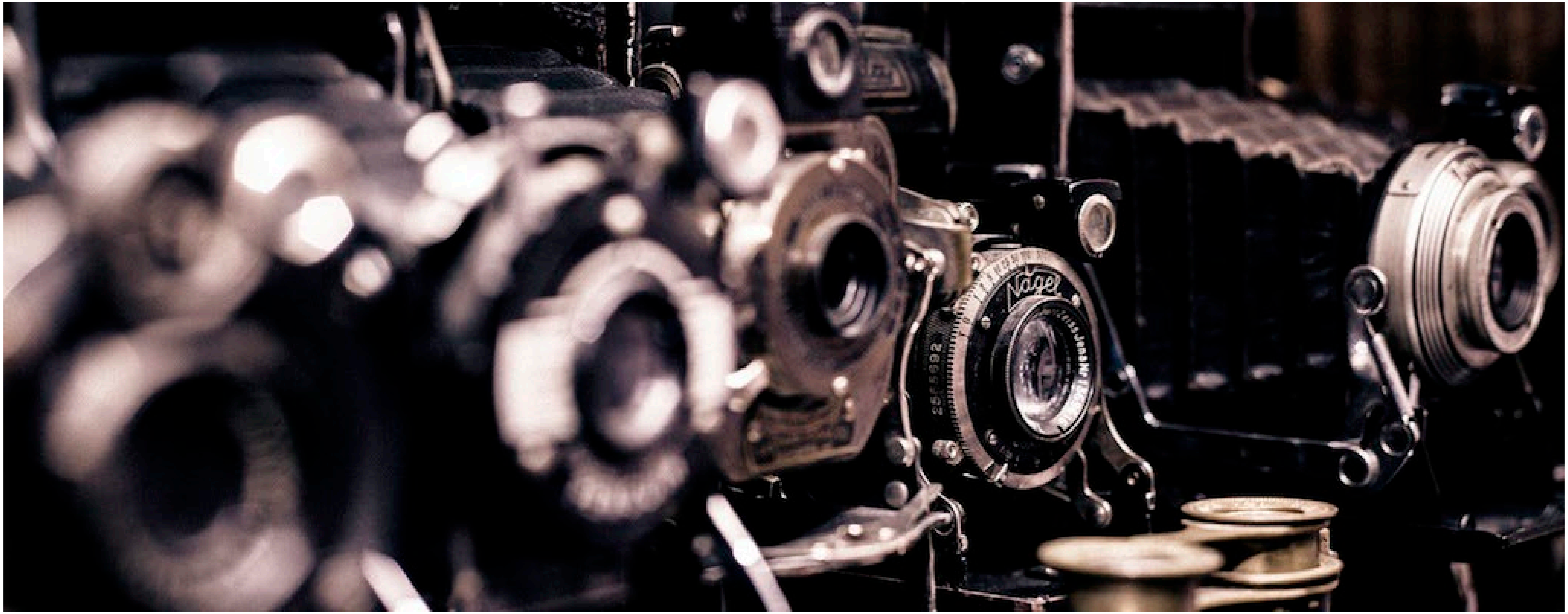
Camera models and calibration



https://cmu-ci-lab.github.io/16705

16-705 Fundamentals of Computer Vision
Spring 2026, Lecture 3

Course announcements

- Anyone not on Slack or Canvas?

Pinhole camera model

Let’s say we have a sensor…

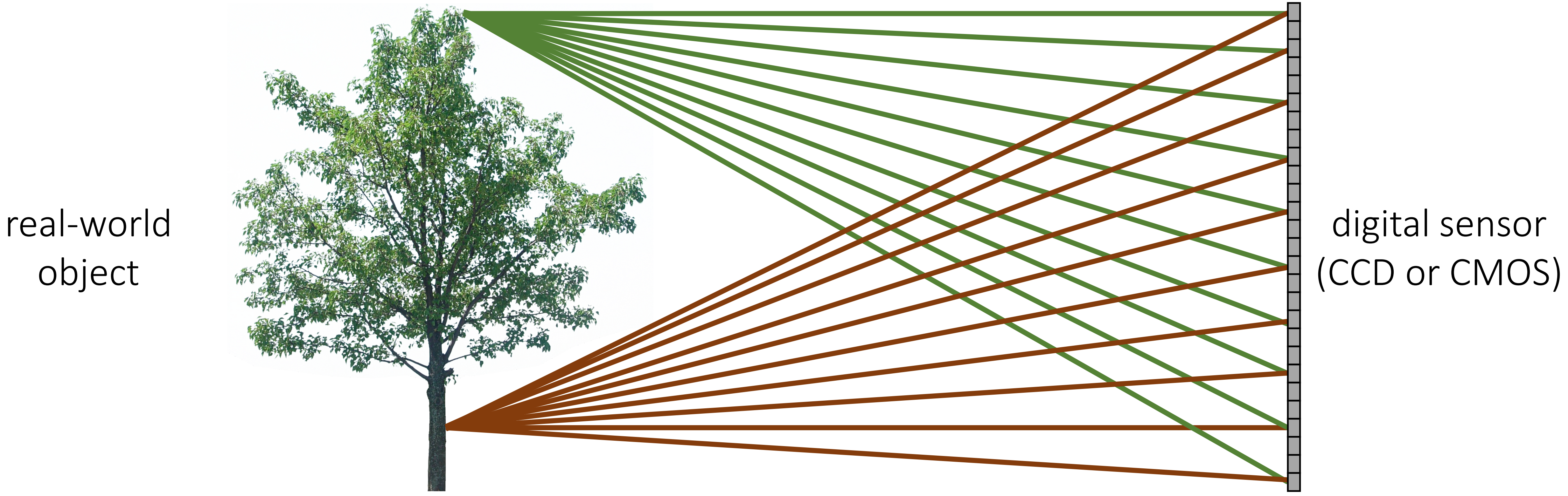
digital sensor
(CCD or CMOS)

… and an object we like to photograph

real-world
object

digital sensor
(CCD or CMOS)

Bare-sensor imaging

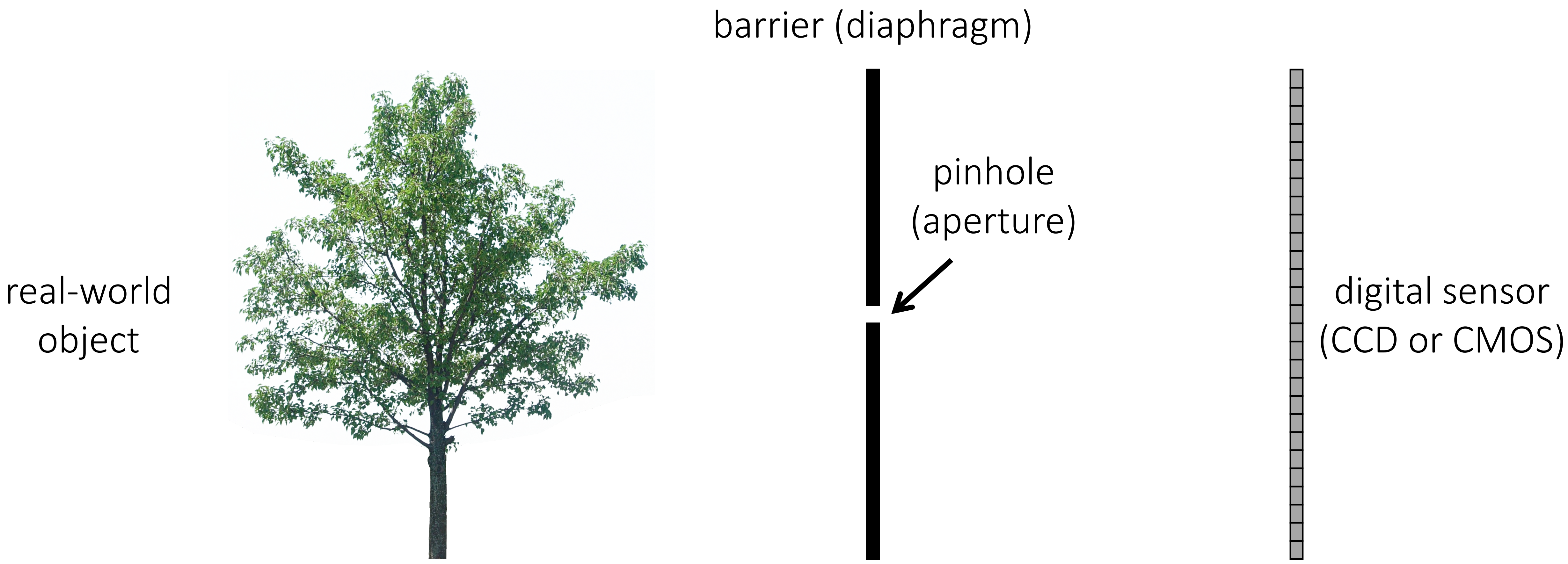


All scene points contribute to all sensor pixels

Bare-sensor imaging

All scene points contribute to all sensor pixels

Let’s add something to this scene

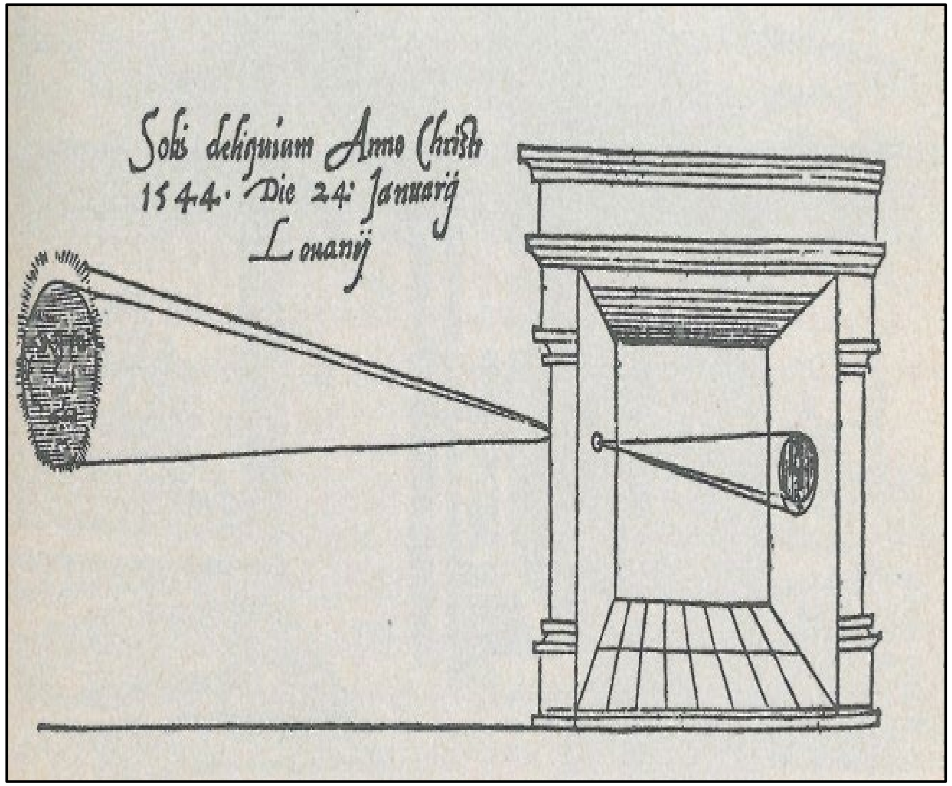


Pinhole camera a.k.a. camera obscura

First mention …

Chinese philosopher Mozi
(470 to 390 BC)

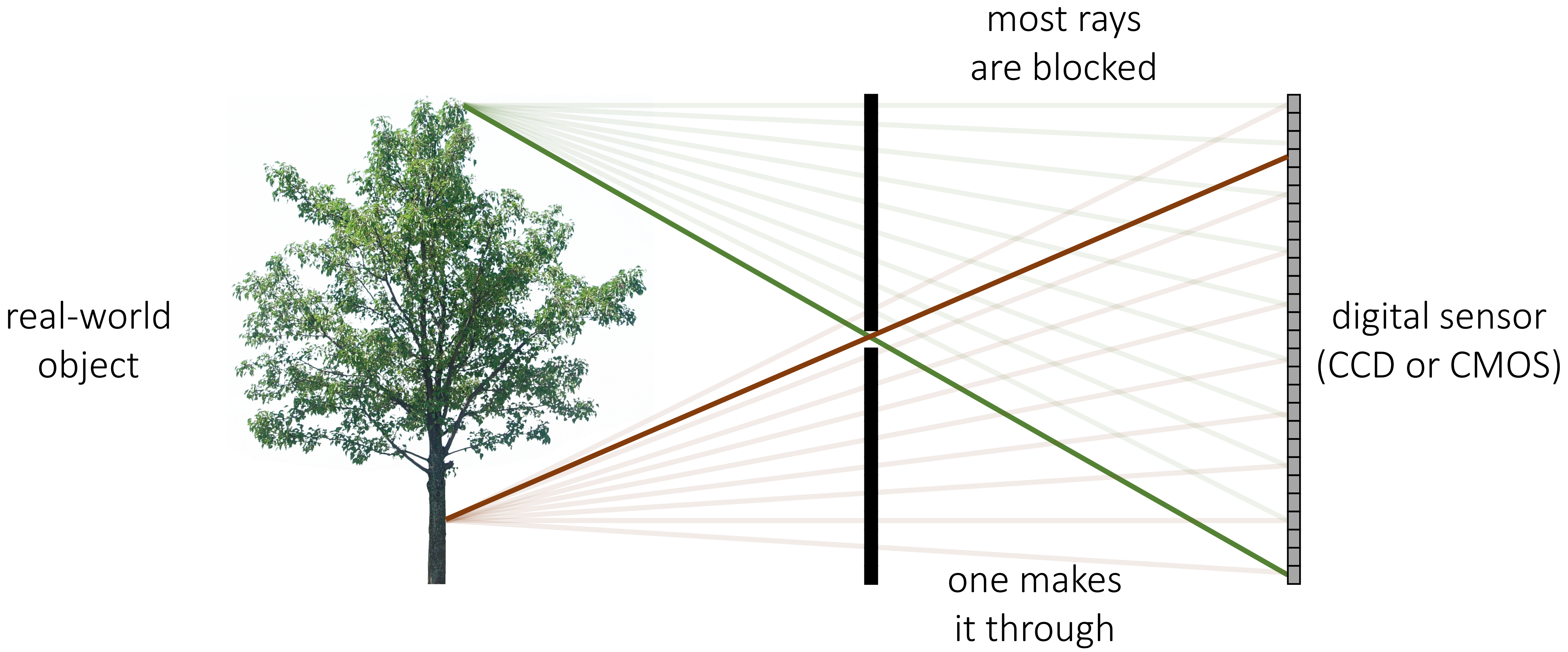
First camera …



Greek philosopher Aristotle
(384 to 322 BC)

Indirect observation of total solar eclipse of August 21, 2017

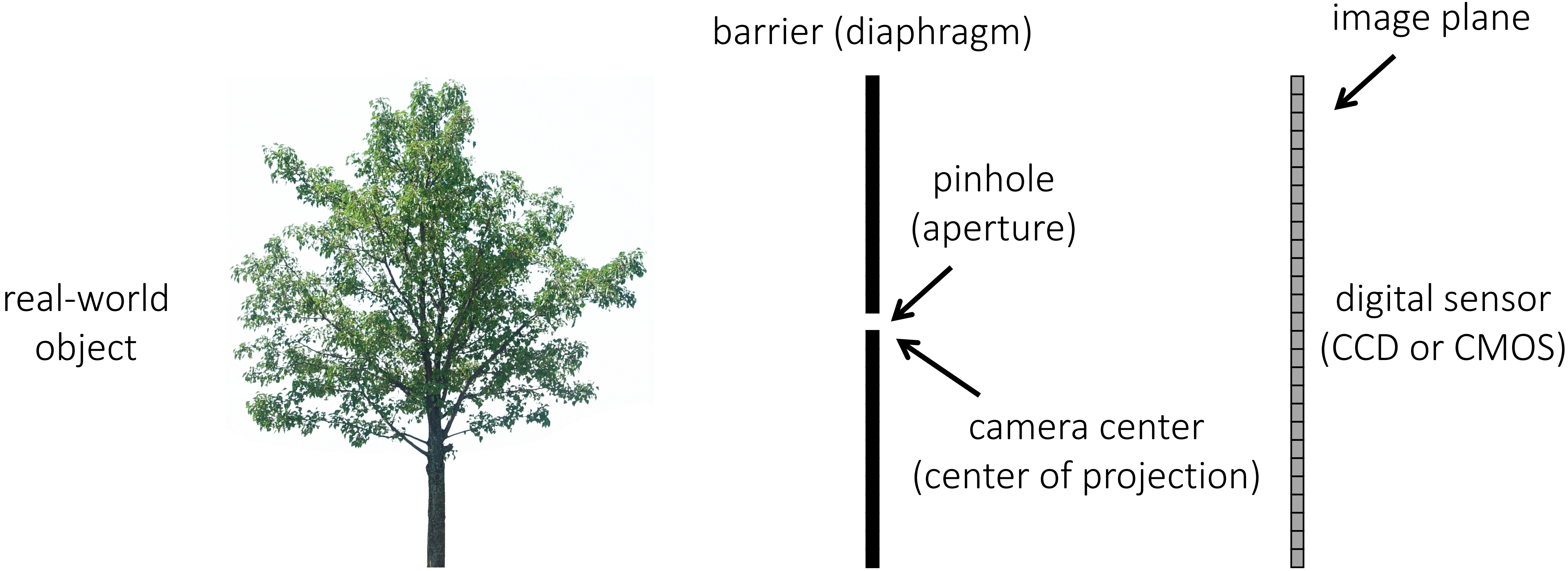
Pinhole imaging



Each scene point contributes to only one sensor pixel

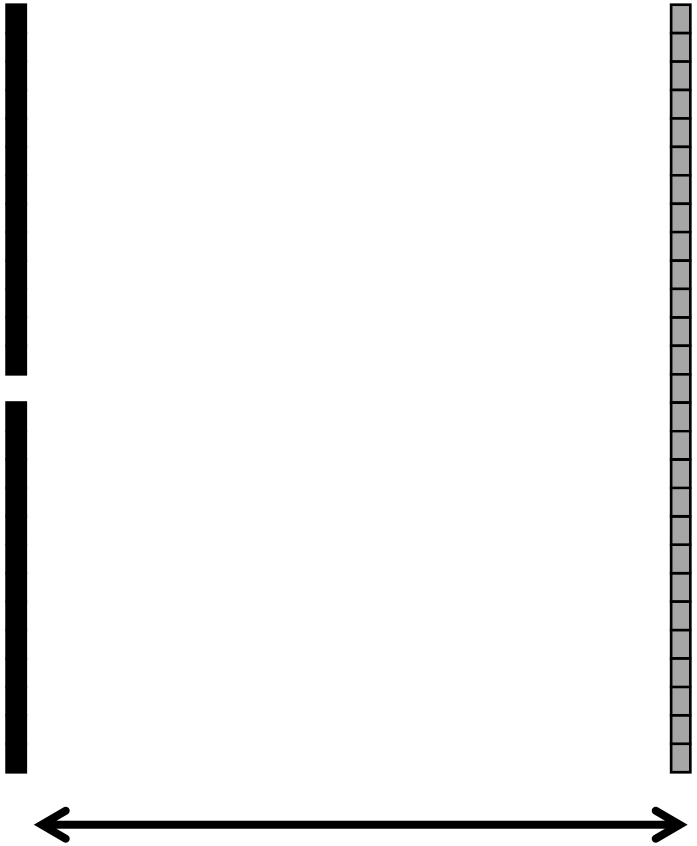
Pinhole imaging

Pinhole camera terms



Focal length

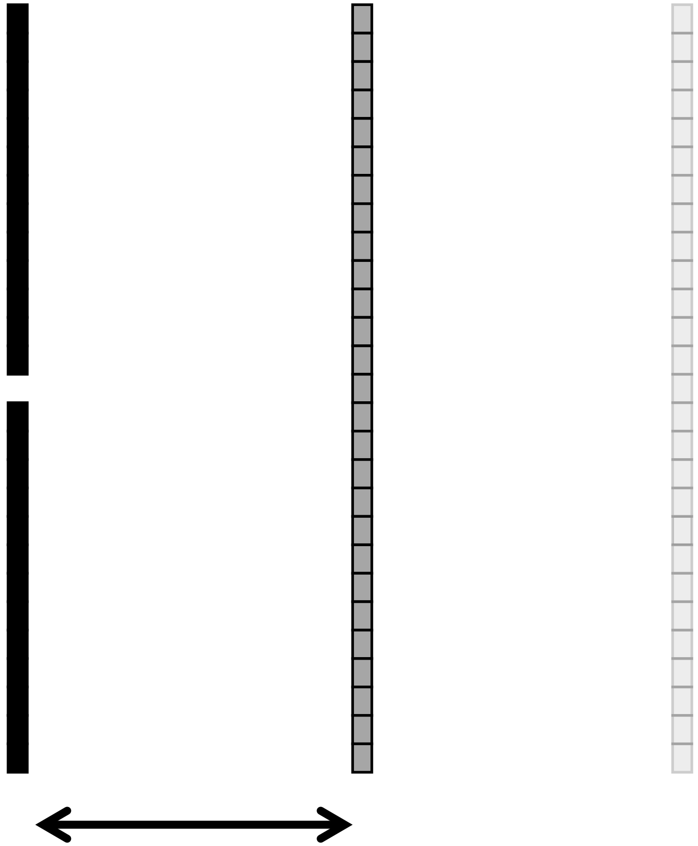
real-world
object



focal length f

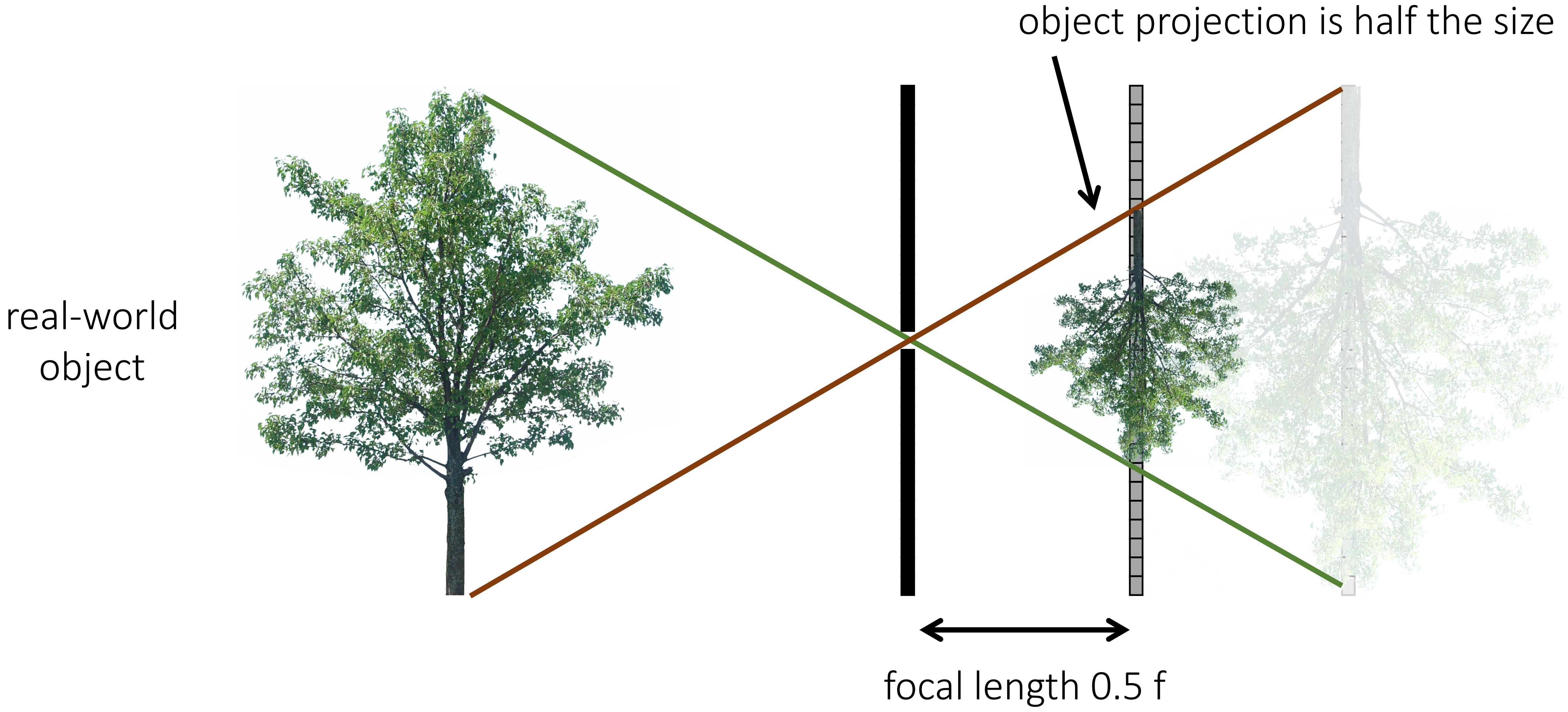
Changing the focal length

real-world
object

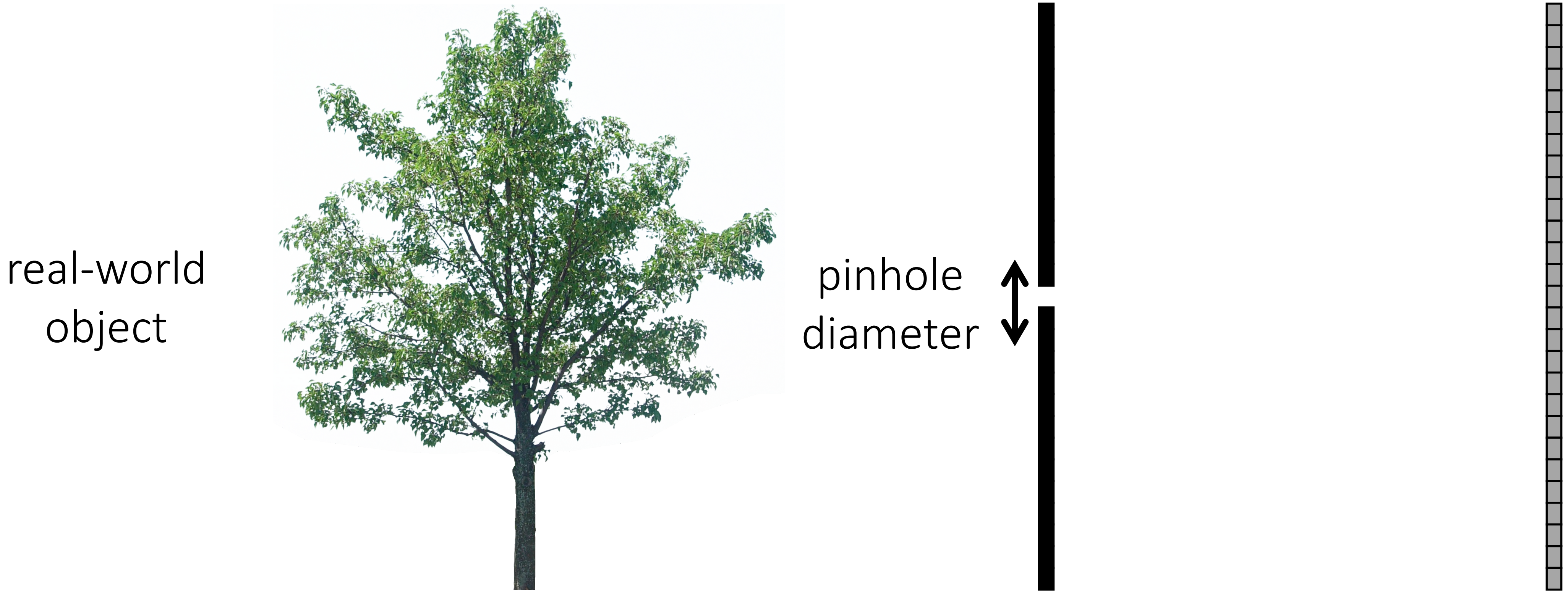


focal length 0.5 f

Changing the focal length



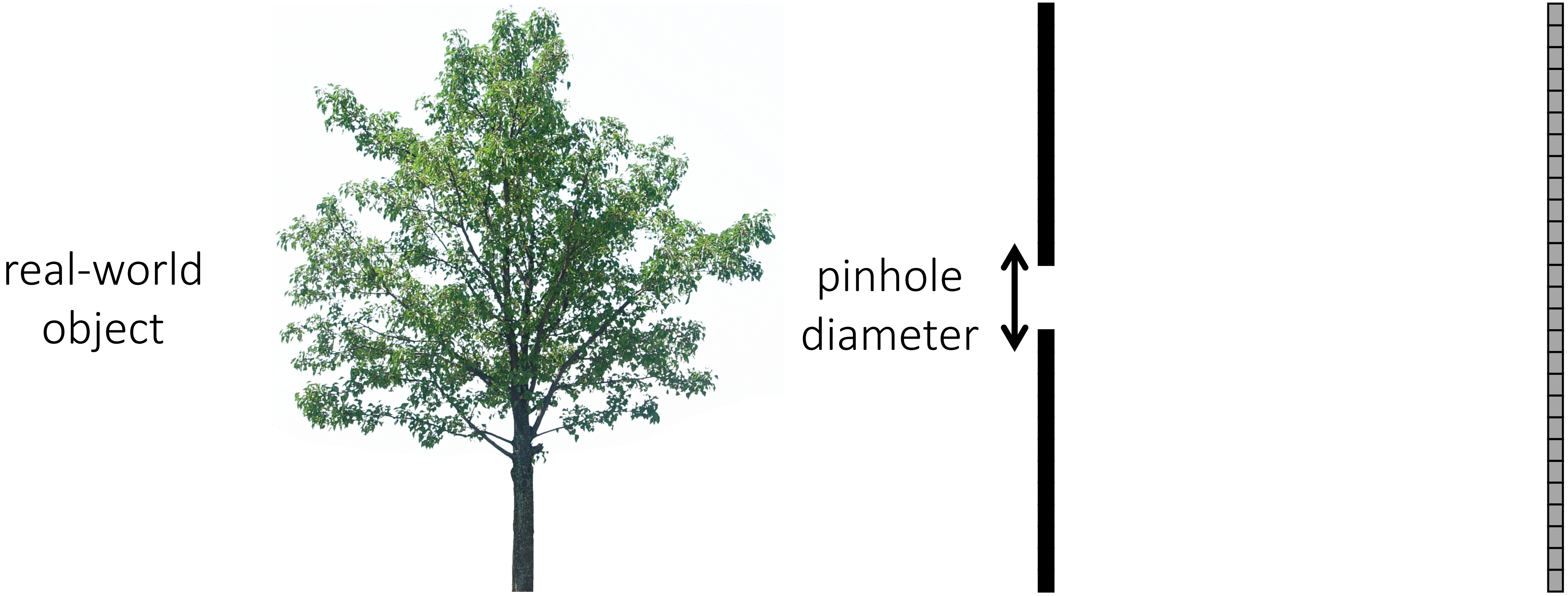
Pinhole size



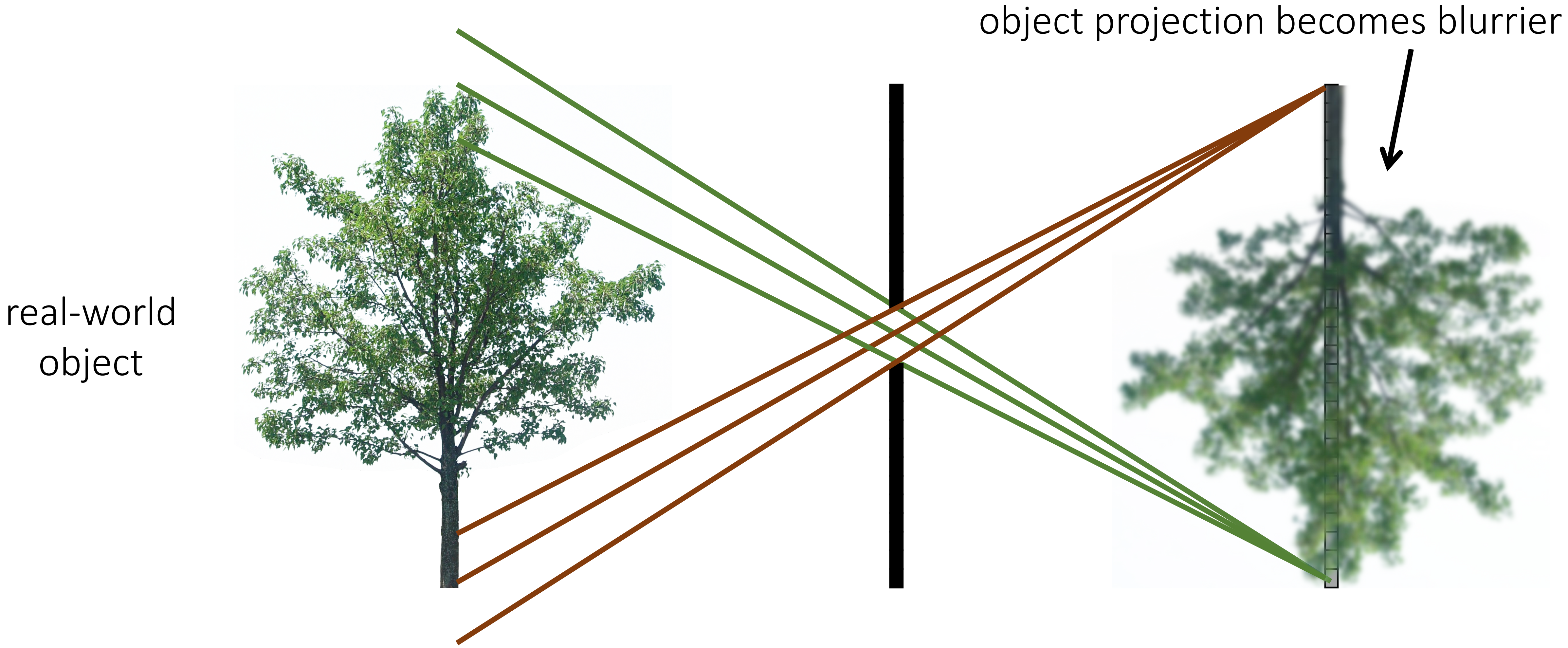
Ideal pinhole has infinitesimally small size

- In practice that is impossible.

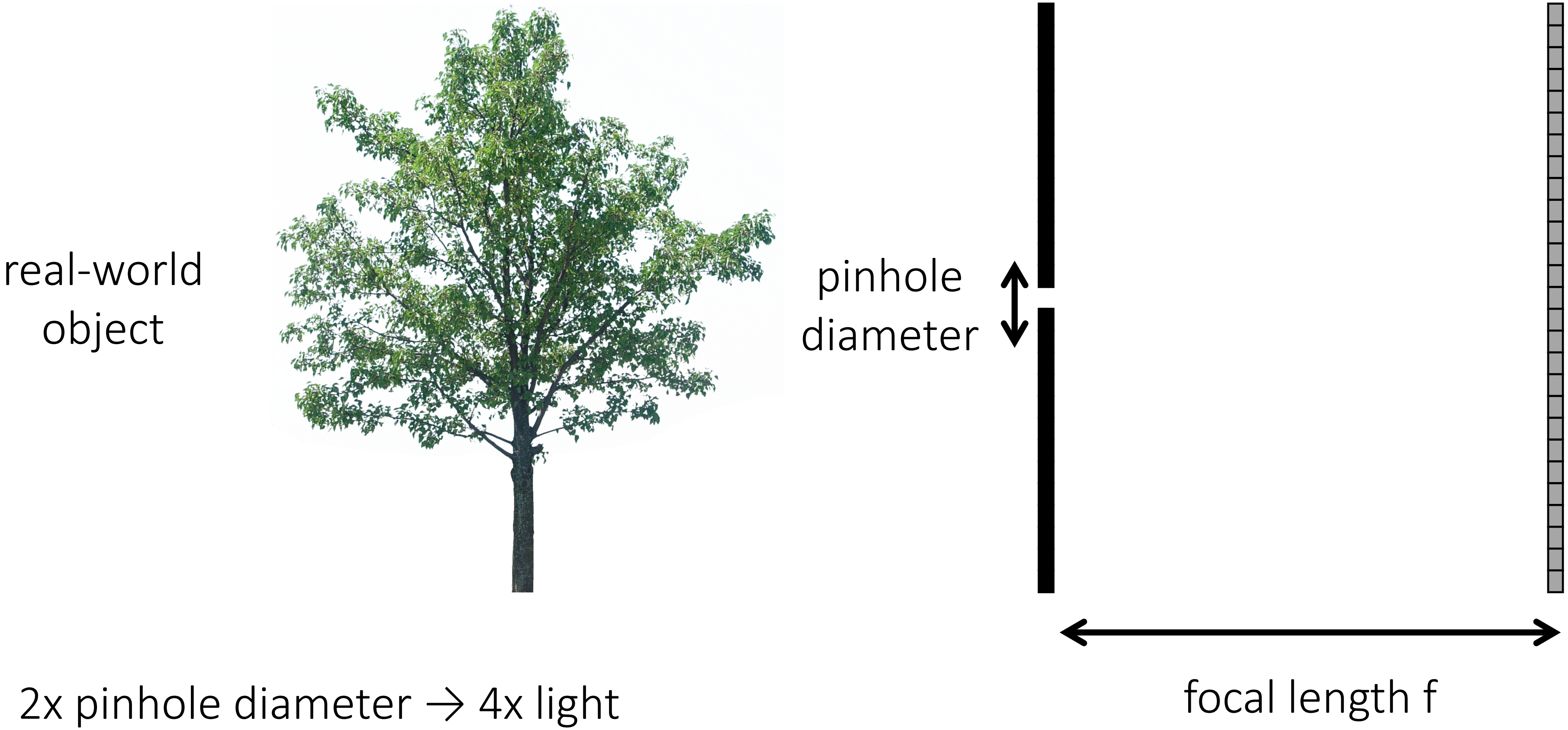
Changing the pinhole size



Changing the pinhole size



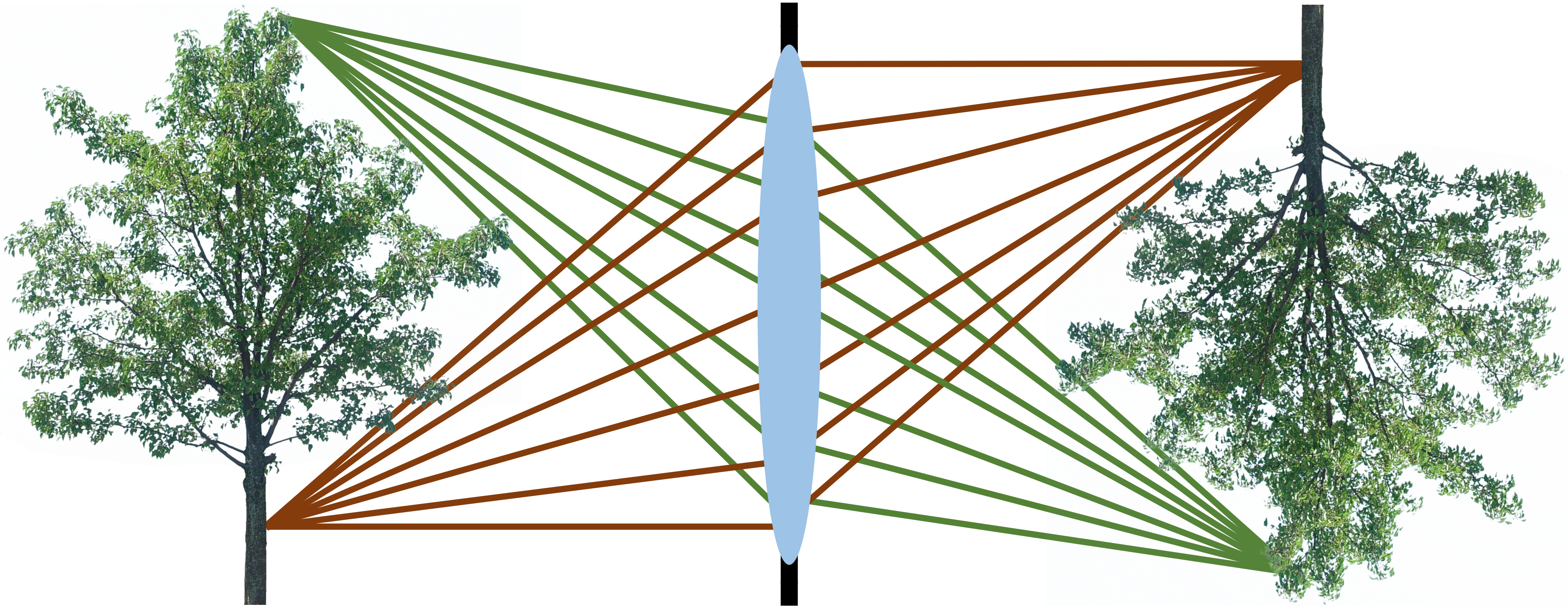
What about light efficiency?



- 2x pinhole diameter → 4x light
- Trade-off between sharpness and light efficiency

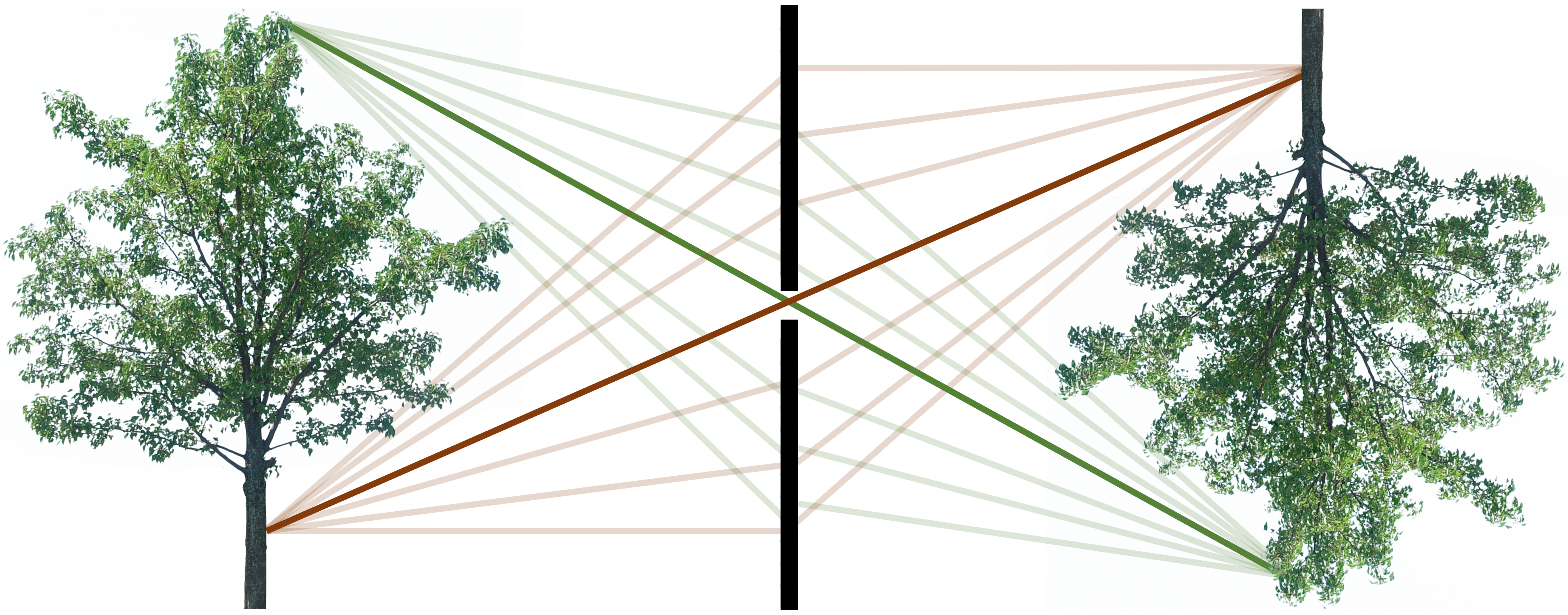
Lenses versus pinholes

The lens camera



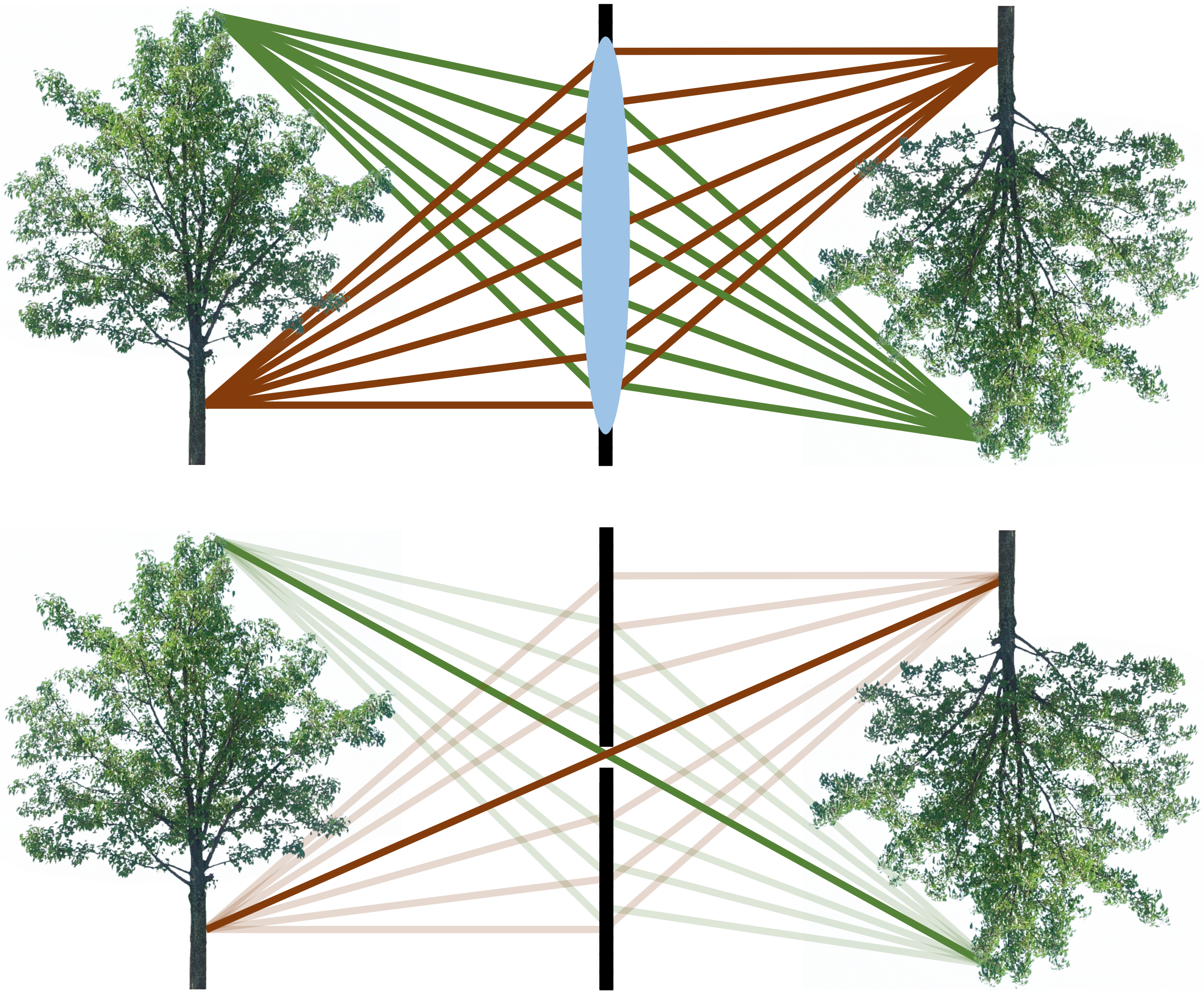
Lenses map “bundles” of rays from points on the scene to the sensor.

The pinhole camera



Central rays propagate in the same way for both models!

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.

Camera matrix

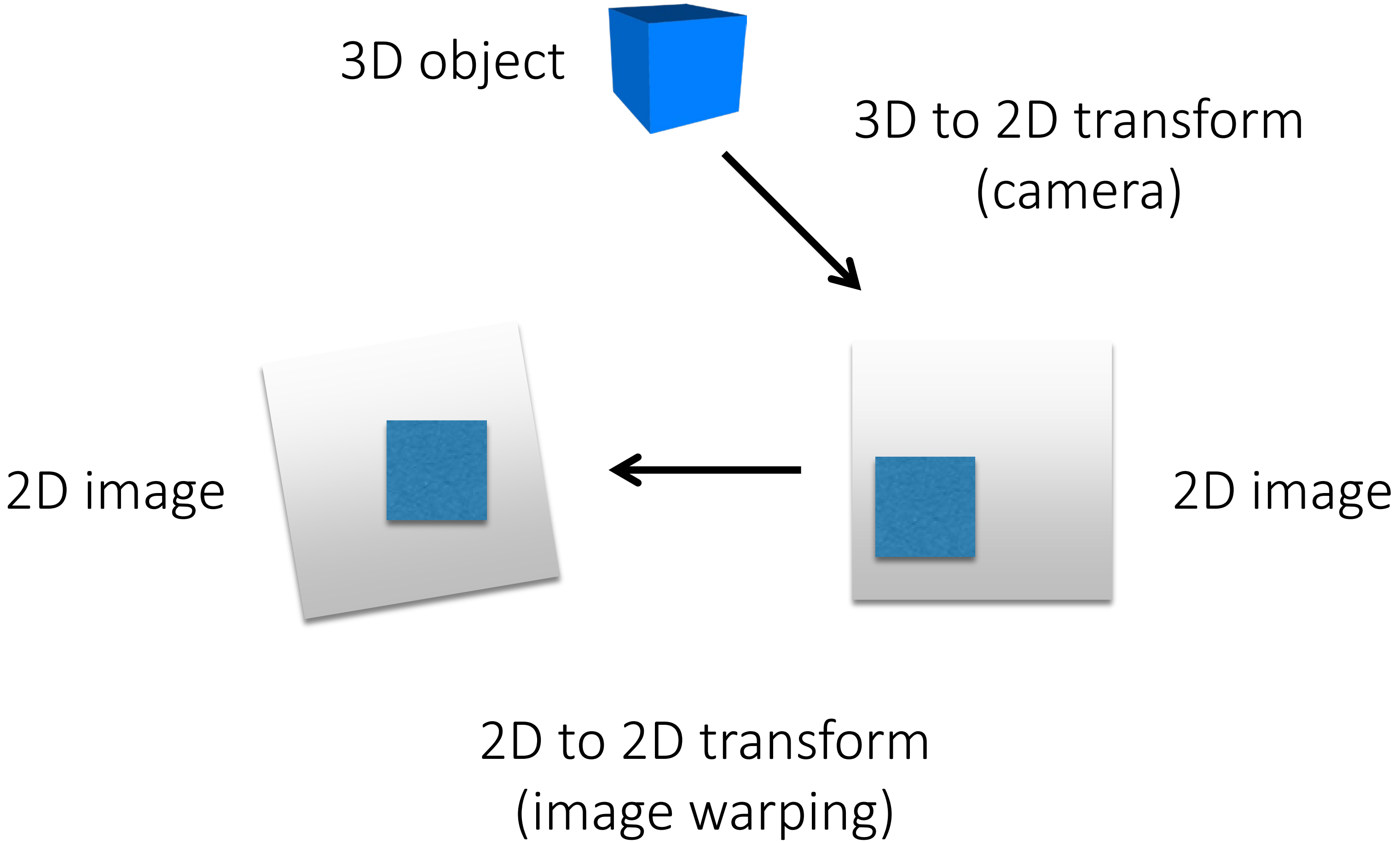
The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image



The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image

homogeneous coordinates

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

2D image point

camera matrix

3D world point

What does this transformation look like?

2D homogeneous coordinates

heterogeneous coordinates

homogeneous coordinates

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

add a 1 here

- Represent 2D point with a 3D vector

2D homogeneous coordinates

heterogeneous coordinates | homogeneous coordinates

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} ax \\ ay \\ a \end{bmatrix}$$

- Represent 2D point with a 3D vector
- 3D vectors are only defined up to scale

2D homogeneous coordinates

Conversion:

- heterogeneous → homogeneous

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- homogeneous → heterogeneous

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} x/z \\ y/z \end{bmatrix}$$

Scale invariance:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = a \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Special points:

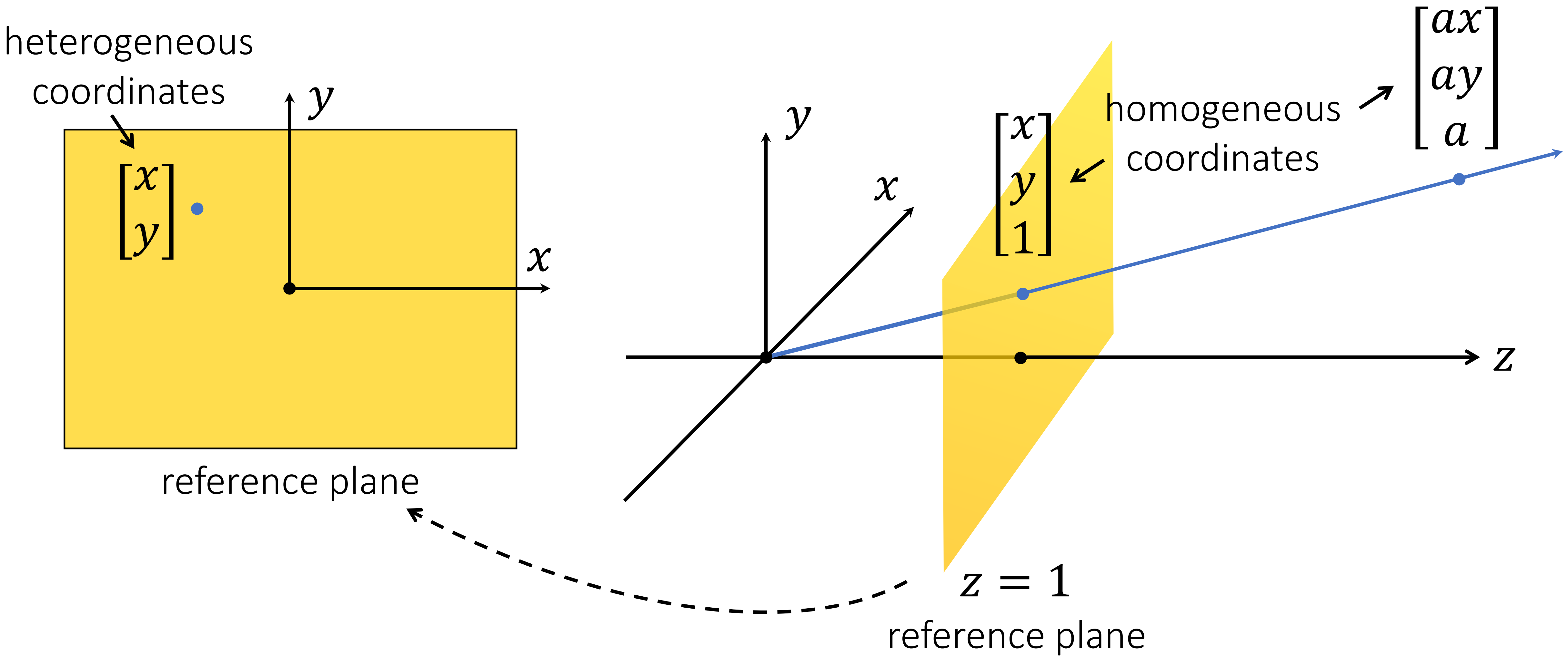
- point at infinity

$$\begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

- undefined

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

2D projective geometry



Through the scale invariance property, homogeneous coordinates map all points on a line passing through the origin to the point where this line intersects the reference plane.

3D homogeneous coordinates

heterogeneous coordinates | homogeneous coordinates

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} aX \\ aY \\ aZ \\ a \end{bmatrix}$$

- Represent 3D point with a 4D vector
- 4D vectors are only defined up to scale

Notation

	heterogeneous coordinates	homogeneous coordinates
2D coordinates	2D vector $\tilde{\boldsymbol{x}} = \begin{bmatrix} x \\ y \end{bmatrix}$	3D vector $\boldsymbol{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
3D coordinates	3D vector $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$	4D vector $\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image

homogeneous coordinates

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

2D image point

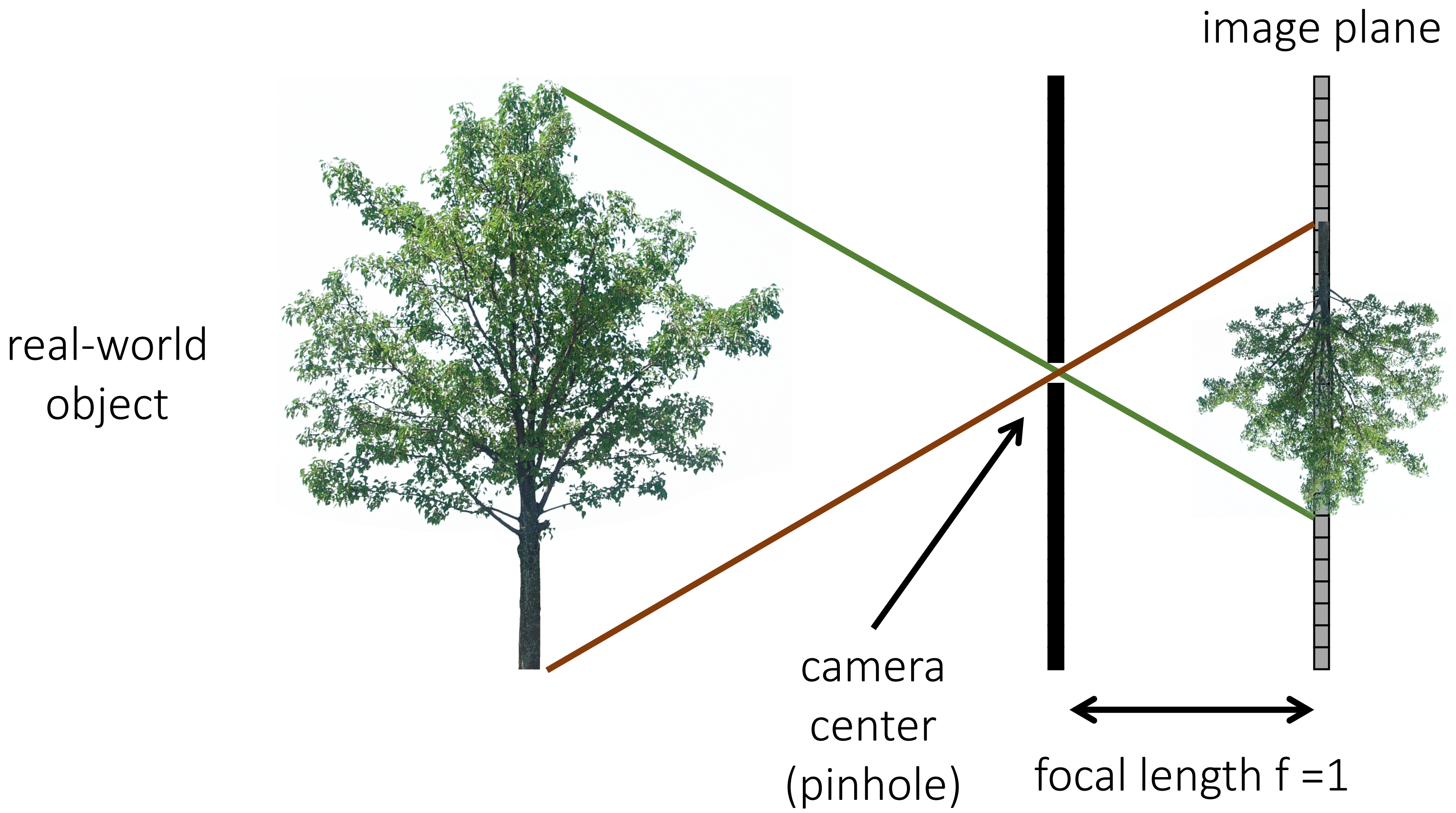
camera matrix

3D world point

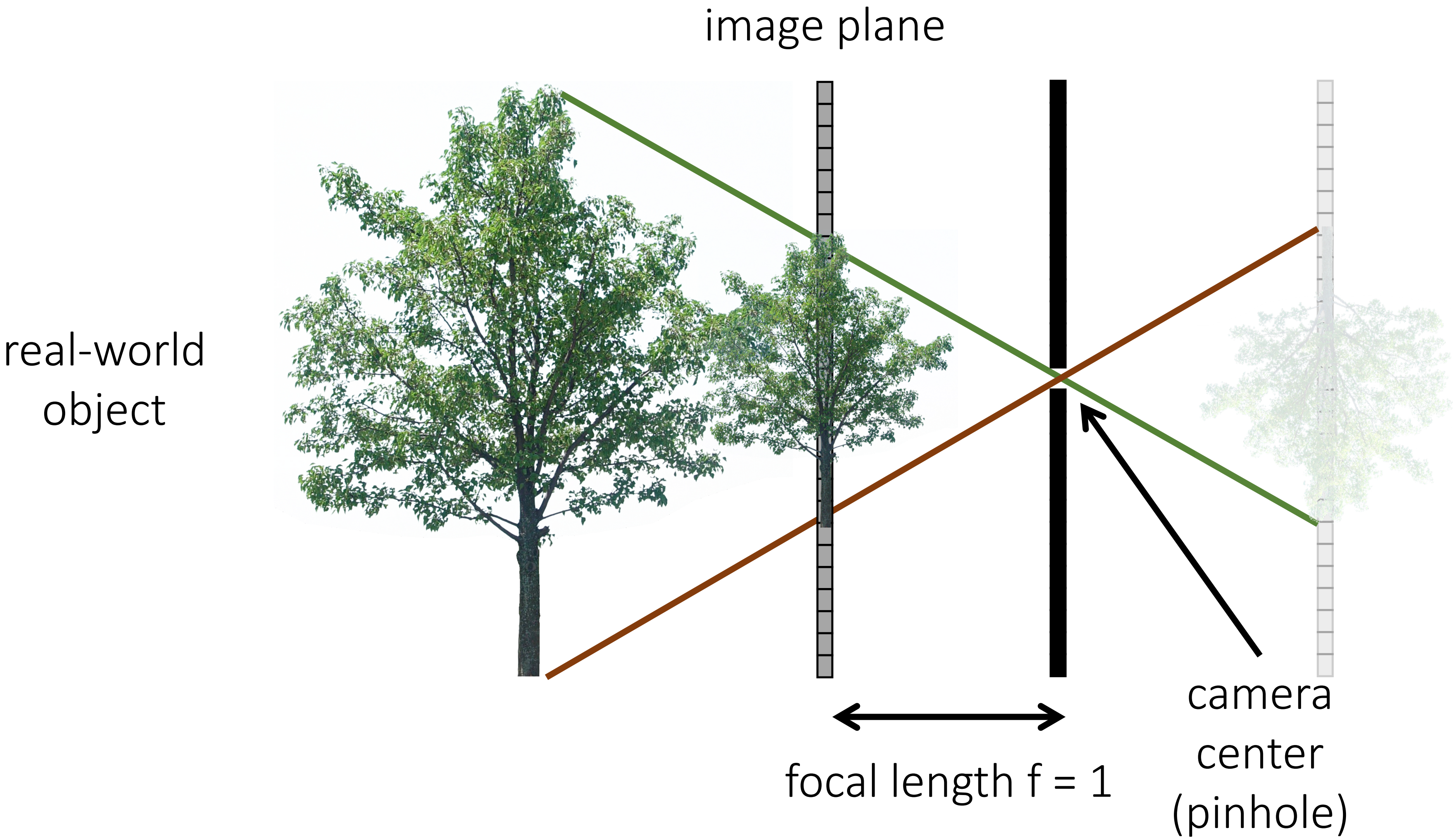
What does this transformation look like?

Perspective projection

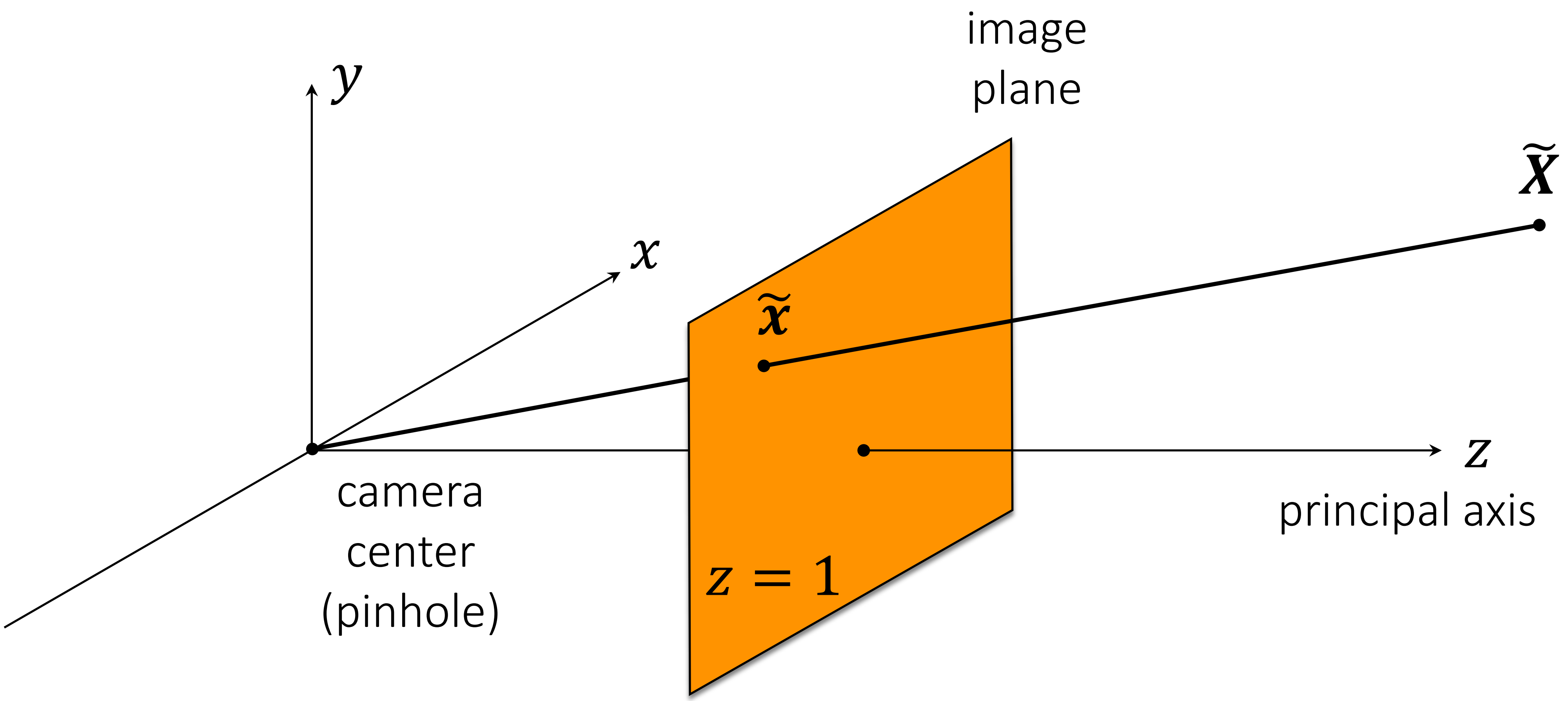
The pinhole camera



The (rearranged) pinhole camera

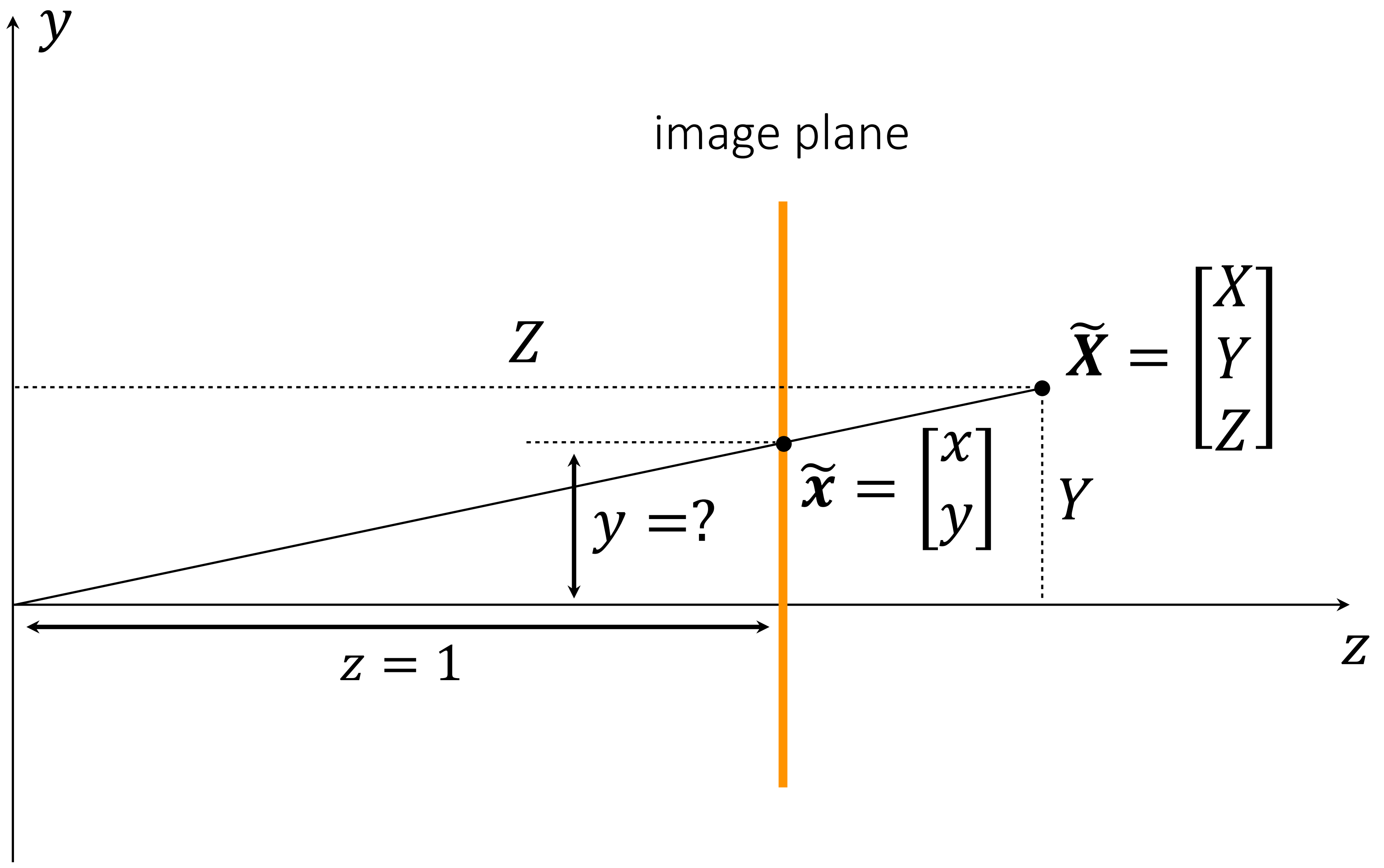


The (rearranged) pinhole camera

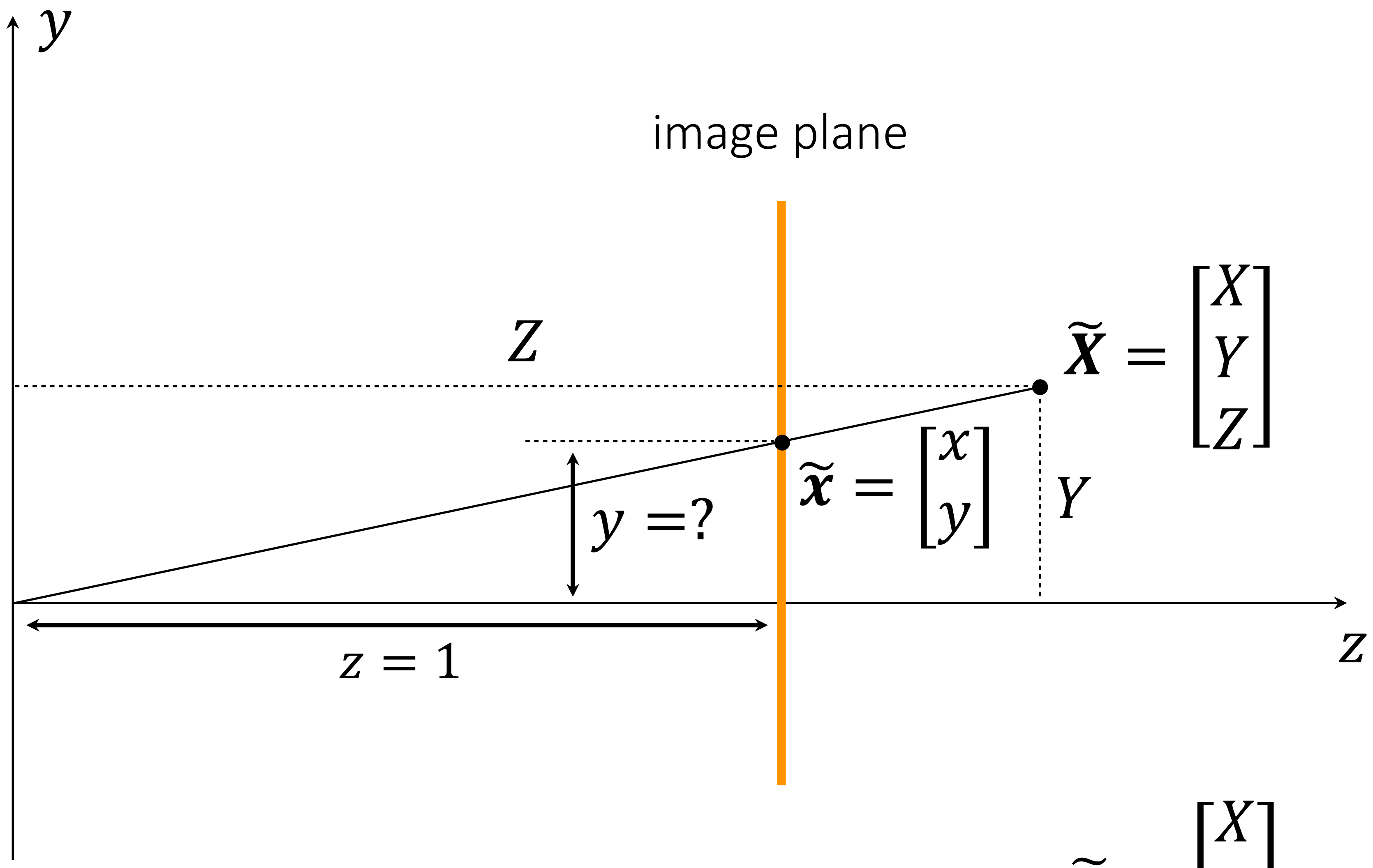


What is the equation for image coordinate $\tilde{\boldsymbol{x}}$ in terms of $\tilde{\boldsymbol{X}}$?

The 2D view of the (rearranged) pinhole camera

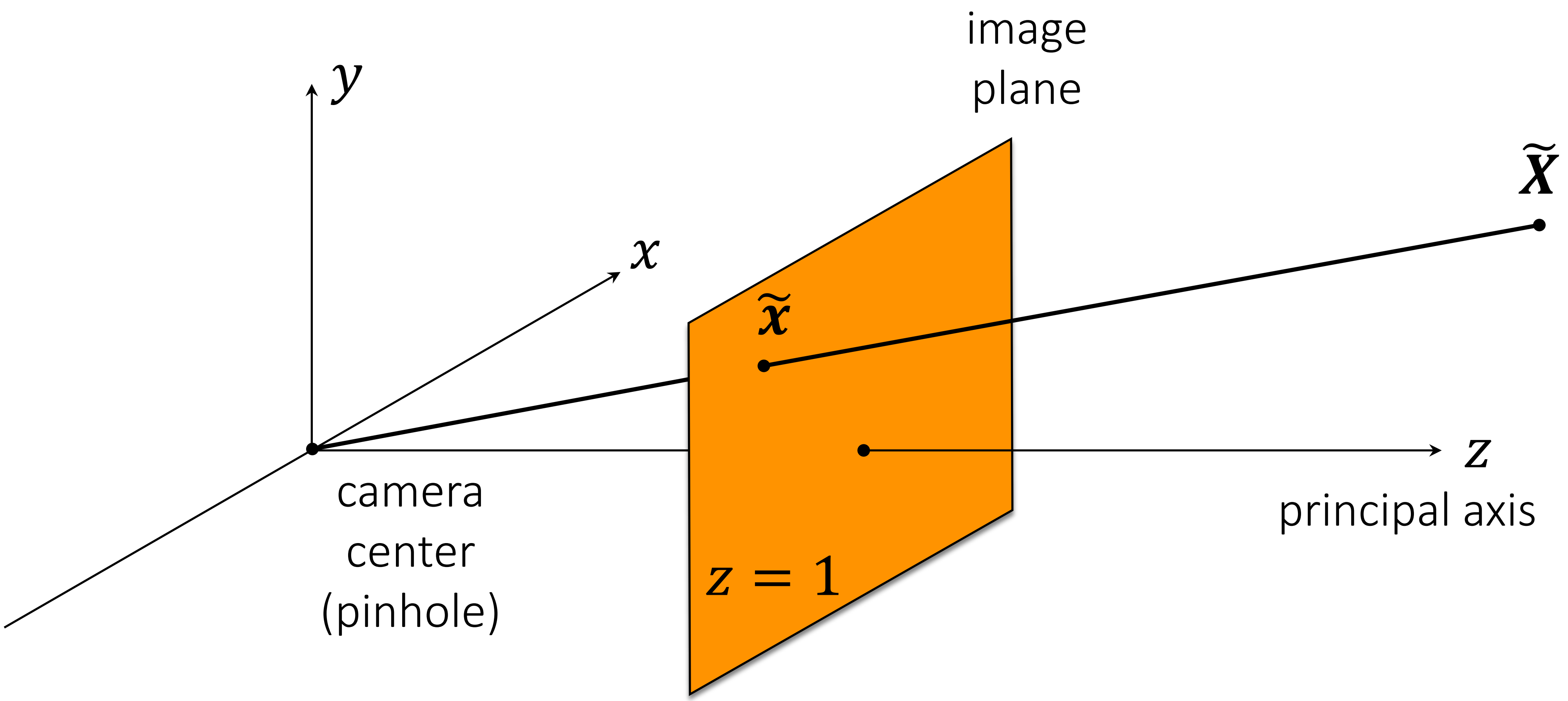


The 2D view of the (rearranged) pinhole camera



Using similar triangles: $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

The (rearranged) pinhole camera



What is the camera matrix $\boldsymbol{P}$ for a pinhole camera?

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

The pinhole camera matrix

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \widetilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = ?$$

The pinhole camera matrix

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \widetilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

General camera model in *homogeneous coordinates*:

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

What does the pinhole camera projection look like?

$$\boldsymbol{P} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

The pinhole camera matrix

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \widetilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

General camera model in *homogeneous coordinates*:

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

What does the pinhole camera projection look like?

The perspective projection matrix

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

The pinhole camera matrix

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \widetilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

General camera model in *homogeneous coordinates*:

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

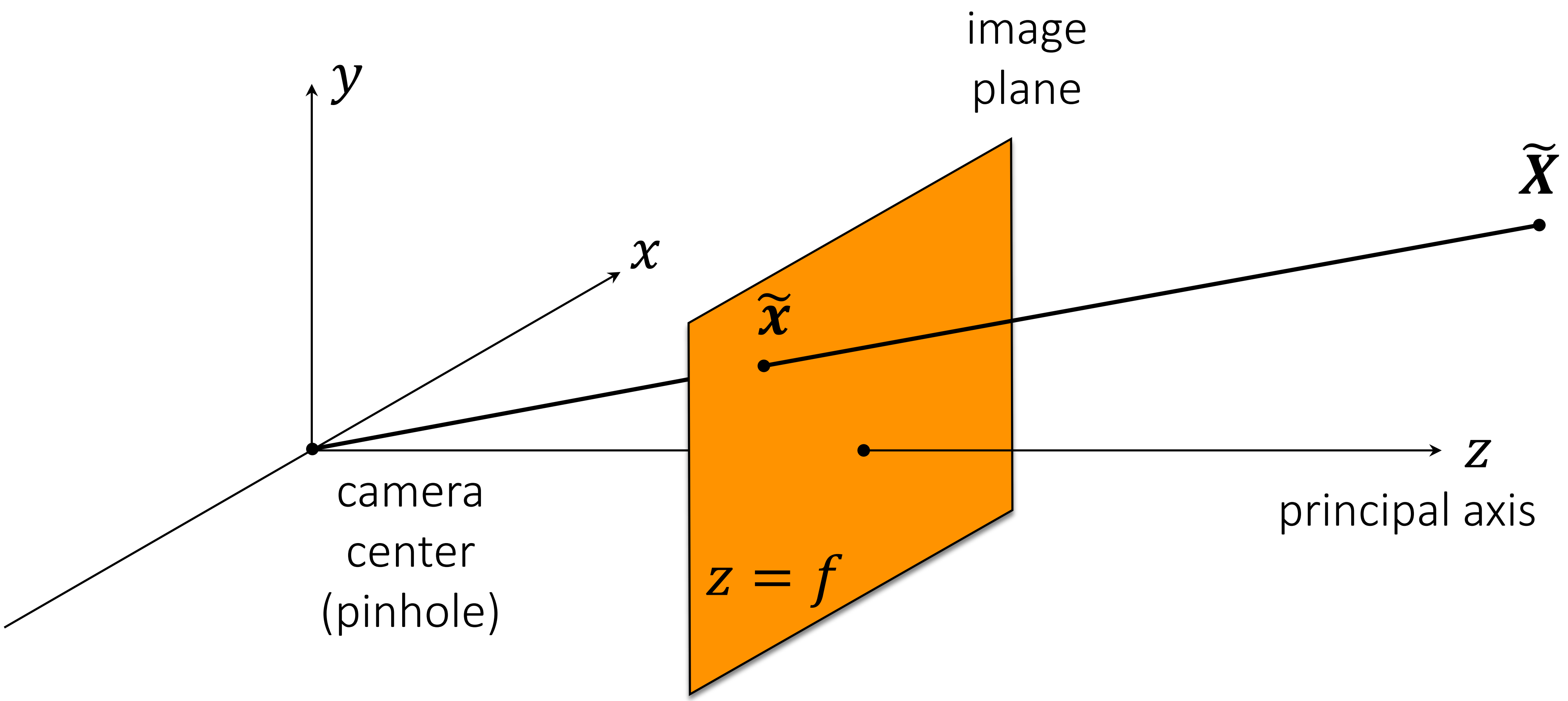
What does the pinhole camera projection look like?

The *perspective projection matrix*

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [\boldsymbol{I} \mid \boldsymbol{0}]$$

alternative way to write the same thing

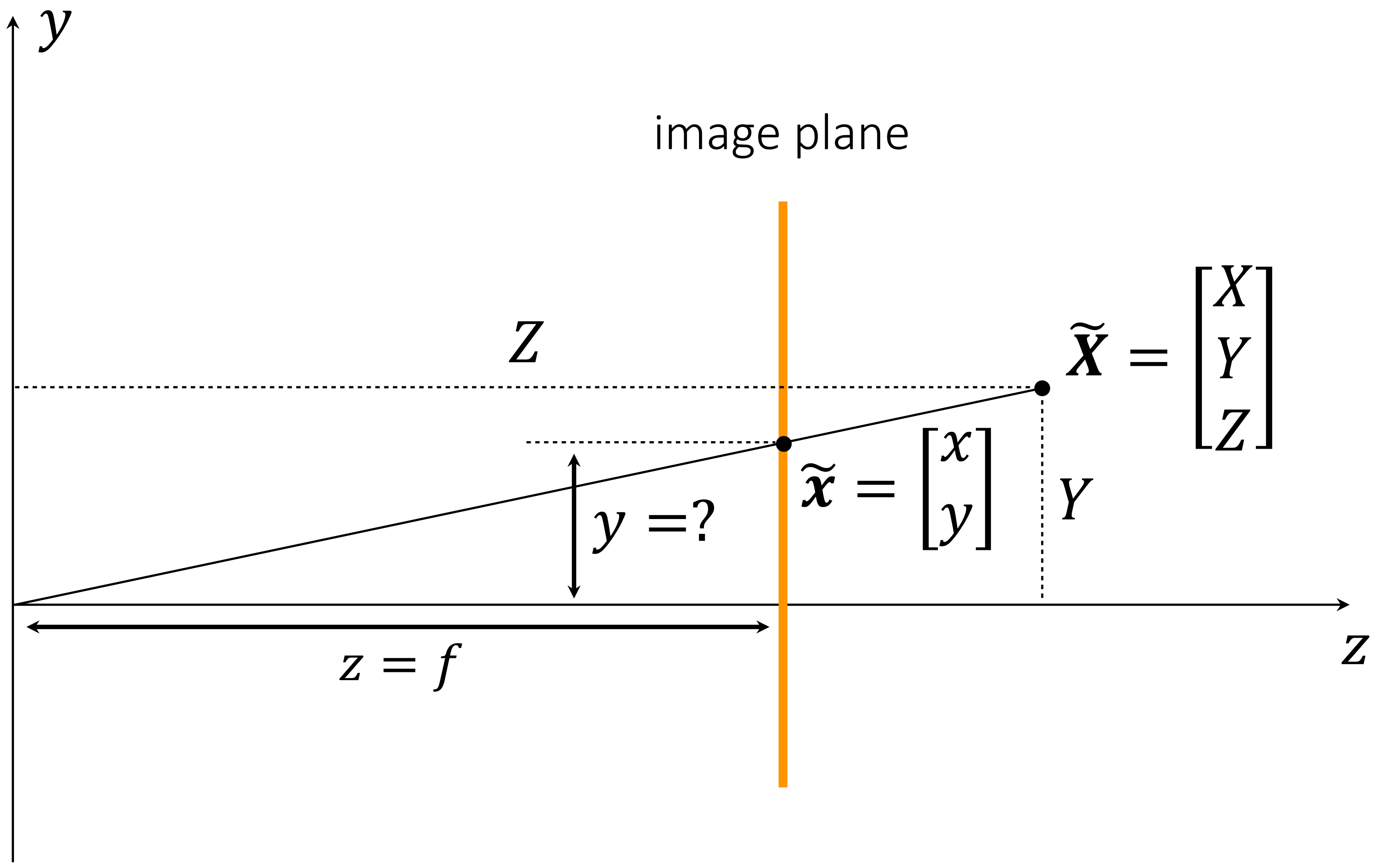
More general case: arbitrary focal length



What is the camera matrix $\boldsymbol{P}$ for a pinhole camera?

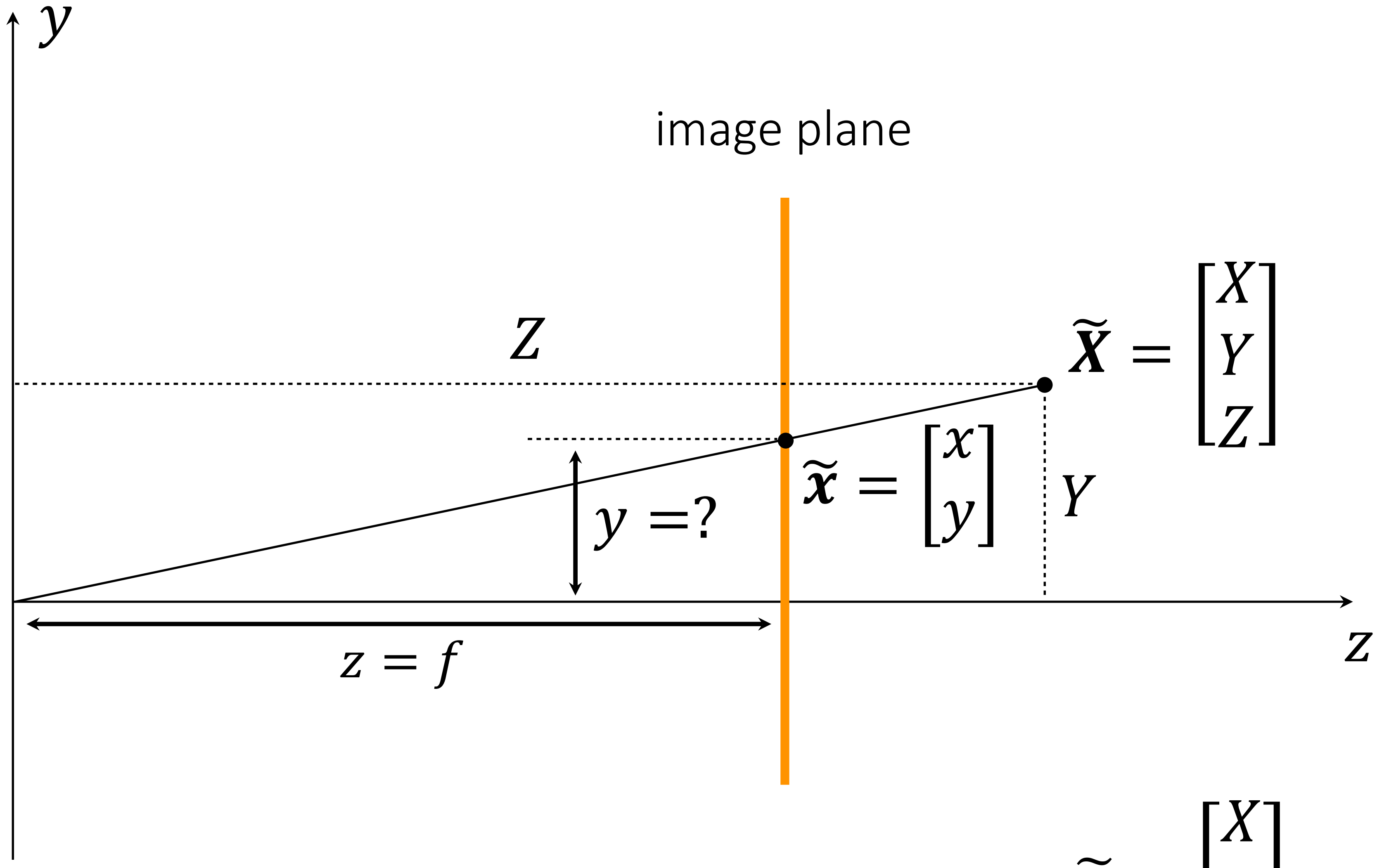
$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X}$$

More general (2D) case: arbitrary focal length



What is the equation for image coordinate $\tilde{\boldsymbol{x}}$ in terms of $\tilde{\boldsymbol{X}}$?

More general (2D) case: arbitrary focal length



Using similar triangles: $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix}$

The pinhole camera matrix for arbitrary focal length

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \widetilde{\boldsymbol{x}} = \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}$$

General camera model in *homogeneous coordinates*:

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

What does the pinhole camera projection look like?

$$\boldsymbol{P} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

The pinhole camera matrix for arbitrary focal length

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}$$

General camera model in *homogeneous coordinates*:

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

What does the pinhole camera projection look like?

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

The pinhole camera matrix for arbitrary focal length

Camera projection relationship expressed:

- in *heterogeneous coordinates*

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \widetilde{\boldsymbol{x}} = \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix}$$

- in *homogeneous coordinates*

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}$$

General camera model in *homogeneous coordinates*:

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

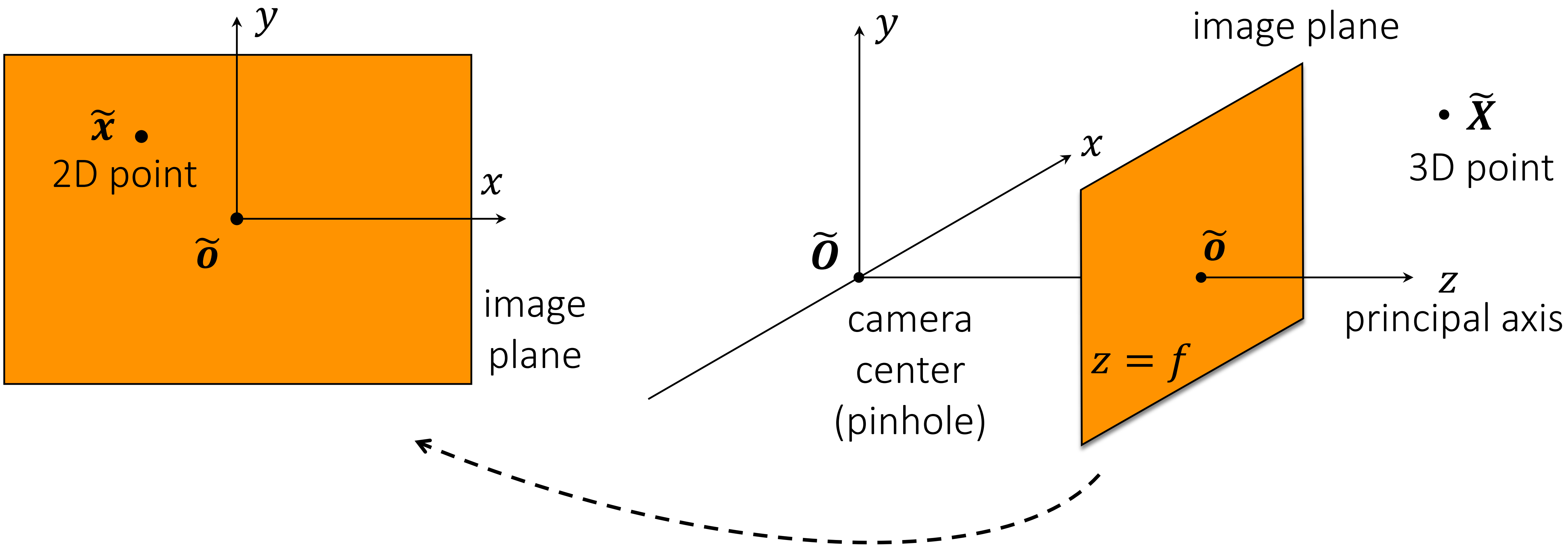
What does the pinhole camera projection look like?

Equivalently we can write:

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

combination of perspective projection and a 2D scaling transformation

Generalizations: coordinate systems

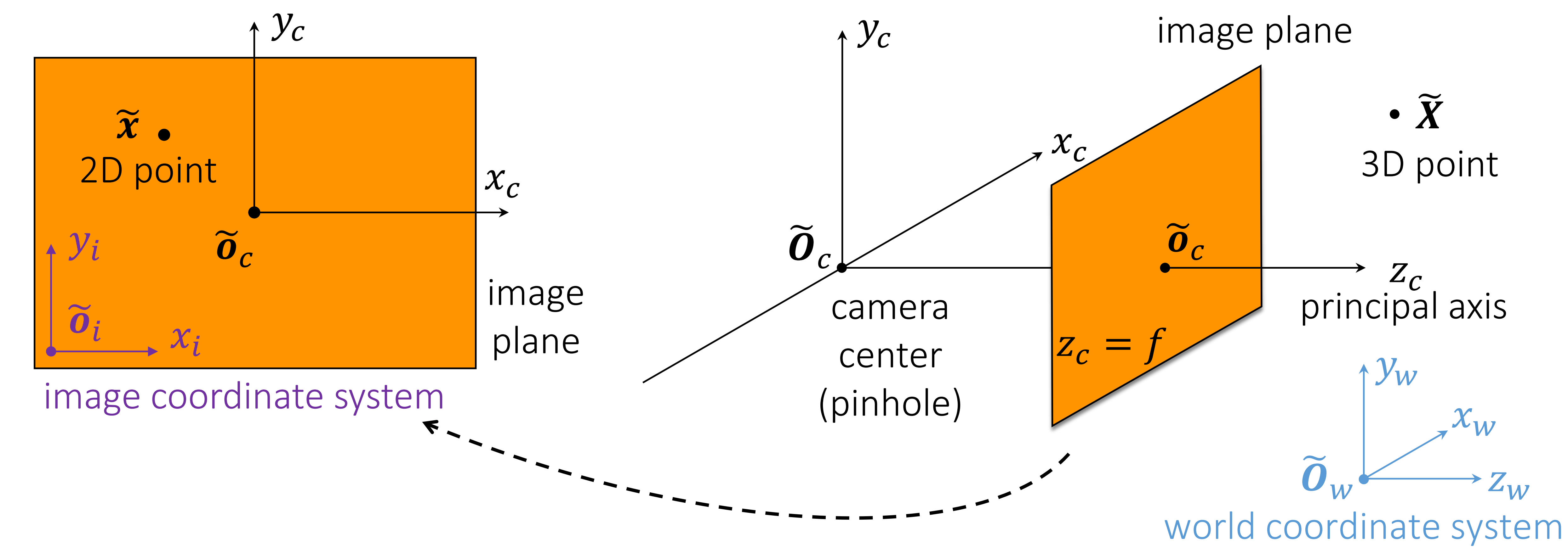


2D camera coordinate system

3D camera coordinate system

- A camera introduces two related coordinate systems, in 3D (world), and in 2D (image plane).

Generalizations: coordinate systems



2D camera coordinate system

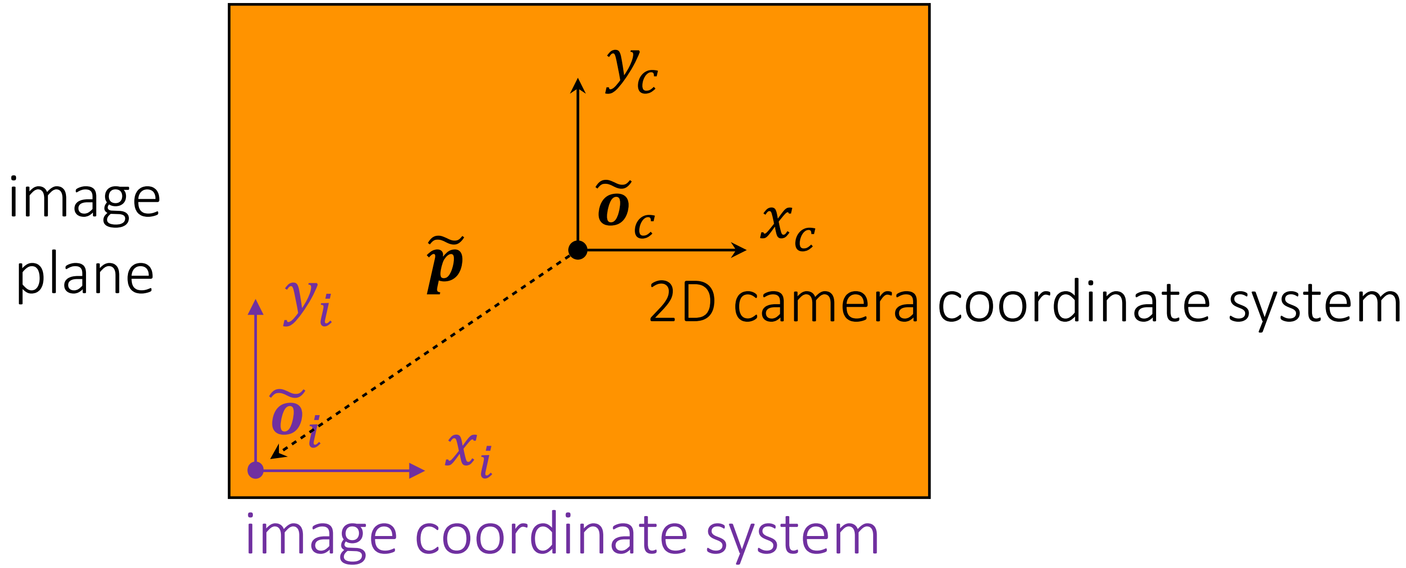
3D camera coordinate system

- A camera introduces two related coordinate systems, in 3D (world), and in 2D (image plane).
- These coordinate systems may be different from the coordinate systems of our application.

Generalization: image coordinate system

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

- Can you think of a case when this happens?



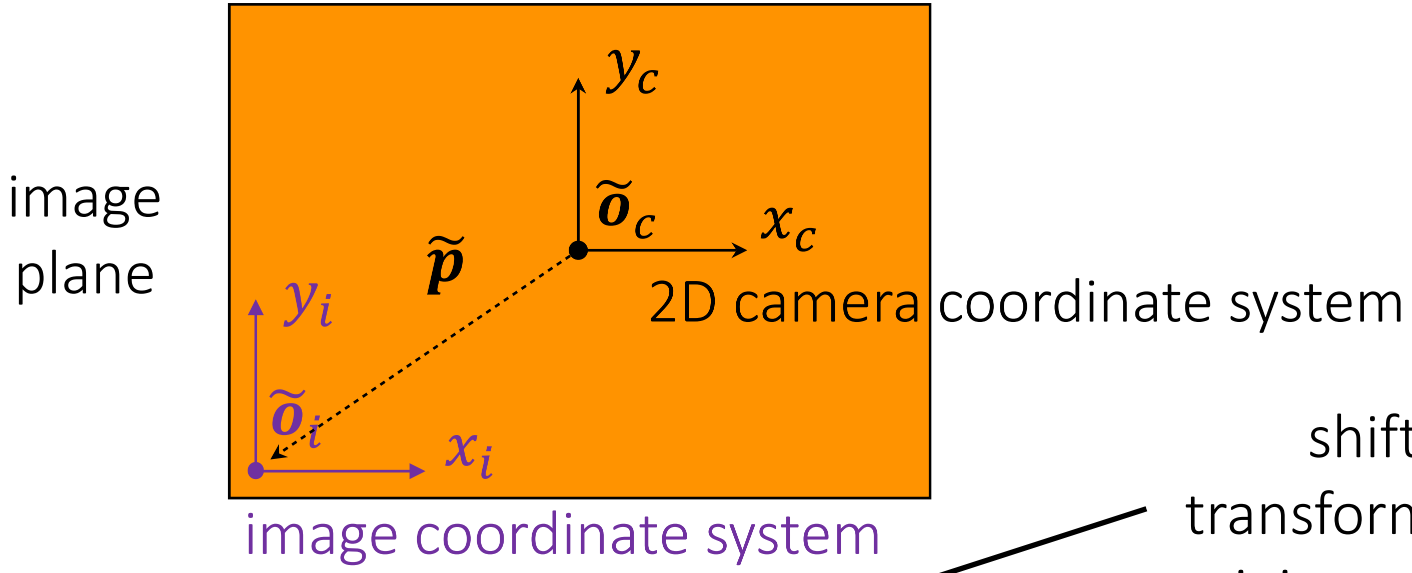
How does the camera matrix change?

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

Generalization: image coordinate system

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

- Can you think of a case when this happens?



shift vector transforming camera origin to image origin

How does the camera matrix change?

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

Camera matrix decomposition

We can decompose the camera matrix like this:

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

What does each part of the matrix represent?

Camera matrix decomposition

We can decompose the camera matrix like this:

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

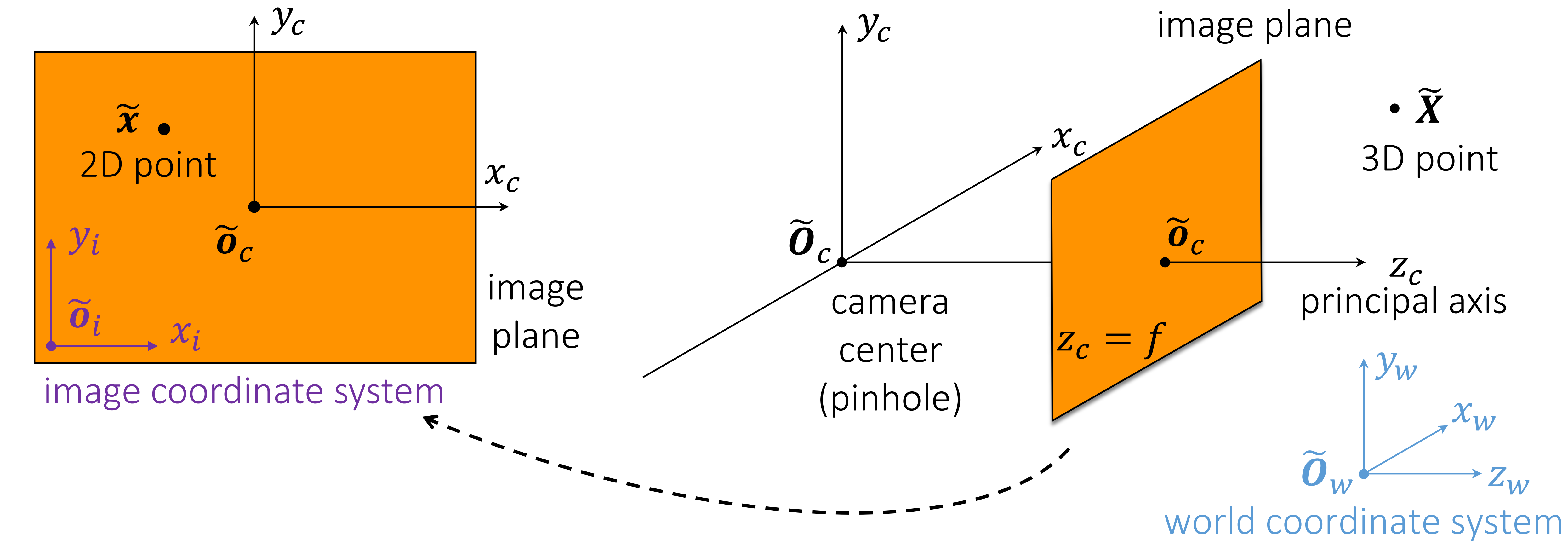
(homogeneous) transformation from 2D to 2D, accounting for non-unit focal length and origin shift

(homogeneous) perspective projection from 3D to 2D, assuming image plane at z = 1 and shared camera/image origin

Also written as:

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{I} \mid \boldsymbol{0}]$$

Generalizations: coordinate systems

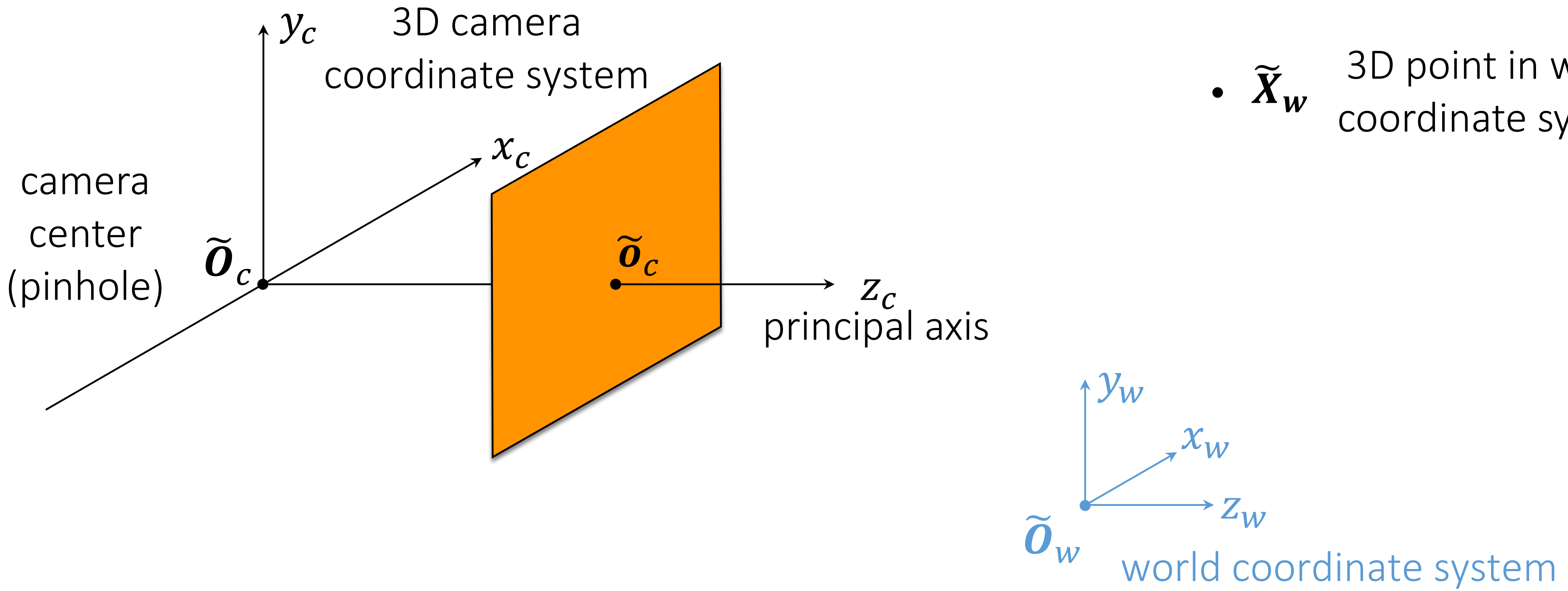


2D camera coordinate system

3D camera coordinate system

- A camera introduces two related coordinate systems, in 3D (world), and in 2D (image plane).
- These coordinate systems may be different from the coordinate systems of our application.

World-to-camera coordinate system transformation

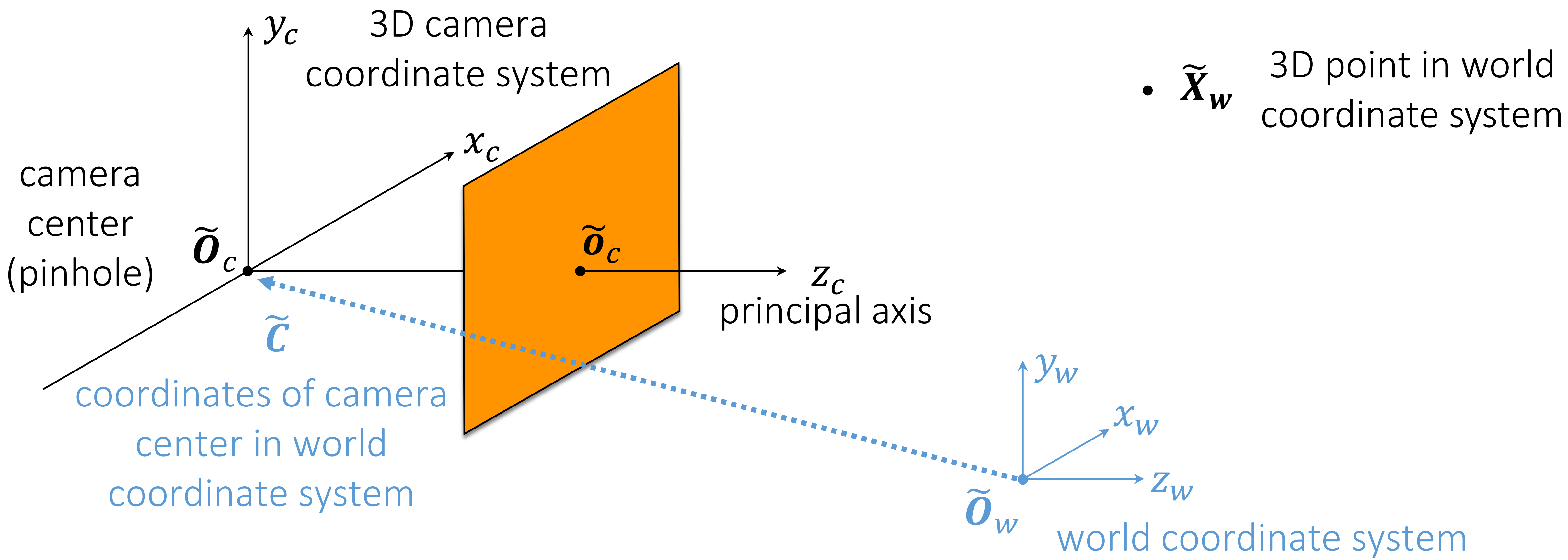


- $\tilde{\boldsymbol{X}}_w$ 3D point in world coordinate system

How do we express $\tilde{\boldsymbol{X}}$ in the 3D camera coordinate system?

$$\tilde{\boldsymbol{X}}_w$$

World-to-camera coordinate system transformation

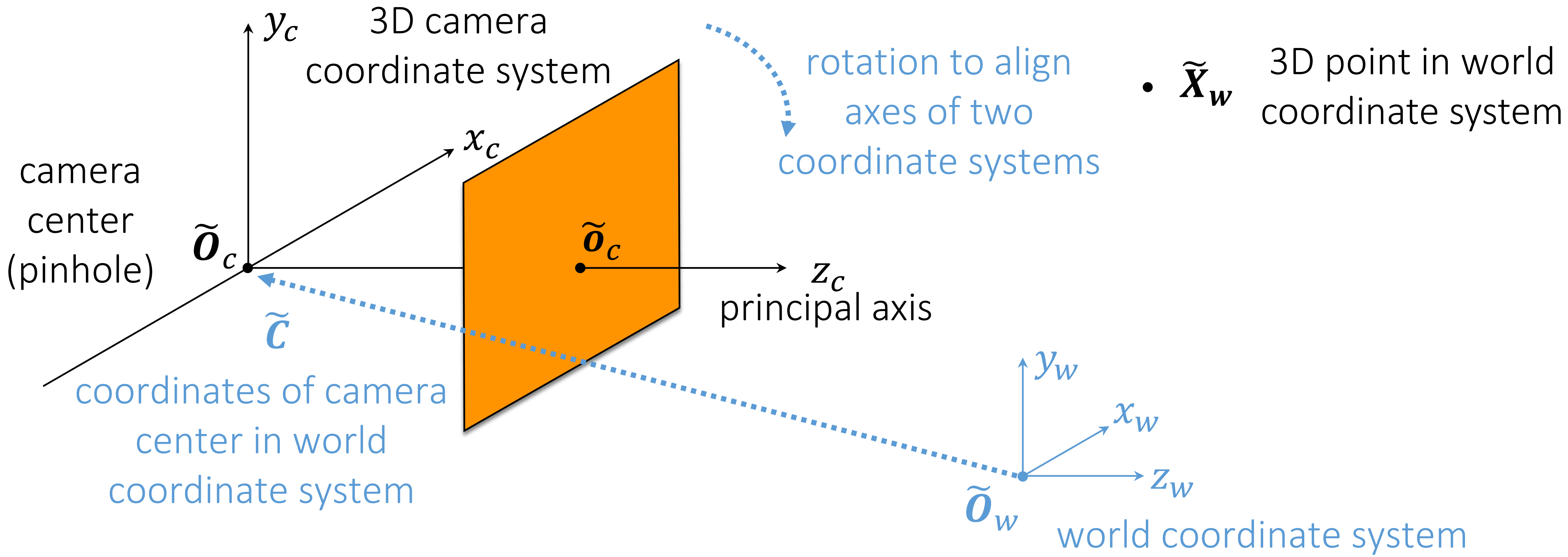


How do we express $\tilde{\boldsymbol{X}}$ in the 3D camera coordinate system?

$$\tilde{\boldsymbol{X}}_w - \tilde{\boldsymbol{C}}$$

translate

World-to-camera coordinate system transformation



How do we express $\tilde{\boldsymbol{X}}$ in the 3D camera coordinate system?

$$\boldsymbol{R} \cdot \left(\tilde{\boldsymbol{X}}_{\boldsymbol{w}} - \tilde{\boldsymbol{C}}\right)$$

rotate translate

Modeling the 3D coordinate system transformation

In heterogeneous coordinates, we have:

$$\tilde{\boldsymbol{X}}_{\boldsymbol{c}} = \boldsymbol{R} \cdot \left(\tilde{\boldsymbol{X}}_{\boldsymbol{w}} - \tilde{\boldsymbol{C}}\right)$$

How do we write this transformation in homogeneous coordinates?

Modeling the 3D coordinate system transformation

In heterogeneous coordinates, we have:

$$\tilde{\boldsymbol{X}}_{\boldsymbol{c}} = \boldsymbol{R} \cdot \left(\tilde{\boldsymbol{X}}_{\boldsymbol{w}} - \tilde{\boldsymbol{C}}\right)$$

In homogeneous coordinates, we have:

$$\boldsymbol{X}_{\boldsymbol{c}} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\tilde{\boldsymbol{C}} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{X}_{\boldsymbol{w}}$$

Incorporating the transform in the camera matrix

The previous camera matrix is for homogeneous 3D coordinates in camera coordinate system:

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X_c} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{I} \mid \boldsymbol{0}] \boldsymbol{X_c}$$

We also just derived:

$$\boldsymbol{X_c} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\widetilde{\boldsymbol{C}} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{X_w}$$

Putting it all together

We can write everything into a single projection:

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X_w}$$

The camera matrix now looks like:

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{I} \mid \boldsymbol{0}] \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\widetilde{\boldsymbol{C}} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

intrinsic parameters (3 x 3): correspond to camera internals (2D image-to-image transformation)

perspective projection (3 x 4): maps 3D to 2D points (camera-to-image transformation)

extrinsic parameters (4 x 4): correspond to camera externals (3D world-to-camera transformation)

Putting it all together

We can write everything into a single projection:

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X_w}$$

The camera matrix now looks like:

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{R} \mid -\boldsymbol{R}\widetilde{\boldsymbol{C}}]$$

It is common to combine the perspective projection and extrinsics in one matrix.

intrinsic parameters (3 x 3): correspond to camera internals (2D image-to-image transformation)

extrinsic parameters (3 x 4): correspond to camera externals (3D world-to-camera transformation) *and* perspective projection

The pinhole camera matrix

More compactly, we can write the pinhole camera matrix as:

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]$$

where

$$\underbrace{\underset{\text{2D Euclidean transform}}{\boldsymbol{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{intrinsic parameters}} \qquad \underbrace{\underset{\text{3D rotation}}{\boldsymbol{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}} \qquad \underset{\text{3D translation}}{\boldsymbol{t} = -\boldsymbol{R}\widetilde{\boldsymbol{C}} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}}}_{\text{extrinsic parameters}}$$

More general pinhole camera matrices

The following is the standard pinhole camera matrix we saw.

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

More general pinhole camera matrices

The following is the standard pinhole camera matrix we saw.

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

- 9 degrees of freedom (3 for intrinsics, 3 for rotation, 3 for translation).

More general pinhole camera matrices

The following is the standard pinhole camera matrix we saw.

$$\boldsymbol{P} = \boxed{\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

- 9 degrees of freedom (3 for intrinsics, 3 for rotation, 3 for translation).

We can get more general pinhole cameras with more degrees of freedom by generalizing the intrinsics matrix, while leaving everything else the same..

More general pinhole camera matrices

CCD camera: pixels may not be square.

$$\boldsymbol{P} = \begin{bmatrix} a_x & 0 & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

More general pinhole camera matrices

CCD camera: pixels may not be square.

$$\boldsymbol{P} = \begin{bmatrix} a_x & 0 & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

- 10 degrees of freedom (4 for intrinsics, 3 for rotation, 3 for translation).

More general pinhole camera matrices

Finite projective camera: sensor may be skewed.

$$\boldsymbol{P} = \begin{bmatrix} a_x & s & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

More general pinhole camera matrices

Finite projective camera: sensor may be skewed.

$$\boldsymbol{P} = \begin{bmatrix} a_x & s & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

- 11 degrees of freedom (5 for intrinsics, 3 for rotation, 3 for translation).

Can we get a *perspective projection* camera with more degrees of freedom?

More general pinhole camera matrices

Finite projective camera: sensor may be skewed.

The finite projective camera is the most general camera implementing perspective projection.

$$\boldsymbol{P} = \begin{bmatrix} a_x & s & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

- 11 degrees of freedom (5 for intrinsics, 3 for rotation, 3 for translation).

Can we get a *perspective projection* camera with more degrees of freedom?

- No, as the entire camera matrix $\boldsymbol{P}$ has 12 elements (3x4) and is defined up to scale.

More general pinhole camera matrices

Finite projective camera: sensor may be skewed.

The finite projective camera is the most general camera implementing $\boxed{\text{perspective projection.}}$

$$\boldsymbol{P} = \begin{bmatrix} a_x & s & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \boxed{\left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

How many degrees of freedom does this matrix have?

- 11 degrees of freedom (5 for intrinsics, 3 for rotation, 3 for translation).

Can we get a *perspective projection* camera with more degrees of freedom?

- No, as the entire camera matrix $\boldsymbol{P}$ has 12 elements (3x4) and is defined up to scale.

Perspective distortion

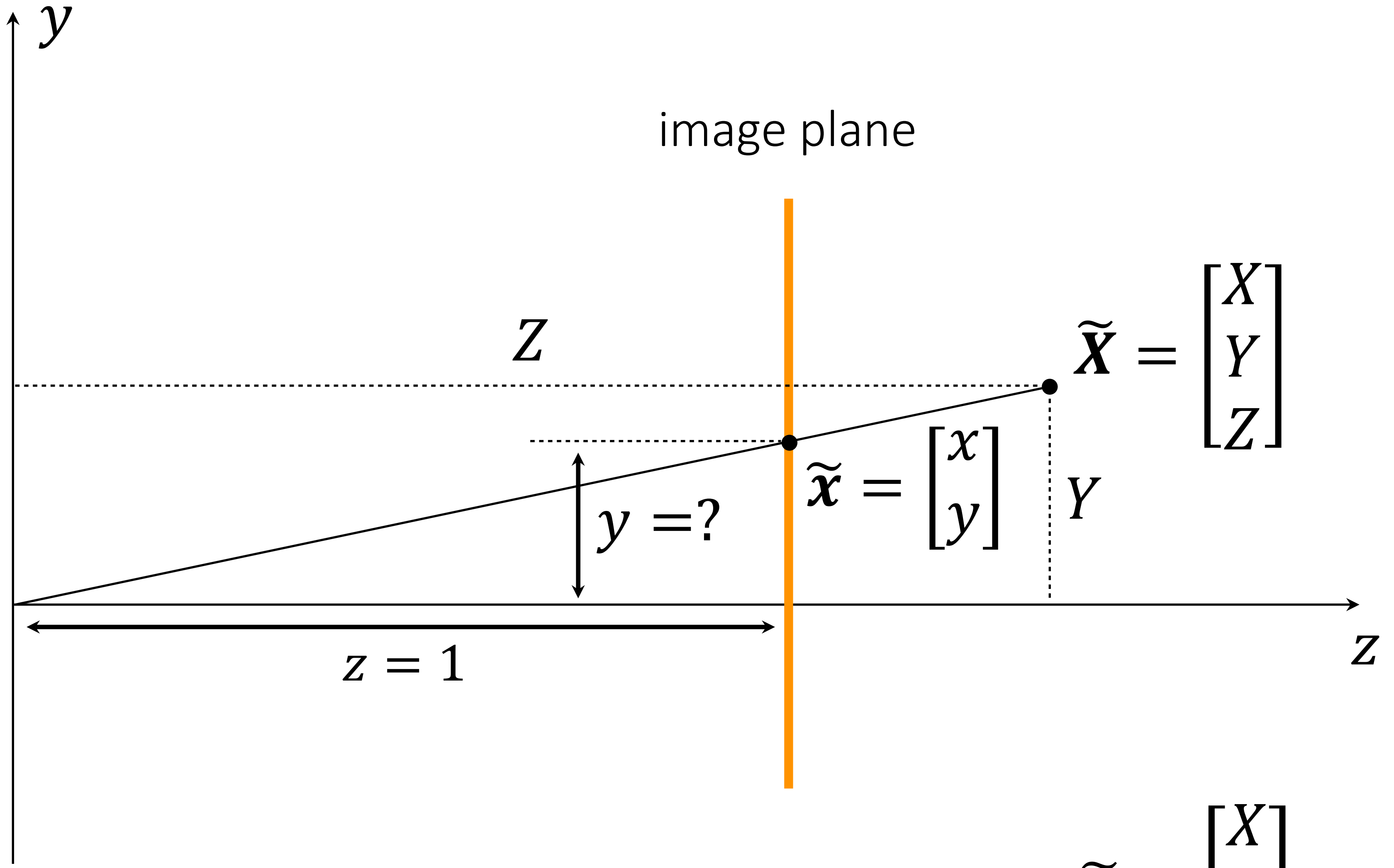
Finite projective camera

Let’s ignore intrinsics and extrinsics for now.

$$\boldsymbol{P} = \begin{bmatrix} a_x & s & p_x \\ 0 & a_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

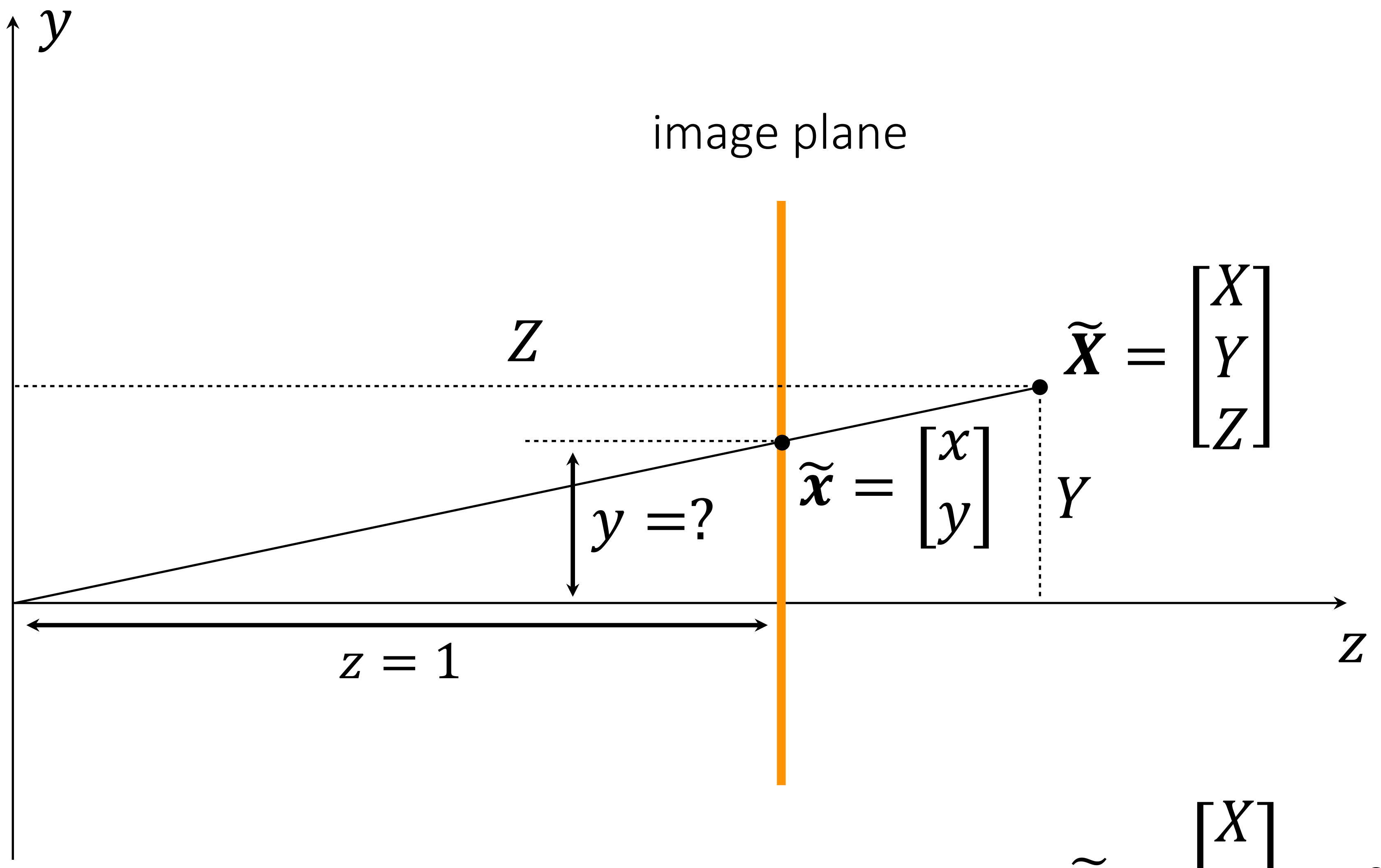
What is the effect of the perspective projection matrix?

The 2D view of the (rearranged) pinhole camera



Using similar triangles: $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

The 2D view of the (rearranged) pinhole camera



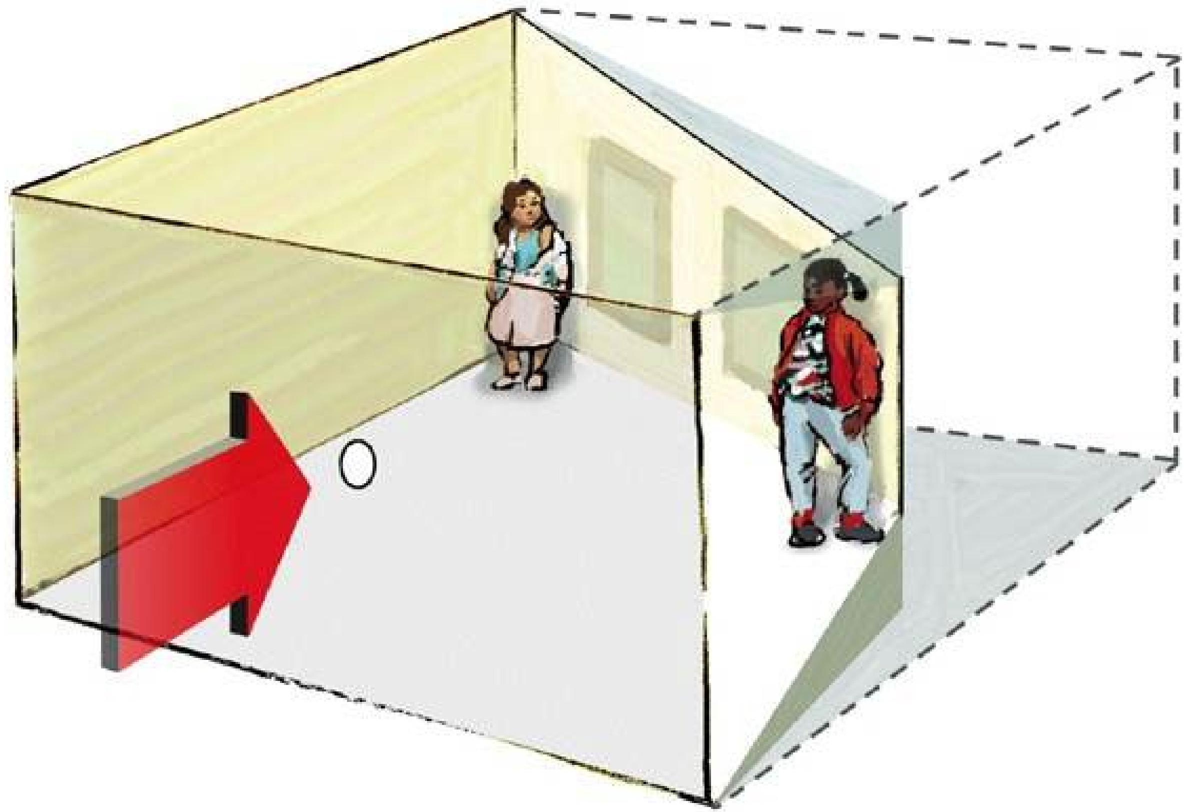
Perspective distortion: magnification changes with depth

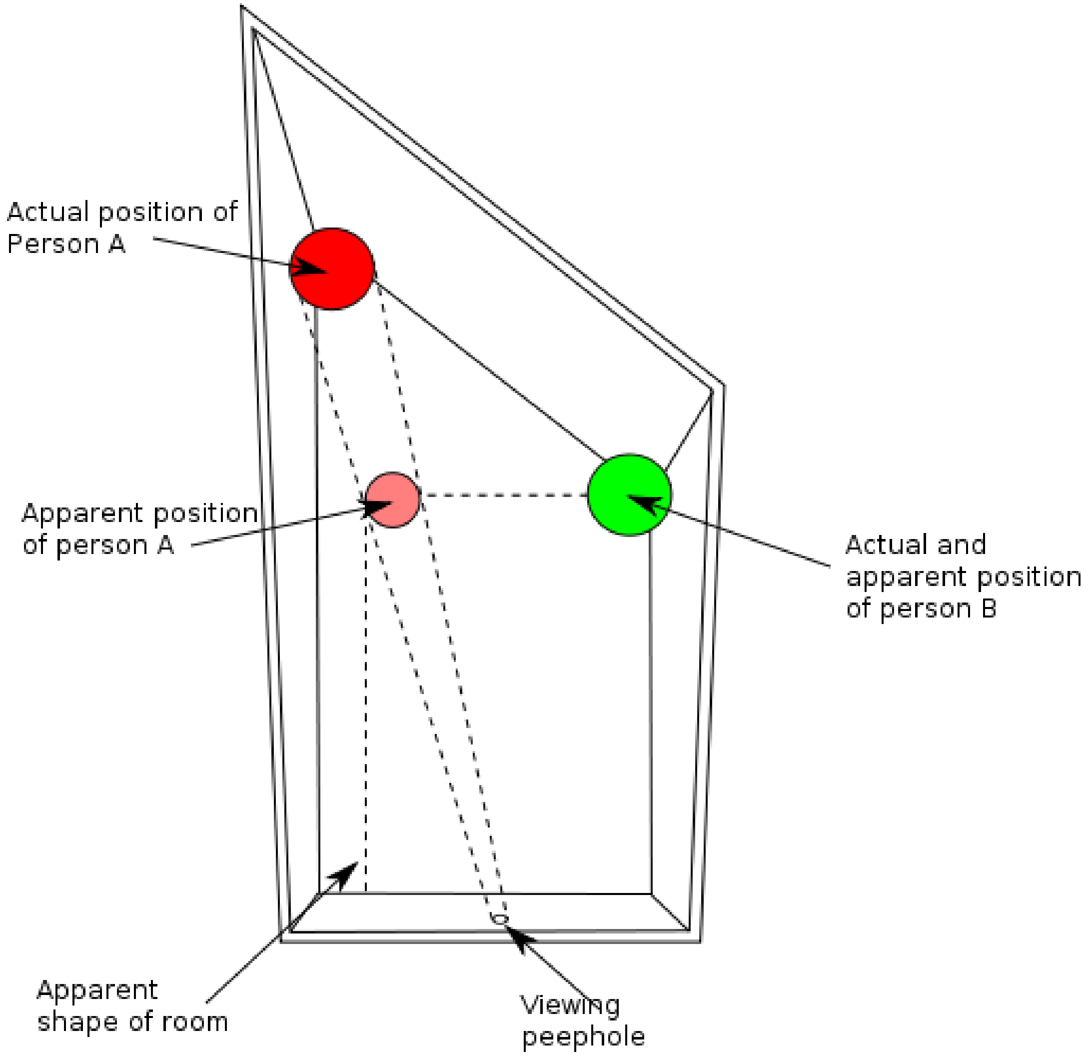
Using similar triangles: $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/\boxed{Z} \\ Y/\boxed{Z} \end{bmatrix}$

Forced perspective

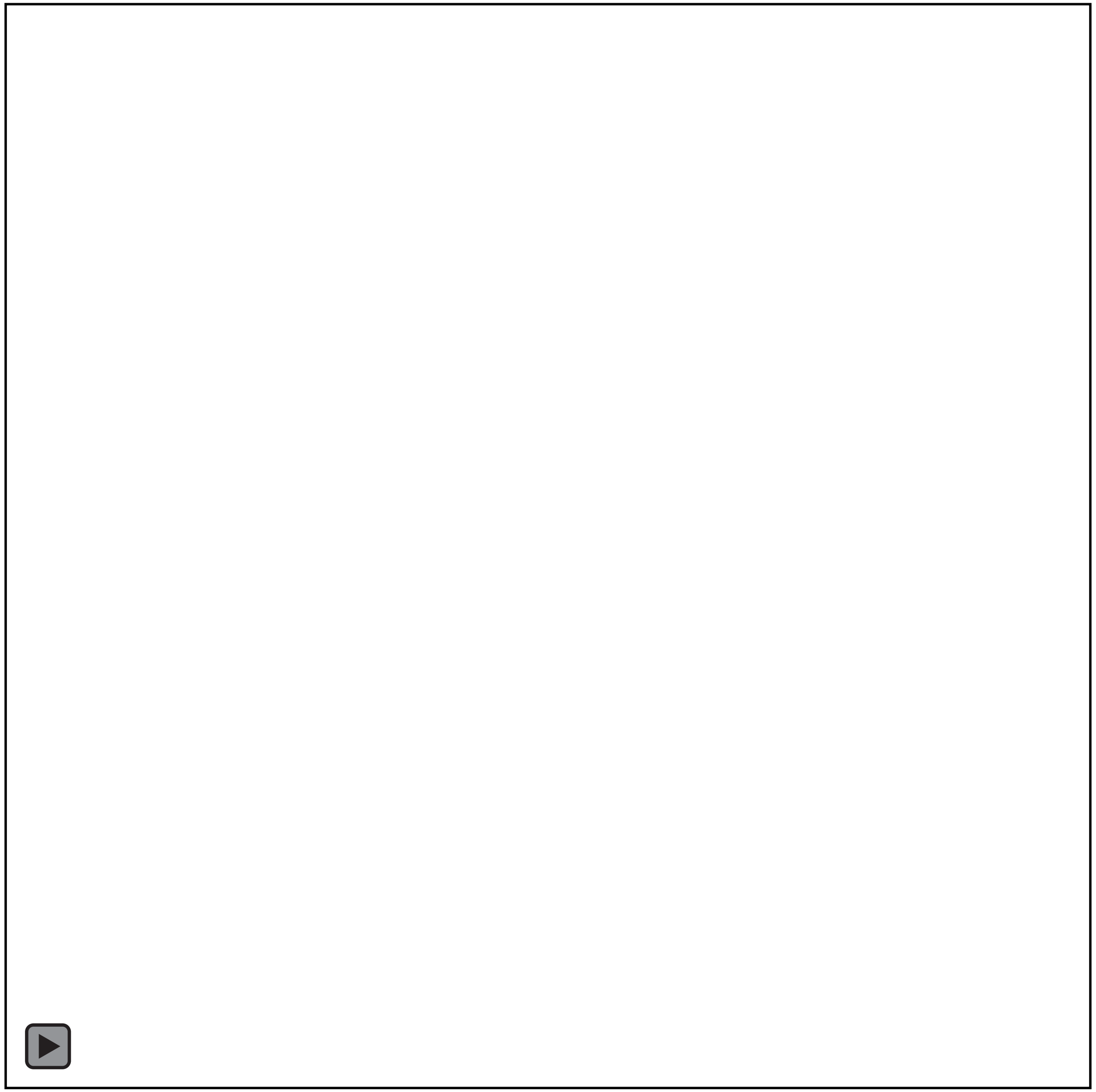
The Ames room illusion

The Ames room illusion





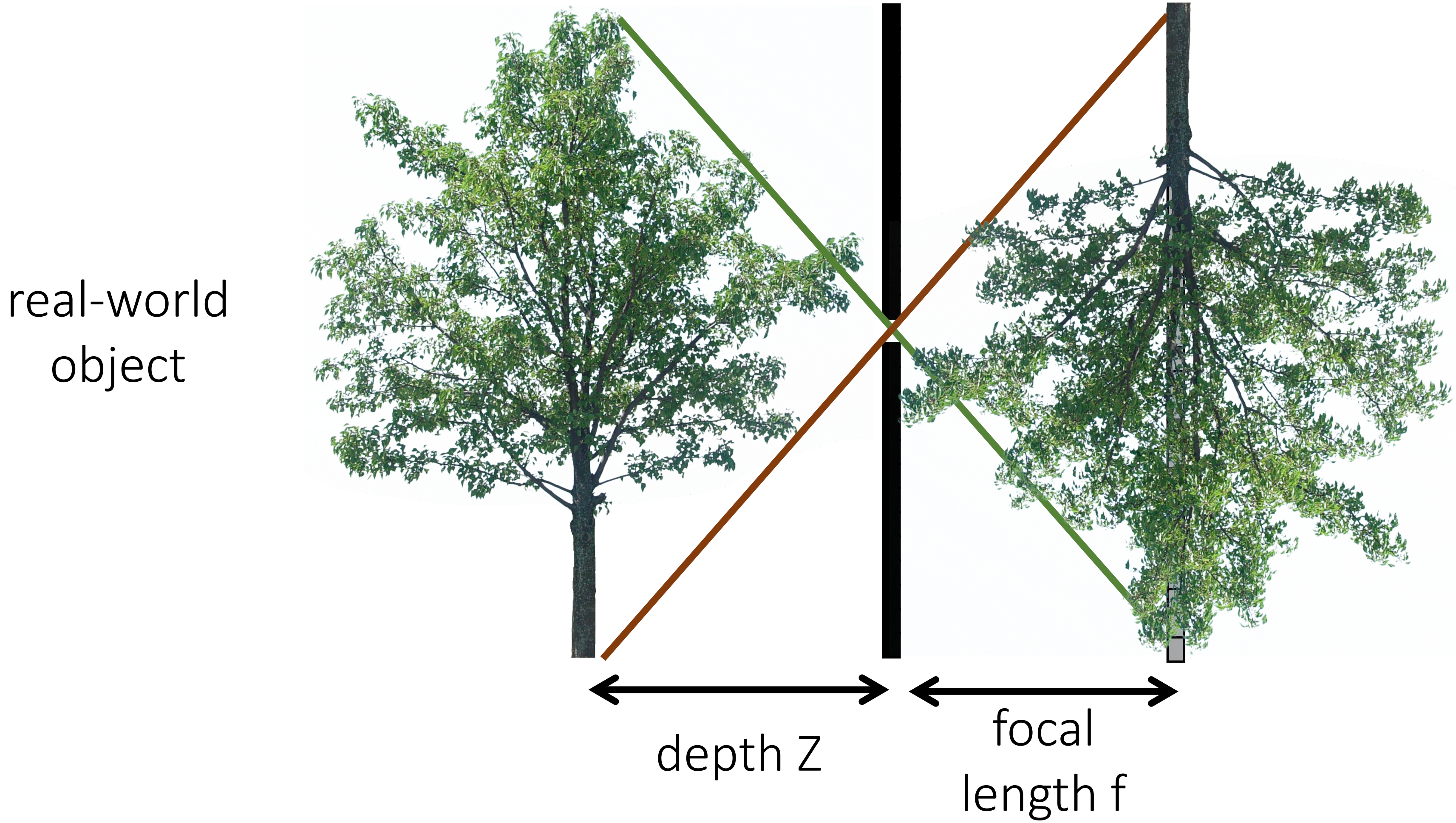
The arrow illusion



Is there a camera without perspective distortion?

Other camera models

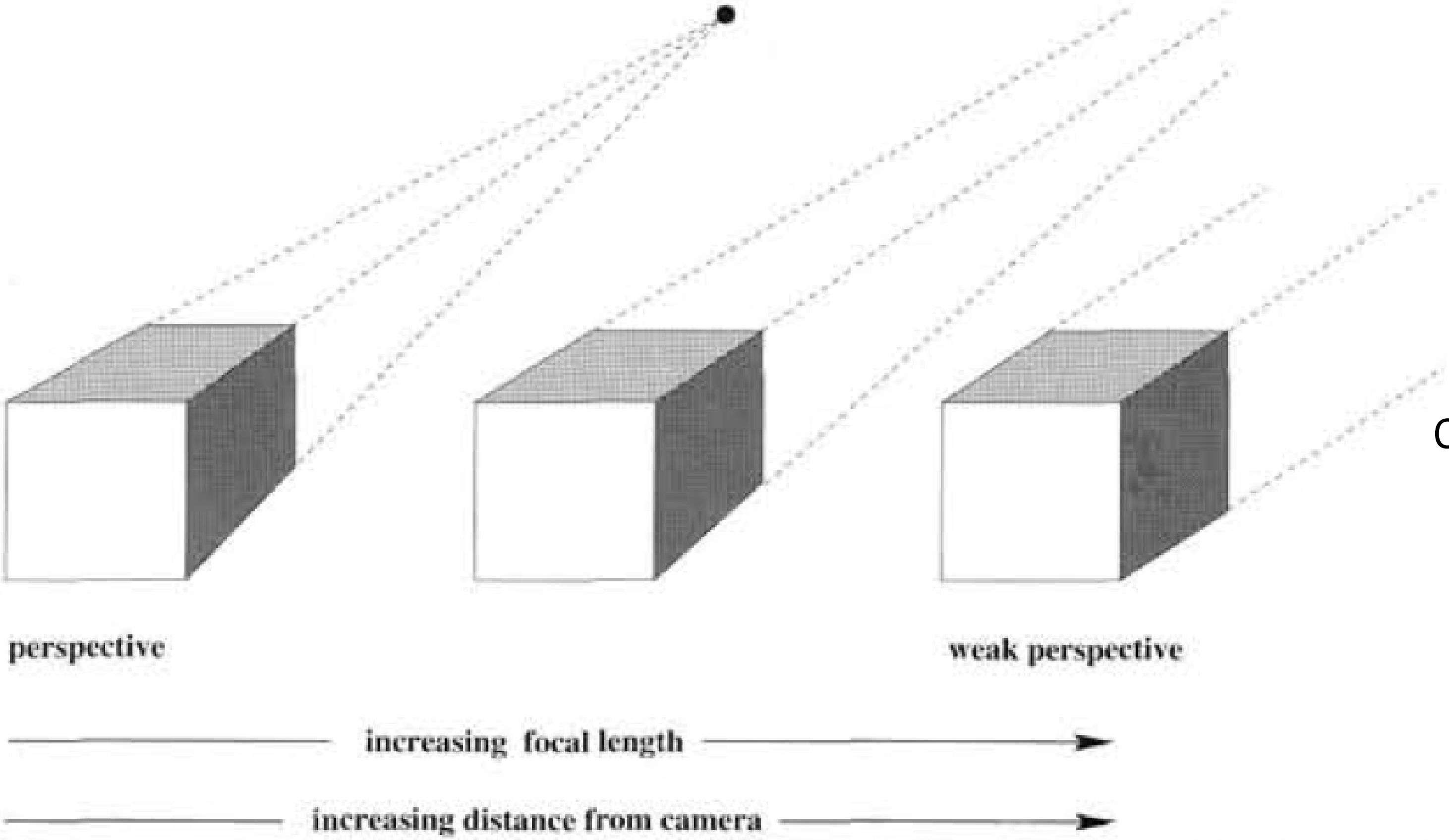
What if...



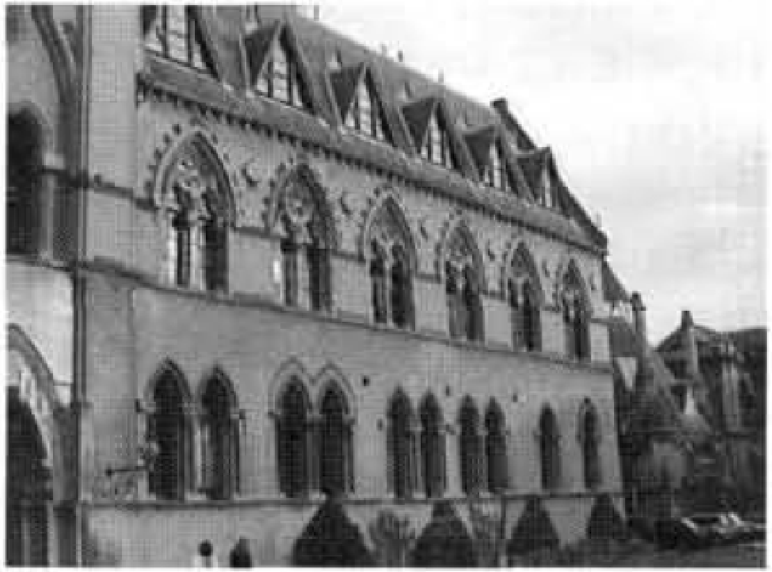
... we continue increasing Z and f while maintaining same magnification?

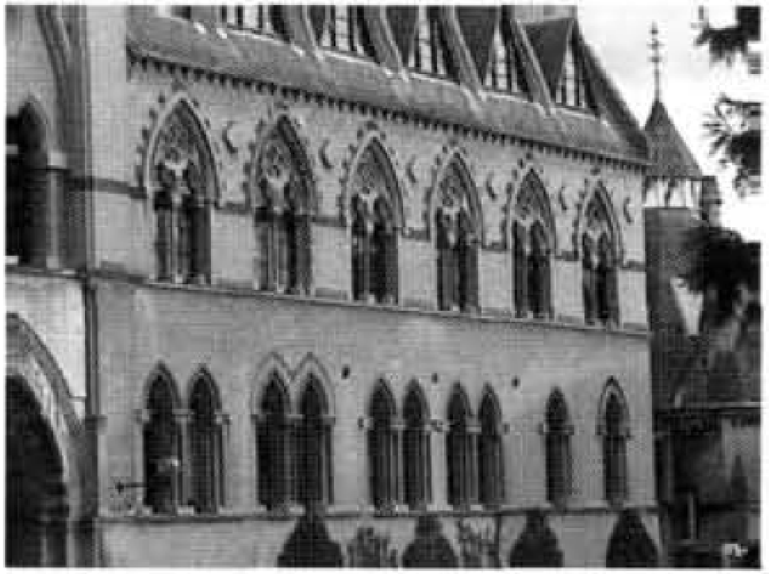
$$f \rightarrow \infty \text{ and } \frac{f}{Z} = \text{constant}$$

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

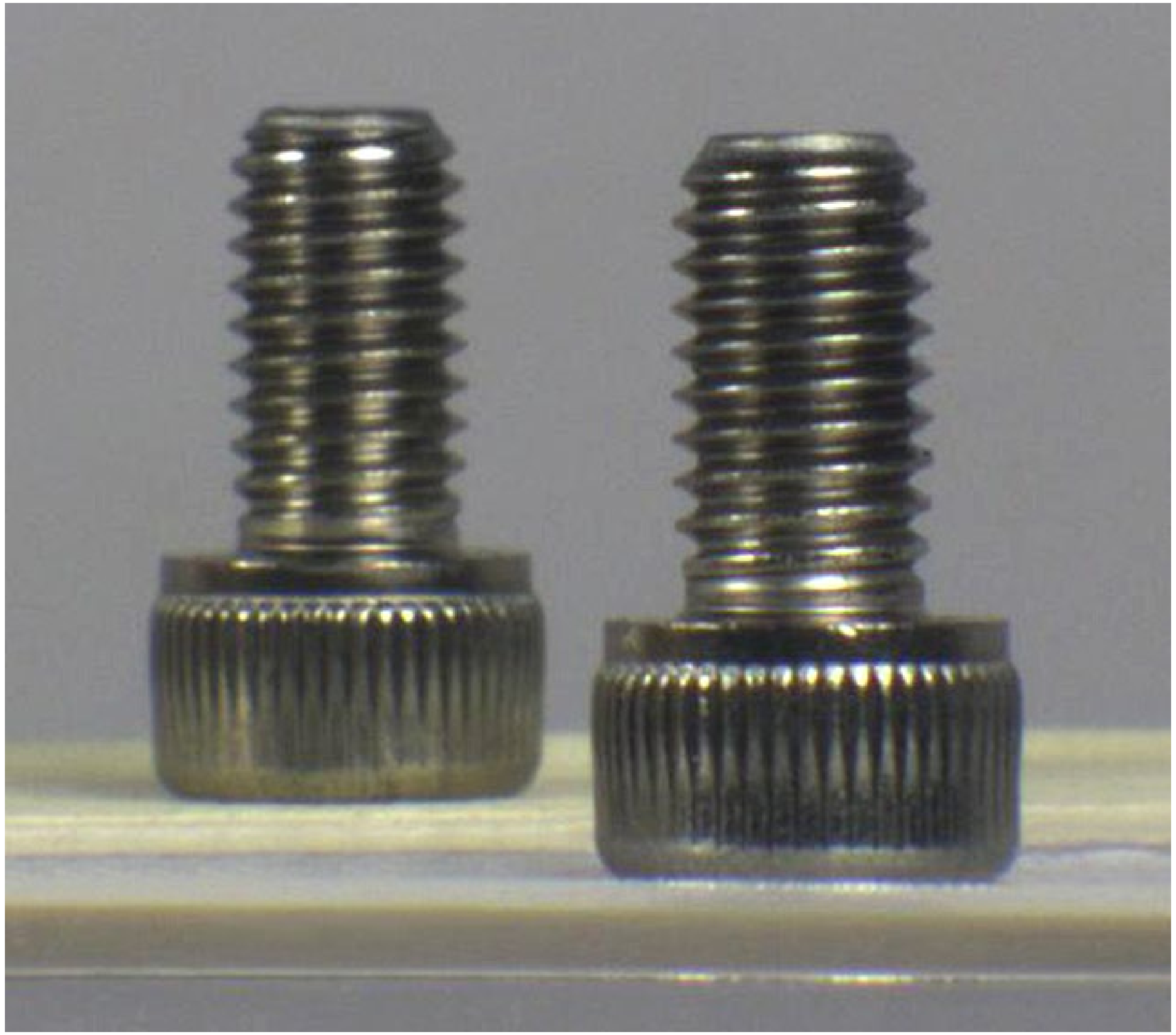


Weak-perspective camera: camera is *far* from object and has *large* focal length

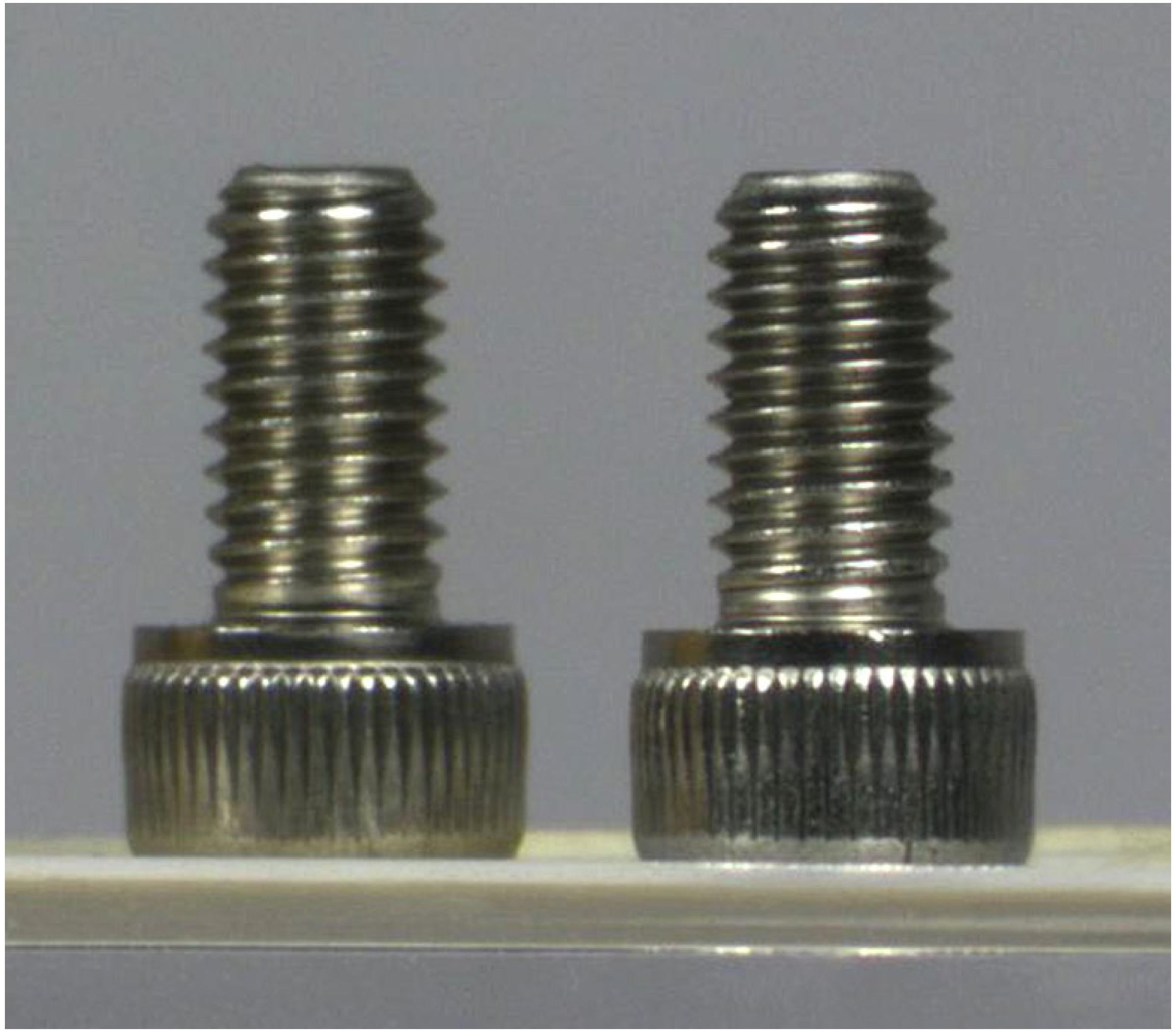




Different cameras

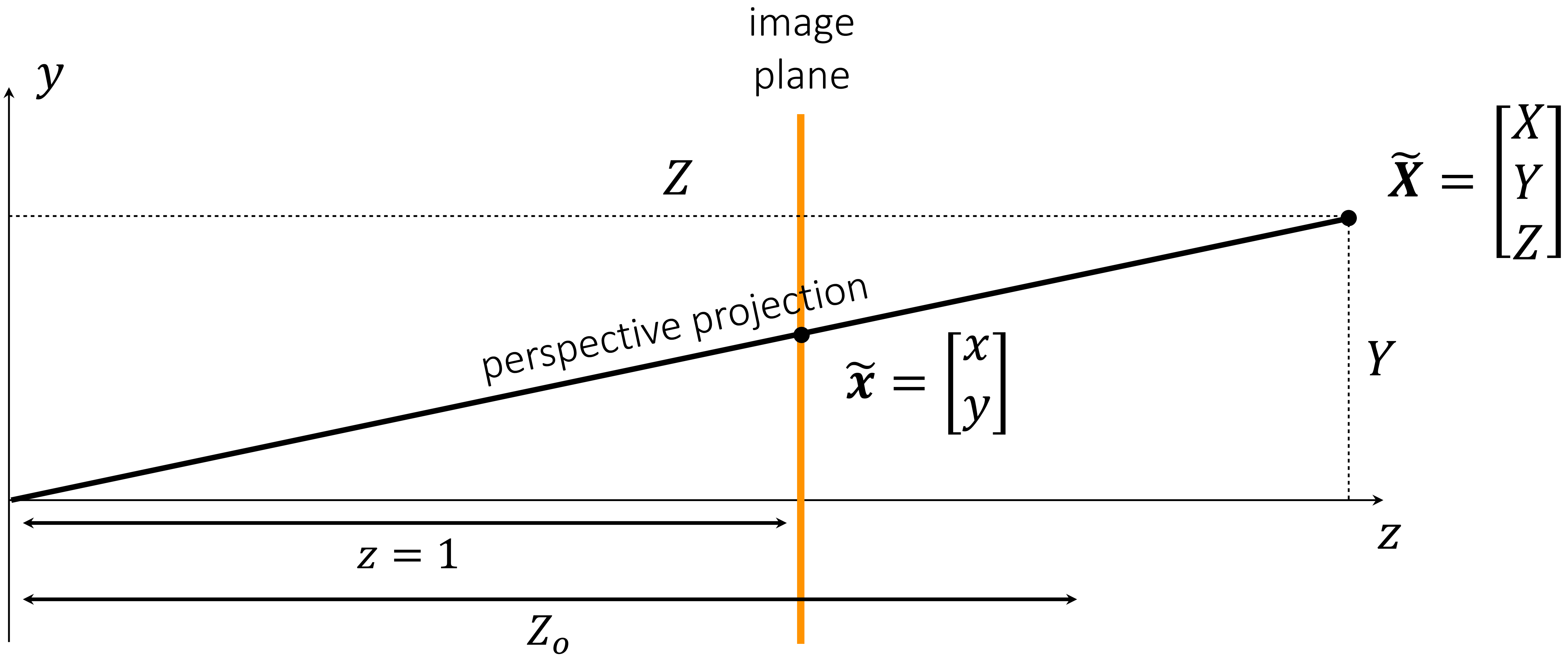


perspective camera



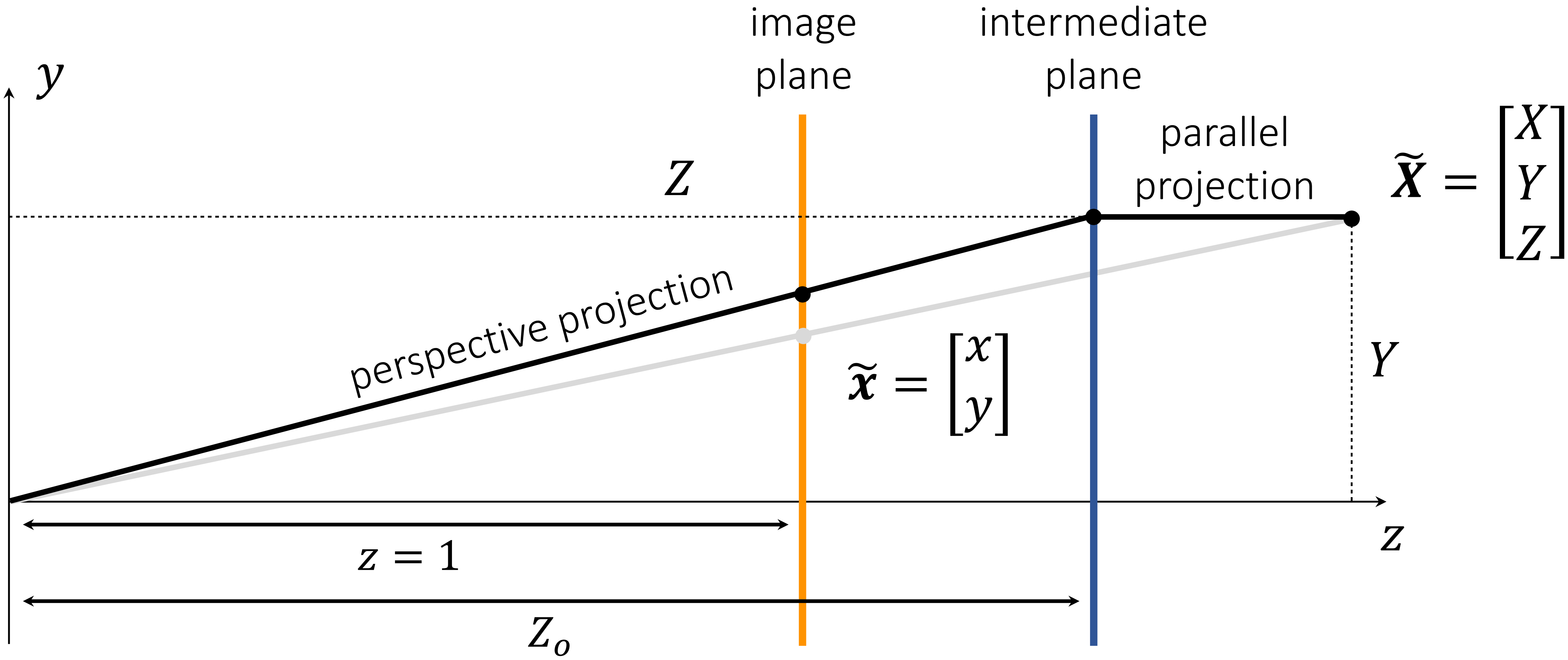
weak perspective camera

Perspective versus weak-perspective camera



perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

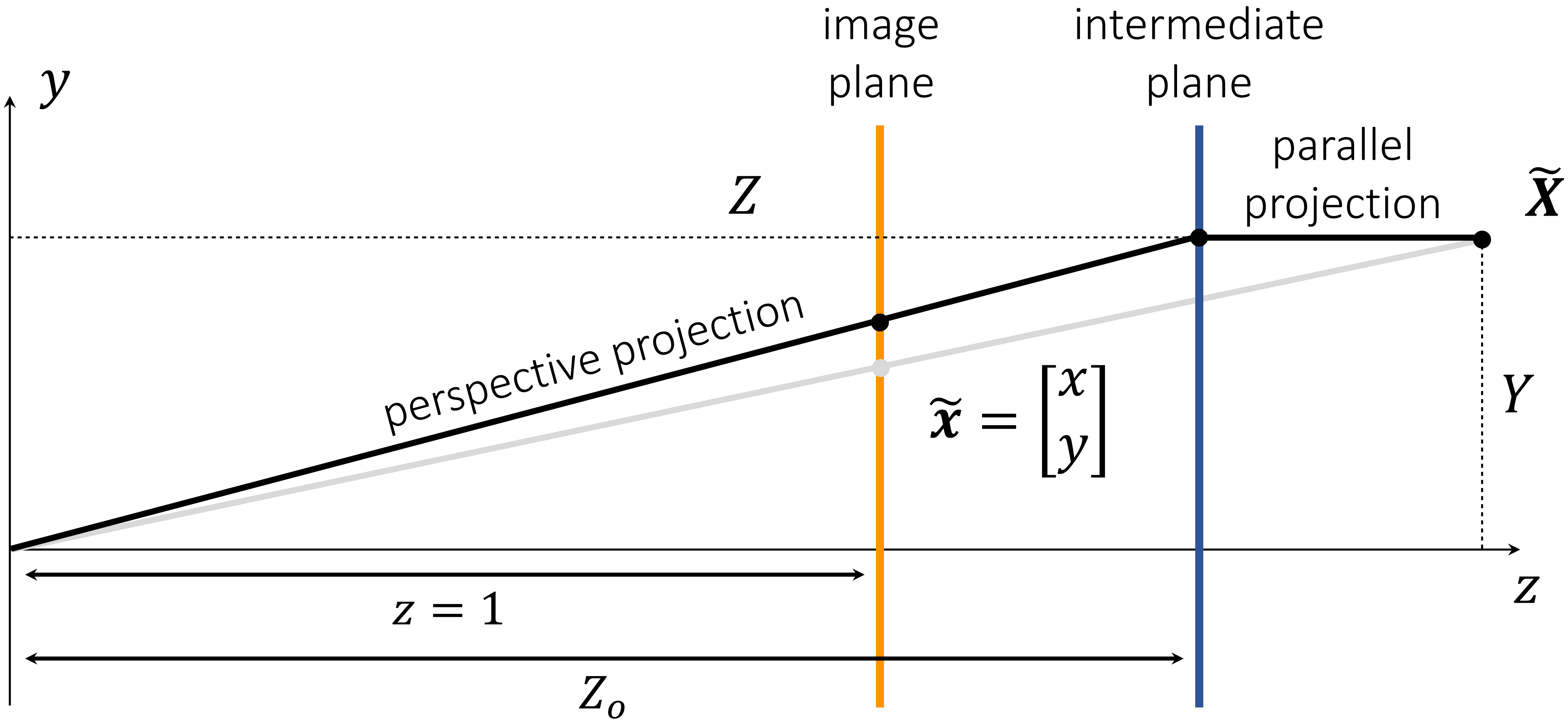
Perspective versus weak-perspective camera



perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

weak-perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z_o \\ Y/Z_o \end{bmatrix}$

Perspective versus weak-perspective camera



$$\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

magnification *independent* of depth, depends only on Z_o

perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

weak-perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z_o \\ Y/Z_o \end{bmatrix}$

Comparing camera projection matrices

Let’s ignore intrinsics and extrinscis for now.

- The *perspective projection matrix* can be written as:

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

- The *weak-perspective projection matrix* can be written as:

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_o \end{bmatrix}$$

Comparing camera matrices

Let's now incorporate intrinsics and extrinsics.

- The *finite projective camera matrix* can be written as:

$$\boldsymbol{P} = \boldsymbol{K}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ \boldsymbol{0} & 1\end{bmatrix}$$

Change only the projection matrix, and use the exact same matrices for intrinsics and extrinsics.

- The *affine camera matrix* can be written as:

$$\boldsymbol{P} = \boldsymbol{K}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & Z_o\end{bmatrix}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ \boldsymbol{0} & 1\end{bmatrix}$$

Special case: orthographic projection

Let's now incorporate intrinsics and extrinsics.

- The *finite projective camera matrix* can be written as:

$$\boldsymbol{P} = \boldsymbol{K}\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\\boldsymbol{0} & 1\end{bmatrix}$$

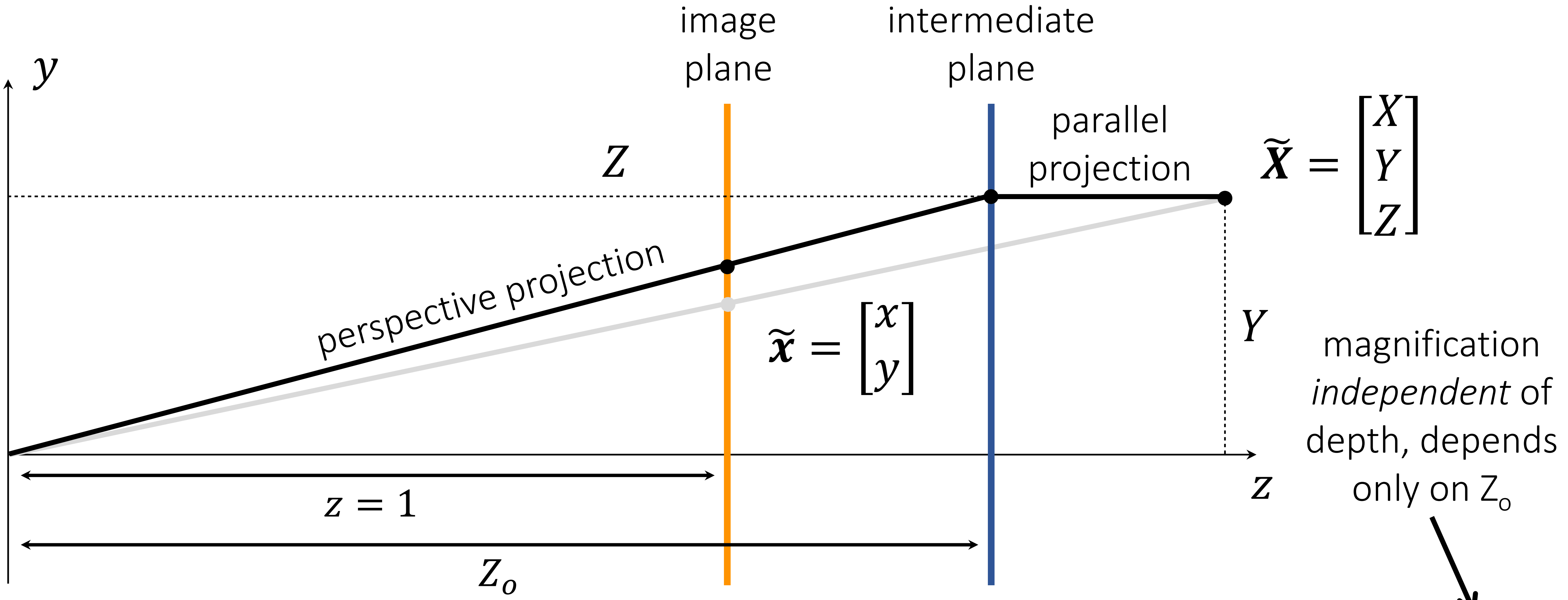
Change only the projection matrix, and use the exact same matrices for intrinsics and extrinsics.

- The *affine camera matrix* can be written as:

What's the effect of setting $Z_o = 1$?

$$\boldsymbol{P} = \boldsymbol{K}\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & \boxed{1}\end{bmatrix}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\\boldsymbol{0} & 1\end{bmatrix}$$

Perspective versus weak-perspective camera



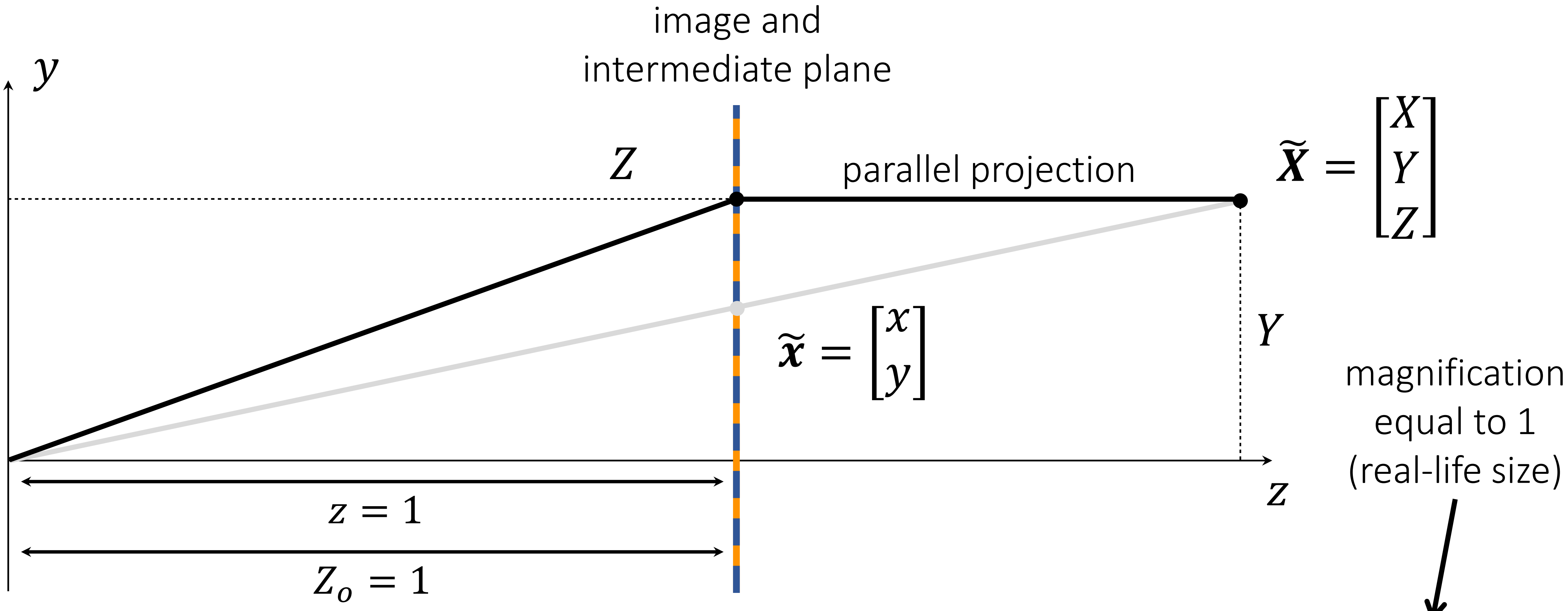
$$\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} x \\ y \end{bmatrix}$$

perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

weak-perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/\boxed{Z_o} \\ Y/\boxed{Z_o} \end{bmatrix}$

Perspective versus orthographic camera



perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$

weak-perspective projection $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} = \begin{bmatrix} X \\ Y \end{bmatrix}$

When can we assume a weak-perspective camera?

When can we assume a weak-perspective camera?

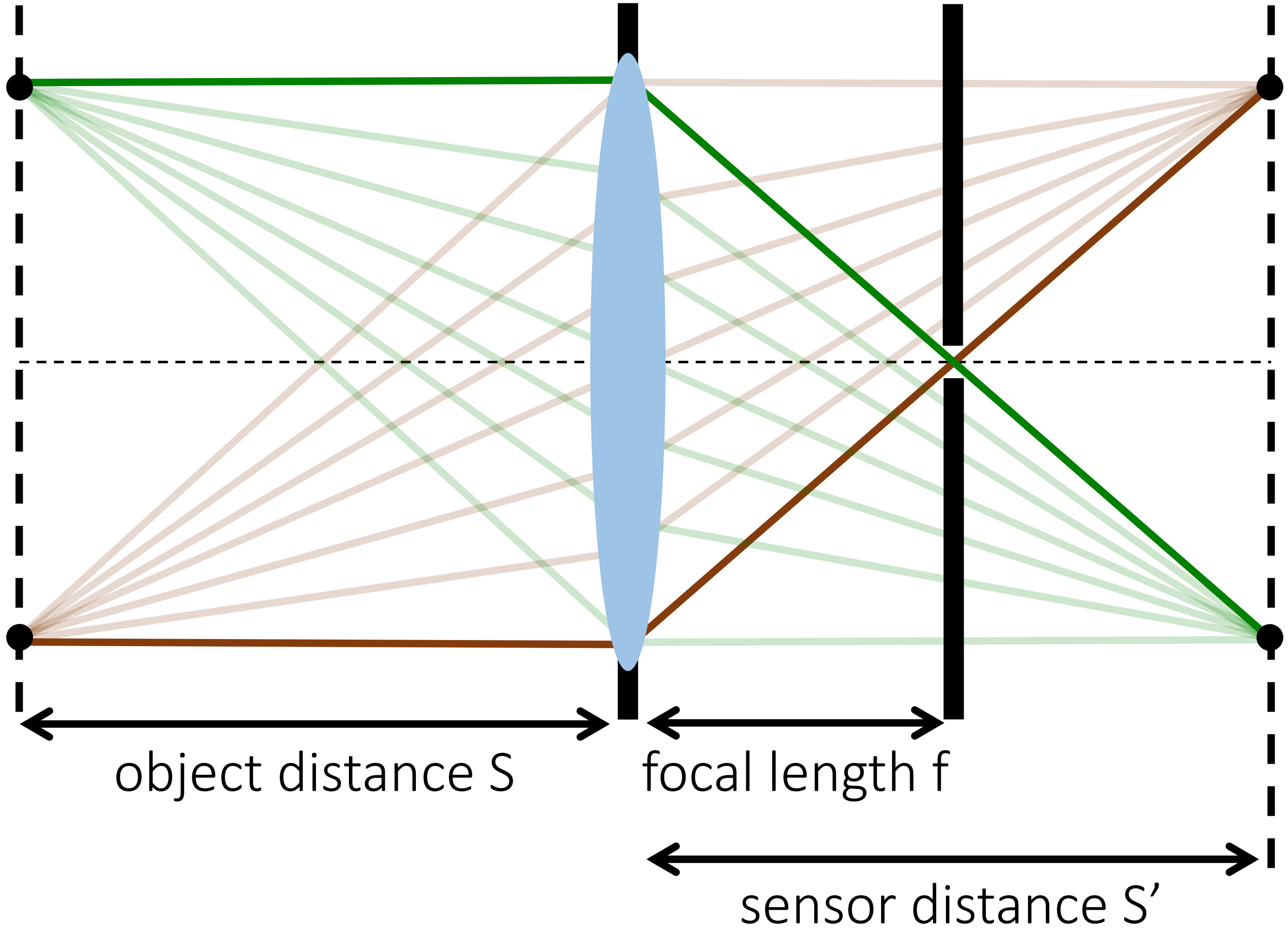
1. When the scene (or parts of it) is very far away.

Weak-perspective projection applies to the mountains.

When can we assume a weak-perspective camera?

2. When we use a telecentric lens.

Place a pinhole at focal length, so that only rays parallel to primary ray pass through.

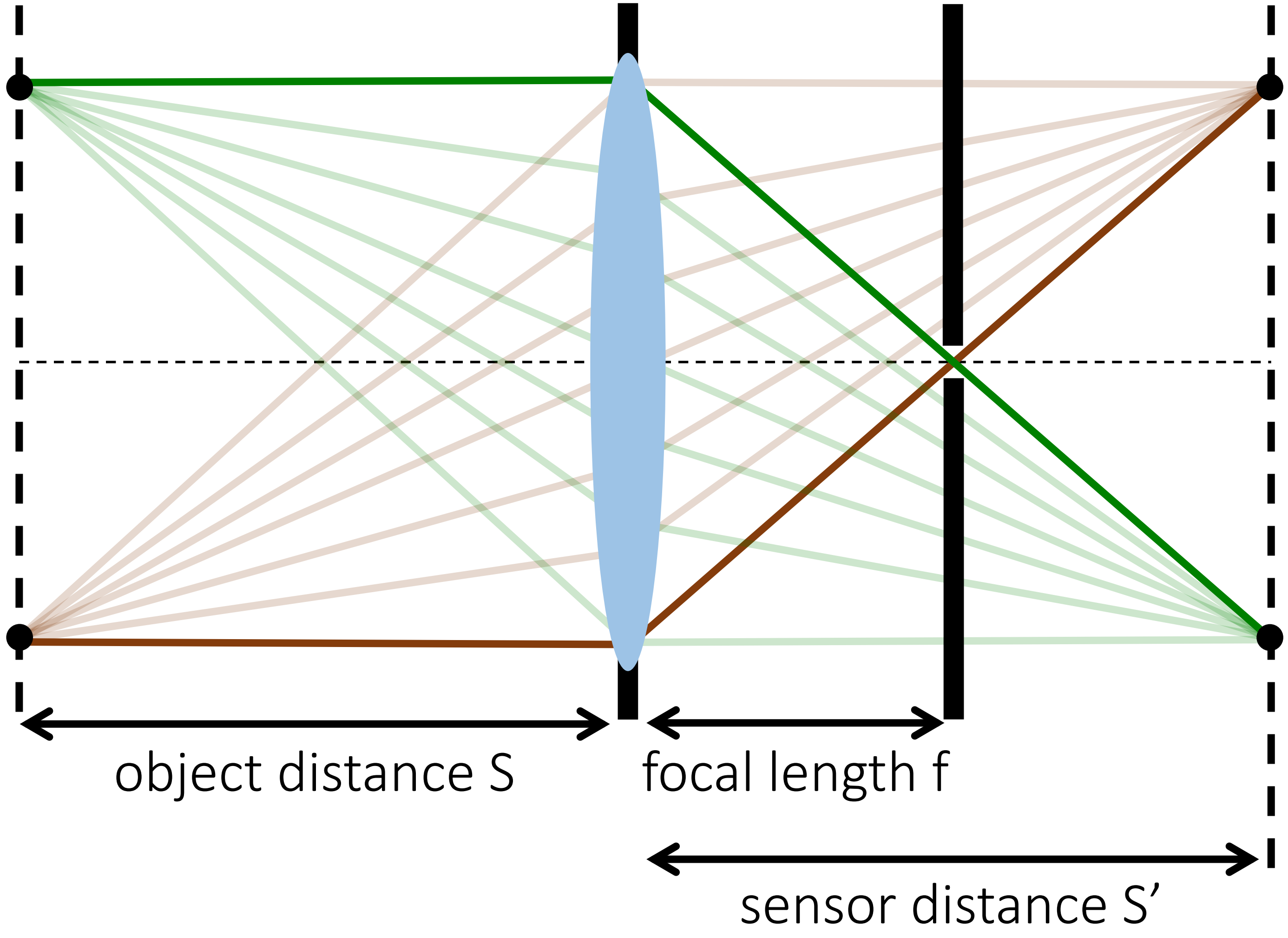


$$Z_o = f$$

remember that *focal length* f refers to different things in pinhole and lens cameras

Orthographic projection using a telecentric lens

How do we make the telecentric lens act as an orthographic camera?

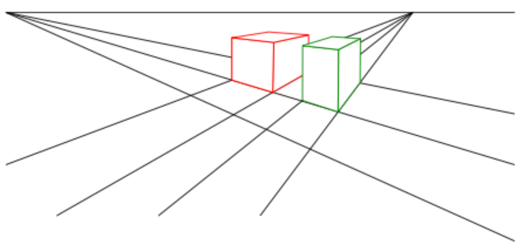


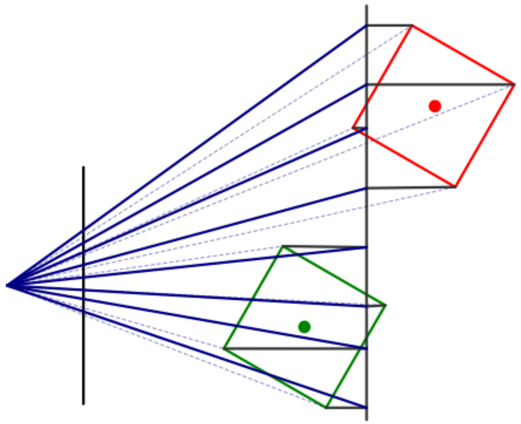
We set the sensor distance as:

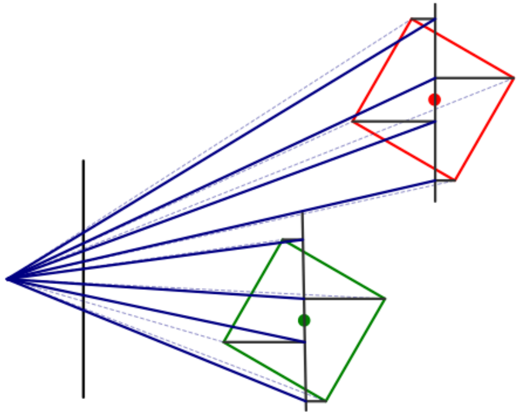
$$S' = 2f$$

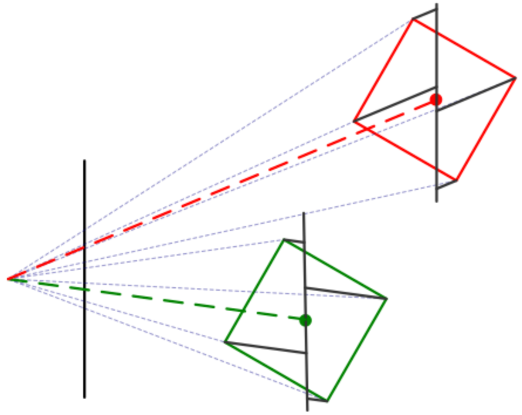
in order to achieve unit magnification.

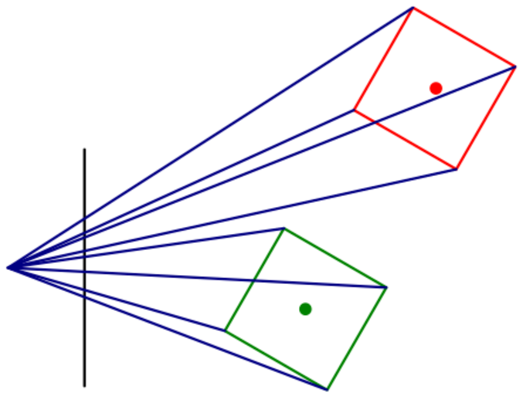
Many other types of cameras

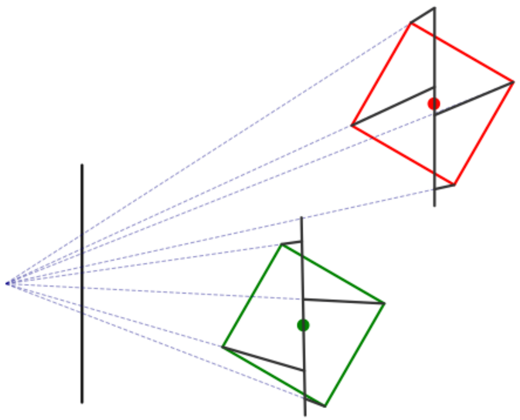

(a) 3D view


(b) orthography


(c) scaled orthography


(d) para-perspective


(e) perspective


(f) object-centered

Geometric camera calibration

Geometric camera calibration

Given a set of matched points…

$$\{\mathbf{X}_i, \boldsymbol{x}_i\}$$

point in 3D space | point in the image

… and camera model…

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

camera matrix

… estimate the camera matrix

$$\mathbf{P}$$

Mapping between 3D point and image points

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

Mapping between 3D point and image points

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

Simplify notation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} — & \boldsymbol{p}_1^\top & — \\ — & \boldsymbol{p}_2^\top & — \\ — & \boldsymbol{p}_3^\top & — \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

Mapping between 3D point and image points

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

Simplify notation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} - & \boldsymbol{p}_1^\top & - \\ - & \boldsymbol{p}_2^\top & - \\ - & \boldsymbol{p}_3^\top & - \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

Heterogeneous coordinates

$$x' = \frac{\boldsymbol{p}_1^\top \boldsymbol{X}}{\boldsymbol{p}_3^\top \boldsymbol{X}} \qquad y' = \frac{\boldsymbol{p}_2^\top \boldsymbol{X}}{\boldsymbol{p}_3^\top \boldsymbol{X}}$$

(non-linear relation between coordinates)

Linearize relations with algebraic manipulation

$$\boldsymbol{p}_2^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} y' = 0$$

$$\boldsymbol{p}_1^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} x' = 0$$

Linearize relations with algebraic manipulation

$$\boldsymbol{p}_2^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} y' = 0$$

$$\boldsymbol{p}_1^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} x' = 0$$

In matrix form … $$\begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{0} & -x'\boldsymbol{X}^\top \\ \boldsymbol{0} & \boldsymbol{X}^\top & -y'\boldsymbol{X}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix} = \boldsymbol{0}$$

Linearize relations with algebraic manipulation

$$\boldsymbol{p}_2^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} y' = 0$$

$$\boldsymbol{p}_1^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} x' = 0$$

In matrix form ...

$$\begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{0} & -x'\boldsymbol{X}^\top \\ \boldsymbol{0} & \boldsymbol{X}^\top & -y'\boldsymbol{X}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix} = \boldsymbol{0}$$

Setup homogeneous linear system for multiple point correspondences

For N points ...

$$\begin{bmatrix} \boldsymbol{X}_1^\top & \boldsymbol{0} & -x'\boldsymbol{X}_1^\top \\ \boldsymbol{0} & \boldsymbol{X}_1^\top & -y'\boldsymbol{X}_1^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_N^\top & \boldsymbol{0} & -x'\boldsymbol{X}_N^\top \\ \boldsymbol{0} & \boldsymbol{X}_N^\top & -y'\boldsymbol{X}_N^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix} = \boldsymbol{0}$$

Linearize relations with algebraic manipulation

$$\boldsymbol{p}_2^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} y' = 0$$

$$\boldsymbol{p}_1^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} x' = 0$$

In matrix form …

$$\begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{0} & -x'\boldsymbol{X}^\top \\ \boldsymbol{0} & \boldsymbol{X}^\top & -y'\boldsymbol{X}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix} = \boldsymbol{0}$$

Setup homogeneous linear system for multiple point correspondences

For N points …

$$\boxed{\begin{bmatrix} \boldsymbol{X}_1^\top & \boldsymbol{0} & -x'\boldsymbol{X}_1^\top \\ \boldsymbol{0} & \boldsymbol{X}_1^\top & -y'\boldsymbol{X}_1^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_N^\top & \boldsymbol{0} & -x'\boldsymbol{X}_N^\top \\ \boldsymbol{0} & \boldsymbol{X}_N^\top & -y'\boldsymbol{X}_N^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix} = \boldsymbol{0}}$$

Solve using SVD!

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array}\right]$$

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array} \right]$$

We want to decompose the matrix in intrinsics and extrinsics

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

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array} \right]$$

We want to decompose the matrix in intrinsics and extrinsics

$$\begin{aligned} \mathbf{P} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ &= \mathbf{K}[\mathbf{R}| - \mathbf{R}\mathbf{c}] \\ &= [\mathbf{M}| - \mathbf{M}\mathbf{c}] \end{aligned}$$

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array}\right]$$

We want to decompose the matrix in intrinsics and extrinsics

$$\begin{aligned} \mathbf{P} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ &= \mathbf{K}[\mathbf{R}| - \mathbf{Rc}] \\ &= [\mathbf{M}| - \mathbf{Mc}] \end{aligned}$$

Find the camera center **c**

$$\mathbf{Pc} = \mathbf{0}$$

Find intrinsic **K** and rotation **R**

$$\mathbf{M} = \mathbf{KR}$$

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array} \right]$$

We want to decompose the matrix in intrinsics and extrinsics

$$\begin{aligned} \mathbf{P} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ &= \mathbf{K}[\mathbf{R}| - \mathbf{Rc}] \\ &= [\mathbf{M}| - \mathbf{Mc}] \end{aligned}$$

Find the camera center **c**

$$\mathbf{Pc} = \mathbf{0}$$

SVD of P

c is the singular vector corresponding to the smallest singular value

Find intrinsic **K** and rotation **R**

$$\mathbf{M} = \mathbf{KR}$$

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array}\right]$$

We want to decompose the matrix in intrinsics and extrinsics

$$\begin{aligned} \mathbf{P} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ &= \mathbf{K}[\mathbf{R}| - \mathbf{R}\mathbf{c}] \\ &= [\mathbf{M}| - \mathbf{M}\mathbf{c}] \end{aligned}$$

Find the camera center **c**

$$\mathbf{P}\mathbf{c} = \mathbf{0}$$

SVD of **P**

c is the singular vector corresponding to the smallest singular value

Find intrinsic **K** and rotation **R**

$$\mathbf{M} = \mathbf{K}\mathbf{R}$$

upper triangular orthogonal

We have computed the camera matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{array}\right]$$

We want to decompose the matrix in intrinsics and extrinsics

$$\begin{aligned} \mathbf{P} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ &= \mathbf{K}[\mathbf{R}| - \mathbf{Rc}] \\ &= [\mathbf{M}| - \mathbf{Mc}] \end{aligned}$$

Find the camera center **c**

$$\mathbf{Pc} = \mathbf{0}$$

SVD of **P**

c is the singular vector corresponding to the smallest singular value

Find intrinsic **K** and rotation **R**

$$\mathbf{M} = \mathbf{KR}$$

upper triangular orthogonal

QR decomposition of **M**

Modern toolboxes

Geometric camera calibration

Given a set of matched points...

$$\{\mathbf{X}_i, \boldsymbol{x}_i\}$$

point in 3D space   point in the image

Where do the matched points come from?

... and camera model...

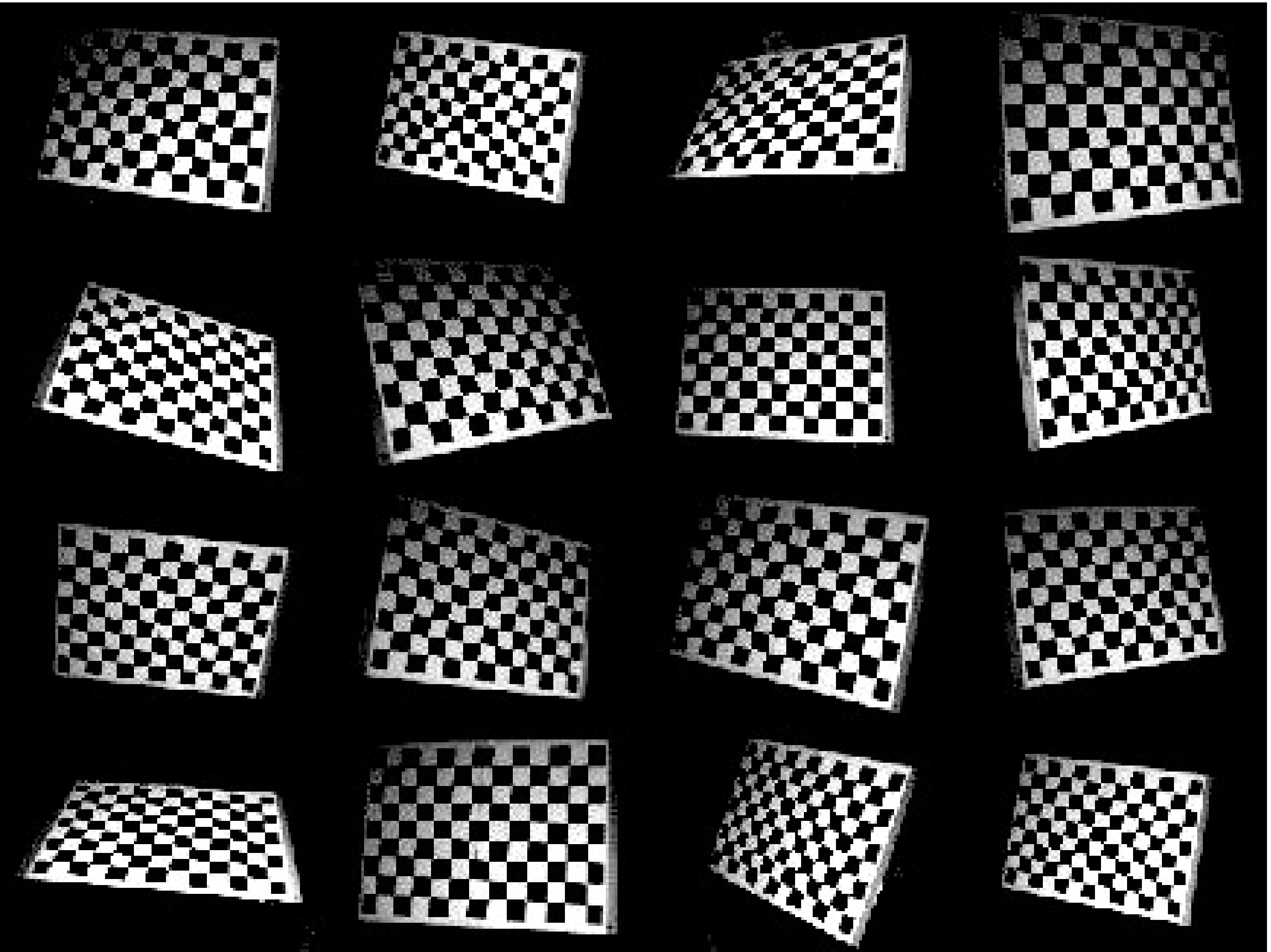
$$\boldsymbol{x} = \mathbf{P}\mathbf{X}$$

camera matrix

... estimate the camera matrix

$$\mathbf{P}$$

Nowadays: multi-plane calibration



Advantages:

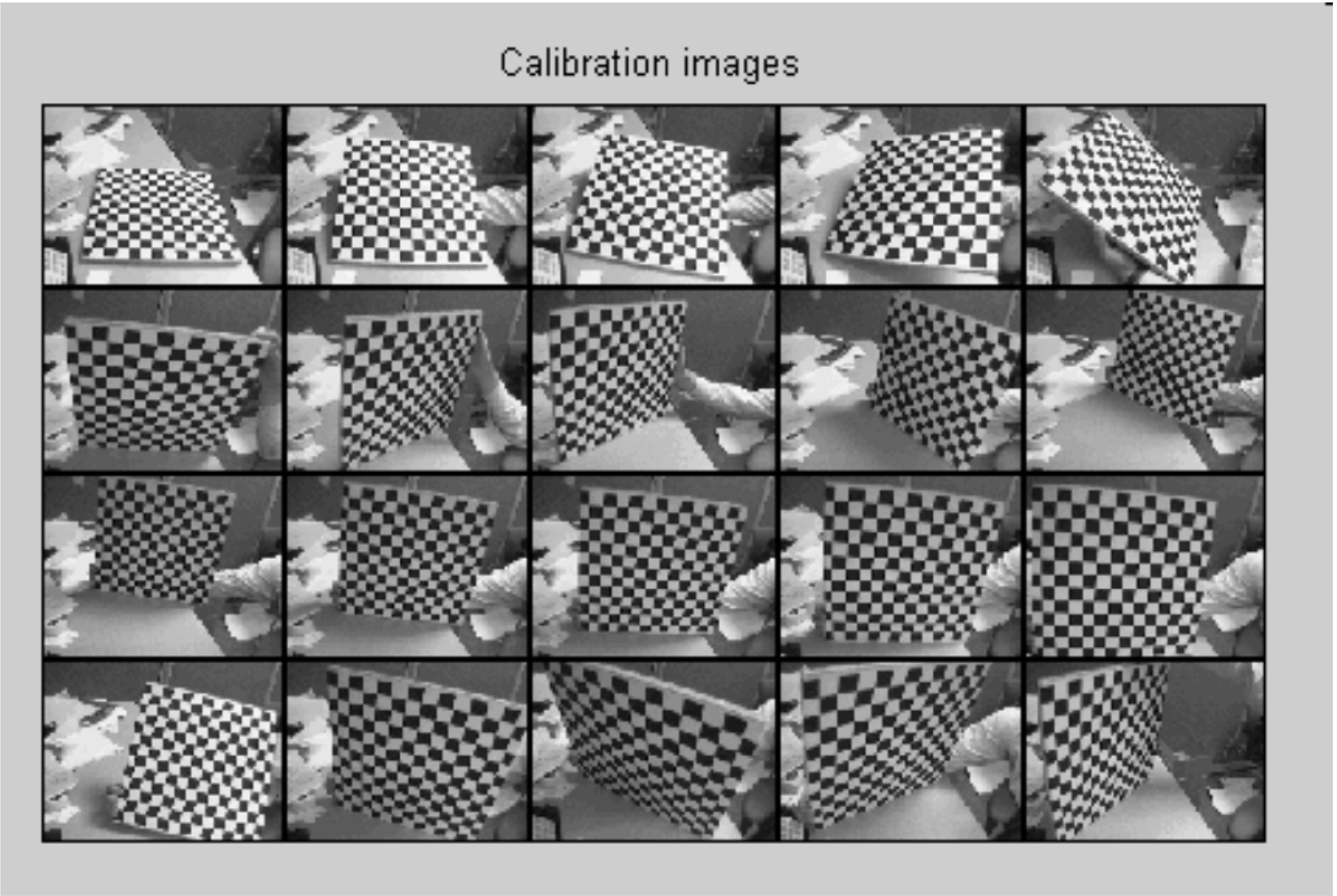
- only requires a planar checkerboard
- does not require positions/orientations
- also corrects lens radial distortion
- great toolboxes freely available!

Correcting radial distortion

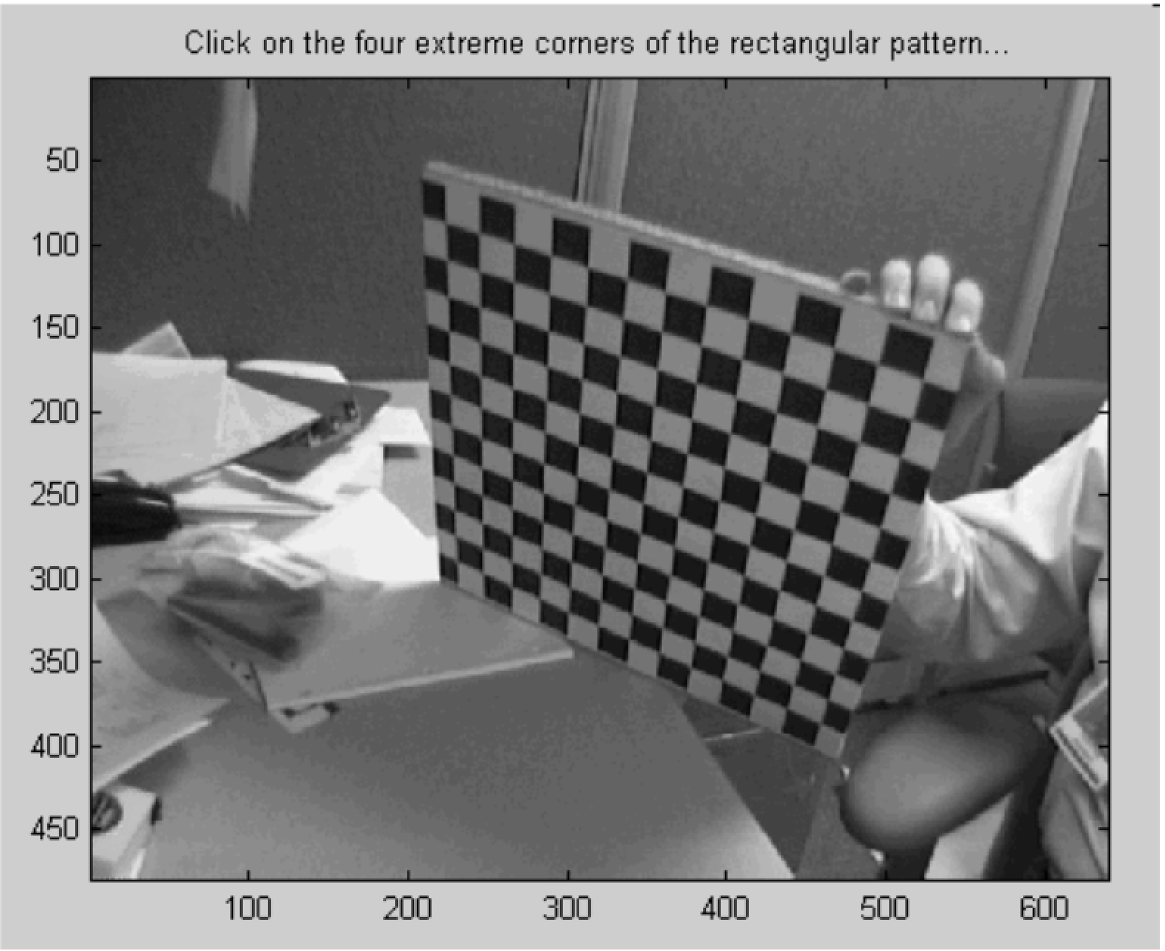
before

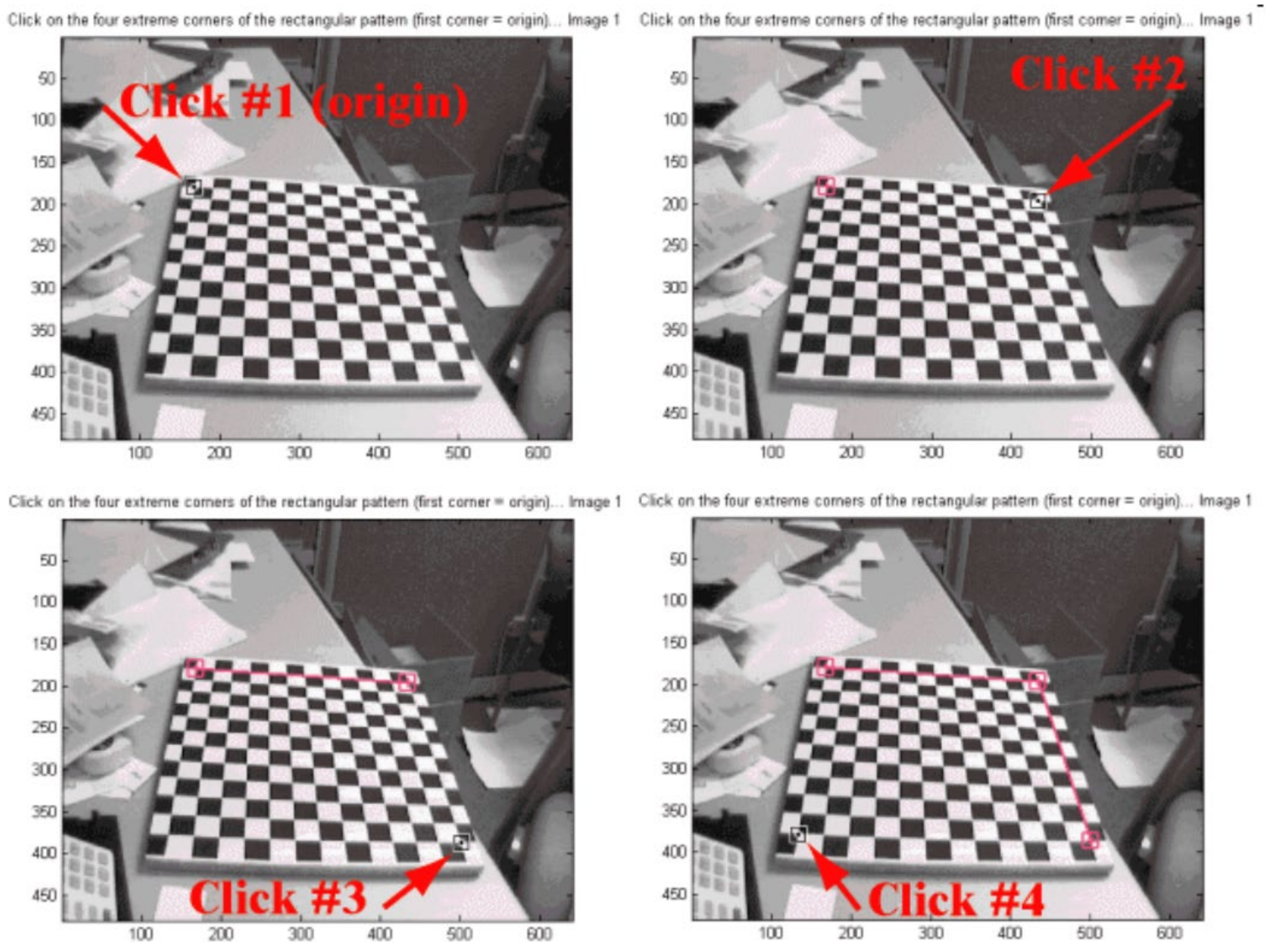
after

Step-by-step demonstration

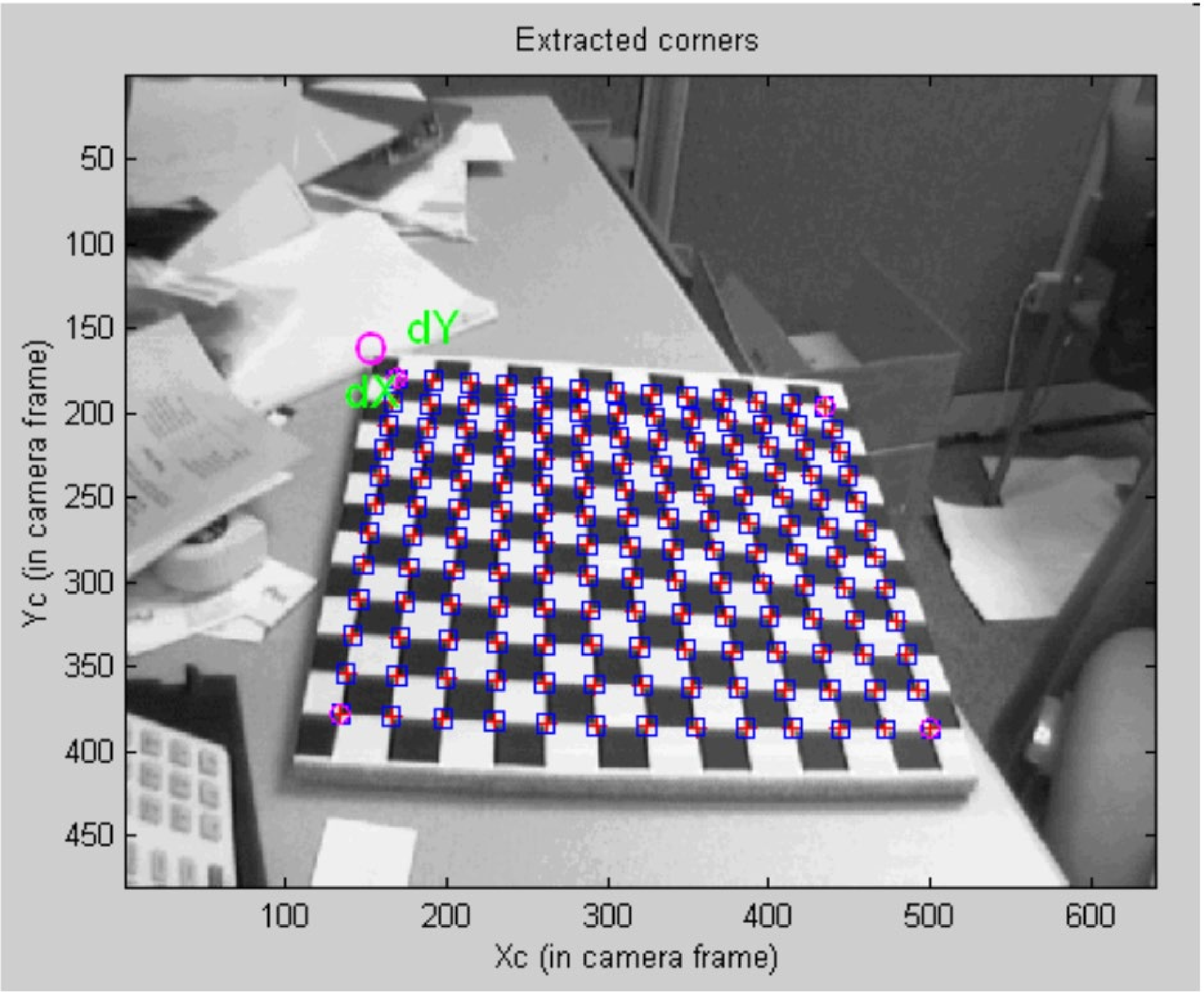


Step-by-step demonstration

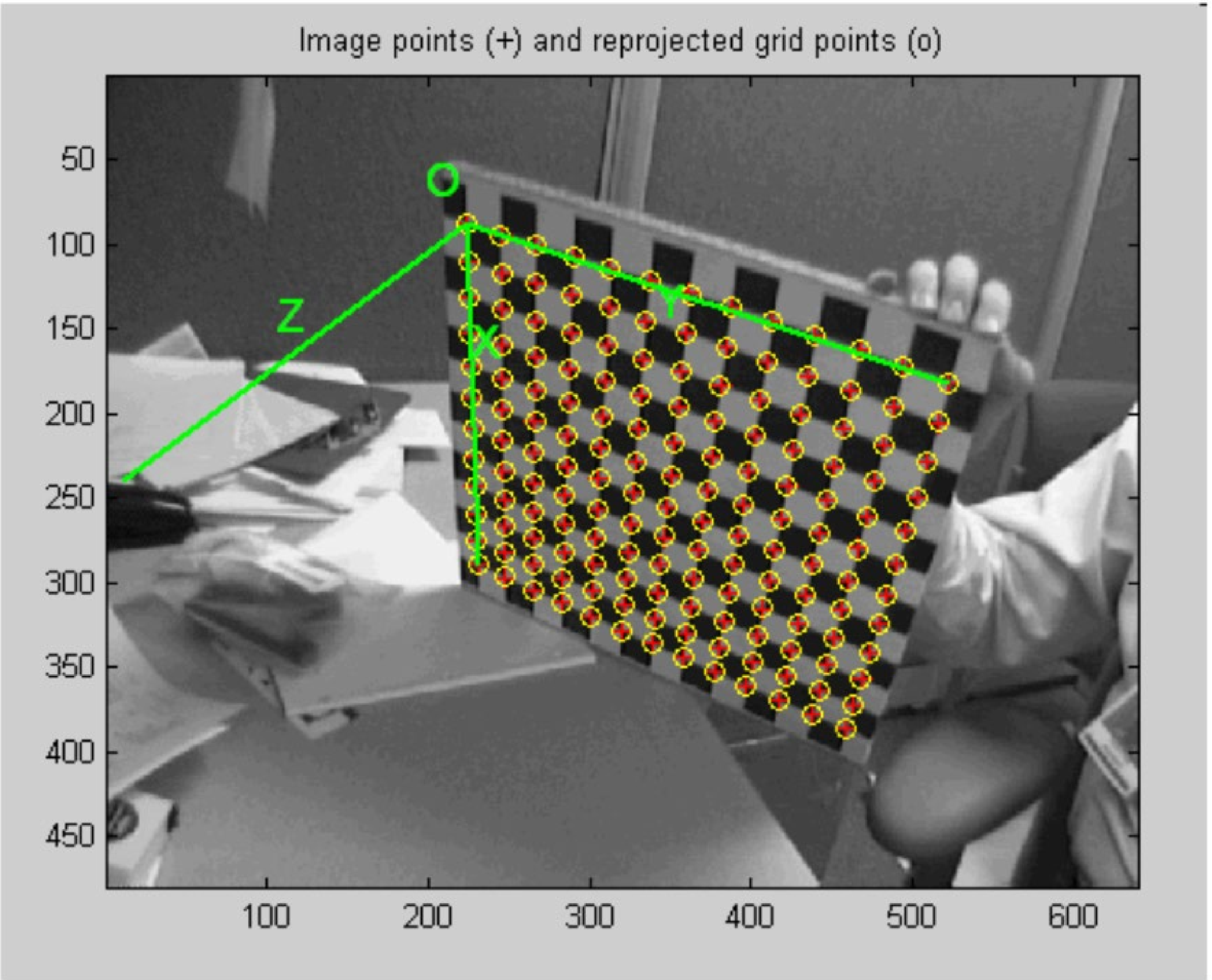




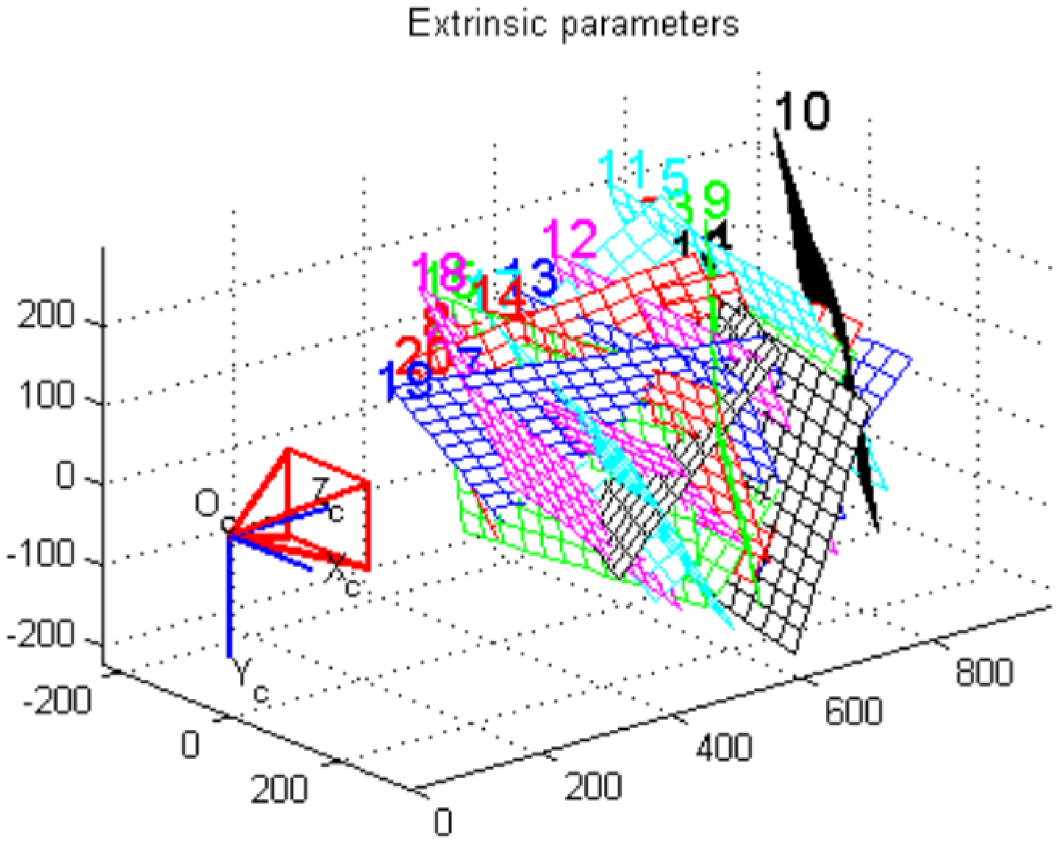
Step-by-step demonstration

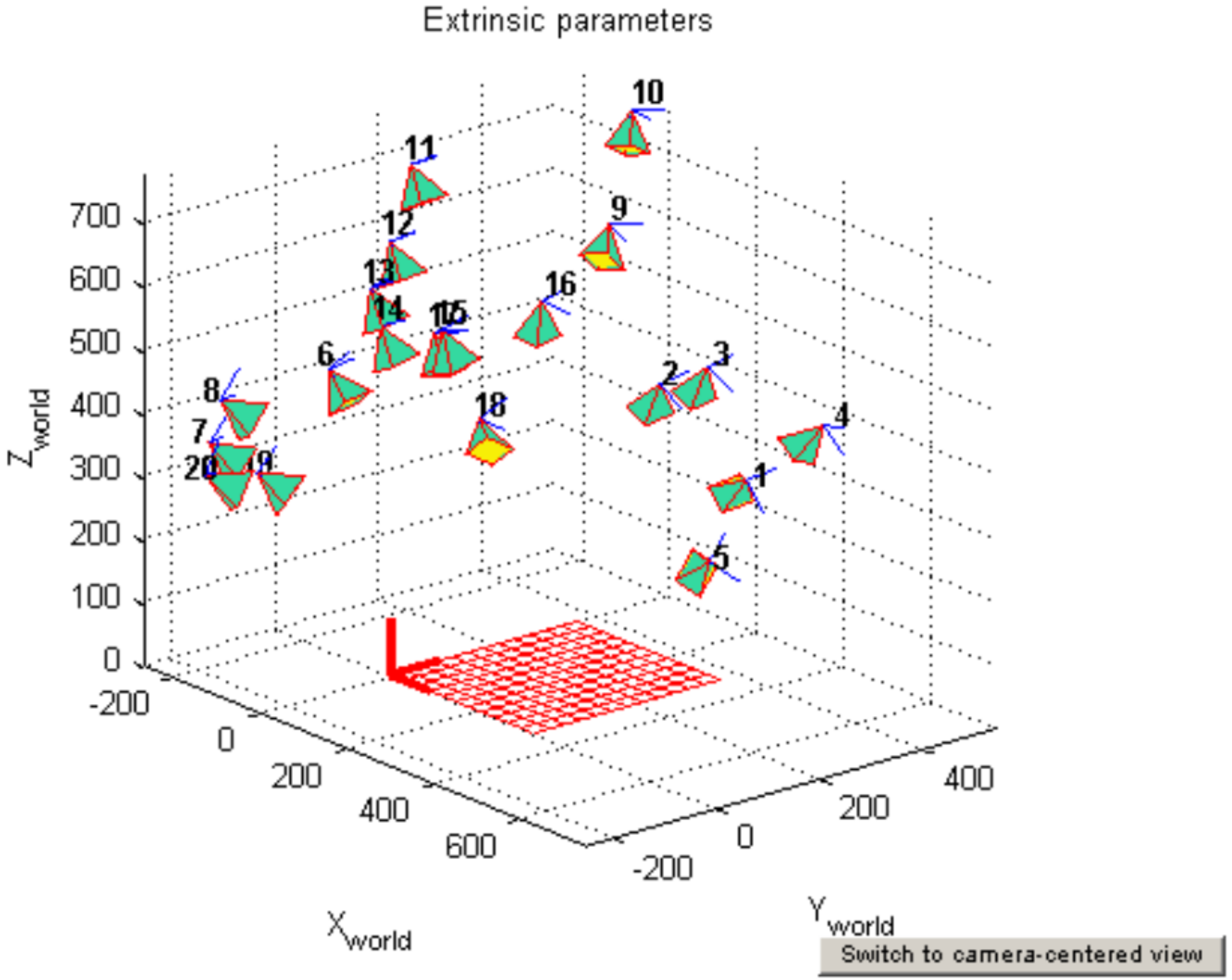


Step-by-step demonstration



Step-by-step demonstration





What does it mean to “calibrate a camera”?

What does it mean to “calibrate a camera”?

Many different ways to calibrate a camera:

- Radiometric calibration.
- Color calibration.
- Geometric calibration.
- Noise calibration.
- Lens (or aberration) calibration.

References

Basic reading:

- Szeliski textbook, Section 2.1.5, 6.2.
- Hartley and Zisserman, Chapter 6.