncompressed formats (tiff, raw)
- Color
  - color fringing artifacts from Bayer patterns
- Blooming
  - charge overflowing into neighboring pixels
- In-camera processing
  - oversharpening can produce halos
- Interlaced vs. progressive scan video
  - even/odd rows from different exposures
- Are more megapixels better?
  - requires higher quality lens
  - noise issues

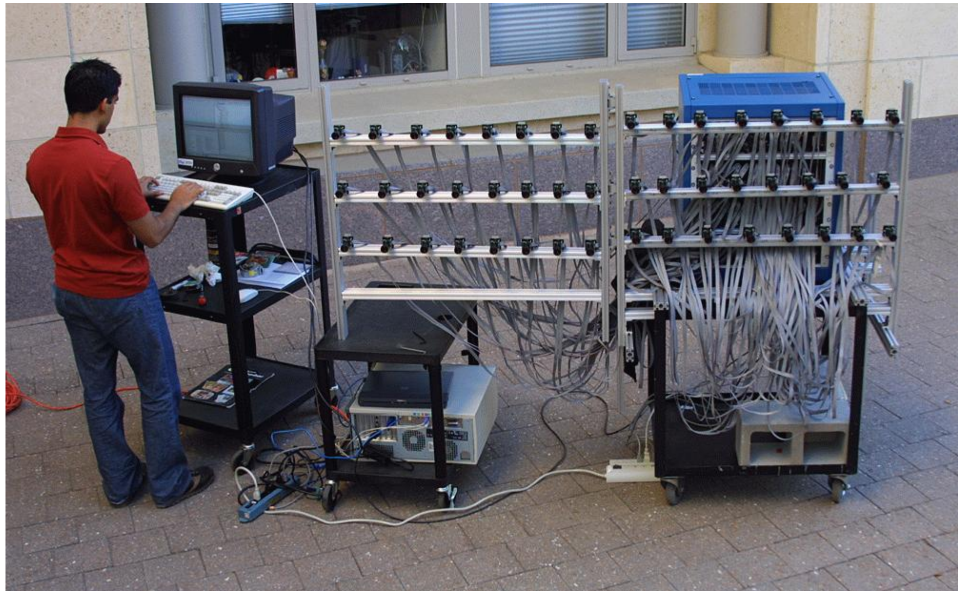
More info online, e.g.,

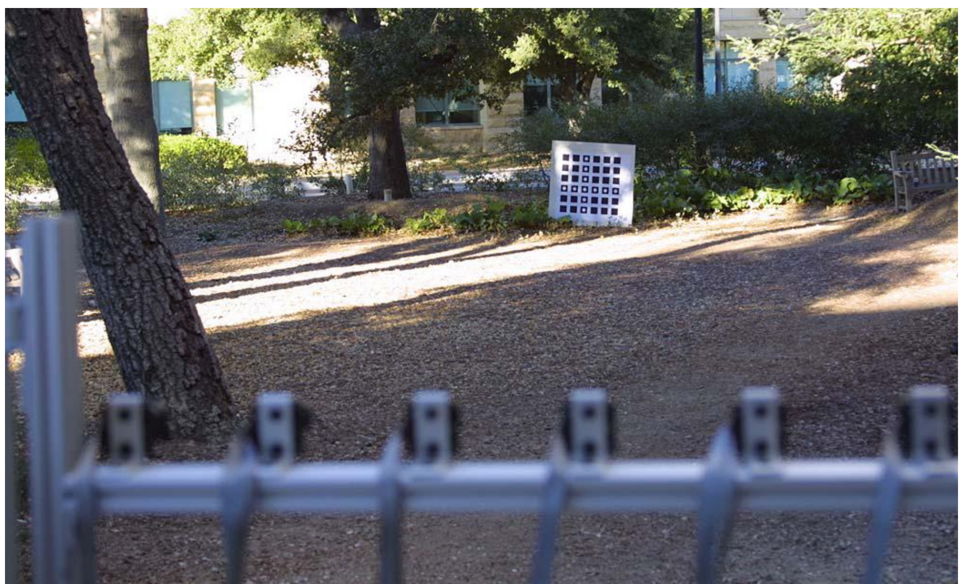
http://electronics.howstuffworks.com/digital-camera.htm

http://www.dpreview.com/

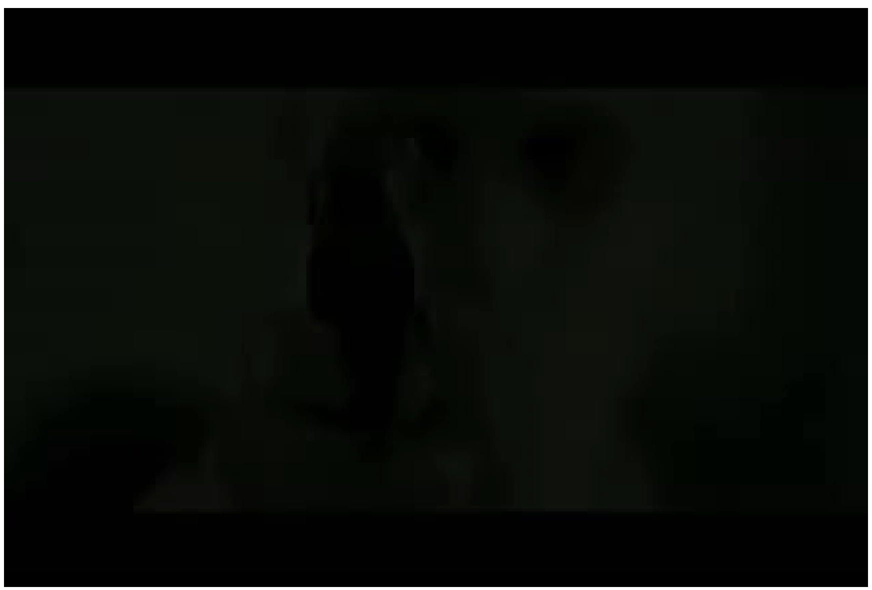
Large Aperture (Shallow Depth of Field)

Synthetic Aperture [Vaish et al.]





Bullet Time (from the Matrix)

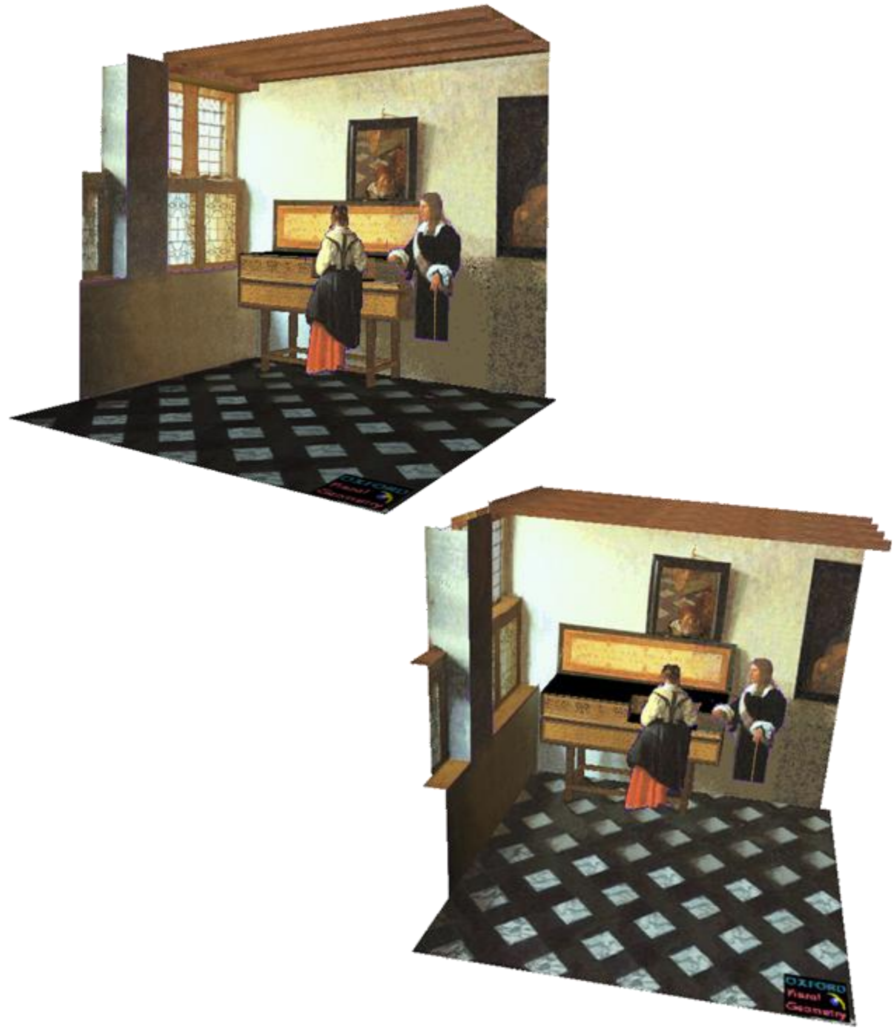


Projective geometry—what's it good for?

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

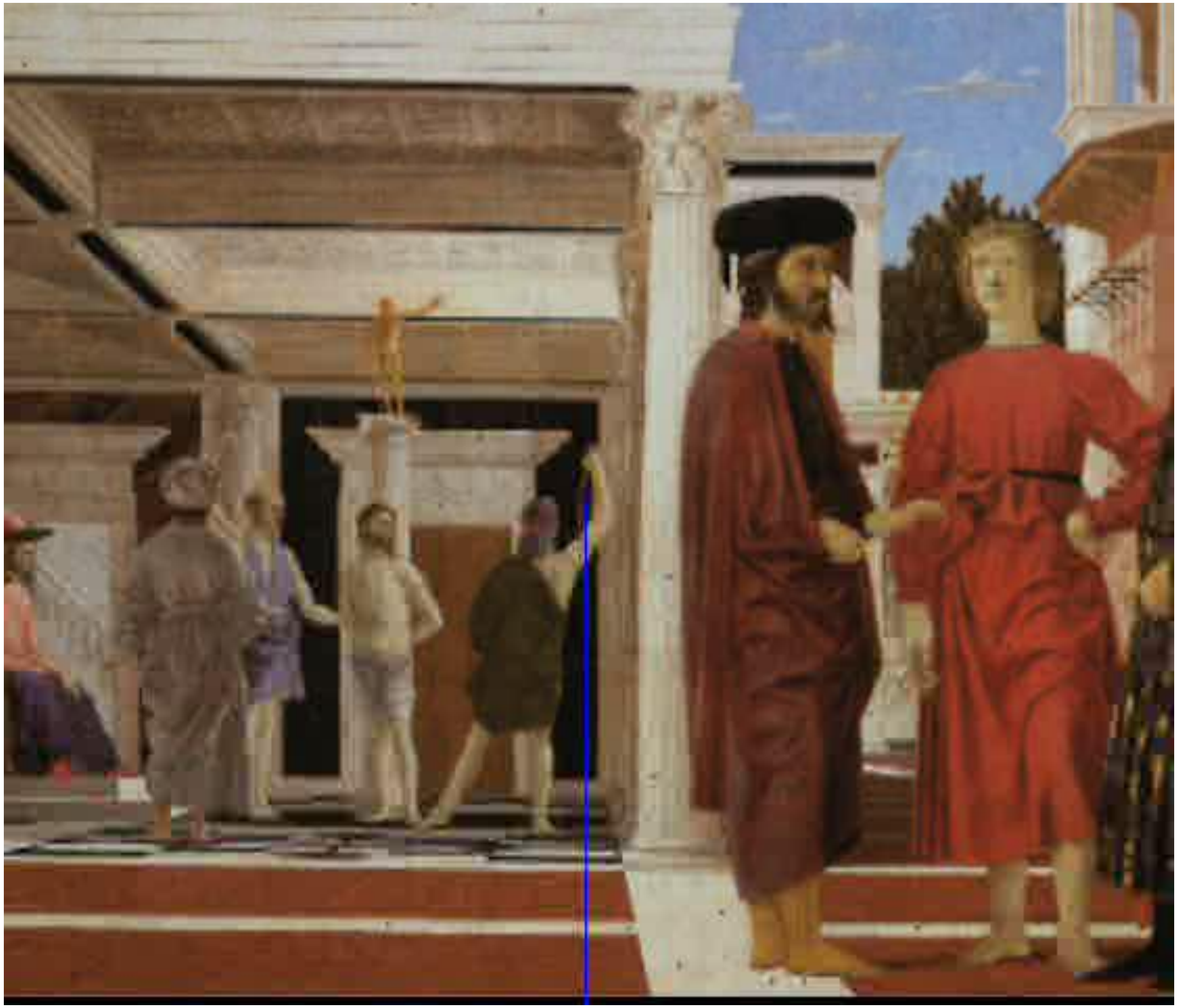
Applications of projective geometry

Vermeer's *Music Lesson*

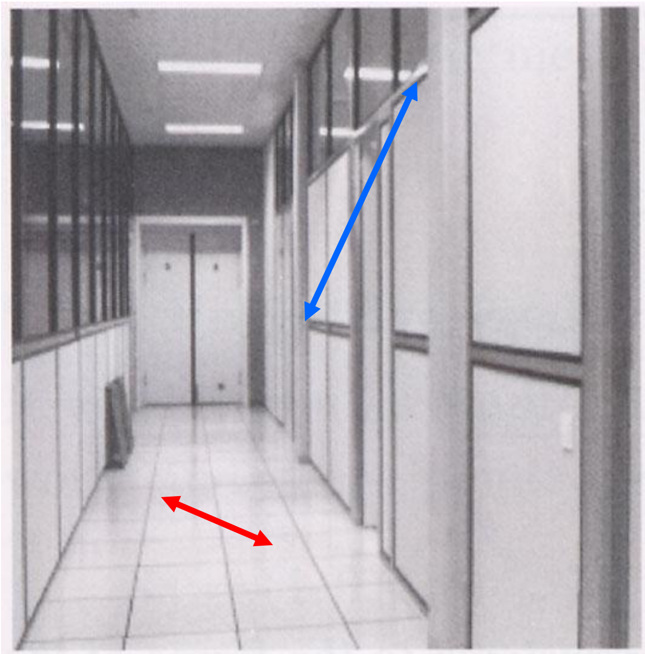


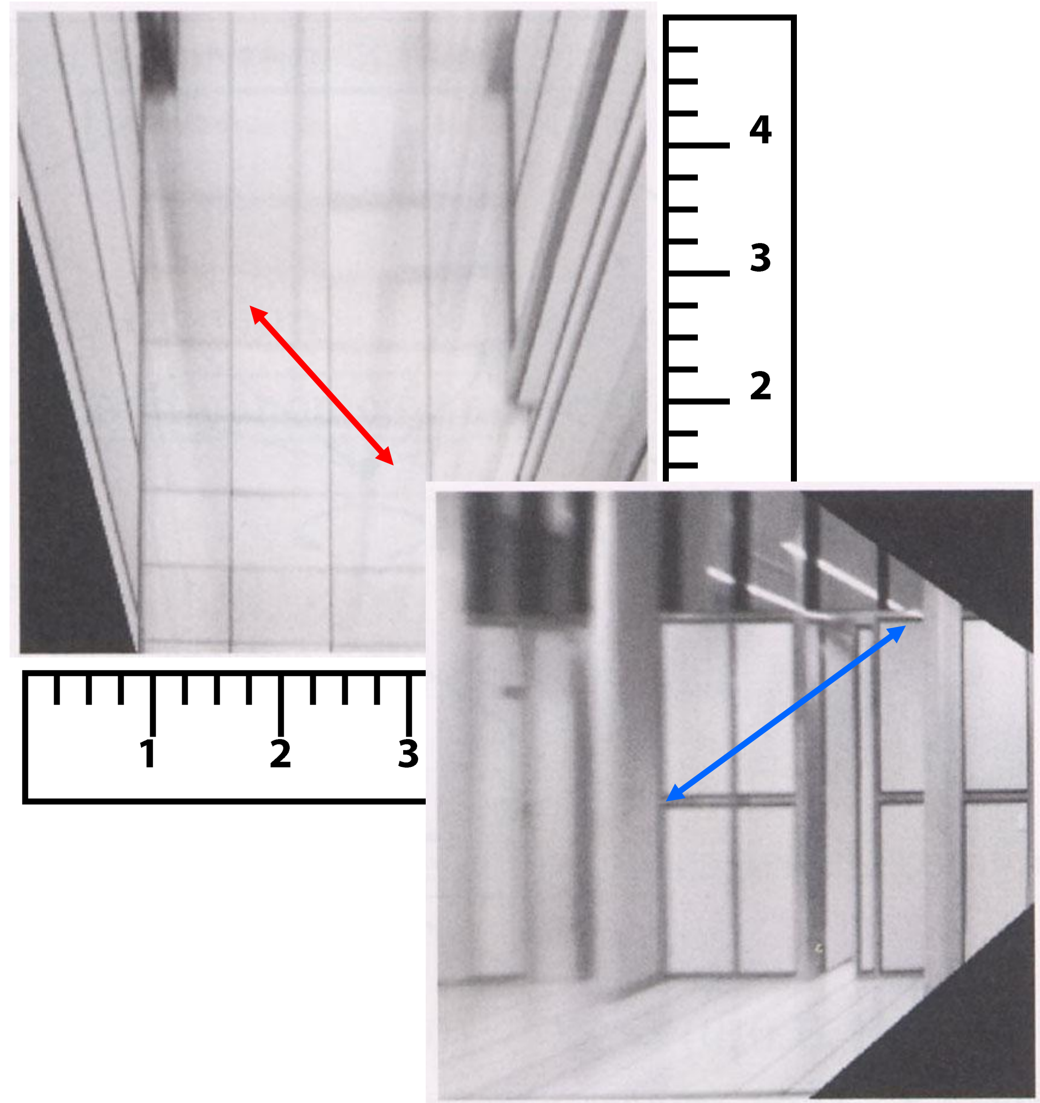
Reconstructions by Criminisi et al.

3D Modeling from a photograph



Measurements on planes

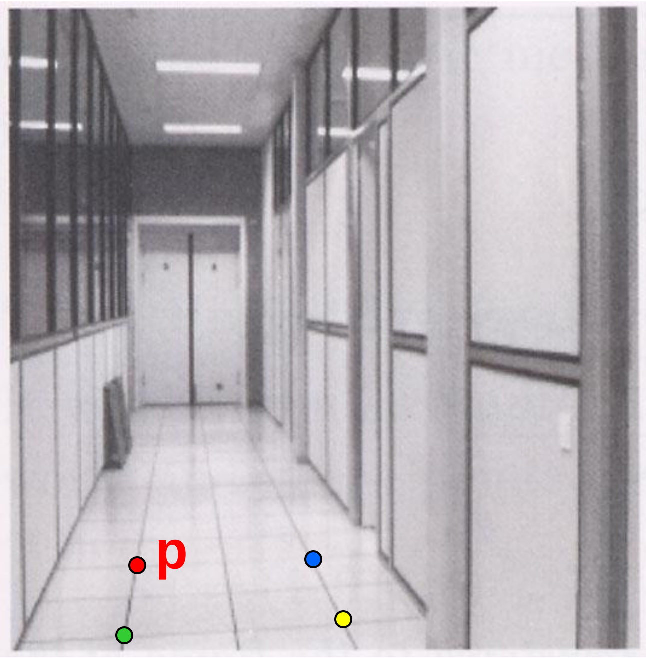


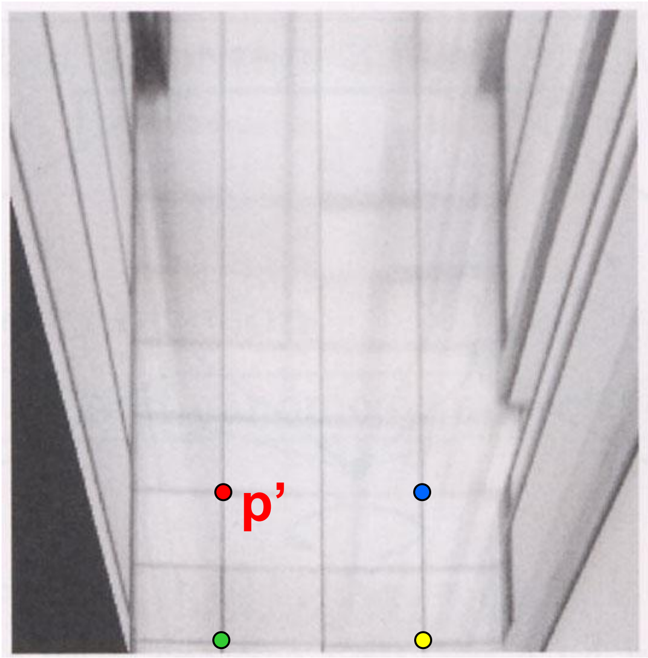


Approach: unwarp then measure

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**?

How many points?

Solving for homographies

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

$$x'_i = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$y'_i = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

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

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

Solving for homographies

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

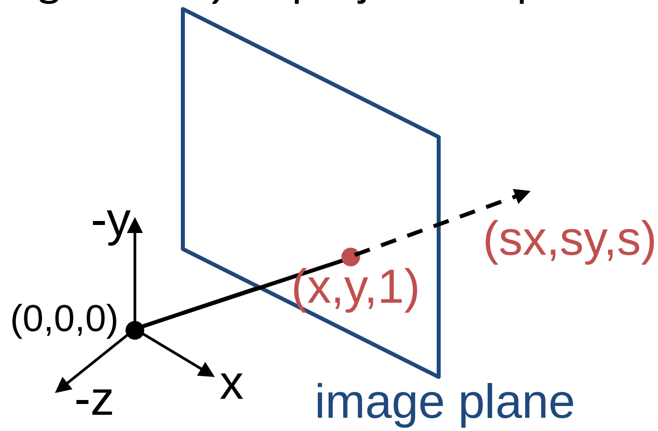
A	**h**	**0**
2n × 9	**9**	**2n**

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

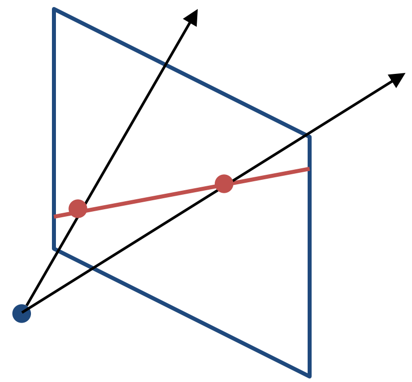
- Why do we need homogeneous coordinates?
  - represent points at infinity, homographies, perspective projection, multi-view relationships
- What is the geometric intuition?
  - a point in the image is a *ray* in projective space



- 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?



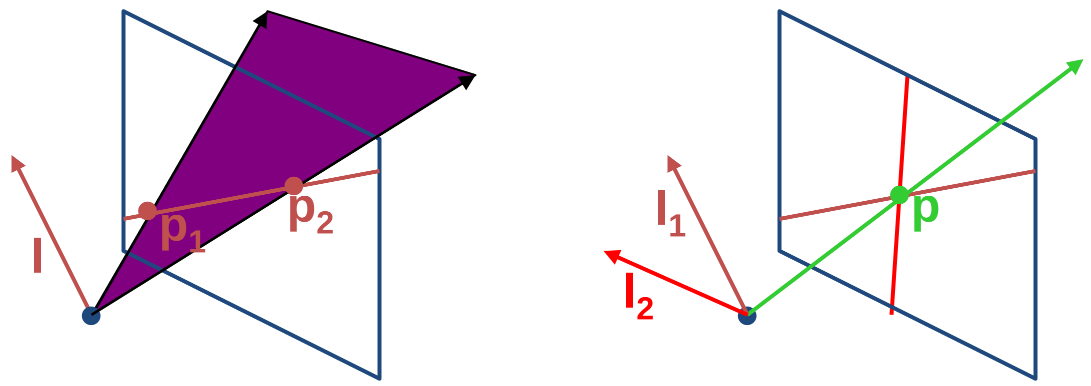
- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

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

- 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 $\perp$ 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 $\perp$ to $\mathbf{p_1}$ and $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- **l** is the plane normal

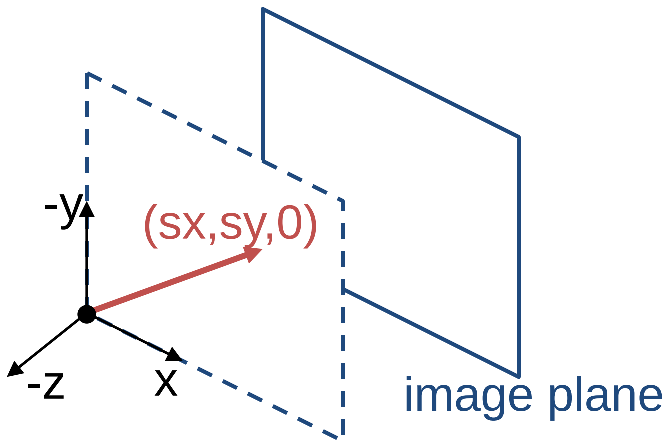
What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

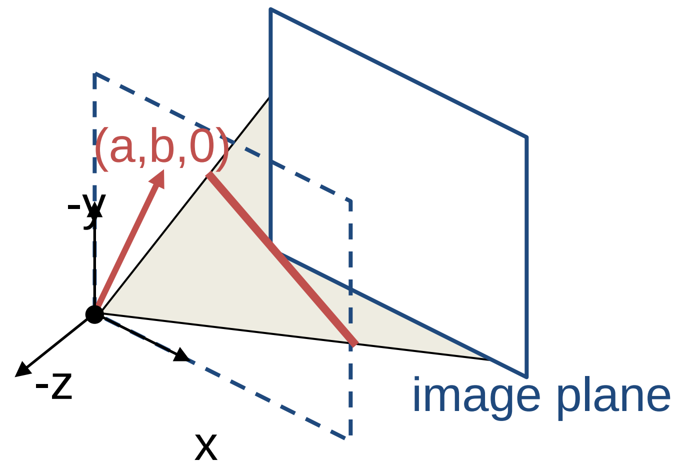
- **p** is $\perp$ to $\mathbf{l_1}$ and $\mathbf{l_2}$ $\Rightarrow$ $\mathbf{p} = \mathbf{l_1} \times \mathbf{l_2}$

Points and lines are *dual* in projective space

- 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 ≅ (x, y, 0) – parallel to image plane
  - It has infinite image coordinates

Ideal line

- l ≅ (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: $\mathbf{p}' = \mathbf{H}\mathbf{p}$
  - To transform a line: $\mathbf{l}\mathbf{p} = 0 \rightarrow \mathbf{l}'\mathbf{p}' = 0$
    - $0 = \mathbf{l}\mathbf{p} = \mathbf{l}\mathbf{H}^{-1}\mathbf{H}\mathbf{p} = \mathbf{l}\mathbf{H}^{-1}\mathbf{p}' \Rightarrow \mathbf{l}' = \mathbf{l}\mathbf{H}^{-1}$
    - 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**: **P' = TP, N' = N** $\mathbf{T}^{-1}$

3D to 2D: "perspective" projection

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

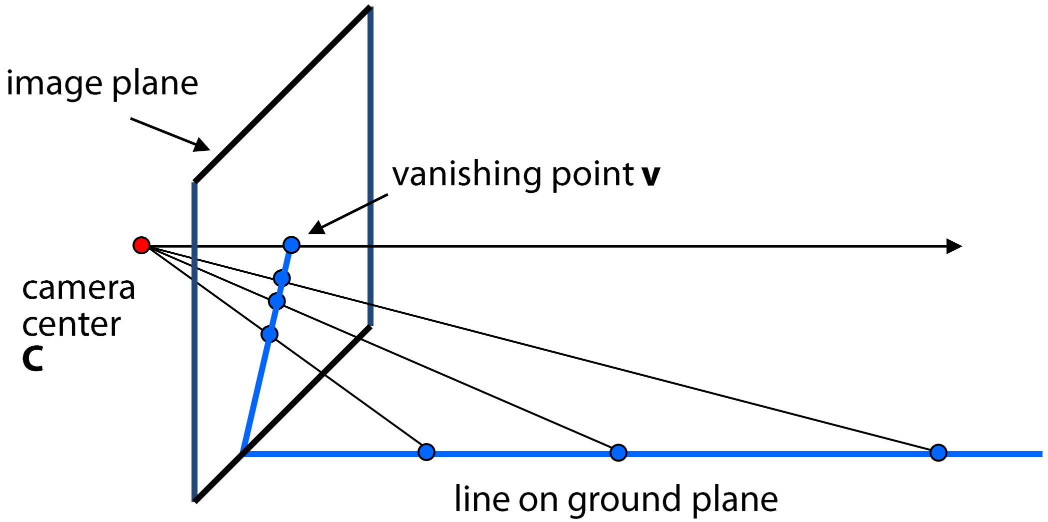
What is *not* preserved under perspective projection?

What IS preserved?

Geometric properties of perspective projection

- Geometric properties of perspective projection
  - Points go to points
  - Lines go to lines
  - Planes go to whole image or half-plane
  - Polygons go to polygons
  - Angles & distances not preserved
- Degenerate cases:
  - line through focal point yields point
  - plane through focal point yields line

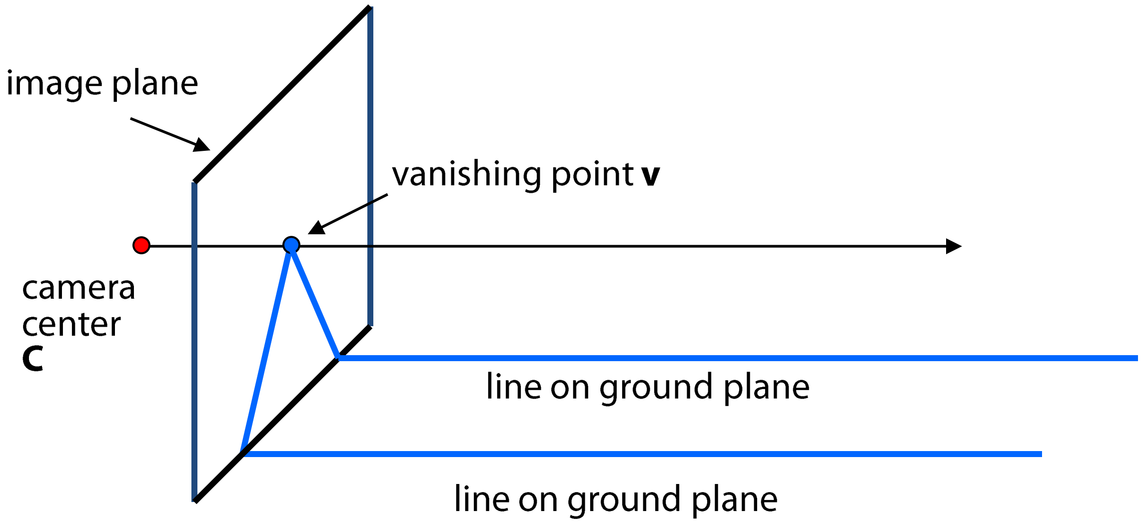
Vanishing points



Vanishing point

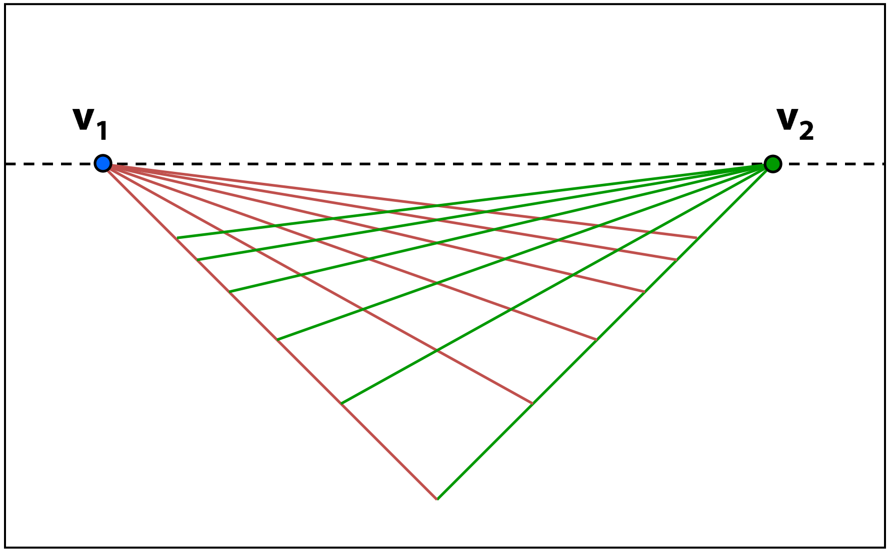
- projection of a point at infinity

Vanishing points



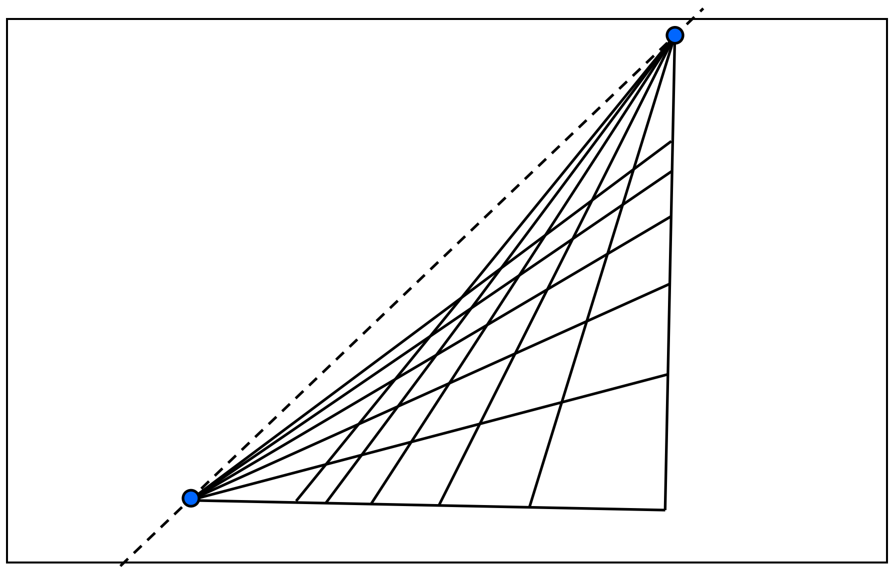
- Properties
  - Any two parallel lines have the same vanishing point **v**
  - The ray from **C** through **v** is parallel to the lines
  - An image may have more than one vanishing point
    - in fact every pixel is a potential vanishing point

Vanishing lines



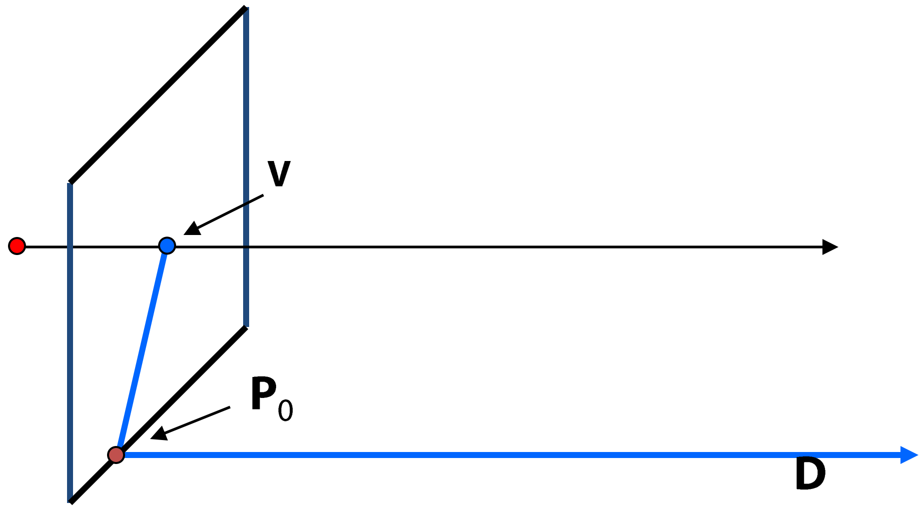
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of vanishing points from lines on the same plane is the *vanishing line*
    - For the ground plane, this is called the *horizon*

Vanishing lines



- Multiple Vanishing Points
  - Different planes define different vanishing lines

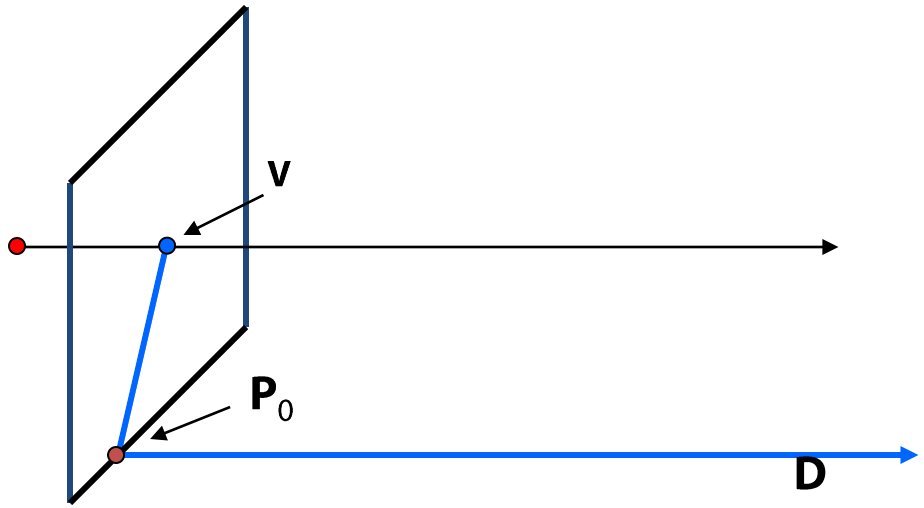
Computing vanishing points



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

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

Computing vanishing points

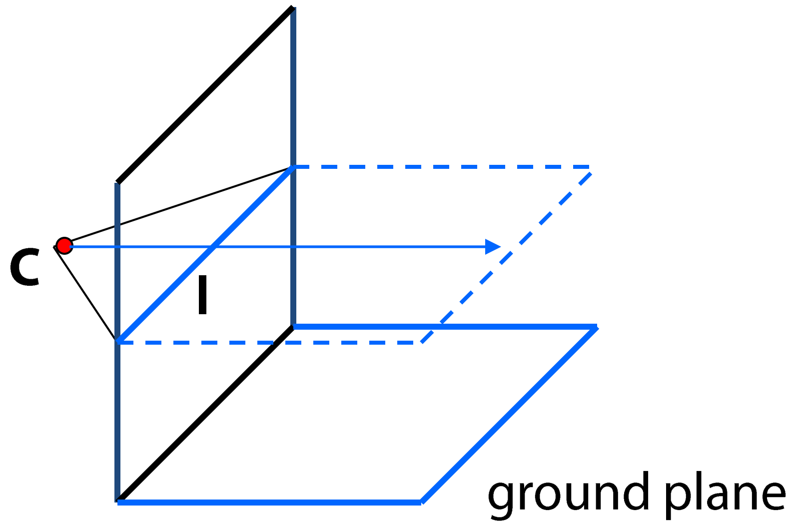


$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

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

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

Above or Below?

Fun with vanishing points