# Camera Models

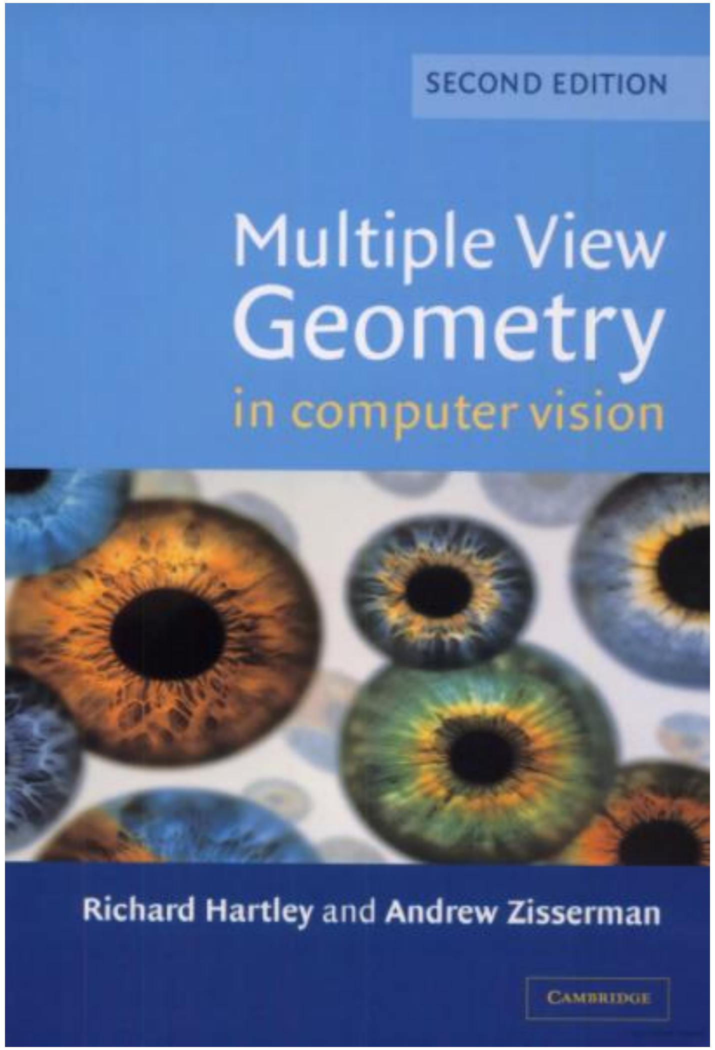
Computer Vision – Lecture 14

(Generative Models postponed to Lecture 16)

# Further Reading

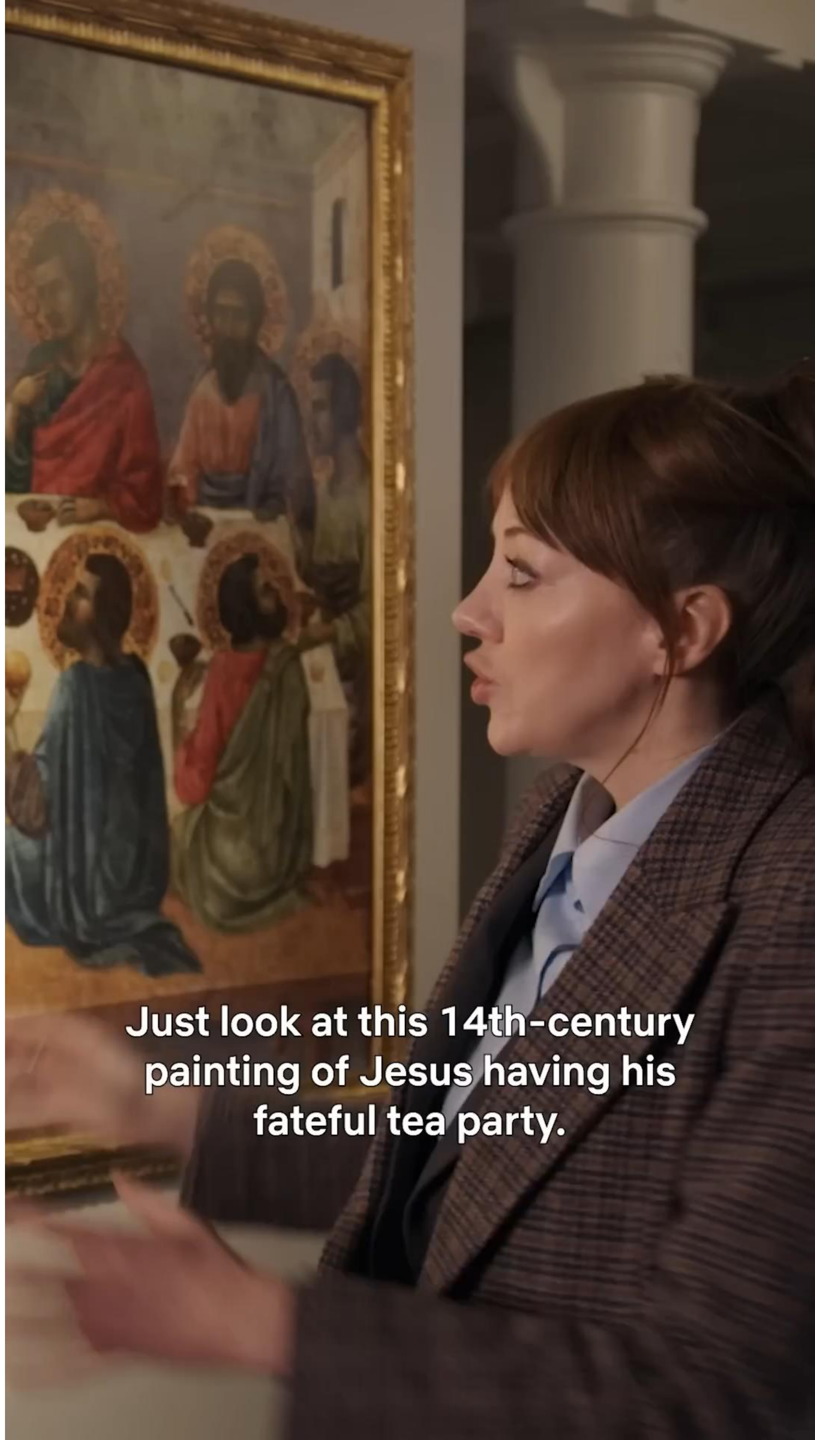
- 7 lectures from S Lazebnik
- Slides from D Fouhey and J Johnson
- Many slides adapted from both sources

# Multi-view geometry “Bible”



# Perspective in Art

- Eastern art: planar perspective 11th century.
- Western art: early Renaissance: 15th century
- Before: attempts to simulate perspective but no good understanding.



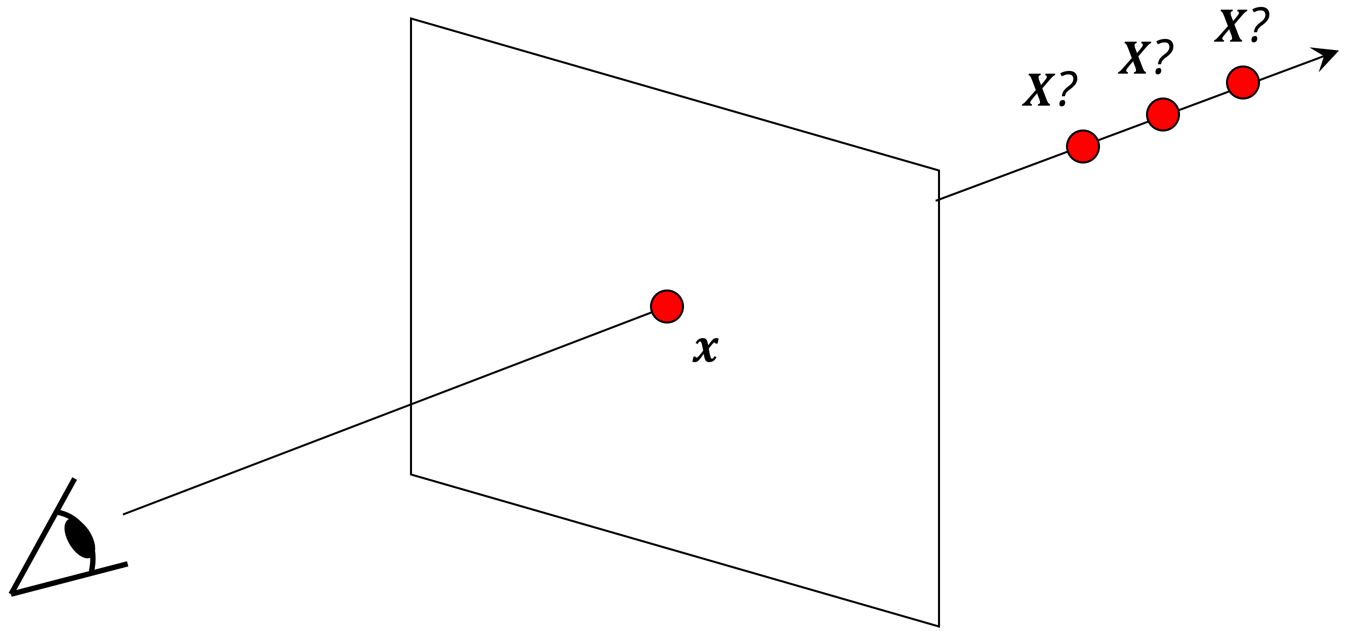

Diane Morgan in “Cunk on Earth”, 2022

# Things aren't always as they appear…




Video source

# Single-view ambiguity



# Single-view ambiguity

[Rashad Alakbarov shadow sculptures]

# Anamorphic perspective

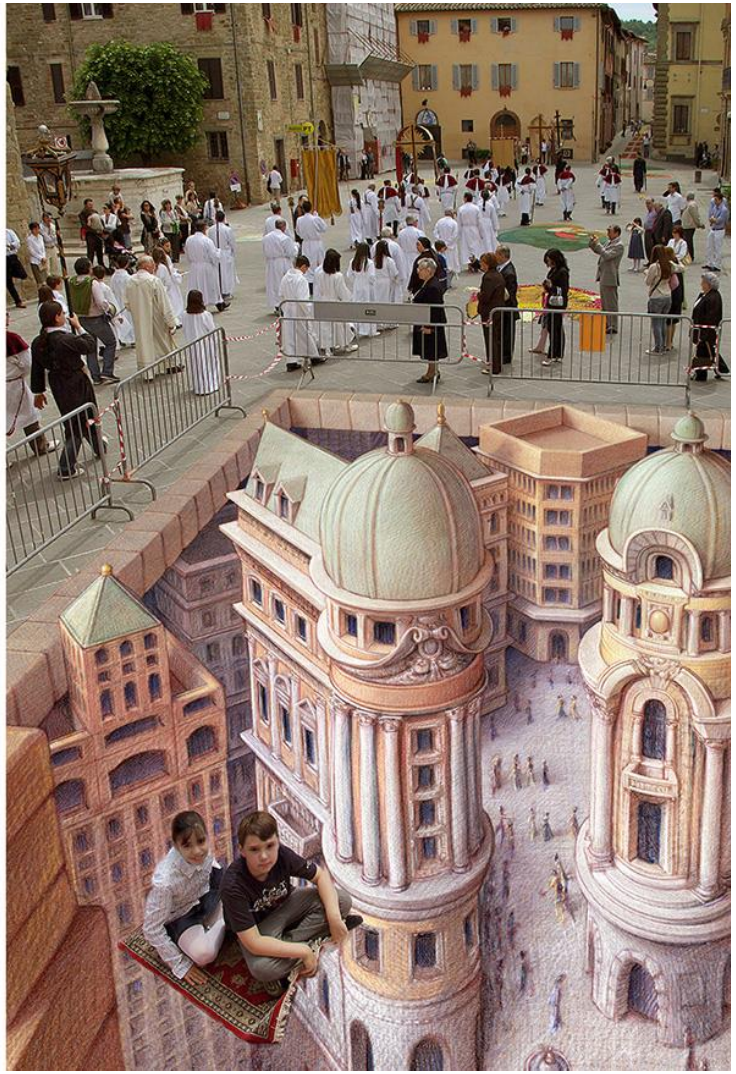




Image source

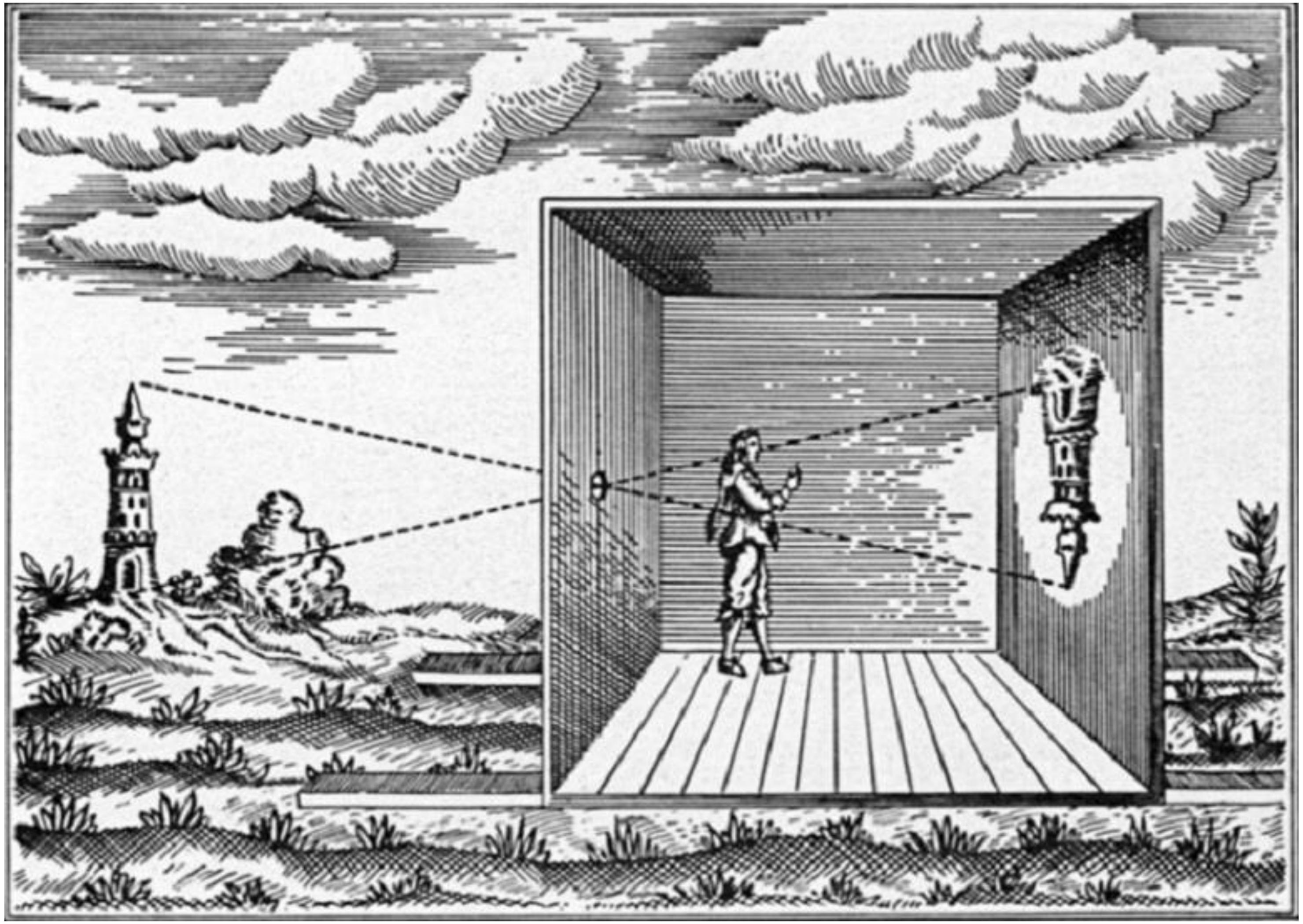
# Anamorphic perspective

H. Holbein The Younger, *The Ambassadors*, 1533

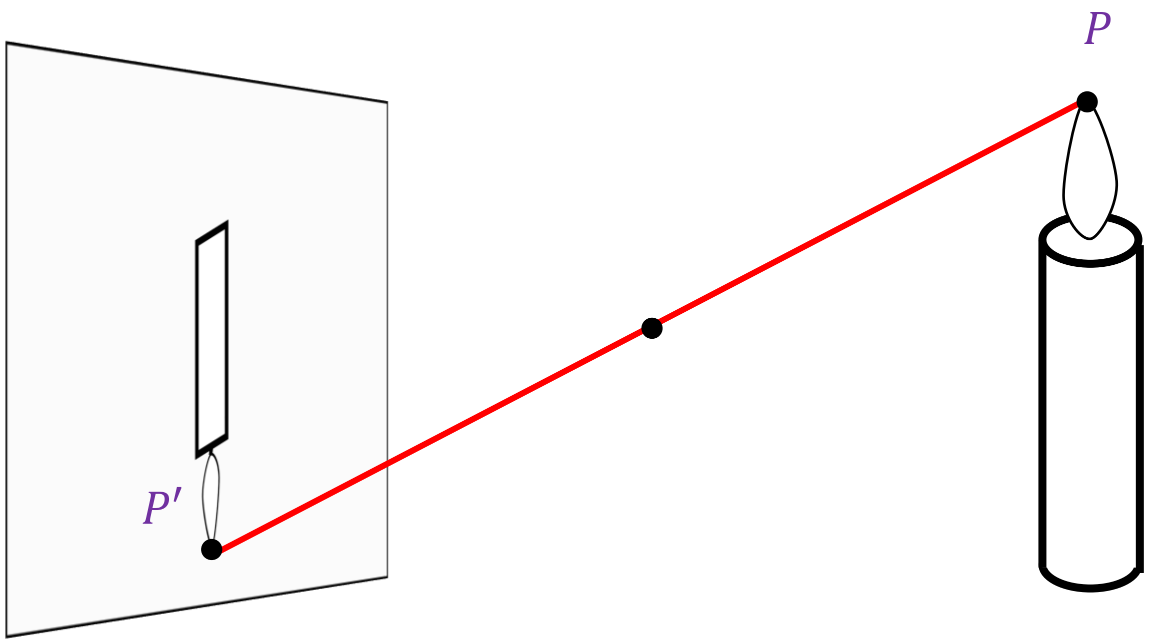


https://en.wikipedia.org/wiki/Anamorphosis

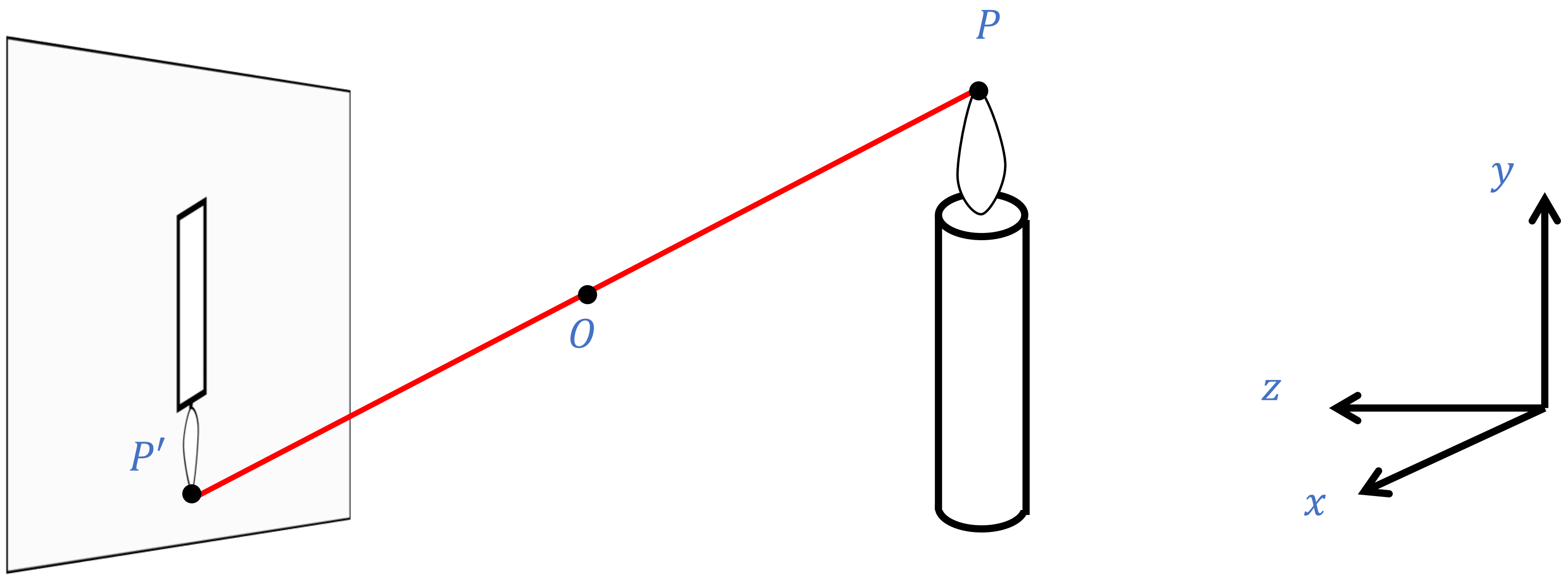
# Camera Obscura



# Pinhole Camera



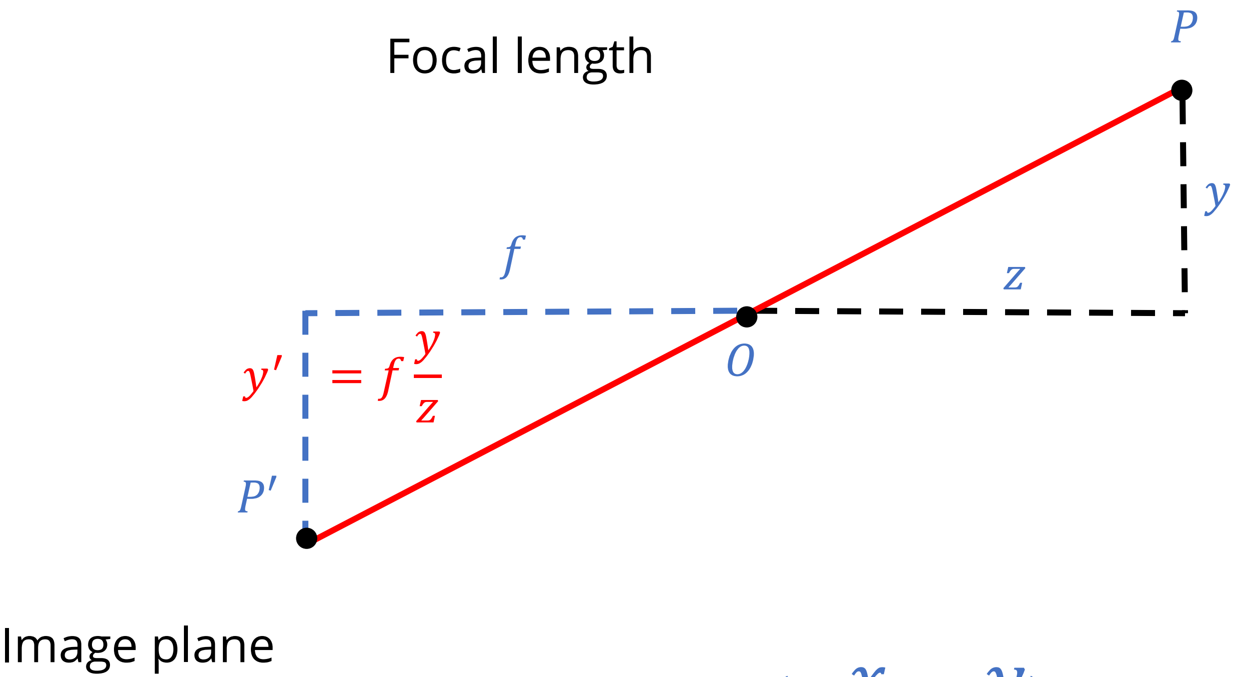
# Perspective Projection




## Camera Coordinate System

- The optical center ($O$) is at the origin
- The z axis is the *optical axis* perpendicular to the image plane
- The xy plane is parallel to the image plane, x and y axes are horizontal and vertical directions of the image plane

# Perspective Projection

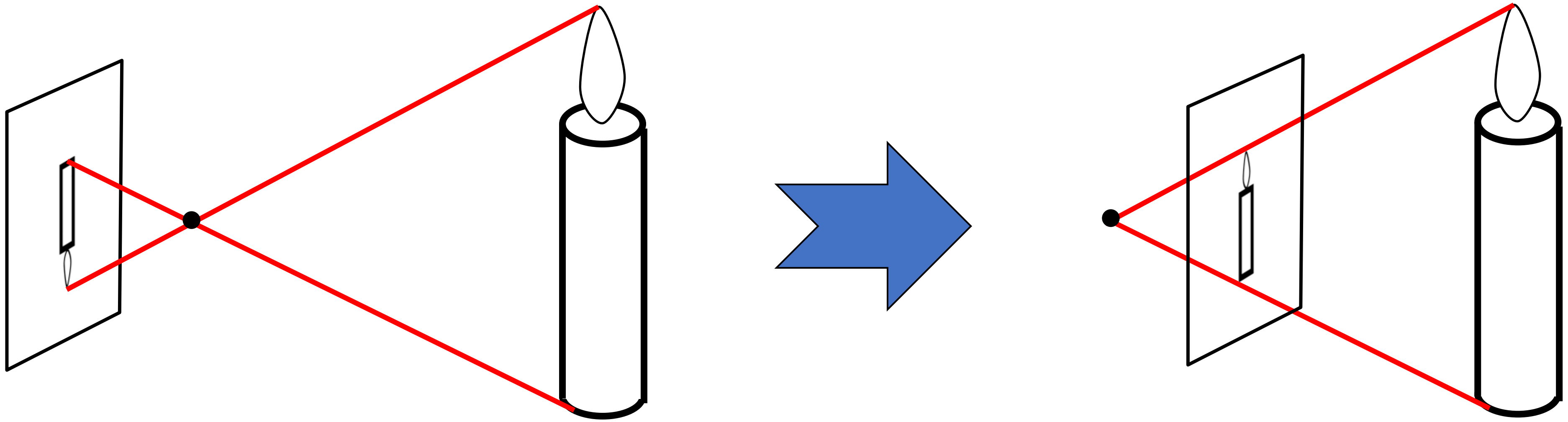



$$(x, y, z) \rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

3D point 2D point

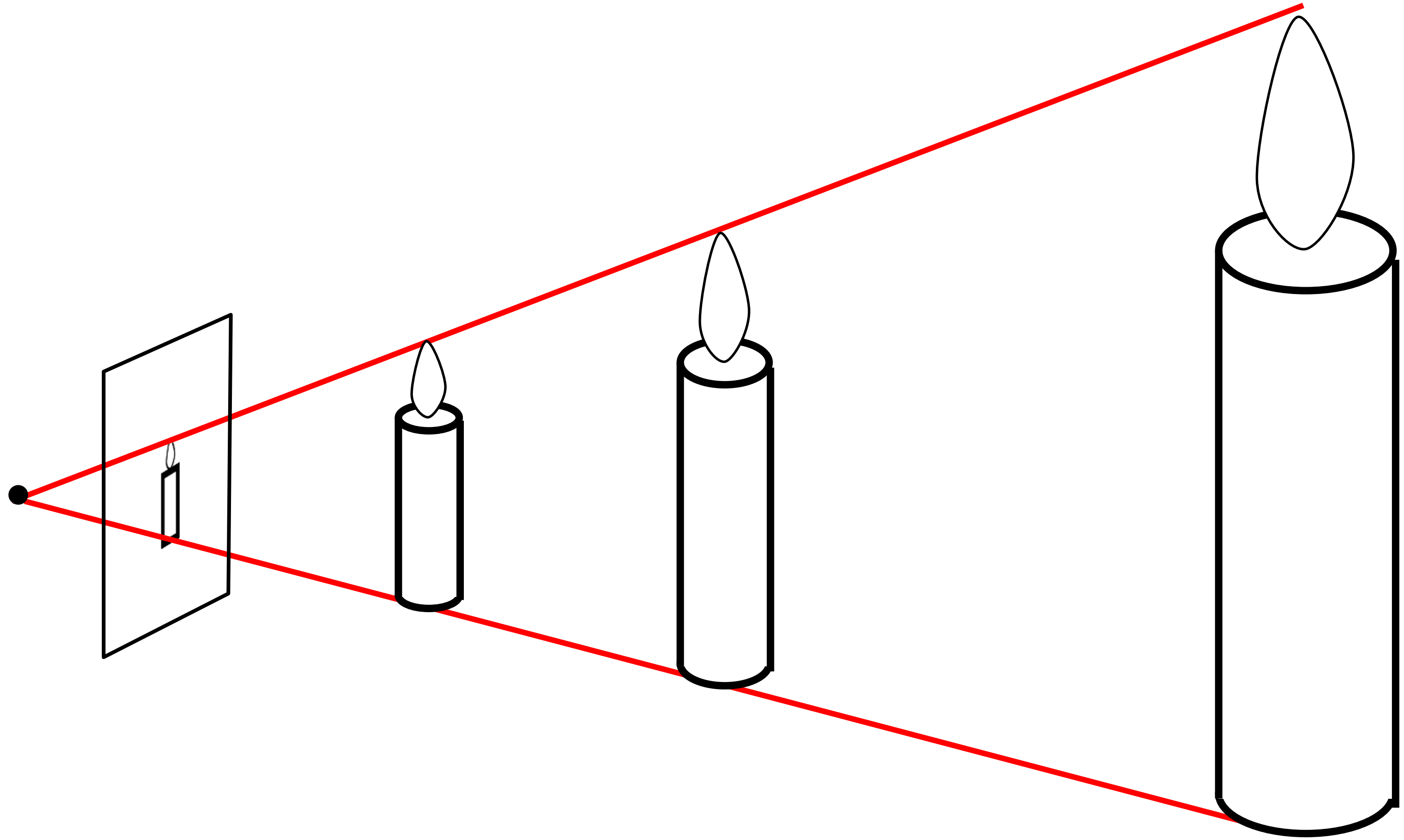
# Perspective Projection

Instead of dealing with an image that is upside down, most of the time we will pretend that the image plane is *in front* of the camera center



# Single View Ambiguity

From a single image, there is an inherent ambiguity about the scale of the observed object.

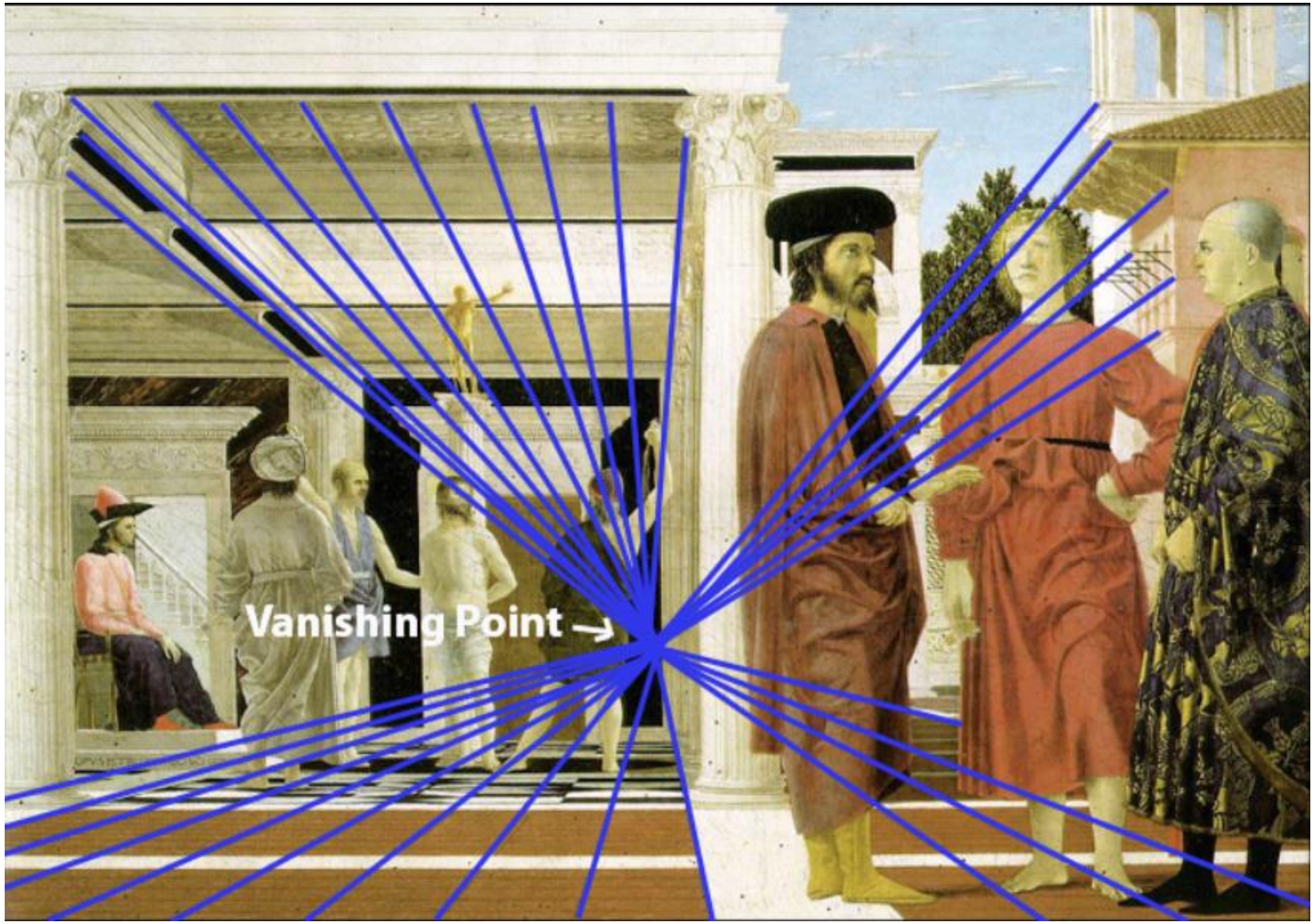


# Projection of lines


Piero della Francesca, *Flagellation of Christ*, 1455-1460

# Projection of lines

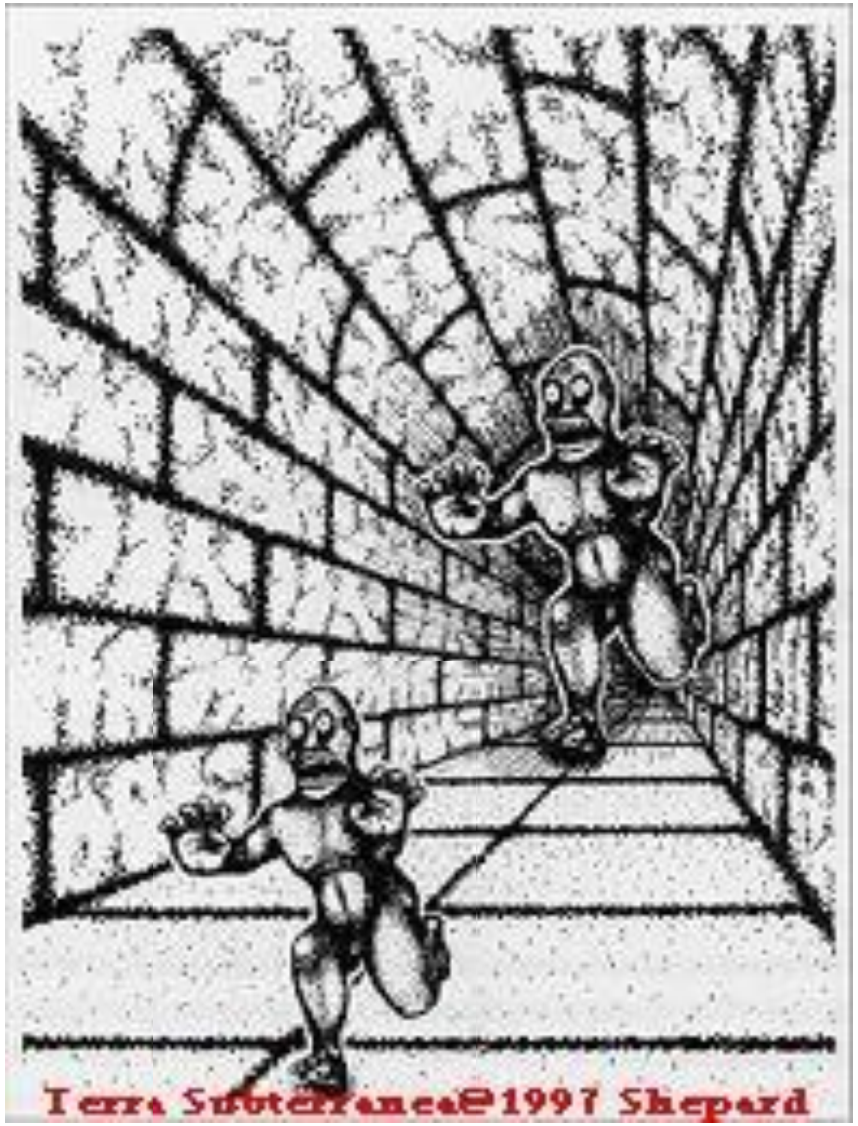
- Parallel lines meet at a *vanishing point*




Piero della Francesca, *Flagellation of Christ*, 1455-1460

Image source

# Perspective Cues






Slide by Steve Seitz

# Projection of 3D shapes

What is the shape
of the projection
of a sphere?

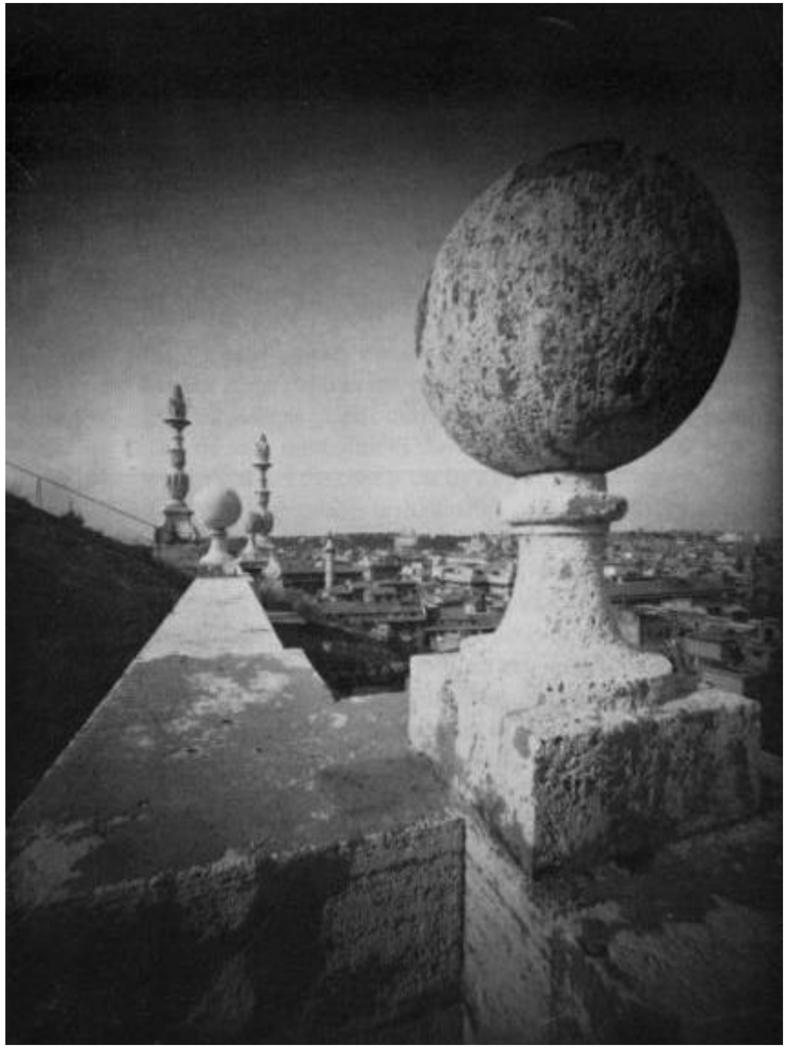
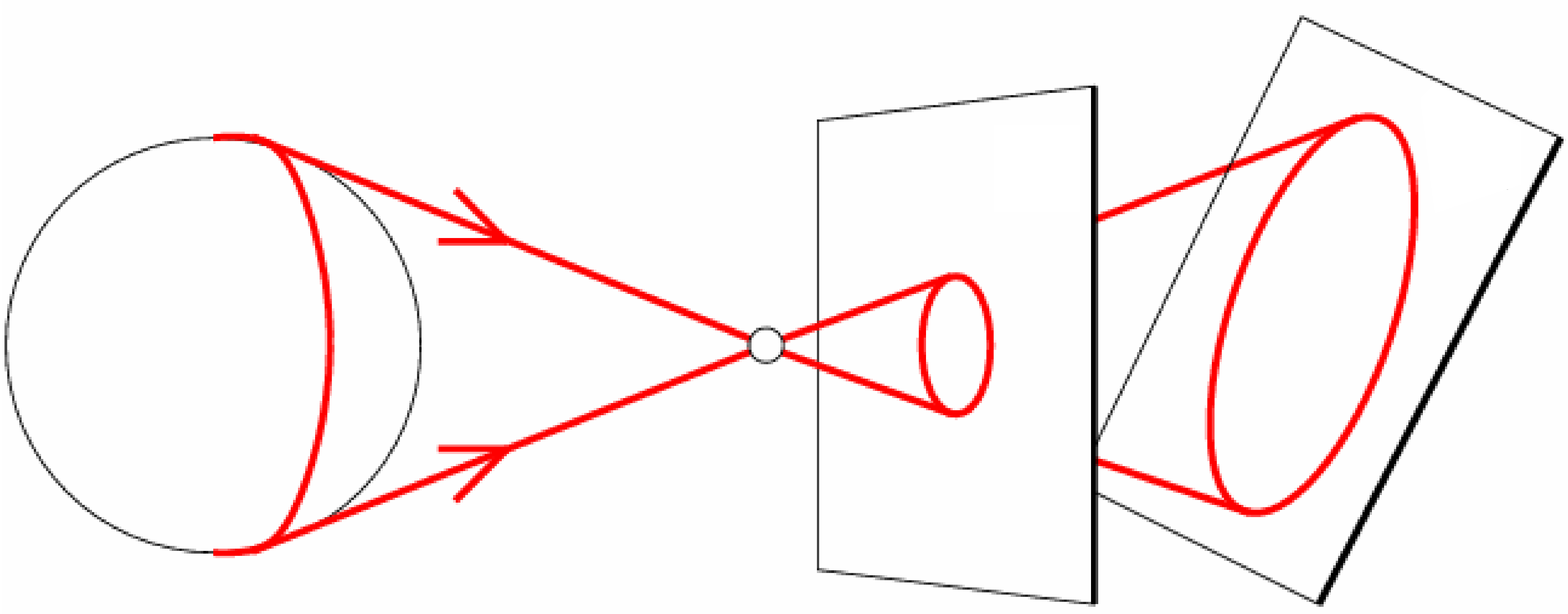


Image source: F. Durand

# Projection of 3D shapes

- What is the shape of the projection of a sphere?



# Homogeneous Coordinates III

To form homogeneous coordinates from normal Euclidean coordinates, append 1 as the last entry:

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

homogeneous *image* coordinates

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

homogeneous *scene* coordinates

To convert *from* homogeneous coordinates, divide by the last entry:

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

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

In homogeneous coordinates, all scalar multiples represent the same point!

# Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates:

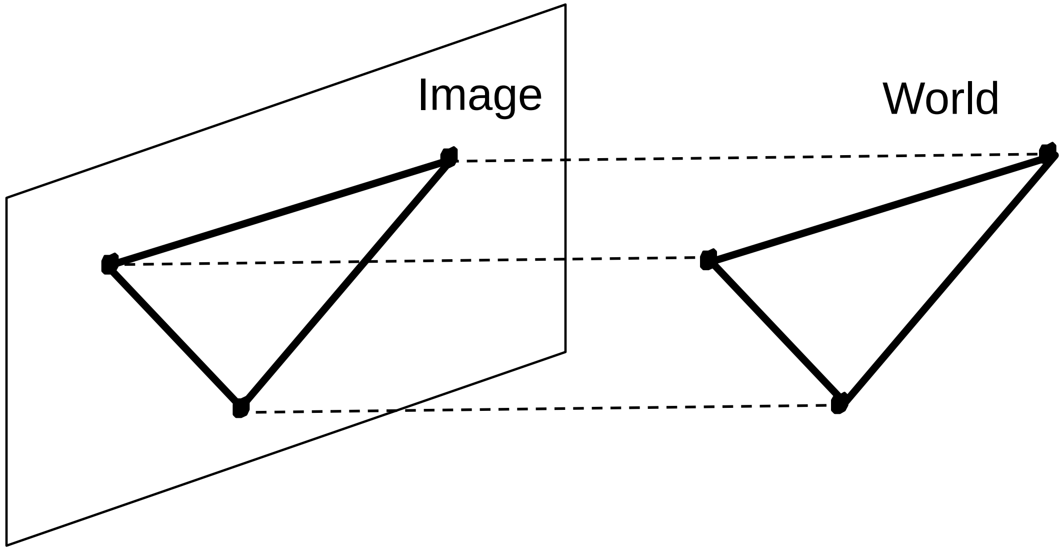
$$\begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix} \Longrightarrow \left(f\,\frac{x}{z}, f\,\frac{y}{z}\right)$$

divide by the third coordinate

# Orthographic projection

Special case of perspective projection

- Distance from center of projection to image plane is infinite
- Also called “parallel projection”



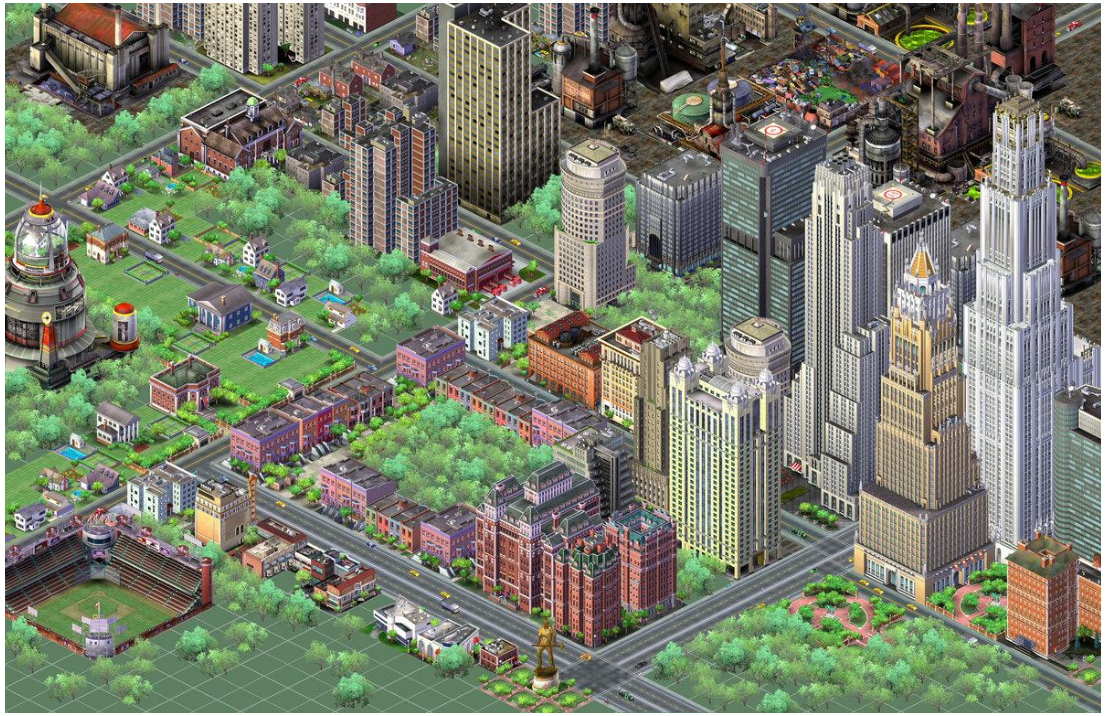



Slide by Steve Seitz

# Orthographic projection

Special case of perspective projection

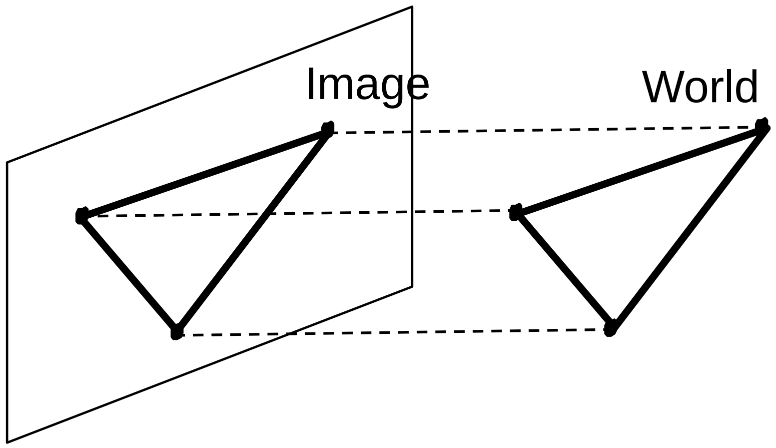
- Distance from center of projection to image plane is infinite
- Also called “parallel projection”



# Orthographic projection

Special case of perspective projection

- Distance from center of projection to image plane is infinite
- Also called “parallel projection”



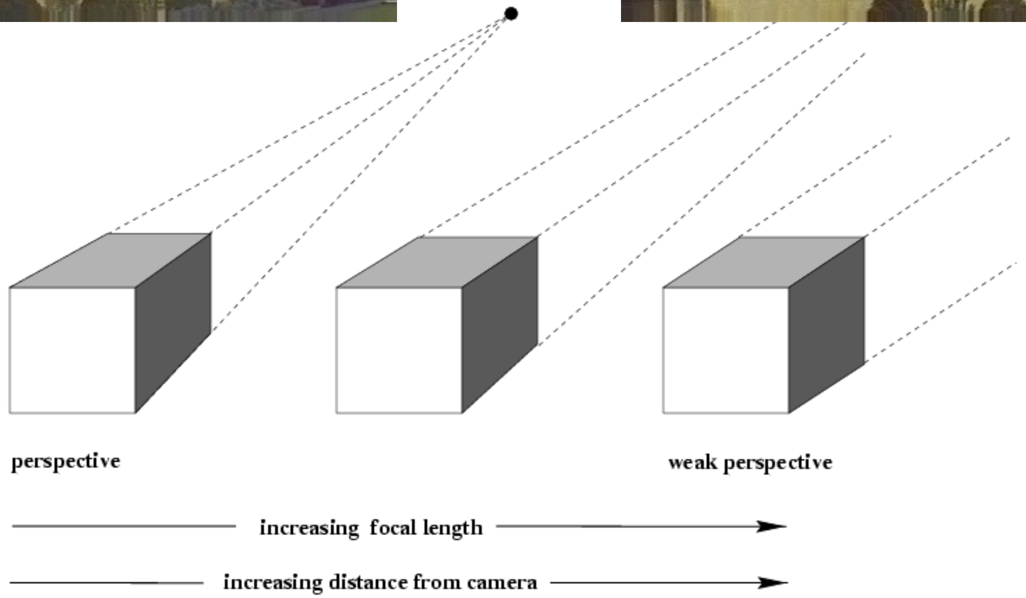

- Assuming projection along the $z$ axis, what’s the matrix?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Slide by Steve Seitz

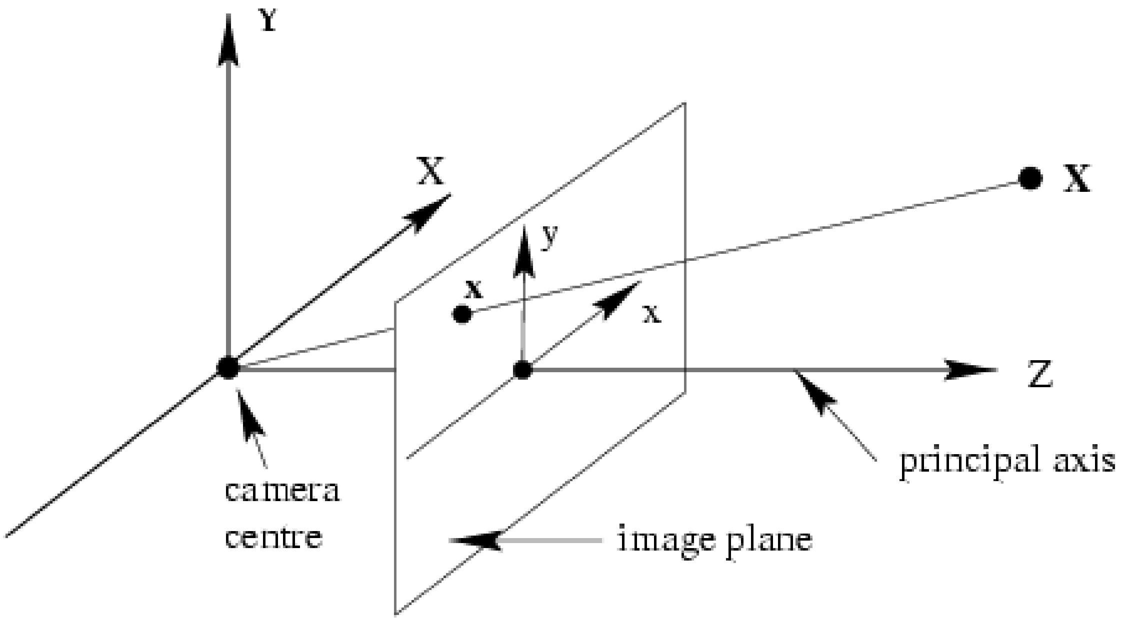
# Approximating an orthographic camera




center at
infinity

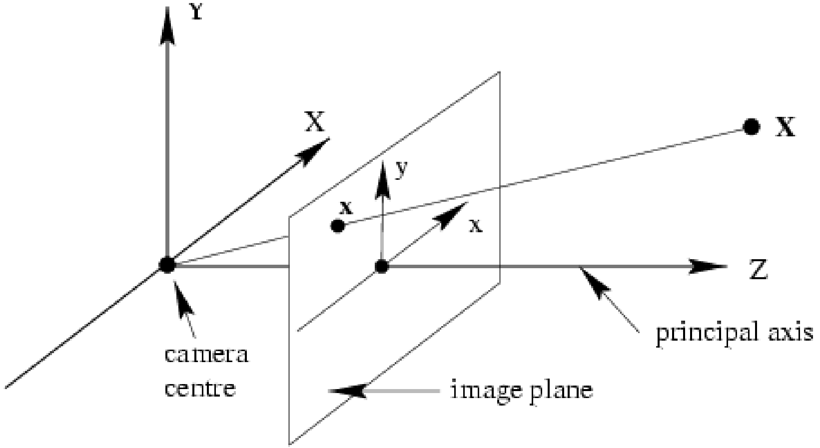
Source: Hartley & Zisserman

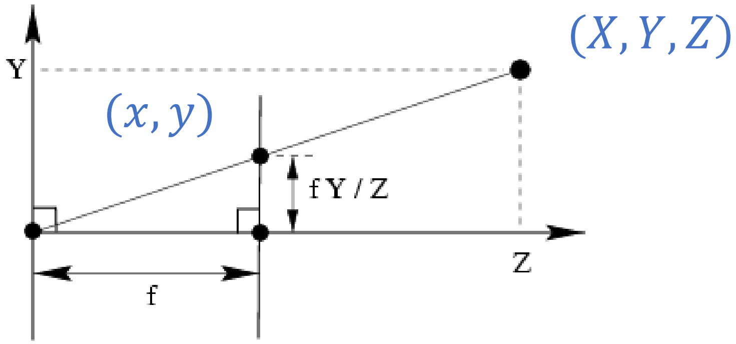
# Normalized Coordinates




**Normalized (camera) coordinate system:** camera center is at the origin, the *principal axis* is the $z$-axis, $x$ and $y$ axes of the image plane are parallel to $x$ and $y$ axes of the world

# Perspective projection in normalized coordinates




$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Homogeneous coord. vec. $\boldsymbol{X}$ of 3D point

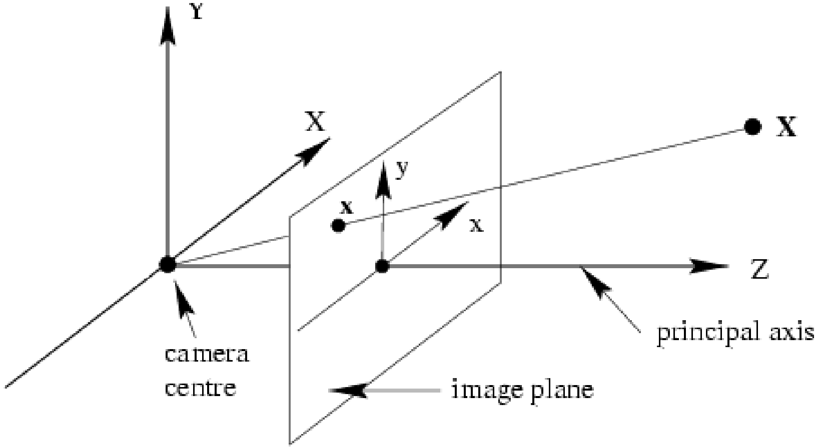
$$\boldsymbol{x} \cong \boldsymbol{PX}$$

Homogeneous coord. vec. $\boldsymbol{x}$ of image point

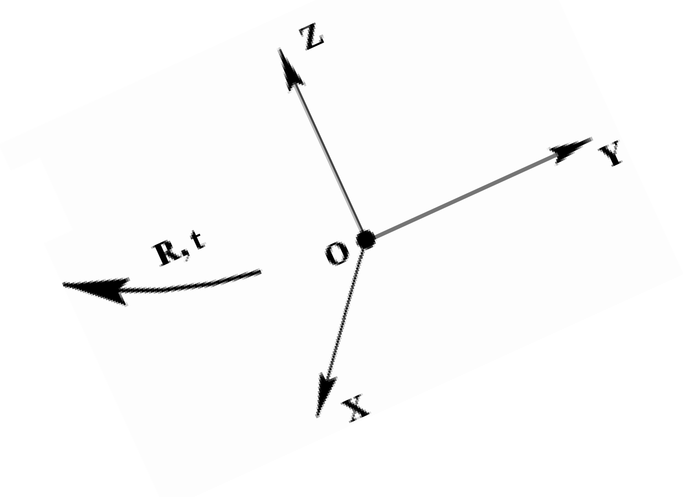
Camera projection matrix $\boldsymbol{P}$

Equality up to scale

# Camera Calibration




world coordinate system



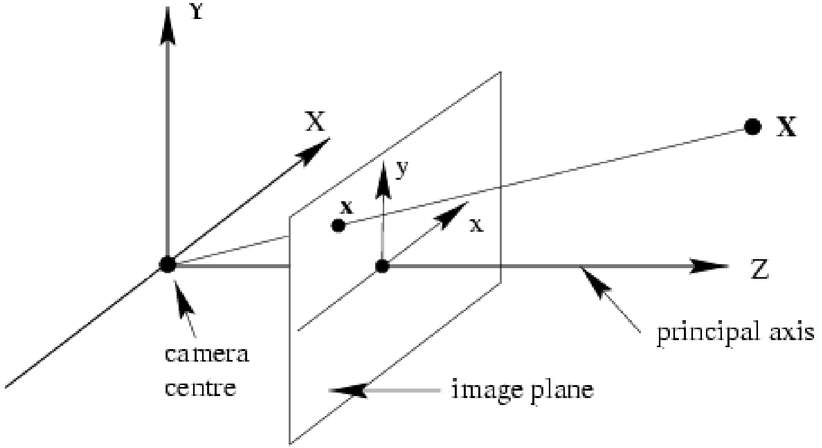

**Camera calibration:** figuring out transformation from *world* coordinate system to *image* coordinate system

(2D point $\boldsymbol{x}$ (3x1)) $\cong$ (Camera to pixel coord. trans. matrix $\boldsymbol{K}$ (3x3)) (*Canonical* projection matrix $[\boldsymbol{I} \mid \mathbf{0}]$ (3x4)) (World to camera coord. trans. matrix $\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$ (4x4)) (3D point $\boldsymbol{X}$ (4x1))

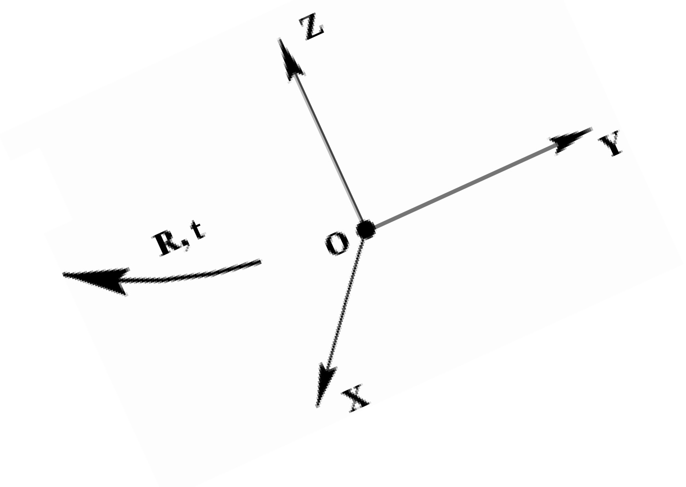
***Intrinsic* camera parameters**: principal point, scaling factors

***Extrinsic* camera parameters**: rotation, translation

# Camera Calibration




world coordinate system

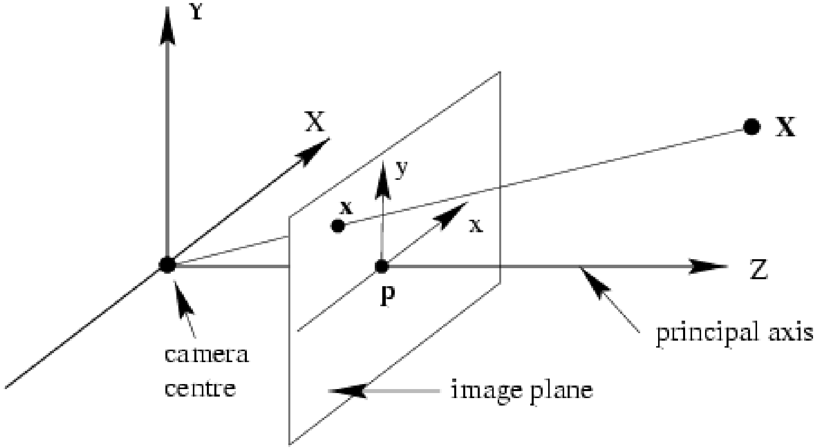


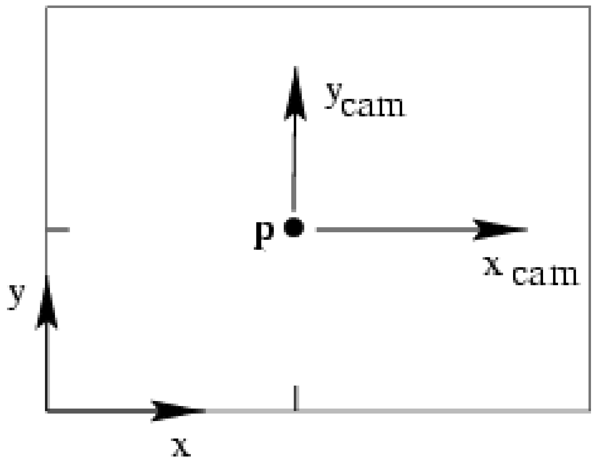

**Camera calibration:** figuring out transformation from *world* coordinate system to *image* coordinate system

$$\begin{pmatrix} \text{2D} \\ \text{point} \\ \boldsymbol{x} \\ \text{(3x1)} \end{pmatrix} \cong \underbrace{\begin{pmatrix} \text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \boldsymbol{K} \text{ (3x3)} \end{pmatrix} \begin{pmatrix} \textit{Canonical} \\ \text{projection matrix} \\ [\boldsymbol{I} \mid \boldsymbol{0}] \text{ (3x4)} \end{pmatrix} \begin{pmatrix} \text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \text{ (4x4)} \end{pmatrix}}_{\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]} \begin{pmatrix} \text{3D} \\ \text{point} \\ \boldsymbol{X} \\ \text{(4x1)} \end{pmatrix}$$

*General* camera projection matrix

# Intrinsic parameters: Principal point

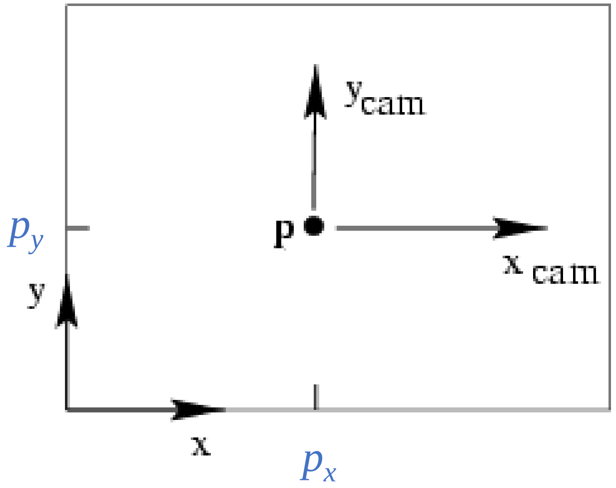



**Principal point (*p*):** point where principal axis intersects the image plane

- In the *normalized* coordinate system, the origin of the image is at the principal point
- In the *image* coordinate system: the origin is in the corner

# Intrinsic parameters: Principal point

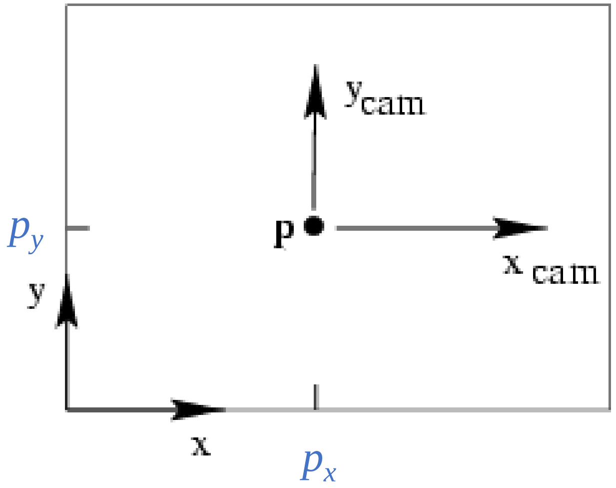



We want the principal point to map to $(p_x, p_y)$ instead of $(0,0)$

$$x = f\frac{X}{Z} + p_x, \qquad y = f\frac{Y}{Z} + p_y$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

# Intrinsic parameters: Principal point




Principal point: $(p_x, p_y)$

$$\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

calibration matrix $\boldsymbol{K}$

Canonical projection matrix $[\boldsymbol{I} \mid \mathbf{0}]$

$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{I}|\mathbf{0}]$

# Intrinsic parameters: Principal point

- What are the units of the focal length $f$ and principal point coordinates $(p_x, p_y)$?
  - Same as world units – likely metric units
- What units do we want for measuring image coordinates?
  - Pixel units
- Thus, we need to introduce *scaling factors* for mapping from world to pixel units

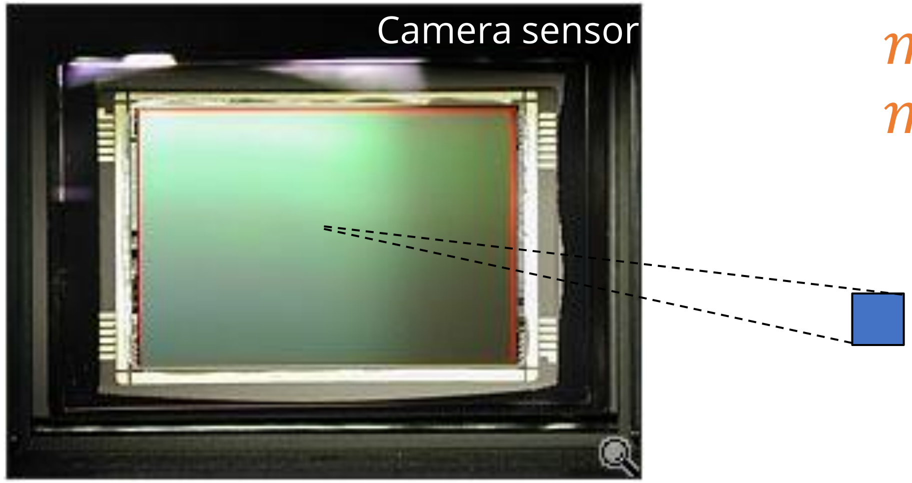
$$\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

calibration matrix $\boldsymbol{K}$

Canonical projection matrix $[\boldsymbol{I} \mid \boldsymbol{0}]$

$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]$

# Intrinsic parameters: Scaling factors

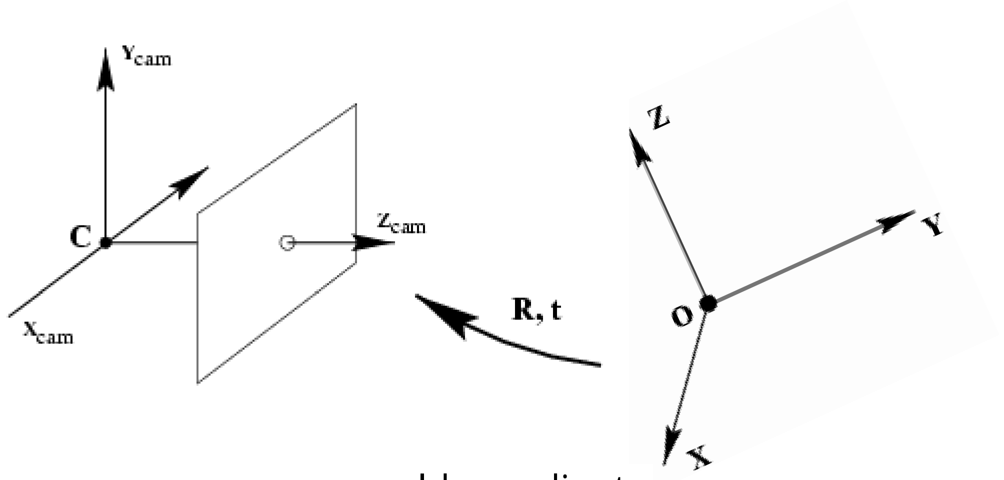



$m_x$ pixels/m in horizontal direction,
$m_y$ pixels/m in vertical direction

Pixel size (m): $\frac{1}{m_x} \times \frac{1}{m_y}$

| Scaling factors | Calibration matrix $\boldsymbol{K}$ in metric units | Calibration matrix $\boldsymbol{K}$ in pixel units |
|---|---|---|
| $\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$ | $= \begin{bmatrix} \alpha_x & 0 & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}$ |
| pixels/m | m | pixels |

# Extrinsic Parameters




In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation.

camera coordinate system

world coordinate system

In non-homogeneous coordinates, the transformation from world to normalized camera coordinate system is given by:

$$\tilde{\boldsymbol{X}}_{\text{cam}} = \boldsymbol{R}(\tilde{\boldsymbol{X}} - \tilde{\boldsymbol{C}}) = \boldsymbol{R}\tilde{\boldsymbol{X}} + \boldsymbol{t}$$

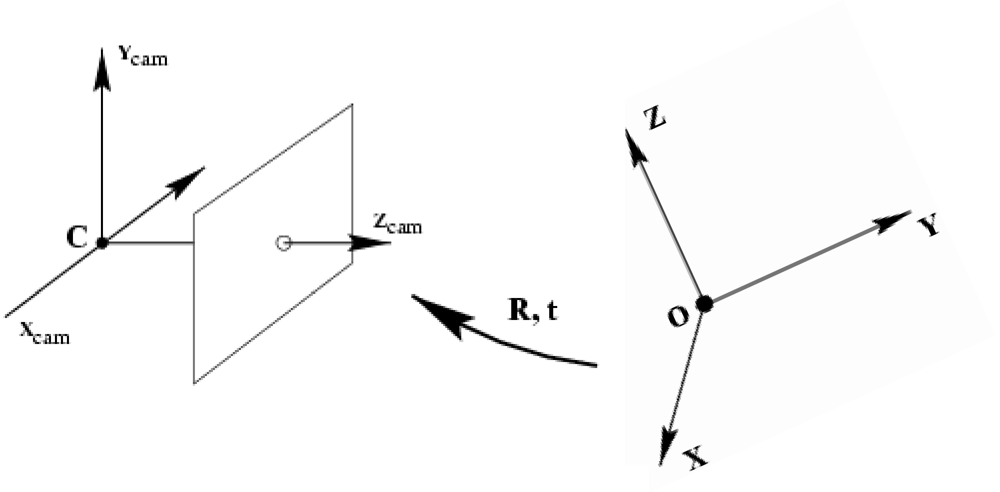
coords. of point in normalized camera frame

3x3 rotation matrix

coords. of a point in world frame

coords. of camera center in world frame

# Extrinsic Parameters




In *non-homogeneous* coordinates:

$$\widetilde{\boldsymbol{X}}_{\text{cam}} = \boldsymbol{R}\widetilde{\boldsymbol{X}} + \boldsymbol{t}$$

In *homogeneous* coordinates:

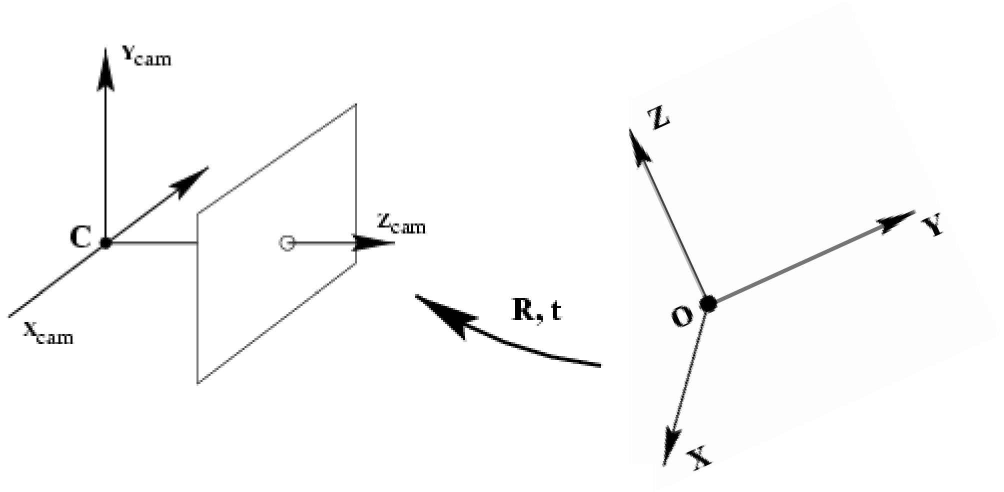
3D transformation matrix (4 x 4)

$$\boldsymbol{X}_{\text{cam}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \boldsymbol{X}$$

Transformation from normalized 3D coordinates to pixel image coordinates:

$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\boldsymbol{X}_{\text{cam}}$$

# Extrinsic Parameters




$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}\boldsymbol{X}$$

Simplifying:

$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X} \qquad \boldsymbol{t} = -\boldsymbol{R}\widetilde{\boldsymbol{C}}$$

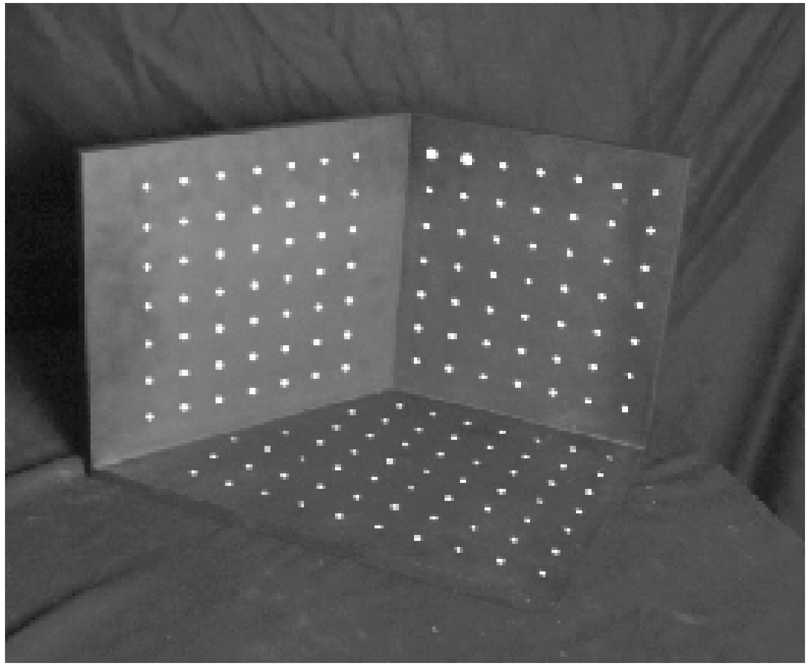
# Camera calibration

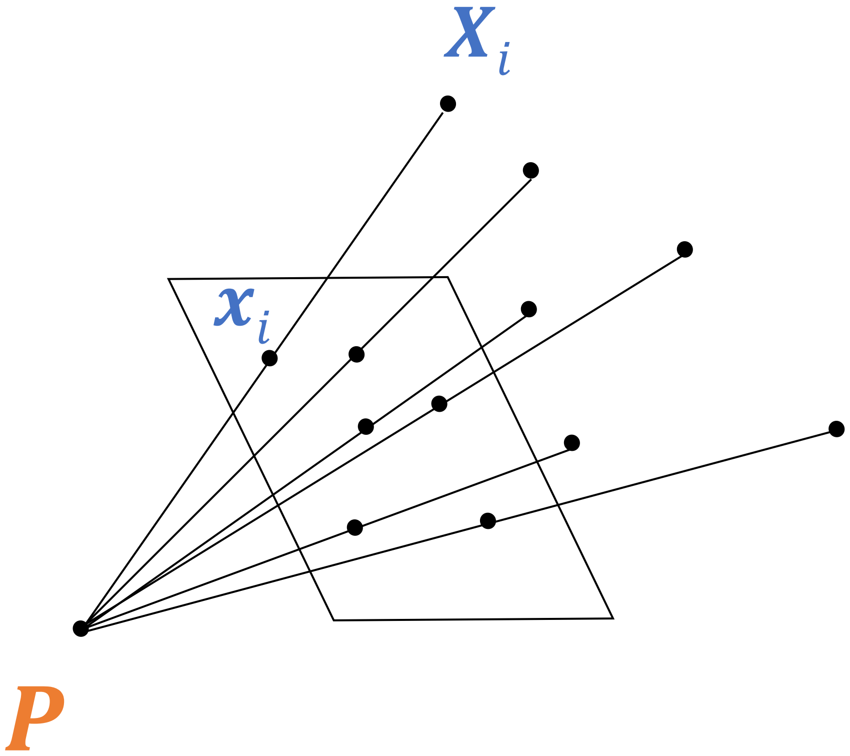
$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R} \;\; \boldsymbol{t}]\boldsymbol{X}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

# Camera calibration

Given $n$ points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters





# Camera calibration

Given $n$ points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters

Image credit: J. Hays

# Camera calibration: Linear method

$$\boldsymbol{x}_i \cong \boldsymbol{P}\boldsymbol{X}_i \qquad \boldsymbol{P}_j = \begin{pmatrix} p_{j1} \\ p_{j2} \\ p_{j3} \\ p_{j4} \end{pmatrix} \qquad \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \cong \begin{pmatrix} \boldsymbol{X}_i^T\boldsymbol{P}_1 \\ \boldsymbol{X}_i^T\boldsymbol{P}_2 \\ \boldsymbol{X}_i^T\boldsymbol{P}_3 \end{pmatrix} \qquad \begin{matrix} \boldsymbol{X}_i^T\boldsymbol{P}_1 - x_i\boldsymbol{X}_i^T\boldsymbol{P}_3 = 0 \\ \boldsymbol{X}_i^T\boldsymbol{P}_2 - y_i\boldsymbol{X}_i^T\boldsymbol{P}_3 = 0 \end{matrix}$$

$$\begin{bmatrix} X_i^T & 0 & -x_iX_i^T \\ 0 & X_i^T & -y_iX_i^T \end{bmatrix} \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{pmatrix} = 0$$

One match gives two linearly independent constraints

# Camera calibration: Linear method

Final linear system:

$$\begin{bmatrix} \mathbf{0}^T & \boldsymbol{X}_1^T & -y_1\boldsymbol{X}_1^T \\ \boldsymbol{X}_1^T & \mathbf{0}^T & -x_1\boldsymbol{X}_1^T \\ \cdots & \cdots & \cdots \\ \mathbf{0}^T & \boldsymbol{X}_n^T & -y_n\boldsymbol{X}_n^T \\ \boldsymbol{X}_n^T & \mathbf{0}^T & -x_n\boldsymbol{X}_n^T \end{bmatrix} \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{pmatrix} = 0 \qquad \boldsymbol{A}\boldsymbol{p} = 0$$

- One 2D/3D correspondence gives two linearly independent equations
  - The projection matrix has 11 degrees of freedom
  - 6 correspondences needed for a minimal solution
- Homogeneous least squares: find $\boldsymbol{p}$ minimizing $\|\boldsymbol{A}\boldsymbol{p}\|^2$
  - Solution is eigenvector of $\boldsymbol{A}^T\boldsymbol{A}$ corresponding to smallest eigenvalue

# Camera calibration: Linear vs. nonlinear

Linear calibration is easy to formulate and solve, but it doesn't directly tell us the camera parameters

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

vs.

$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X}$$

# Camera calibration: Linear vs. nonlinear

In practice, *non-linear* methods are preferred

- Write down objective function in terms of intrinsic and extrinsic parameters, as sum of squared distances between measured 2D points and estimated projections of 3D points:

$$\sum_i \|\mathrm{proj}(\boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X}_i) - \boldsymbol{x}_i\|_2^2$$

- Can include radial distortion and constraints such as known focal length, orthogonality, visibility of points
- Minimize error using non-linear optimization package
- Can initialize solution with output of linear method (perform QR decomposition to get $\boldsymbol{K}$ and $\boldsymbol{R}$ from $\boldsymbol{P}$)