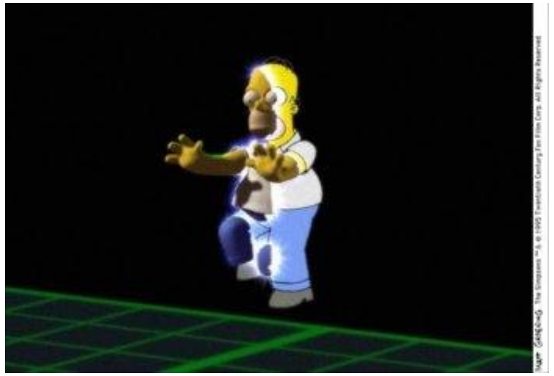
Lecture 15: Camera Models -1

Instructor: Roni Sengupta
ULA: Andrea Dunn Beltran, William Li,
Liujie Zheng

Course Website:
Scan Me!

Breaking out of 2D

…now we are ready to break out of 2D


And enter the real world!

NeRF in the wild (will get to it towards the end!)

NeRF in the Wild, Martin-Brualla, Radwan et al. CVPR 2021

Lectures in 3D Vision

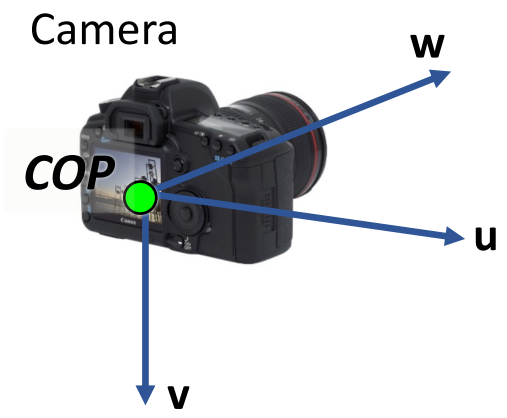
- Fundamental Concepts (4 lectures)
  - Modeling camera and 3D->2D projection (2 lectures)
  - 2-view geometry & Stereo Vision (3 lectures)

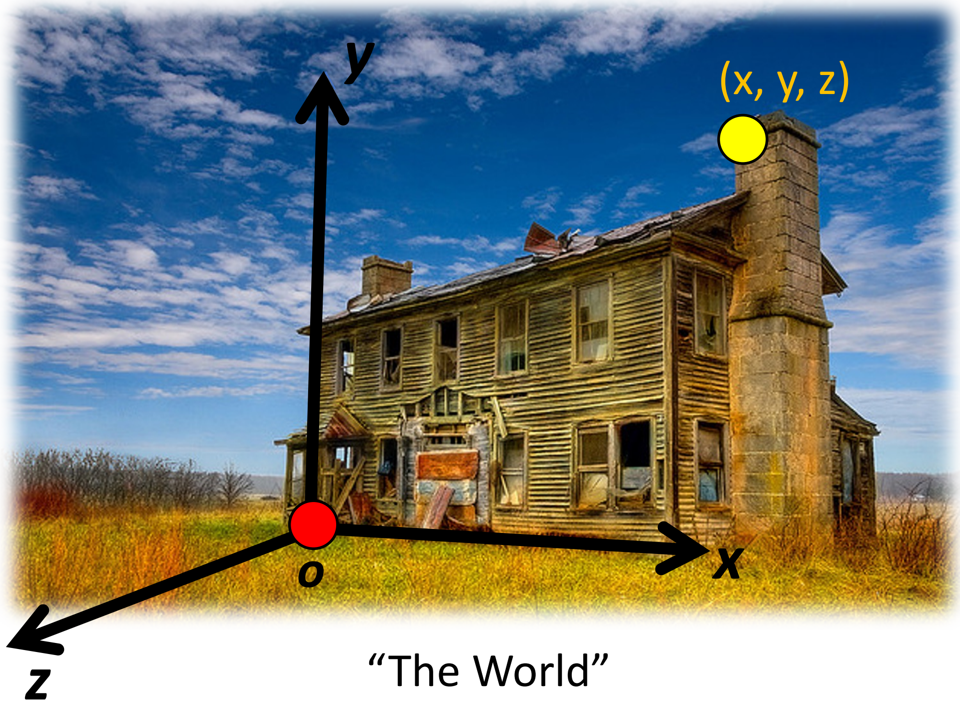
- 3D Reconstruction techniques (3 lectures)
  - Multiview Stereo (MVS)
  - Structure from Motion (SfM) + SLAM
  - Photometric Stereo (PS)

- Deep Learning + 3D Vision (2 lectures)
  - Deep Learning + MVS, SfM, PS
  - Neural Radiance Fields (NeRFs)

Camera parameters

- How can we model the geometry of a camera?





"The World"

Three important coordinate systems:

1. *World* coordinates
2. *Camera* coordinates
3. *Image* coordinates

How do we project a given world point (x, y, z) to an image point?

Coordinate frames

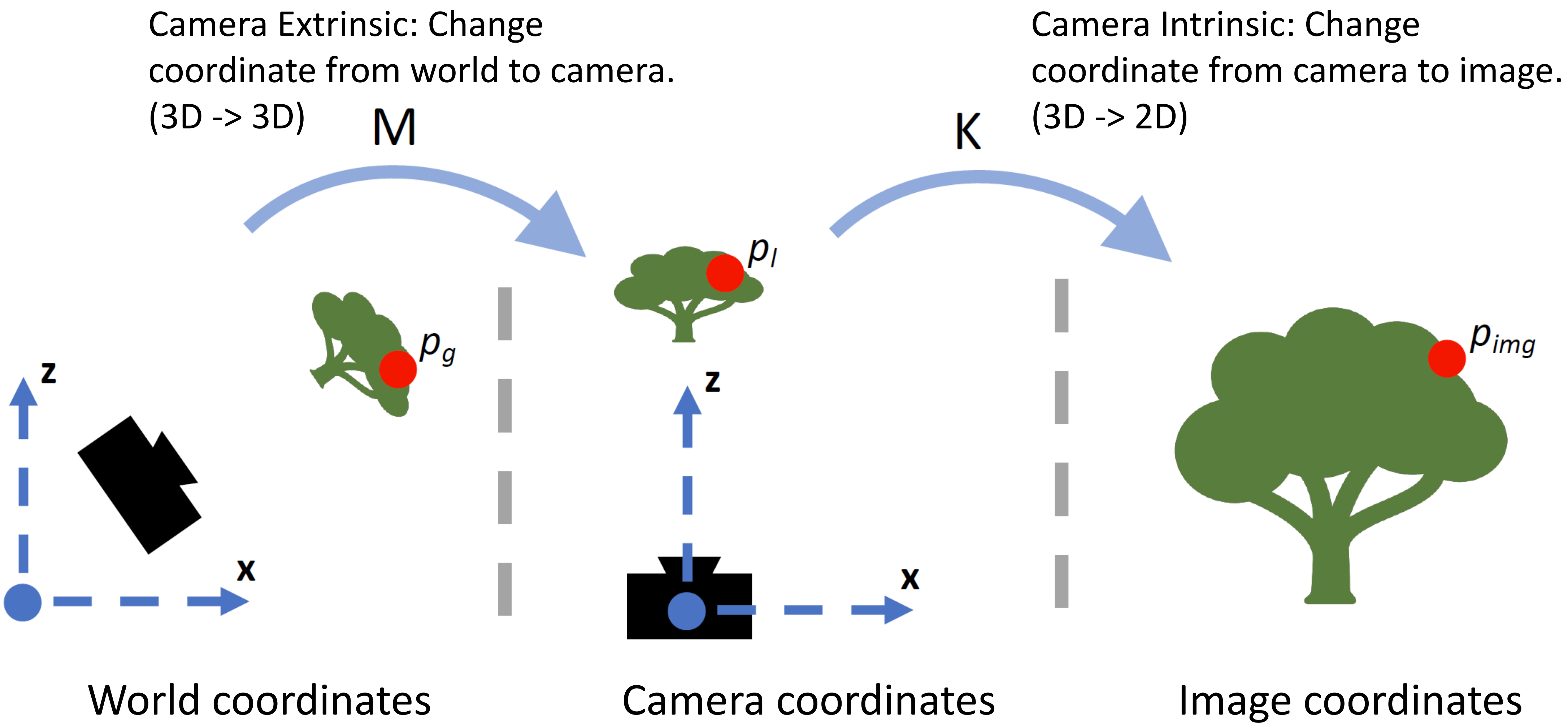


Figure credit: Peter Hedman

A camera is a mapping between the 3D world and a 2D image

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

2D image point

camera matrix

3D world point

$$\boldsymbol{x} = \mathbf{PX}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image 3 x 1	Camera matrix 3 x 4	homogeneous world point 4 x 1

Camera parameters

To project a point (*x*, *y*, *z*) in *world* coordinates into a camera

- First transform (*x*, *y*, *z*) into *camera* coordinates
- Need to know
  - Camera position (in world coordinates)
  - Camera orientation (in world coordinates)
  - Together they form *Camera Extrinsics*
- Then project into the image plane to get *image (pixel) coordinates*
  - Need to know *Camera Intrinsics*

Today’s Class

- Camera Extrinsic
- Camera Intrinsics and pinhole camera model
- Perspective Distortion
- Other Projection models
- How to calibrate camera, i.e. estimate camera parameters (next class)

Today's Class

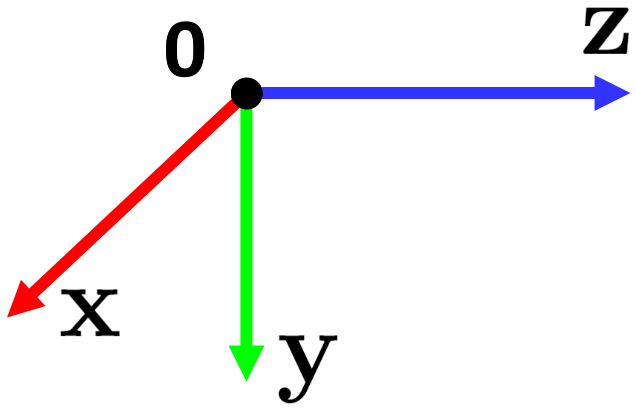
- Camera Extrinsic
- Camera Intrinsics and pinhole camera model
- Perspective Distortion
- Other Projection models
- How to calibrate camera, i.e. estimate camera parameters (next class)

Extrinsics

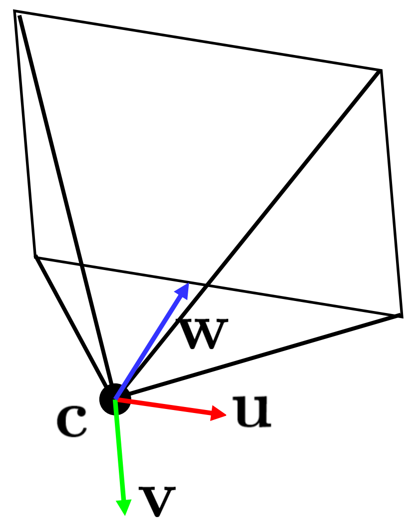
- How do we get the camera to "canonical form"?
  - Canonical form: Center of projection at the origin, x-axis points right, y-axis points down, z-axis points forwards

$\mathbf{X_w}$ -> location of a point in world coordinate.

$\mathbf{X_c}$ -> location of a point in camera coordinate.

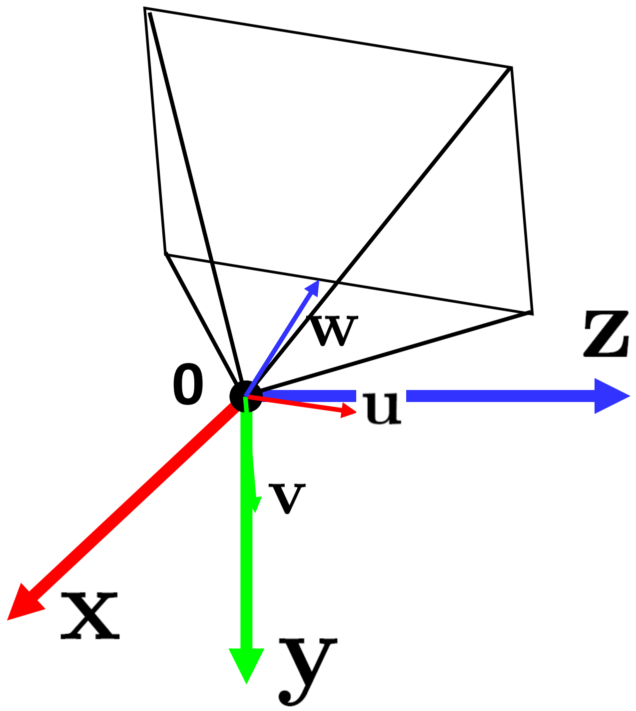


Step 1: Translate by -**c**



Extrinsics

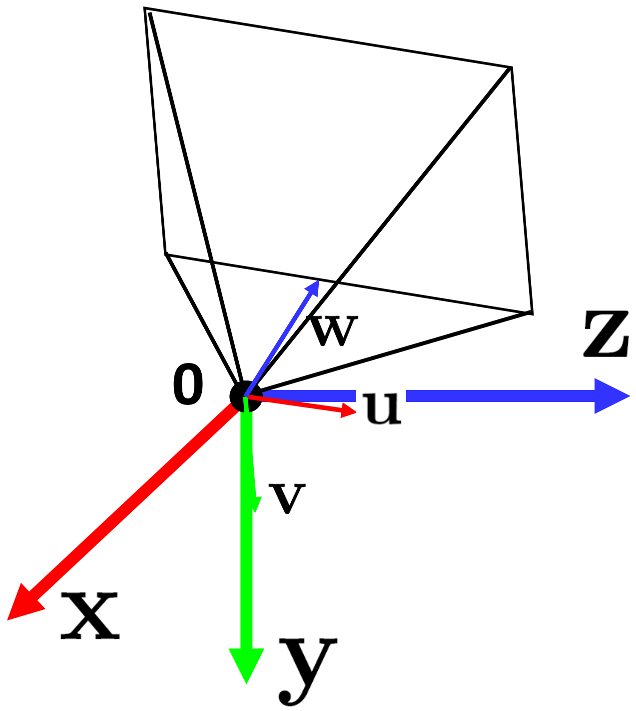
- How do we get the camera to “canonical form”?
  - Canonical form: Center of projection at the origin, x-axis points right, y-axis points down, z-axis points forwards



Step 1: Translate by -**c**

Extrinsics

- How do we get the camera to "canonical form"?
  - Canonical form: Center of projection at the origin, x-axis points right, y-axis points down, z-axis points forwards

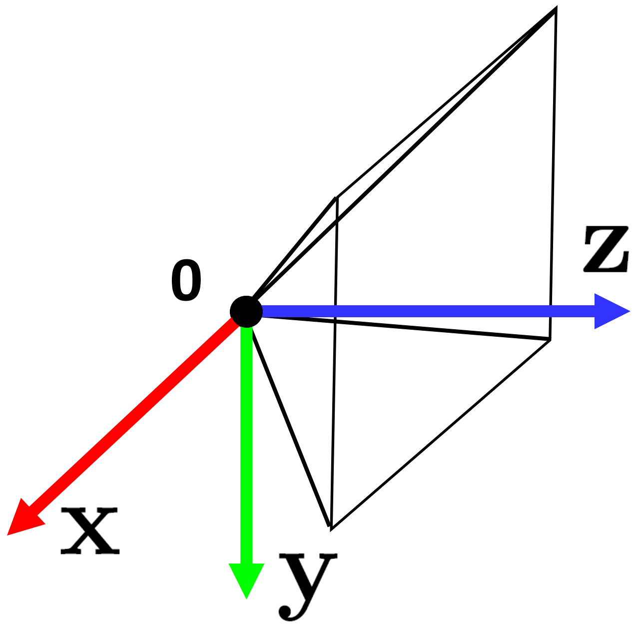


Step 1: Translate by -**c**

Step 2: Rotate by **R**

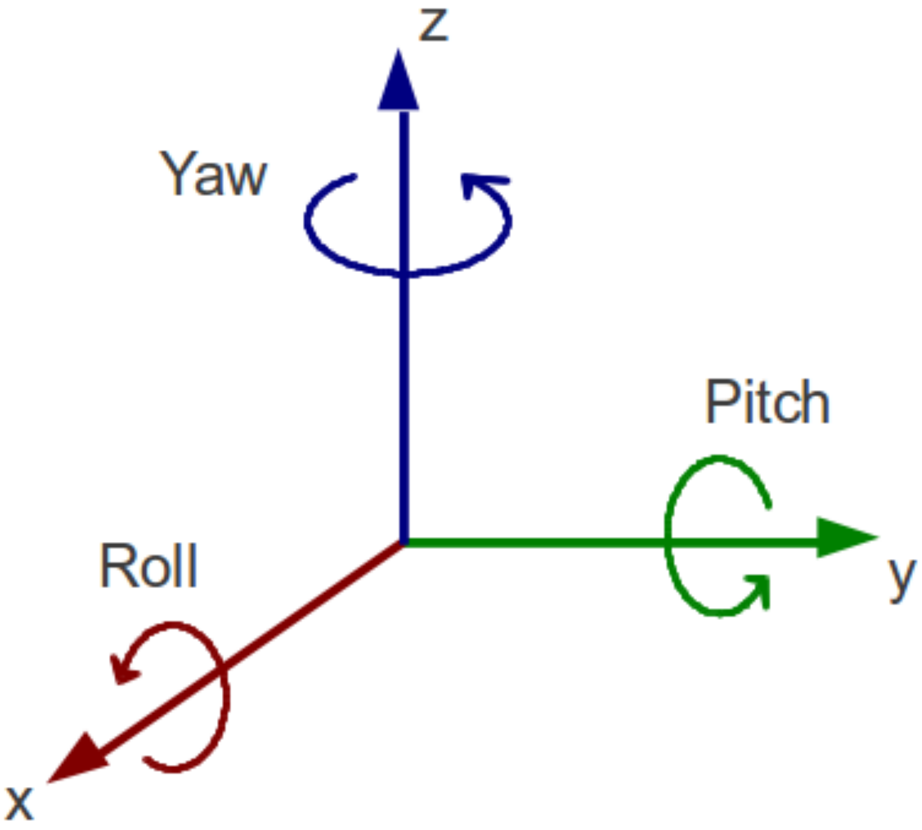
Extrinsics

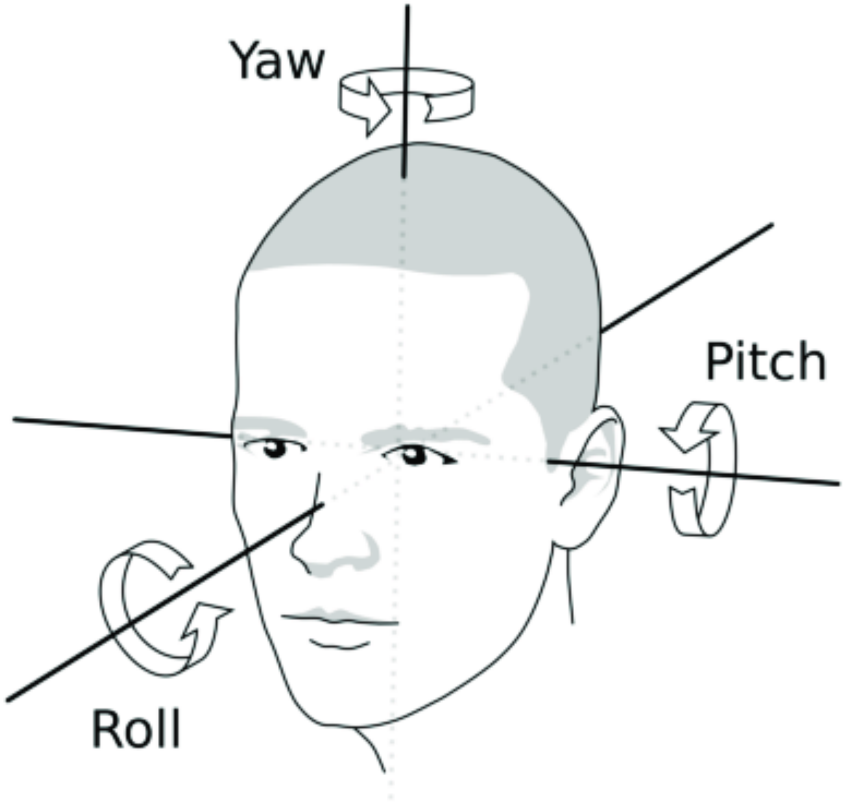
- How do we get the camera to "canonical form"?
  - Canonical form: Center of projection at the origin, x-axis points right, y-axis points down, z-axis points forwards



Step 1: Translate by -**c**
Step 2: Rotate by **R**

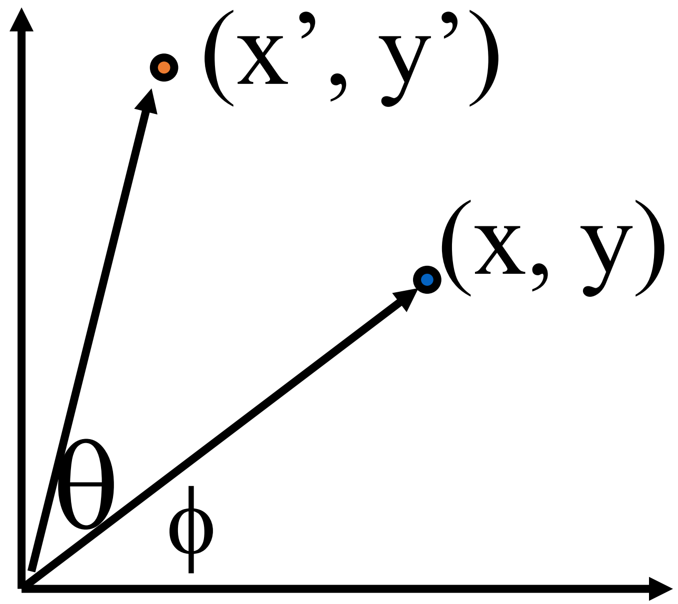
How do we represent 3D rotation?





Euler Angles

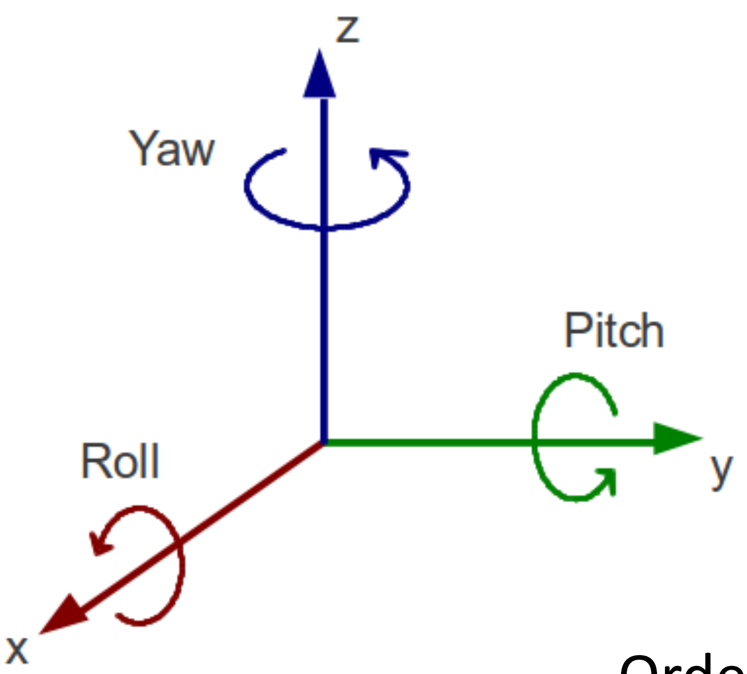
What did we do with 2D rotation?



$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation matrix in 3D



$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Order of applying rotation matters
(composition of 3D rotations is not commutative.)

$$R = R_z(\alpha)\,R_y(\beta)\,R_x(\gamma) = \overset{\text{yaw}}{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{pitch}}{\begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}} \overset{\text{roll}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}}$$

How to derive camera extrinsics?

$M = R\,[I \mid -C]$ (translate first then rotate)

or

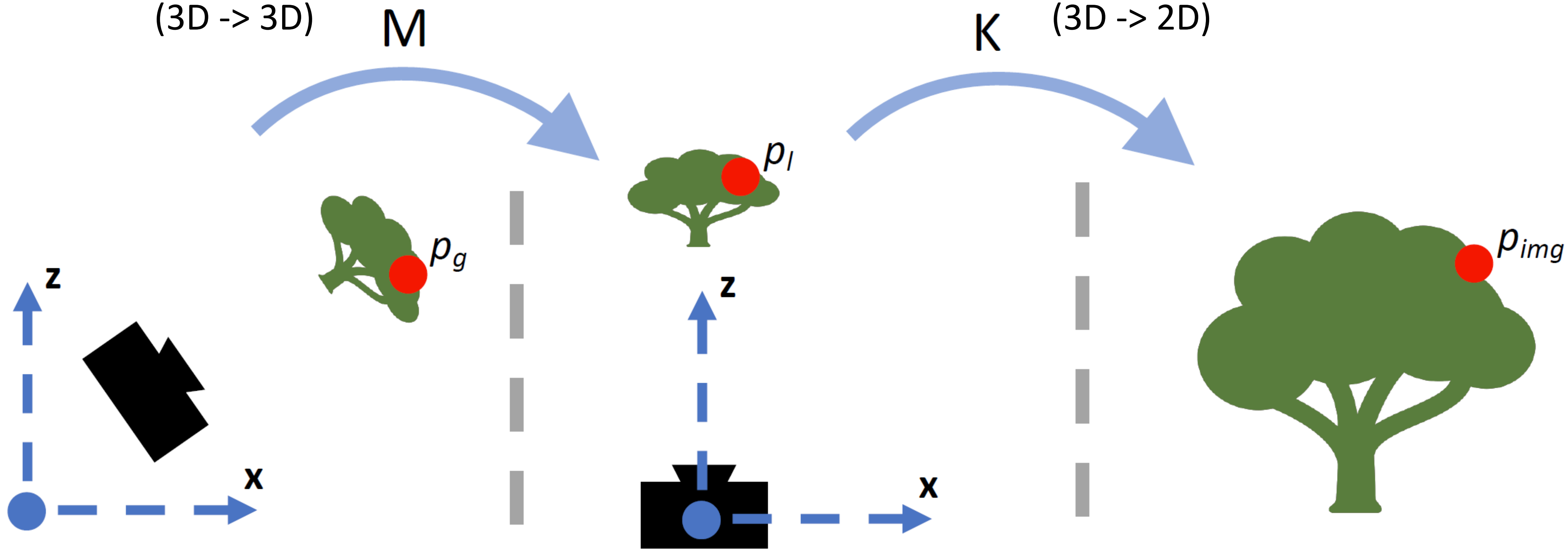
$M = [R \mid t]$, where $t = -RC$ (rotate first then translate)

Show on board

Coordinate frames

Camera Extrinsic: Change coordinate from world to camera. (3D -> 3D)

Camera Intrinsic: Change coordinate from camera to image. (3D -> 2D)



World coordinates

Camera coordinates

Image coordinates

Figure credit: Peter Hedman

Today's Class

- Camera Extrinsic
- Camera Intrinsics and pinhole camera model
- Perspective Distortion
- Other Projection models
- How to calibrate camera, i.e. estimate camera parameters (next class)

Geometric Model: A Pinhole Camera

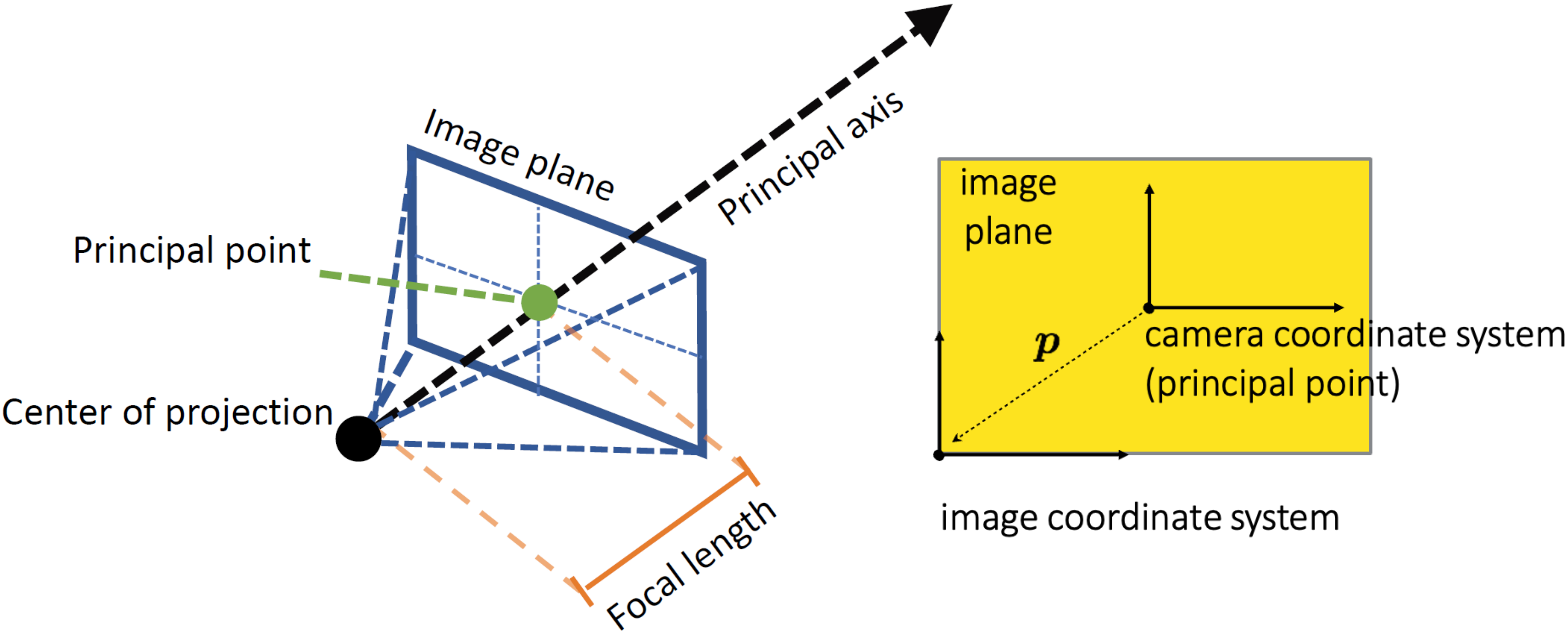
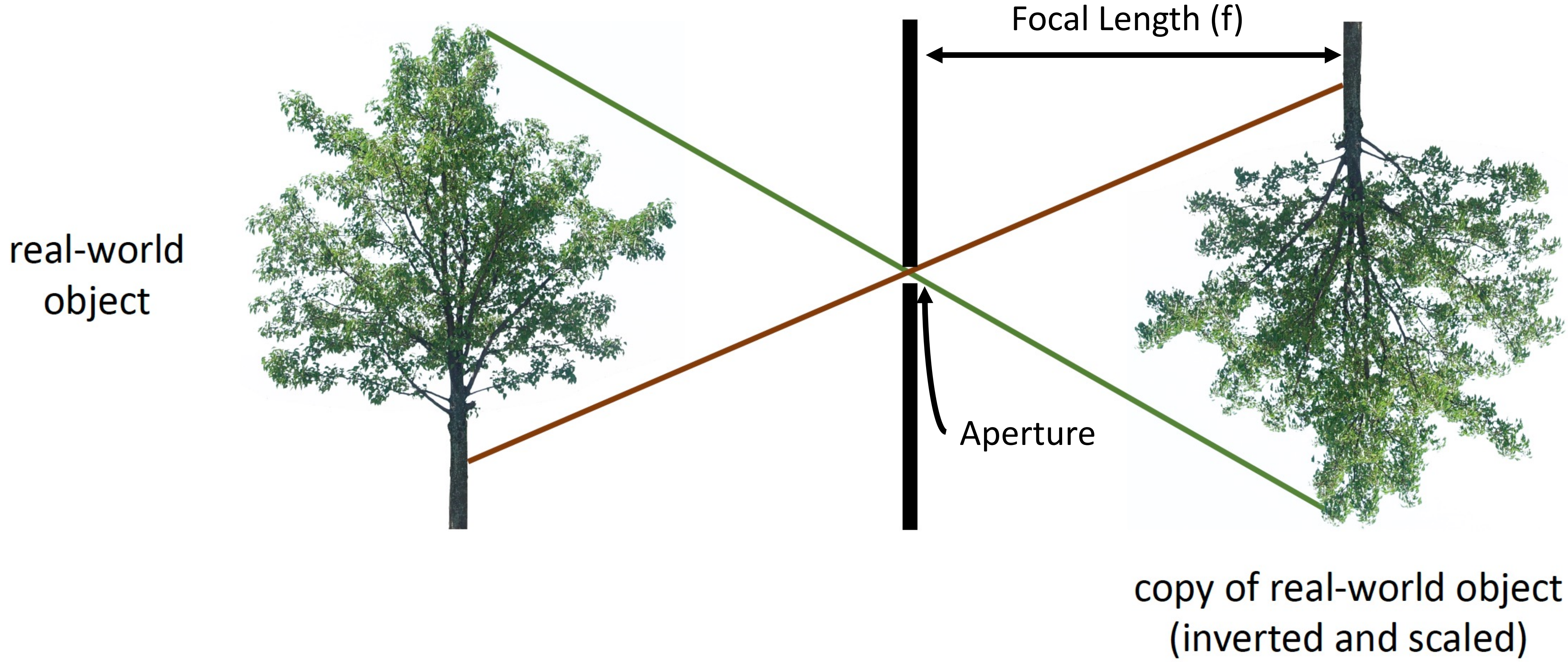
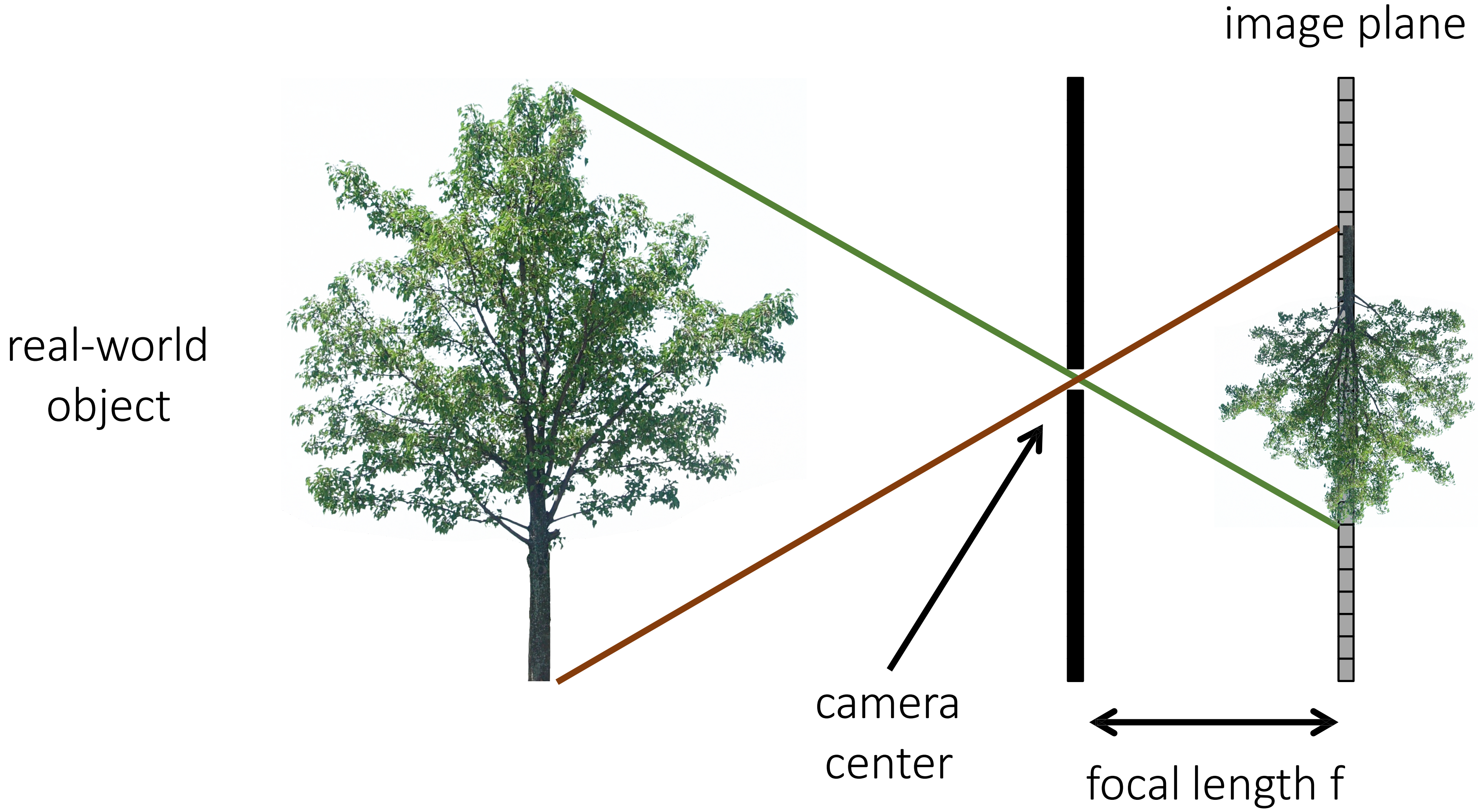


Figure credit: Peter Hedman

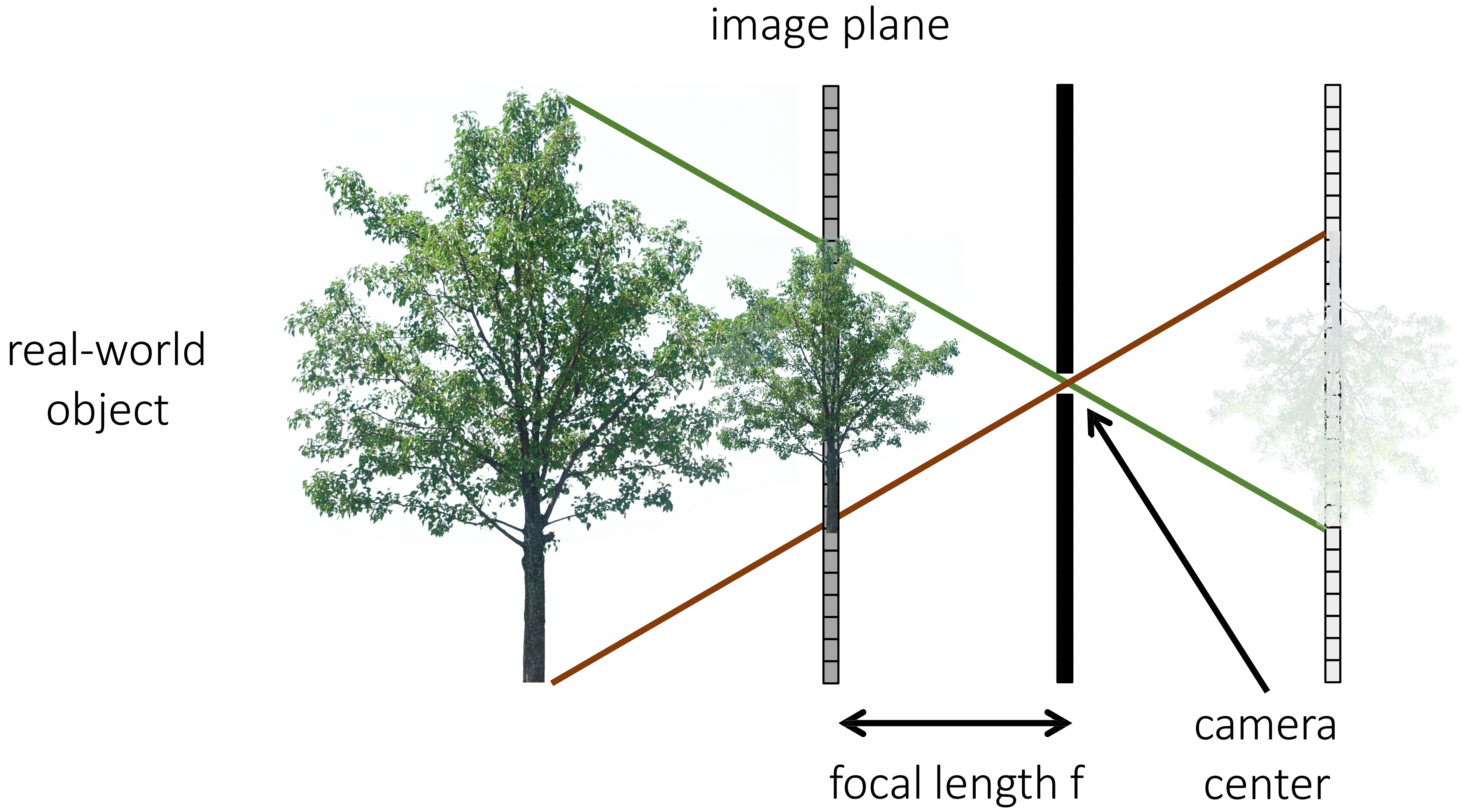
Pinhole Camera



The pinhole camera



The (rearranged) pinhole camera



Focal length

- Can think of as “zoom”

24mm

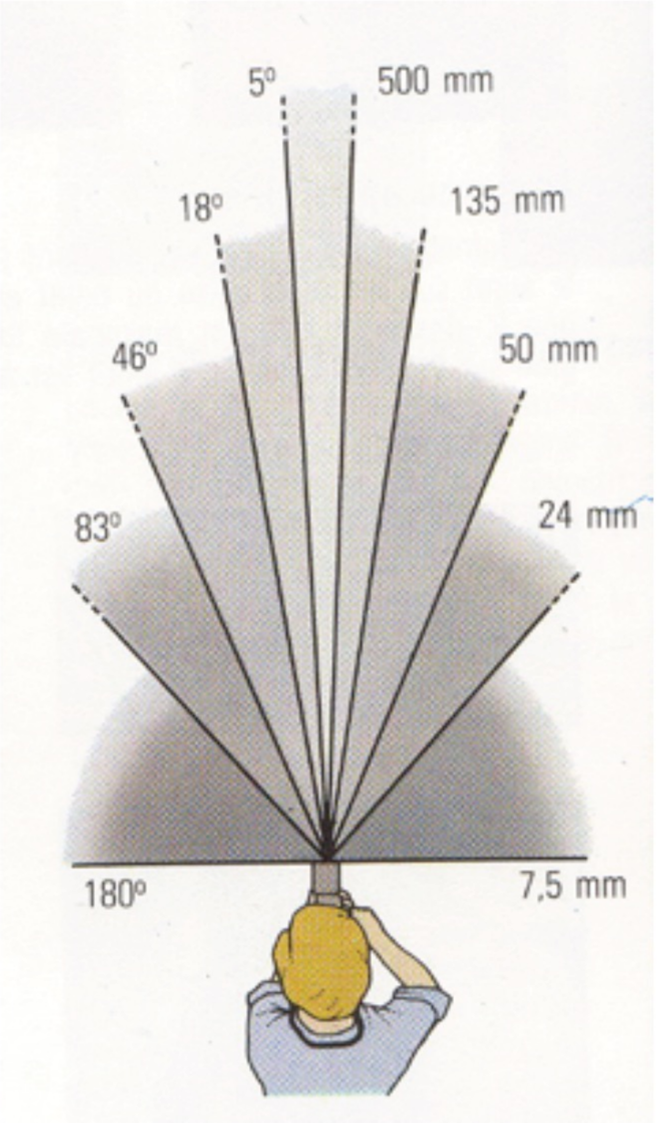
50mm

200mm

800mm

- Also related to *field of view (inversely)*

Focal length in practice



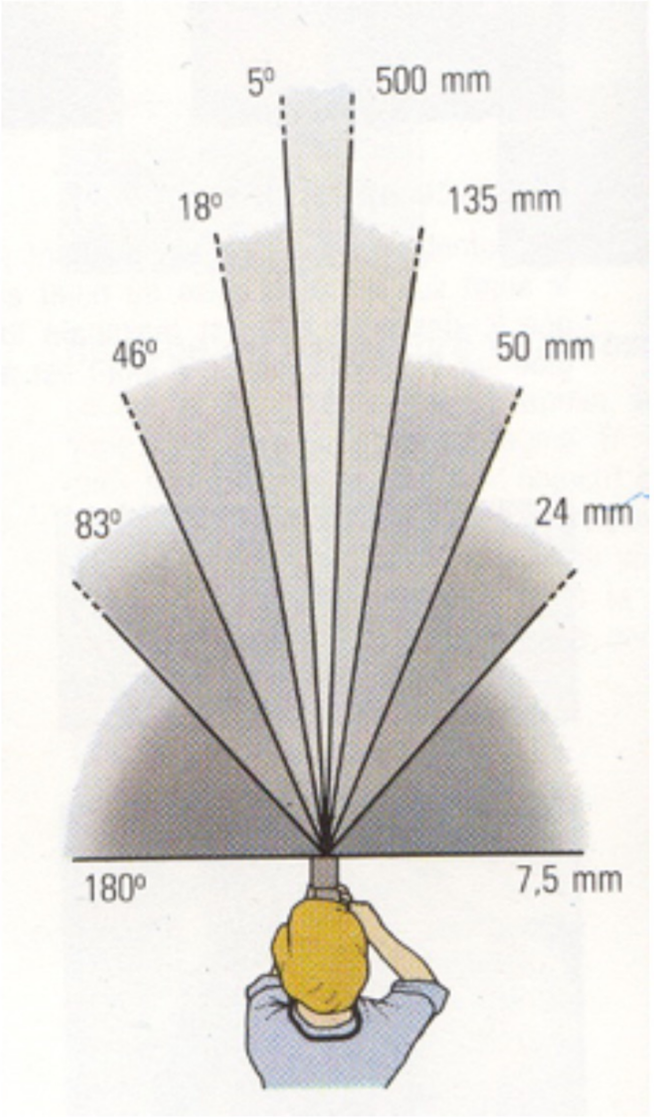
24mm

50mm

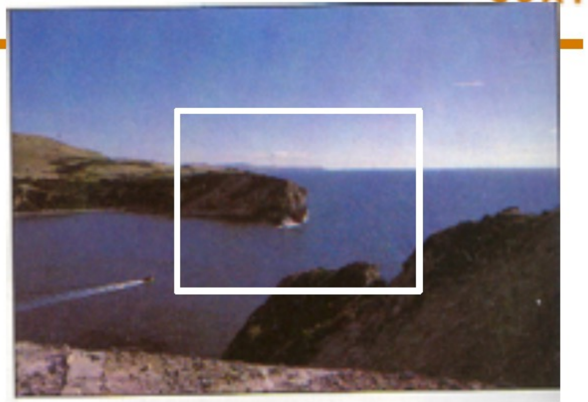
135mm

Fredo Durand

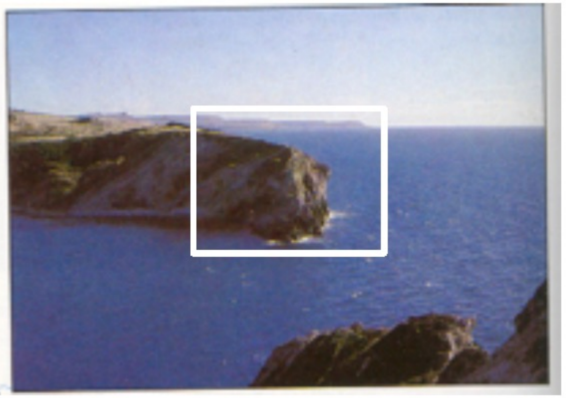
Focal length = cropping



24mm

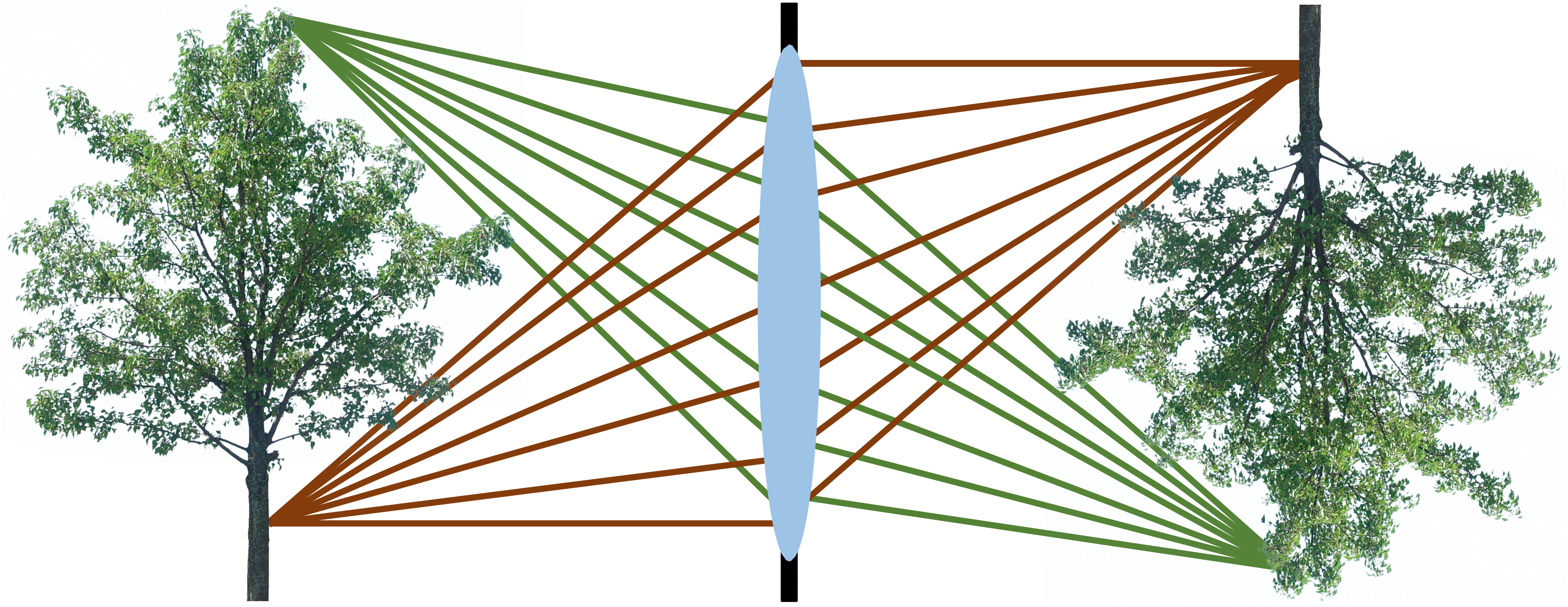


50mm



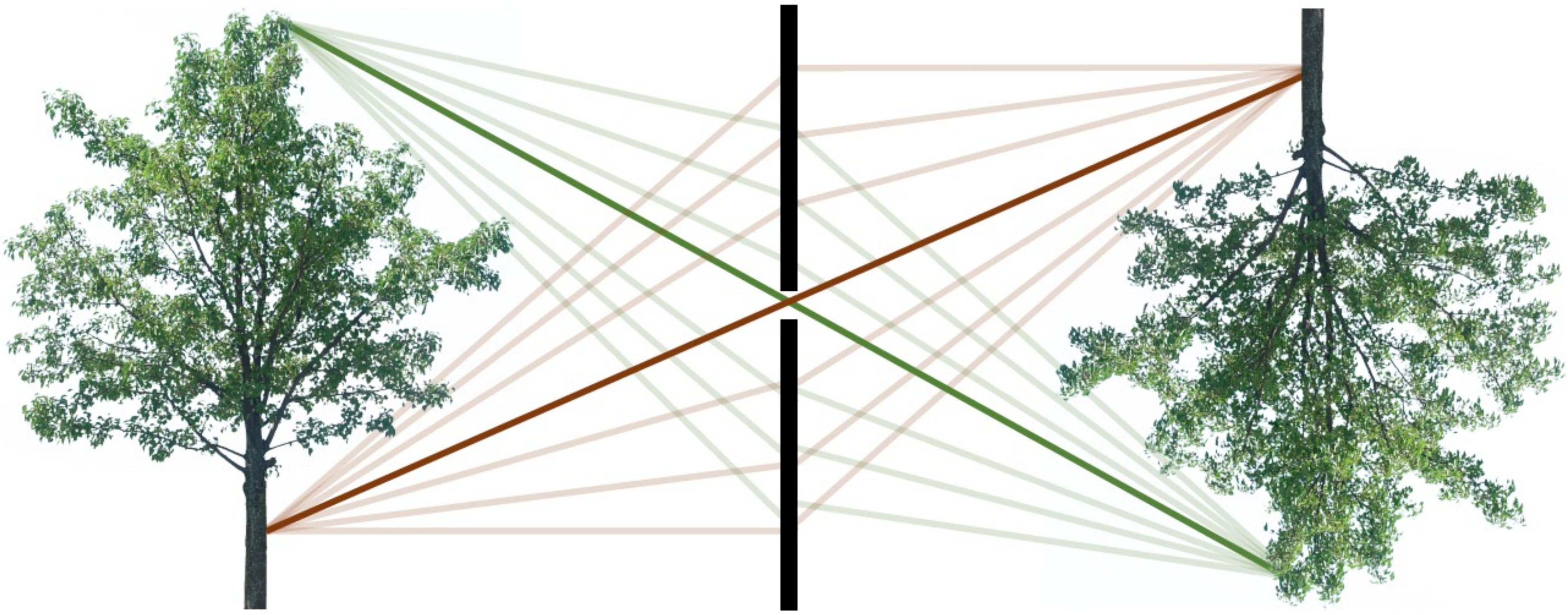
135mm

The lens camera



Lenses map “bundles” of rays from points on the scene to the sensor.

The pinhole camera

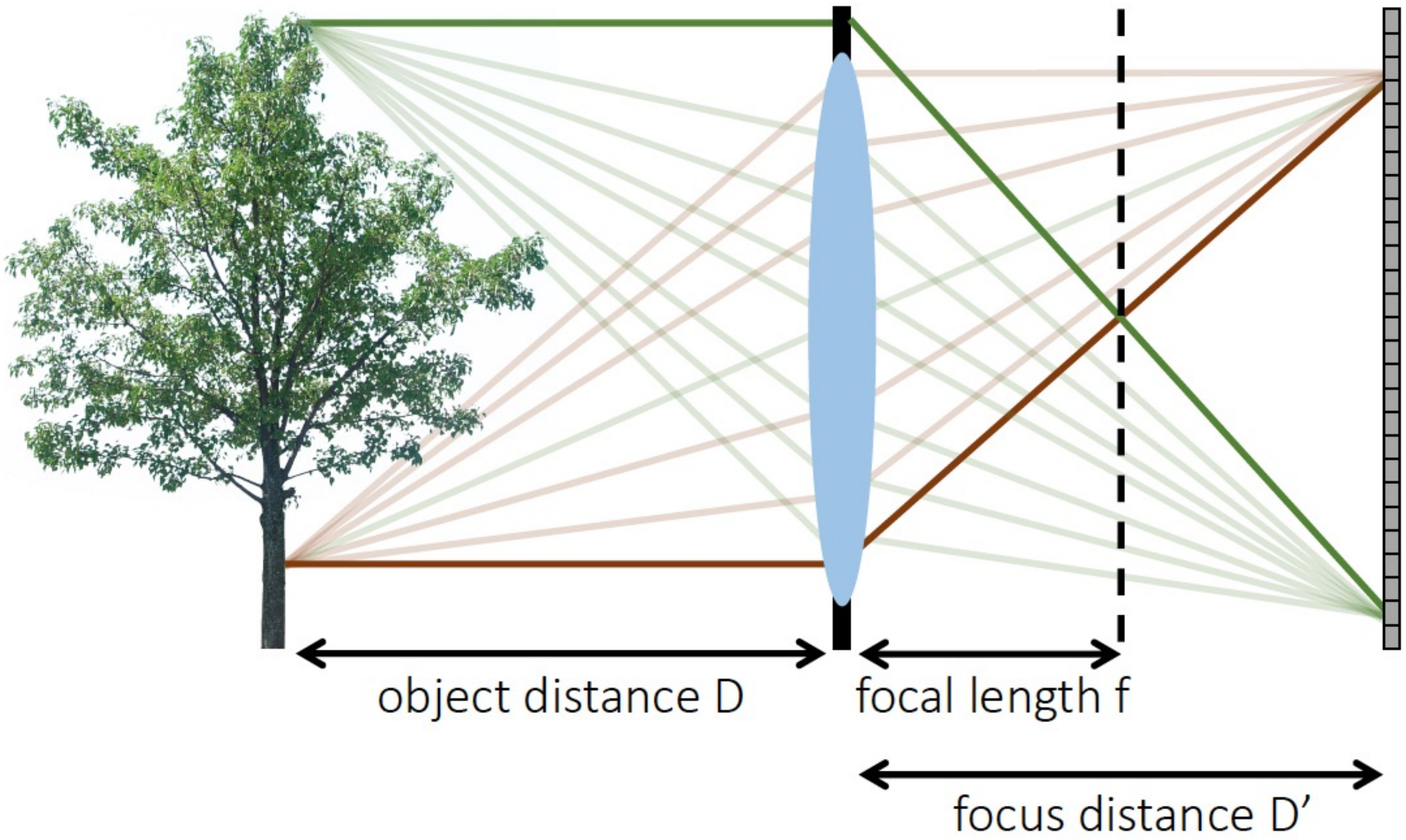


Central rays propagate in the same way for both models!

Important Difference: focal length

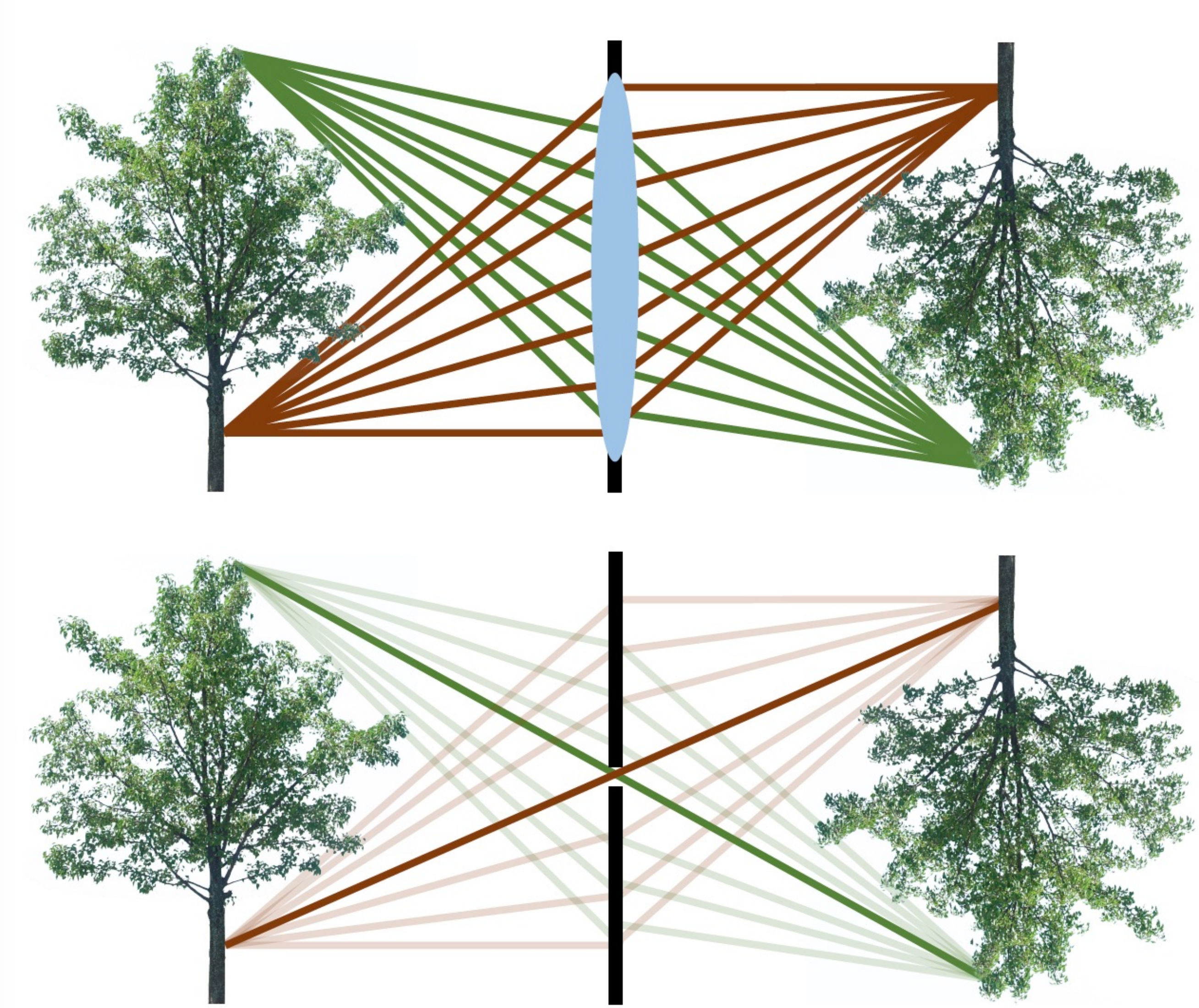
In a pinhole camera, focal length is distance between aperture and sensor

Important Difference: focal length



In a lens camera, focal length is distance where parallel rays intersect

Describing both lens and pinhole cameras



We can derive properties and descriptions that hold for both camera models if:

- We use only central rays.
- We assume the lens camera is in focus.
- We assume that the focus distance of the lens camera is equal to the focal length of the pinhole camera.

From now on, we will describe cameras as pinhole cameras!
Focal length will mean aperture-sensor distance.

Geometric Model: A Pinhole Camera

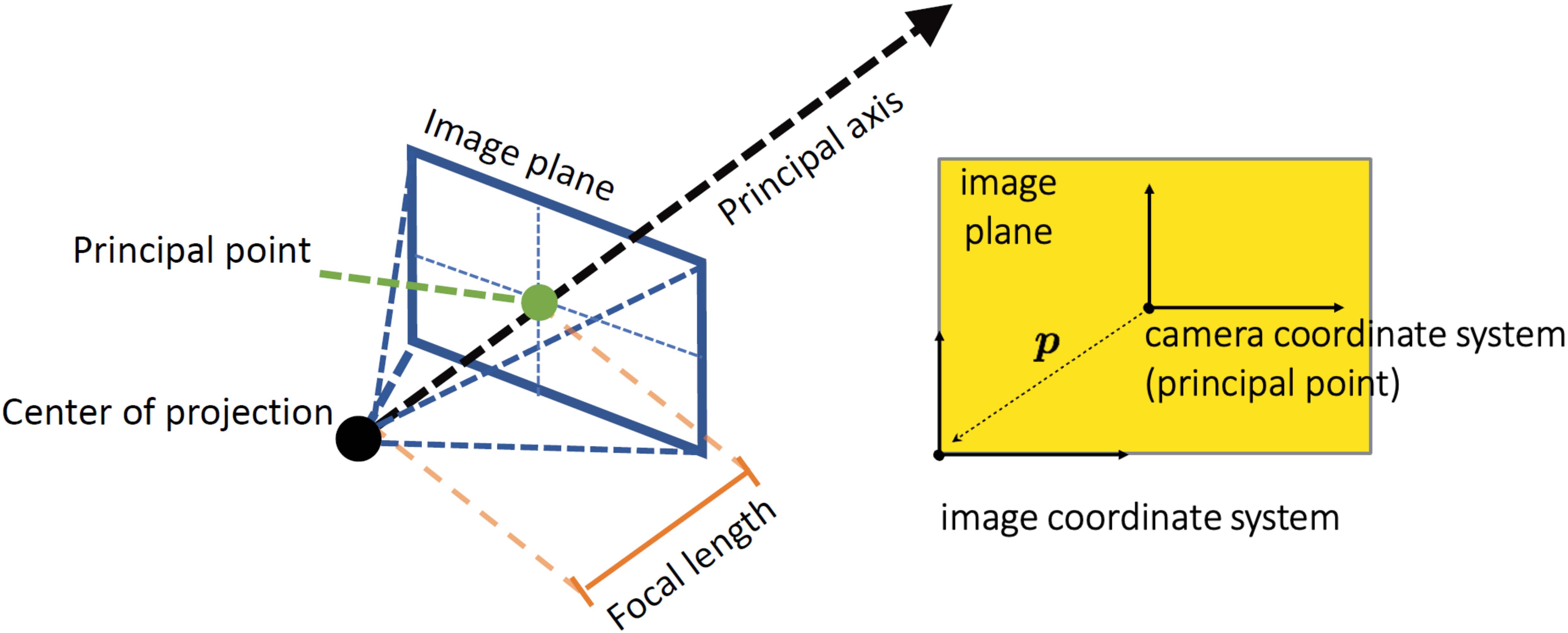
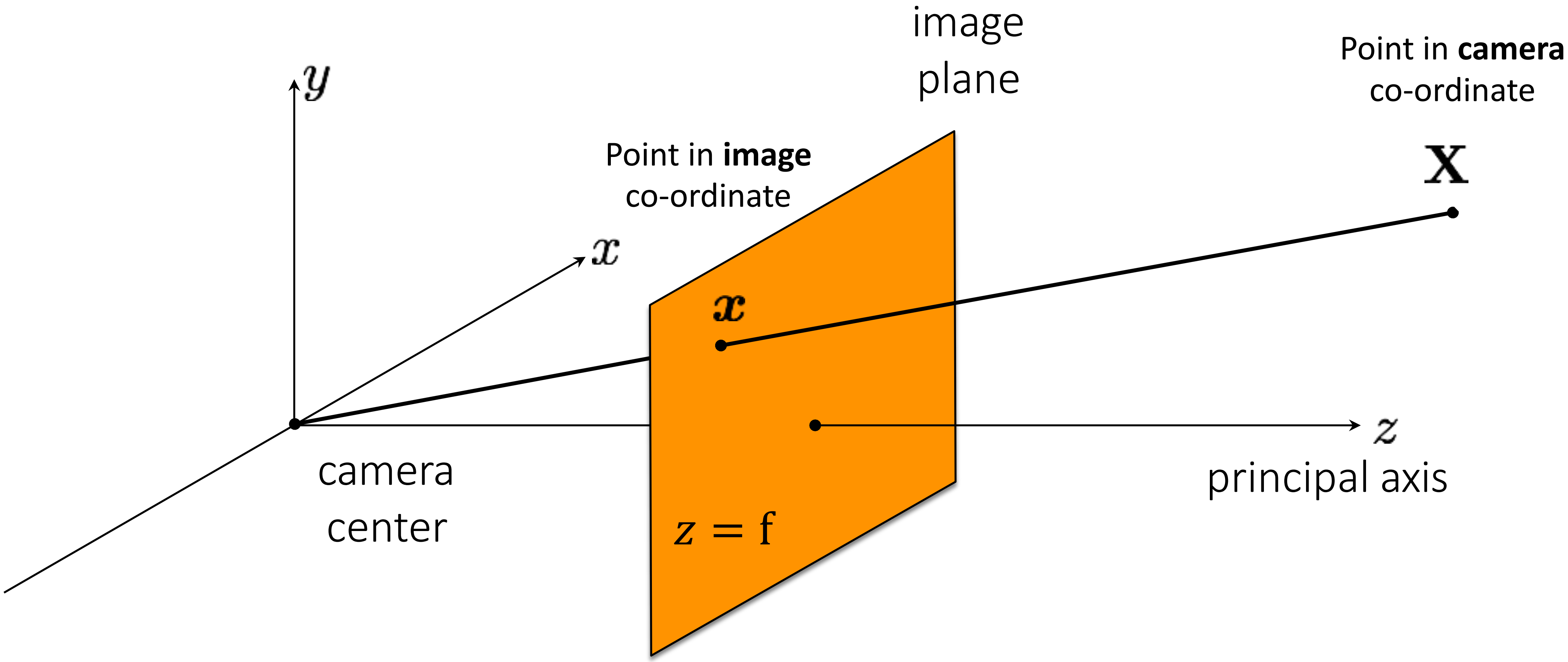


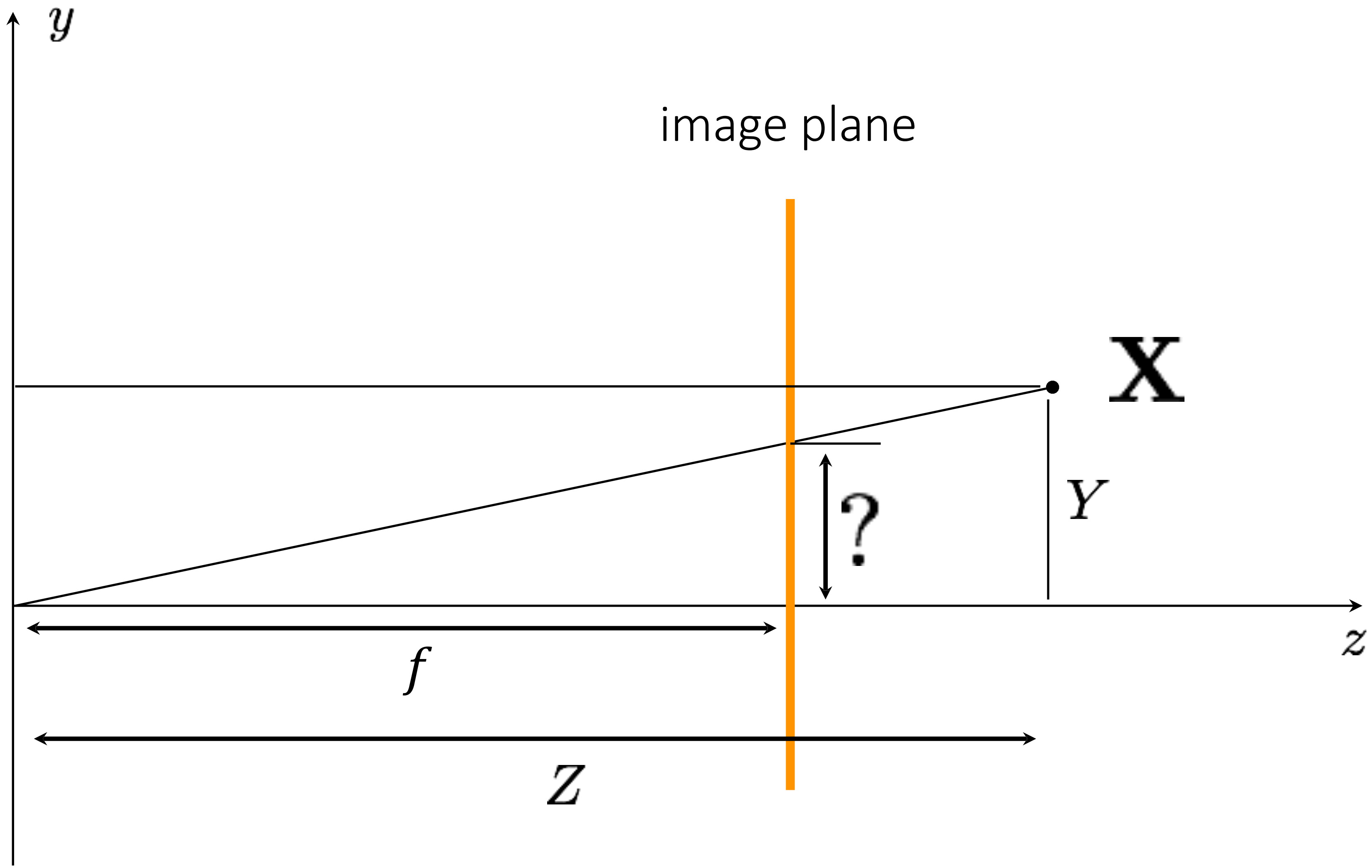
Figure credit: Peter Hedman

The (rearranged) pinhole camera



What is the equation for image coordinate x in terms of X?

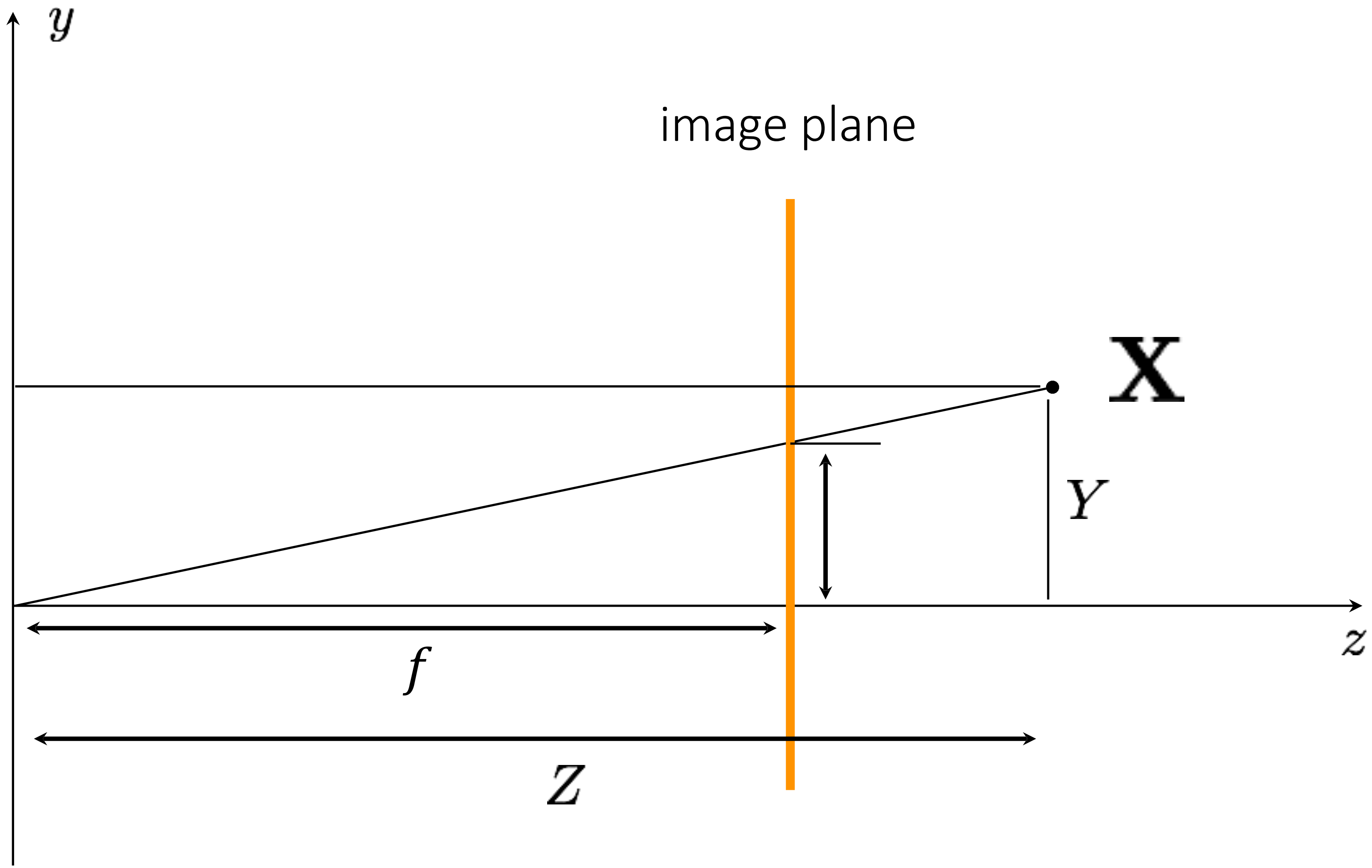
The 2D view of the (rearranged) pinhole camera



What is the equation for image coordinate x in terms of X?

Show on board

The 2D view of the (rearranged) pinhole camera [*]



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

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

Perspective Projection

Projection is a matrix multiply using homogeneous coordinates:

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

divide by third coordinate

This is known as **perspective projection**

- The matrix is the **projection matrix**
- (Can also represent as a 4x4 matrix – OpenGL does something like this)

Perspective Projection

How does scaling the projection matrix change the transformation?

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

Scale by *f*:

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

Scaling a projection matrix produces an equivalent projection matrix!

The pinhole camera matrix for arbitrary focal length

Relationship from similar triangles:

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

General camera model *in homogeneous coordinates*:

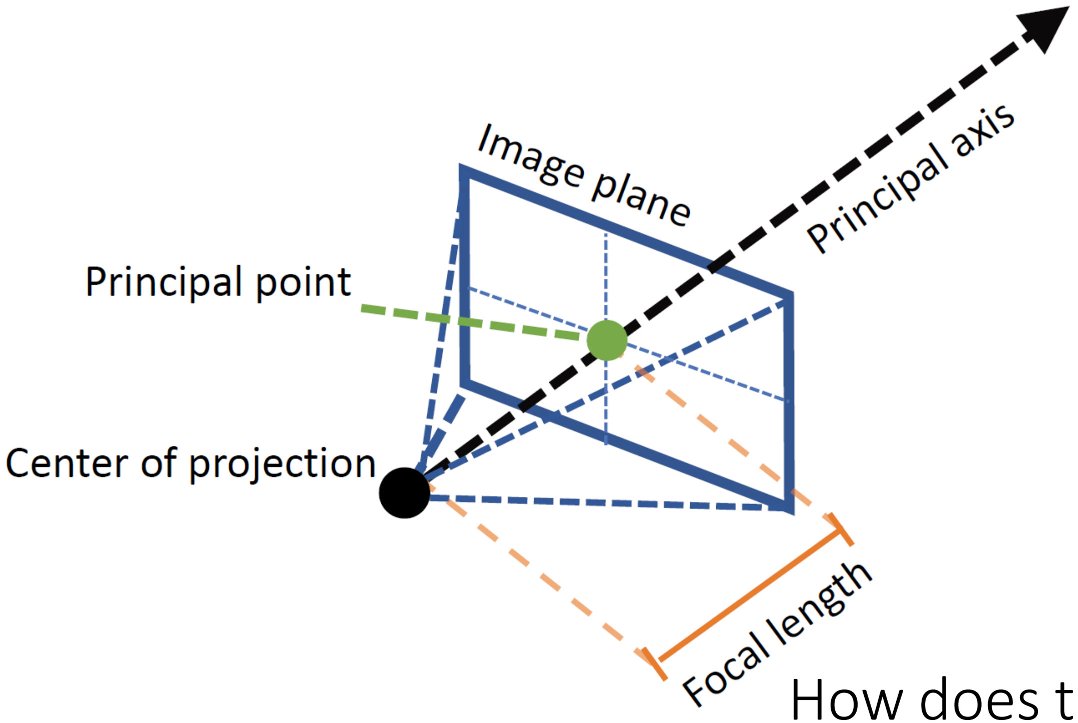
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

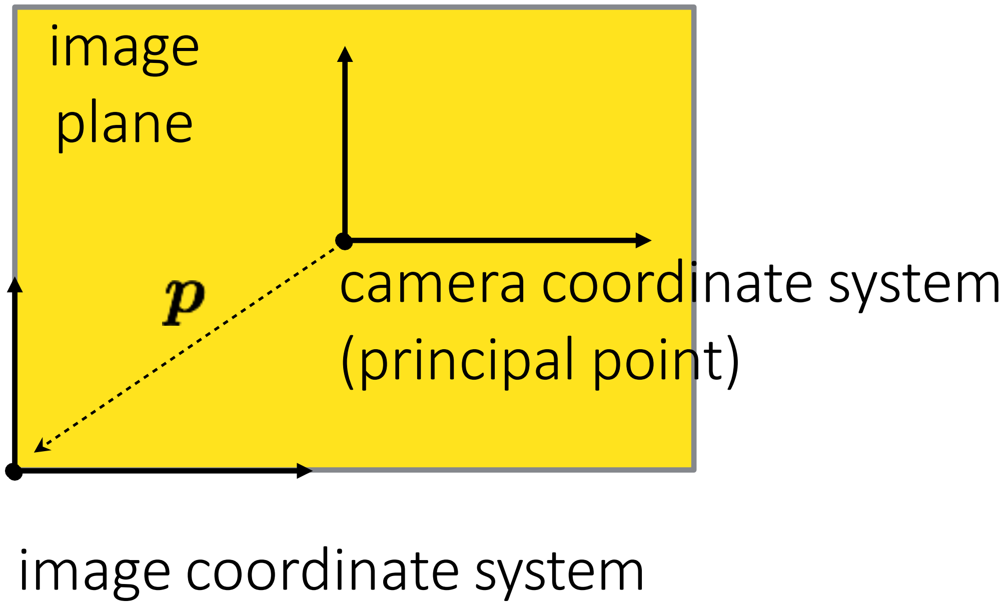
What does the pinhole camera projection look like?

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Generalizing the camera matrix

In particular, the camera origin and image origin may be different:



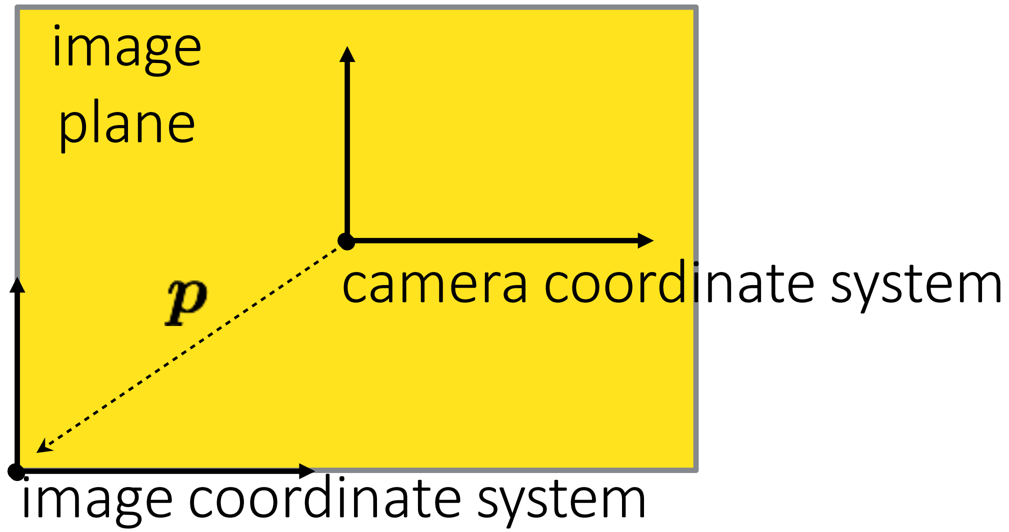


How does the camera matrix change?

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Generalizing the camera matrix

In particular, the camera origin and image origin may be different:



How does the camera matrix change?

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

shift vector transforming camera origin to image origin

Show on board, why?

Typical Intrinsics matrix

General case

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

2D affine transform corresponding to a scale by *f* (focal length) and a translation by (c_x, c_y) (principal point)

Maps 3D rays to 2D pixels

α : **aspect ratio** (1 unless pixels are not square)

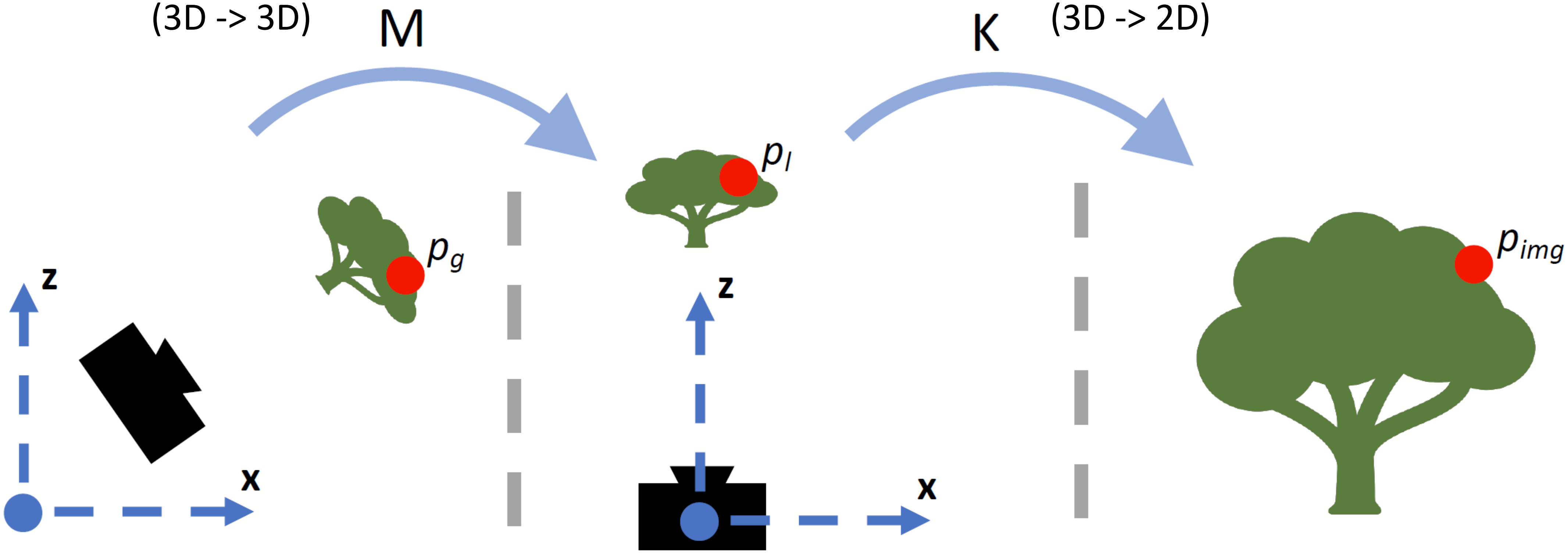
s : **skew** (0 unless pixels are shaped like rhombi/parallelograms)

(c_x, c_y) : **principal point** ((w/2,h/2) unless optical axis doesn't intersect projection plane at image center)

Coordinate frames

Camera Extrinsic: Change coordinate from world to camera. (3D -> 3D)

Camera Intrinsic: Change coordinate from camera to image. (3D -> 2D)



World coordinates

Camera coordinates

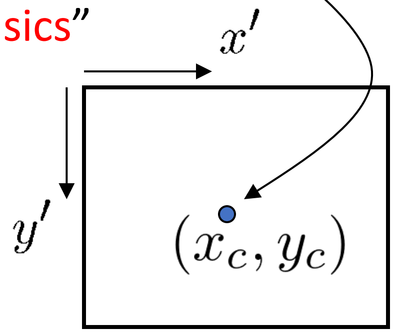
Image coordinates

Figure credit: Peter Hedman

Camera parameters

A camera is described by several parameters

- Translation T of the optical center from the origin of world coords
- Rotation R of the image plane
- focal length f, principal point (c_x, c_y), pixel aspect size α
- blue parameters are called “extrinsics,” red are “intrinsics”



Projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$

- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{bmatrix}}_{\text{translation}}$$

- The definitions of these parameters are **not** completely standardized
  - especially intrinsics—varies from one book to another

General pinhole camera matrix

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

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array} \right]$$

intrinsic parameters

extrinsic parameters

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

3D rotation

3D translation

Recap

What is the size and meaning of each term in the camera matrix?

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}]$$

$\mathbf{K}$	$\mathbf{R}$	$\mathbf{I}$	$\mathbf{C}$
3x3 intrinsics	3x3 3D rotation	3x3 identity	3x1 3D translation

A sample HW/Mid-term problem

- Suppose a camera center is located at (1,1,2) in world co-ordinate and is pointing towards Z-direction with a focal length of f.

a) Write the equation of the image plane.

b) For a point (10,10,17) in world-coordinate, where will that point be projected on the image plane. Write the co-ordinate of that point in:

    i) world co-ordinate

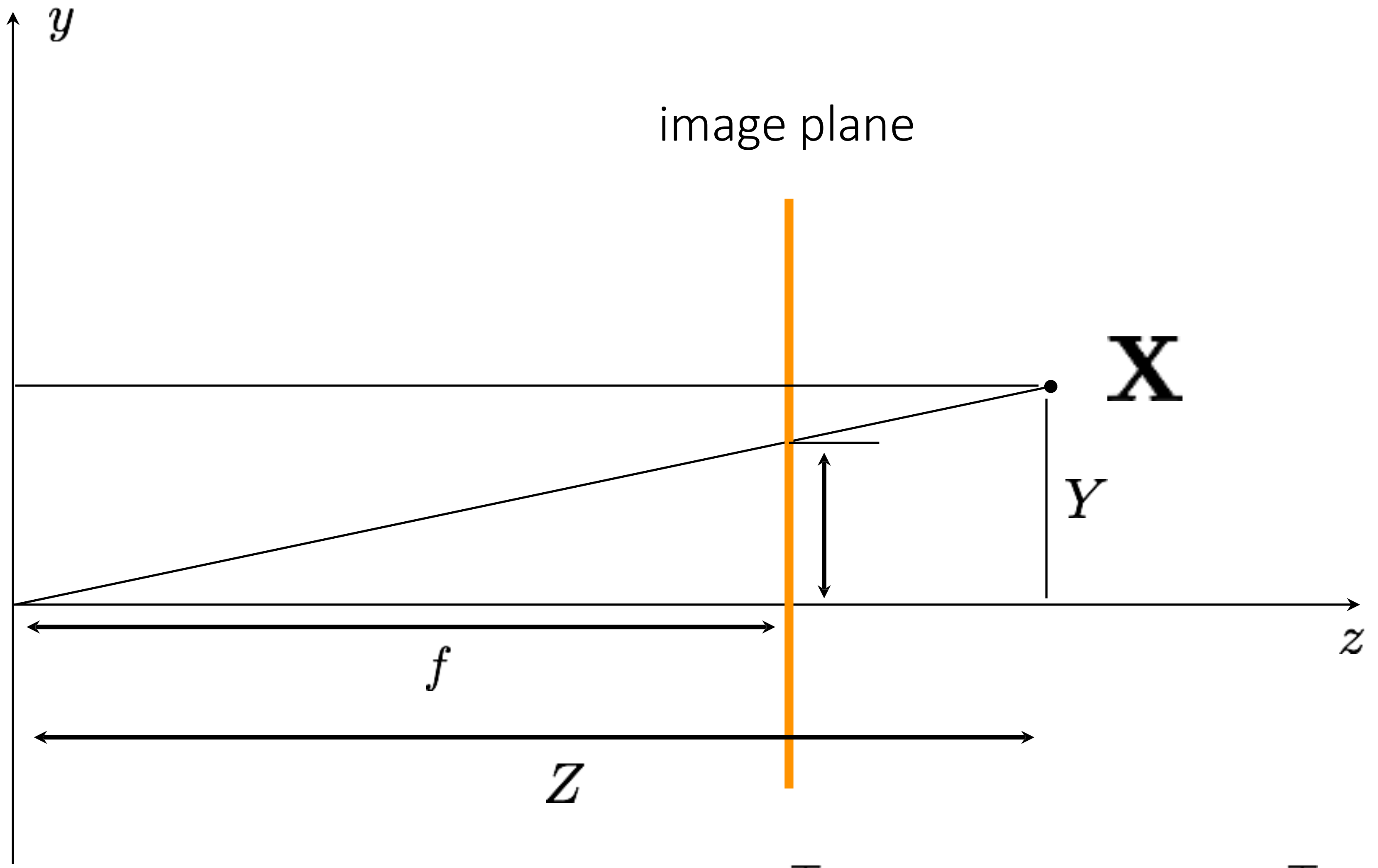
    ii) camera co-ordinate

    iii) image co-ordinate (?)

Today’s Class

- Camera Extrinsic
- Camera Intrinsics and pinhole camera model
- Perspective Distortion
- Other Projection models
- How to calibrate camera, i.e. estimate camera parameters (next class)

The 2D view of the (rearranged) pinhole camera



Perspective distortion: magnification changes with depth

Perspective projection in 2D

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

Perspective distortion

long focal length

mid focal length

short focal length

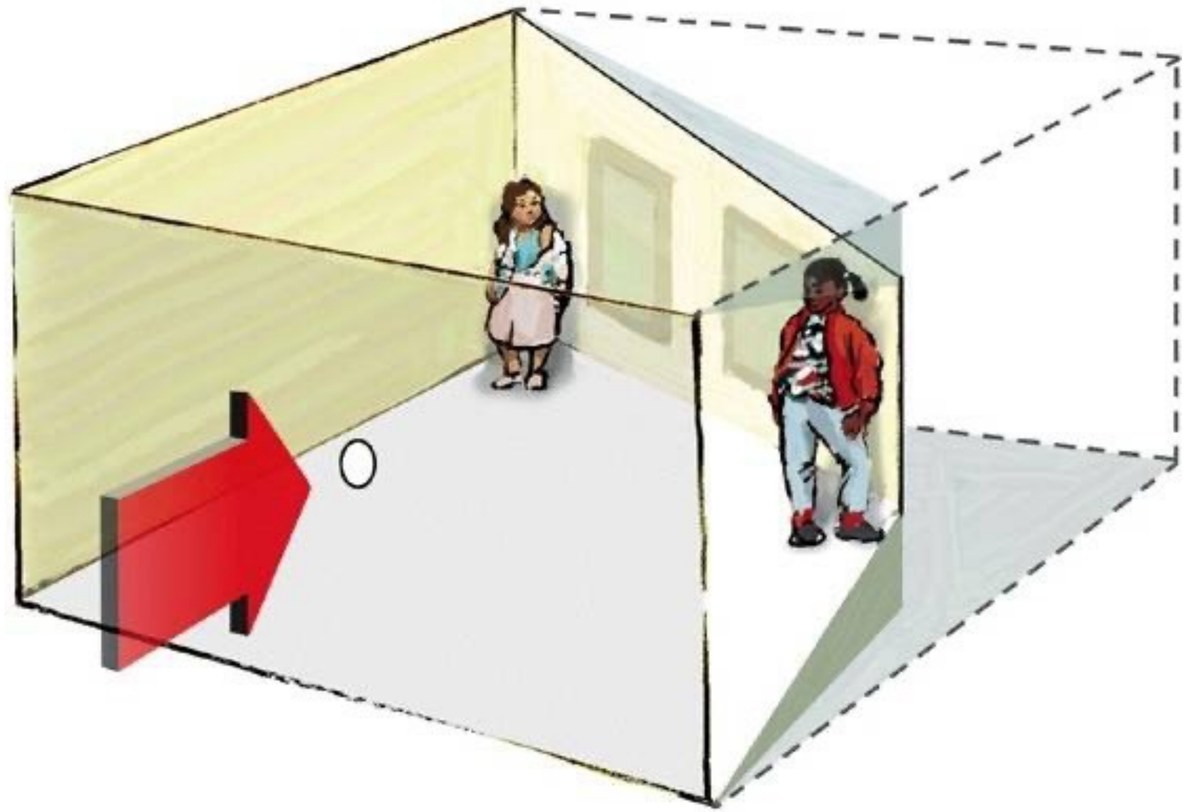
Perspective distortion

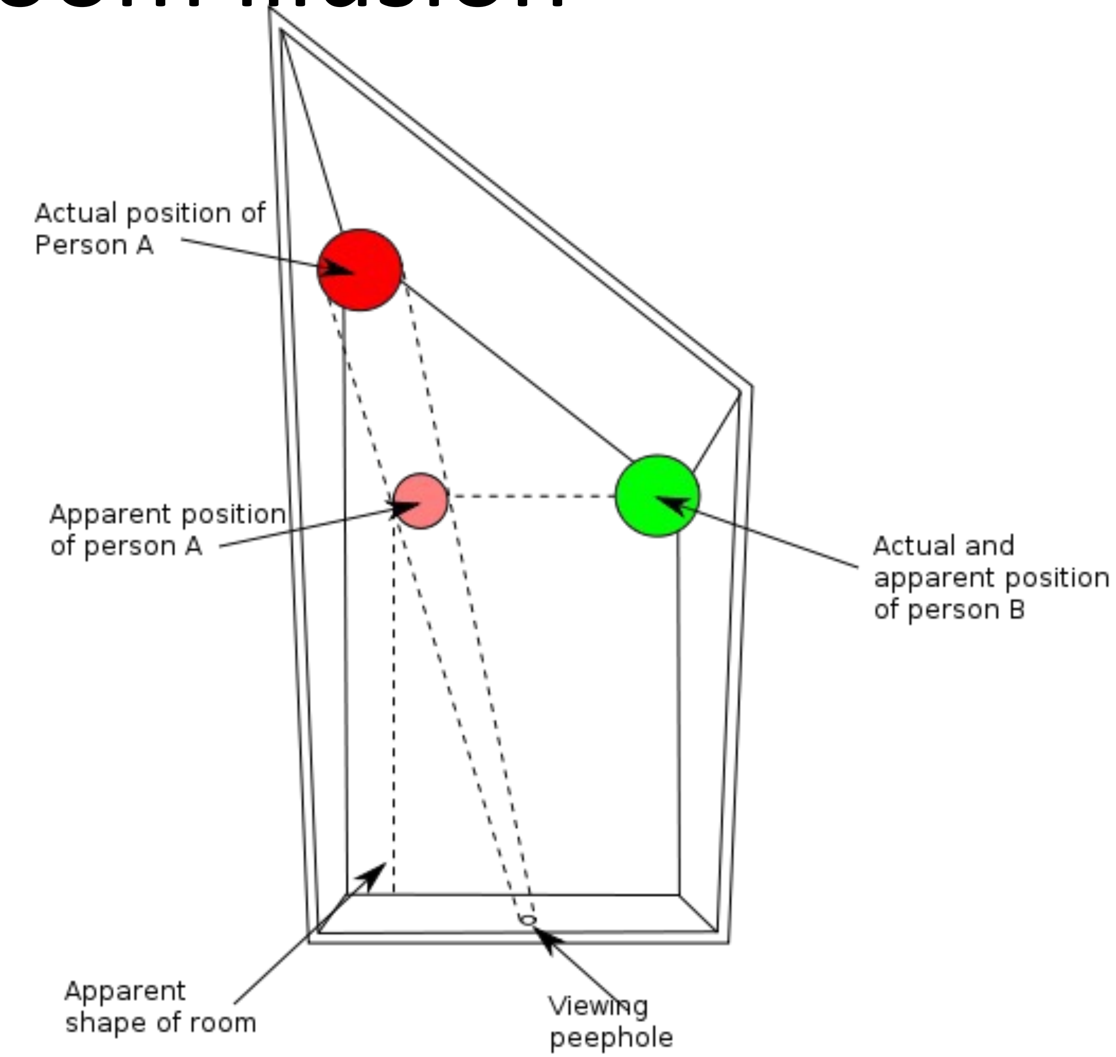
http://petapixel.com/2013/01/11/how-focal-length-affects-your-subjects-apparent-weight-as-seen-with-a-cat/

Forced perspective

The Ames room illusion

The Ames room illusion

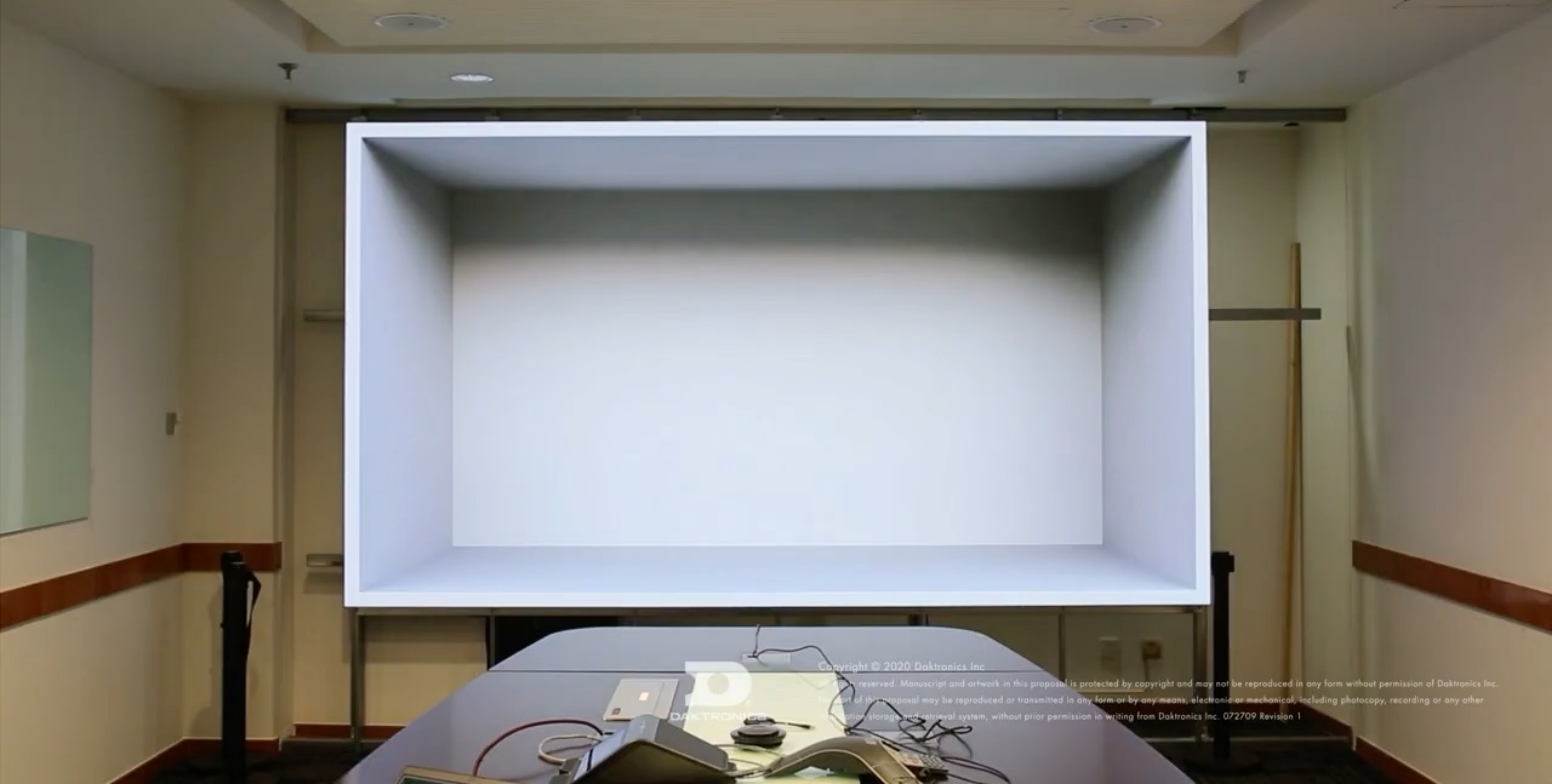




Dolly Zoom aka Vertigo Effect


Fredo Durand

Forced Perspective in displays

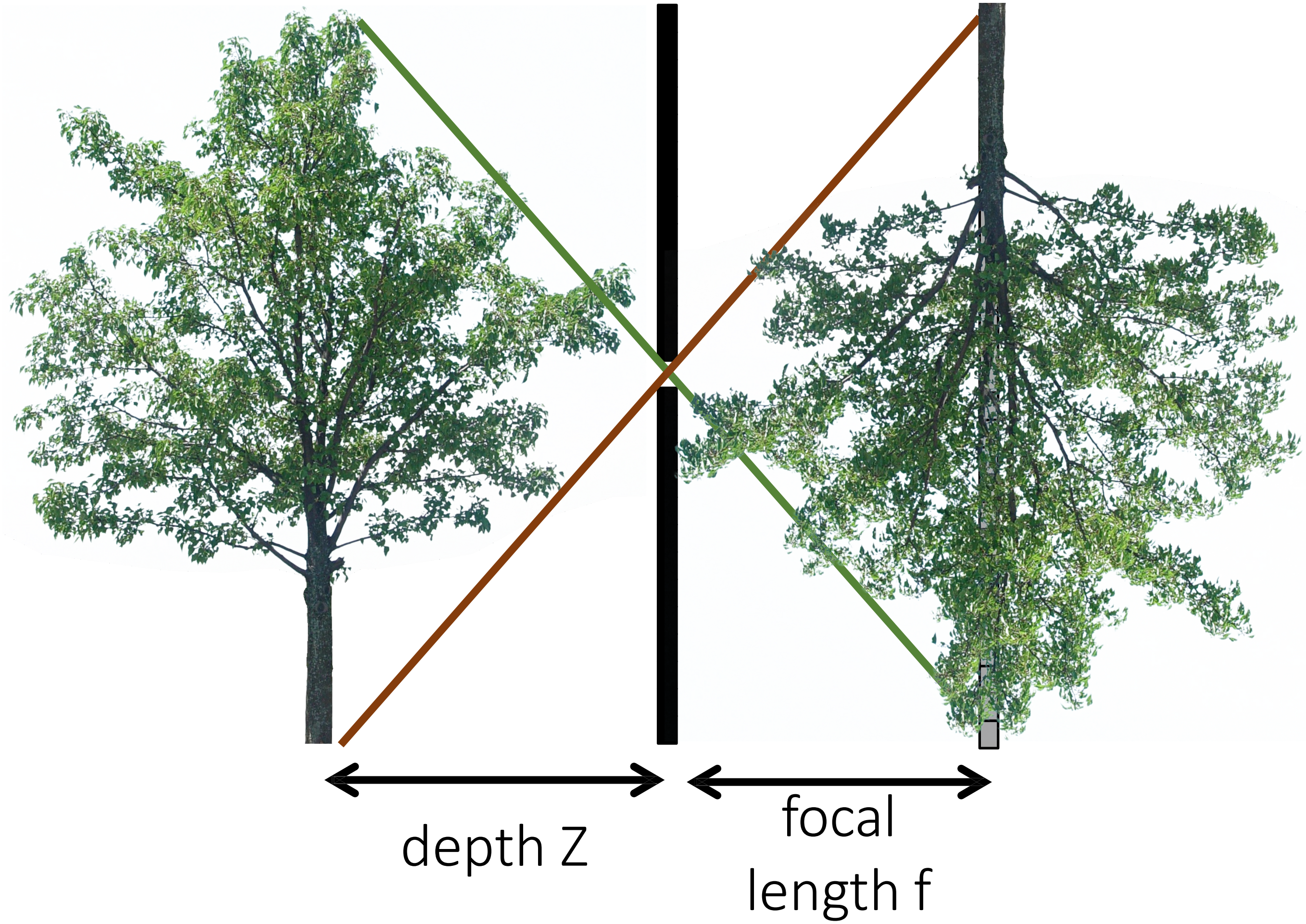


Today’s Class

- Camera Extrinsic
- Camera Intrinsics and pinhole camera model
- Perspective Distortion
- Other Projection models
- How to calibrate camera, i.e. estimate camera parameters (next class)

What if…

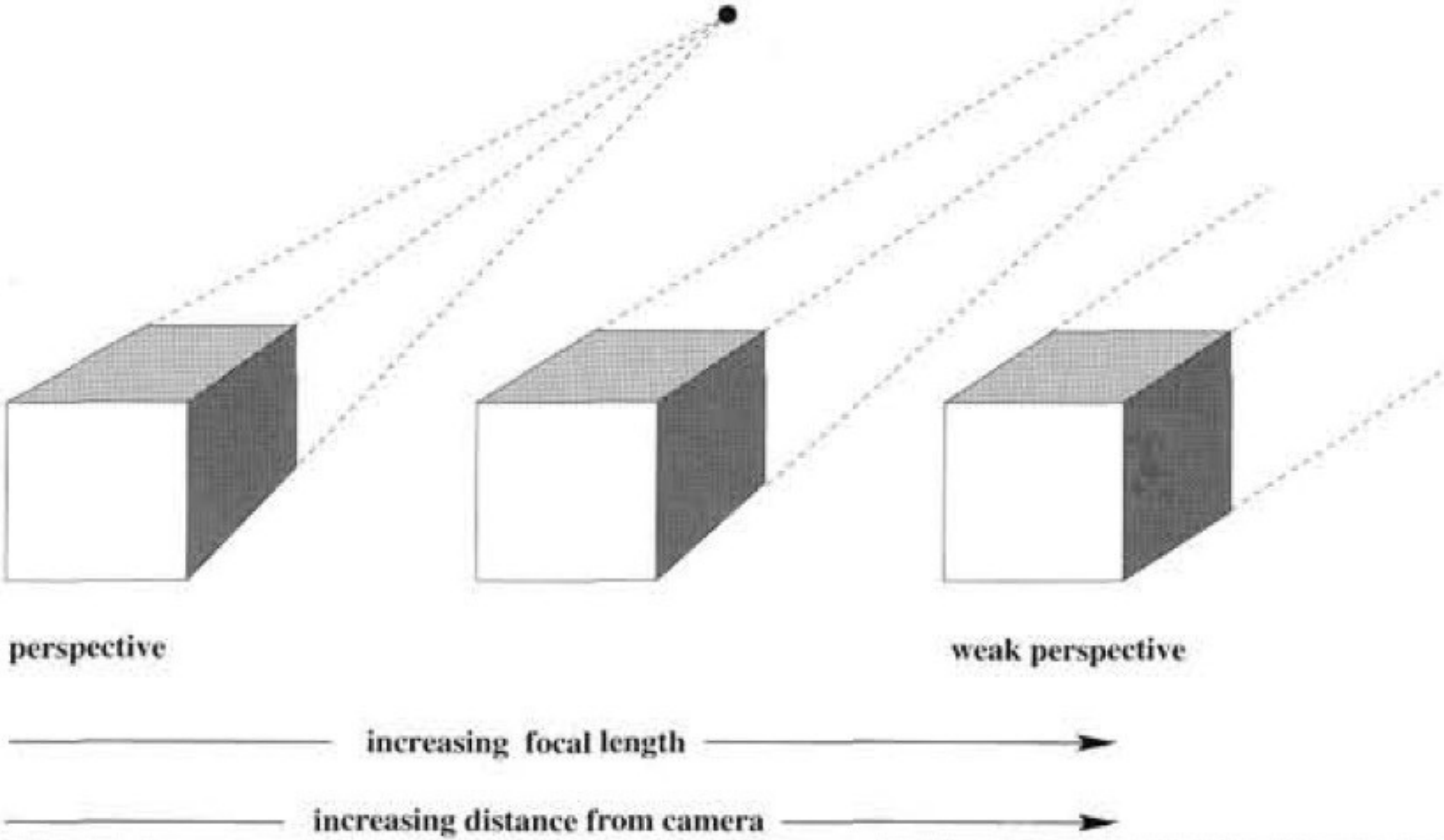
real-world object



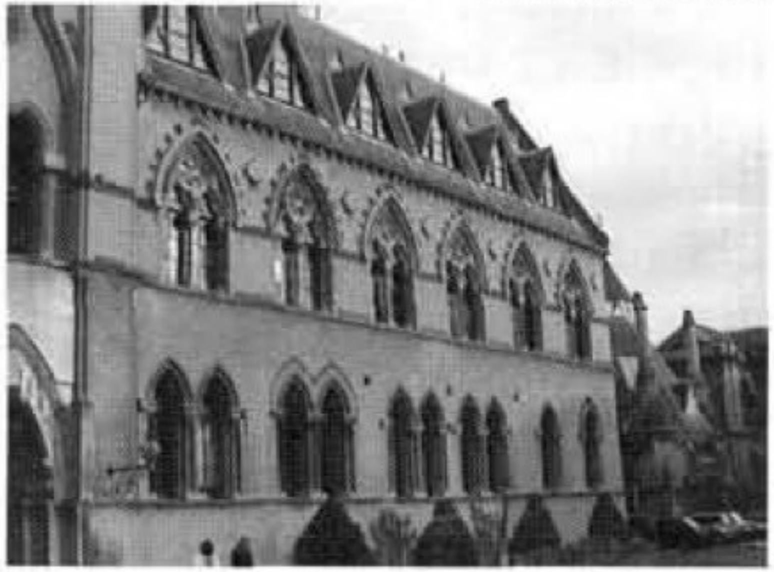
… we continue increasing Z and f while maintaining same magnification?

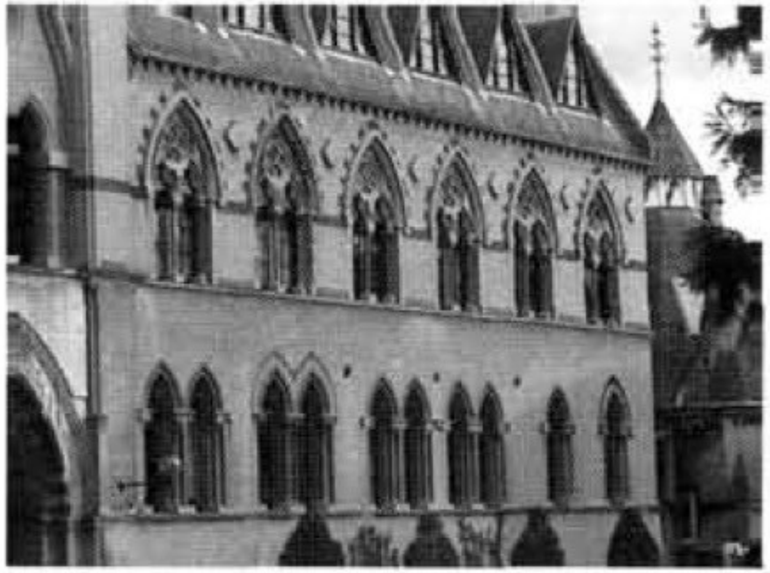
$$f \rightarrow \infty \text{ and } \frac{f}{Z} = \text{constant}$$

camera is *close* to object and has *small* focal length

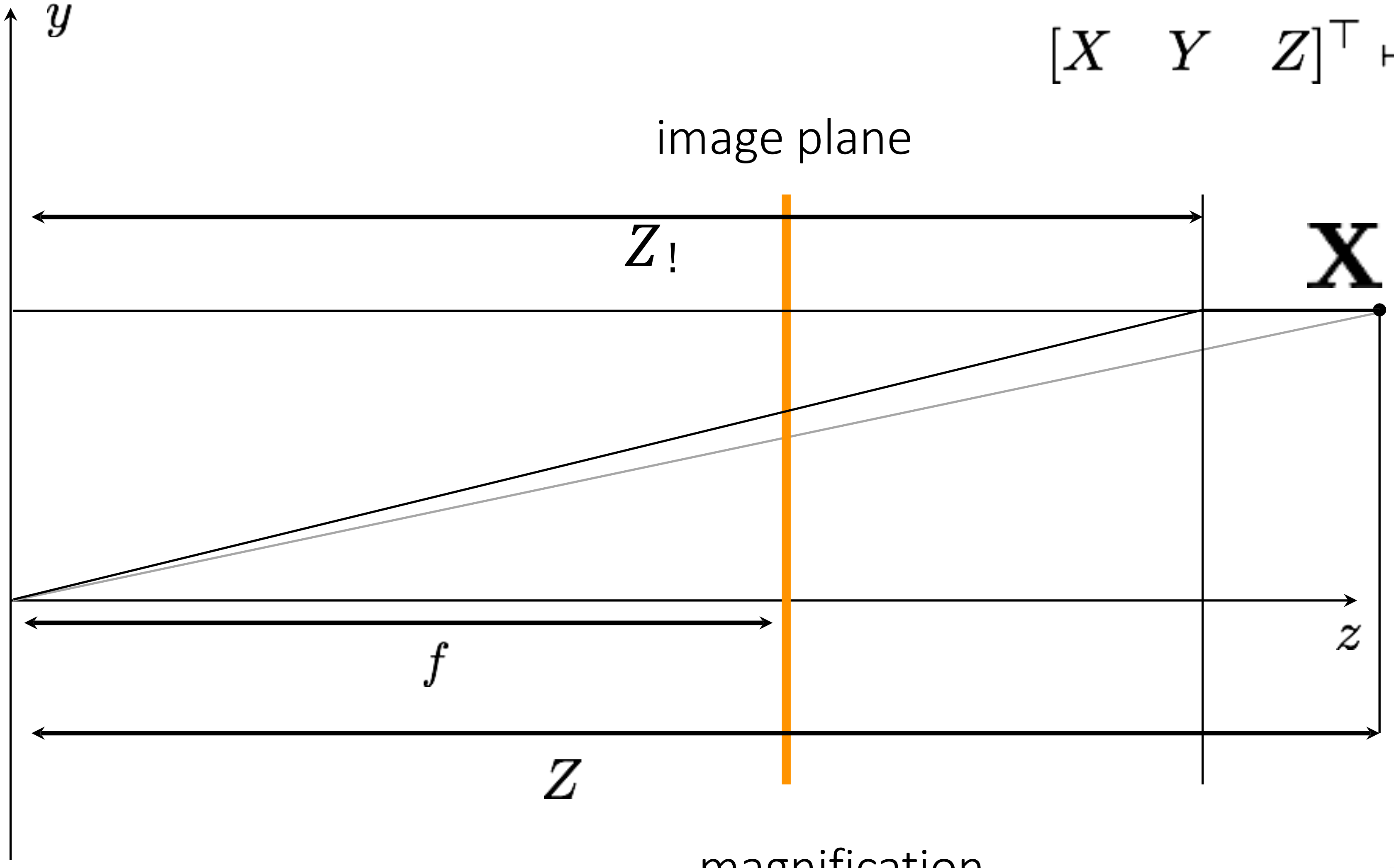


camera is *far* from object and has *large* focal length





Weak perspective vs perspective camera



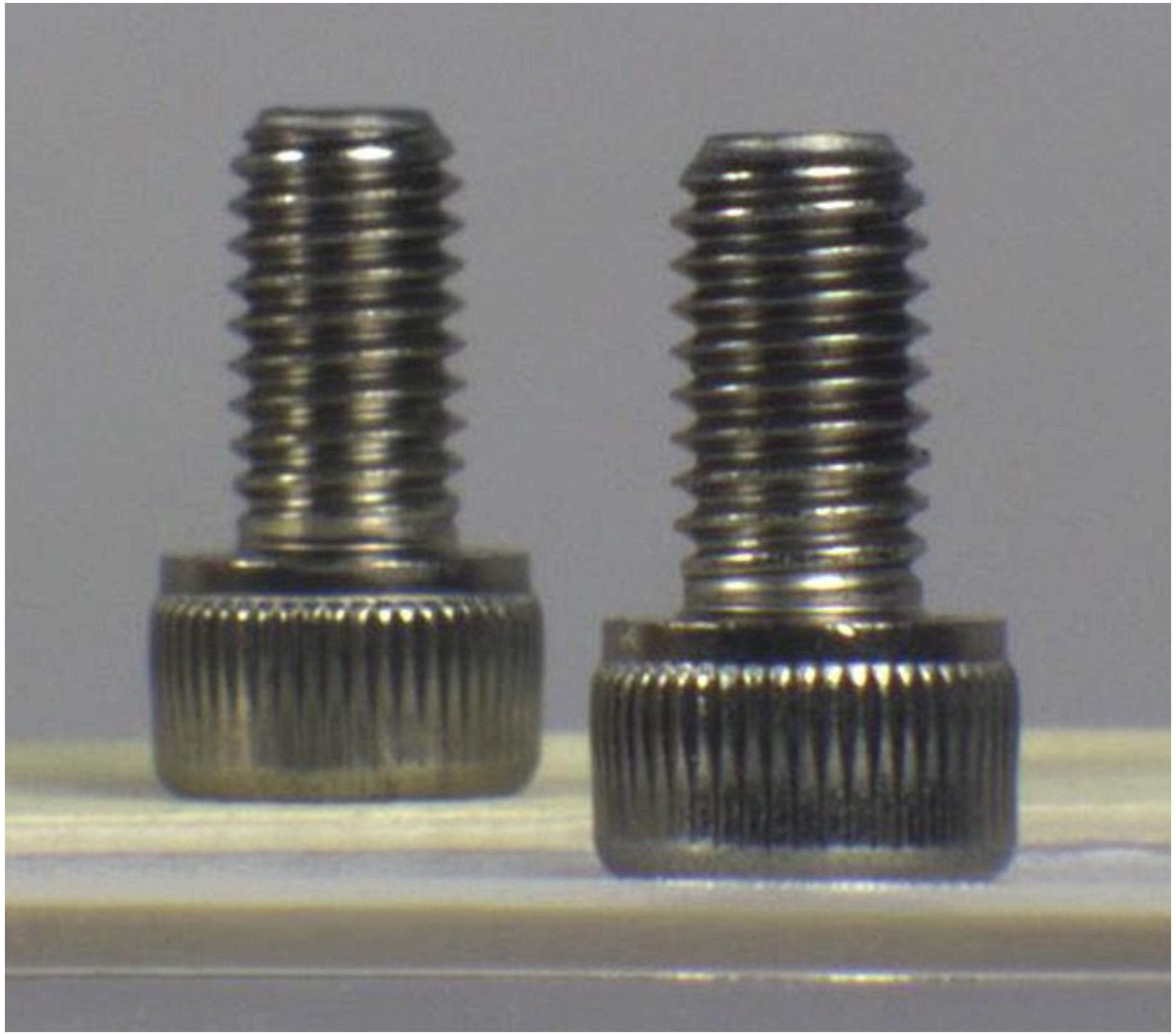
$$[X \quad Y \quad Z]^\top \mapsto [fX/Z_o \quad fY/Z_o]^\top$$

- magnification does not change with depth
- *constant* magnification depending on f and Z_o

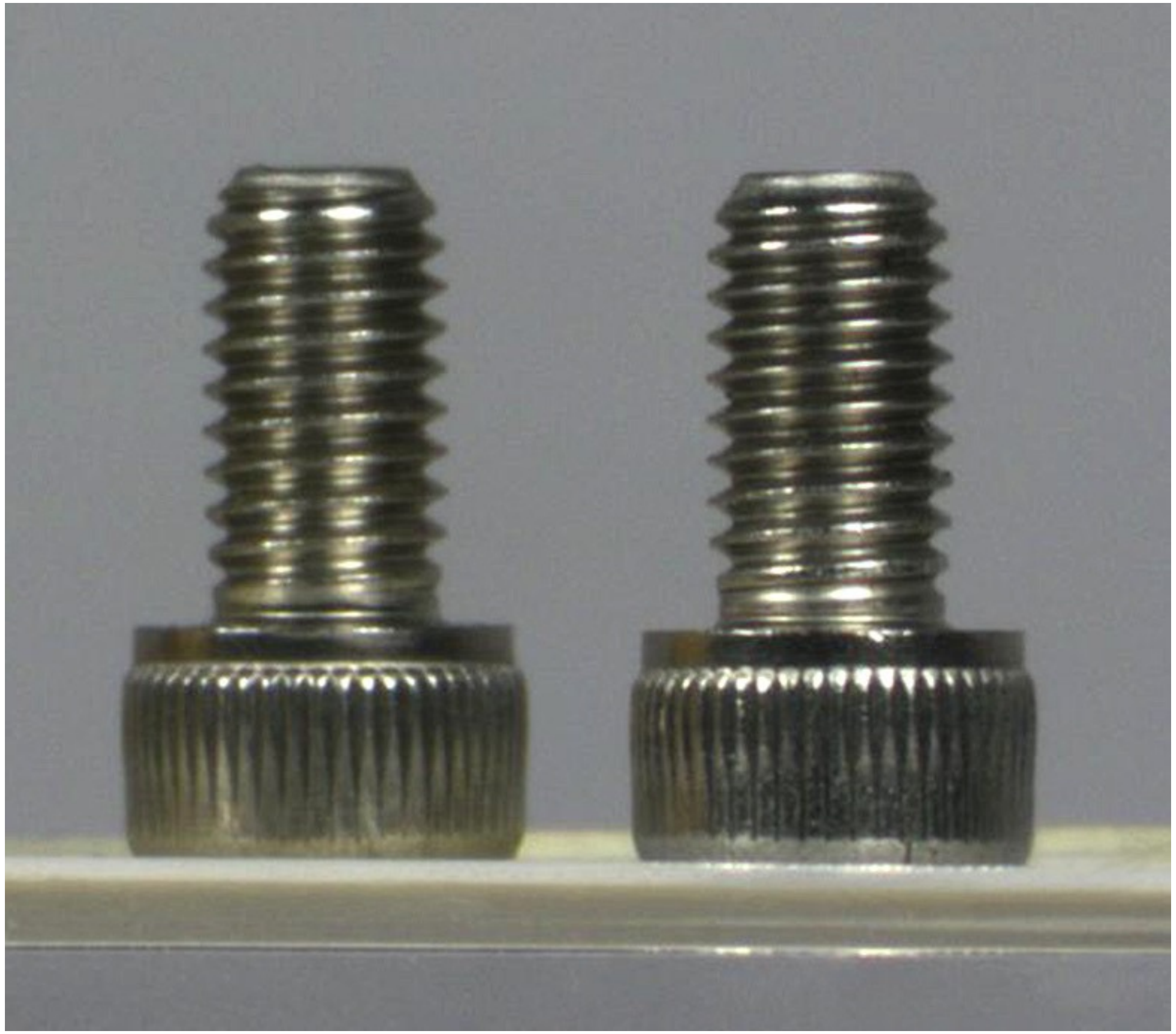
magnification changes with depth

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

Different cameras



perspective camera



weak perspective camera

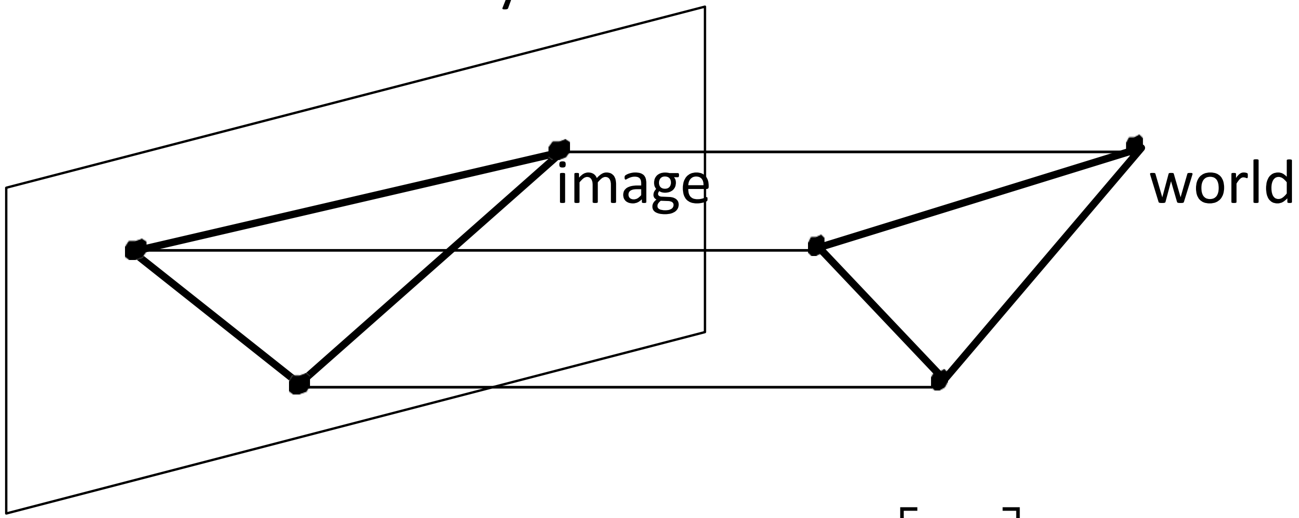
When can we assume a weak perspective camera?

When the scene (or parts of it) is very far away.

Weak perspective projection applies to the mountains.

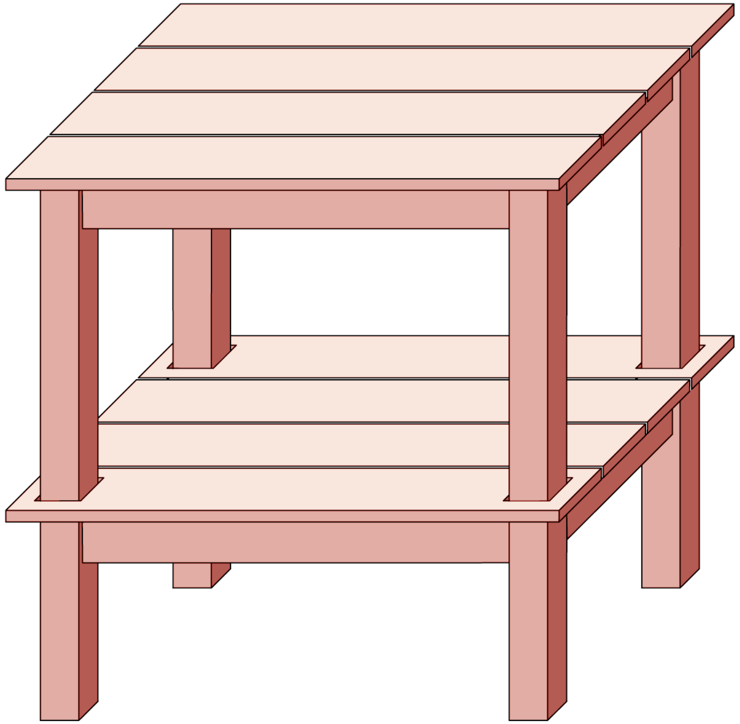
Orthographic camera

- Special case of weak perspective camera where:
- constant magnification is equal to 1.
- there is no shift between camera and image origins.
- the world and camera coordinate systems are the same.

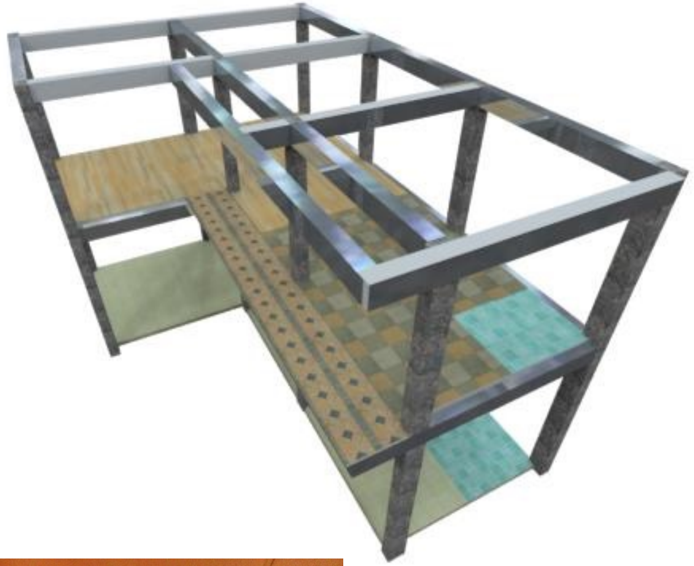


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

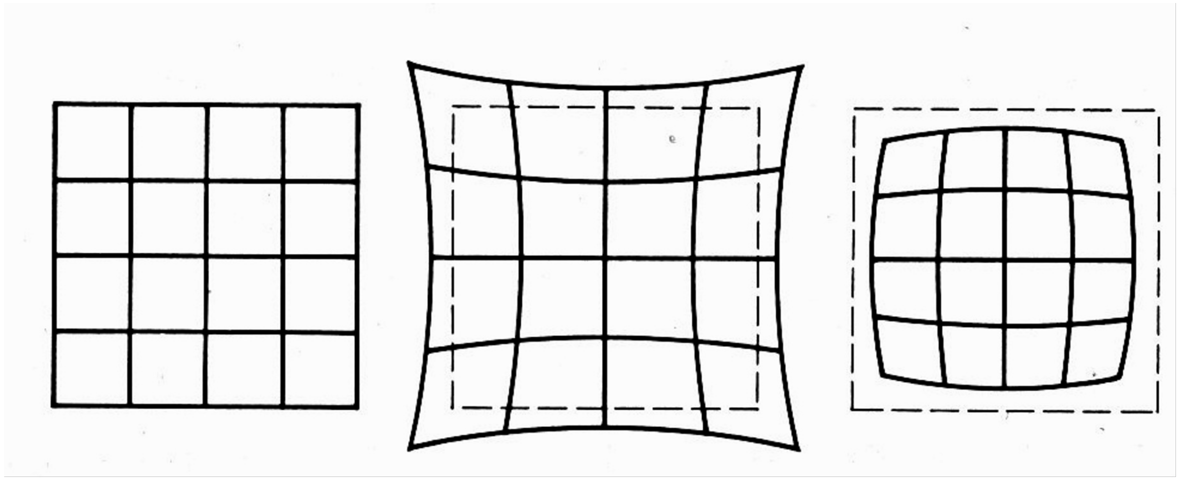
Orthographic projection



Perspective projection



Distortion



No distortion Pin cushion Barrel

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens


©2004 Vincent Bockaert 123di.com


©2004 Vincent Bockaert 123di.com

Correcting radial distortion

from Helmut Dersch

Today’s Class

- Camera Extrinsic
- Camera Intrinsics and pinhole camera model
- Perspective Distortion
- Other Projection models
- How to calibrate camera, i.e. estimate camera parameters (next class)

Slide Credits

- CS5670, Introduction to Computer Vision, **Cornell Tech, by Noah Snavely.**
- CS 194-26/294-26: Intro to Computer Vision and Computational Photography, **UC Berkeley, by Angjoo Kanazawa.**
- CS 16-385: Computer Vision, **CMU, by Matthew O'Toole**

Additional Reading

- Multiview Geometry, Hartley & Zisserman, Chapter 6.1, 6.2.

Related Readings from the past

- Multiview Geometry, Hartley & Zisserman, Chapter 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 3.1, 3.2, 4.1, 4.2