

6. Camera and Images

Image formation, Camera models and Camera calibration

3D through images

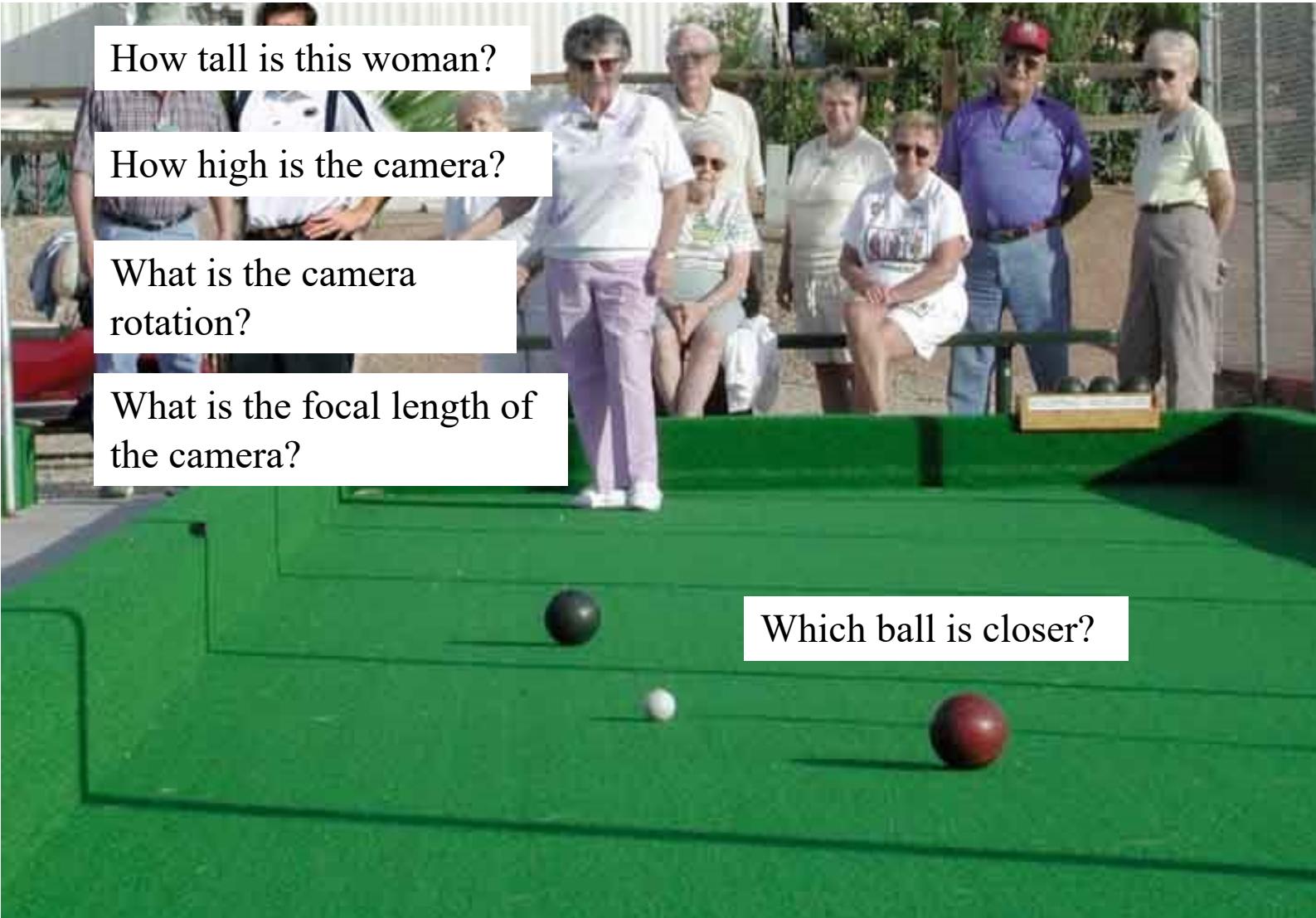
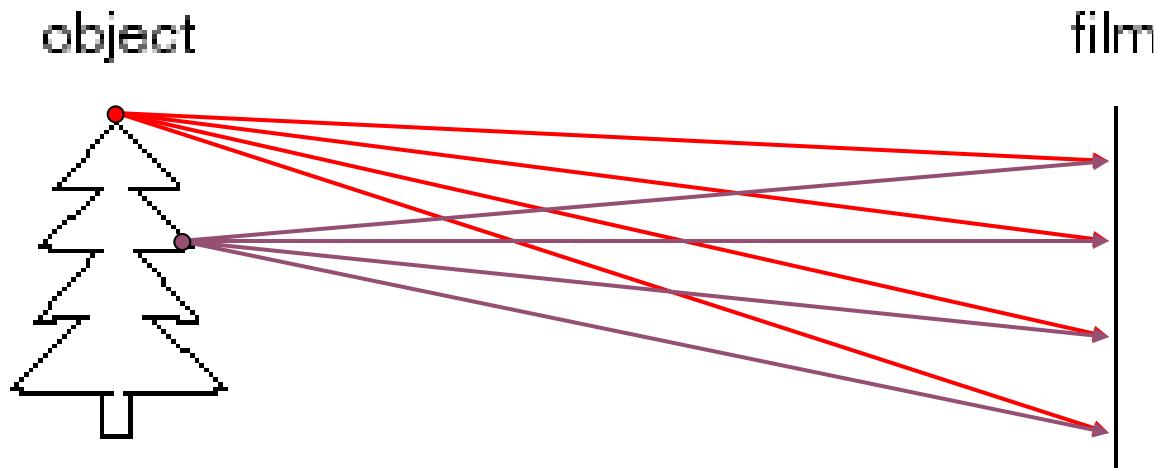


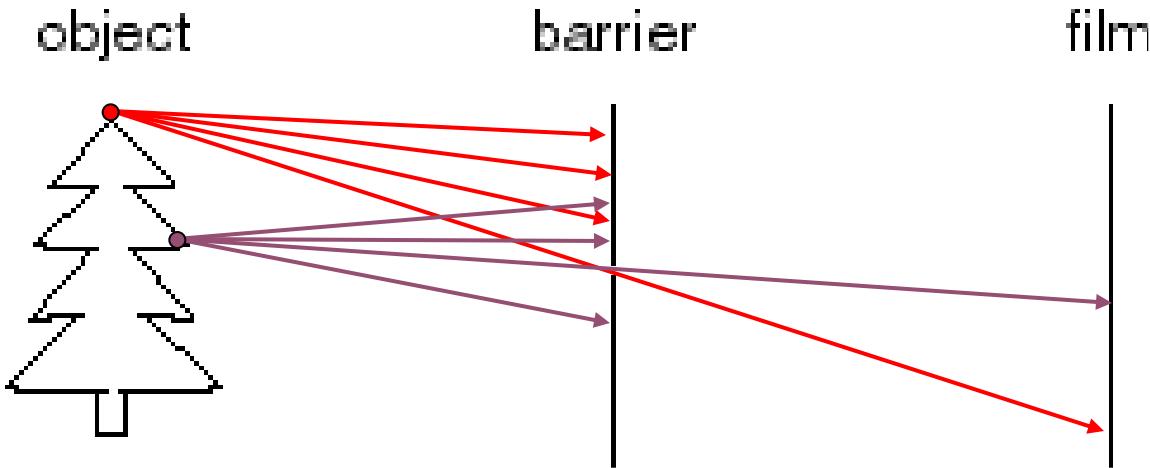
Image formation



Let's design a camera

- Idea 1: put a piece of film in front of an object
- Do we get a reasonable image?

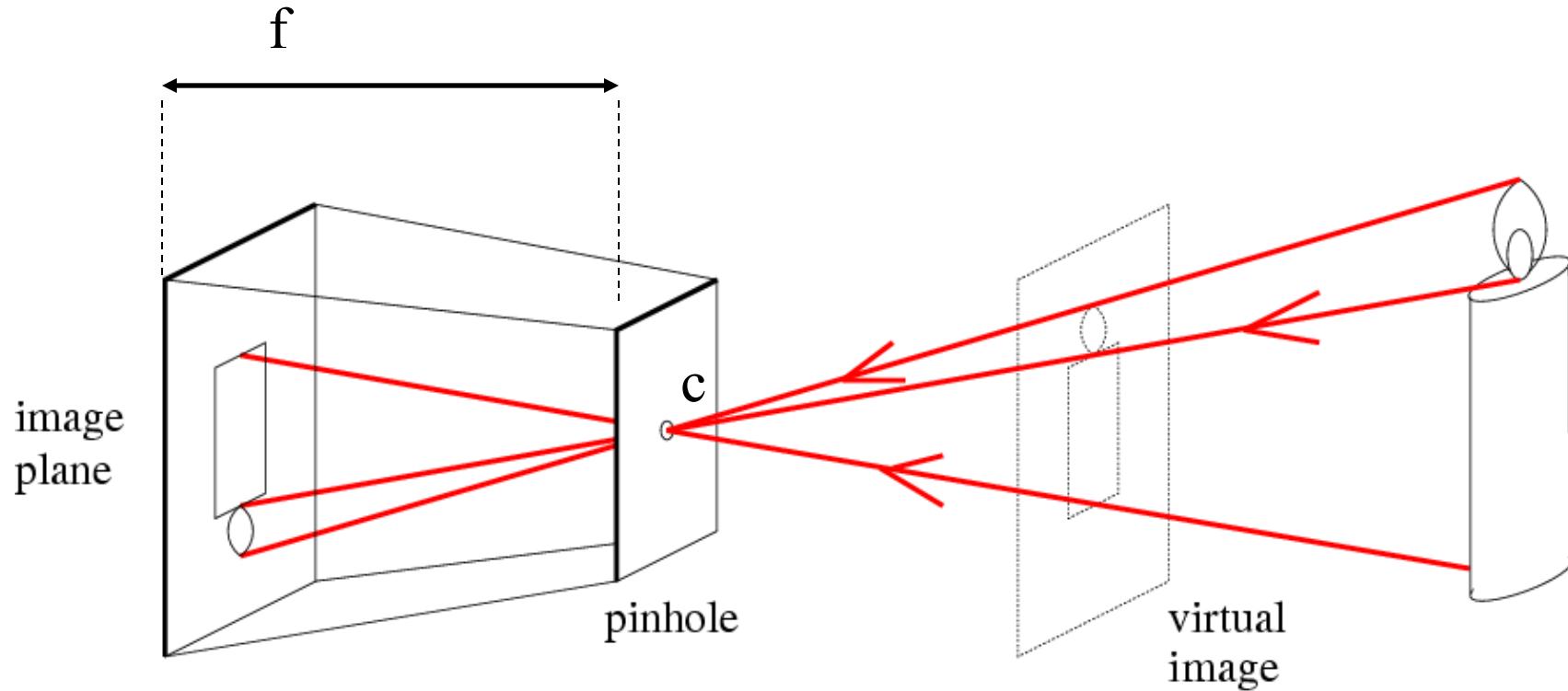
Pinhole camera



Idea 2: add a barrier to block off most of the rays

- This reduces blurring
- The opening known as the **aperture**

Pinhole camera

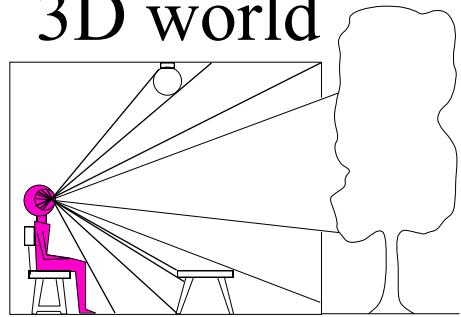


f = focal length

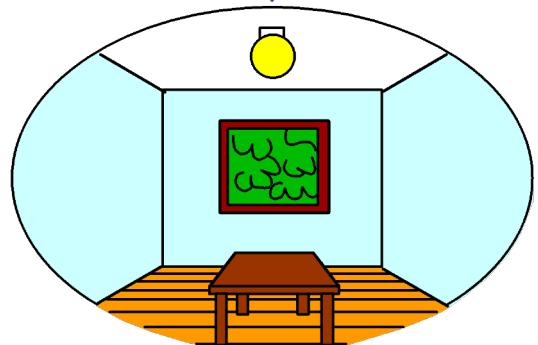
c = center of the camera

Dimensionality Reduction Machine (3D to 2D)

3D world



Point of observation



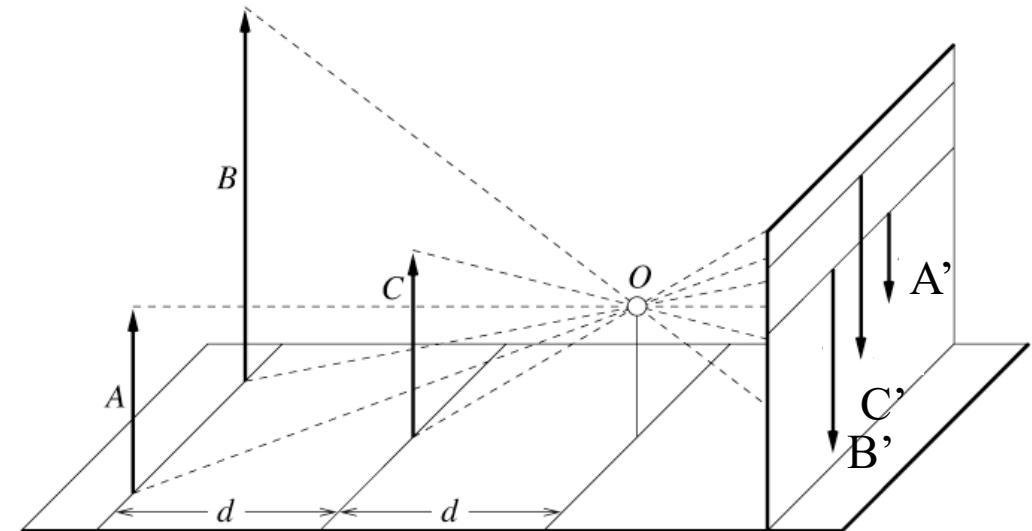
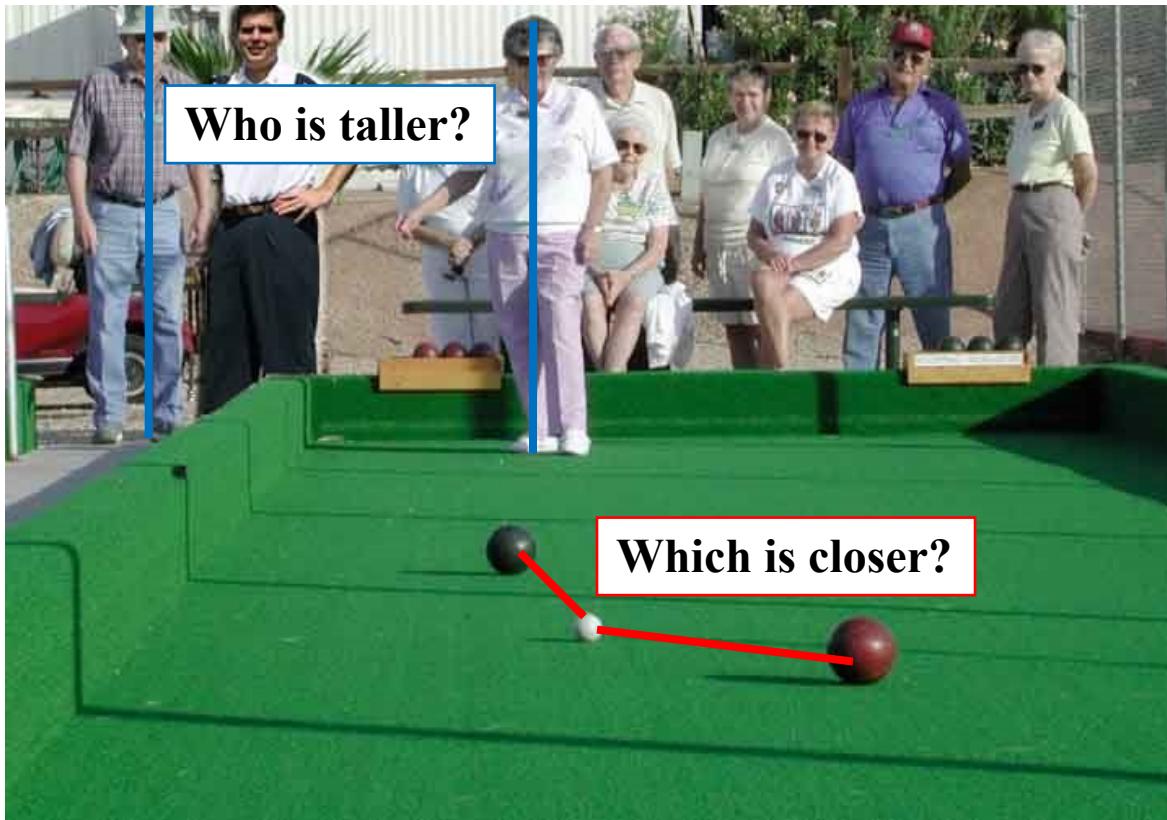
2D image

Projection depends on camera placement



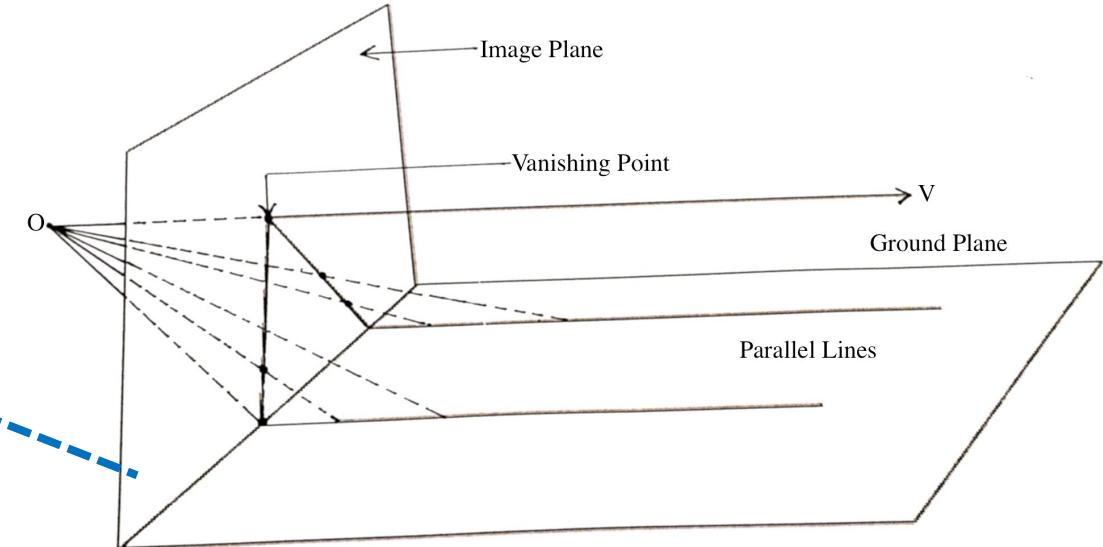
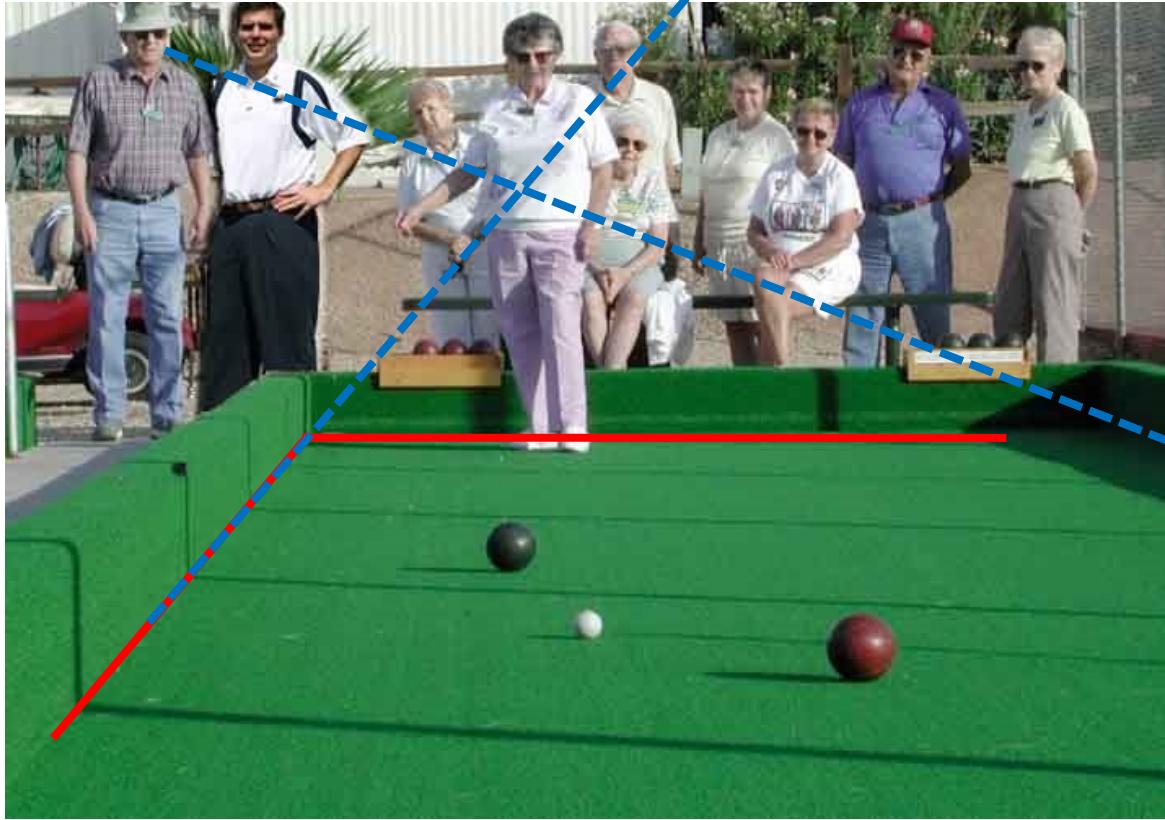
Projective Geometry

Length/distances/geodesics are lost

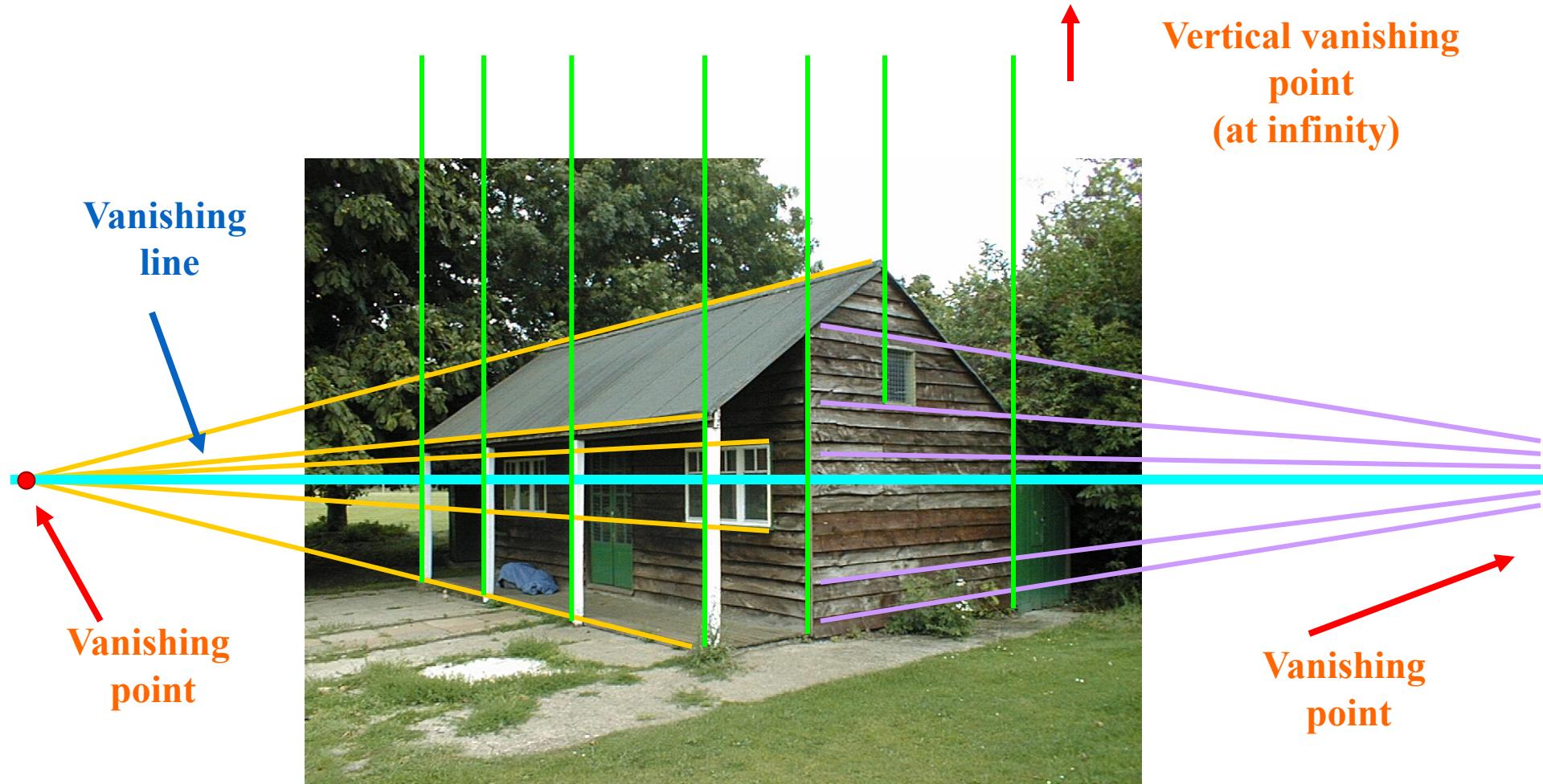


Projective Geometry

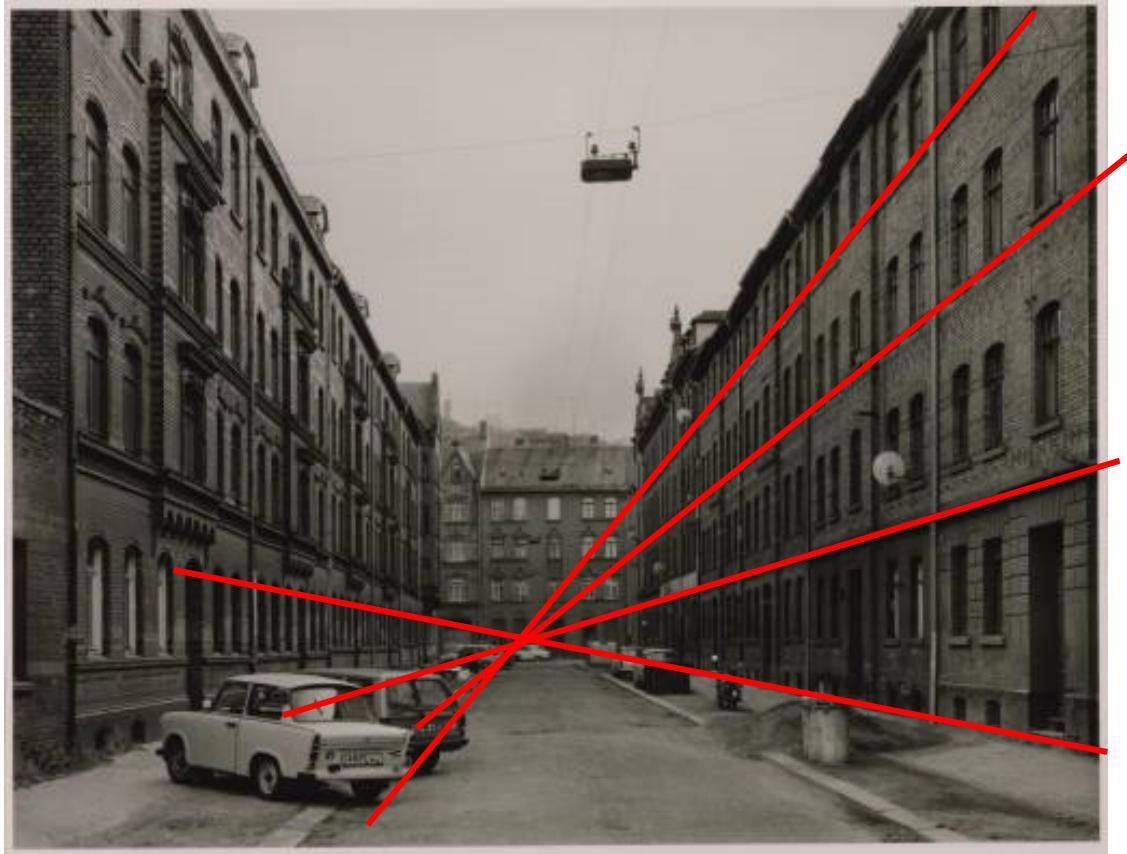
Angles are lost too, straight lines are still straight



Vanishing points and lines



Note on estimating vanishing points



Use multiple lines for better accuracy

... but lines will not intersect at exactly the same point in practice

One solution: take mean of intersecting pairs

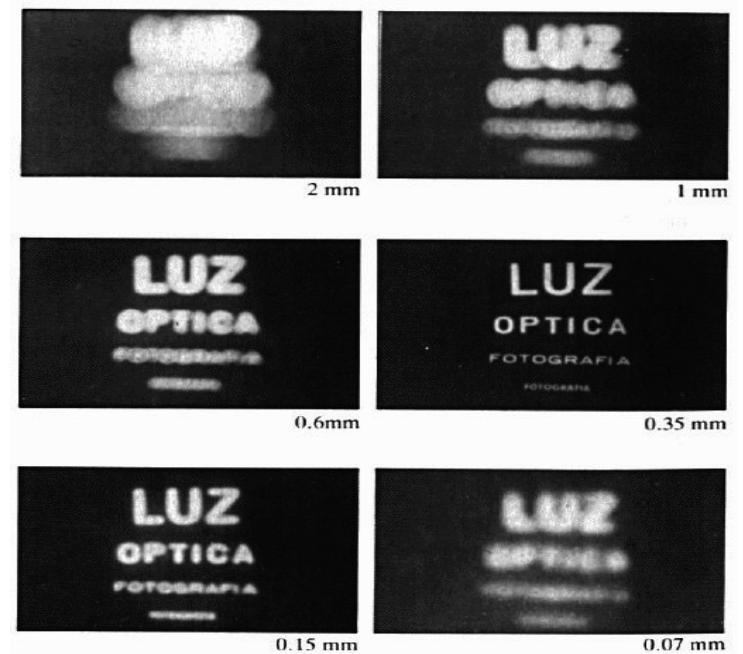
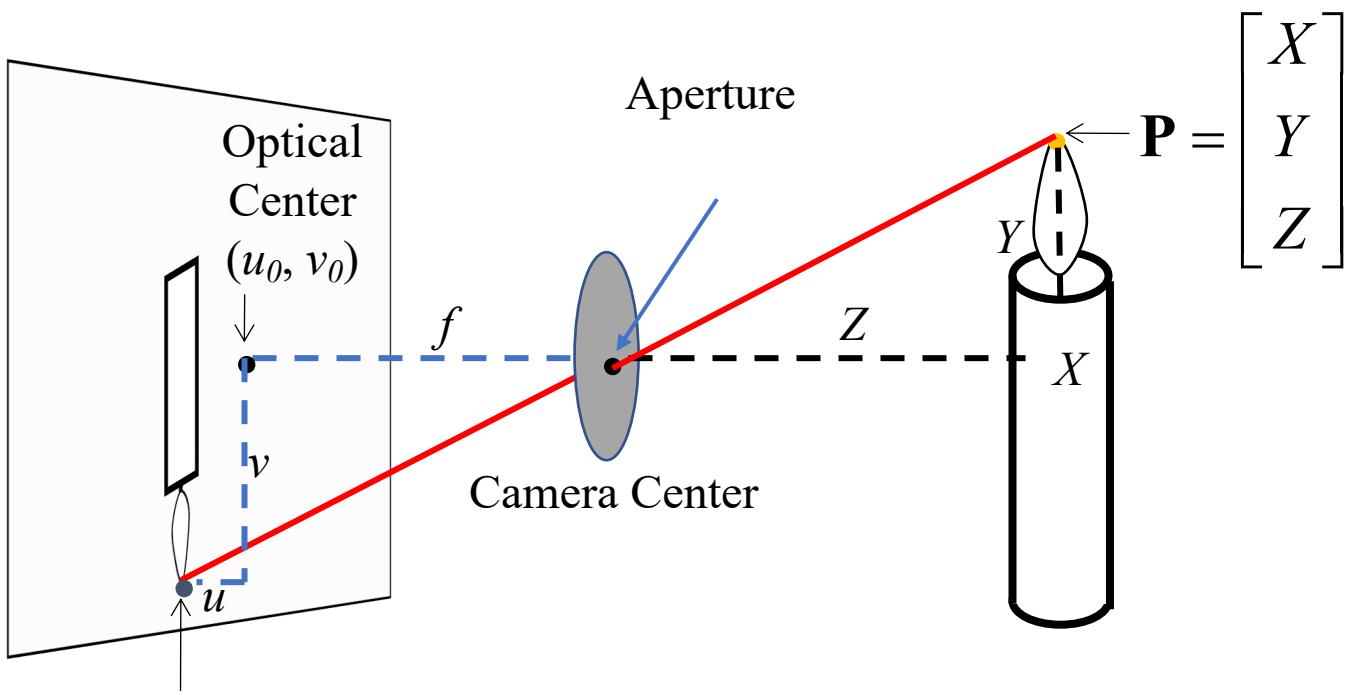
... bad idea!

Instead, minimize angular differences → Angles are stronger indicator of parallelism

Vanishing objects



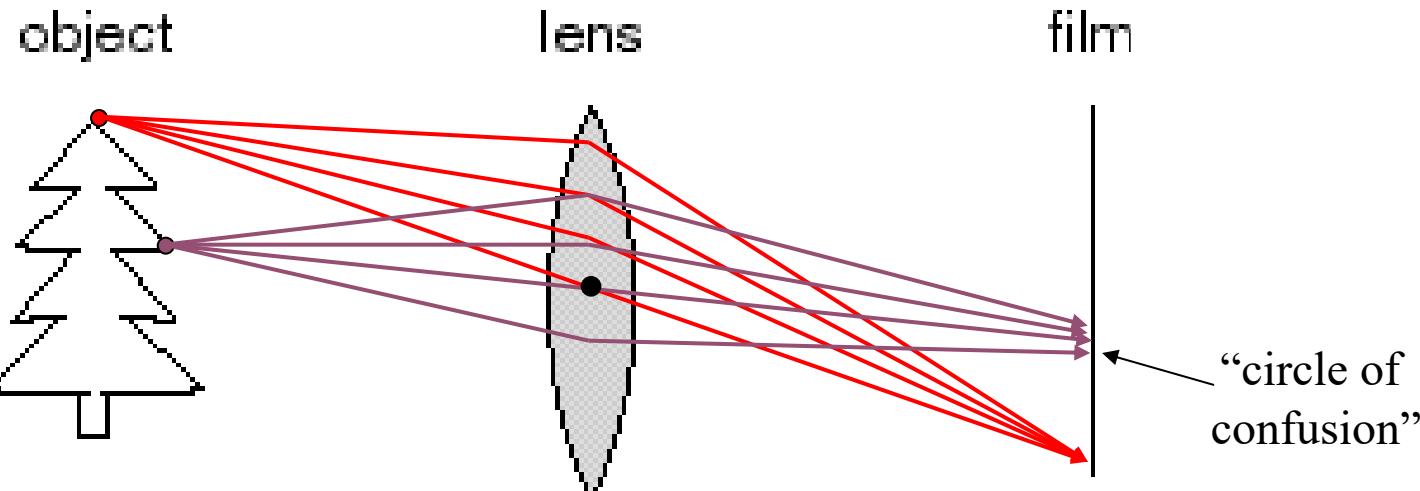
Aperture



Aperture: should be small but not too much

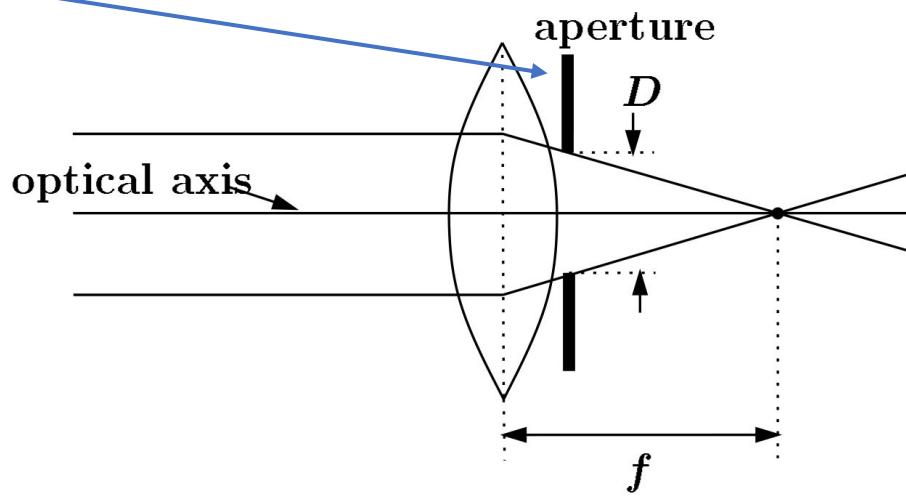
In modern camera, pinhole is replaced by a lens and a shutter is used to control exposure

Adding a lens



There is a specific distance at which objects are “in focus”

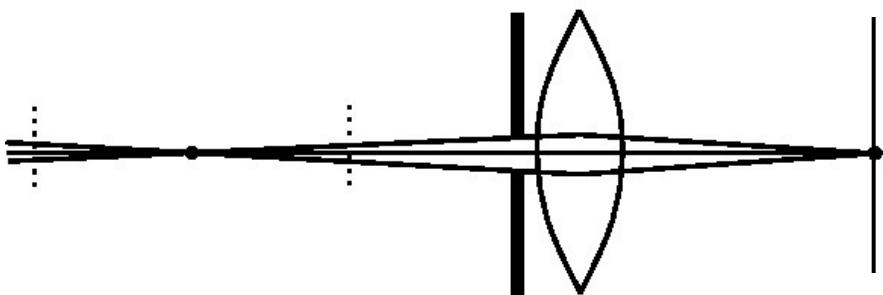
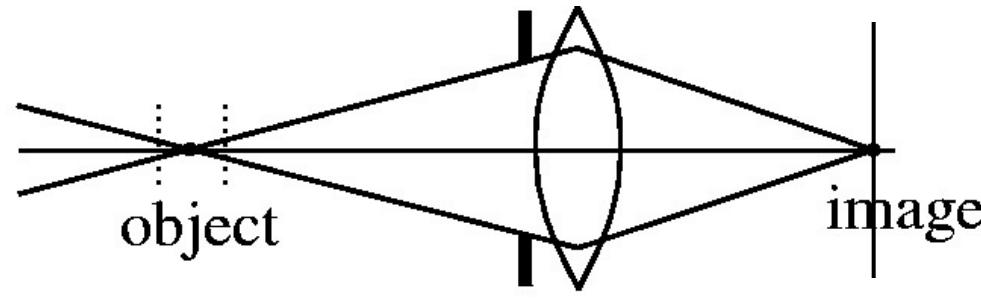
Solution:



A lens focuses parallel rays onto a single focal point

- focal point at a distance f beyond the plane of the lens
- Aperture of diameter D restricts the range of rays

Depth of field



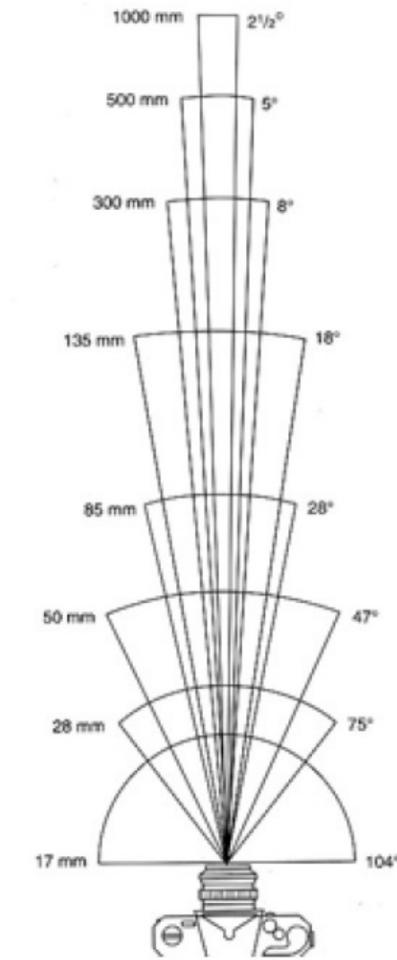
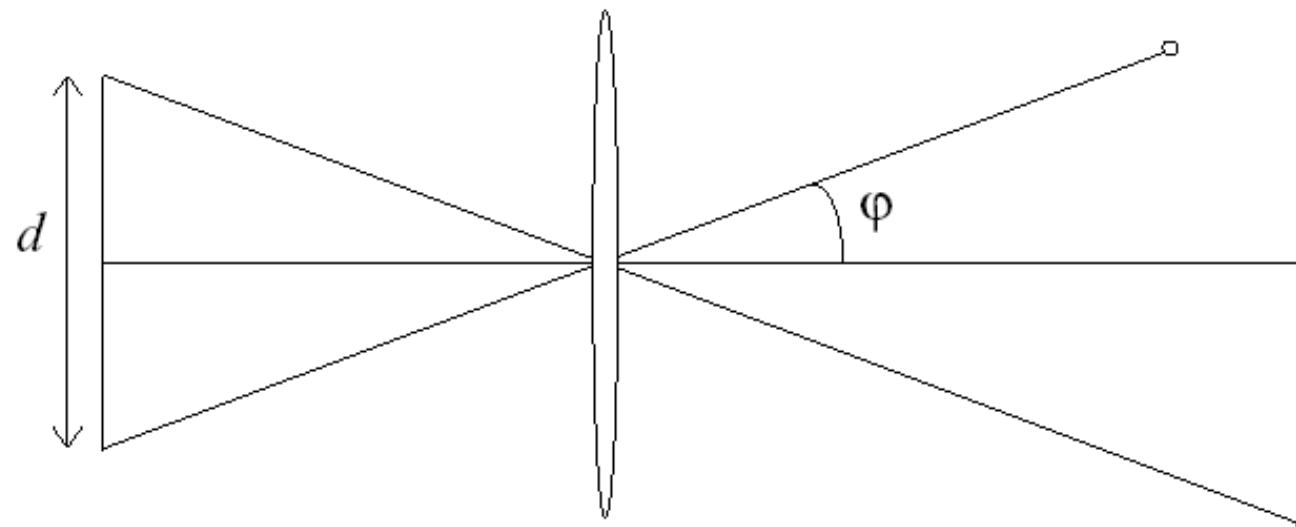
Changing the aperture size or focal length affects depth of field

Relation between field of view and focal length

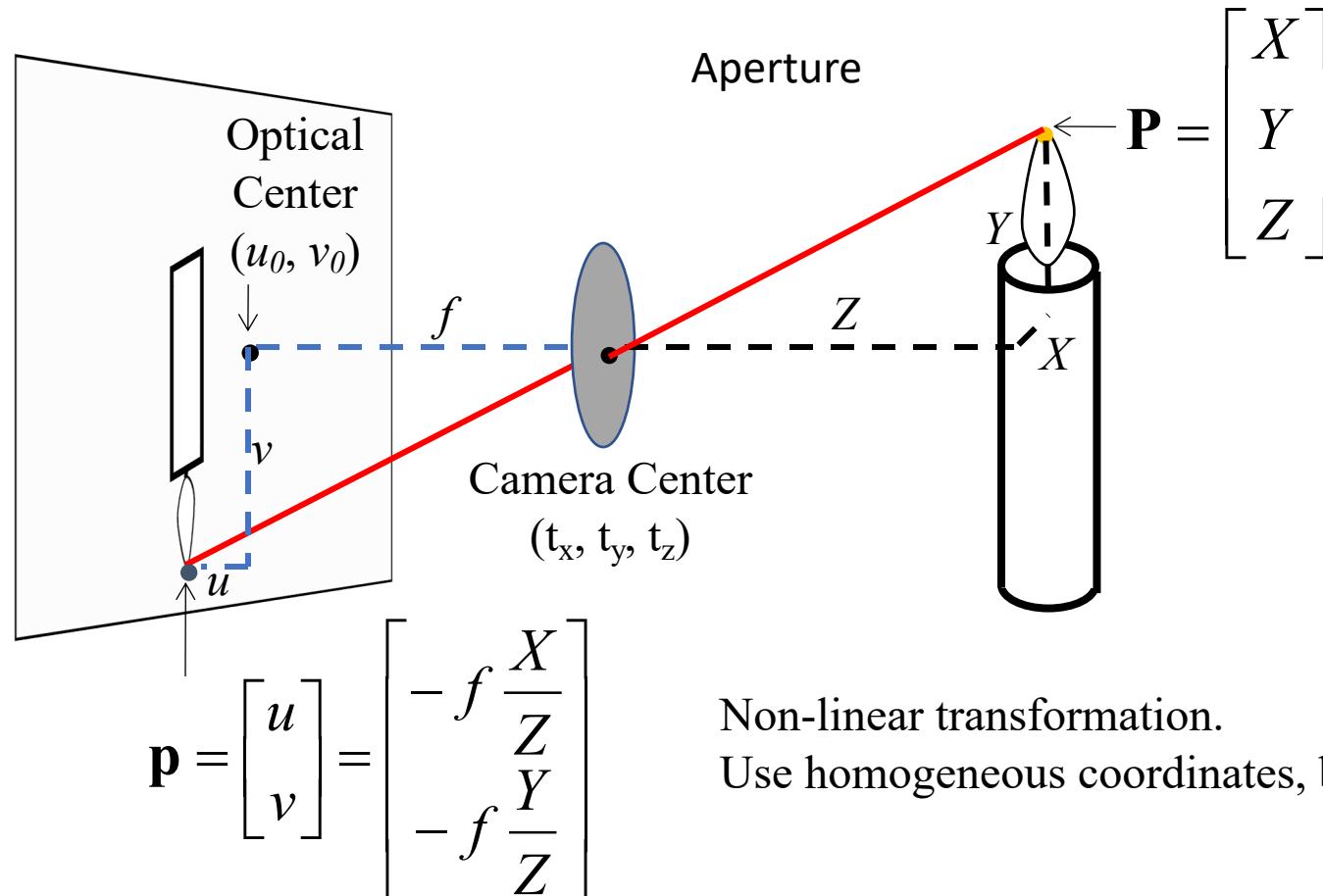
Field of view (angle width)

$$fov = \tan^{-1} \frac{d}{2f}$$

Film/Sensor Width
Focal length



Projection: world coordinates \rightarrow image coordinates



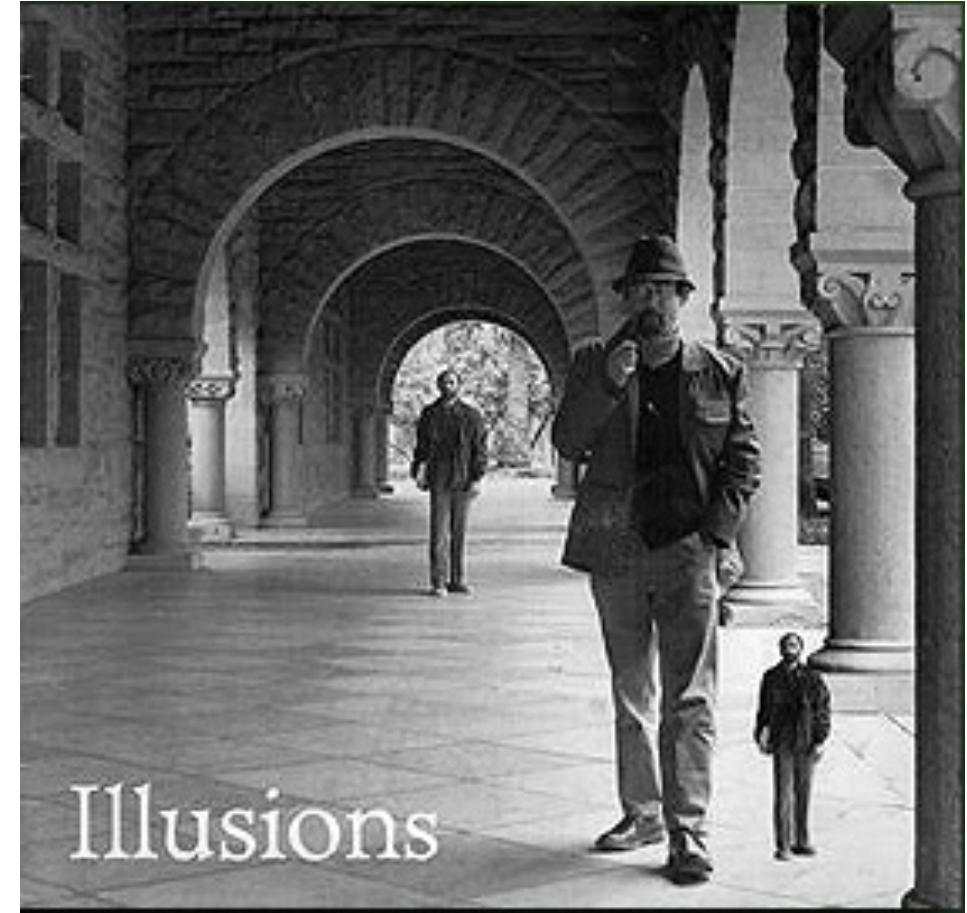
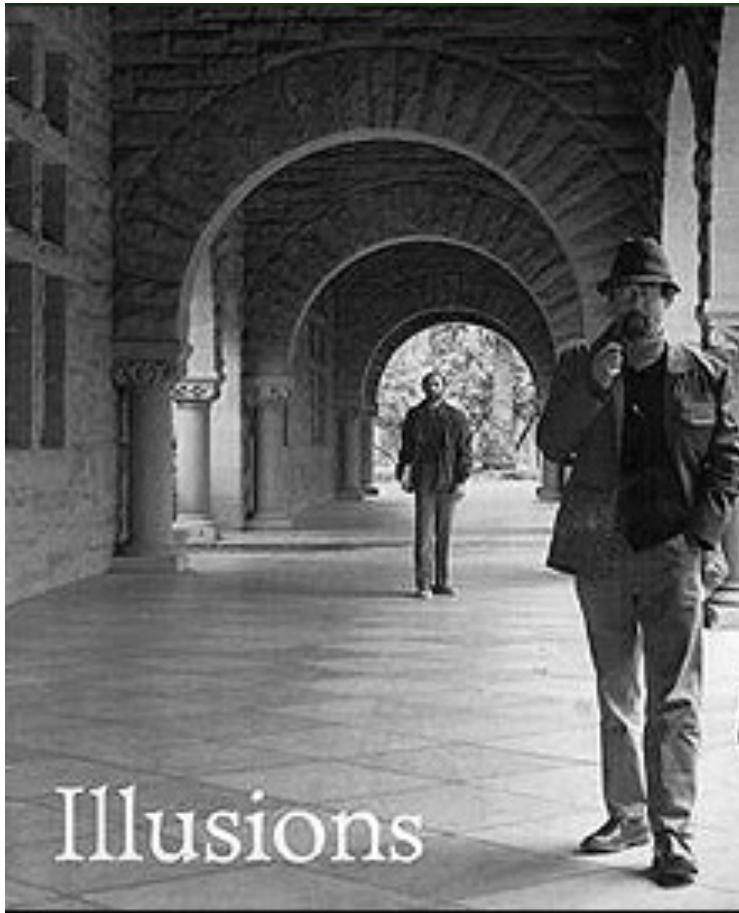
Why Homogeneous Coordinates?



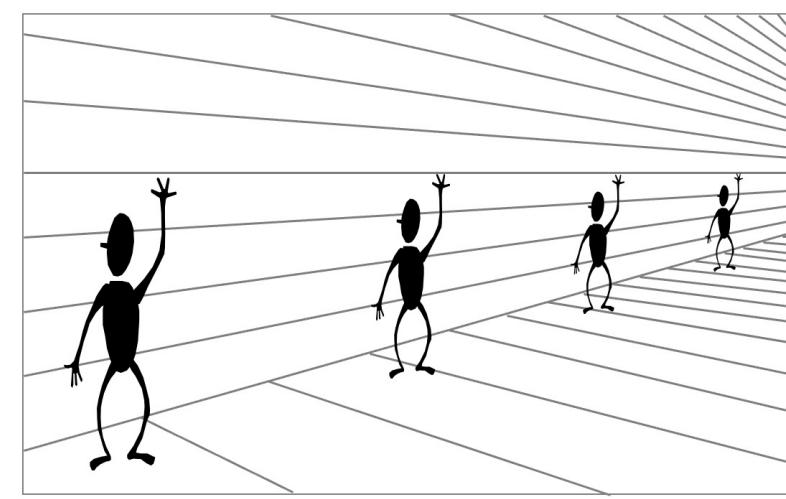
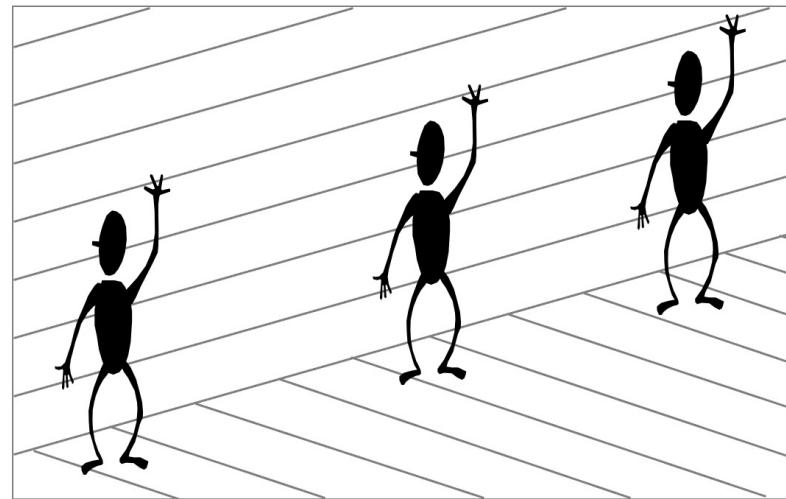
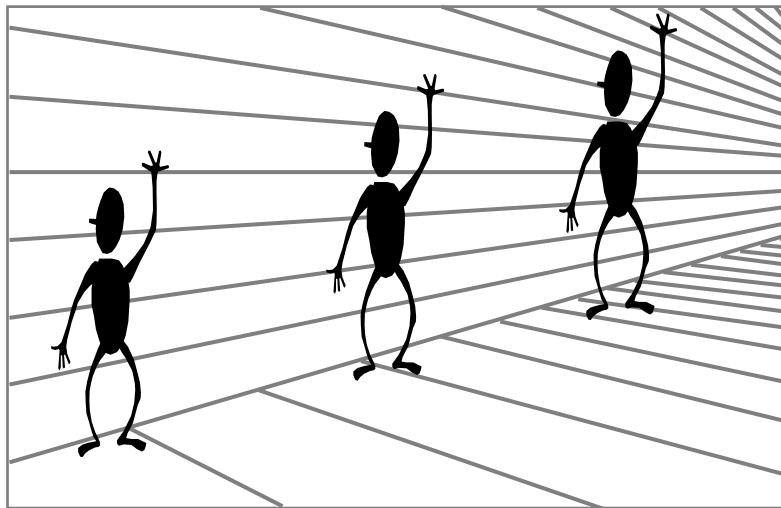
Single view reconstruction: Siggraph 2005

Analyse what happen when camera moves in
(x,y) or in (z) wrt to the object

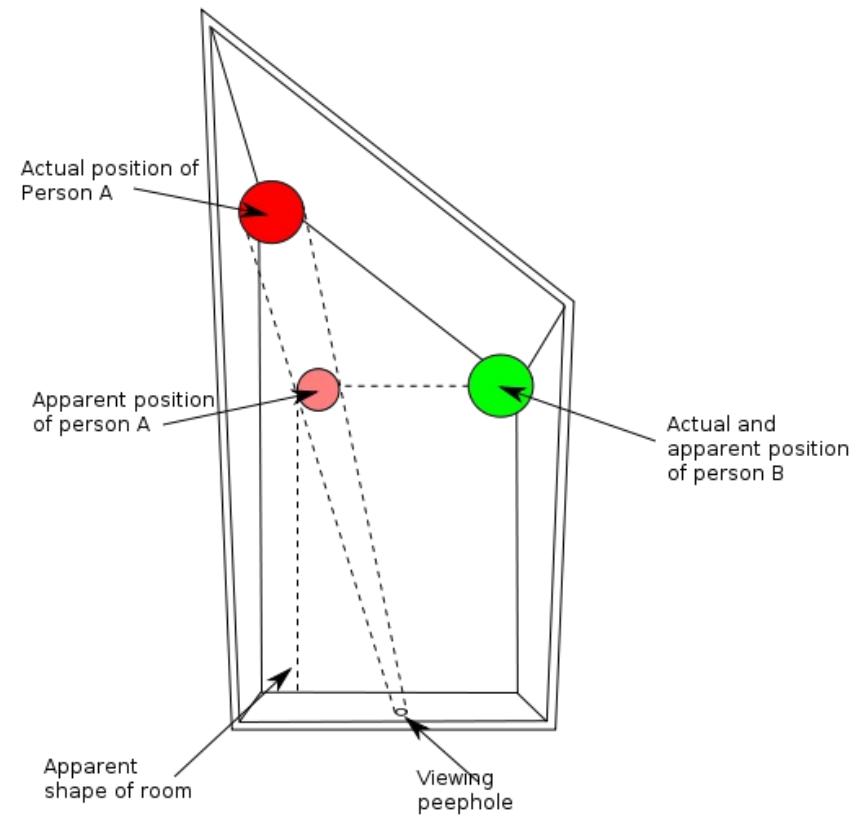
Perspective nature of cameras



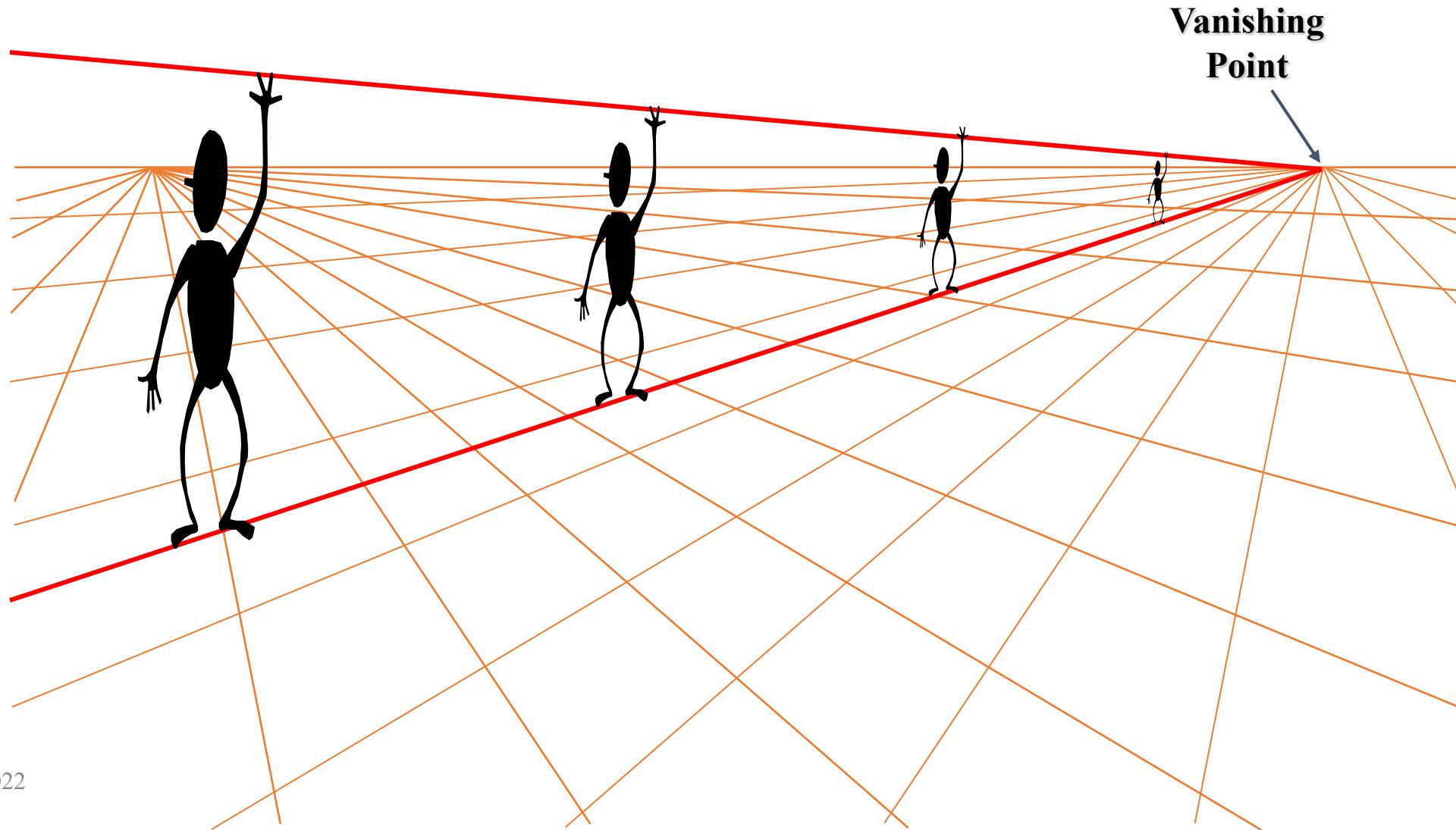
Perspective cues



Ames Room

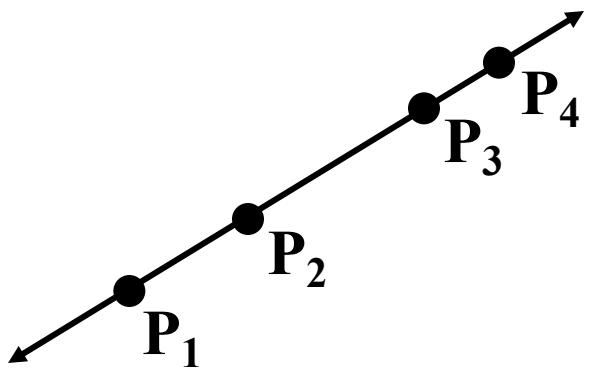


Comparing heights



Cross ratio. A Projective Invariant

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

Homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image
coordinates

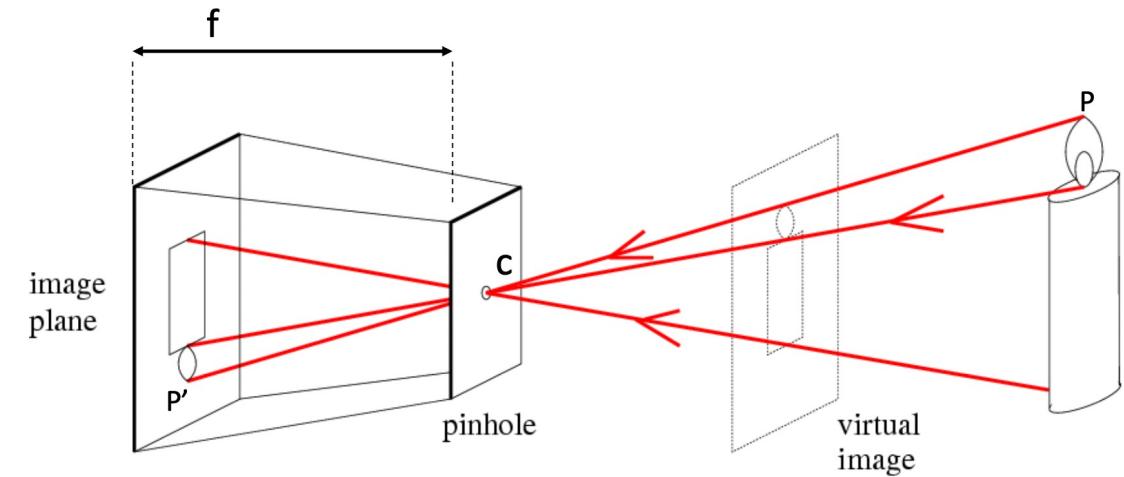
$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene
coordinates

Converting from homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

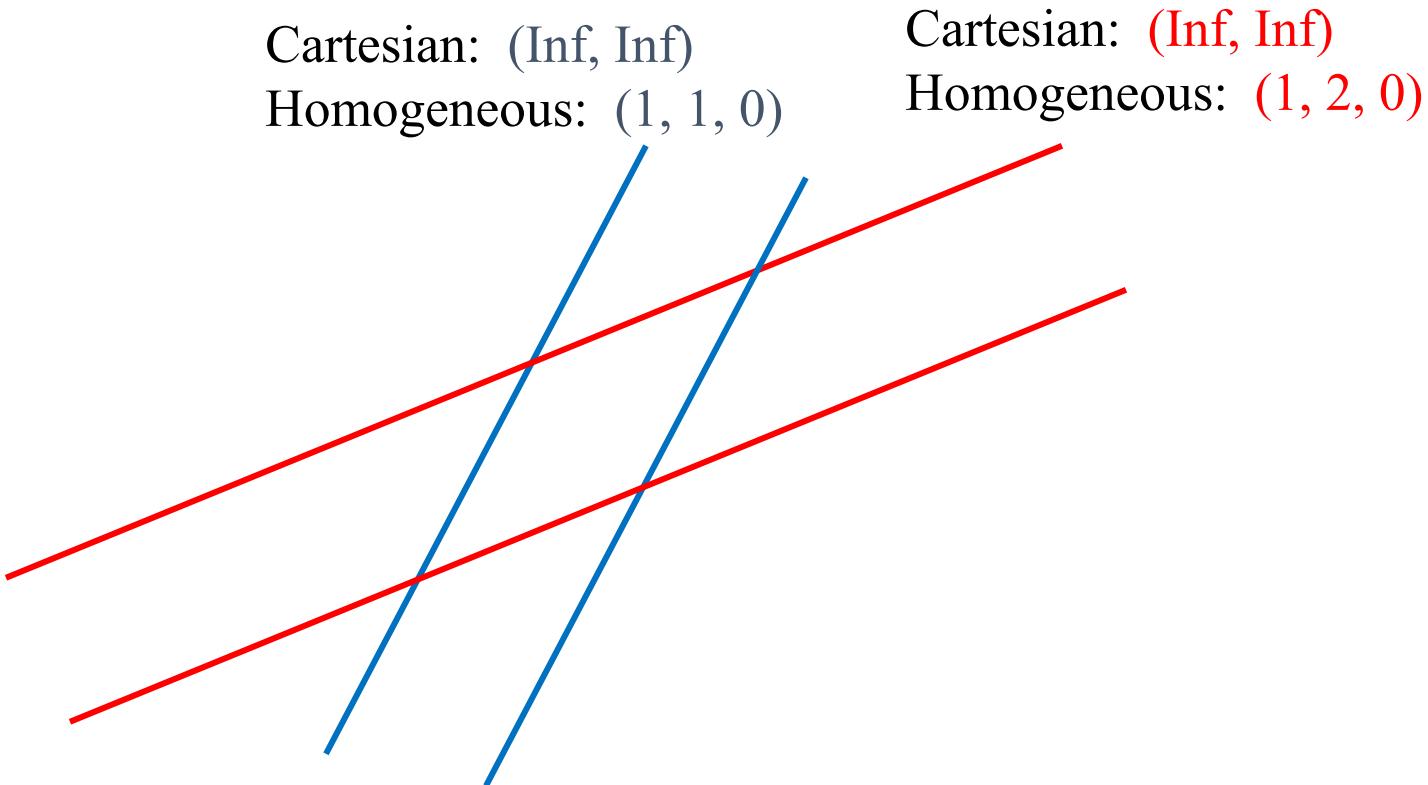
$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$



Also invariant to scale, multiplication with a scalar results movement along a ray

Another problem solved by homogeneous coordinates

Intersection of parallel lines



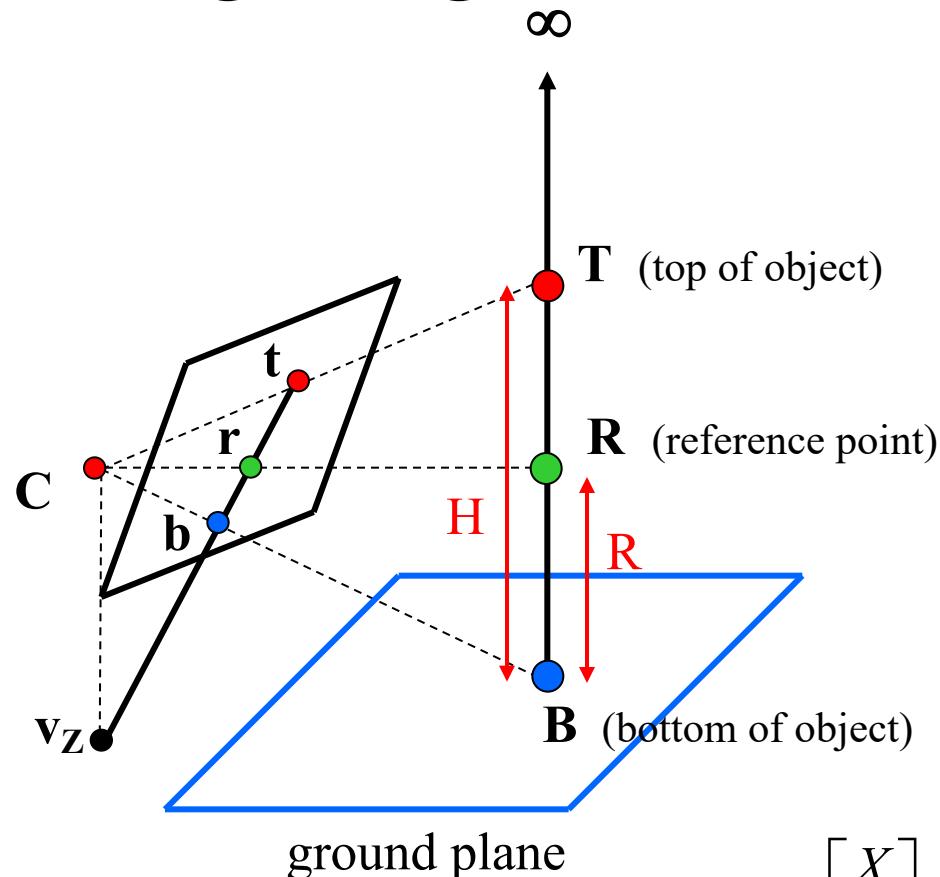
Basic geometry in homogeneous coordinates

- Line equation: $au + bv + c = 0$ $\text{line}_i = [a \ b \ c]^\top$
- Line given by cross product of two points $\text{line}_{ij} = p_i \times p_j$
- Intersection of two lines given by cross product of the lines

$$q_{ij} = \text{line}_i \times \text{line}_j$$

- Three points lies on the same line $p_k^\top (p_i \times p_j) = 0$
- Three lines intersect at the same point $\text{line}_k^\top (\text{line}_i \times \text{line}_j) = 0$

Measuring height



scene points represented as

$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

image points as

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

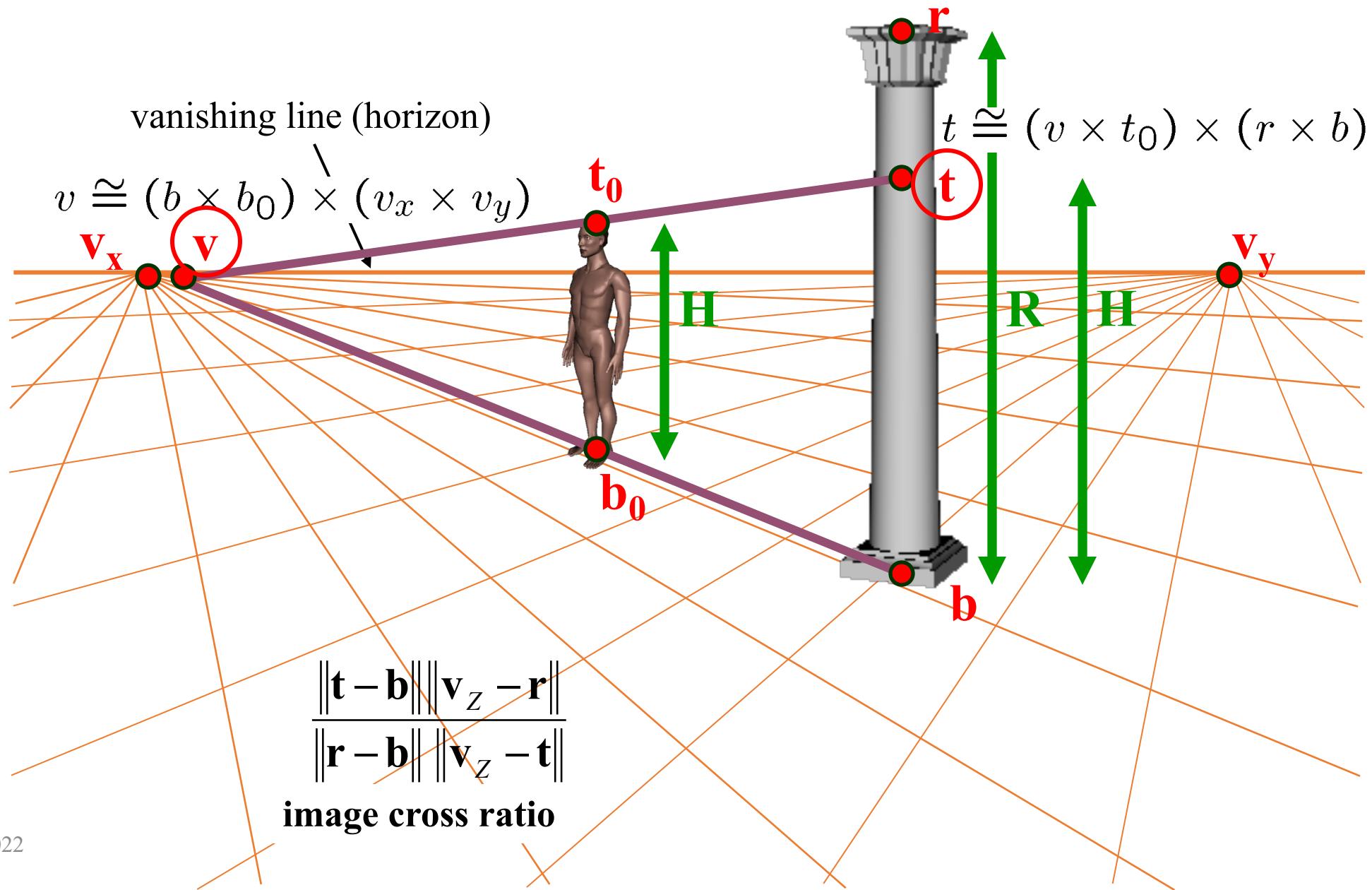
$$\frac{\|\mathbf{B} - \mathbf{T}\| \|\infty - \mathbf{R}\|}{\|\mathbf{B} - \mathbf{R}\| \|\infty - \mathbf{T}\|} = \frac{H}{R}$$

scene cross ratio

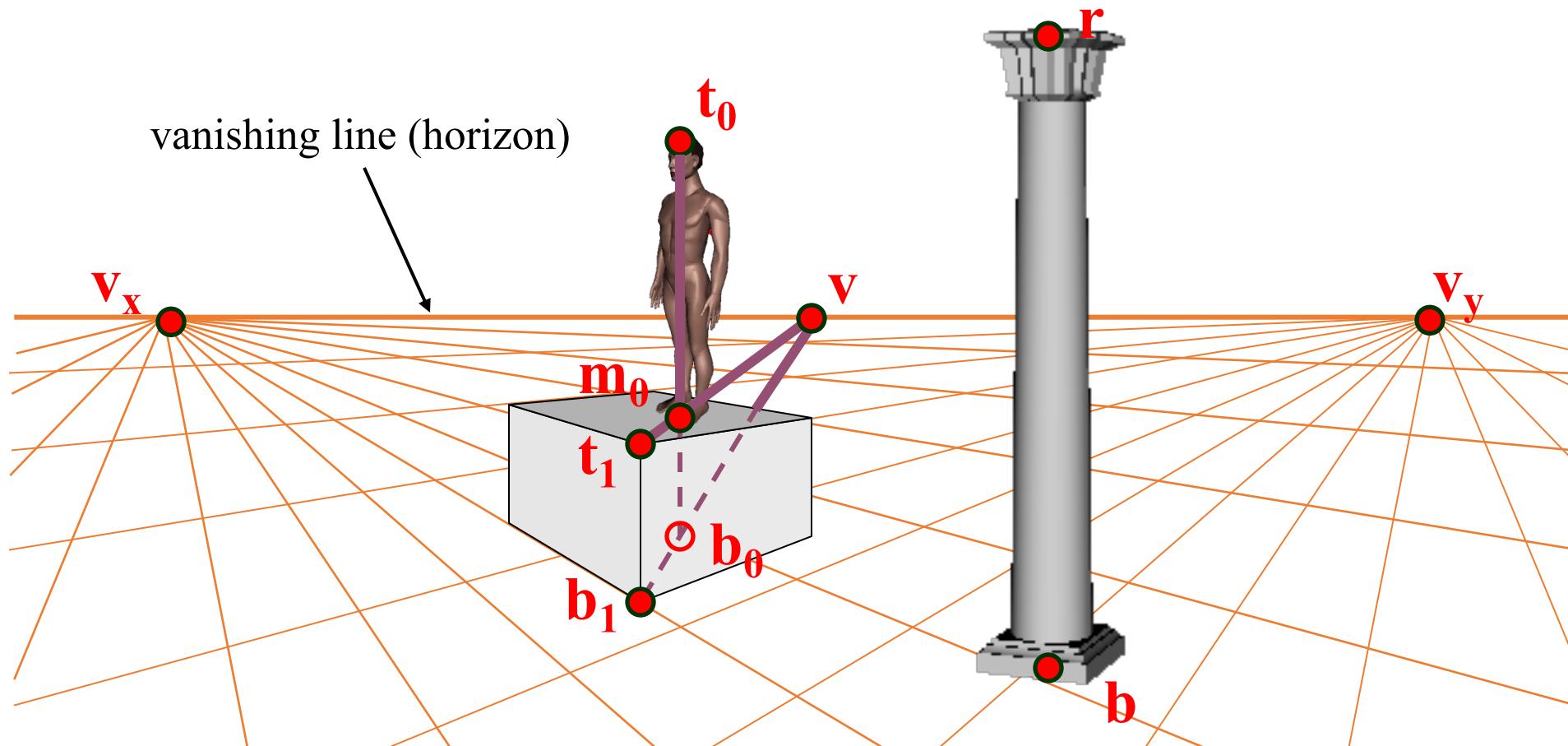
$$\frac{\|\mathbf{b} - \mathbf{t}\| \|\mathbf{v}_z - \mathbf{r}\|}{\|\mathbf{b} - \mathbf{r}\| \|\mathbf{v}_z - \mathbf{t}\|} = \frac{H}{R}$$

image cross ratio

Measuring height



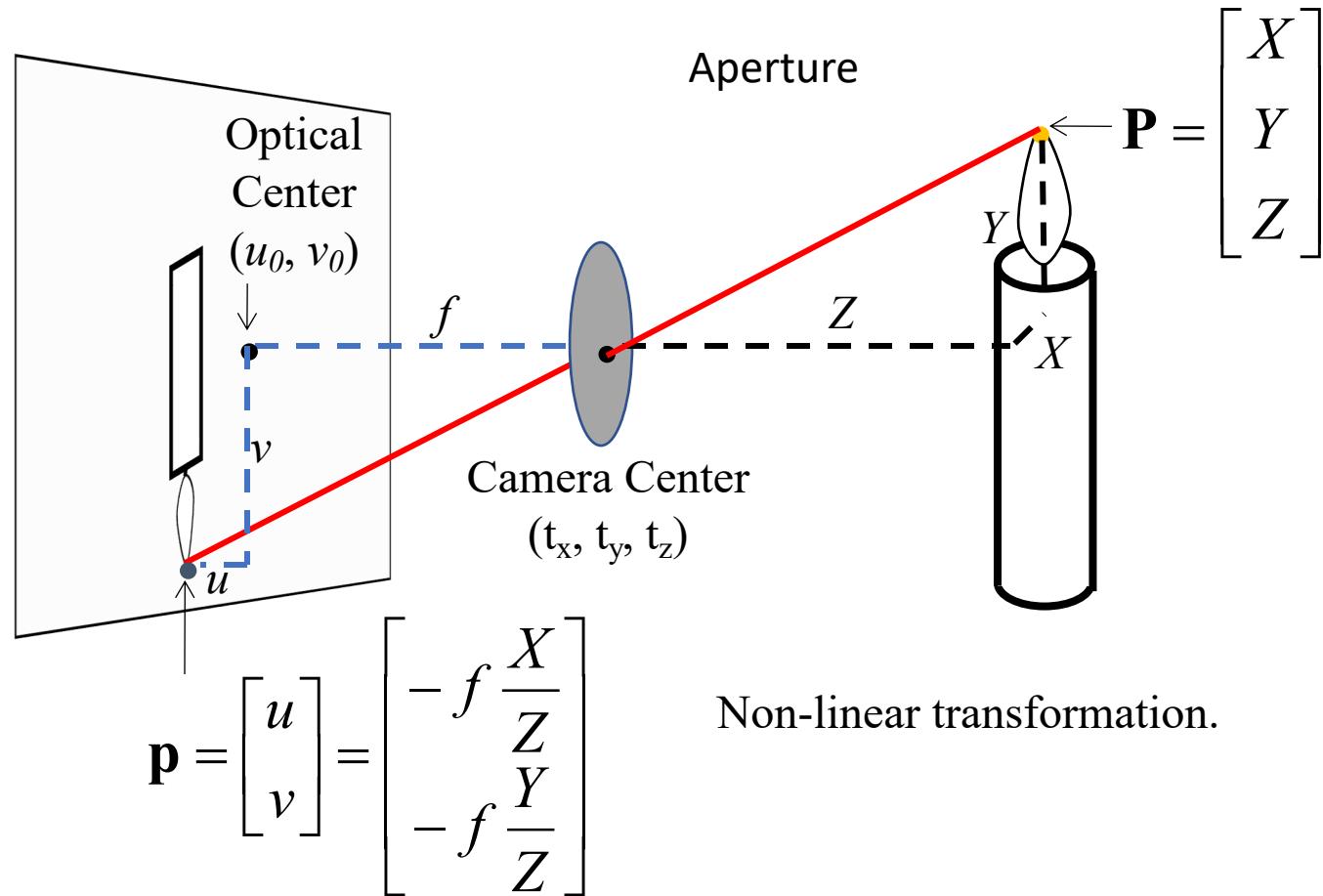
Measuring height



What if the point on the ground plane 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 b_0 as shown above

Projection: world coordinates \rightarrow image coordinates



Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates

$$\mathbf{P}' = \begin{bmatrix} f & x \\ f & y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Basically, $\mathbf{P}' = \mathbf{MP}$

Has 2 components: intrinsics->related to camera specifications
Extrinsics-> related to camera orientation

Projection matrix

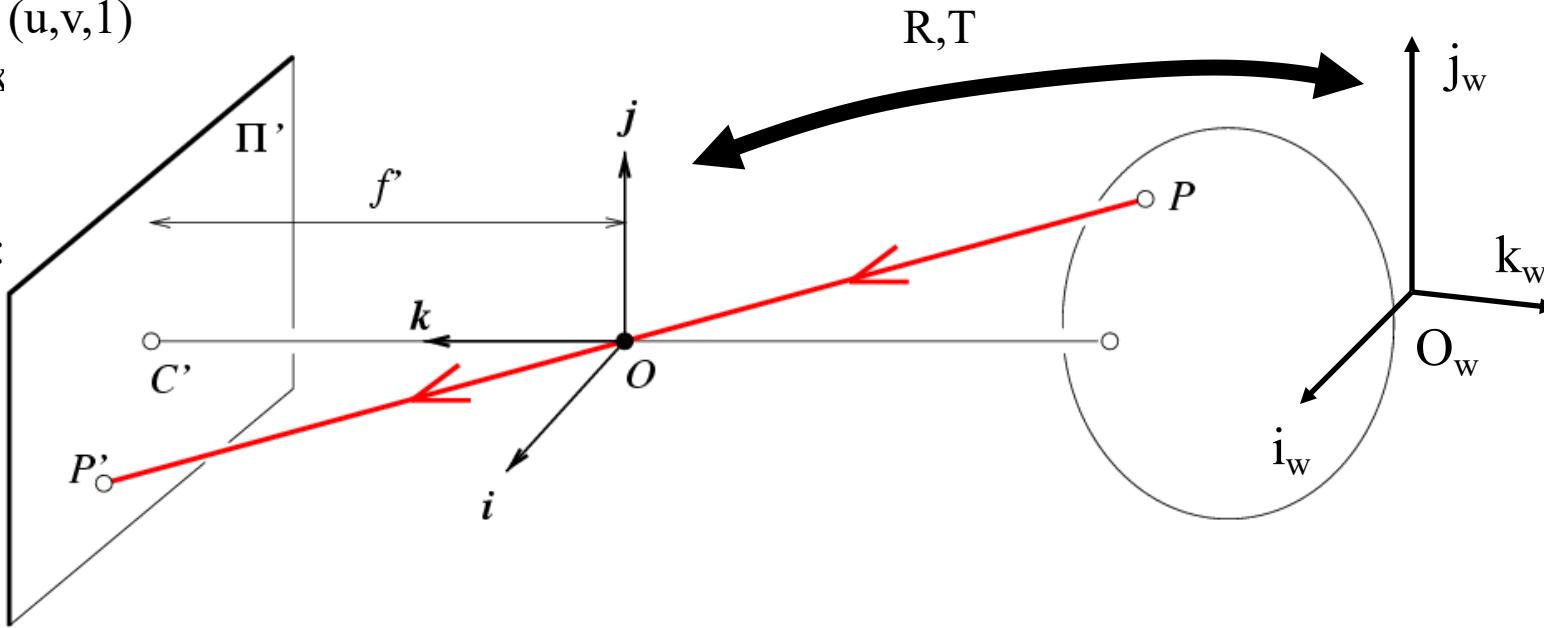
\mathbf{x} : Image Coordinates: $(u, v, 1)$

\mathbf{K} : Intrinsic Matrix (3x3)

\mathbf{R} : Rotation (3x3)

\mathbf{t} : Translation (3x1)

\mathbf{X} : World Coordinates:



$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X}$$

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Degrees of freedom?

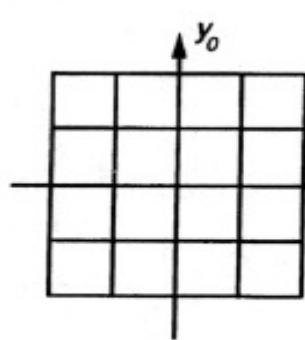
Vanishing Point = Projection from Infinity

$$\mathbf{p} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \Rightarrow \mathbf{p} = \mathbf{K}\mathbf{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \mathbf{p} = \mathbf{K} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix}$$

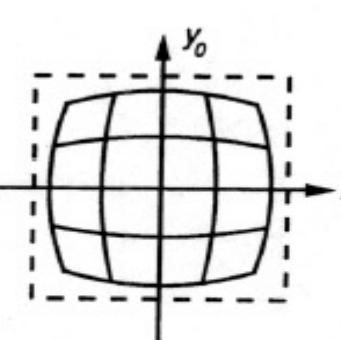
$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \Rightarrow \begin{aligned} u &= \frac{fx_R}{z_R} + u_0 \\ v &= \frac{fy_R}{z_R} + v_0 \end{aligned}$$

Beyond Pinholes: Radial Distortion

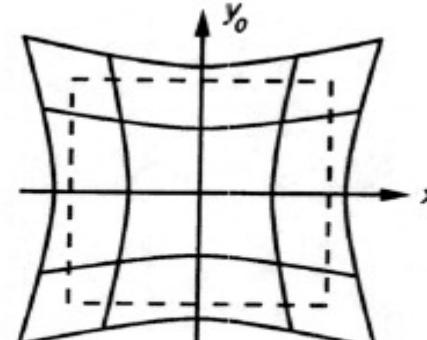
- Common in wide-angle lenses or for special applications (e.g., security)
- Creates non-linear terms in projection
- Usually handled by through solving for non-linear terms and then correcting image



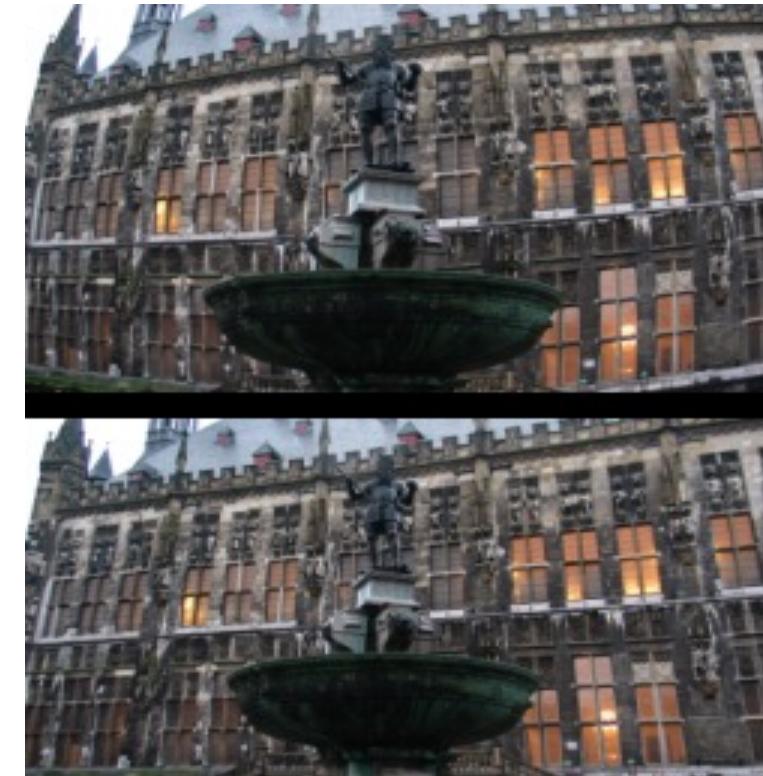
No Distortion



Barrel Distortion



Pincushion Distortion

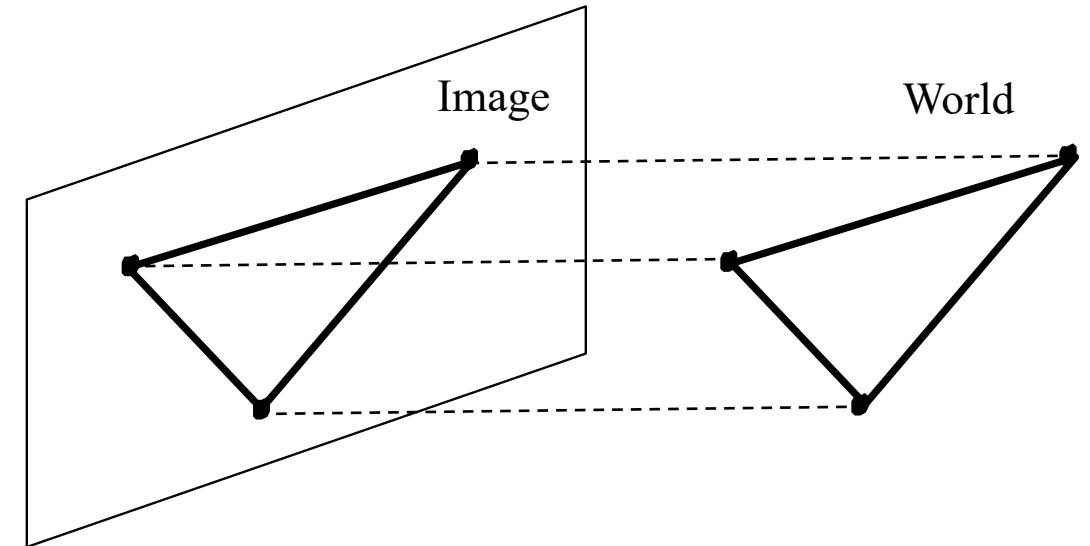


Corrected Barrel Distortion

Scaled Orthographic Projection

- Special case of perspective projection
 - Object dimensions are small compared to distance to camera

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & s \end{bmatrix}}_S \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



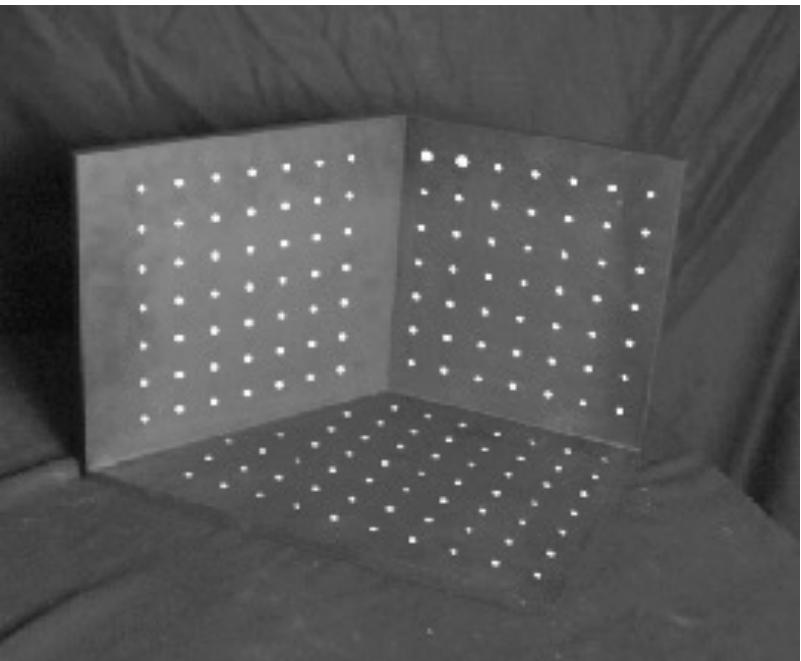
- Also called “weak perspective”
 $s=1$: orthographic (parallel projection)

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \; \mathbf{t}] \mathbf{S} \mathbf{X}$$

Camera Calibration: finding intrinsics and extrinsics

Method 1: Use an object (calibration grid) with known geometry

- Correspond image points to 3d points
- Get least squares solution (or non-linear solution)



Known 2d image coordinates

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Unknown Camera Parameters

Camera Calibration: Finding intrinsics and extrinsics

Unknown Camera Parameters

Known 2d image coords

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Known 3d locations



$$m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

Camera Calibration: Finding intrinsics and extrinsics

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

- Method 1 – homogeneous linear system. Solve for m using linear least squares

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & \vdots & & & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$[U, S, V] = \text{svd}(A);$
 $M = V(:, \text{end});$
 $M = \text{reshape}(M, [], 3)';$

Camera Calibration: Finding intrinsics and extrinsics

Method 2 – nonhomogeneous linear system. Solve for \mathbf{m} using linear least squares

$$\begin{array}{l} \text{Known 2d} \\ \text{image coords} \end{array} \quad \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \begin{array}{l} \text{Known 3d} \\ \text{locations} \end{array}$$

$\mathbf{Ax}=\mathbf{b}$ form

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & \vdots & & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{bmatrix}$$

$\mathbf{M} = \mathbf{A} \setminus \mathbf{Y};$
 $\mathbf{M} = [\mathbf{M}; 1];$
 $\mathbf{M} = \text{reshape}(\mathbf{M}, [], 3)';$

Calibration with linear method

- Advantages
 - Easy to formulate and solve
 - Provides initialization for non-linear methods
- Disadvantages
 - Doesn't directly give you camera parameters
 - Doesn't model radial distortion
 - Can't impose constraints, such as known focal length
 - Doesn't minimize projection error
- Non-linear methods are preferred
 - Define error as difference between projected points and measured points
 - Minimize error using Newton's method or other non-linear optimization

Can we factorize M back to K [R | T]?

- Yes!
- You can use RQ factorization (note – not the more familiar QR factorization). R (right diagonal) is K, and Q (orthogonal basis) is R. T, the last column of $[R | T]$, is $\text{inv}(K) * \text{last column of } M$.
 - But you need to do a bit of post-processing to make sure that the matrices are valid. See <http://ksimek.github.io/2012/08/14/decompose/>