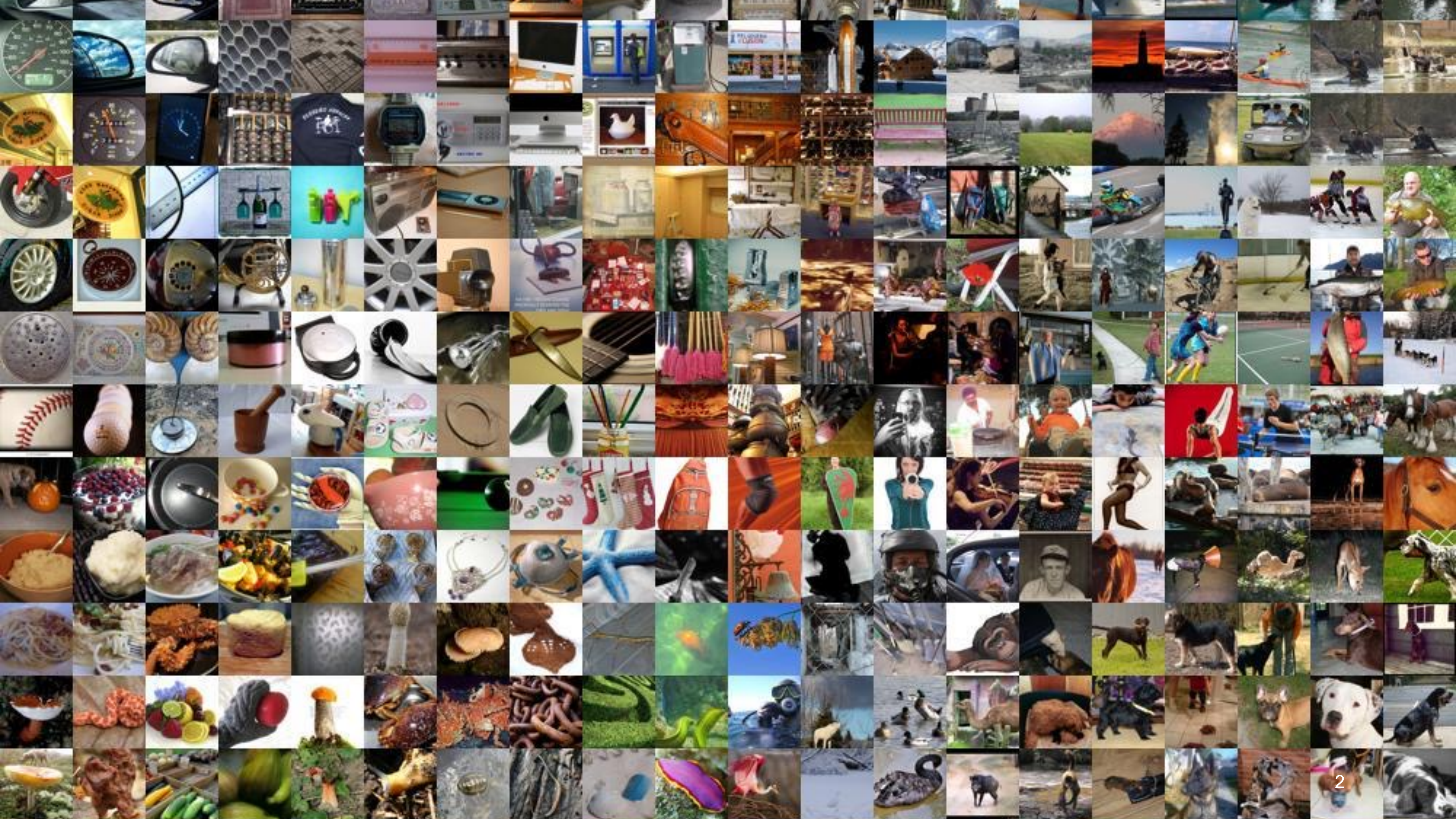
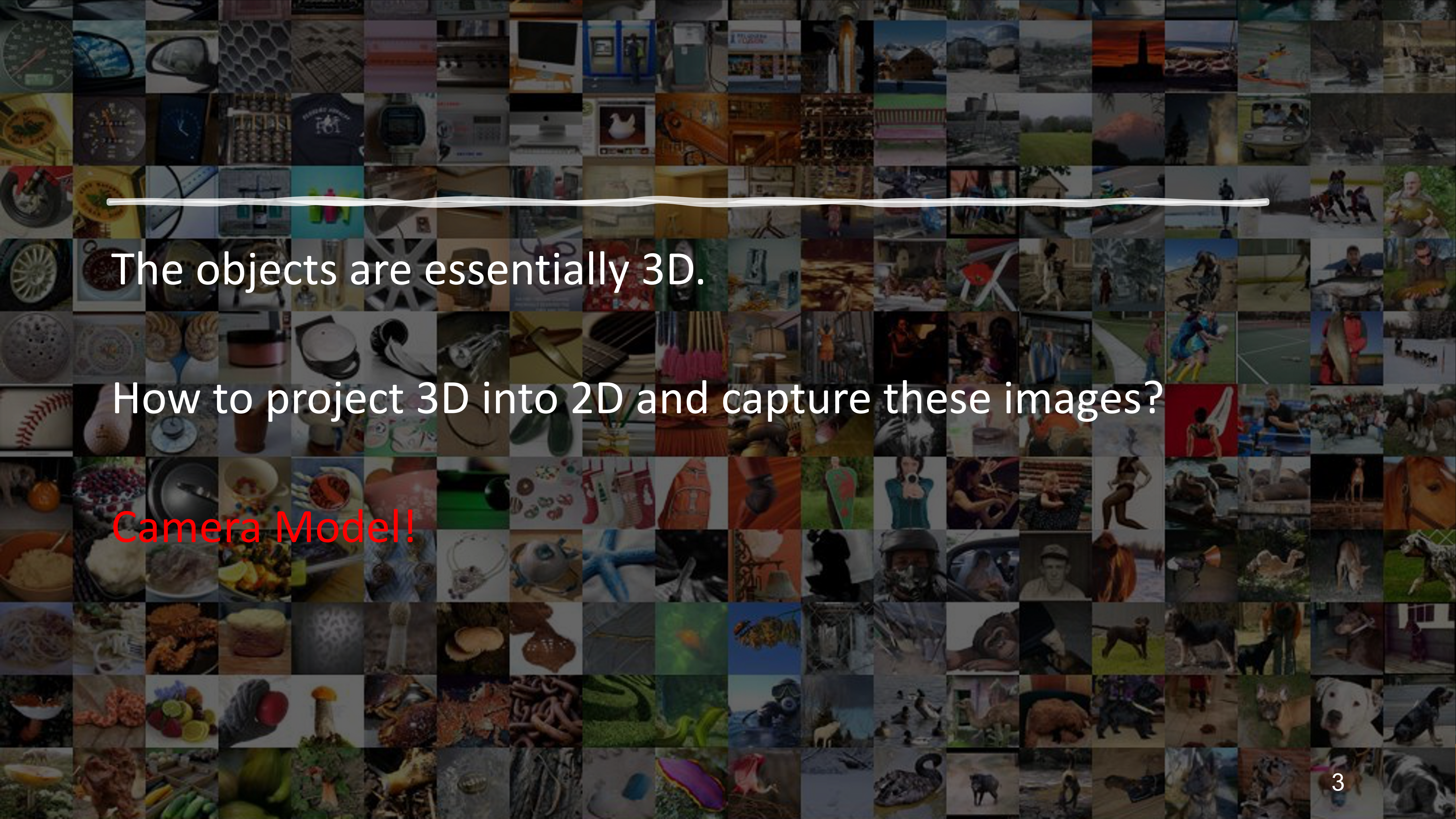
THE UNIVERSITY OF TEXAS AT DALLAS

Image Formulation: Camera Models

CS 4391 Computer Vision

Professor Yapeng Tian
Department of Computer Science



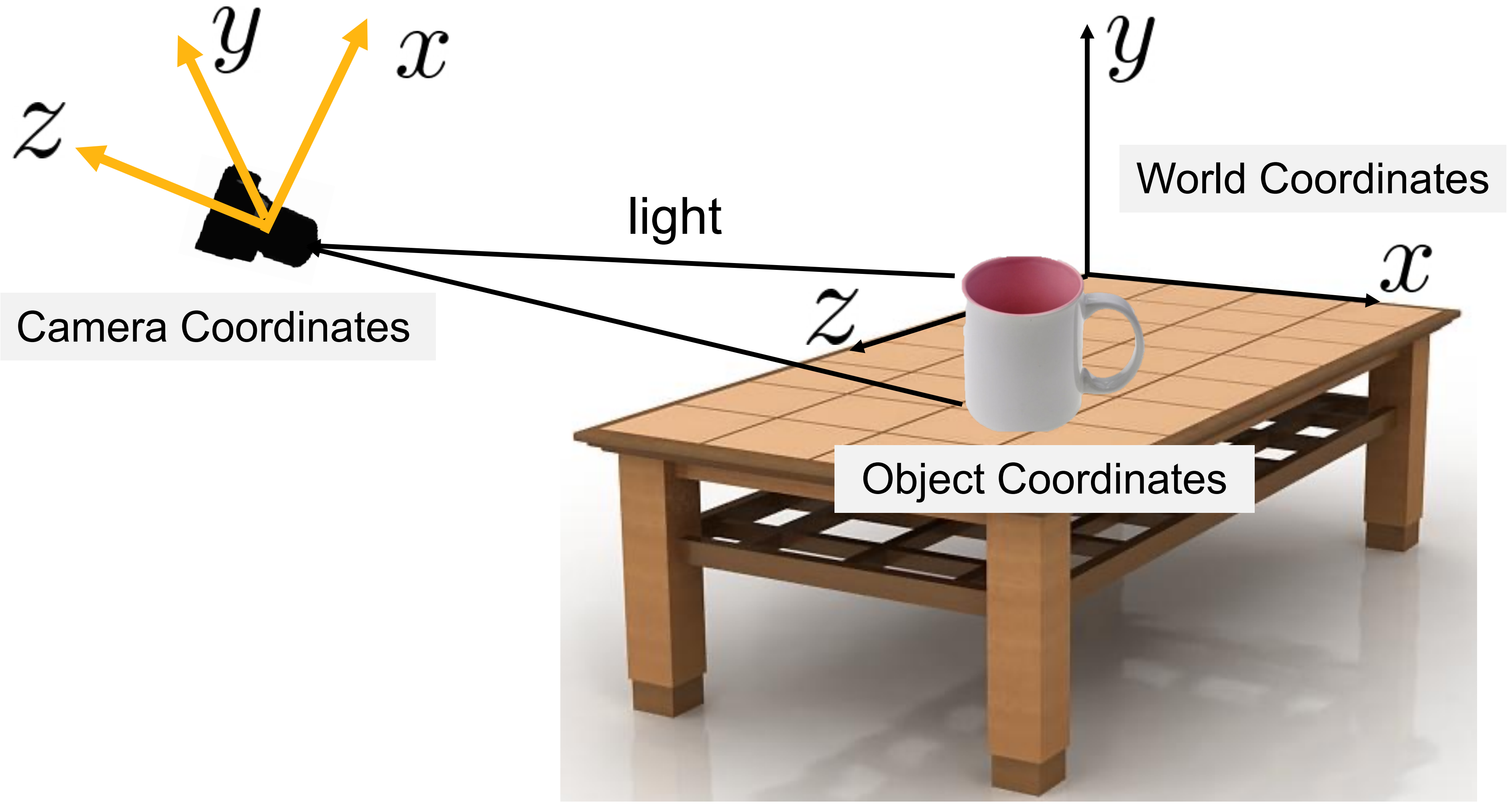


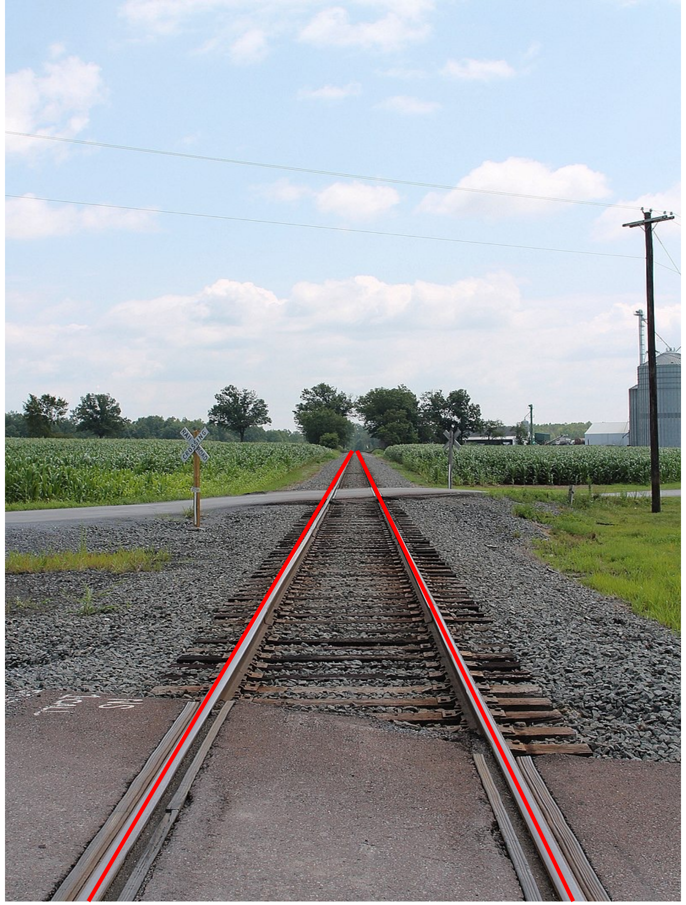
The objects are essentially 3D.

How to project 3D into 2D and capture these images?

Camera Model!

Camera Models: 3D-to-2D Projection





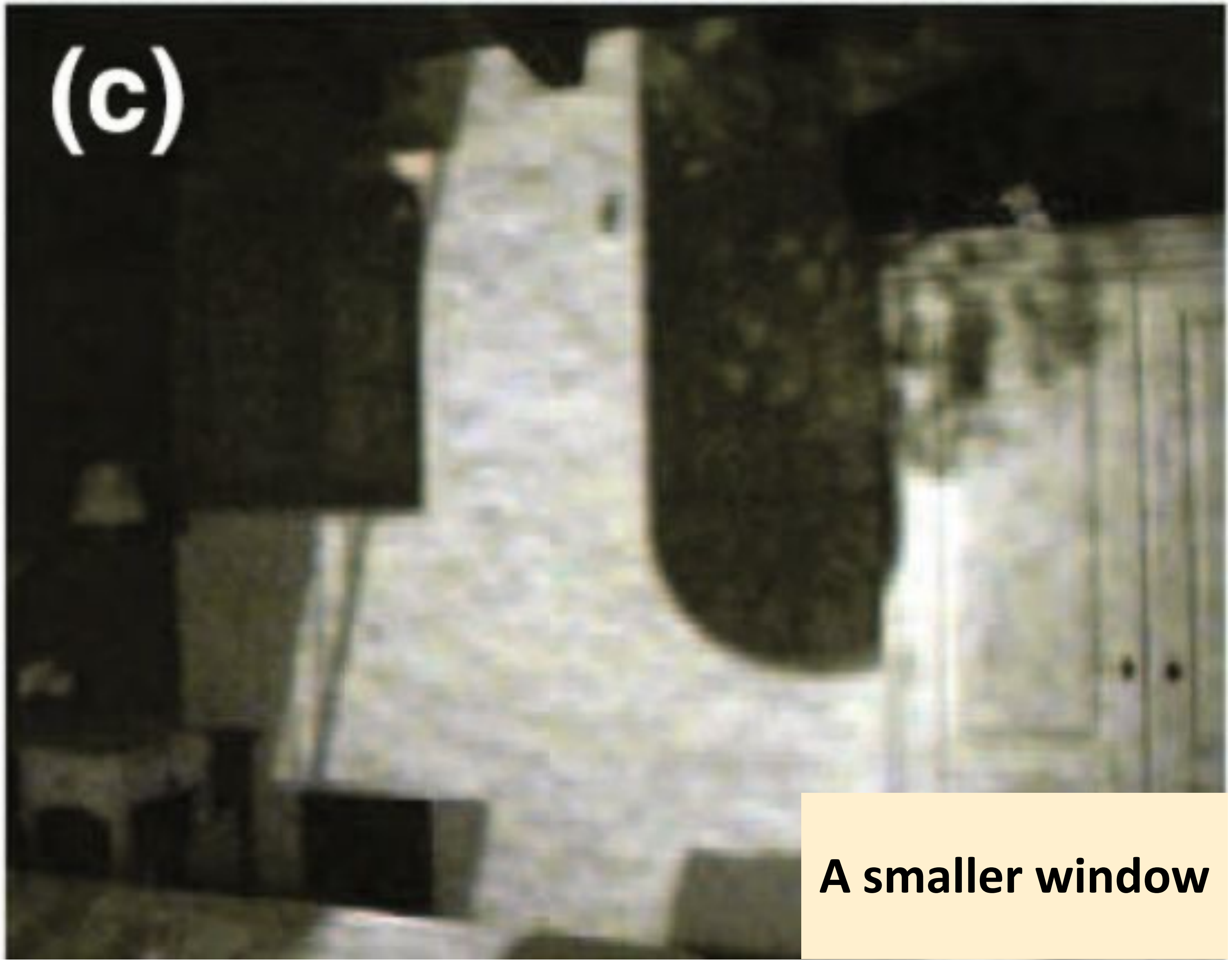
Q1: Are three balls in a same size?

Q2: Are the two rail lines parallel?

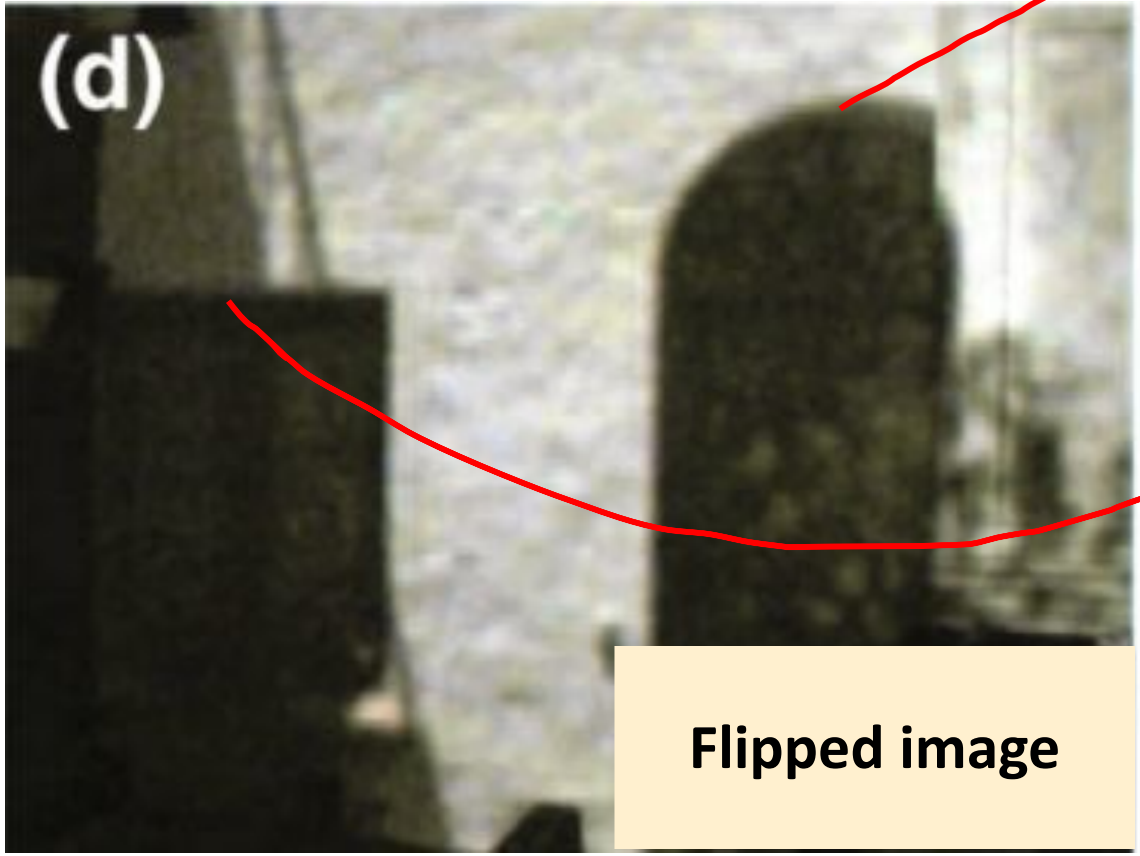
A1&A2: No?

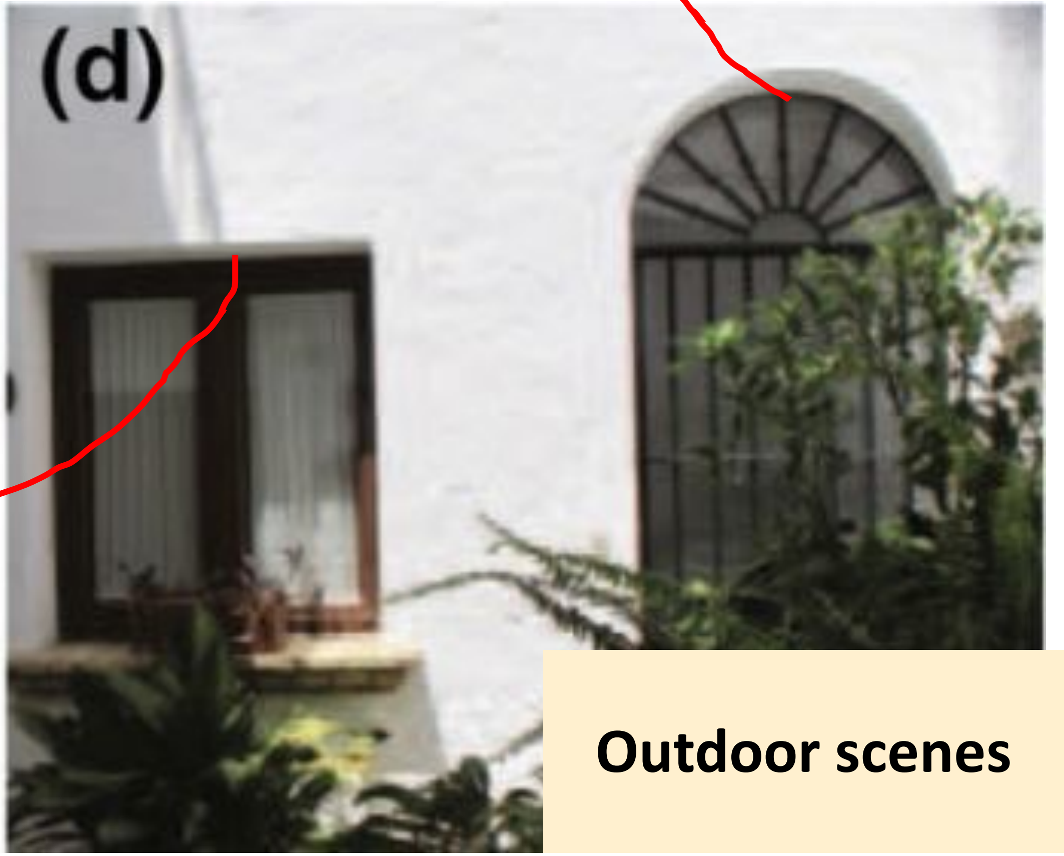
A Living Room
What objects or scenes?

Largely opened window

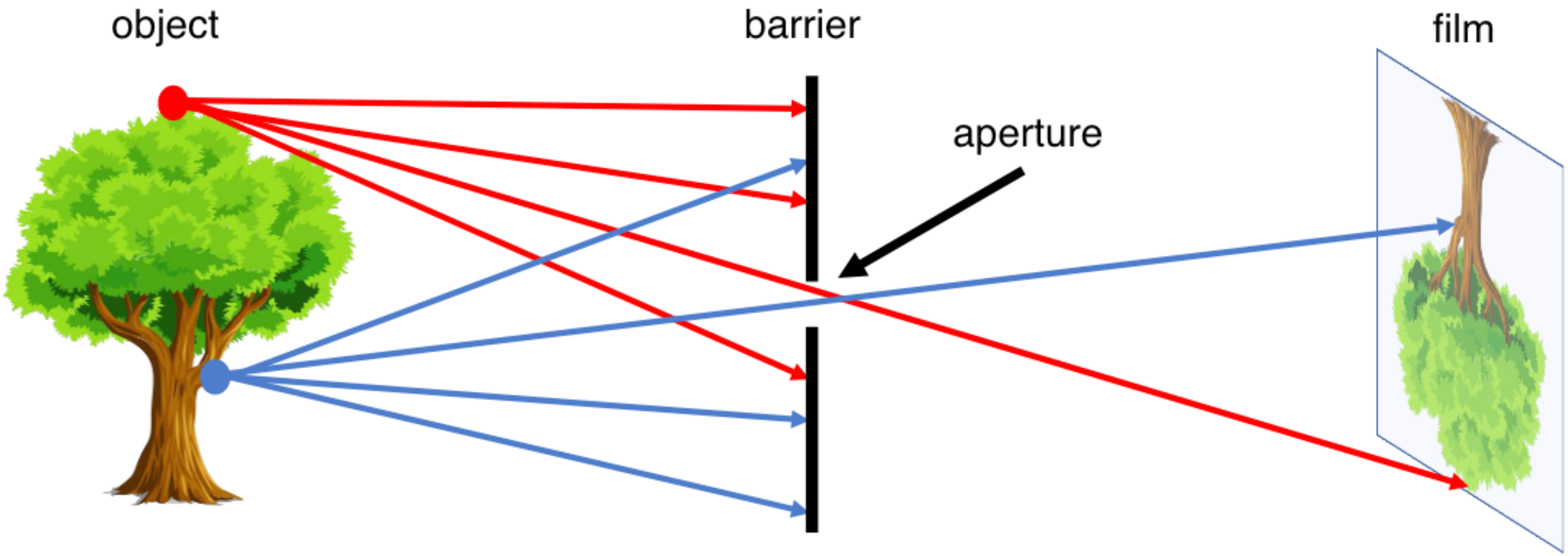

(c)
A smaller window

Nature Example of Pinhole Camera

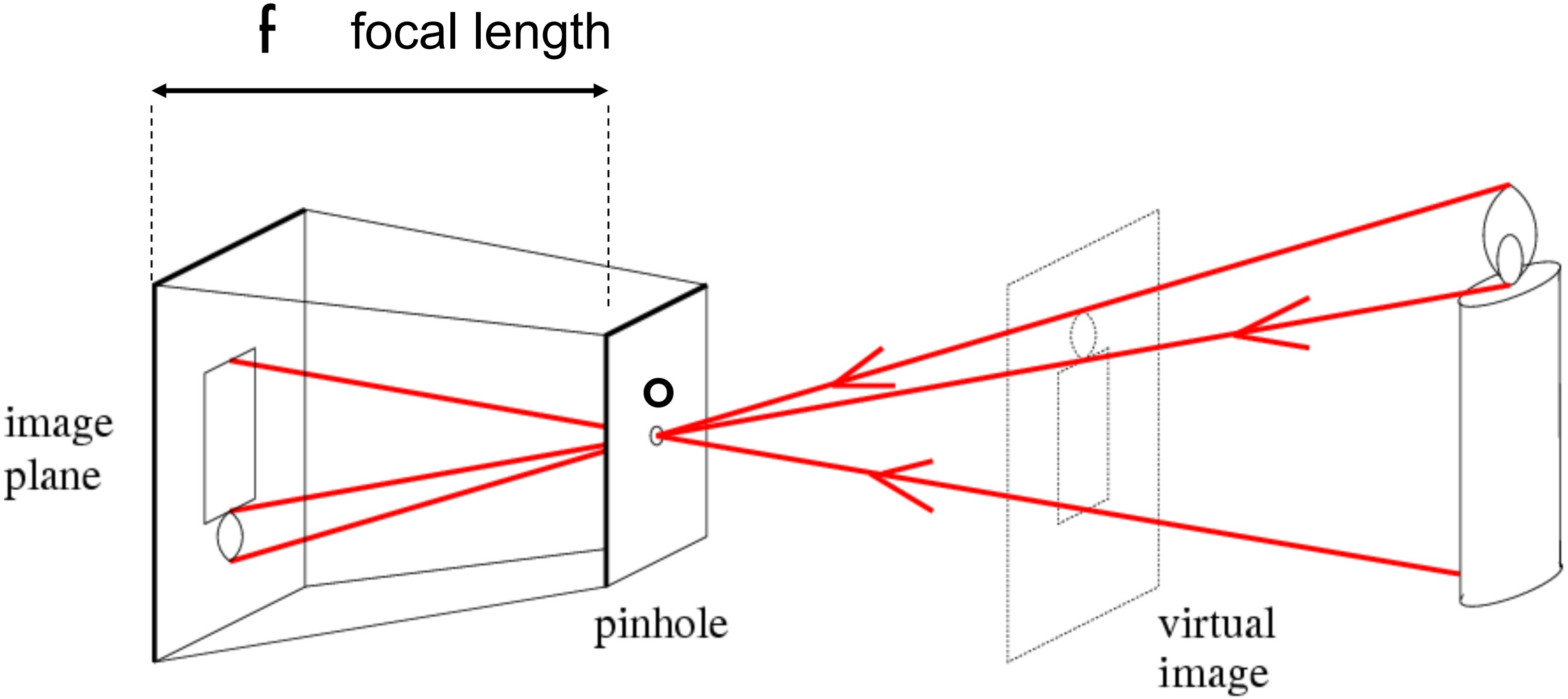




Pinhole Camera



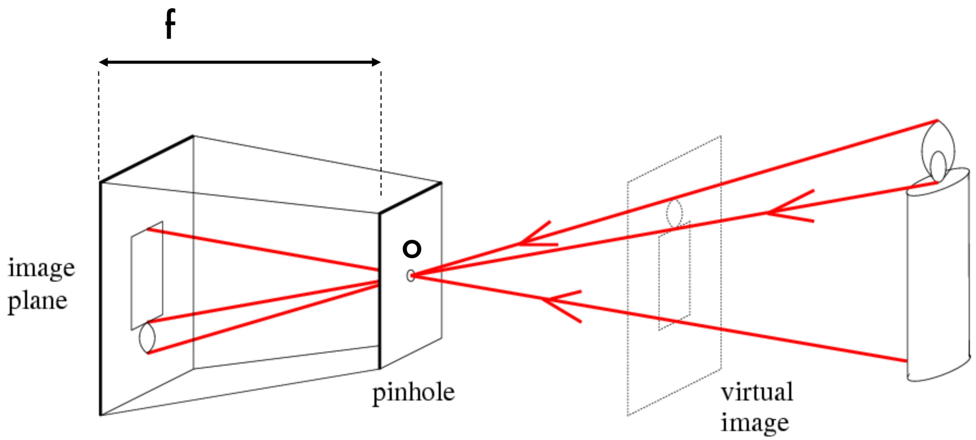
Pinhole Camera



Rotate the image plane by 180°

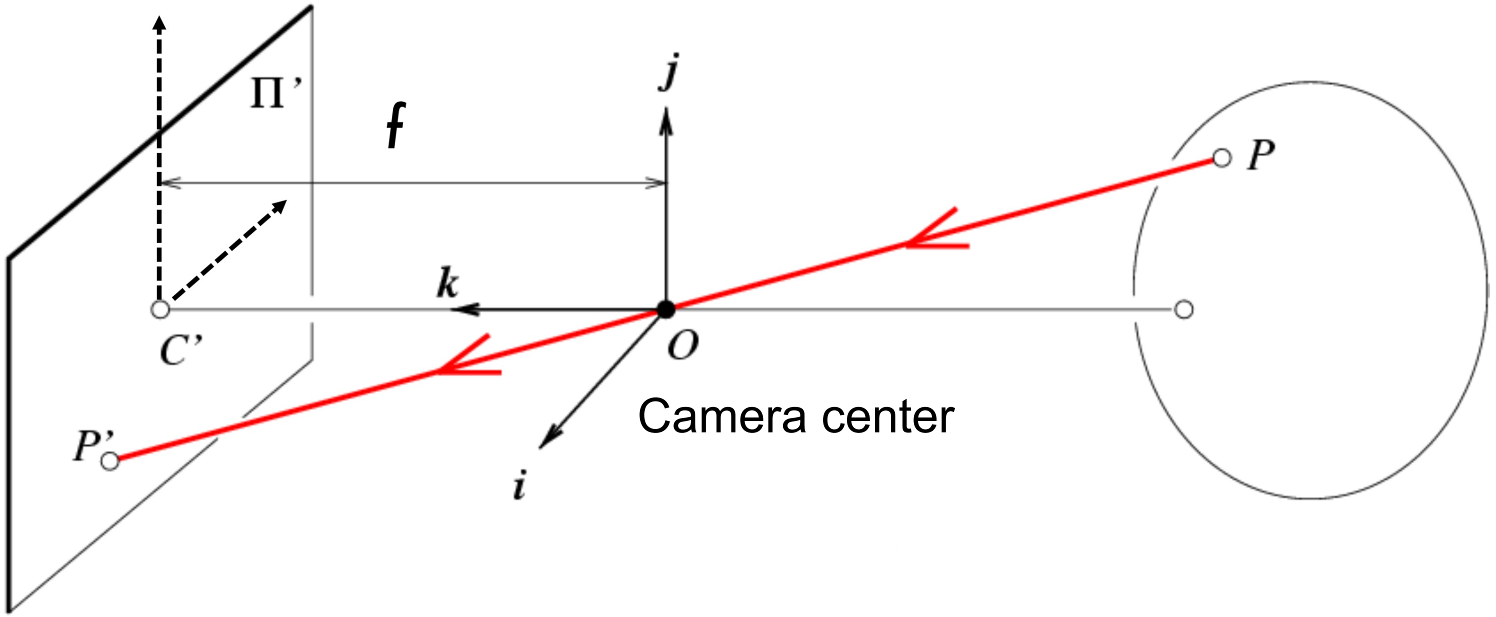
Cannot be implemented in practice
Useful for theoretic analysis

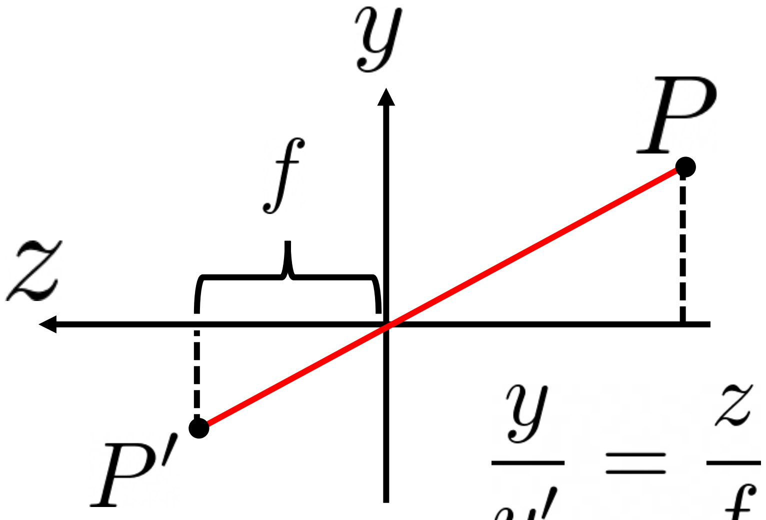
Natural Pinhole Cameras



Object: the sun
Pinhole: gaps between the leaves
Image plane: the ground

Central Projection in Camera Coordinates





$$\frac{y}{y'} = \frac{z}{f}$$

Camera coordinates

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \xrightarrow{\text{Nonlinear}} P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{cases} x' = f\,\dfrac{x}{z} \\ y' = f\,\dfrac{y}{z} \end{cases}$$

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

Conversion

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

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

Central Projection with Homogeneous Coordinates

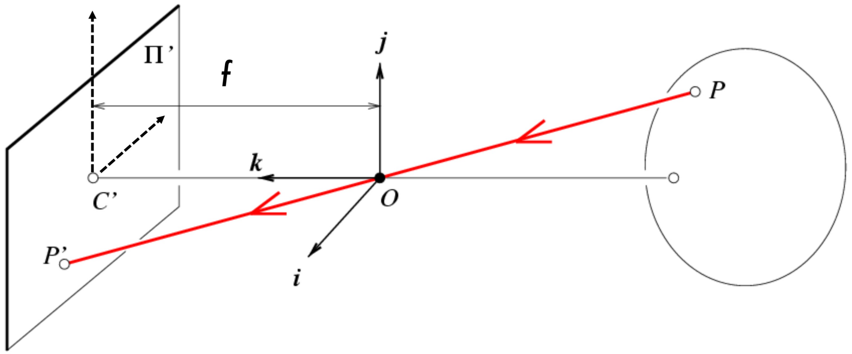
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \longrightarrow \begin{bmatrix} f\,\frac{x}{z} \\ f\,\frac{y}{z} \end{bmatrix}$$

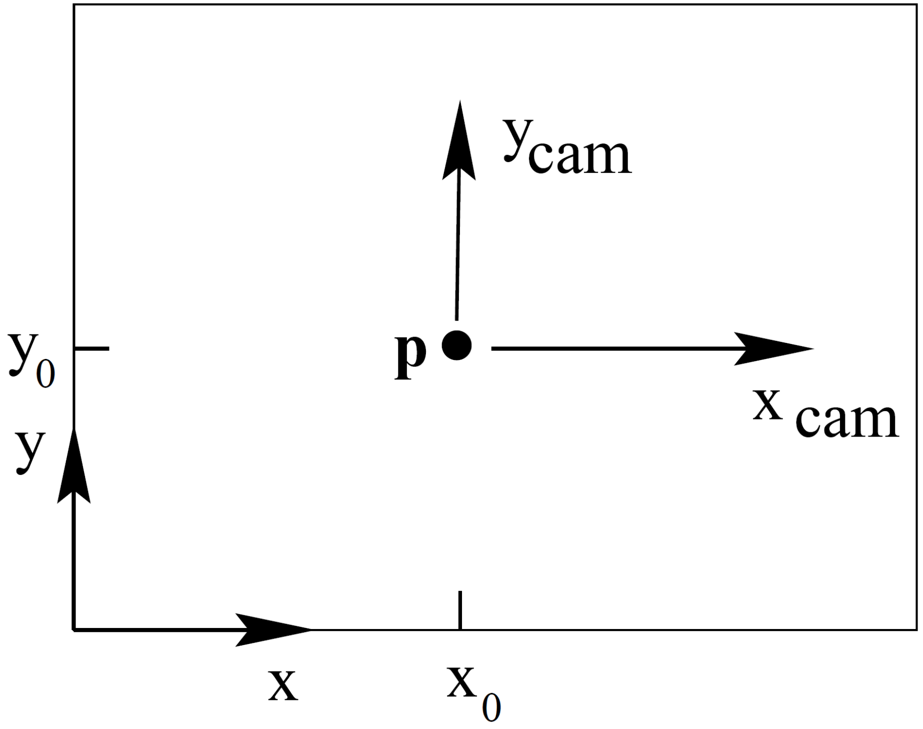
Central projection

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3x4 matrix

Principal Point Offset





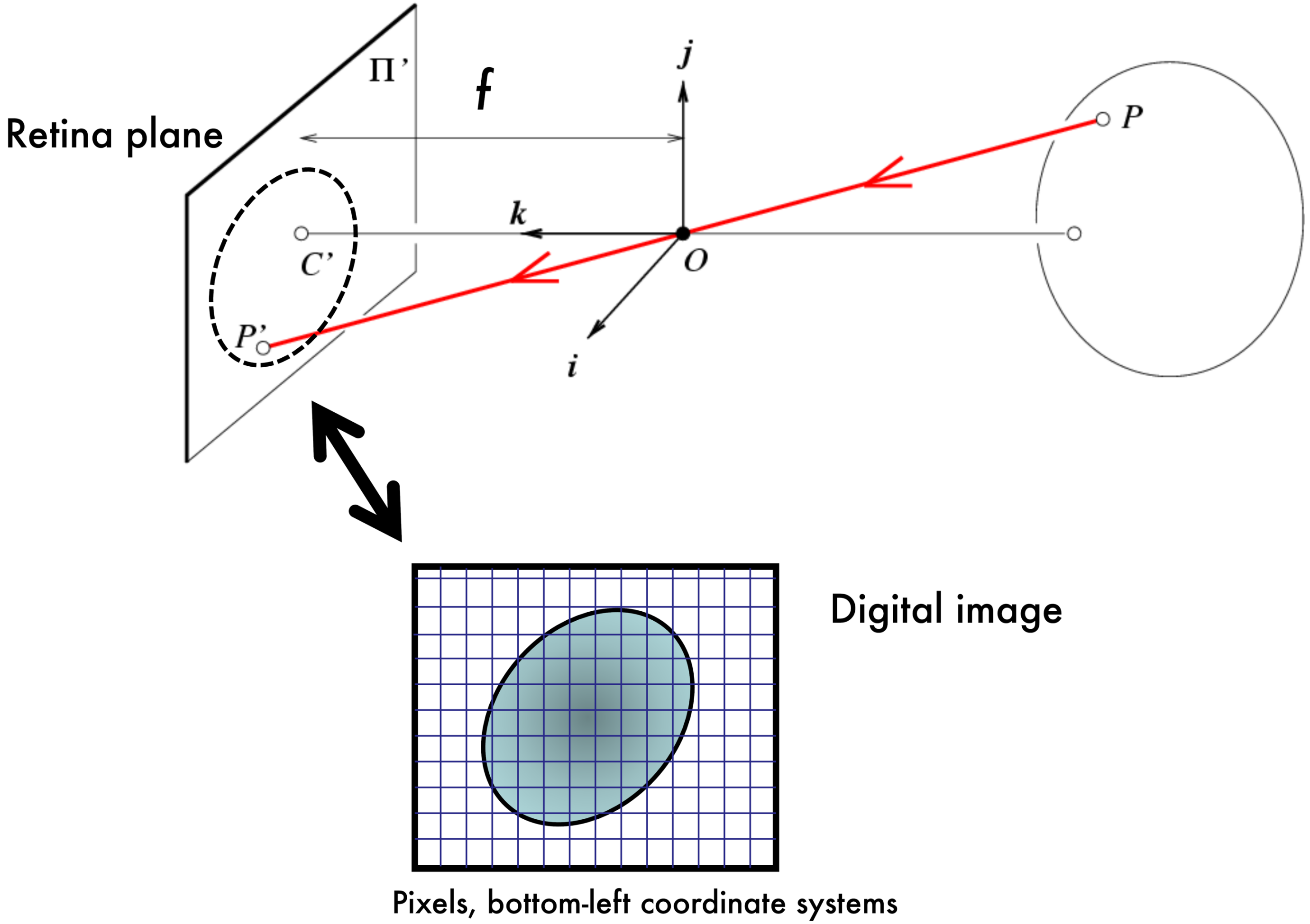
Principle point: projection of the camera center

Principal point $\mathbf{p} = (p_x, p_y)$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \longrightarrow \begin{bmatrix} f\frac{x}{z} + p_x \\ f\frac{y}{z} + p_y \end{bmatrix}$$

$$\begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

From Metric to Pixels



From Metric to Pixels

Metric space, i.e., meters

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

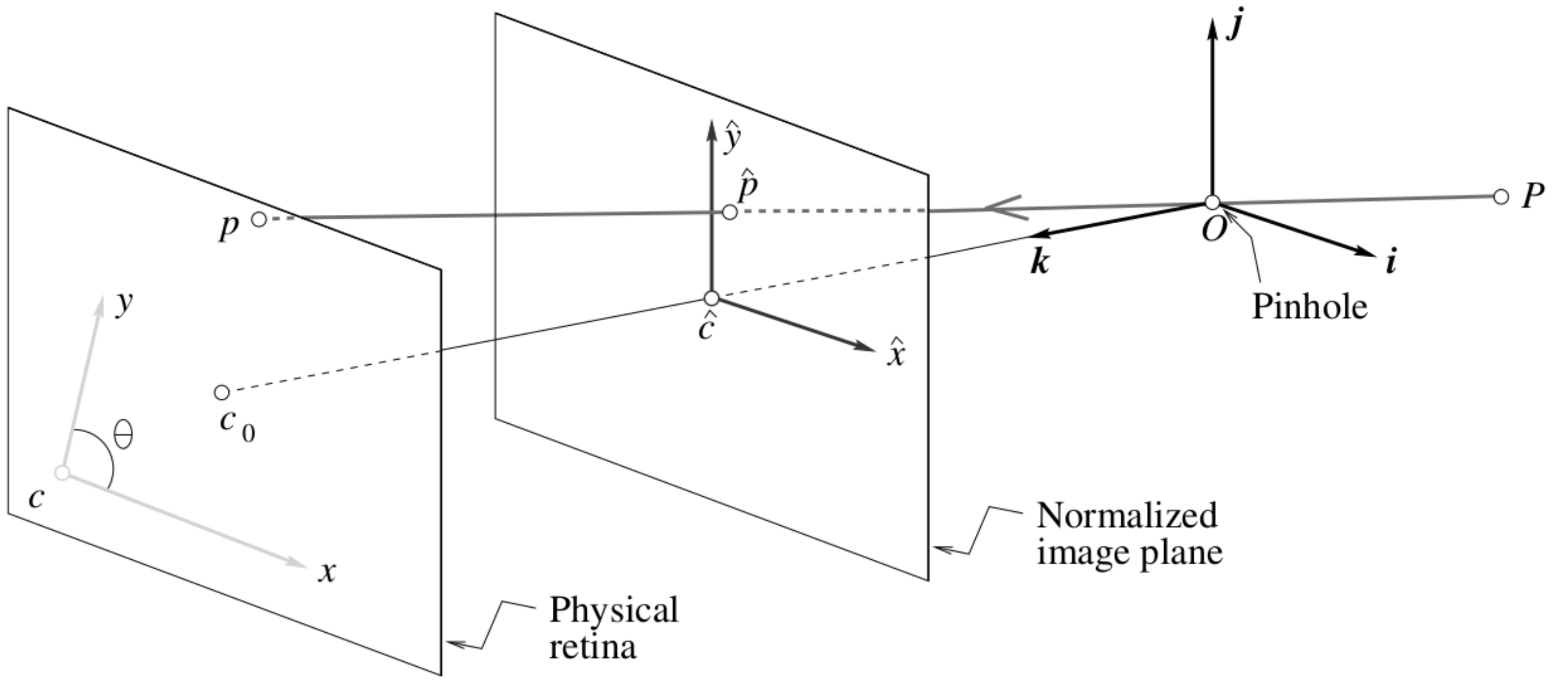
Pixel space

$$\begin{bmatrix} \alpha_x & & x_0 & 0 \\ & \alpha_y & y_0 & 0 \\ & & 1 & 0 \end{bmatrix}$$

$$\alpha_x = f m_x$$
$$\alpha_y = f m_y$$
$$x_0 = p_x m_x$$
$$y_0 = p_y m_y$$

m_x, m_y Number of pixel per unit distance

Axis Skew



The skew parameter will be zero for most normal cameras.

$$\begin{bmatrix} \alpha_x & & x_0 & 0 \\ & \alpha_y & y_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \alpha_x \frac{x}{z} + x_0 \\ \alpha_y \frac{y}{z} + y_0 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_x & -\alpha_x \cot(\theta) & x_0 & 0 \\ & \frac{\alpha_y}{\sin(\theta)} & y_0 & 0 \\ & & 1 & 0 \end{bmatrix}$$

https://blog.immenselyhappy.com/post/camera-axis-skew/

Camera Intrinsics

$$\begin{bmatrix} \alpha_x & -\alpha_x \cot(\theta) & x_0 & 0 \\ & \frac{\alpha_y}{\sin(\theta)} & y_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Camera intrinsics

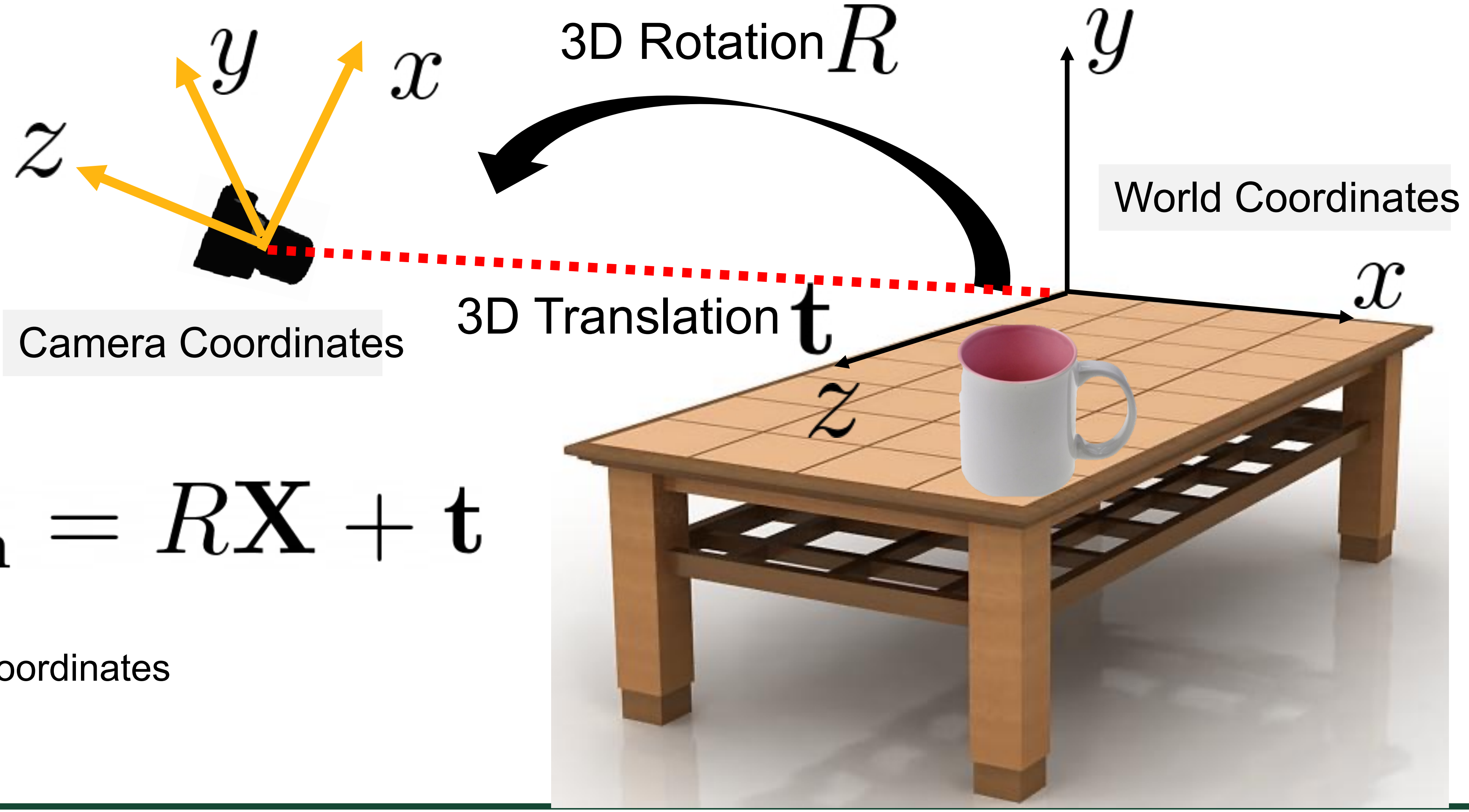
$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X}_{\mathrm{cam}}$$

3x1 3x3 3x4 4x1

Homogeneous coordinates

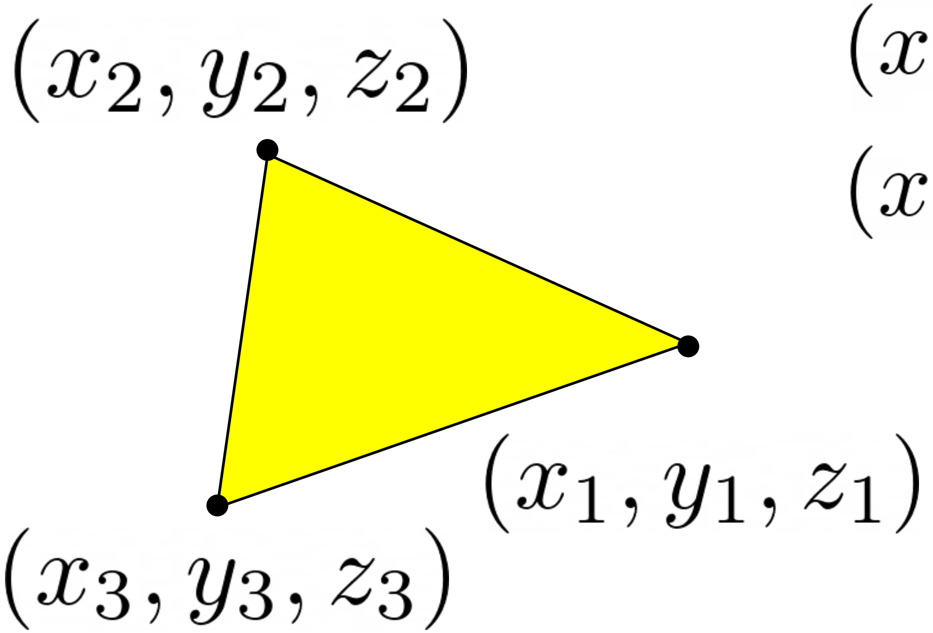
Camera Extrinsics: Camera Rotation and Translation



$$\mathbf{X}_{\mathrm{cam}} = R\mathbf{X} + \mathbf{t}$$

Euclidean coordinates

3D Translation



$$(x_1, y_1, z_1) \mapsto (x_1 + x_t, y_1 + y_t, z_1 + z_t)$$
$$(x_2, y_2, z_2) \mapsto (x_2 + x_t, y_2 + y_t, z_2 + z_t)$$
$$(x_3, y_3, z_3) \mapsto (x_3 + x_t, y_3 + y_t, z_3 + z_t)$$

$$\mathbf{v_1} \mapsto \mathbf{v_1} + \mathbf{t}$$
$$\mathbf{v_2} \mapsto \mathbf{v_2} + \mathbf{t}$$
$$\mathbf{v_3} \mapsto \mathbf{v_3} + \mathbf{t}$$

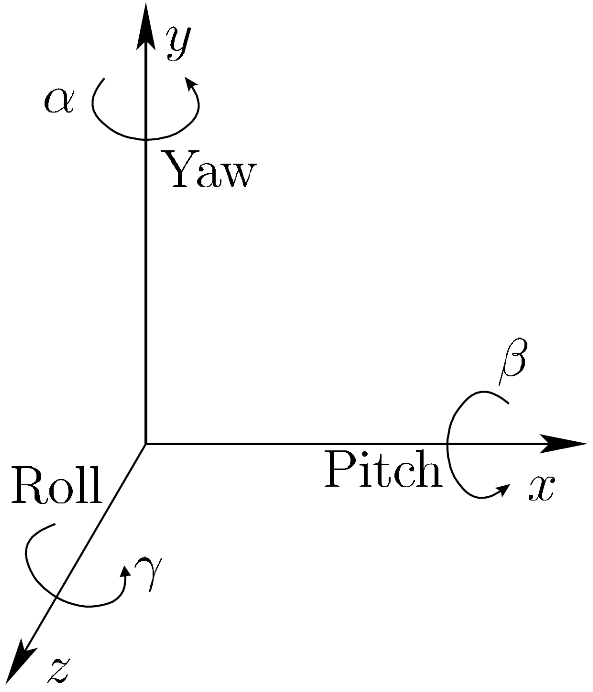
3D Translation $\mathbf{t} = (x_t, y_t, z_t)$

3D Rotation

The yaw, pitch, and roll rotations can be combined sequentially to attain any possible 3D rotation.

$$R(\alpha, \beta, \gamma) = R_y(\alpha) R_x(\beta) R_z(\gamma)$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_x(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \quad R_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$



Camera Projection Matrix $P = K[R|\mathbf{t}]$

Homogeneous coordinates

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X}_{\text{cam}}$$
$$= K[R|\mathbf{t}]\mathbf{X}$$

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

3x1 3x3 3x4 4x1

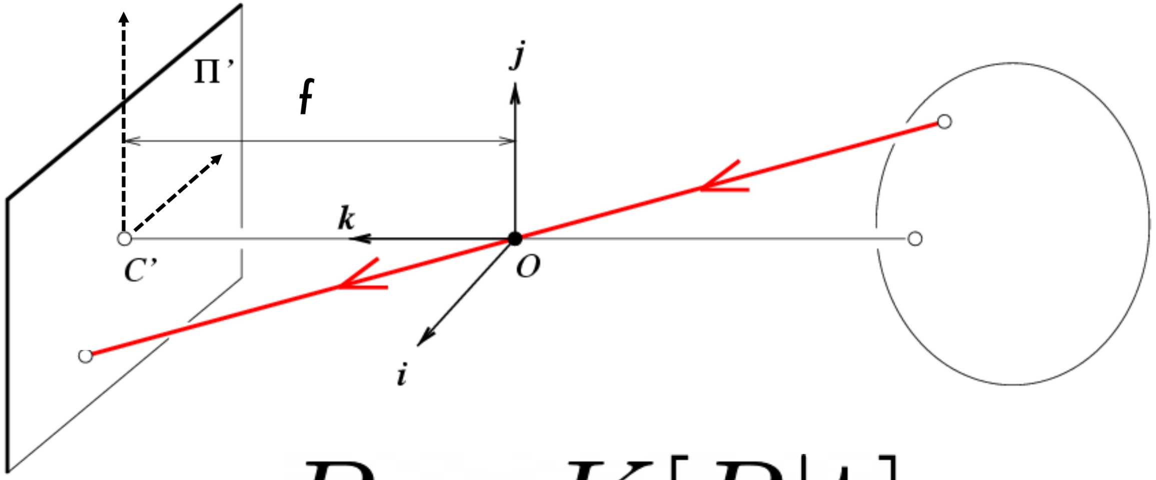
Image coordinates

Camera intrinsics

Camera extrinsics:
rotation and translation

World coordinates

Back-projection in World Coordinates



$$P = K[R|\mathbf{t}]$$

$$\mathbf{x} = P\mathbf{X}$$

- A pixel on the image backprojects to a ray in 3D

- The camera center O is on the ray
- $P^+\mathbf{x}$ is on the ray

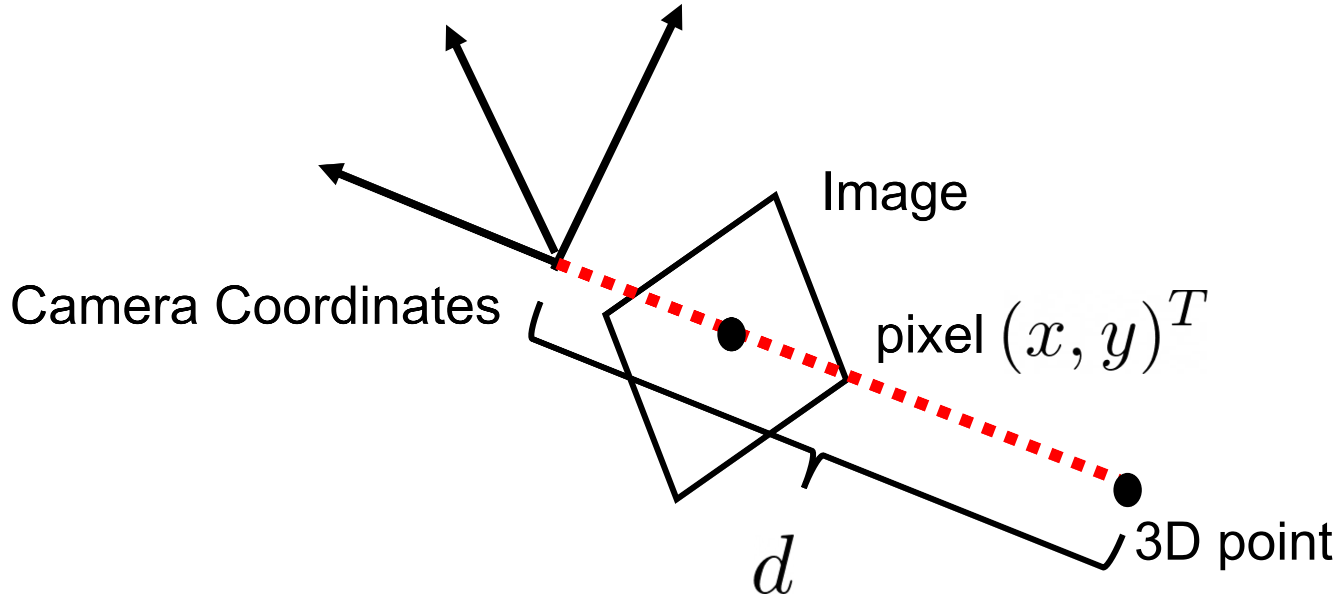
$$P^+ = P^T(PP^T)^{-1}$$

Pseudo-inverse

The ray can be written as

$$P^+\mathbf{x} + \lambda O$$

Back-projection in Camera Coordinates



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

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X}_{\mathrm{cam}}$$

$$K^{-1}\mathbf{x}$$

3D point with depth d : $dK^{-1}\mathbf{x}$

3D camera coordinates $\begin{bmatrix} d\frac{x-p_x}{f_x} \\ d\frac{y-p_y}{f_y} \\ d \end{bmatrix}$

Summary: Camera Models

Camera projection matrix: intrinsics and extrinsics

$$P = K[R|\mathbf{t}]$$

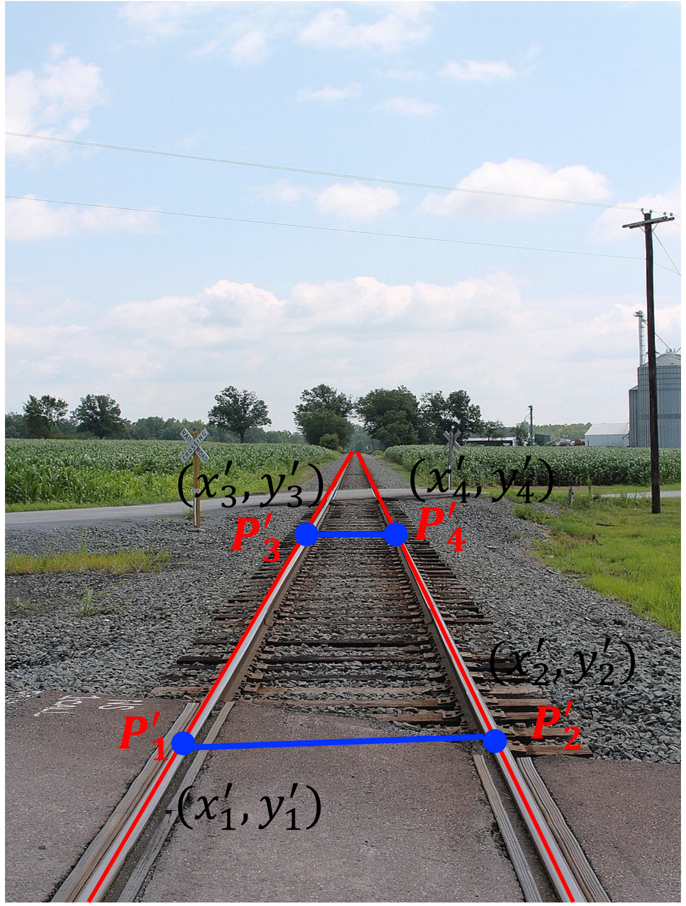
3x3

3x4

Camera intrinsics

Camera extrinsics:
rotation and translation

Interpreting Perceived Images



The lengths of two lines P_1P_2 and P_3P_4 in 3D space are equal

$$\overset{\text{3D}}{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \overset{\text{2D}}{P'} = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad \begin{cases} x' = f\,\dfrac{x}{z} \\ y' = f\,\dfrac{y}{z} \end{cases}$$

Why is $P'_3P'_4$ shorter than $P'_1P'_2$ in the 2D image?

- For the two 3D points P_1 and P_3, let's assume we have $x_1 = x_3, y_1 = y_3$, and $z_1 < z_3$ in the 3D coordinate system
- After 3D-to-2D projection, we have $x'_1 > x'_3$ and $y'_1 > y'_3$
- Larger depth and shorter length due to the projection

Further Reading

Stanford CS231A: Computer Vision, From 3D Reconstruction to Recognition, Course Notes 1: Camera Models

Multiview Geometry in Computer Vision, Richard Hartley and Andrew Zisserman, Chapter 6, Camera Models

Computer Vision: Algorithms and Applications. Richard Szeliski, Chapter 2.1.4, 3D to 2D projections