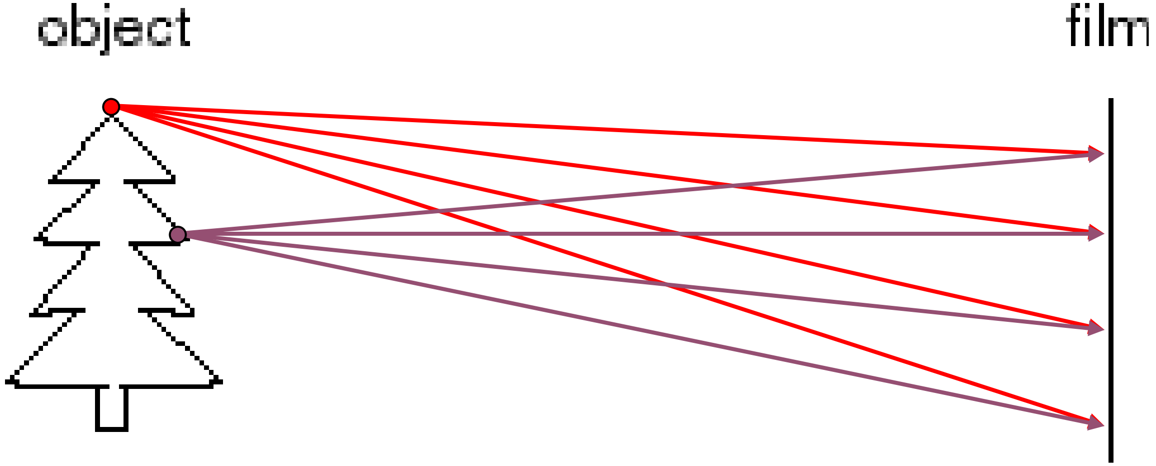
5. Camera and Images

Image formation, Camera models and Camera calibration

Image formation

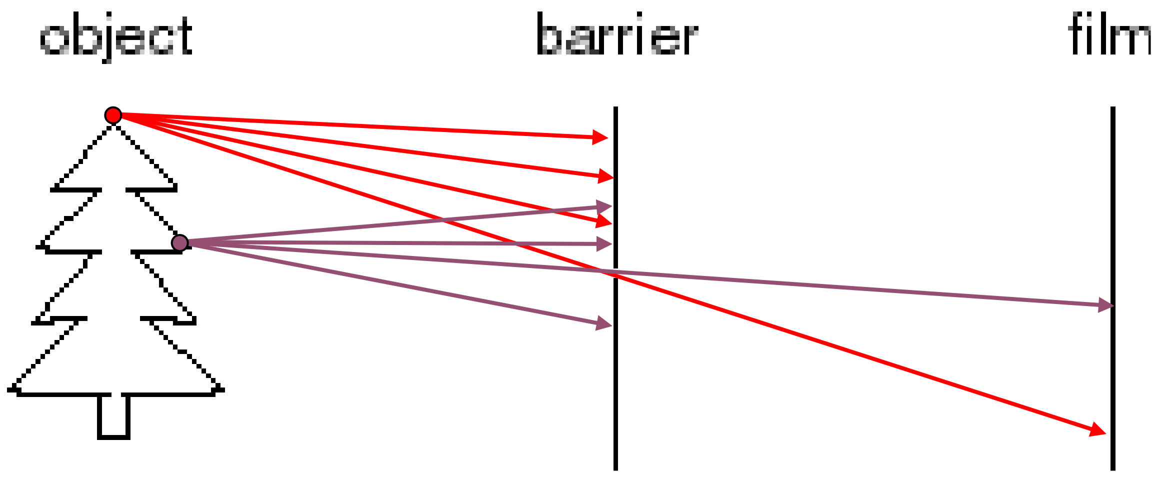


Let’s design a camera

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

Slide source: Seitz

Pinhole camera

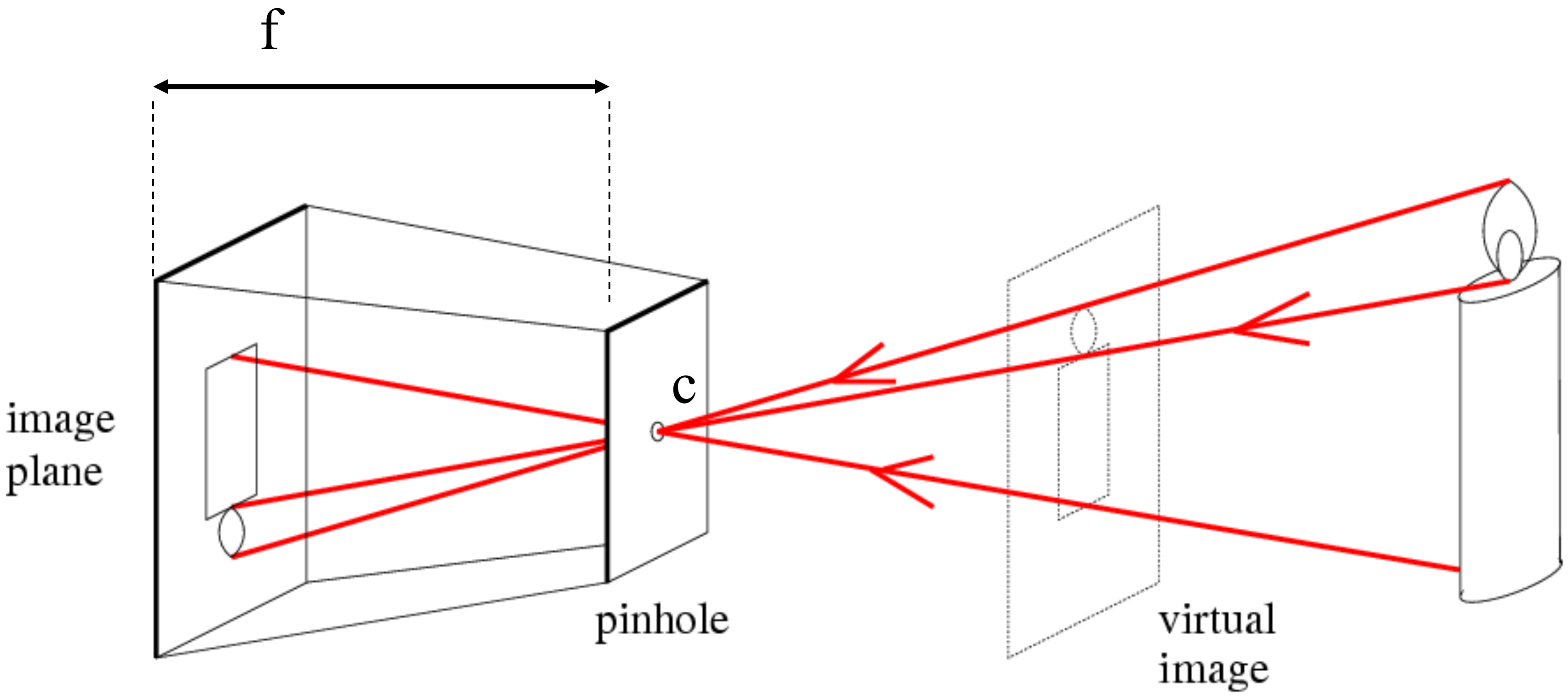


Idea 2: add a barrier to block off most of the rays

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

Slide source: Seitz

Pinhole camera



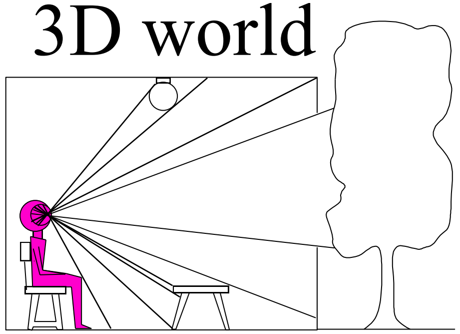
f = focal length

c = center of the camera

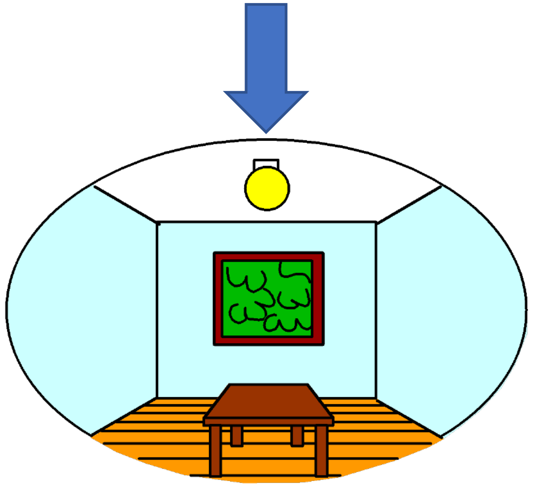
Figure from Forsyth

Dimensionality Reduction Machine (3D to 2D)

3D world



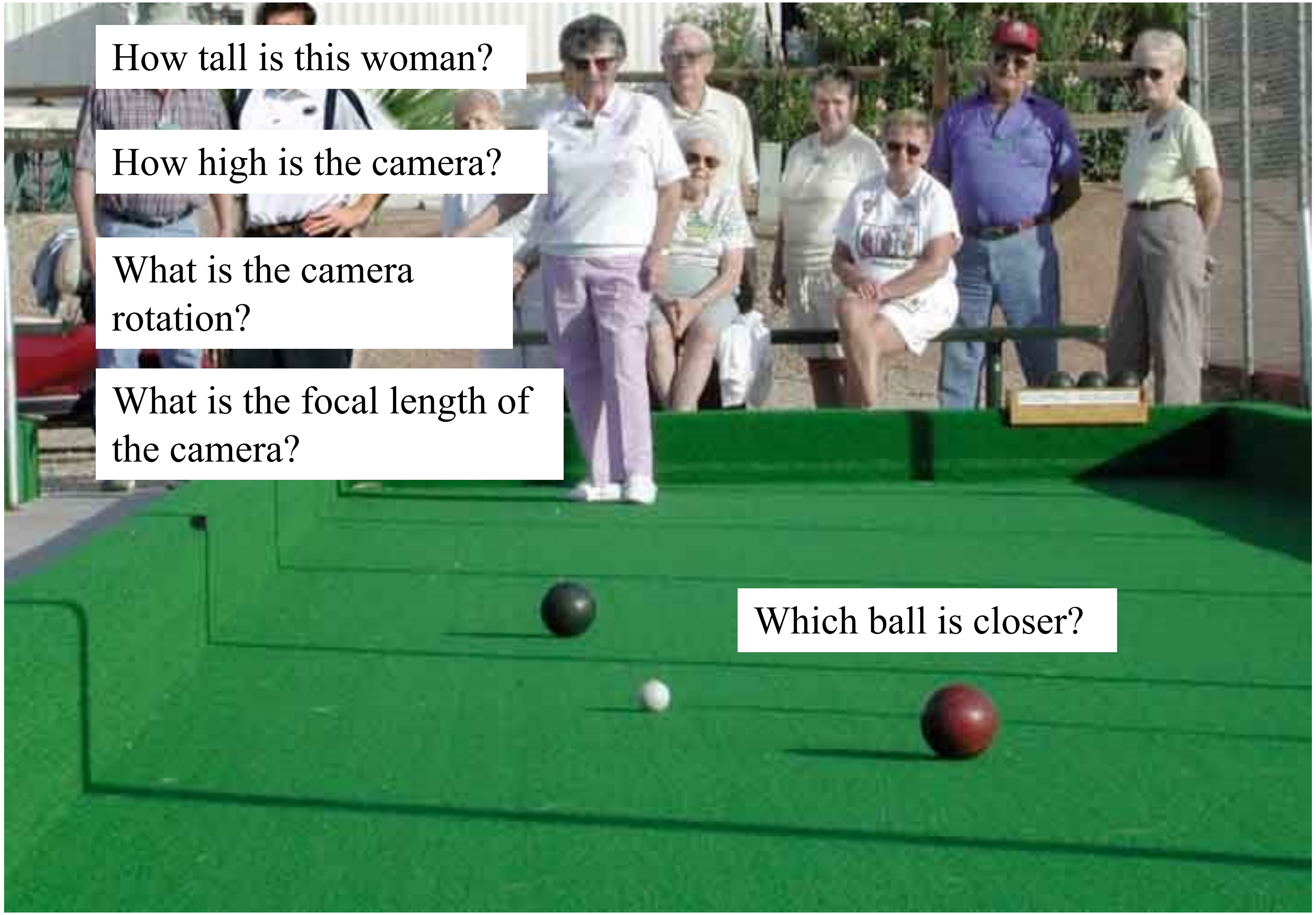
Point of observation



2D image

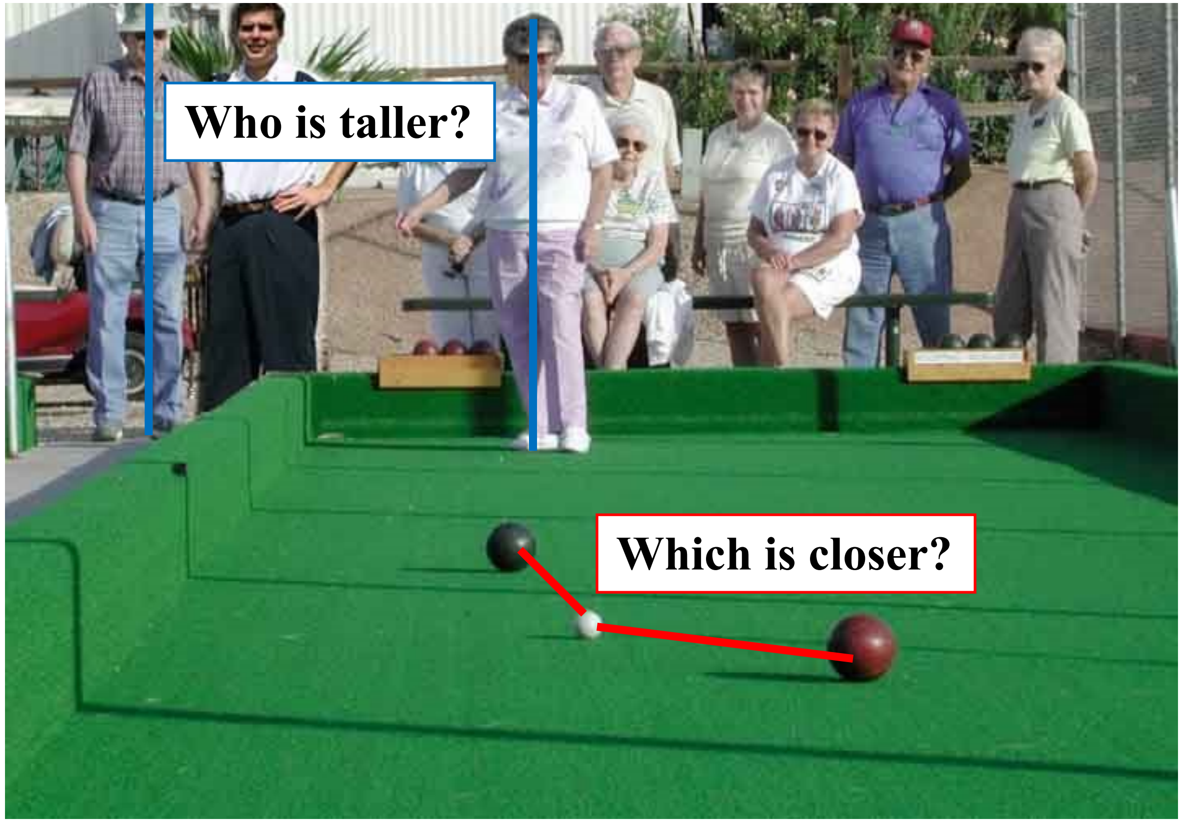
Projection depends on camera placement

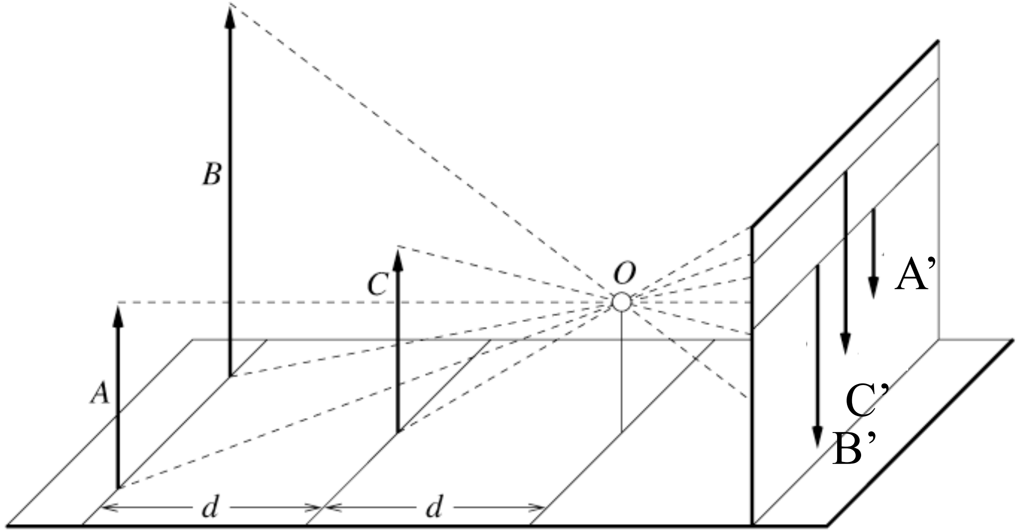
3D through images



Projective Geometry

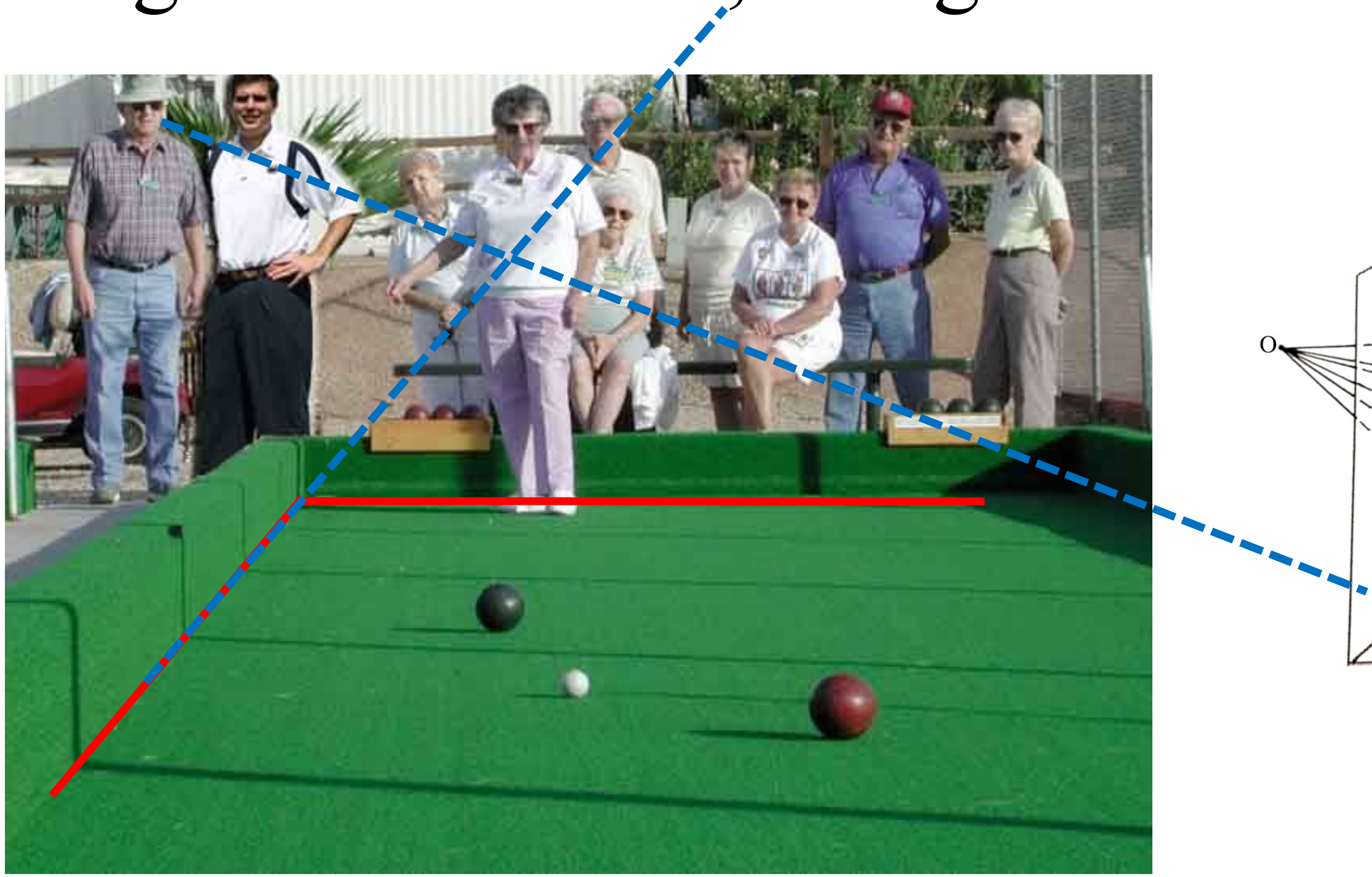
Length/distances/geodesics are lost

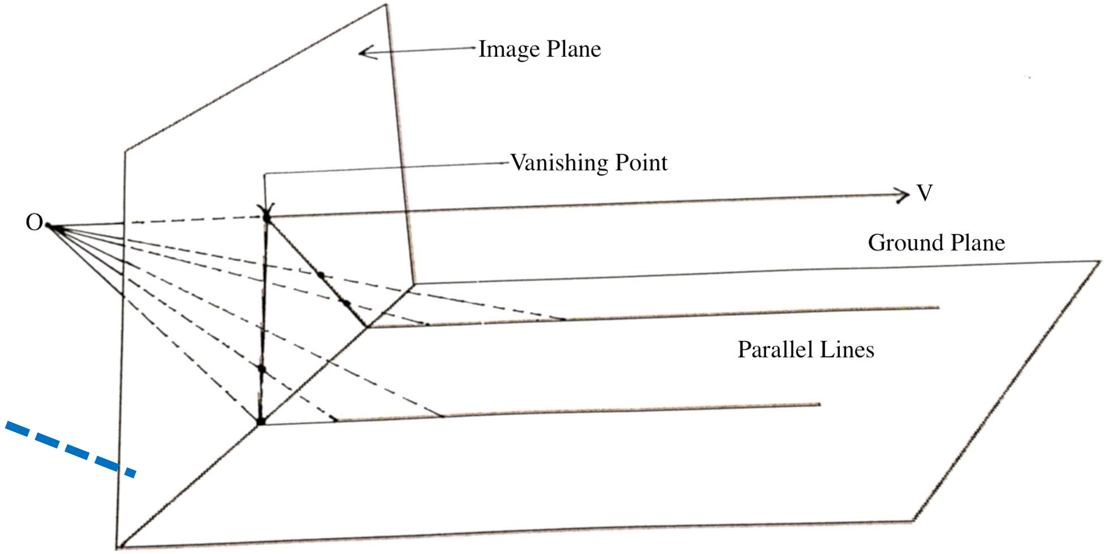




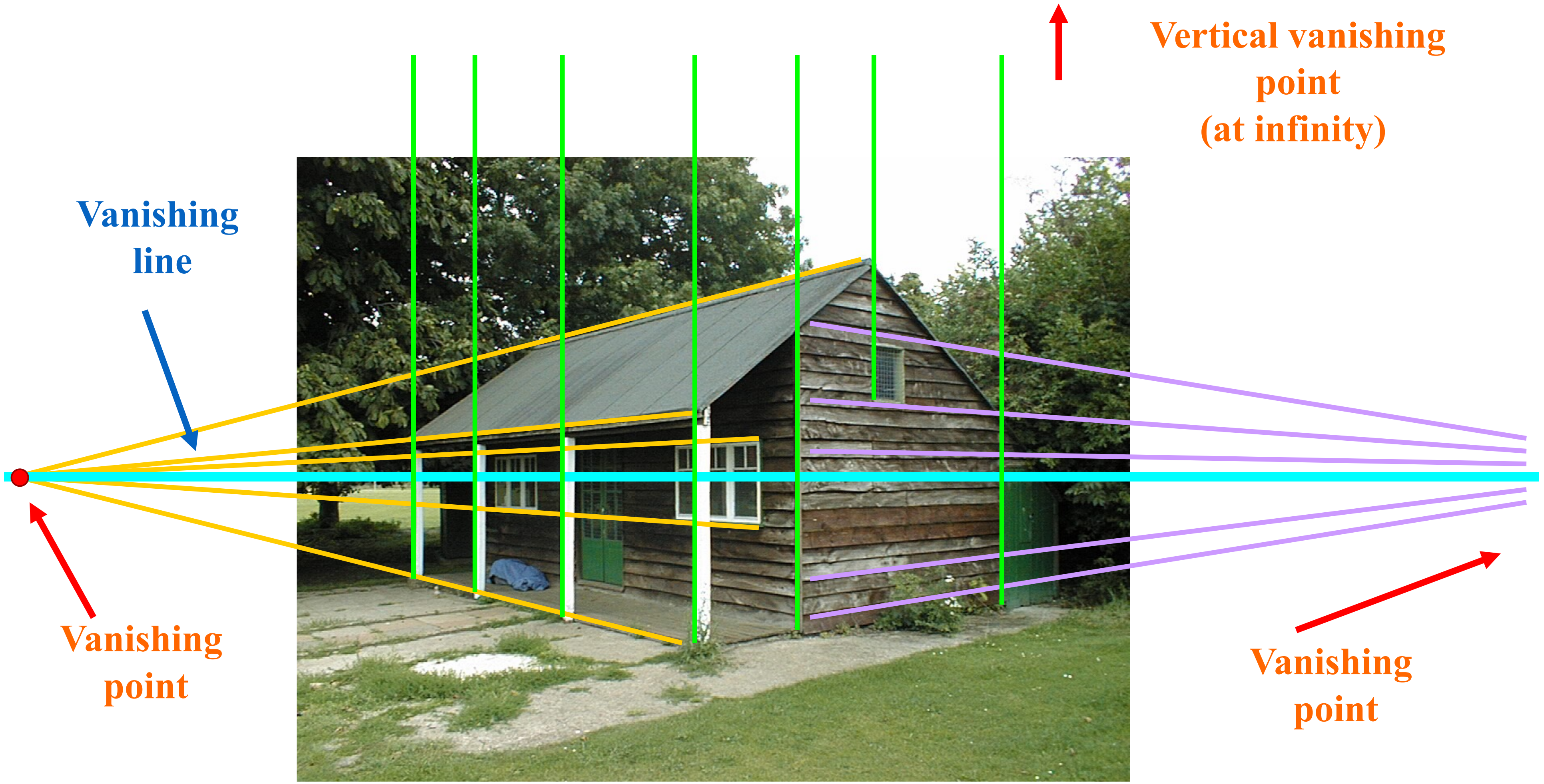
Projective Geometry

Angles are lost too, straight lines are still straight



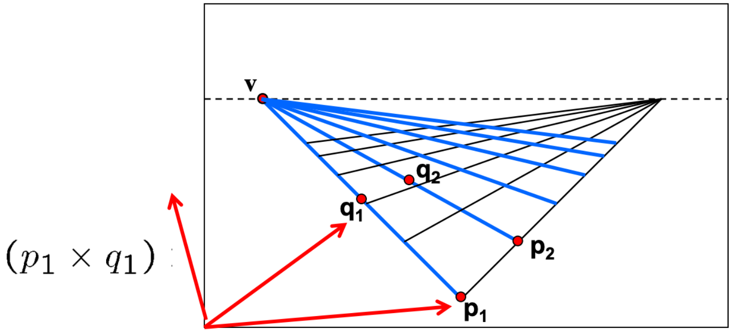


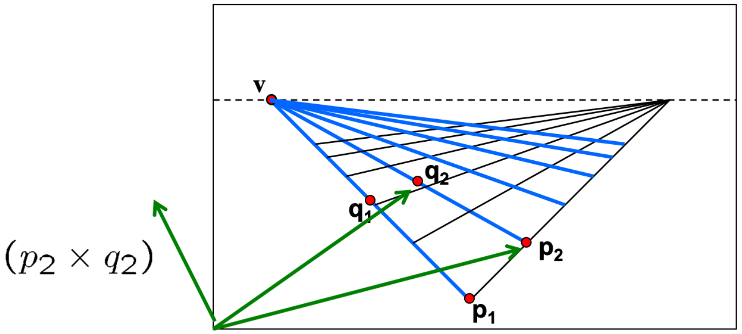
Vanishing points and lines

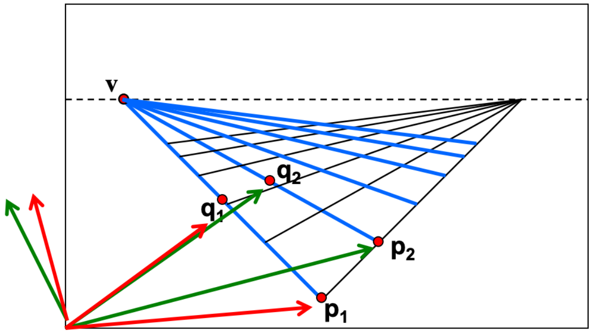


Slide from Efros, Photo from Criminisi

Vanishing points and lines



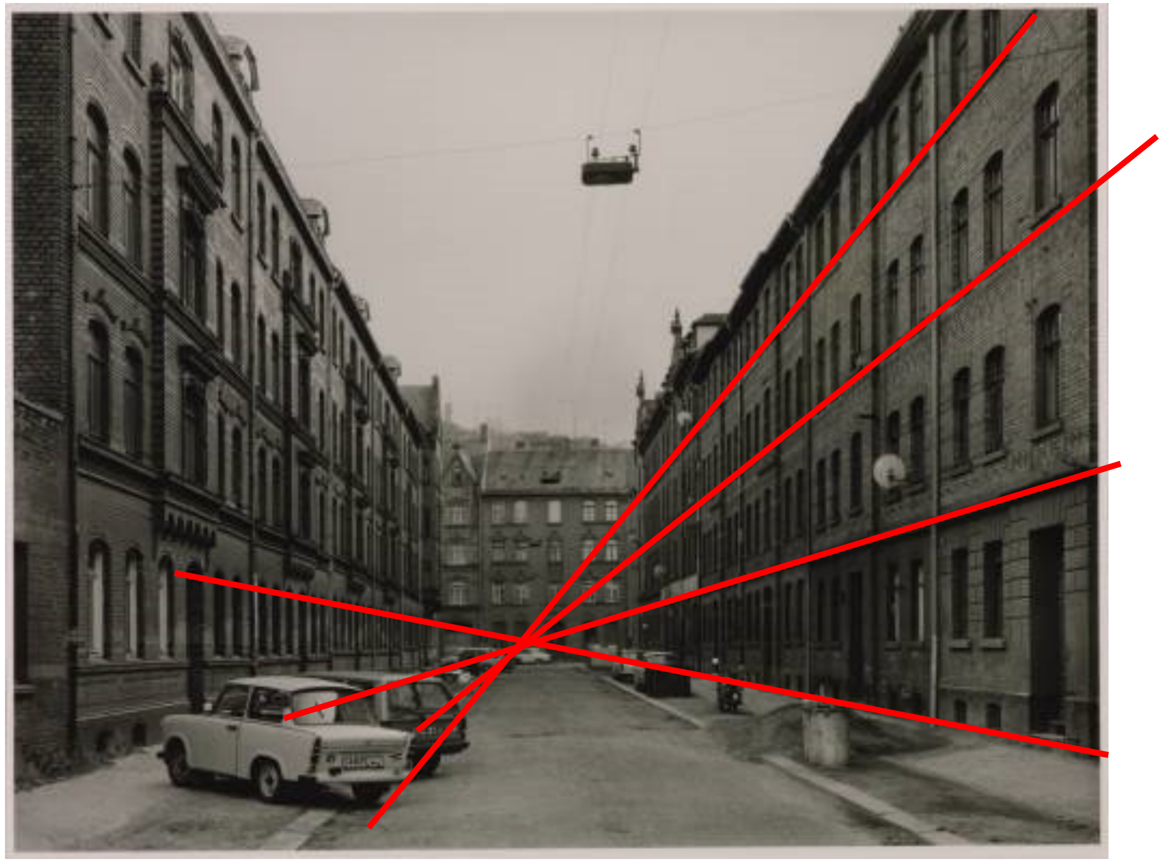




$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Slide from Efros, Photo from Criminisi

Note on estimating vanishing points



Use multiple lines for better accuracy

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

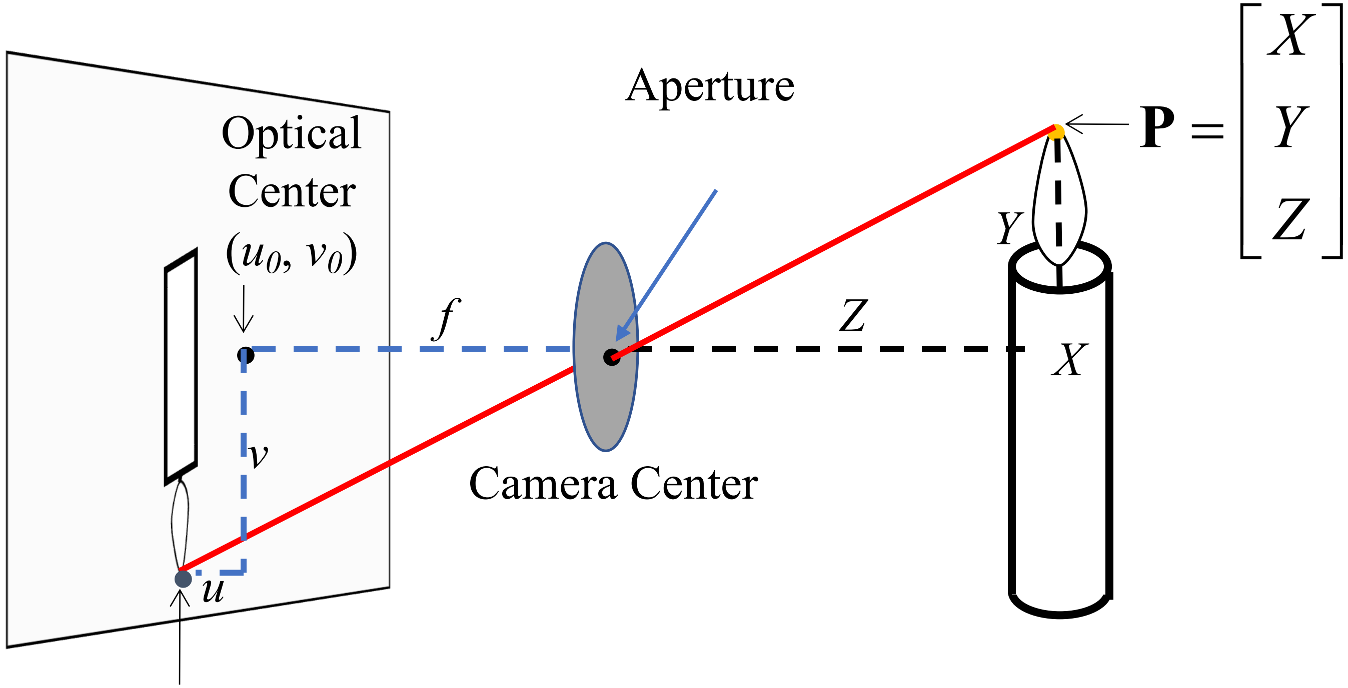
One solution: take mean of intersecting pairs

… bad idea!

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

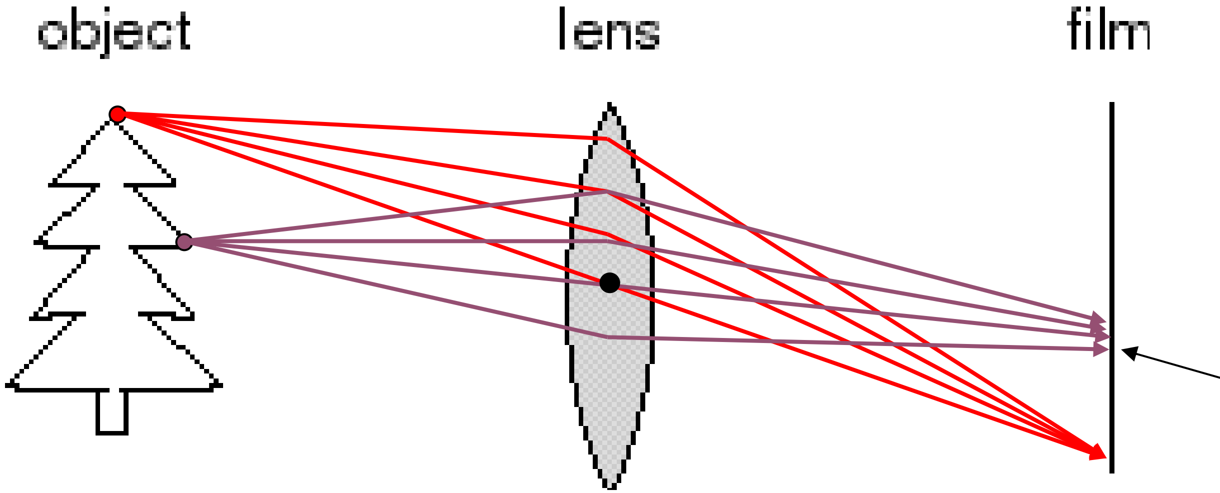
Vanishing objects

Aperture



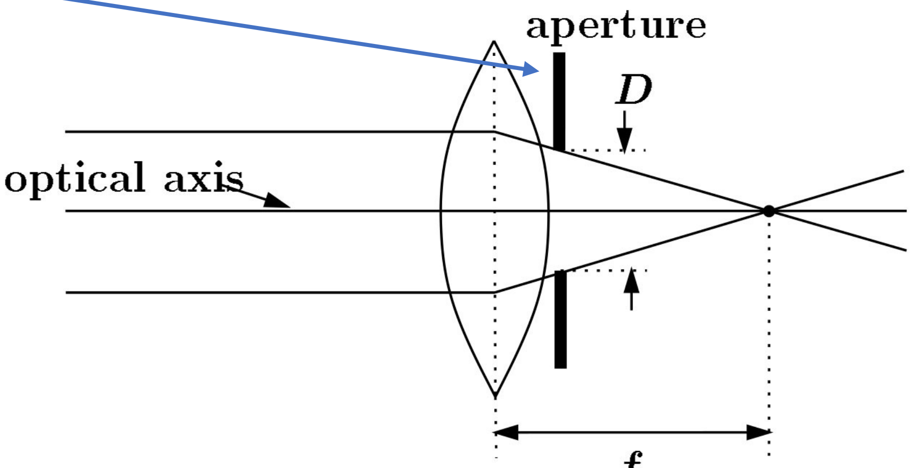
Aperture: should be small but not too much
In modern camera, pinhole is replace by a lens and a shutter is used to control explore

Adding a lens



There is a specific distance at which objects are "in focus"

Solution:

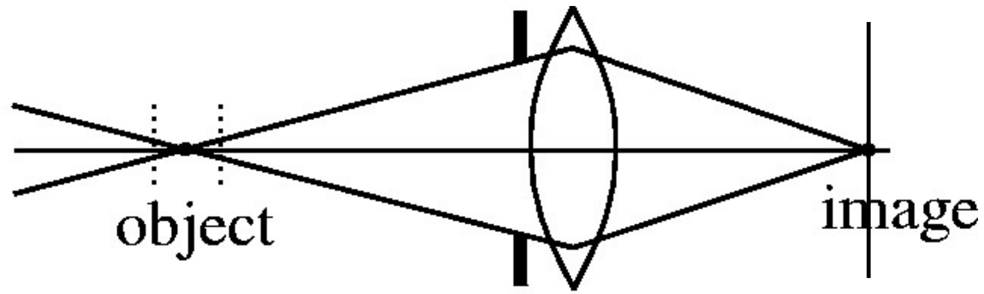


A lens focuses parallel rays onto a single focal point

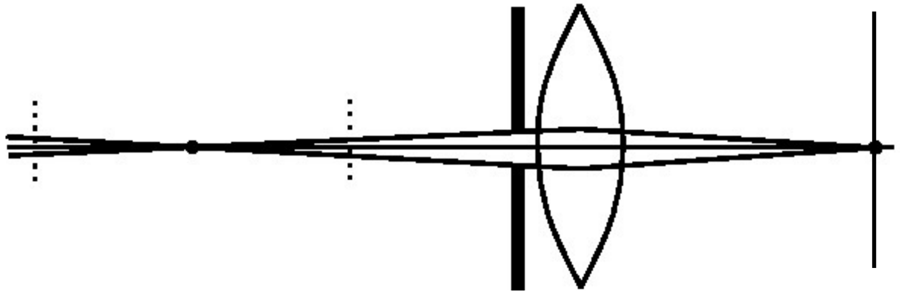
- focal point at a distance f beyond the plane of the lens
- Aperture of diameter D restricts the range of rays

Slide source:
Seitz

Depth of field



f / 5.6



f / 32

Changing the aperture size or focal length affects depth of field

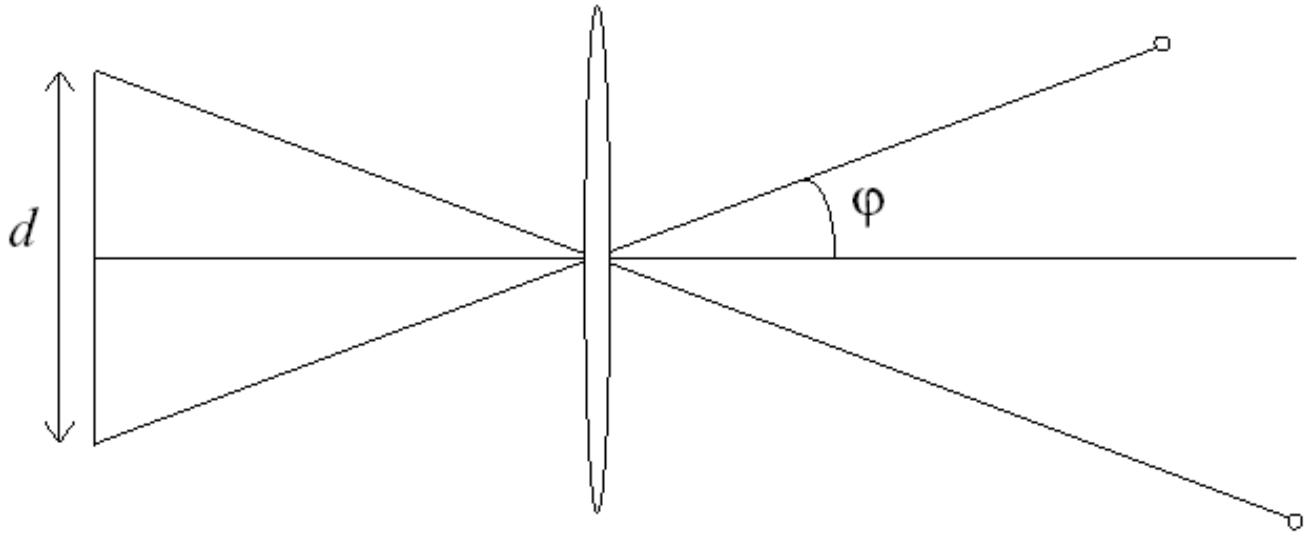
Relation between field of view and focal length

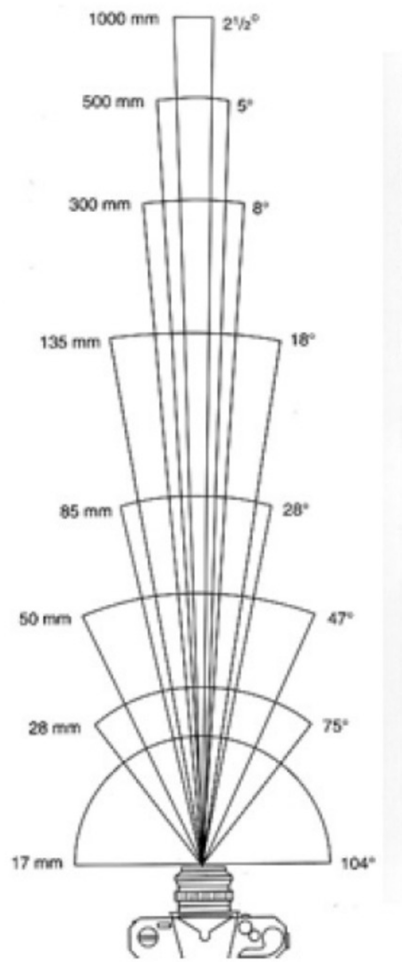
Field of view (angle width)

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

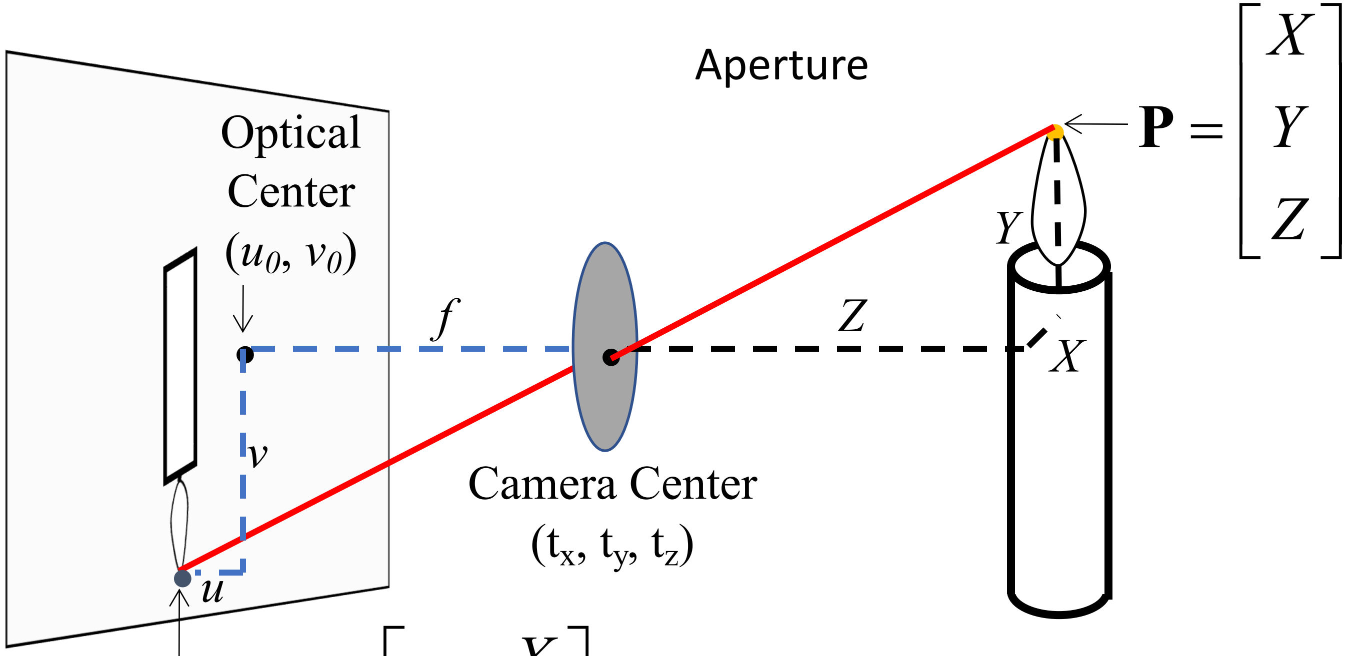
Film/Sensor Width

Focal length





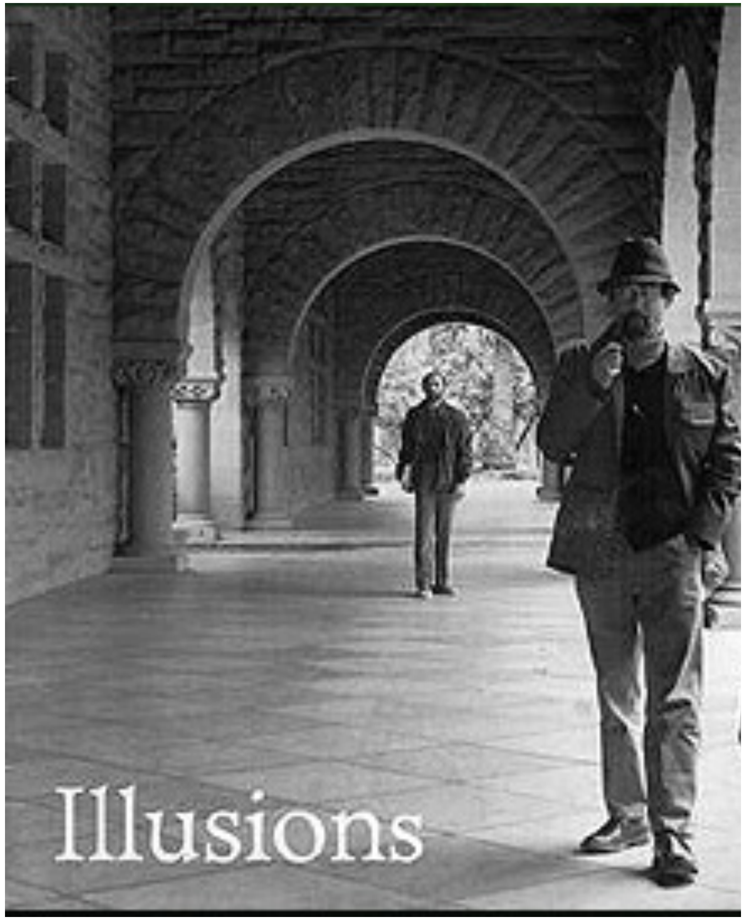
Projection:
world coordinates→image coordinates

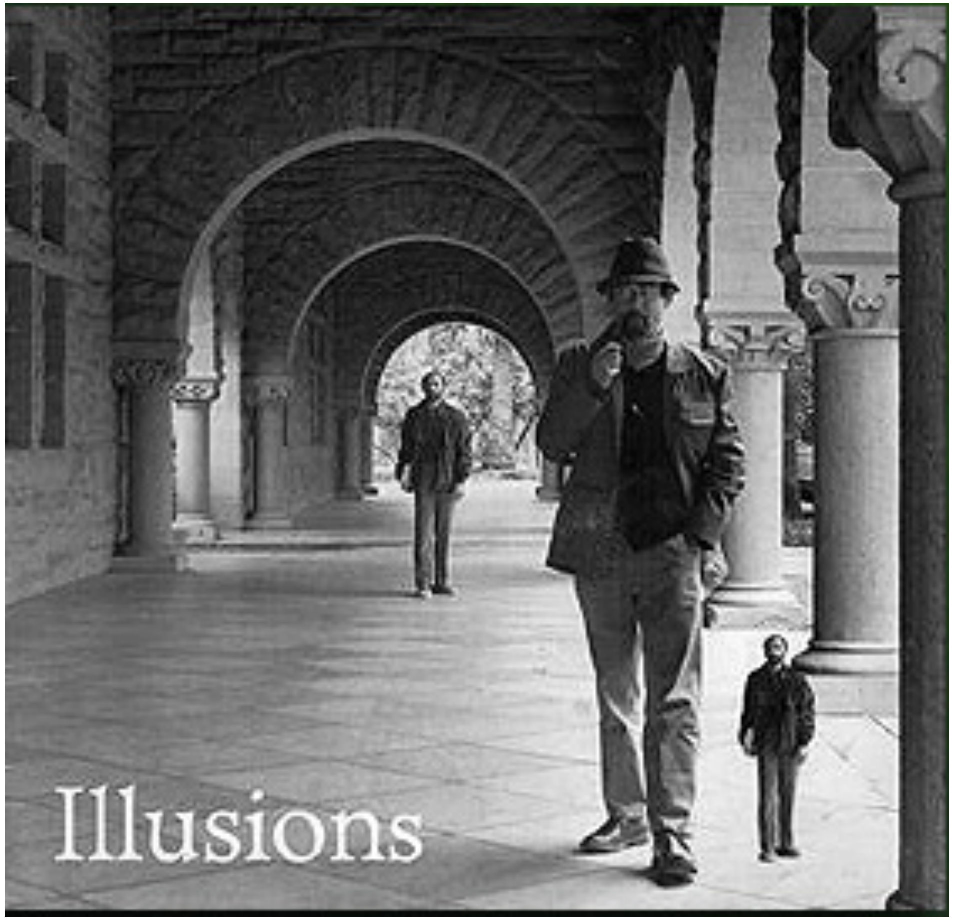


$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f\dfrac{X}{Z} \\ -f\dfrac{Y}{Z} \end{bmatrix}$$

Non-linear transformation.
Use homogeneous coordinates, because???

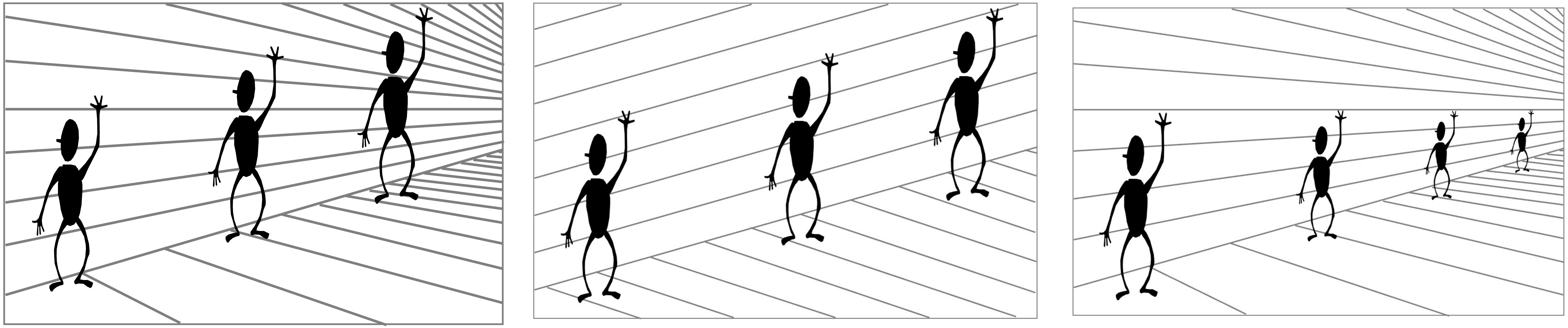
Perspective nature of cameras





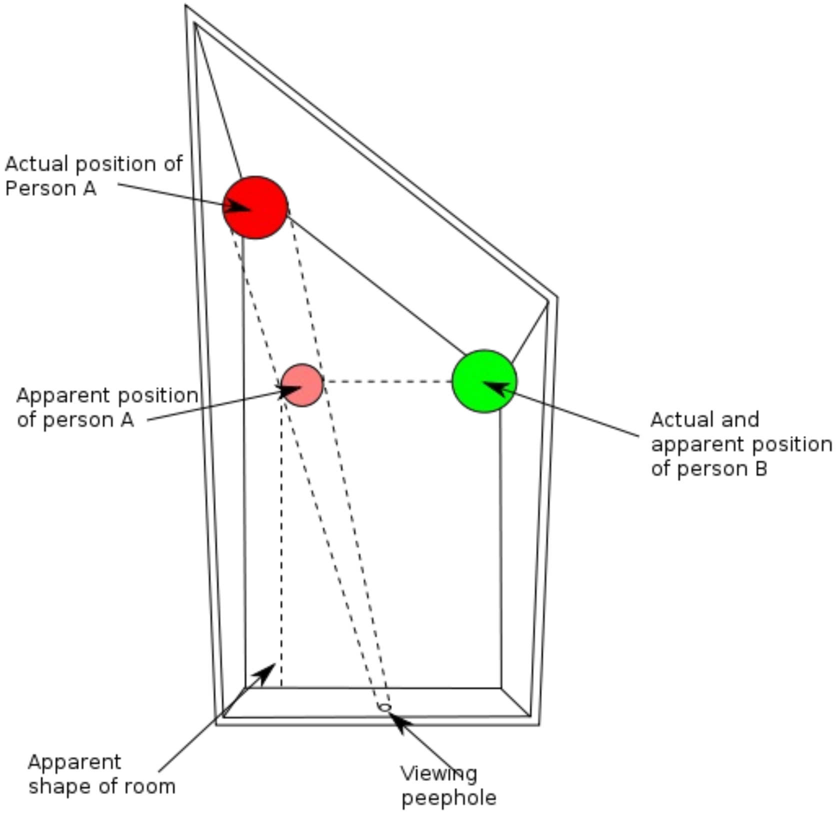
Slide by Steve Seitz

Perspective cues



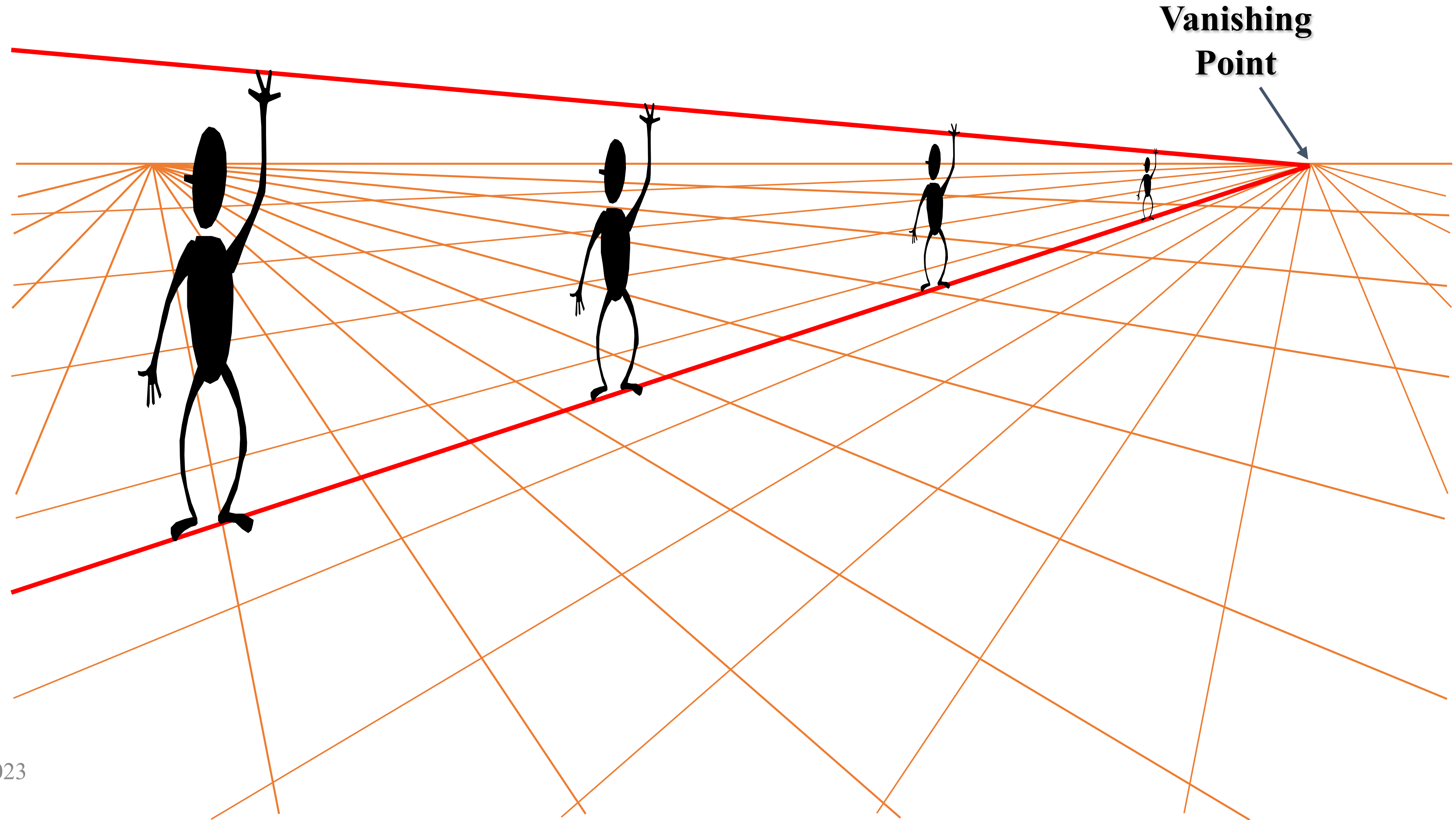
Slide by Steve Seitz

Ames Room



Slide by Steve Seitz

Comparing heights

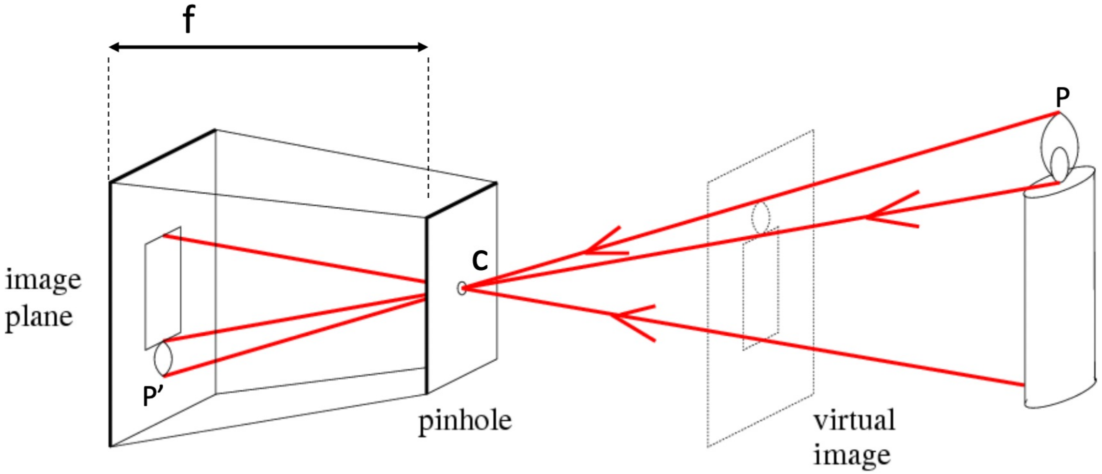


Homogeneous coordinates

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

homogeneous image coordinates

homogeneous scene coordinates



Converting from homogeneous coordinates

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

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

Why Homogeneous Coordinates?

Single view reconstruction: Siggraph 2005

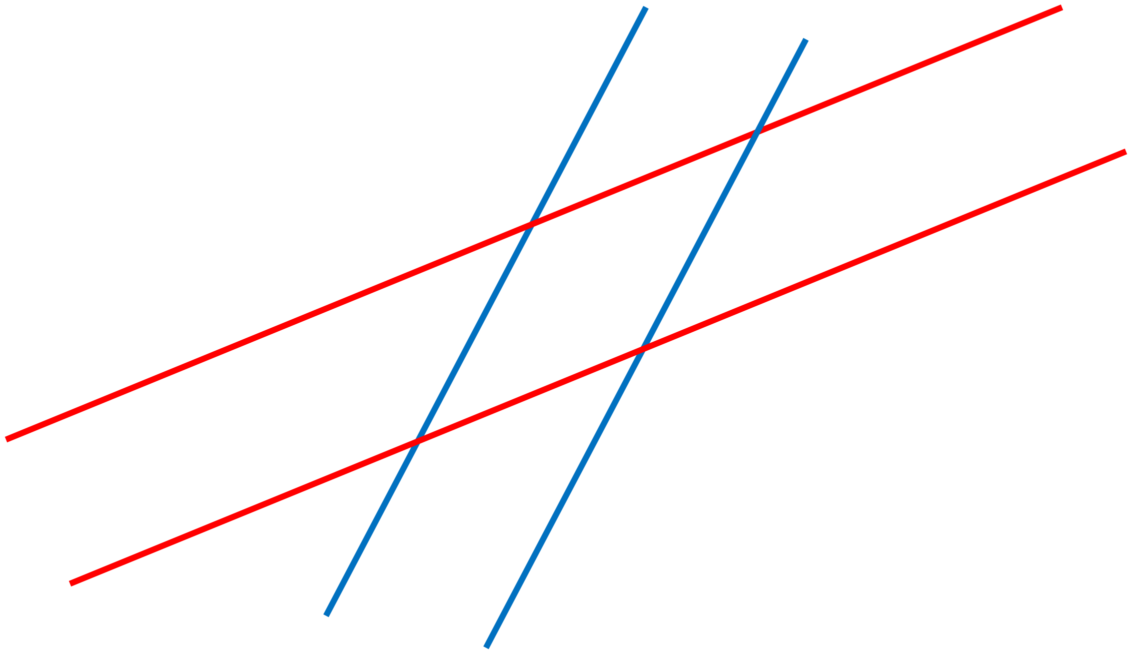
Analyse what happen when camera moves in (x,y) or in (z) wrt to the object

Another problem solved by homogeneous coordinates

Intersection of parallel lines

Cartesian: (Inf, Inf)
Homogeneous: (1, 1, 0)

Cartesian: (Inf, Inf)
Homogeneous: (1, 2, 0)



Basic geometry in homogeneous coordinates

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

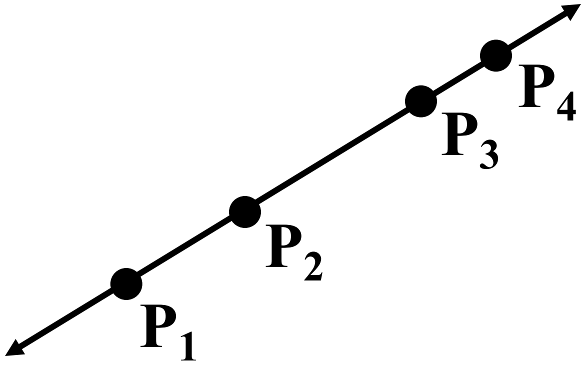
$$q_{ij} = line_i \times line_j$$

- Three points lies on the same line $p_k^{\top}(p_i \times p_j) = 0$
- Three lines intersect at the same point $line_k^{\top}(line_i \times line_j) = 0$

Slide by Steve Seitz

Cross ratio. A Projective Invariant

The cross-ratio of 4 collinear points



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

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

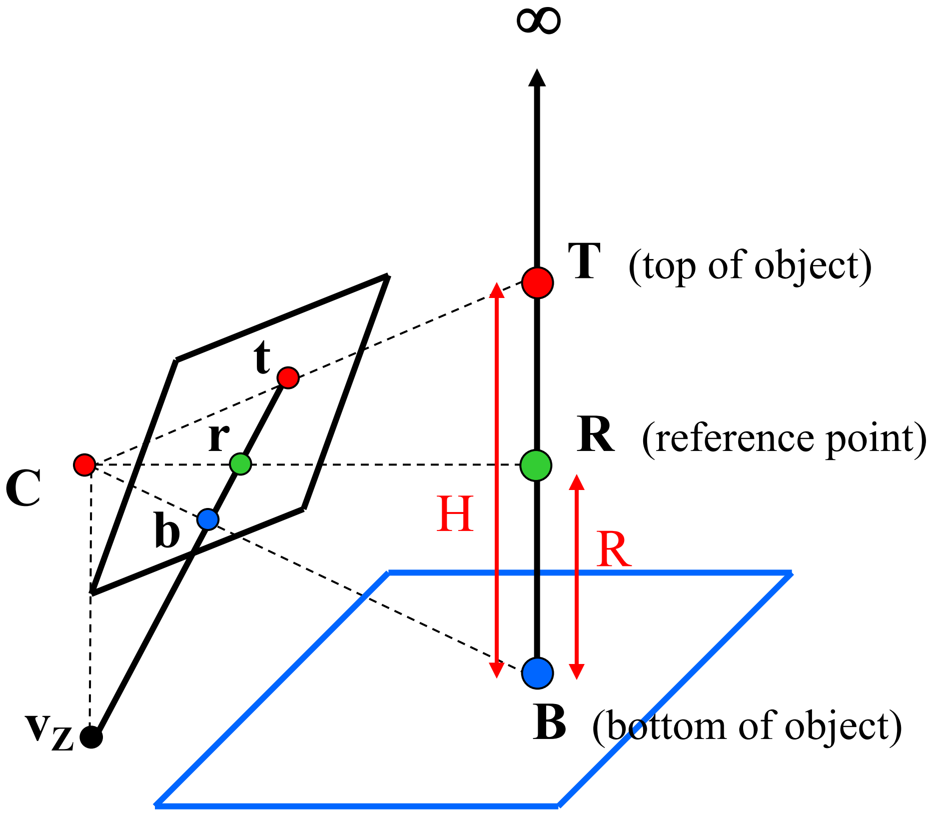
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

Measuring height



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

scene cross ratio

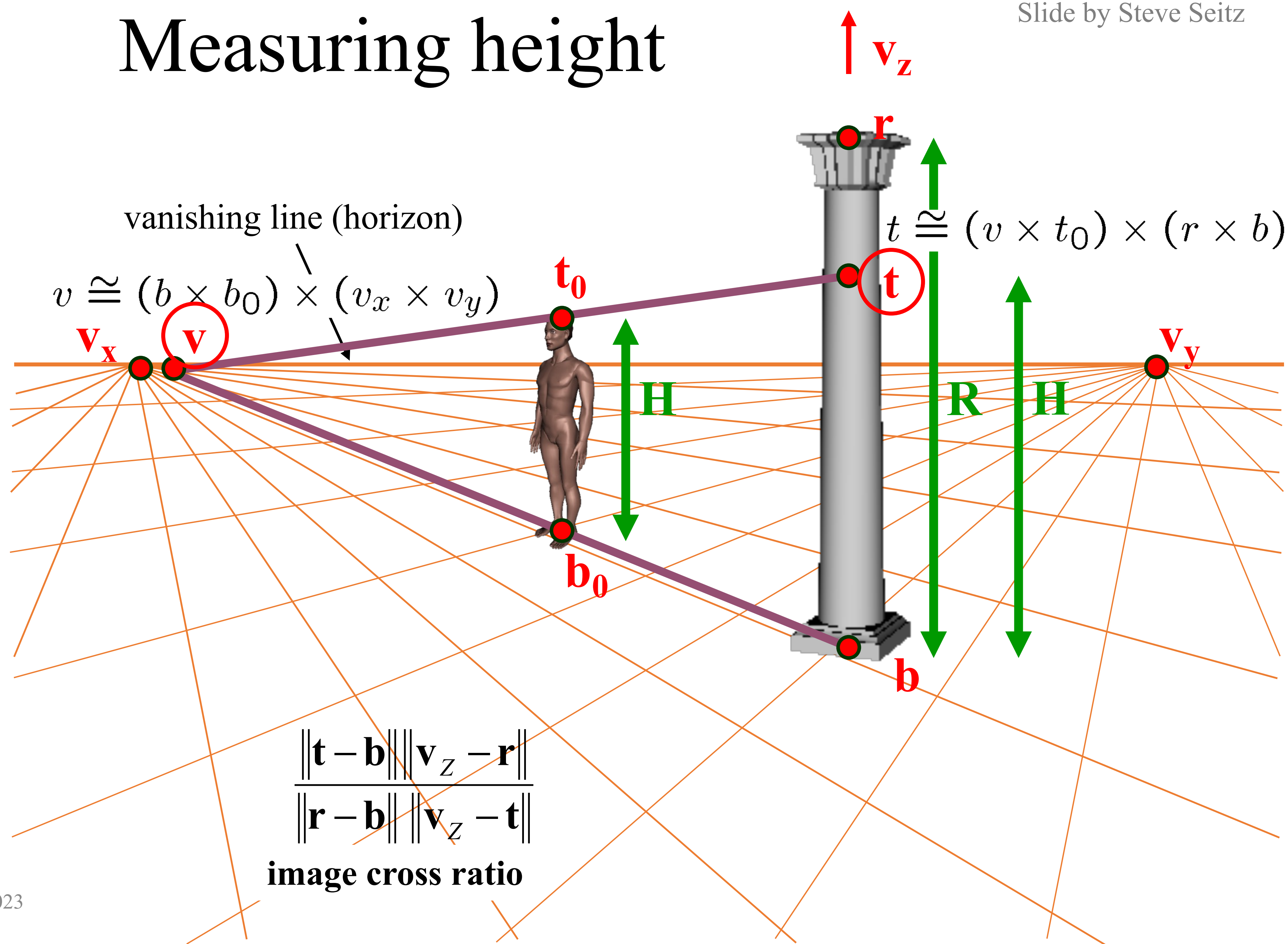
$$\frac{\|\mathbf{b}-\mathbf{t}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{b}-\mathbf{r}\|\,\|\mathbf{v}_Z-\mathbf{t}\|} = \frac{H}{R}$$

image cross ratio

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

Slide by Steve Seitz

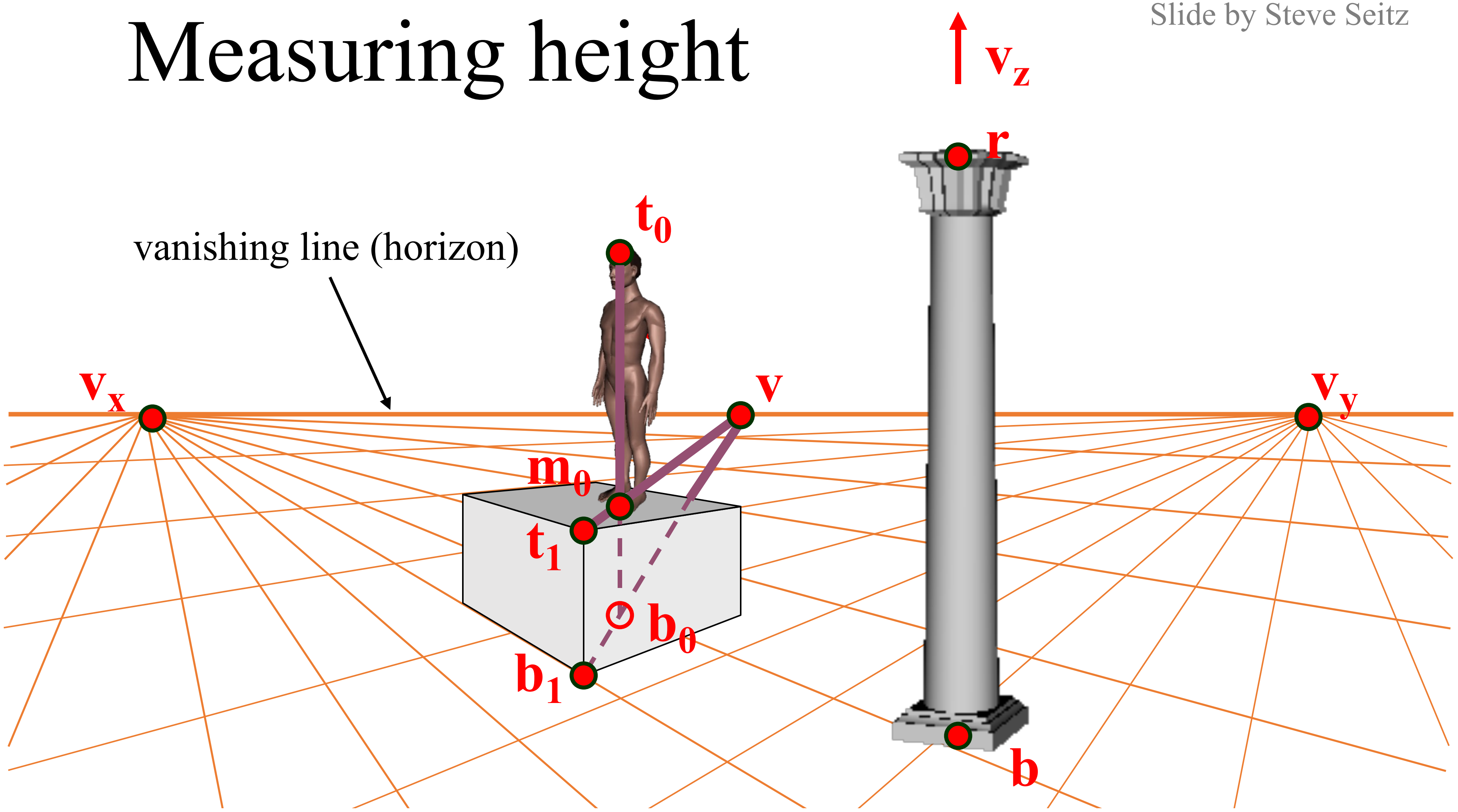


$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}$$

image cross ratio

Measuring height

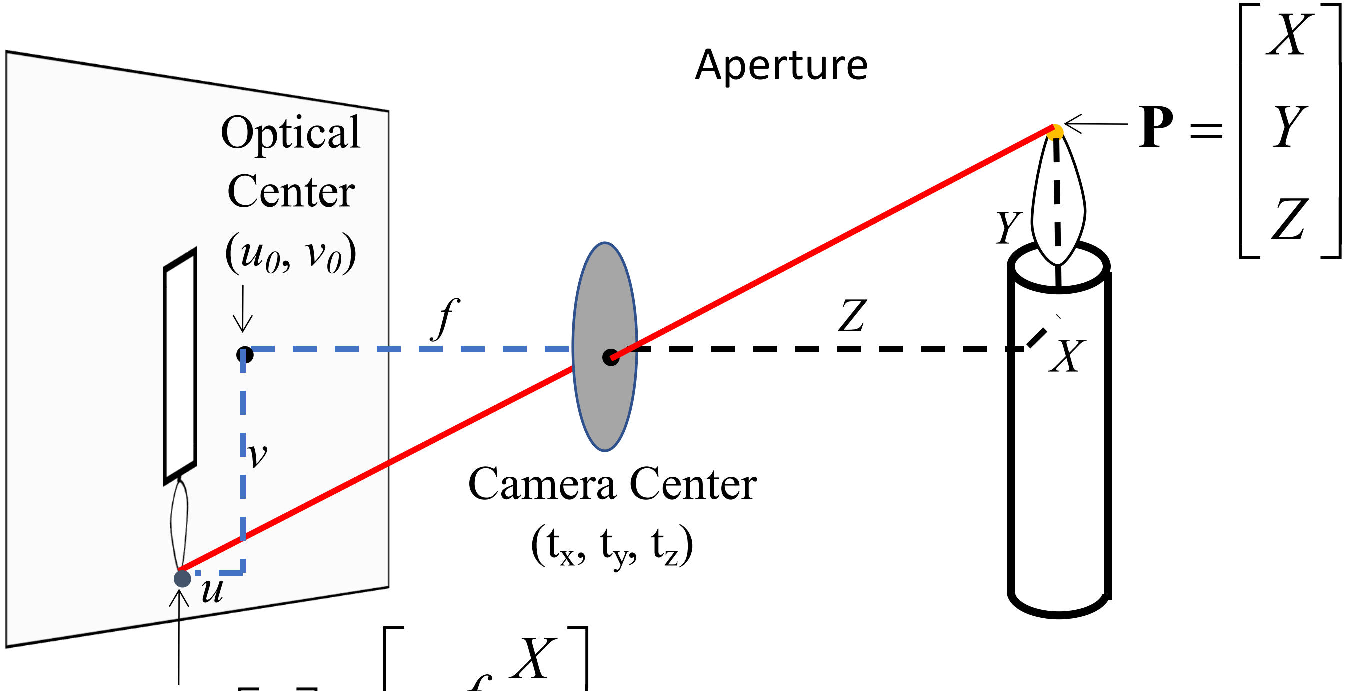
Slide by Steve Seitz



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

Projection:
world coordinates→image coordinates



$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f\dfrac{X}{Z} \\ -f\dfrac{Y}{Z} \end{bmatrix}$$

Non-linear transformation.

Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates

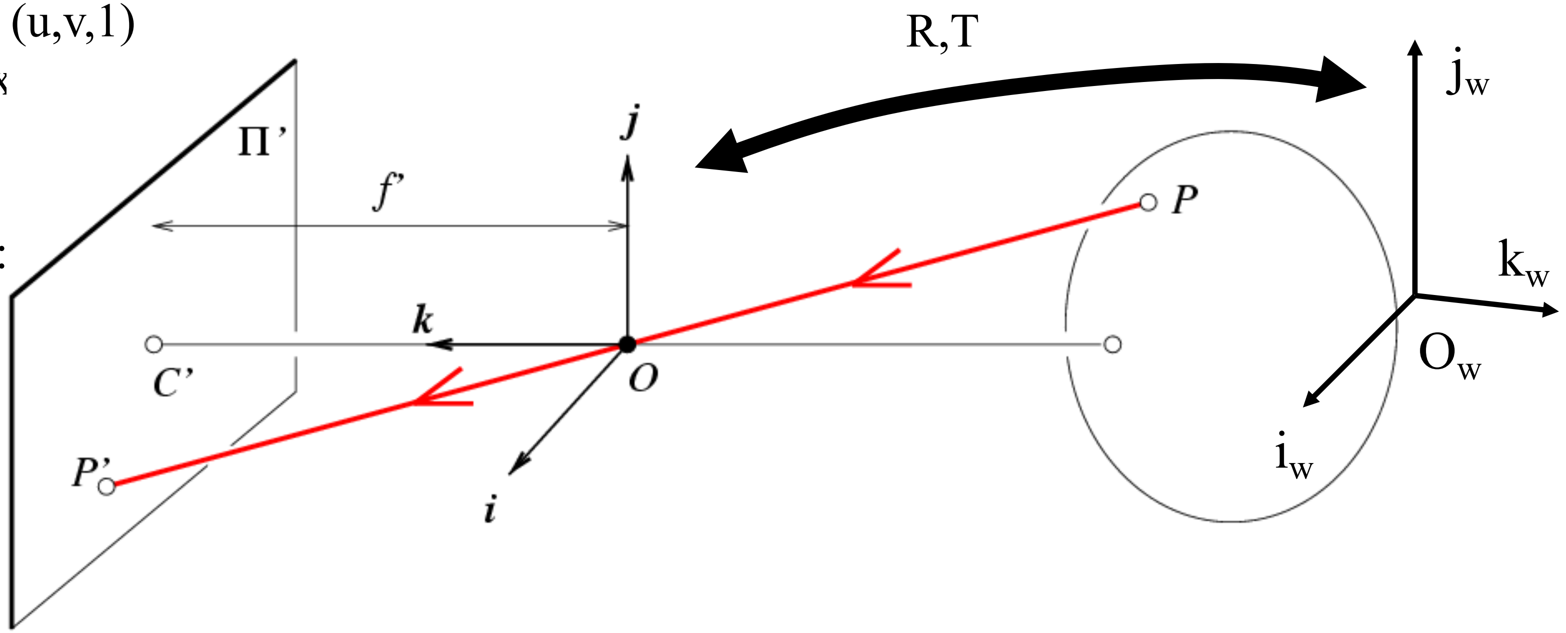
$$\mathbf{P'} = \begin{bmatrix} f\,x \\ f\,y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Basically, **P' = MP**

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

Projection matrix

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x
R: Rotation (3x3)
t: Translation (3x1)
X: World Coordinates:



$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

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

Degrees of freedom?

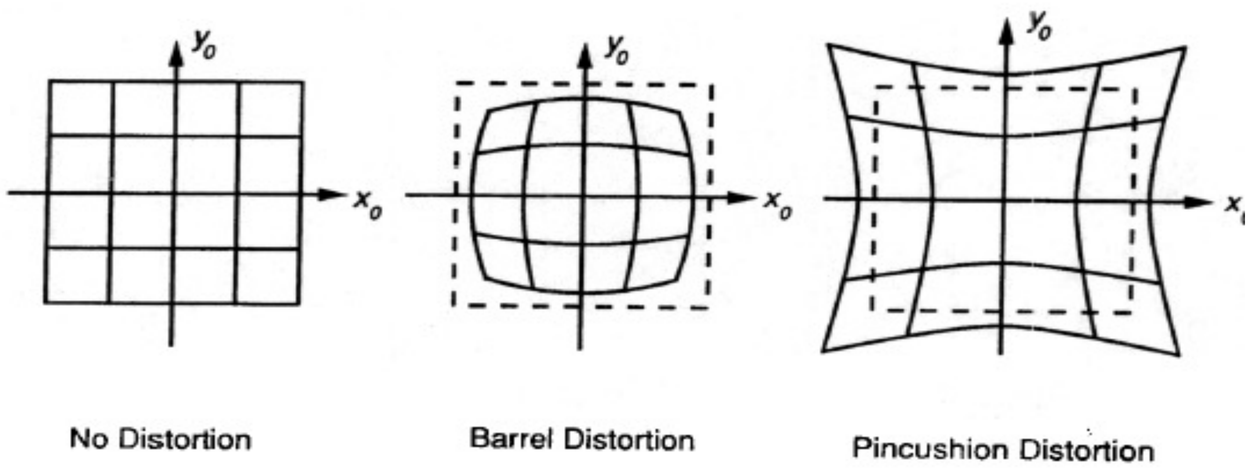
Vanishing Point = Projection from Infinity

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

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

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



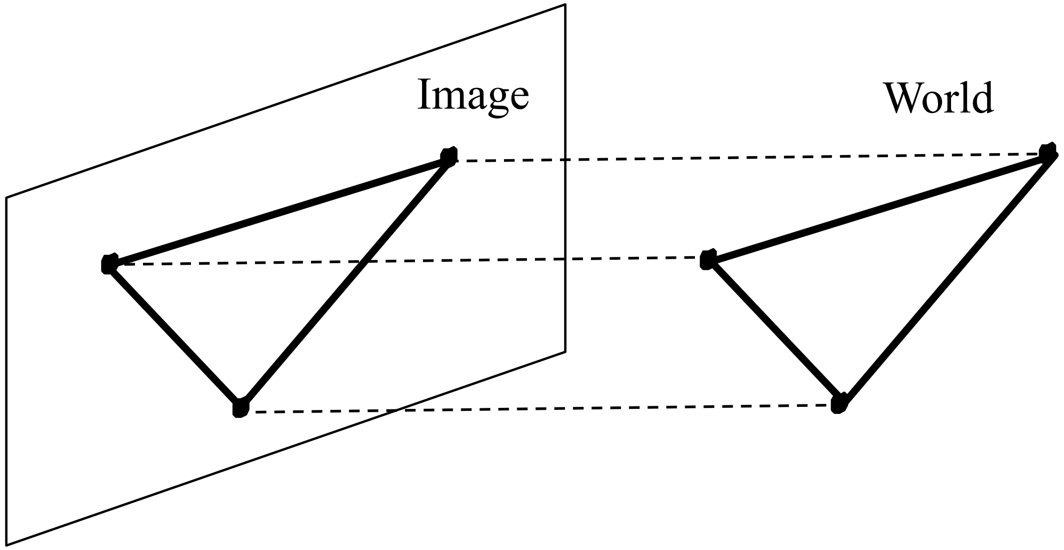
Corrected Barrel Distortion

Image from Martin Habbecke

Scaled Orthographic Projection

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

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



- Also called "weak perspective"

  s =1: orthographic (parallel projection)

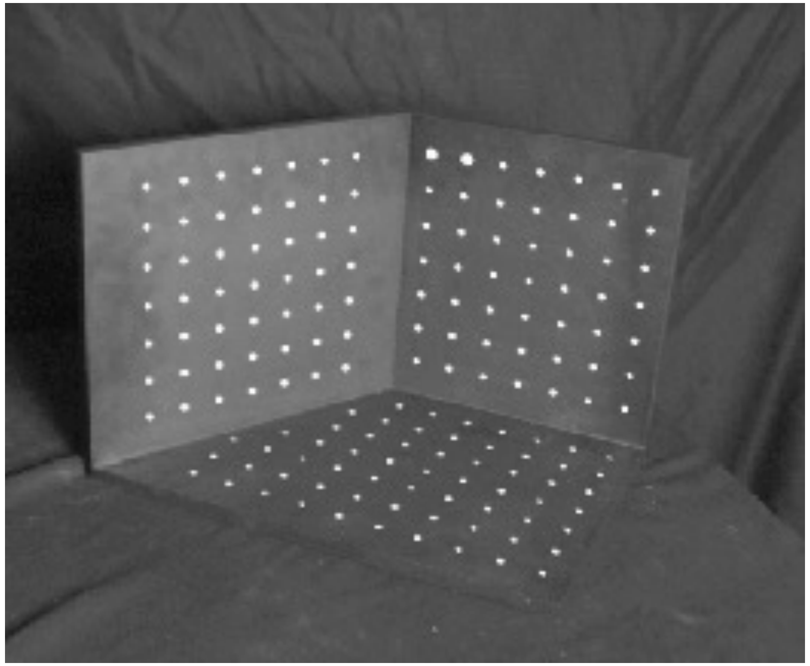
$$\mathbf{x = K[R\ t]SX}$$

Slide by Steve Seitz

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

Known 3d locations

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

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

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

```
[U, S, V] = svd(A);
M = V(:,end);
M = reshape(M,[],3)';
```

Camera Calibration: Finding intrinsics and extrinsics

Method 2 – nonhomogeneous linear system. Solve for m using linear least squares

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

Ax=**b** form

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

```
M = A\Y;
M = [M;1];
M = reshape(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]?

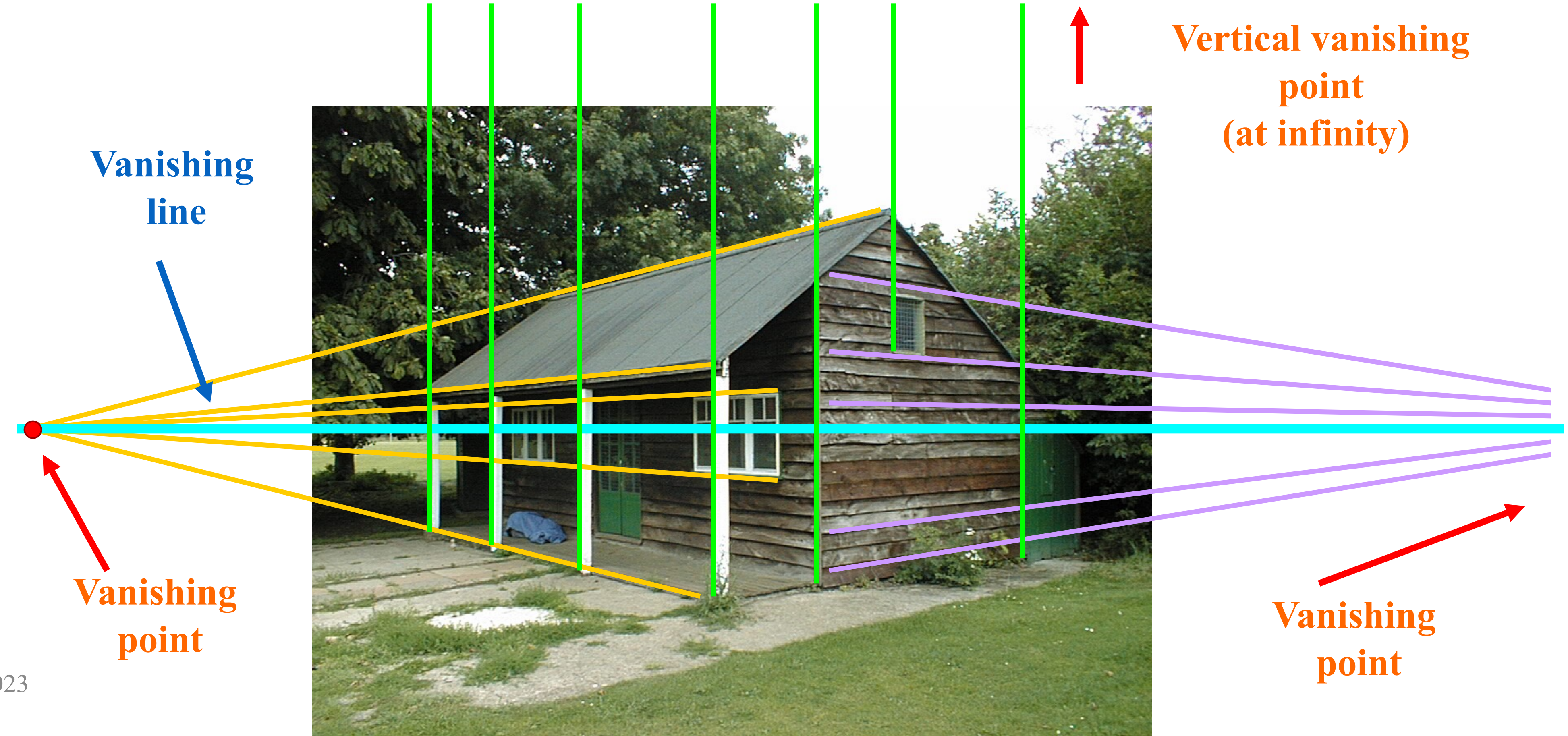
- Yes!
- You can use *RQ* factorization (note – not the more familiar *QR* factorization). *R* (right diagonal) is K, and *Q* (orthogonal basis) is R. T, the last column of [R | T], is inv(K) * last column of M.
  - But you need to do a bit of post-processing to make sure that the matrices are valid.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

Slide credit: J. Hays

Calibrating the Camera

Method 2: Use vanishing points

- Find vanishing points corresponding to orthogonal directions



Calibration by orthogonal vanishing points

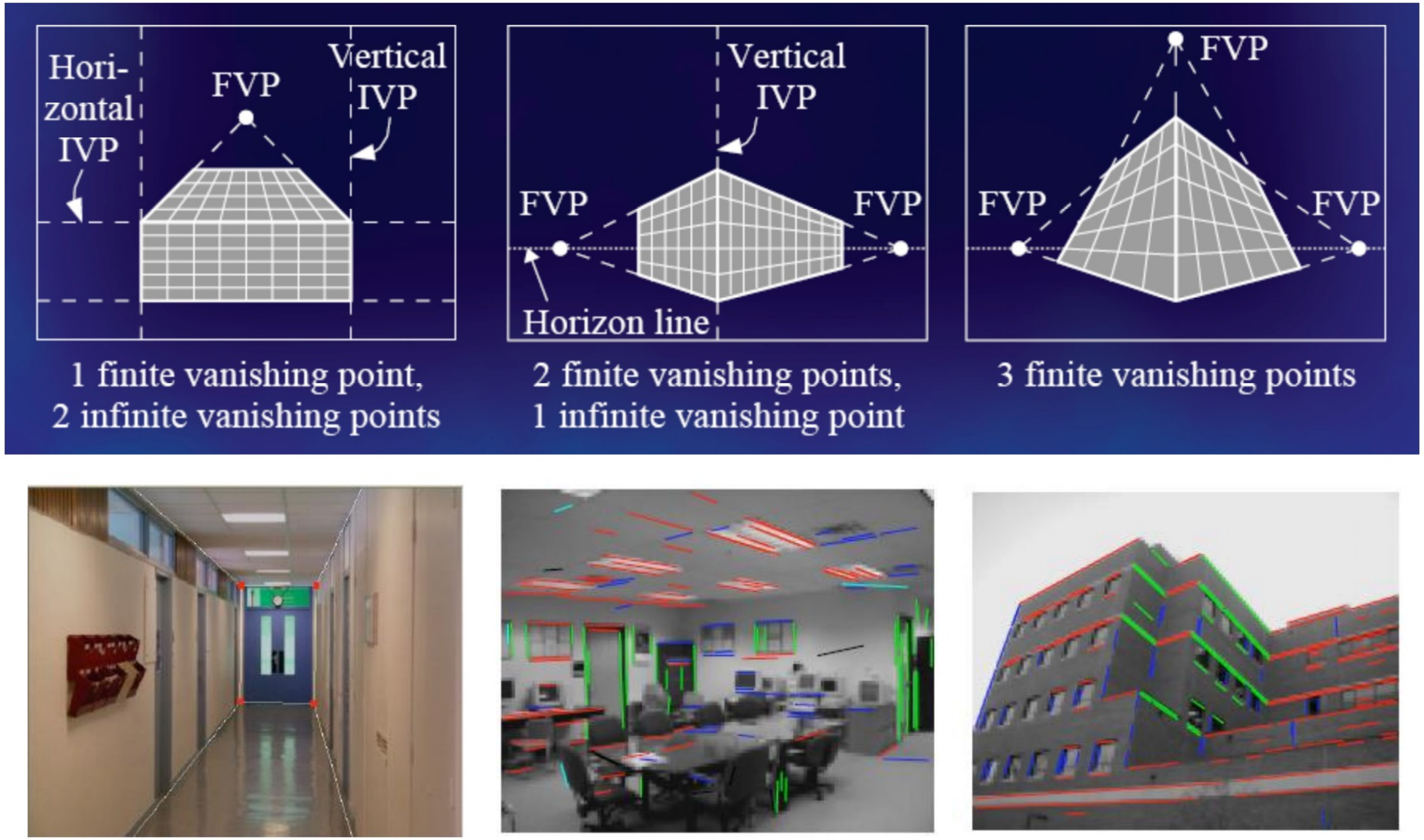
- Align world coordinate frame to 3 orthogonal vanish point directions. Model K with only f, u_0, v_0

$$\mathbf{e_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e_2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e_3} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \lambda \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} = \mathbf{KRe}_i$$

$$\mathbf{e}_i = \lambda \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_i, \quad \mathbf{e}_i^T \mathbf{e}_j = 0$$

$$\mathbf{v}_i^T \mathbf{K}^{-T} \mathbf{R} \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_j = \mathbf{v}_i^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{v}_j = 0$$

Calibration by orthogonal vanishing points



What if you don't have three finite vanishing points?

Two finite VP: solve f, get valid u_0, v_0 closest to image center

One finite VP: u_0, v_0 is at vanishing point; can't solve for f

Calibration by vanishing points

- Rotation matrix

$$\lambda \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} = \mathbf{K}\mathbf{R}\mathbf{e}_i$$

$$\lambda \mathbf{K}^{-1}\mathbf{v}_1 = \mathbf{R}\mathbf{e}_1 = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3]\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{r}_1$$

Thus, $\lambda \mathbf{K}^{-1}\mathbf{v}_i = \mathbf{r}_i$.

Get λ by using the constraint $||\mathbf{r}_i||^2=1$.

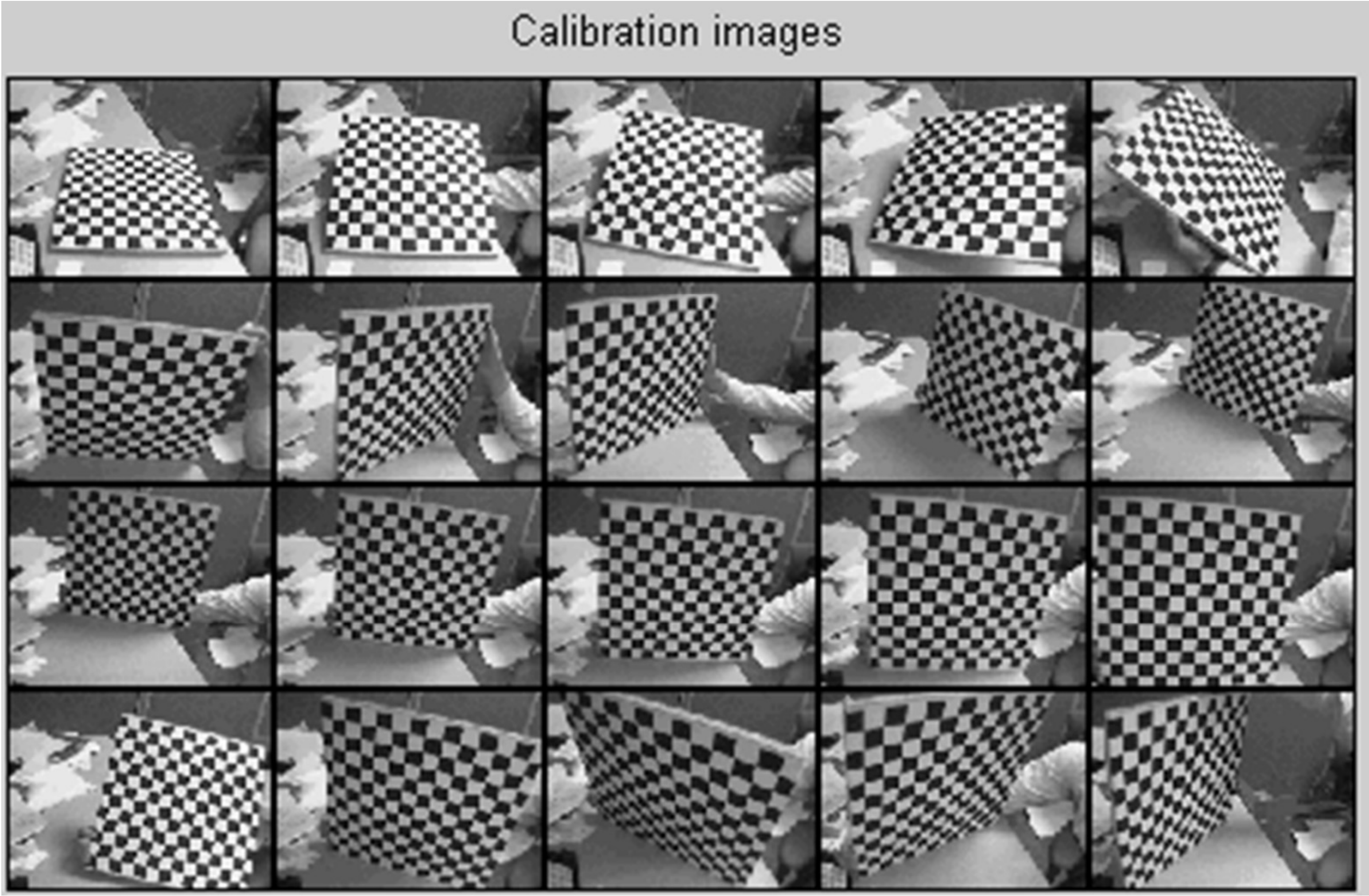
Calibration by vanishing points

- Advantage: Could be automatic, no calibration chart needed
- Disadvantage: Need to have 2-3 vanishing points, not all scenes have that. Inaccurate computation of vanishing point will impact.

- More on camera calibration:

http://vision.stanford.edu/teaching/cs231a_autumn1112/lecture/lecture8_camera_calibration_cs231a.pdf

Calibration Images



Project papers:

Tangent Sampson Error: Fast Approximate Two-view Reprojection Error for Central Camera Models. ICCV 2023. Mikhail Terekhov, Viktor Larsson.

https://openaccess.thecvf.com/content/ICCV2023/html/Terekhov_Tangent_Sampson_Error_Fast_Approximate_Two-view_Reprojection_Error_for_Central_ICCV_2023_paper.html

Fast Globally Optimal Surface Normal Estimation from an Affine Correspondence. ICCV 2023. Levente Hajder, Lajos Lóczi, Daniel Barath.

https://openaccess.thecvf.com/content/ICCV2023/papers/Hajder_Fast_Globally_Optimal_Surface_Normal_Estimation_from_an_Affine_Correspondence_ICCV_2023_paper.pdf

Deep Geometry-Aware Camera Self-Calibration from Video. ICCV 2023.Annika Hagemann, Moritz Knorr, Christoph Stiller.

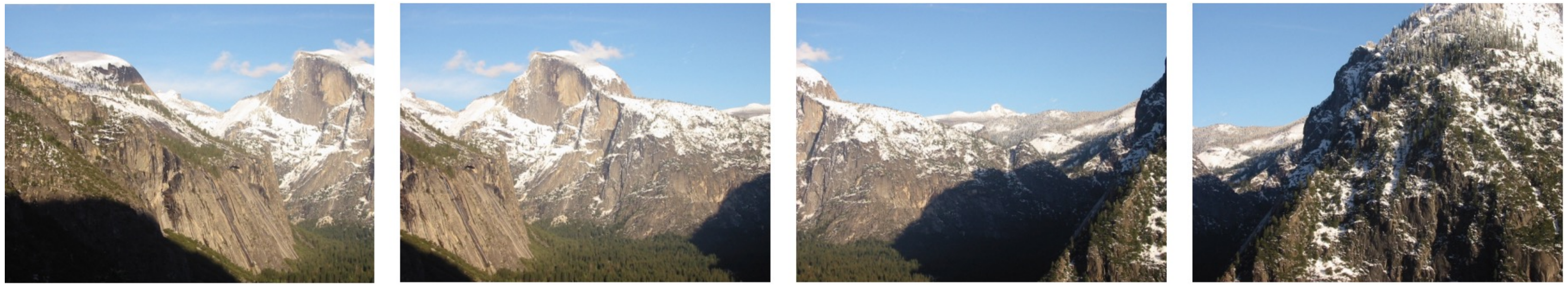
https://openaccess.thecvf.com/content/ICCV2023/papers/Hagemann_Deep_Geometry-Aware_Camera_Self-Calibration_from_Video_ICCV_2023_paper.pdf

P1AC: Revisiting Absolute Pose From a Single Affine Correspondence. ICCV 2023.Jonathan Ventura, Zuzana Kukelova, Torsten Sattler, Dániel Baráth.

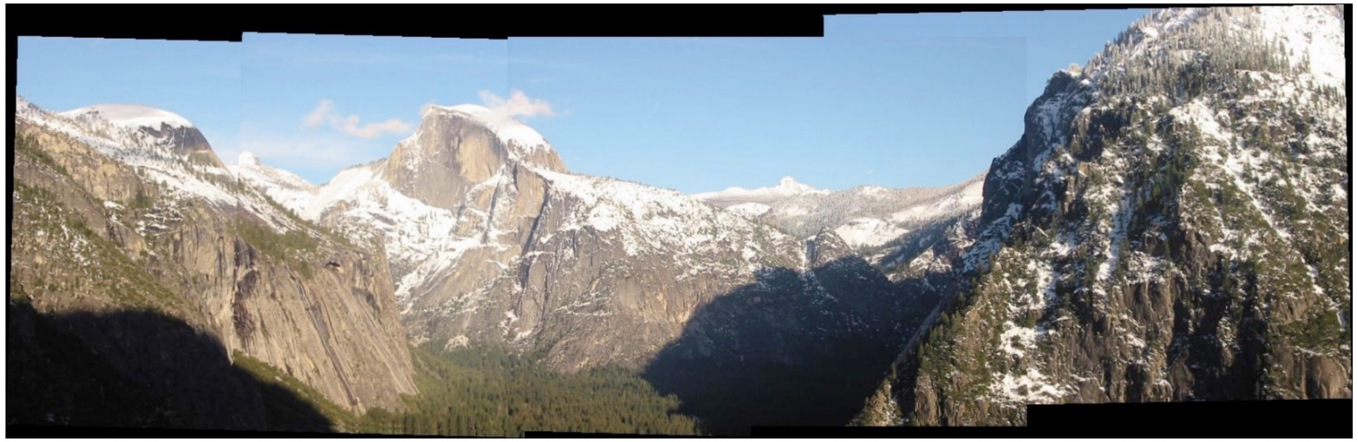
https://openaccess.thecvf.com/content/ICCV2023/papers/Ventura_P1AC_Revisiting_Absolute_Pose_From_a_Single_Affine_Correspondence_ICCV_2023_paper.pdf

Panoramic Image Stiching

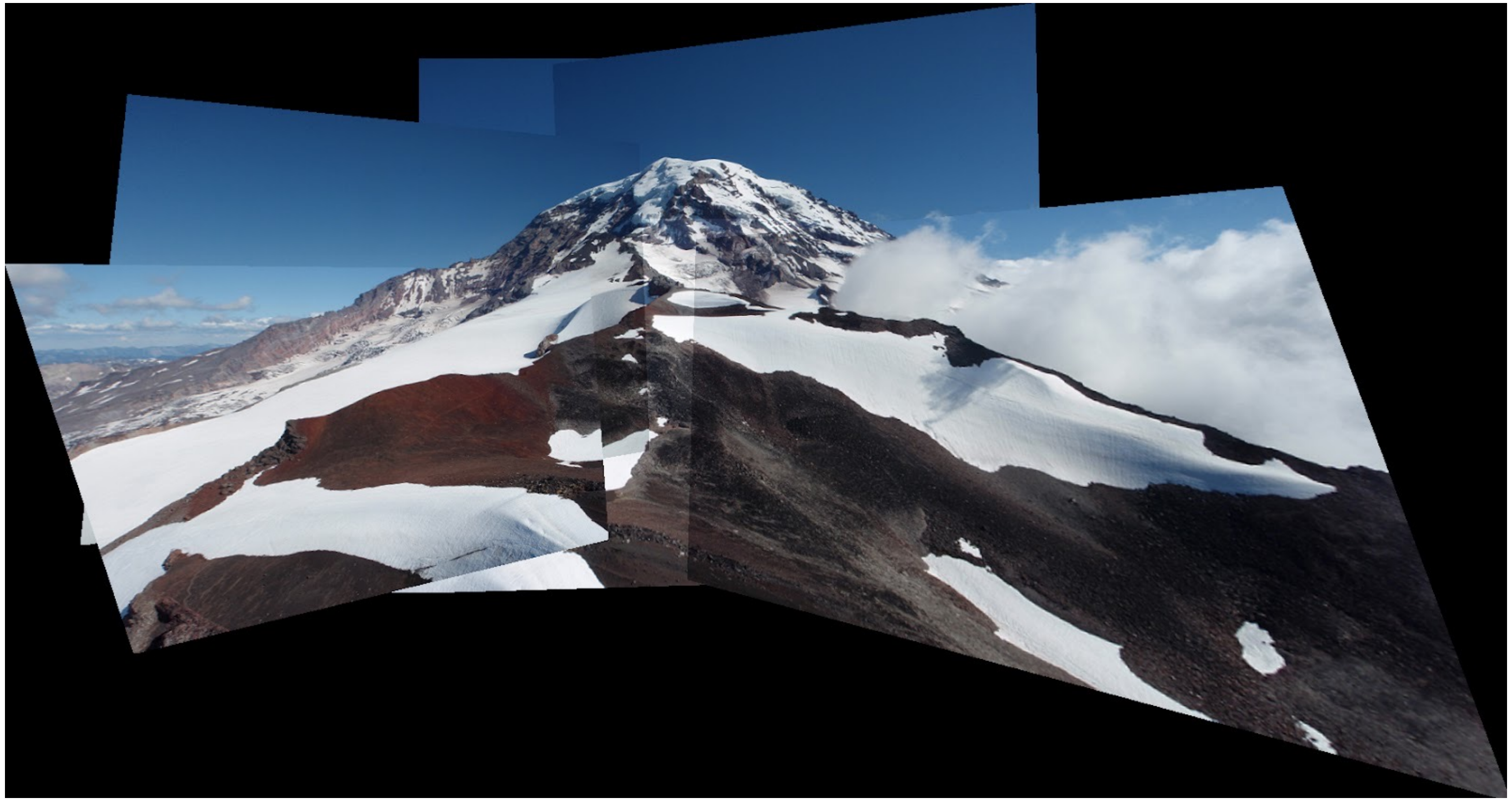
Given a set of images

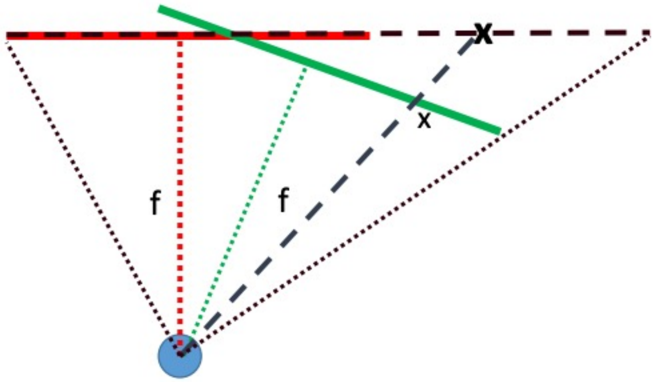


Obtain:



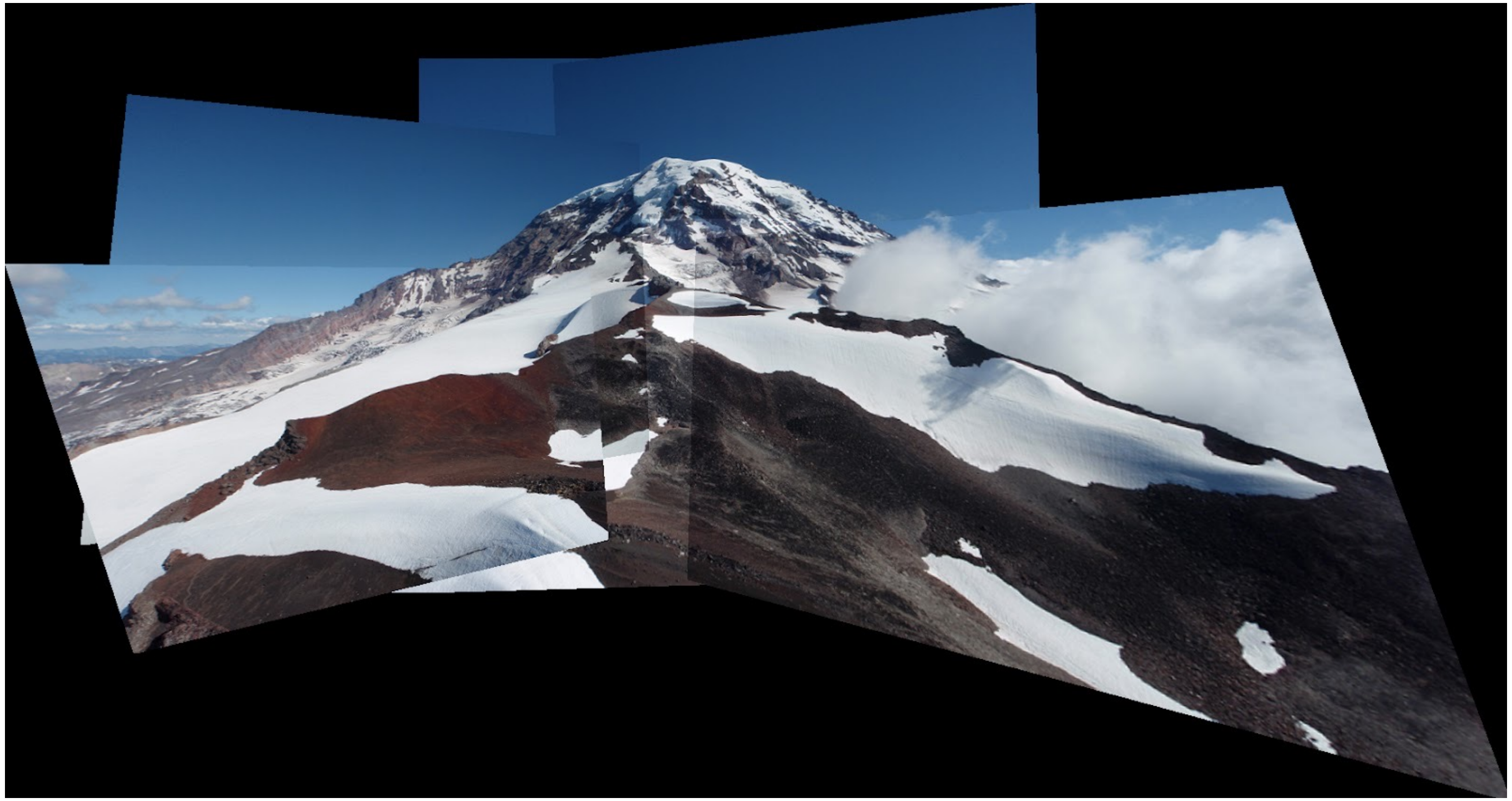
In Practice, we map planes

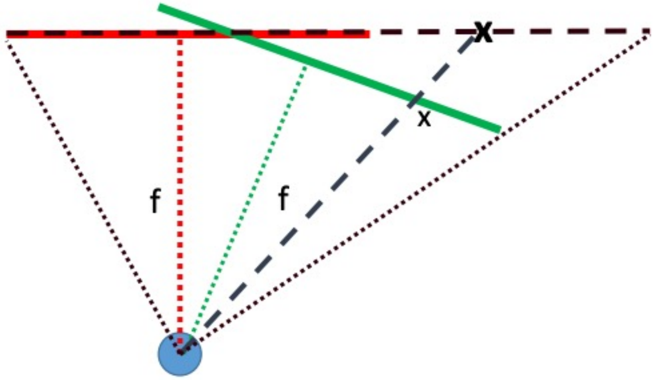




For red image: pixels are already on the plane
For green image: map to first image plane

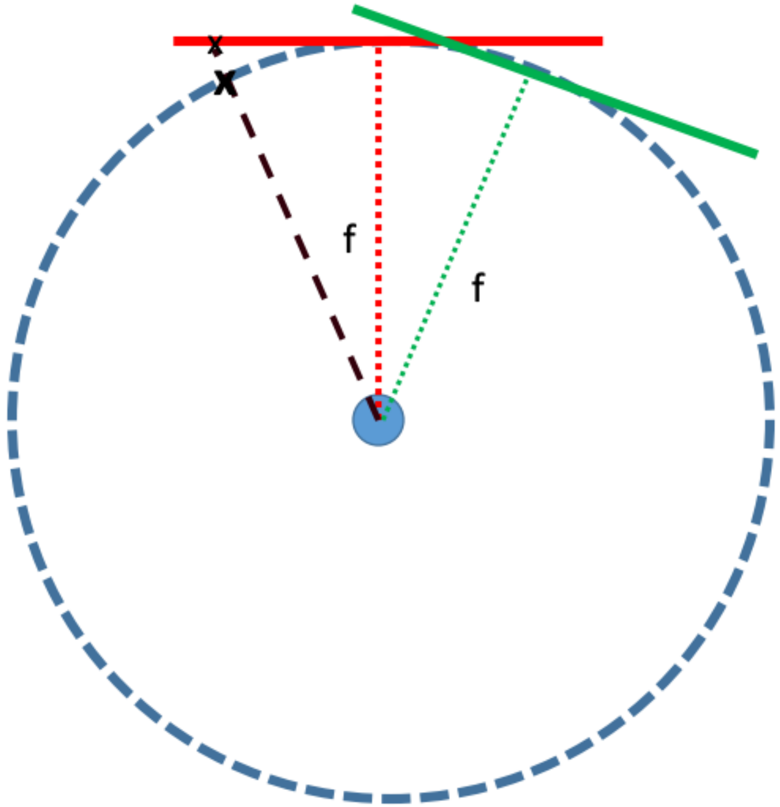
Planar mapping





For red image: pixels are already on the plane
For green image: map to first image plane

Solution: Use cylindrical mapping instead



For red image: compute h, theta on cylindrical surface from (u, v)
For green image: map to first image plane, than map to cylindrical surface

Solution: Use cylindrical mapping instead

Calculate angle and height:
$\theta = (x - xc) / f$
$h = (y - yc) / f$

Find unit cylindrical coords:
$X' = \sin(\theta)$
$Y' = h$
$Z' = \cos(\theta)$

Project to image plane:
$x' = f\, X'/Z' + xc$
$y' = f\, Y'/Z' + yc$