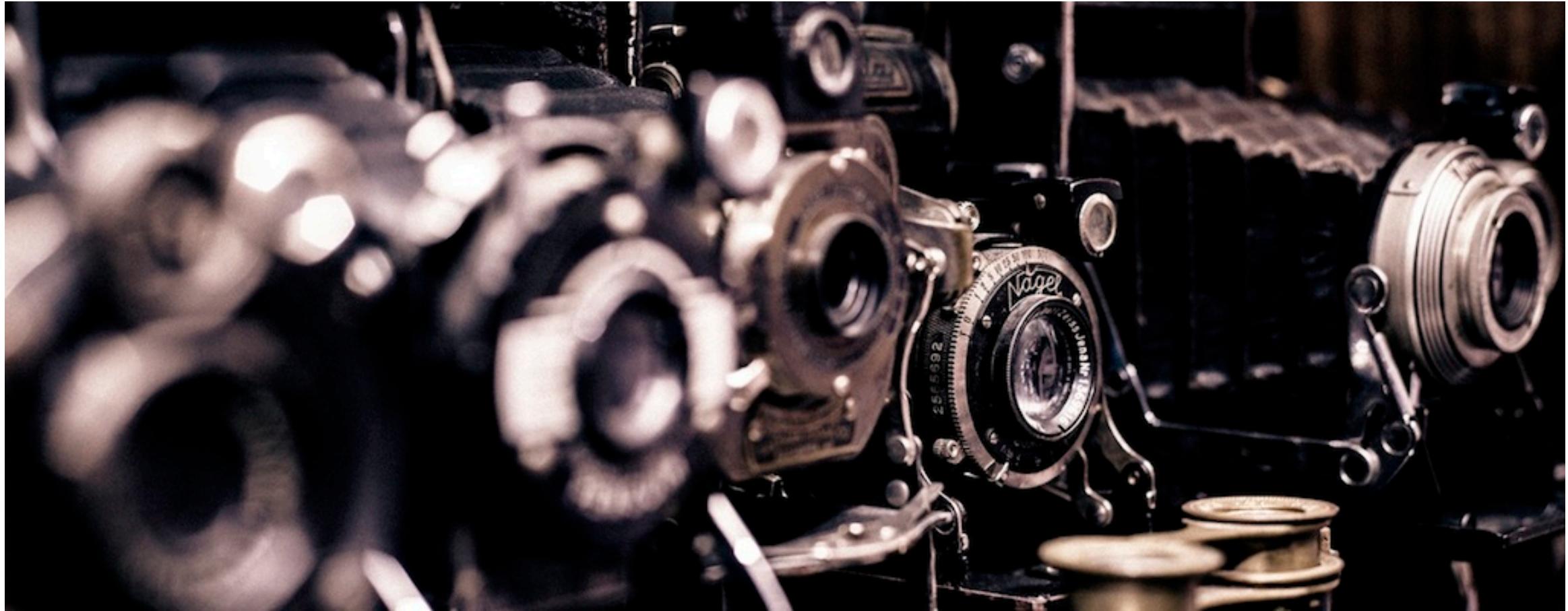


Geometric camera models



Slide credits

Most of these slides were adapted from:

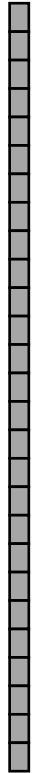
- Kris Kitani (CMU, Fall 2016).
- Ioannis Gkioulekas (CMU, Fall 2019)

Some slides inspired from:

- Fredo Durand (MIT).

Some motivational imaging experiments

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)

What would an image taken like this look like?

Bare-sensor imaging

real-world
object



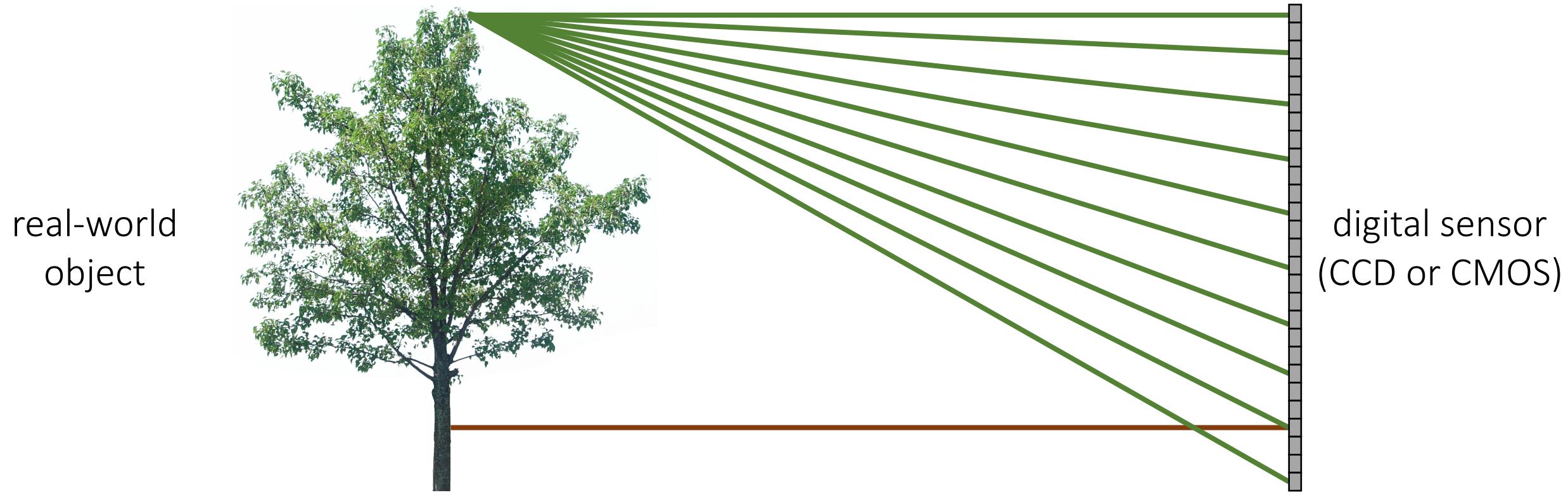
digital sensor
(CCD or CMOS)



Bare-sensor imaging

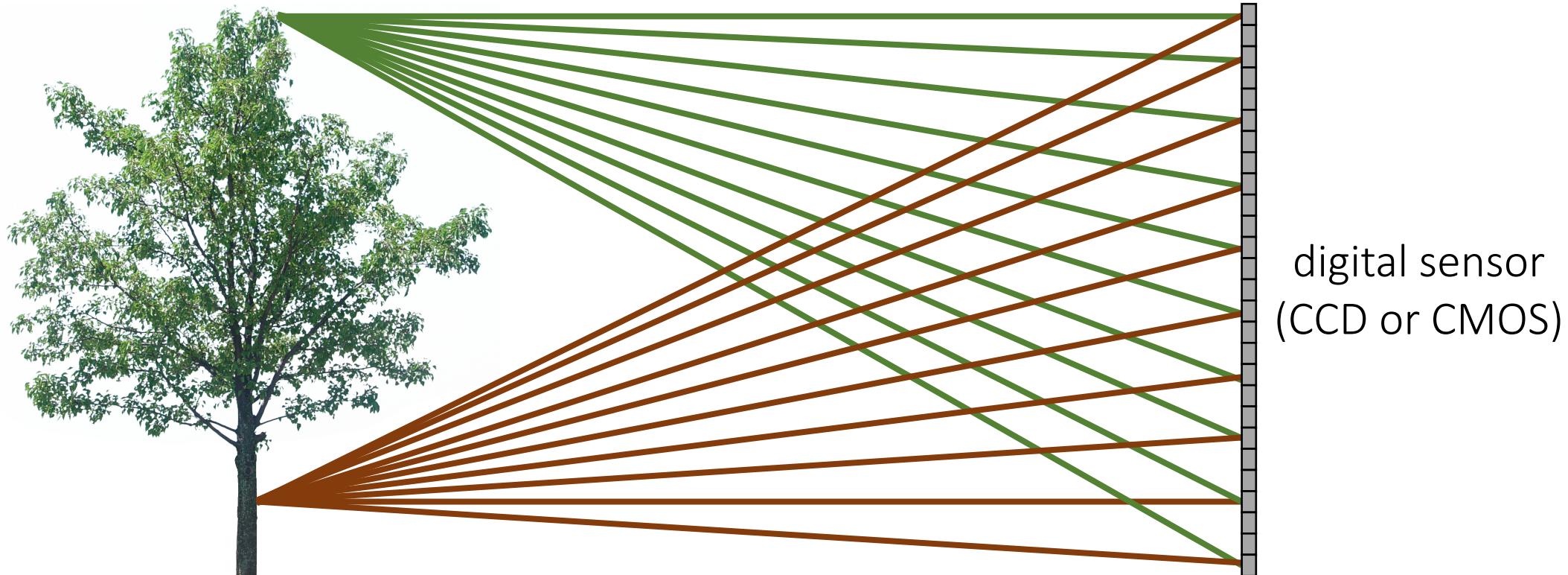


Bare-sensor imaging



Bare-sensor imaging

real-world
object



All scene points contribute to all sensor pixels

What does the
image on the
sensor look like?

Bare-sensor imaging



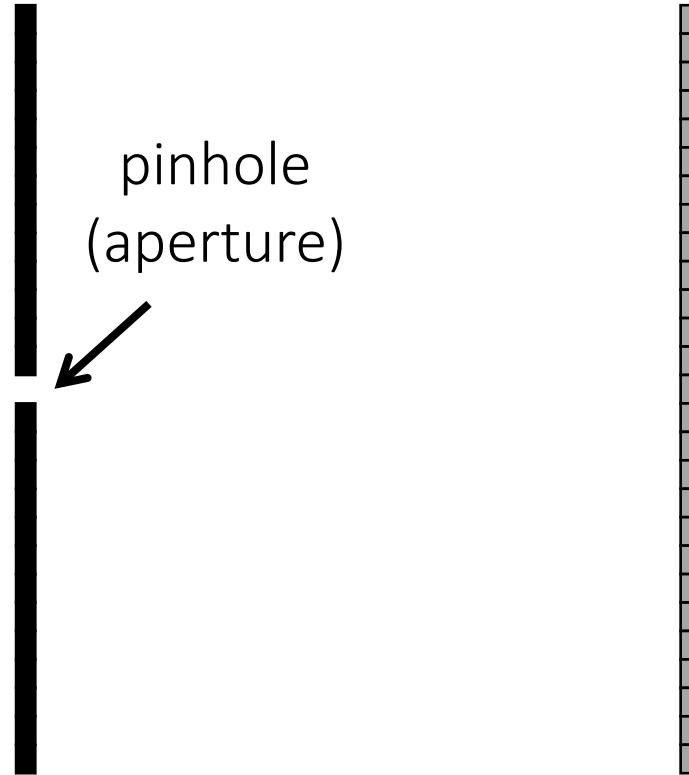
All scene points contribute to all sensor pixels

Let's add something to this scene

real-world
object

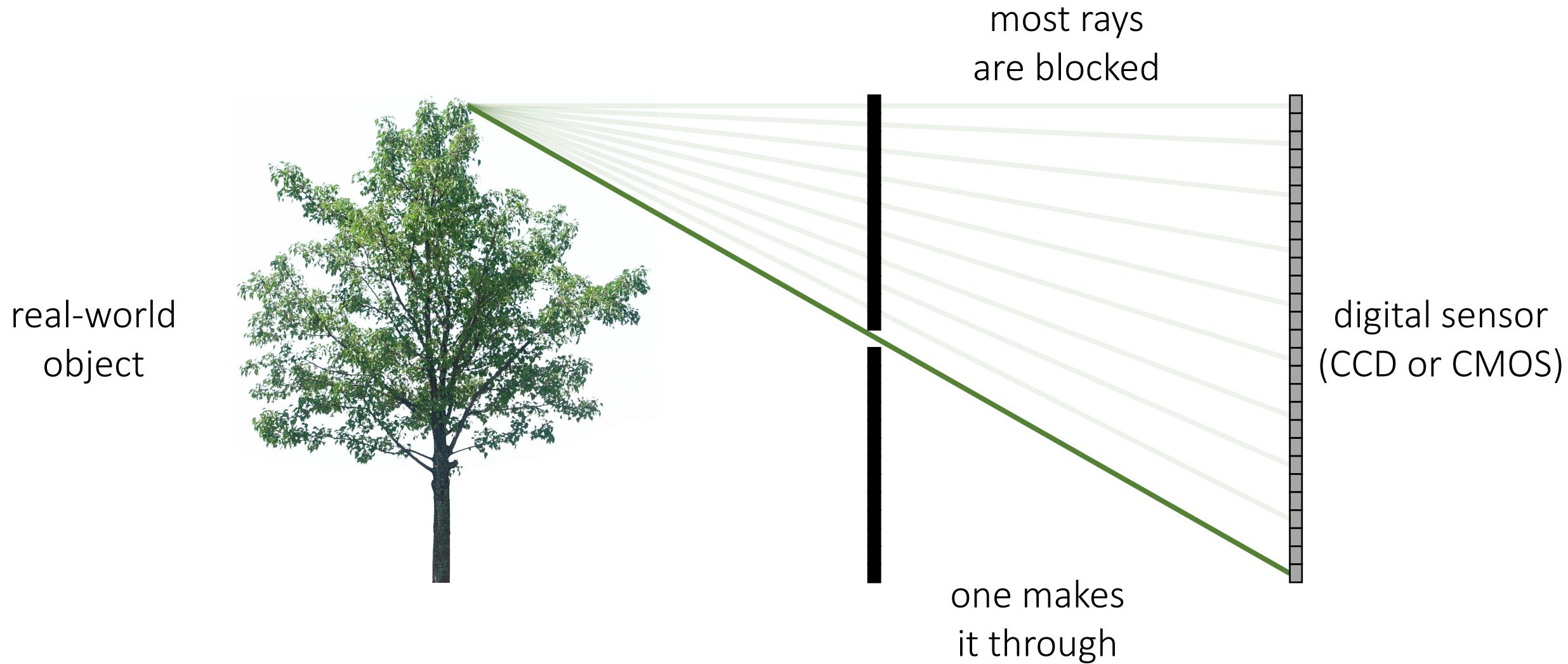


barrier (diaphragm)

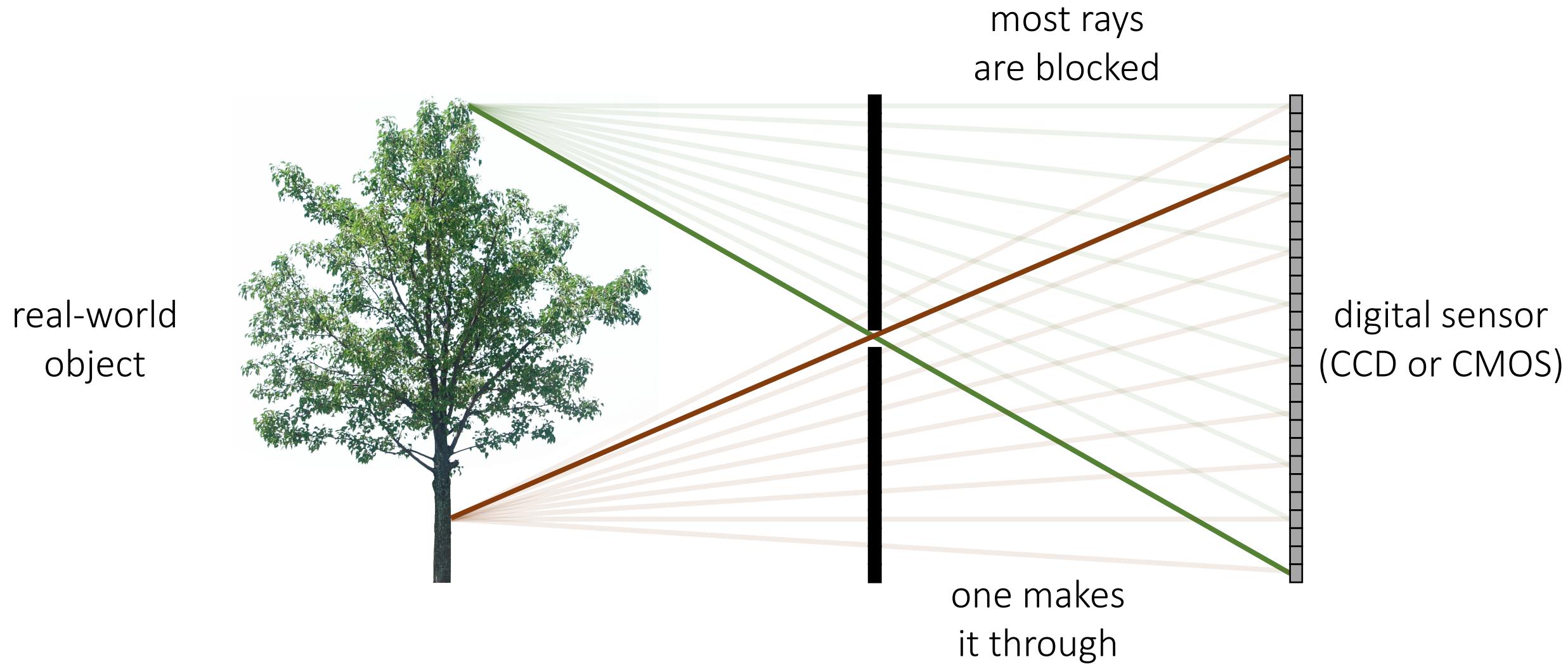


What would an image taken like this look like?

Pinhole imaging

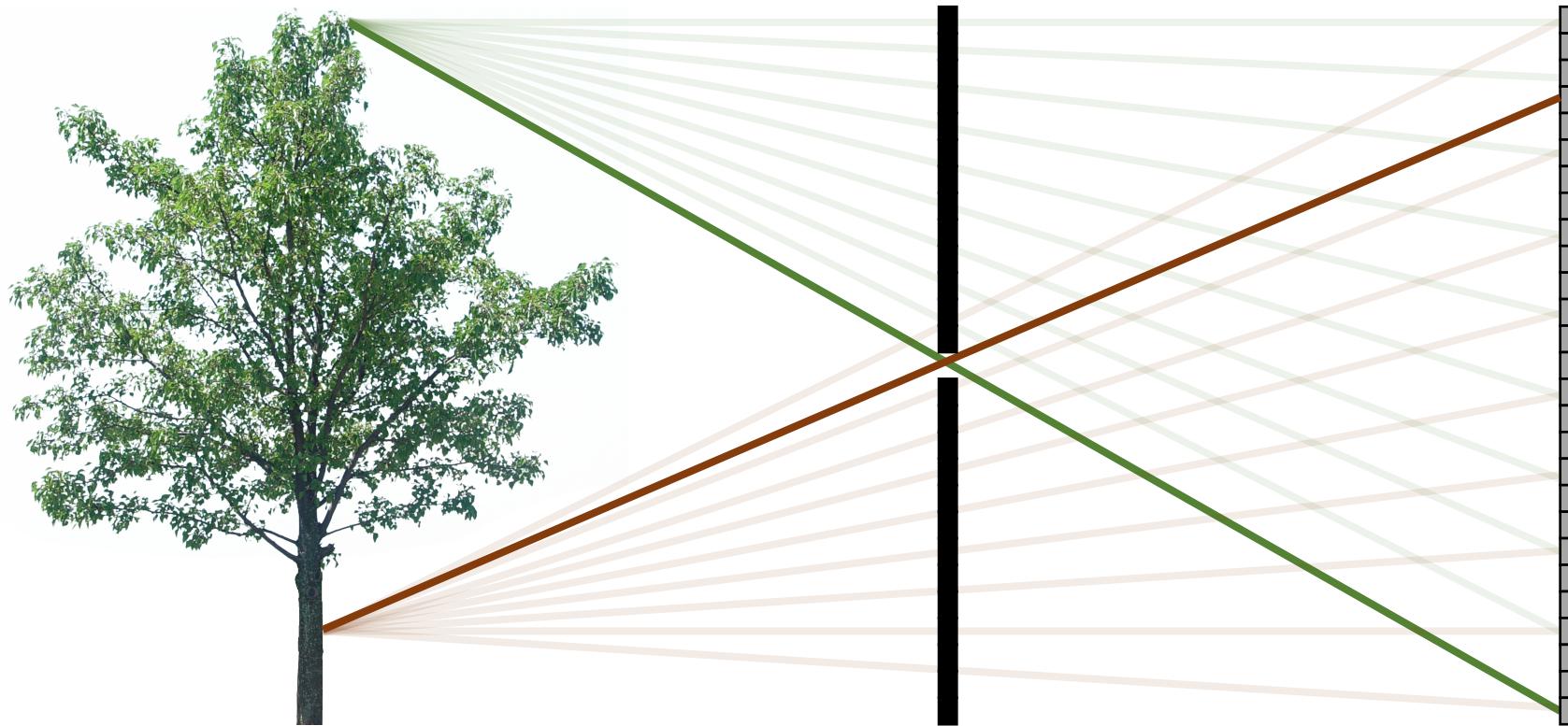


Pinhole imaging



Pinhole imaging

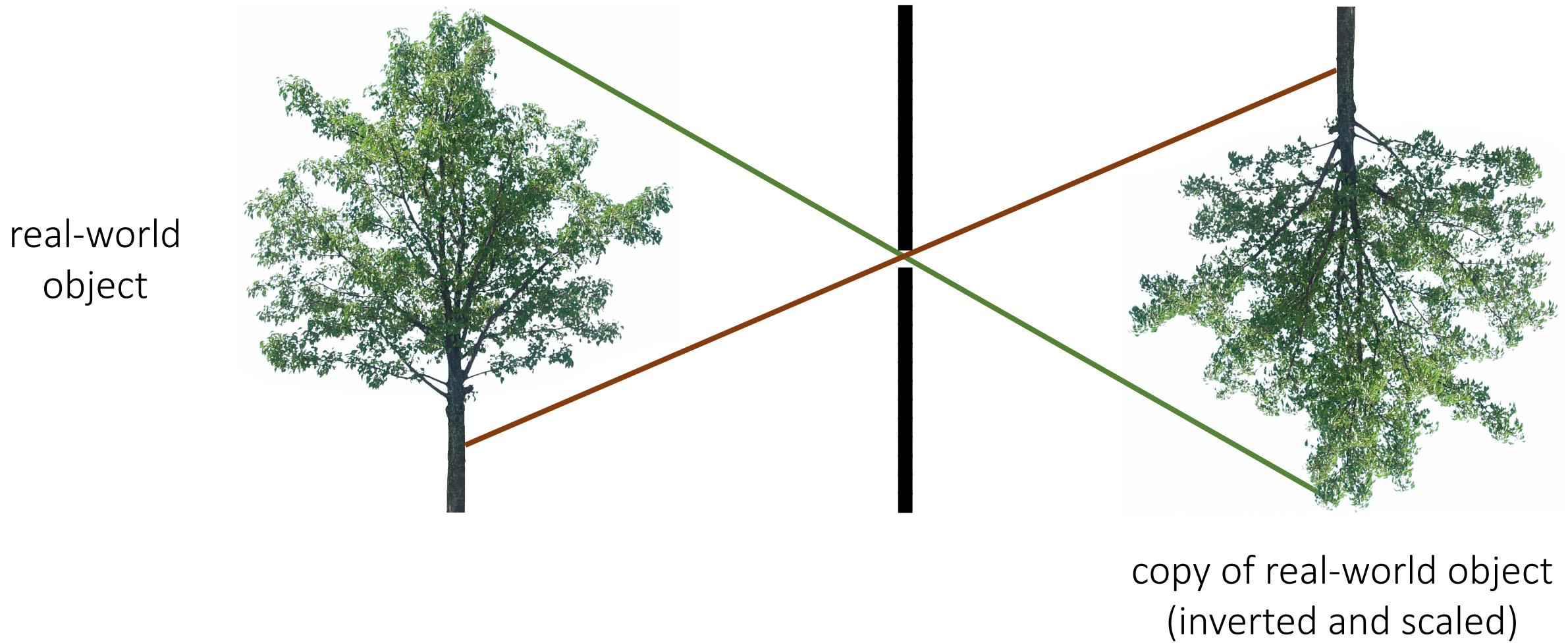
real-world
object



Each scene point contributes to only one sensor pixel

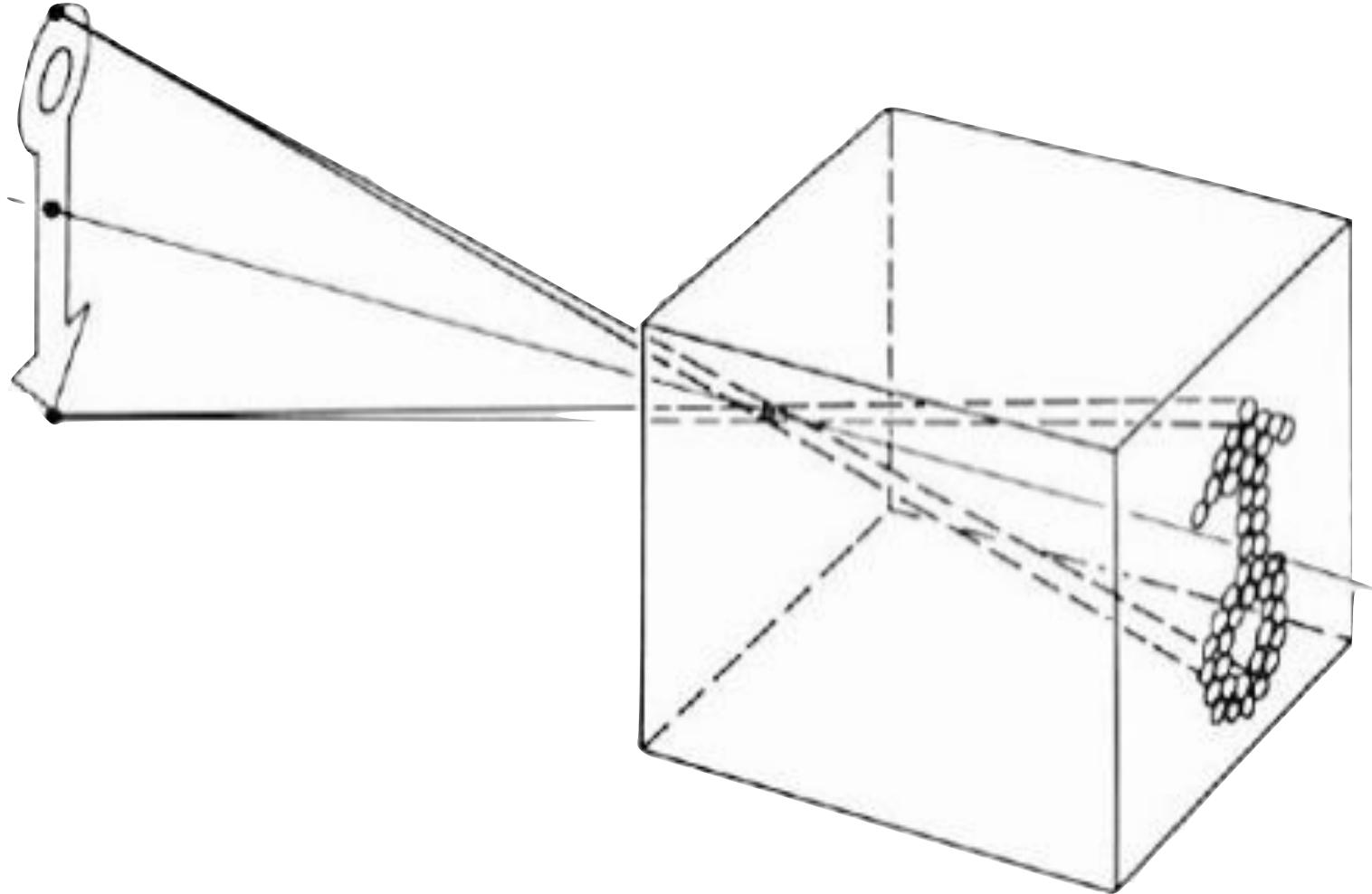
What does the
image on the
sensor look like?

Pinhole imaging



Pinhole camera

Pinhole camera a.k.a. camera obscura



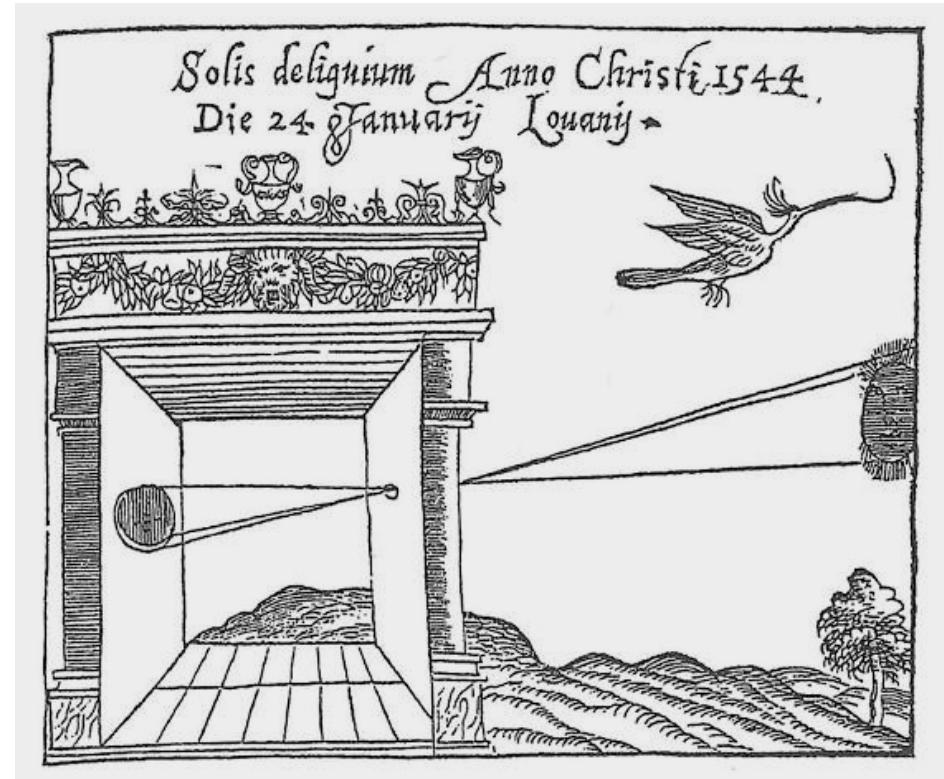
Pinhole camera a.k.a. camera obscura

First mention ...



Chinese philosopher Mozi
(470 to 390 BC)

First camera ...



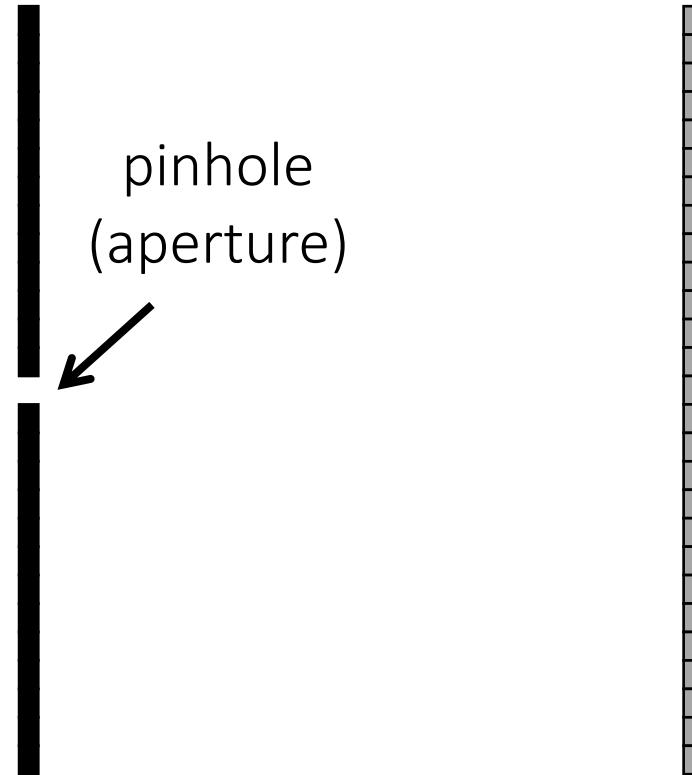
Greek philosopher Aristotle
(384 to 322 BC)

Pinhole camera terms

real-world
object



barrier (diaphragm)



digital sensor
(CCD or CMOS)

Pinhole camera terms

real-world
object



barrier (diaphragm)

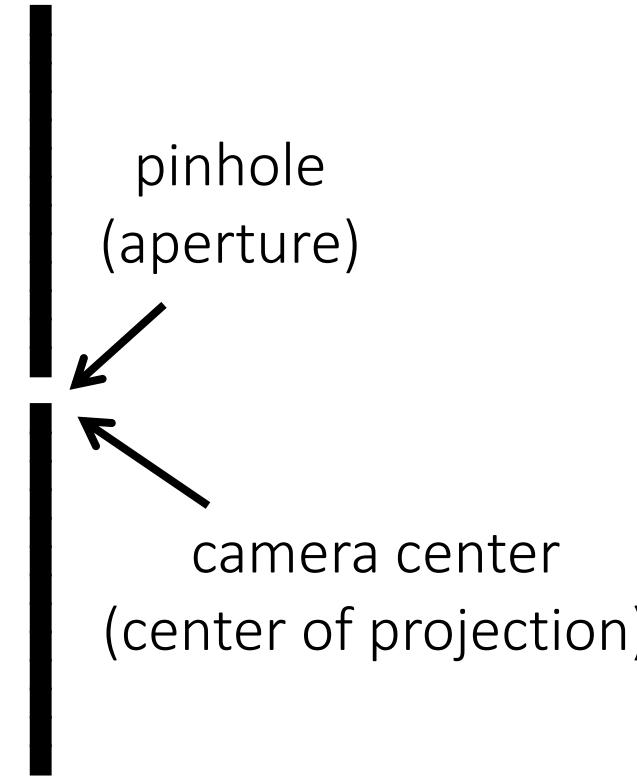
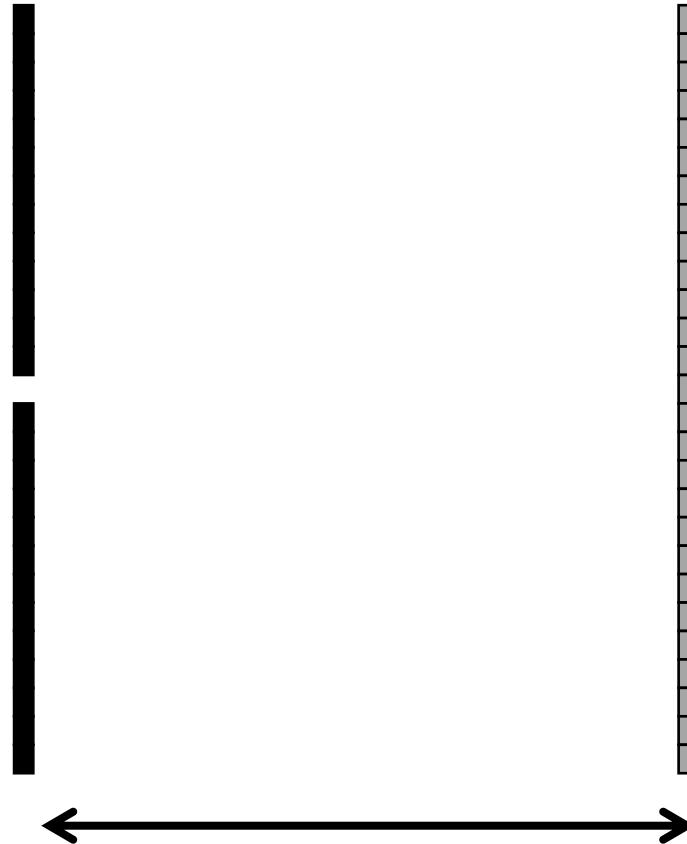


image plane

digital sensor
(CCD or CMOS)

Focal length

real-world
object

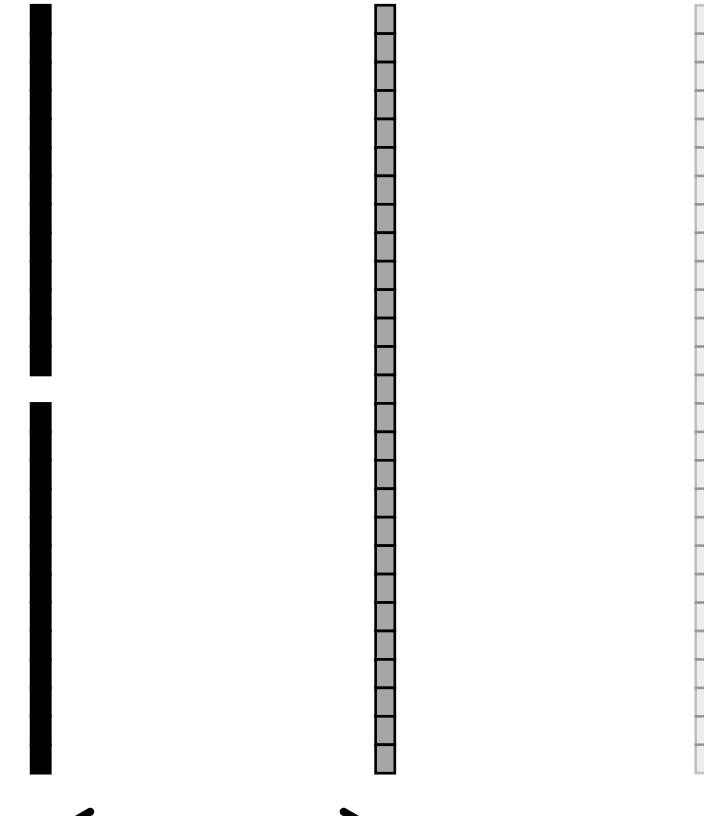


focal length f

Focal length

What happens as we change the focal length?

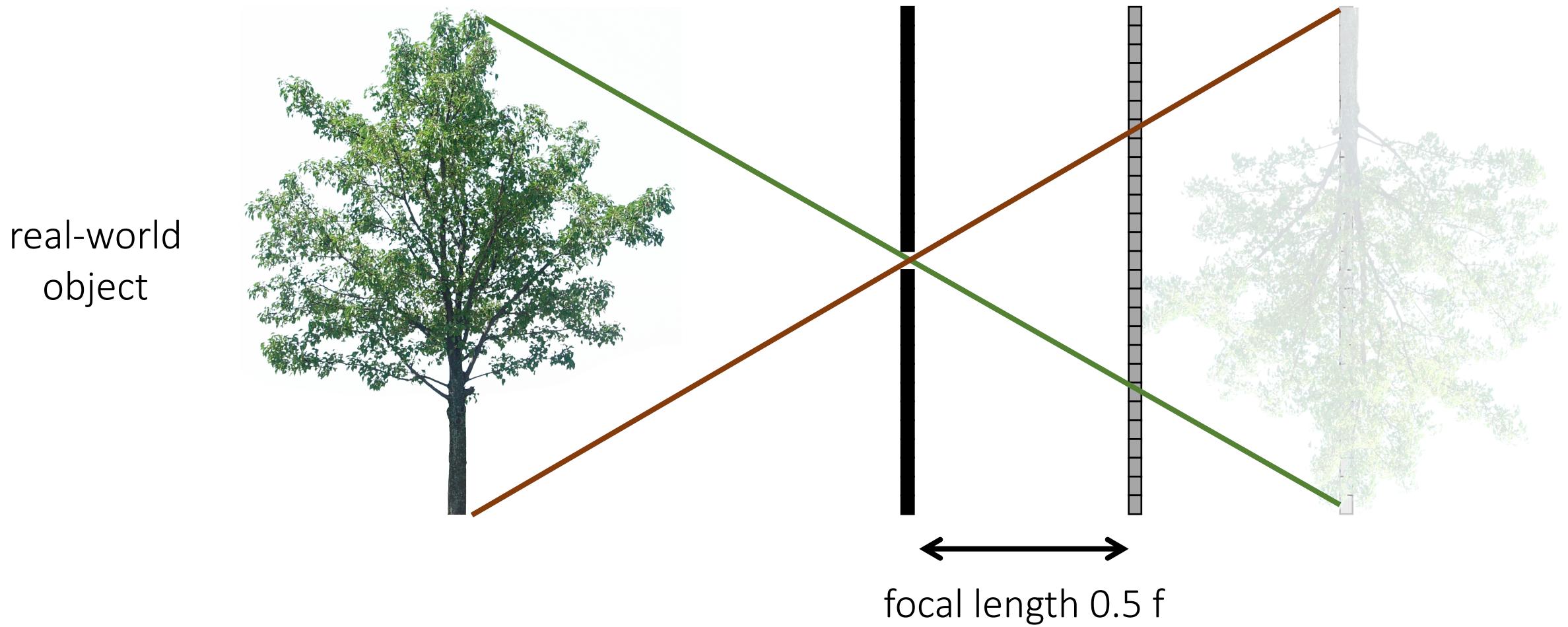
real-world
object



focal length $0.5 f$

Focal length

What happens as we change the focal length?

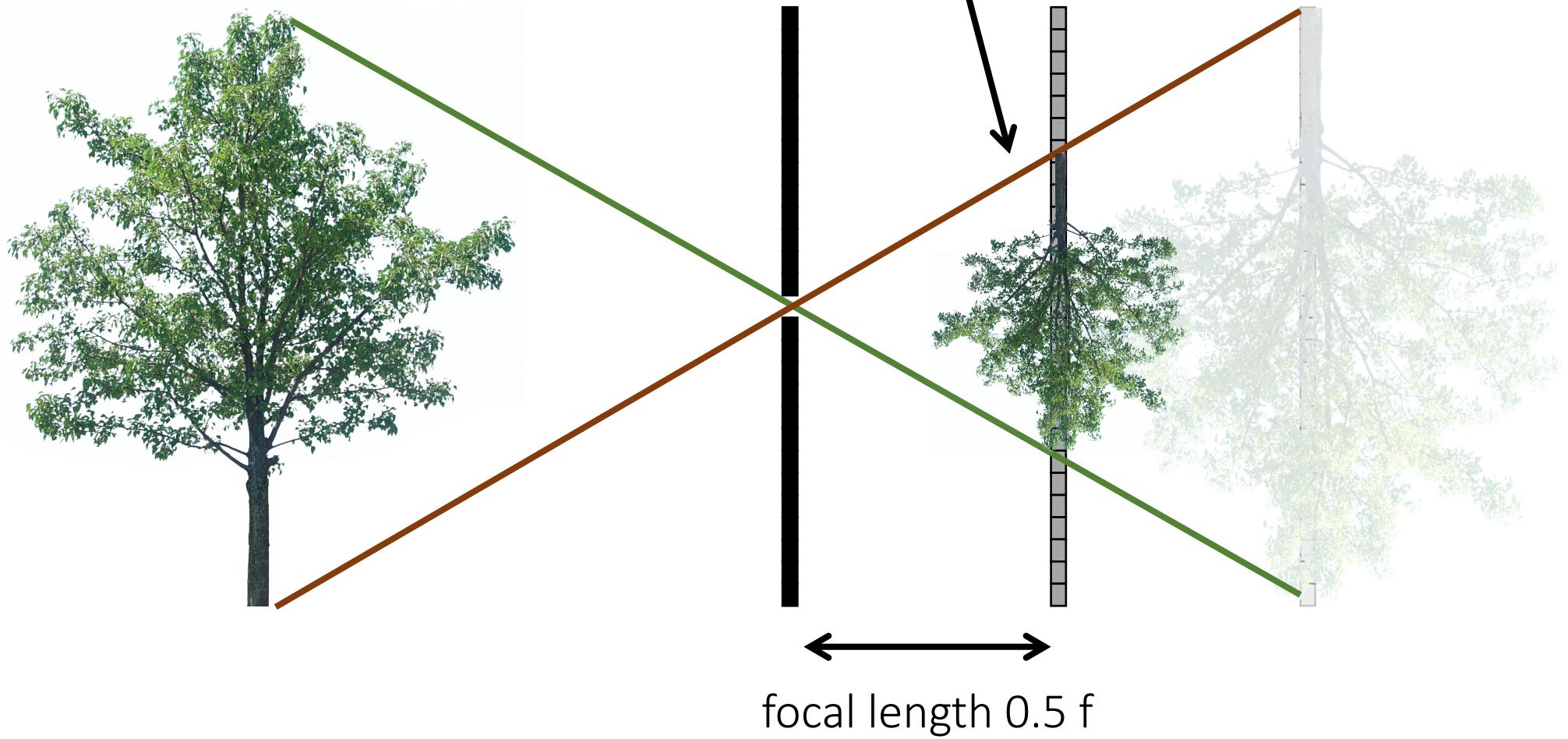


Focal length

What happens as we change the focal length?

object projection is half the size

real-world
object

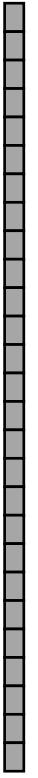


Pinhole size

real-world
object



pinhole
diameter



Ideal pinhole has infinitesimally small size

- In practice that is impossible.

Pinhole size

What happens as we change the pinhole diameter?

real-world
object



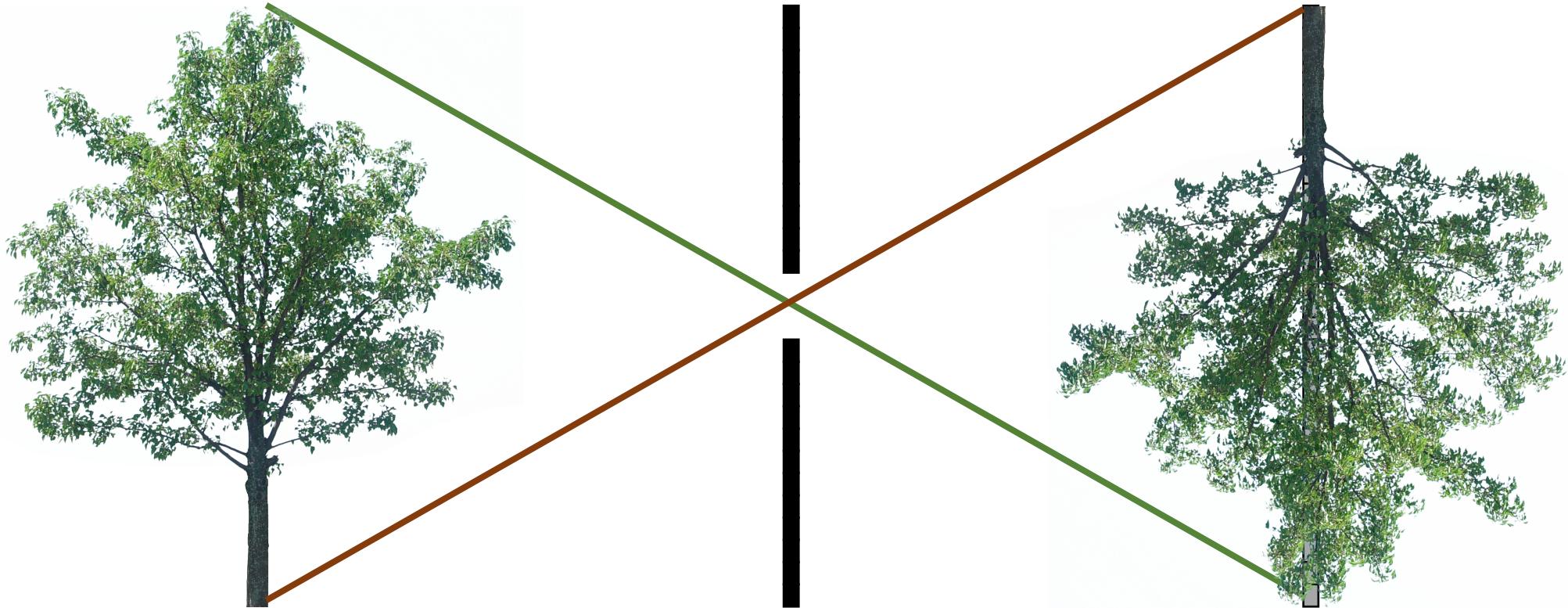
pinhole
diameter



Pinhole size

What happens as we change the pinhole diameter?

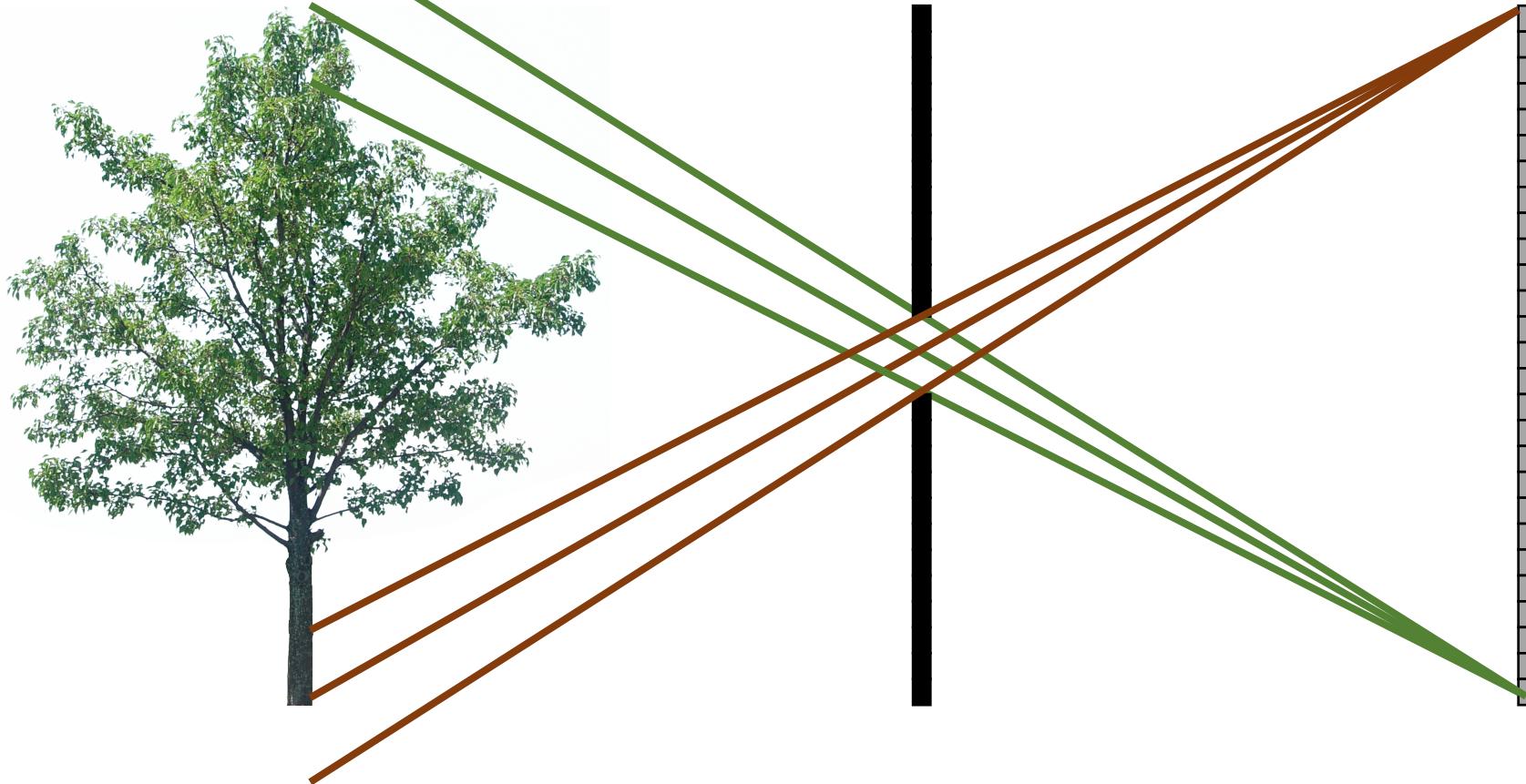
real-world
object



Pinhole size

What happens as we change the pinhole diameter?

real-world
object

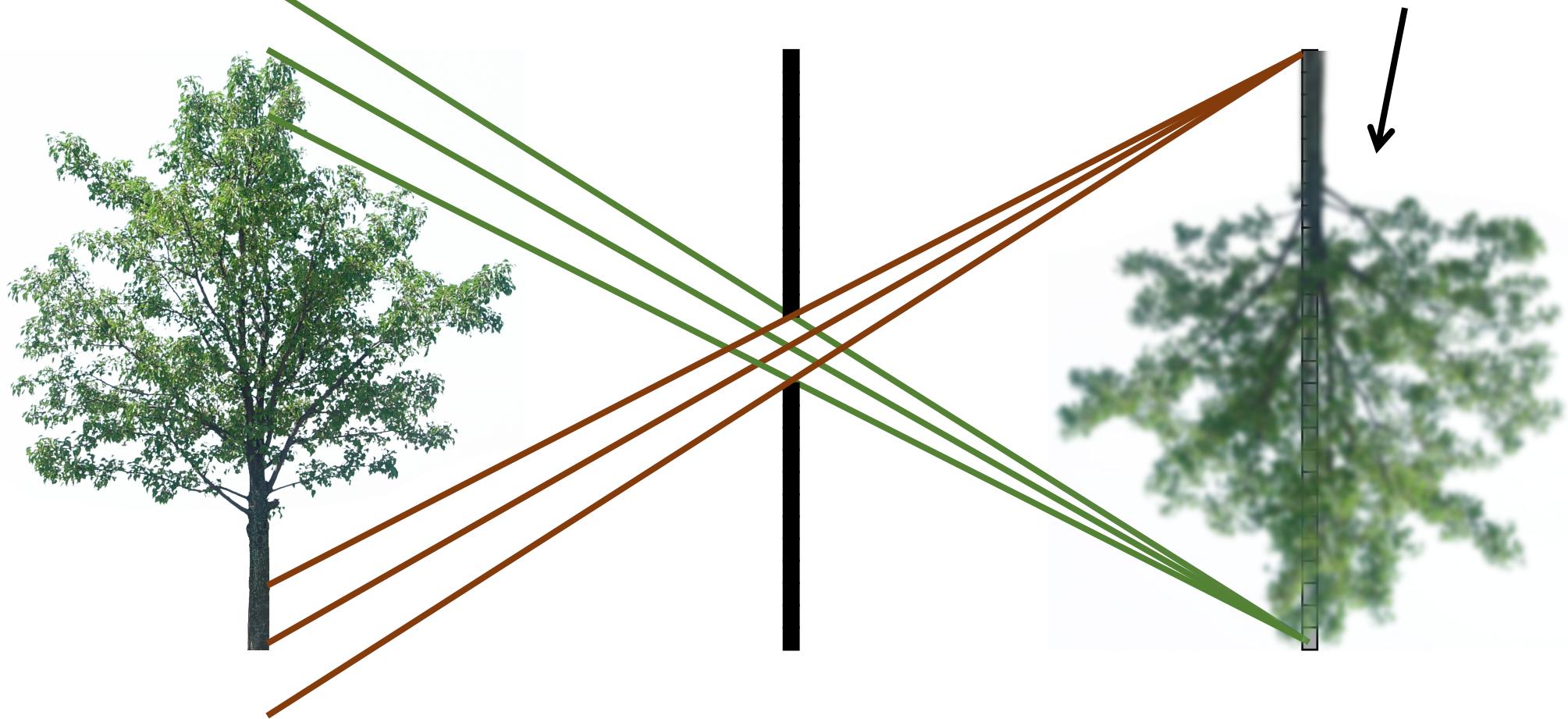


Pinhole size

What happens as we change the pinhole diameter?

object projection becomes blurrier

real-world
object

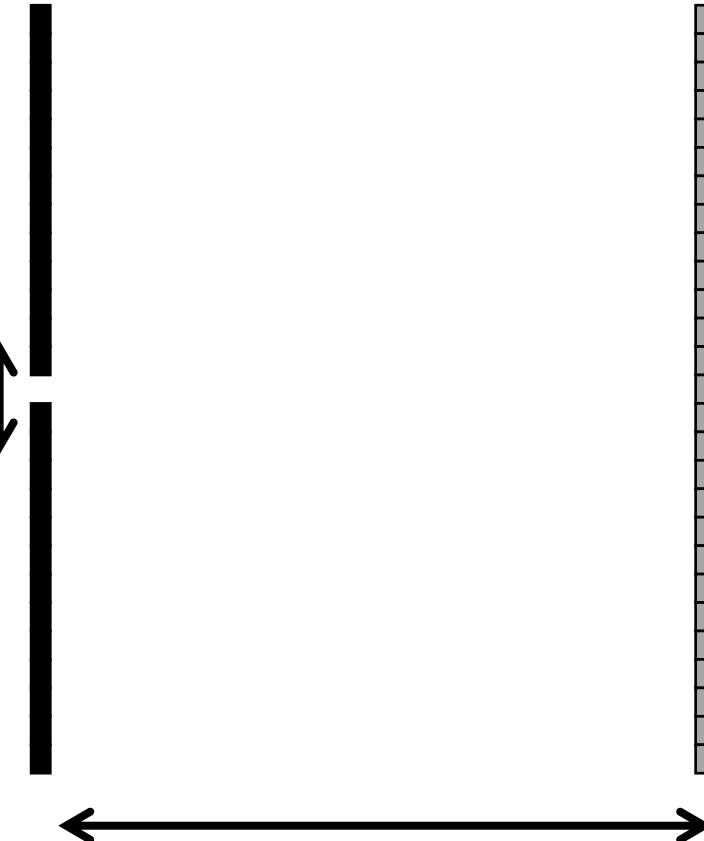


What about light efficiency?

real-world
object



pinhole
diameter



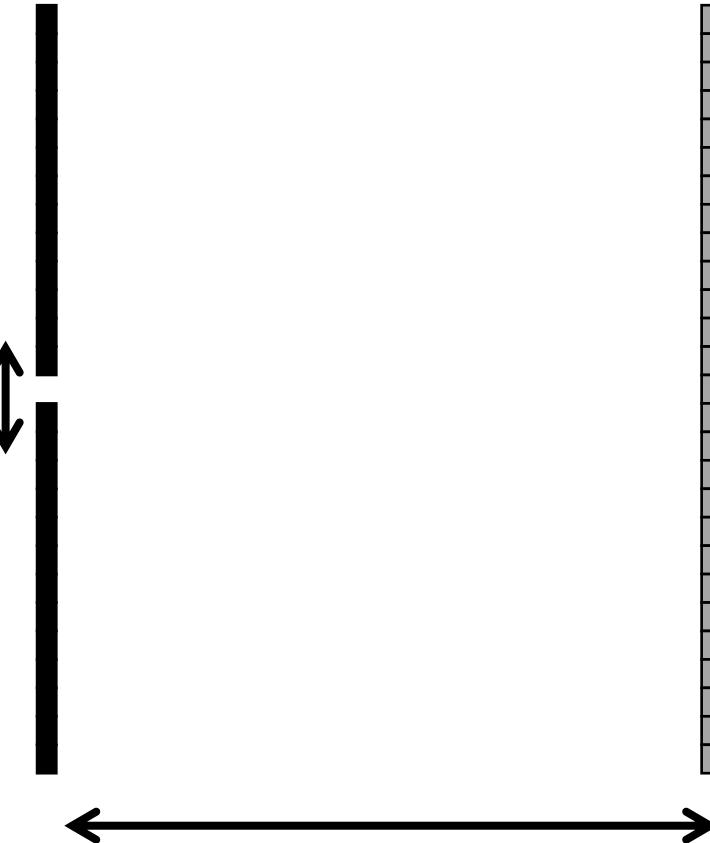
- What is the effect of doubling the pinhole diameter?
- What is the effect of doubling the focal length?

What about light efficiency?

real-world
object

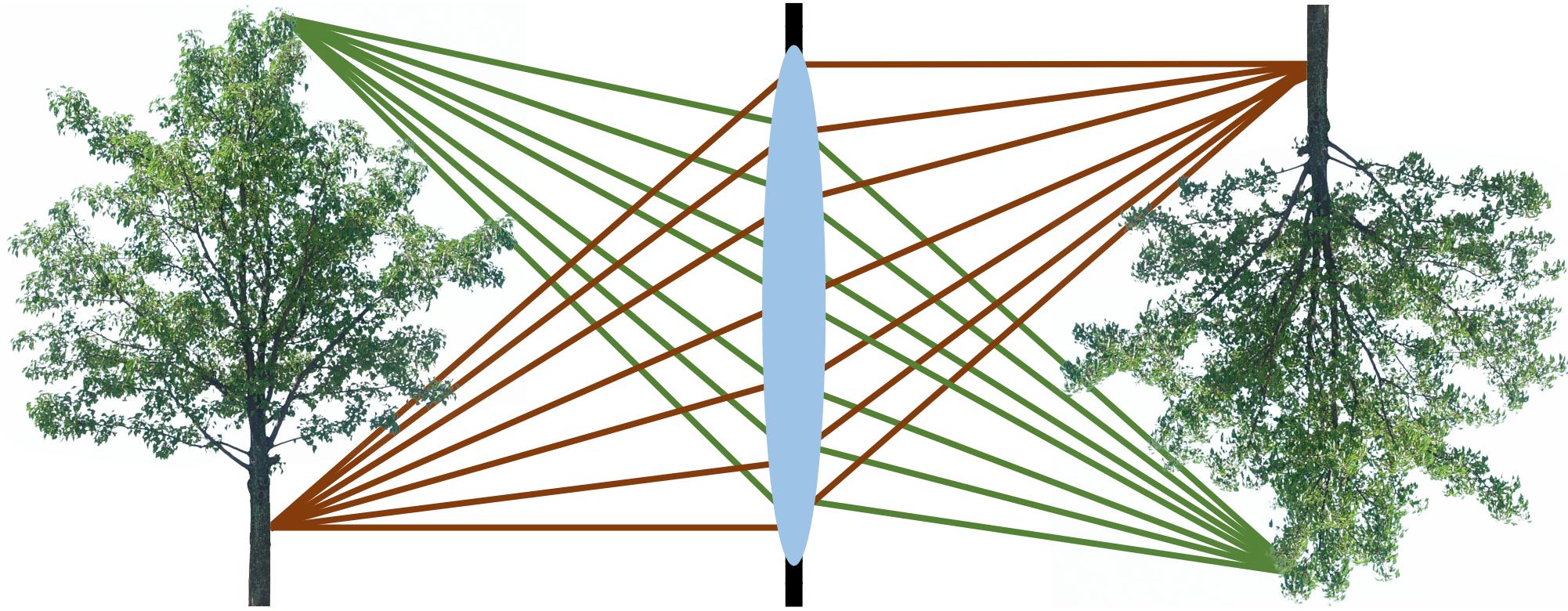


pinhole
diameter



- $2 \times$ pinhole diameter $\rightarrow 4 \times$ light
- $2 \times$ focal length $\rightarrow \frac{1}{4} \times$ light

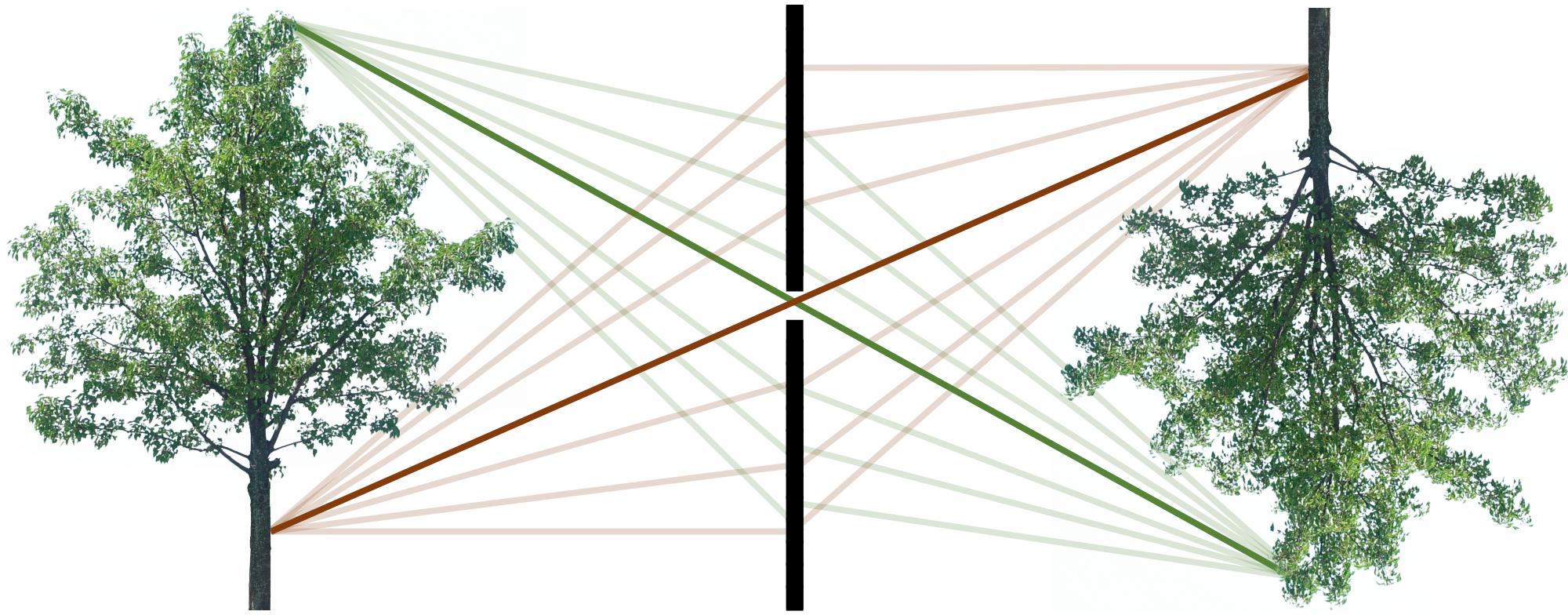
The lens camera



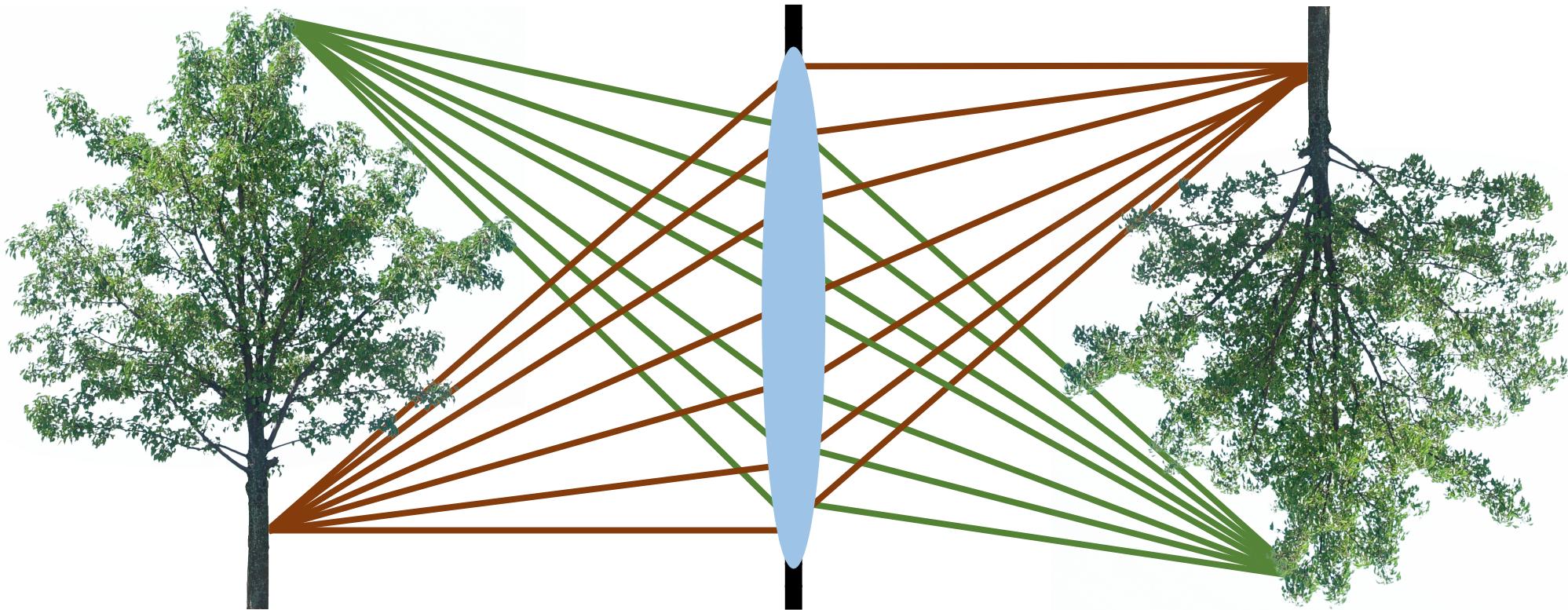
Lenses map “bundles” of rays from points on the scene to the sensor.

How does this mapping work exactly?

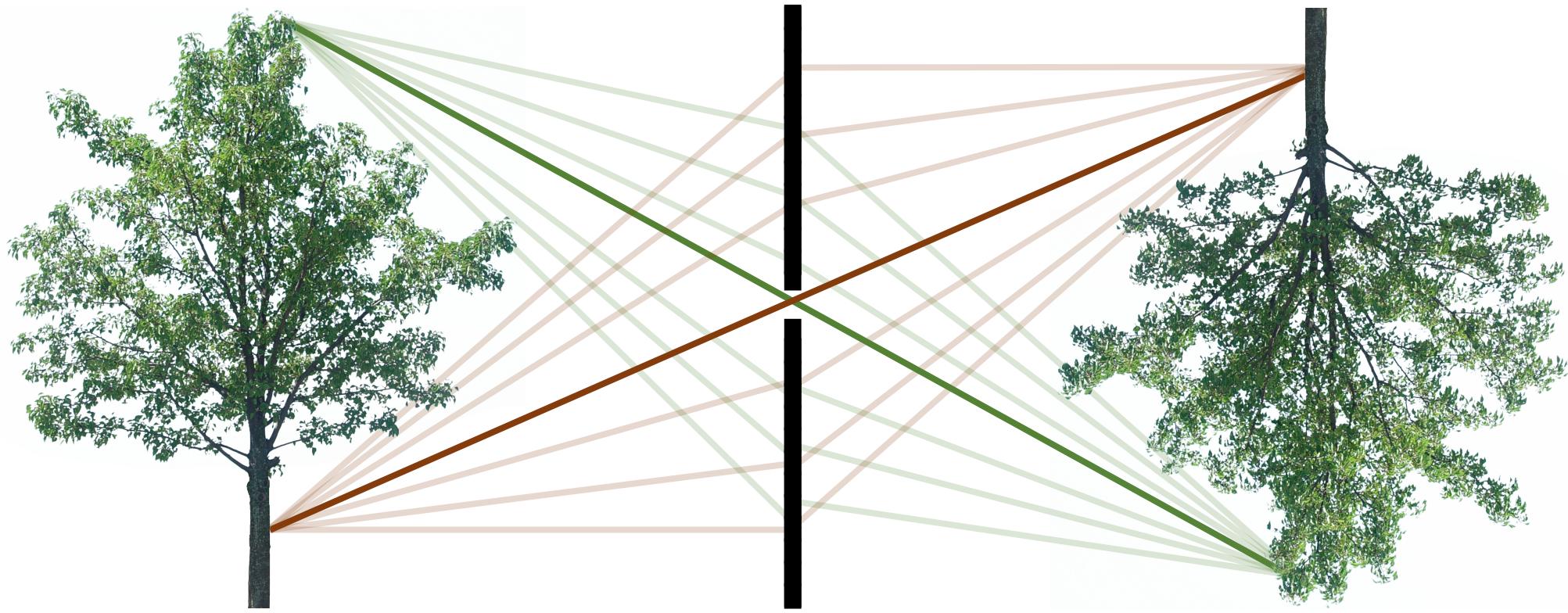
The pinhole camera



The lens camera

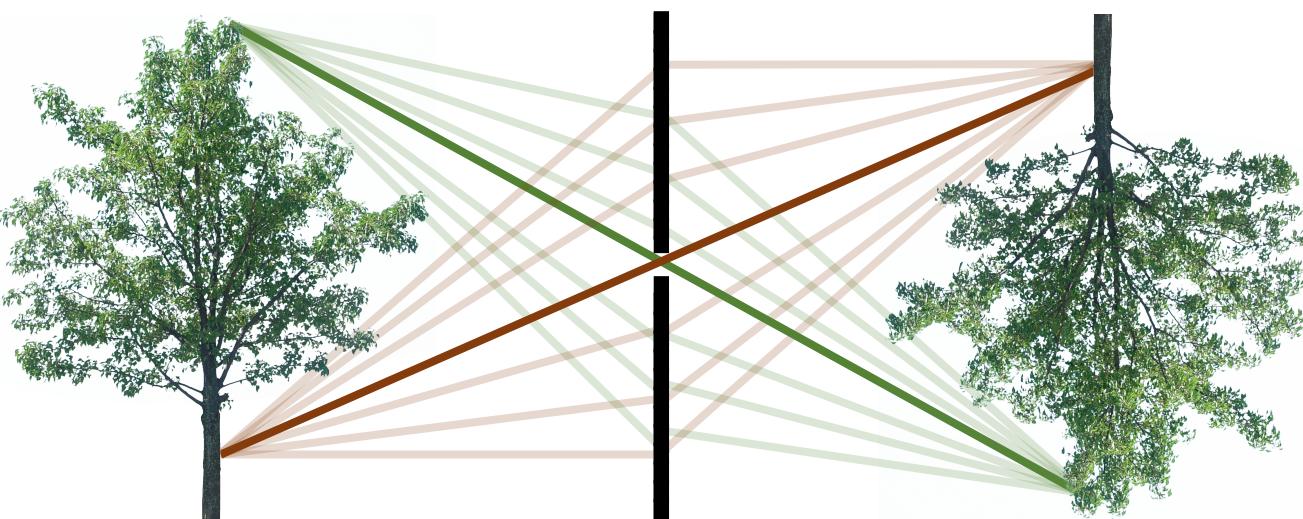
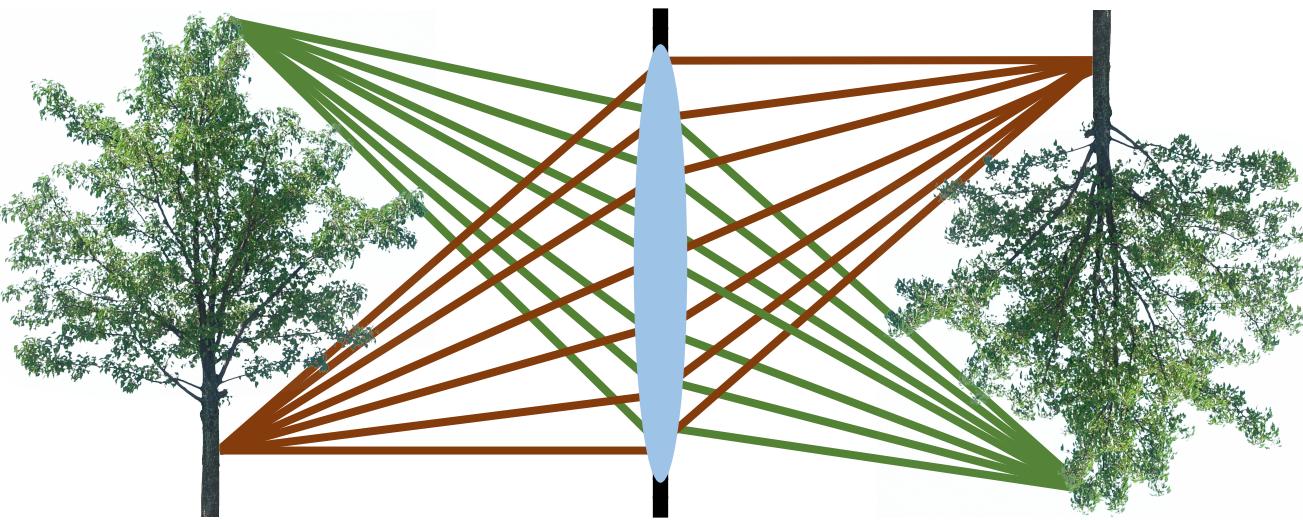


The pinhole camera



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.

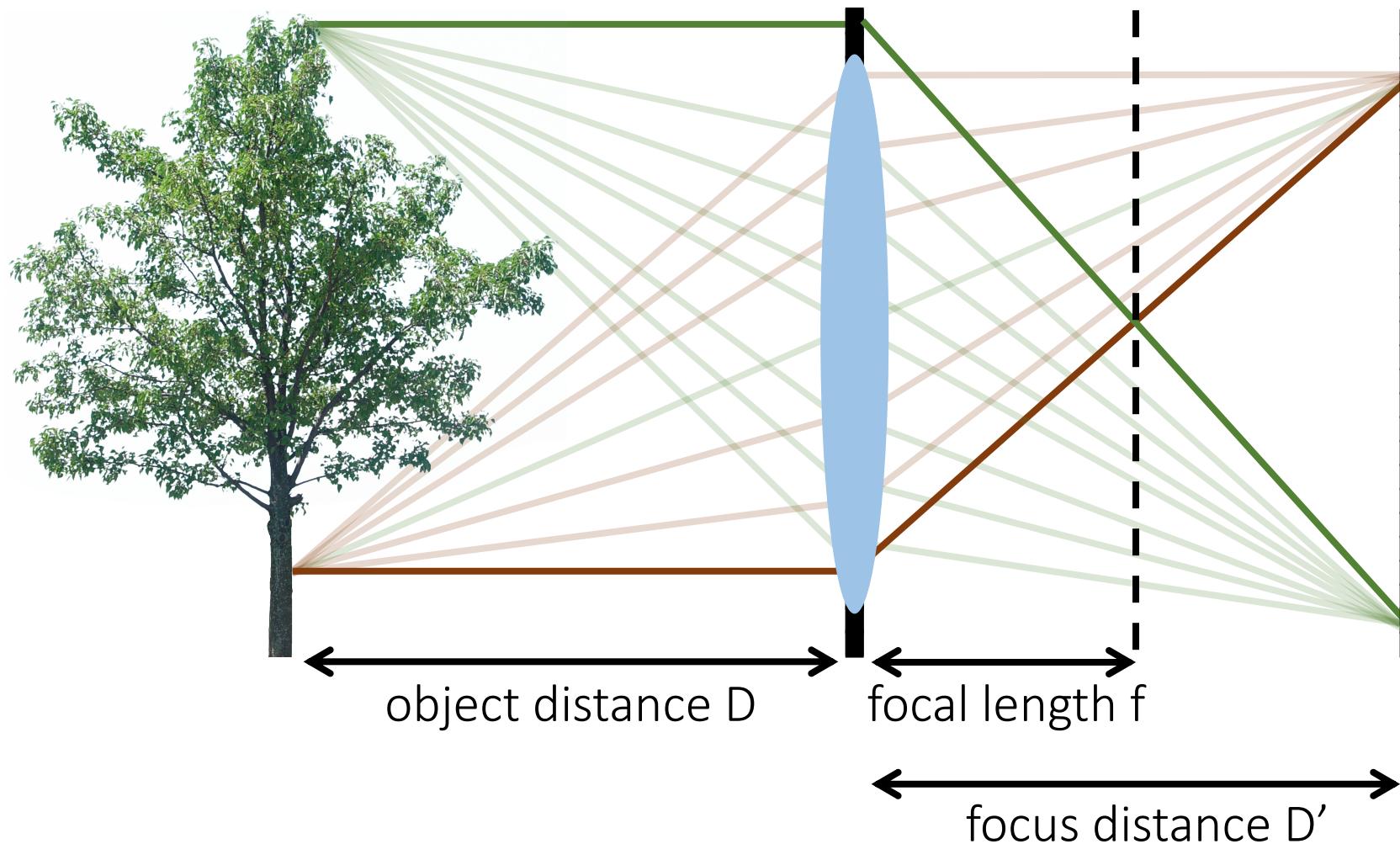
Important difference: focal length

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

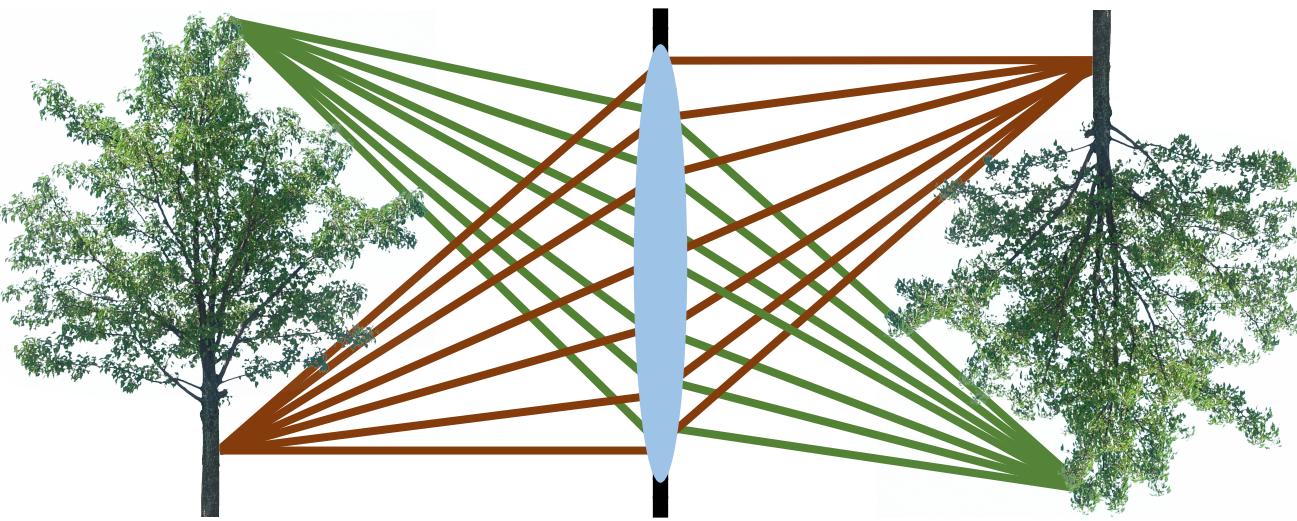


Important difference: focal length

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

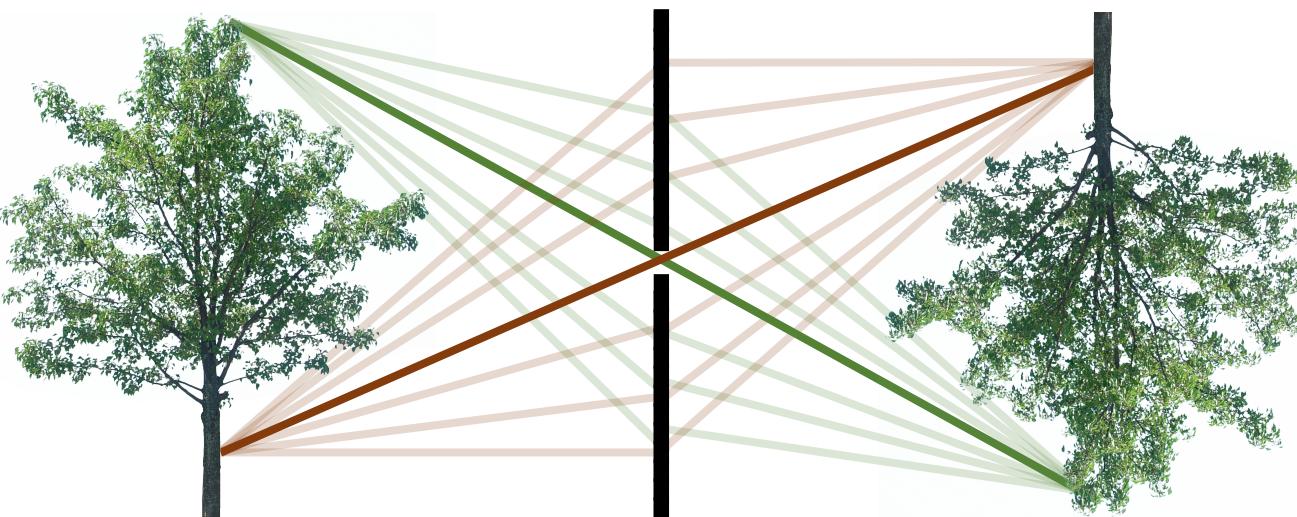


Describing both lens and pinhole cameras



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.



Remember: *focal length f* refers to different things for lens and pinhole cameras.

- In this lecture, we use it to refer to the aperture-sensor distance, as in the pinhole camera case.

Accidental pinholes





What does this image say about the world outside?



Accidental pinhole camera



Antonio Torralba, William T. Freeman
Computer Science and Artificial Intelligence Laboratory (CSAIL)
MIT
torralba@mit.edu, billf@mit.edu

Accidental pinhole camera

projected pattern on the wall



window is an
aperture

window with smaller gap



view outside window



Pinhole cameras

What are we imaging here?



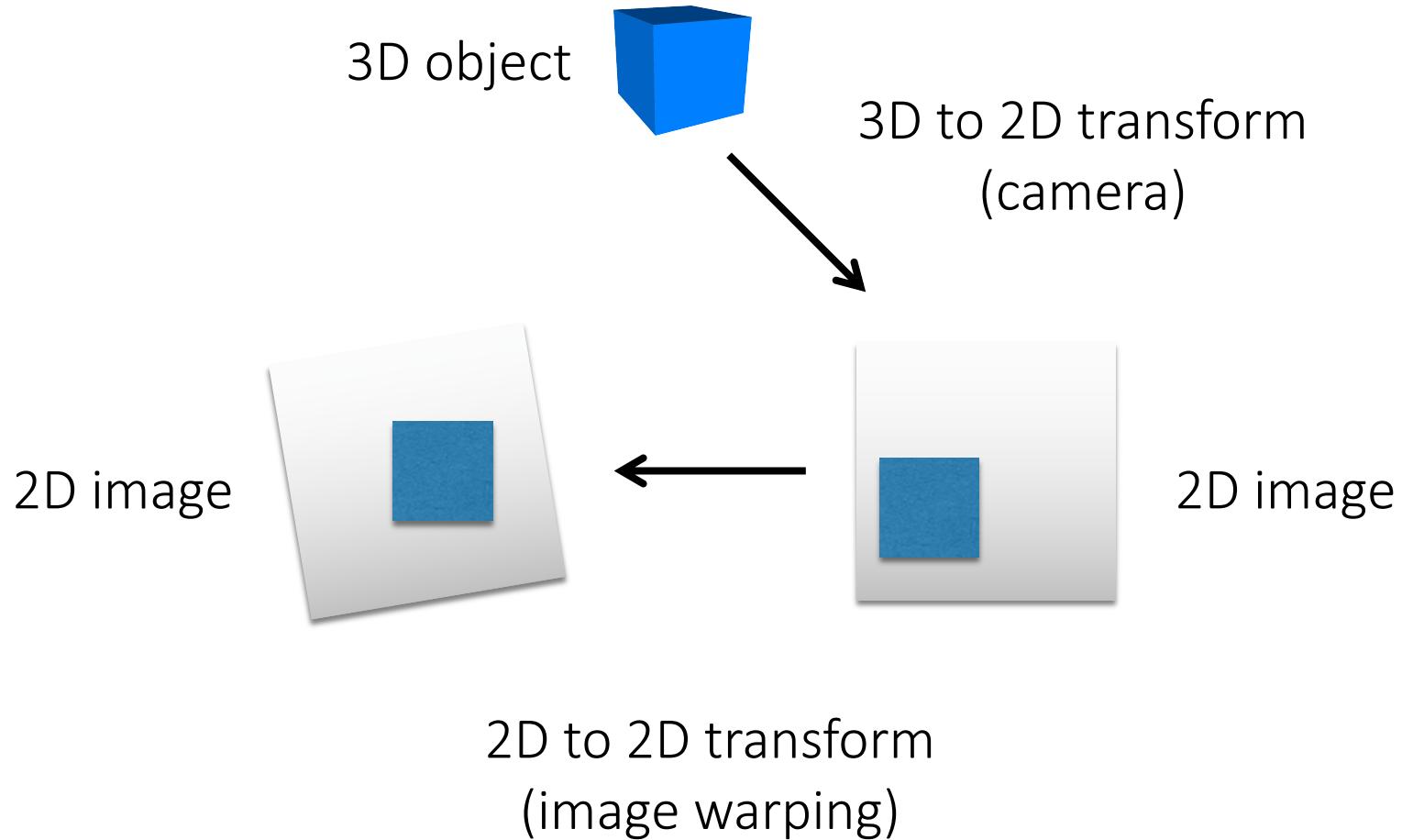
Camera matrix

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

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

homogeneous coordinates



2D image
point

camera matrix 3D world point

What are the dimensions of each variable?

The camera as a coordinate transformation

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

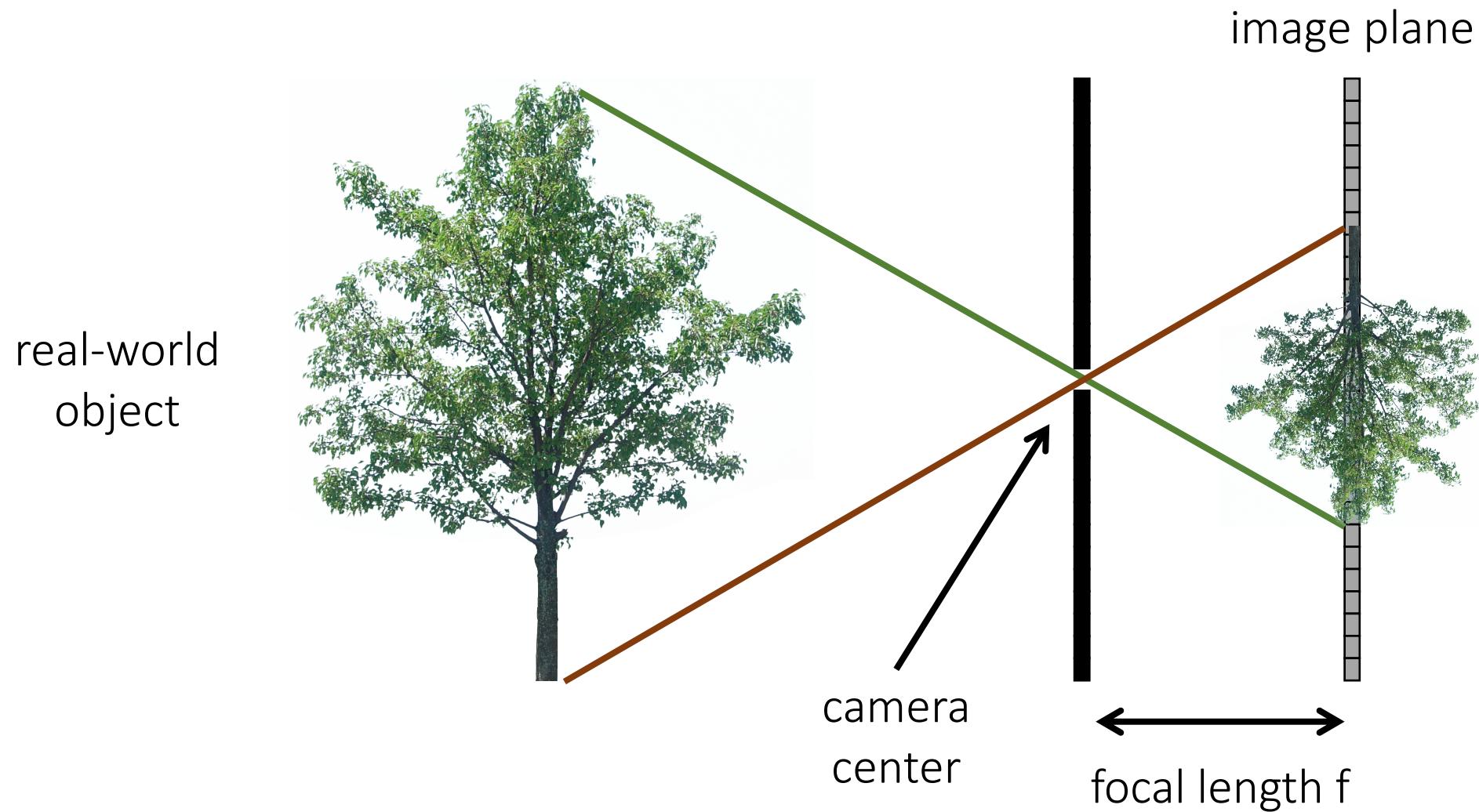
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous
image coordinates
 3×1

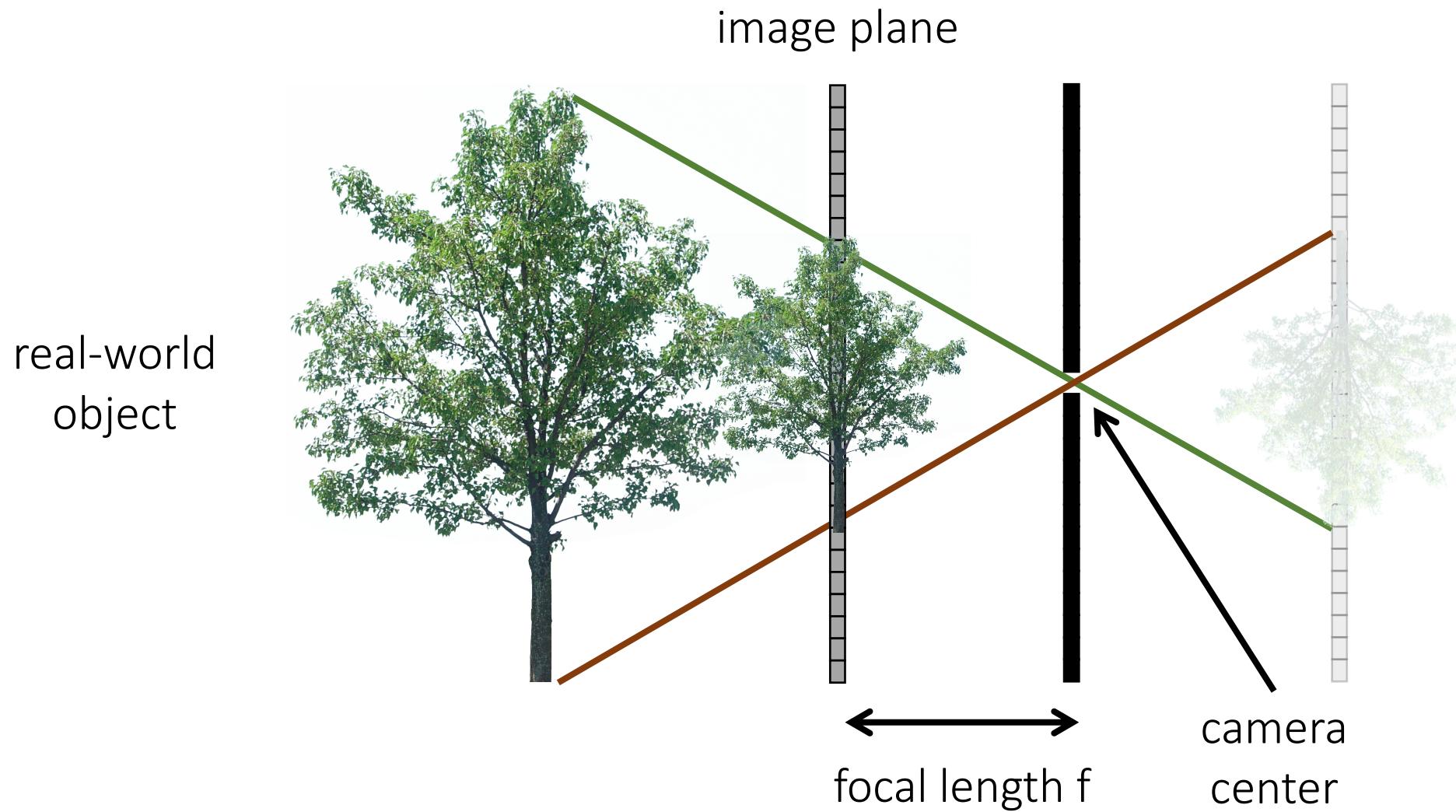
camera
matrix
 3×4

homogeneous
world coordinates
 4×1

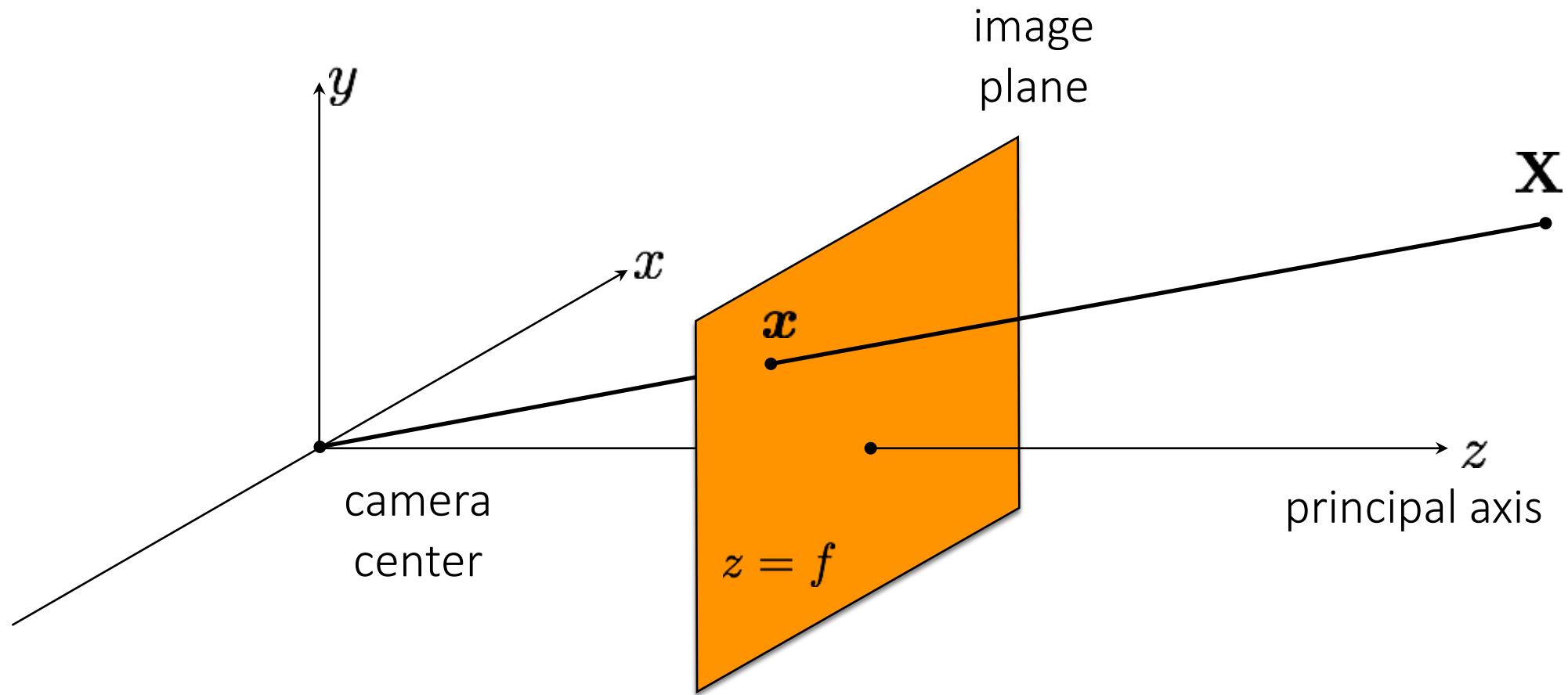
The pinhole camera



The (rearranged) pinhole camera

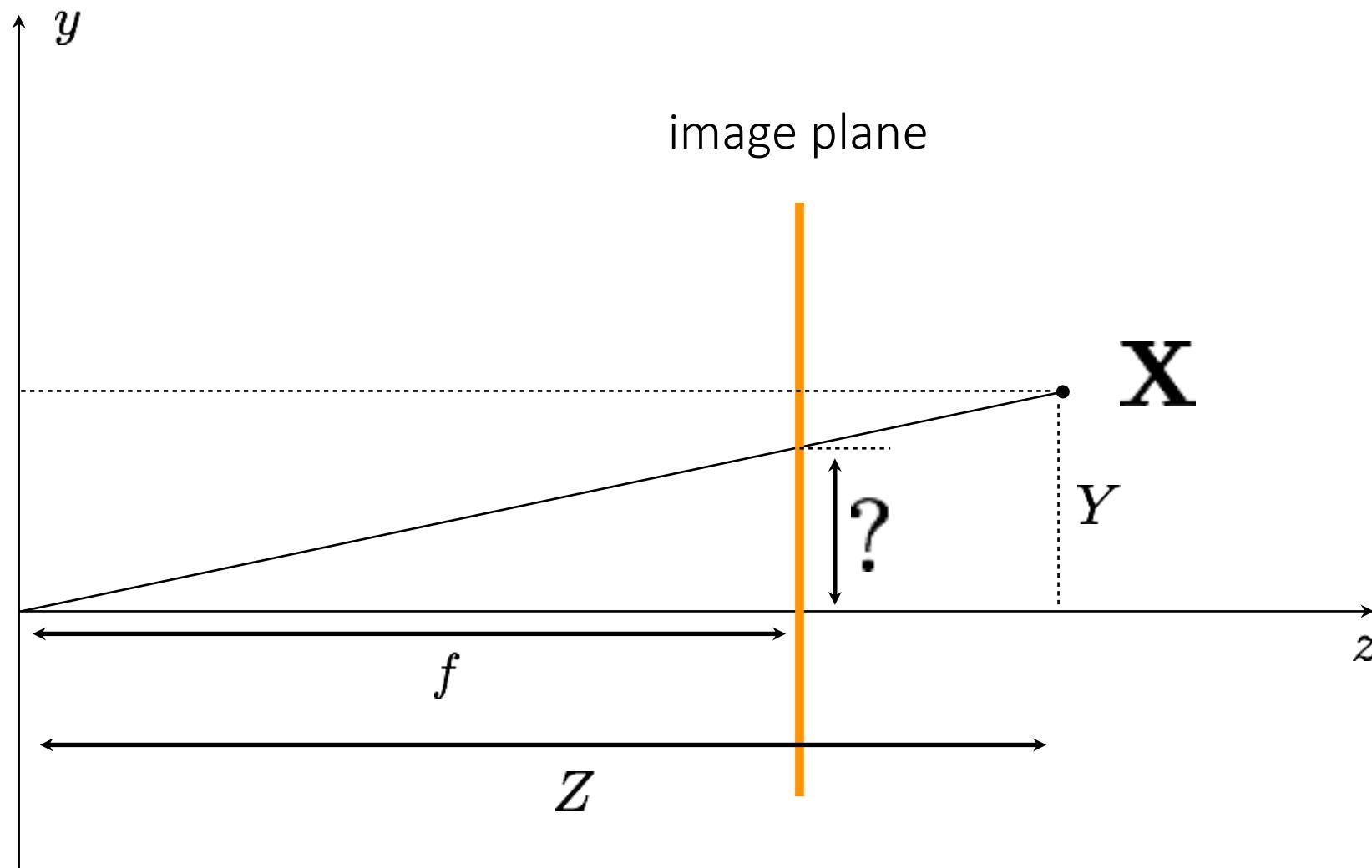


The (rearranged) pinhole camera



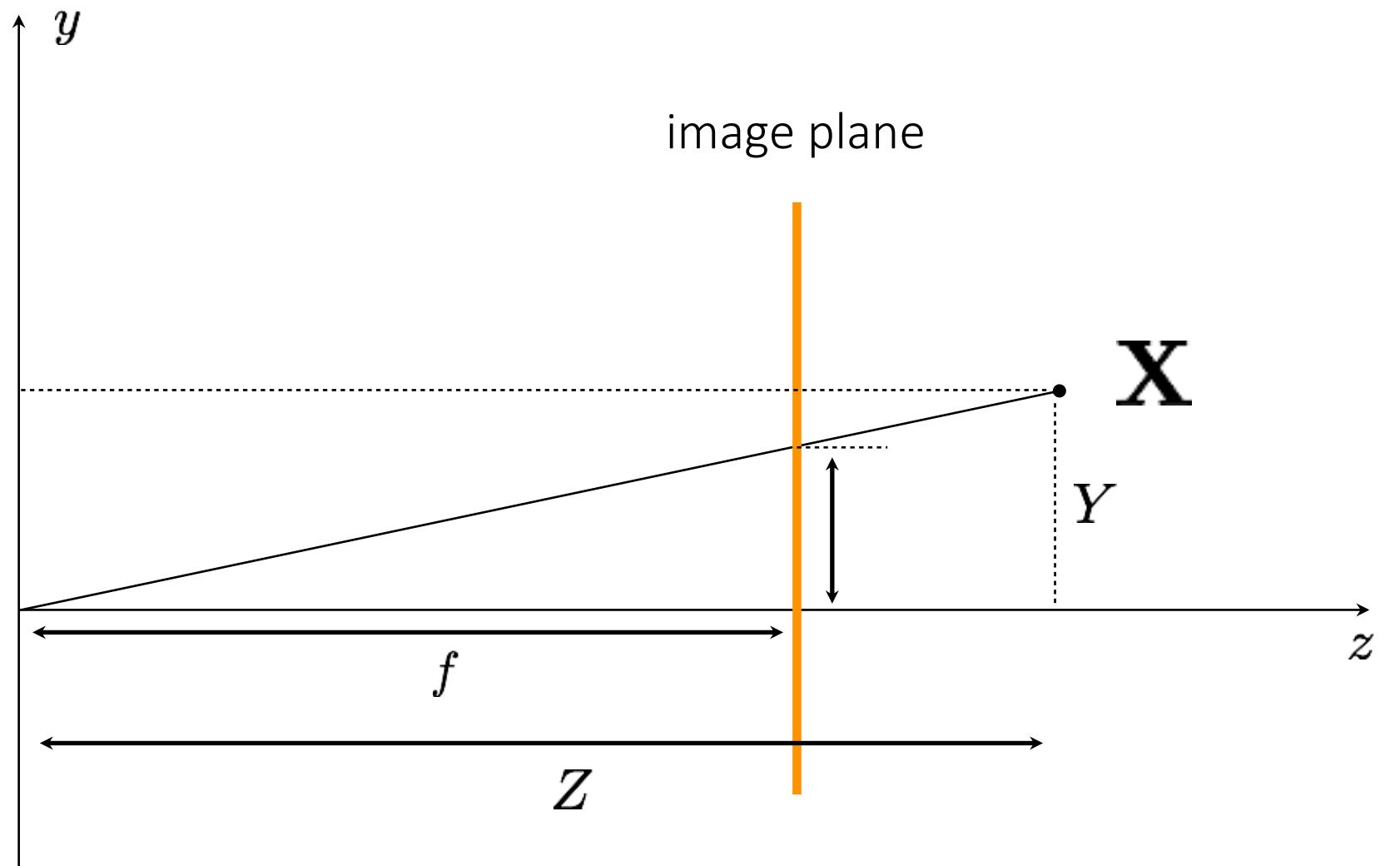
What is the equation for image coordinate \mathbf{x} in terms of \mathbf{X} ?

The 2D view of the (rearranged) pinhole camera



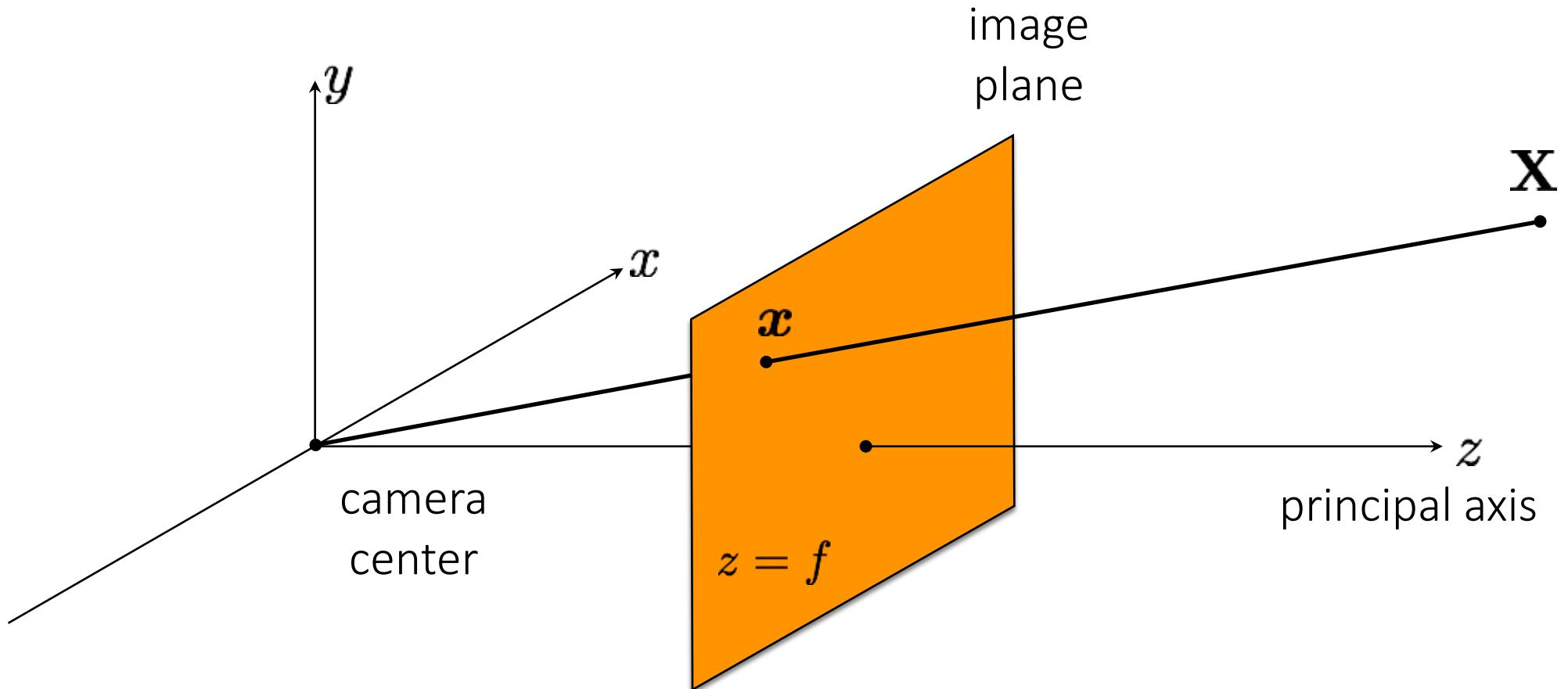
What is the equation for image coordinate x in terms of X ?

The 2D view of the (rearranged) pinhole camera



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

The (rearranged) pinhole camera



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

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

The pinhole camera matrix

Relationship from similar triangles:

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

General camera model:

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

What does the pinhole camera projection look like?

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

The pinhole camera matrix

Relationship from similar triangles:

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

General camera model:

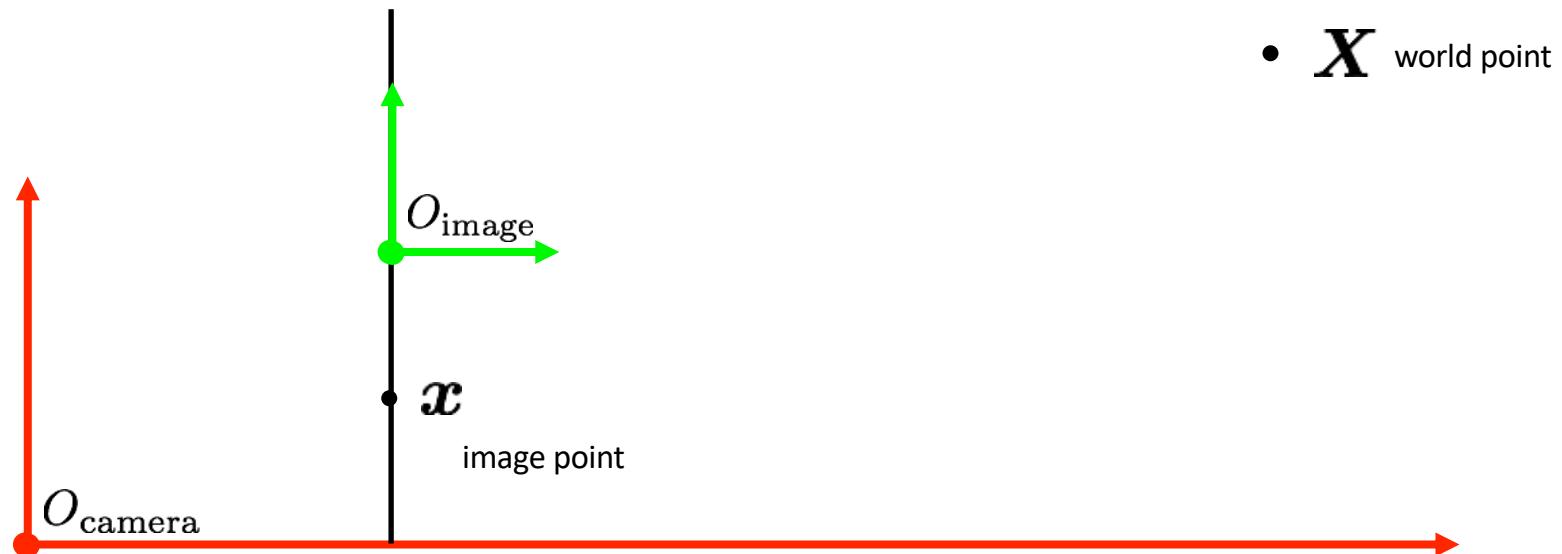
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

What does the pinhole camera projection look like?

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

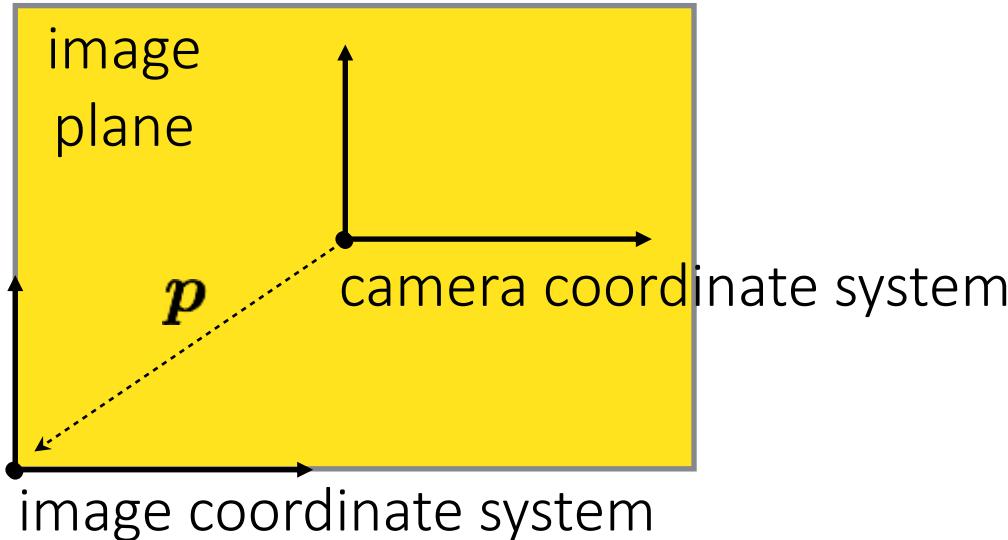
Generalizing the camera matrix

In general, the camera and image have *different* coordinate systems.



Generalizing the camera matrix

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

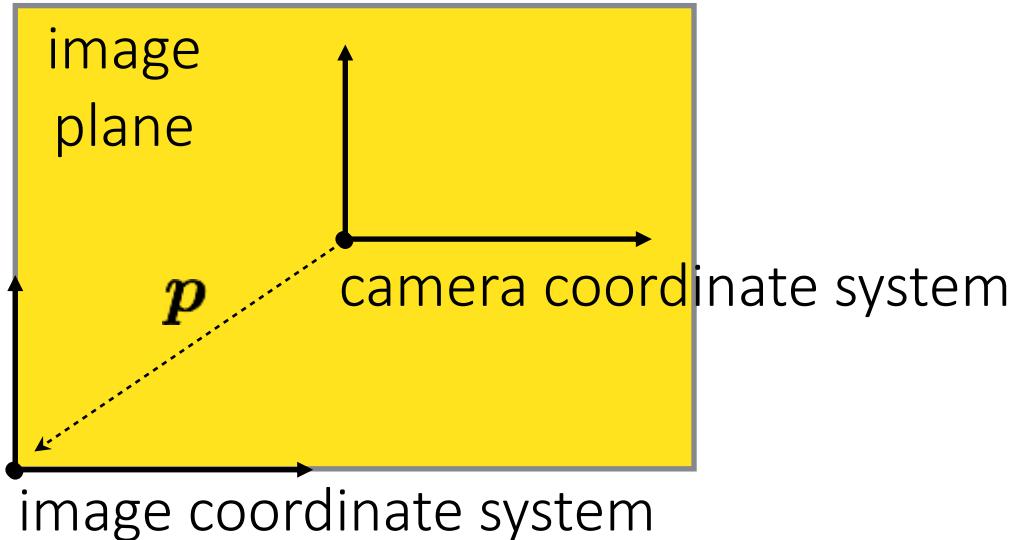


How does the camera matrix change?

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

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

Camera matrix decomposition

We can decompose the camera matrix like this:

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

What does each part of the matrix represent?

Camera matrix decomposition

We can decompose the camera matrix like this:

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



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

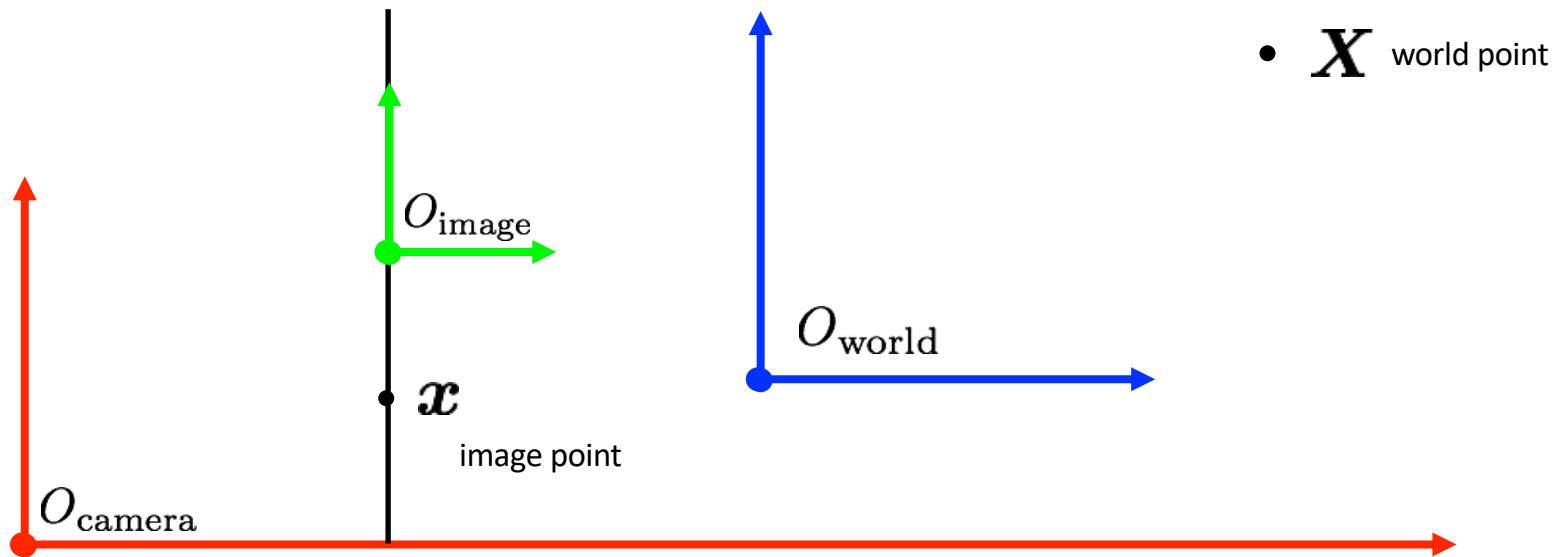
(homogeneous) projection from 3D
to 2D, assuming image plane at z = 1
and shared camera/image origin

Also written as: $\mathbf{P} = \mathbf{K}[\mathbf{I}|0]$

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

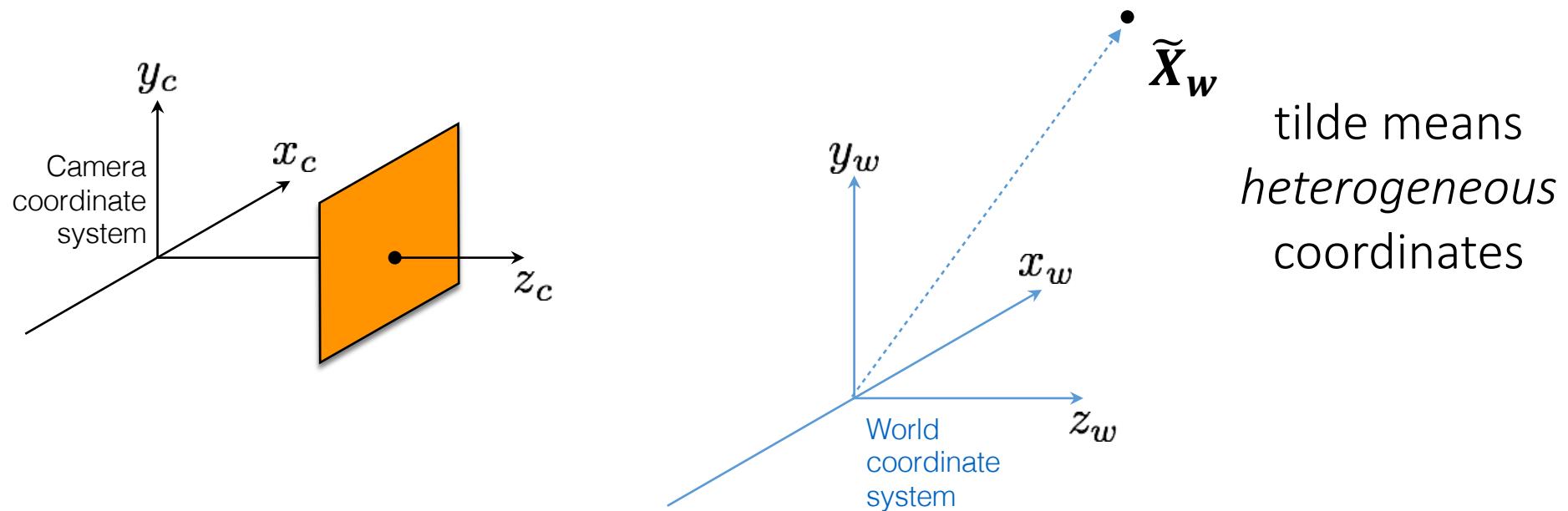
Generalizing the camera matrix

In general, there are *three*, generally different, coordinate systems.

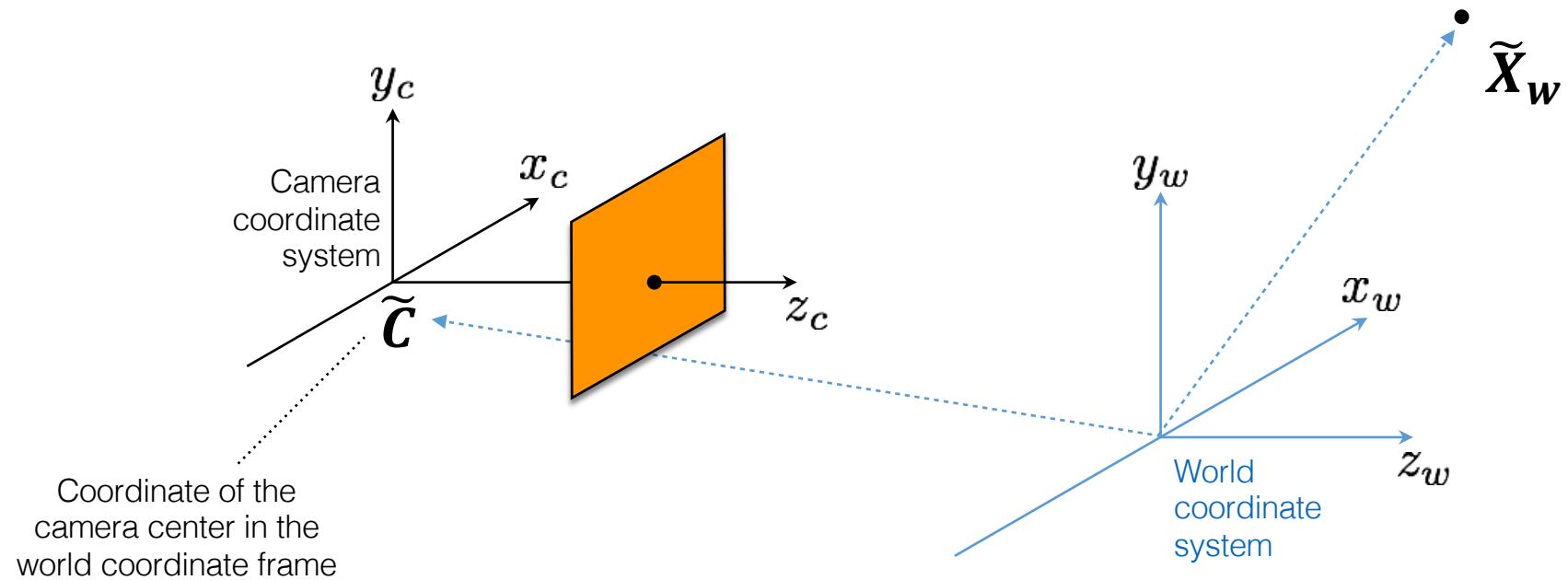


We need to know the transformations between them.

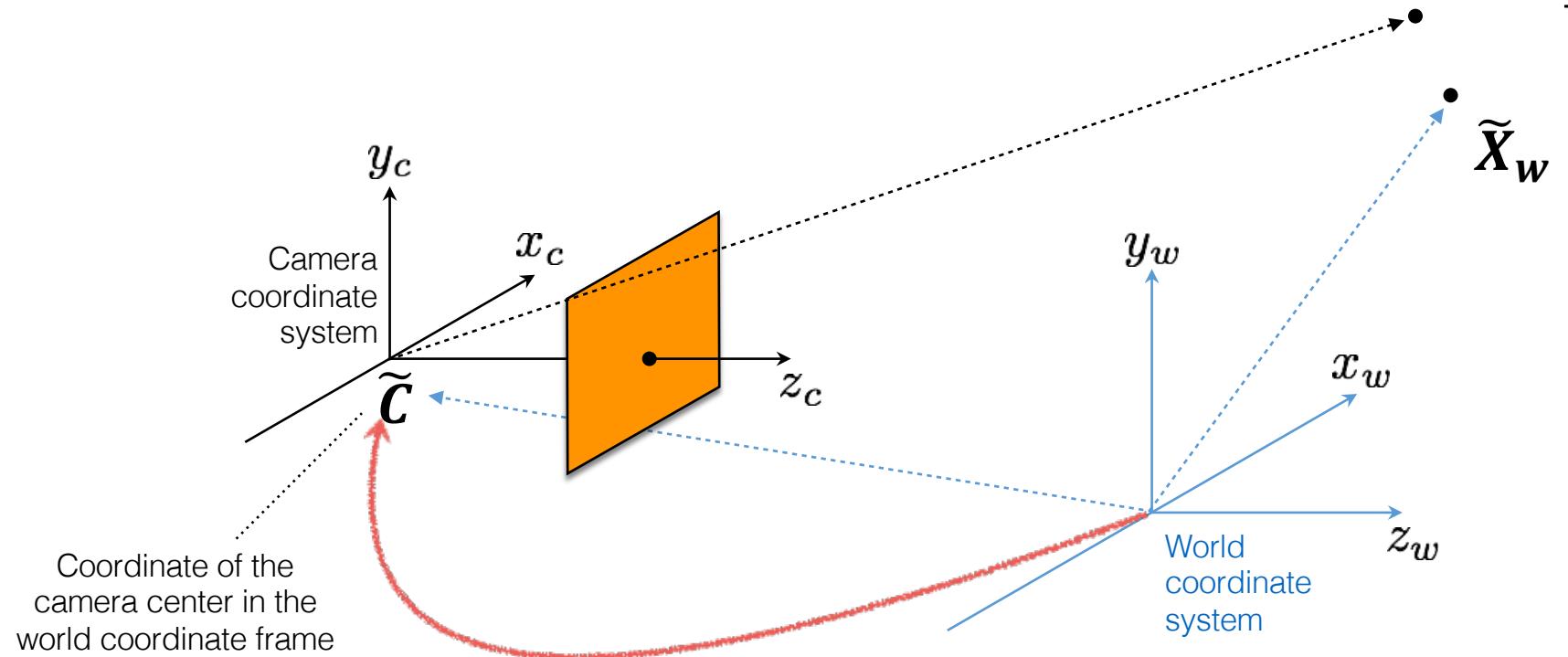
World-to-camera coordinate system transformation



World-to-camera coordinate system transformation



World-to-camera coordinate system transformation

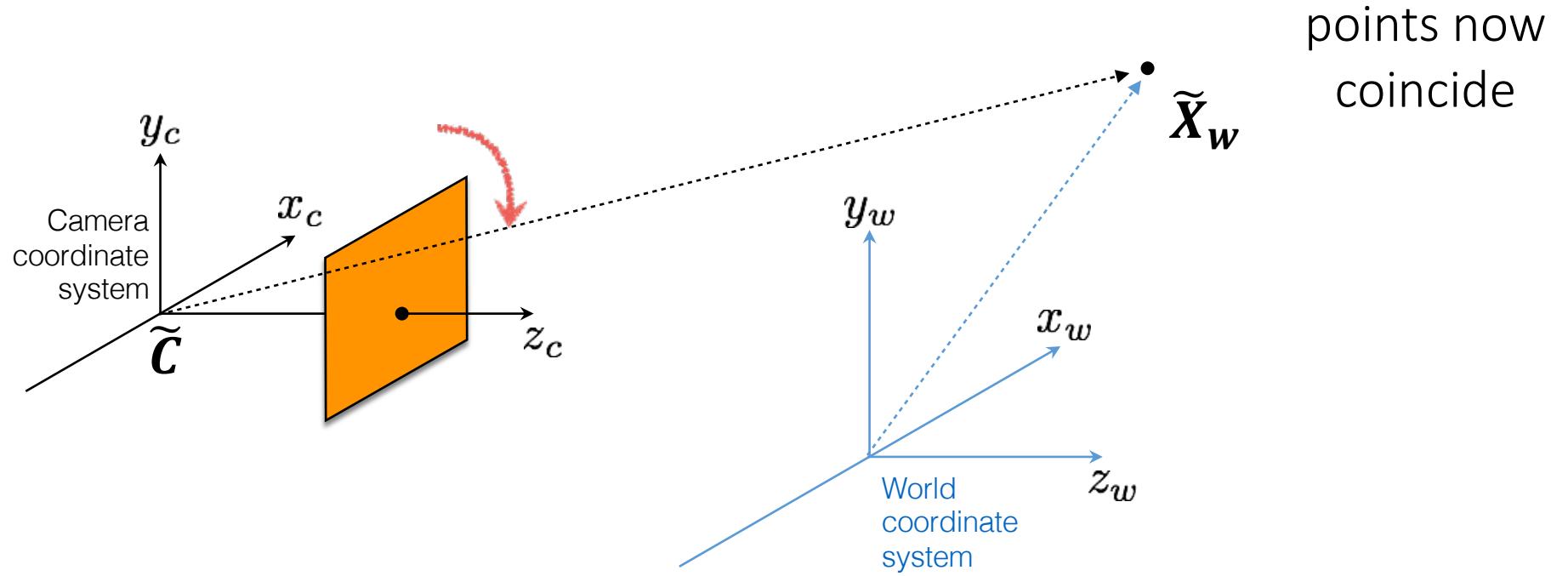


Why aren't
the points
aligned?

$$(\tilde{X}_w - \tilde{C})$$

translate

World-to-camera coordinate system transformation



$$R \cdot (\tilde{X}_w - \tilde{C})$$

rotate translate

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

$$\tilde{\mathbf{X}}_c = \mathbf{R} \cdot (\tilde{\mathbf{X}}_w - \tilde{\mathbf{C}})$$

How do we write this transformation in homogeneous coordinates?

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

$$\tilde{\mathbf{X}}_c = \mathbf{R} \cdot (\tilde{\mathbf{X}}_w - \tilde{\mathbf{C}})$$

In homogeneous coordinates, we have:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{C} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{or} \quad \mathbf{x}_c = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{x}_w$$

Incorporating the transform in the camera matrix

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

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

We also just derived:

$$\mathbf{X}_c = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}_w$$

Putting it all together

We can write everything into a single projection:

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

The camera matrix now looks like:

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

intrinsic parameters (3×3):
correspond to camera internals
(sensor not at $f = 1$ and origin shift)

extrinsic parameters (3×4):
correspond to camera externals
(world-to-image transformation)

General pinhole camera matrix

We can decompose the camera matrix like this:

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

(translate first then rotate)

Another way to write the mapping:

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

where $\mathbf{t} = -\mathbf{R}\mathbf{C}$

(rotate first then translate)

General pinhole camera matrix

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

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

intrinsic extrinsic
parameters parameters

$$\mathbf{R} = \left[\begin{array}{ccc} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{array} \right] \quad \mathbf{t} = \left[\begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array} \right]$$

3D rotation 3D translation

Recap

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

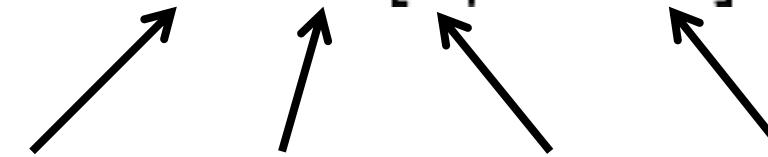
$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}] - \mathbf{C}$$

The diagram consists of a mathematical equation $\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}] - \mathbf{C}$ centered on the page. Below the equation, there are four question marks: one under \mathbf{K} , one under \mathbf{R} , one under $[\mathbf{I}]$, and one under \mathbf{C} . Four black arrows originate from these question marks and point diagonally upwards towards their respective terms in the equation above.

Recap

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

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



3x3

intrinsics

?

?

?

Recap

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

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

3x3
intrinsics 3x3
3D rotation ? ?

Recap

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

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

3x3 3x3 3x3 ?

intrinsics 3D rotation identity

Recap

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

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

3x3

3x3

3x3

3x1

intrinsics

3D rotation

identity

3D translation

Quiz

The camera matrix relates what two quantities?

Quiz

The camera matrix relates what two quantities?

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

homogeneous 3D points to 2D image points

Quiz

The camera matrix relates what two quantities?

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

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?

Quiz

The camera matrix relates what two quantities?

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

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?

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

intrinsic and extrinsic parameters

More general camera matrices

The following is the standard camera matrix we saw.

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad [\mathbf{R} \quad \vdots \quad -\mathbf{RC}]$$

More general camera matrices

CCD camera: pixels may not be square.

$$\mathbf{P} = \begin{bmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad \left[\begin{array}{c|c} \mathbf{R} & -\mathbf{RC} \end{array} \right]$$

How many degrees of freedom?

More general camera matrices

CCD camera: pixels may not be square.

$$\mathbf{P} = \begin{bmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad \left[\begin{array}{c|c} \mathbf{R} & -\mathbf{RC} \end{array} \right]$$

How many degrees of freedom?

10 DOF

More general camera matrices

Finite projective camera: sensor be skewed.

$$\mathbf{P} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{c|c} \mathbf{R} & -\mathbf{RC} \end{array} \right]$$

How many degrees of freedom?

More general camera matrices

Finite projective camera: sensor be skewed.

$$\mathbf{P} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \end{bmatrix}$$

How many degrees of freedom?

11 DOF

Perspective distortion

Finite projective camera

$$\mathbf{P} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \end{bmatrix}$$


What does this matrix look like if the camera and world have the same coordinate system?

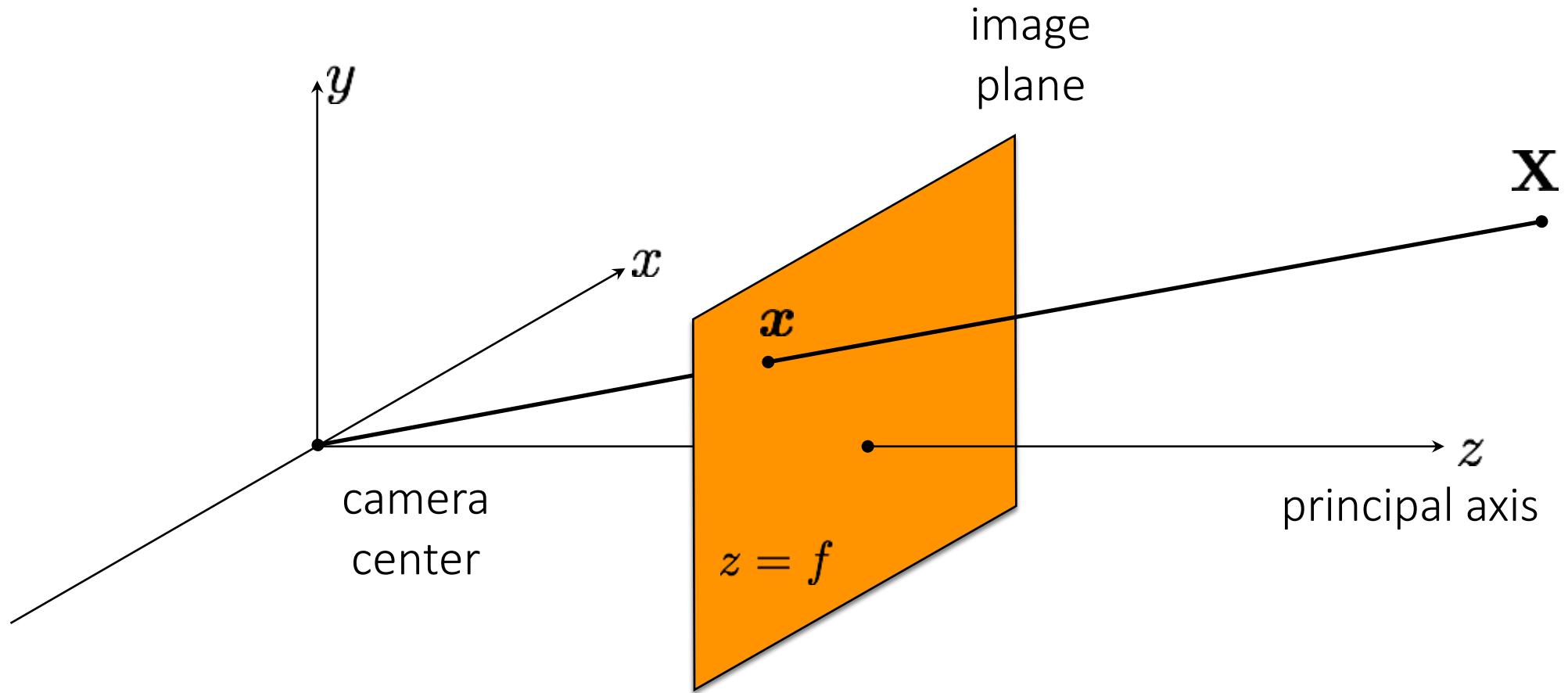
Finite projective camera

The pinhole camera and all of the more general cameras we have seen so far have “*perspective distortion*”.

$$\mathbf{P} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$


*Perspective projection from
(homogeneous) 3D to 2D coordinates*

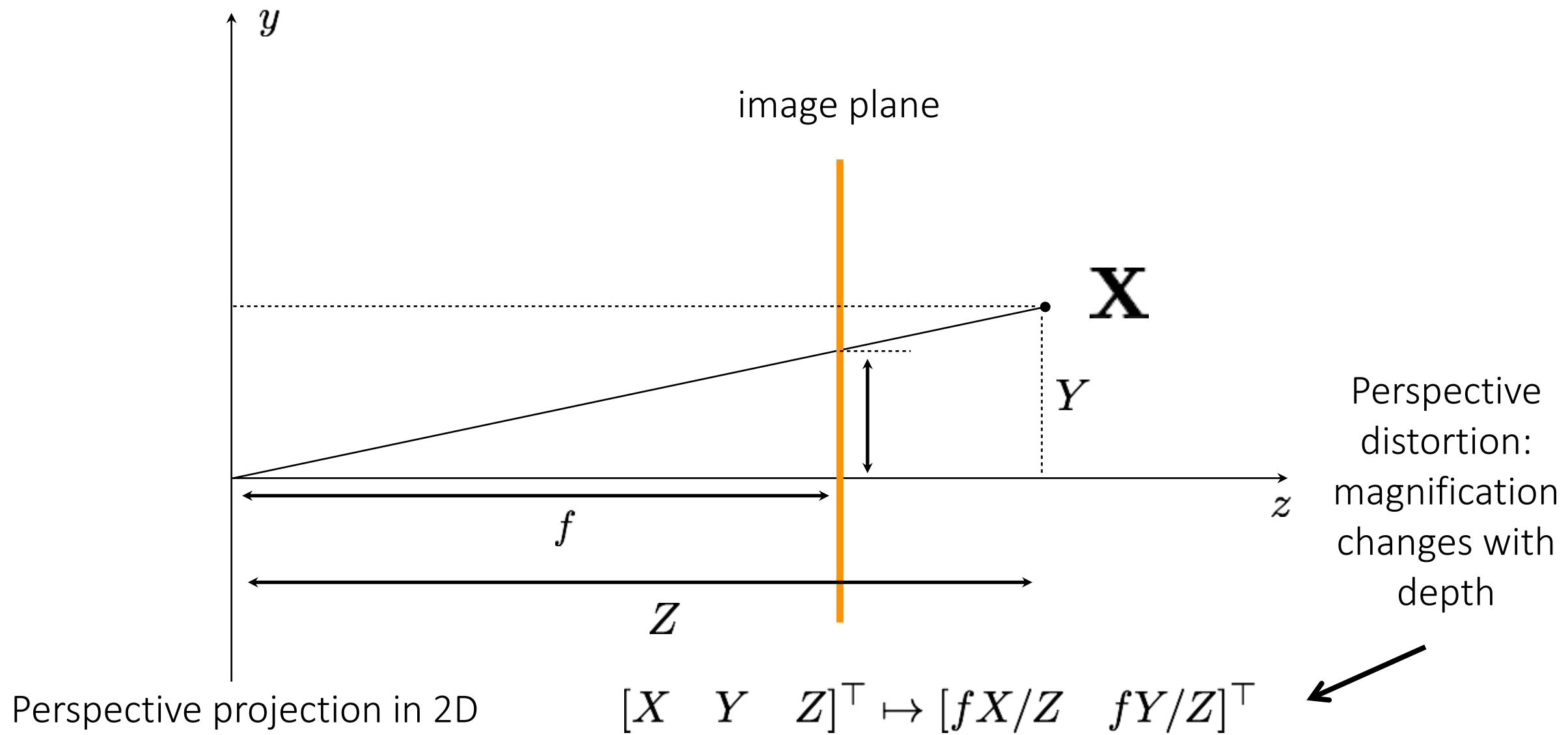
The (rearranged) pinhole camera



Perspective projection in 3D

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

The 2D view of the (rearranged) pinhole camera



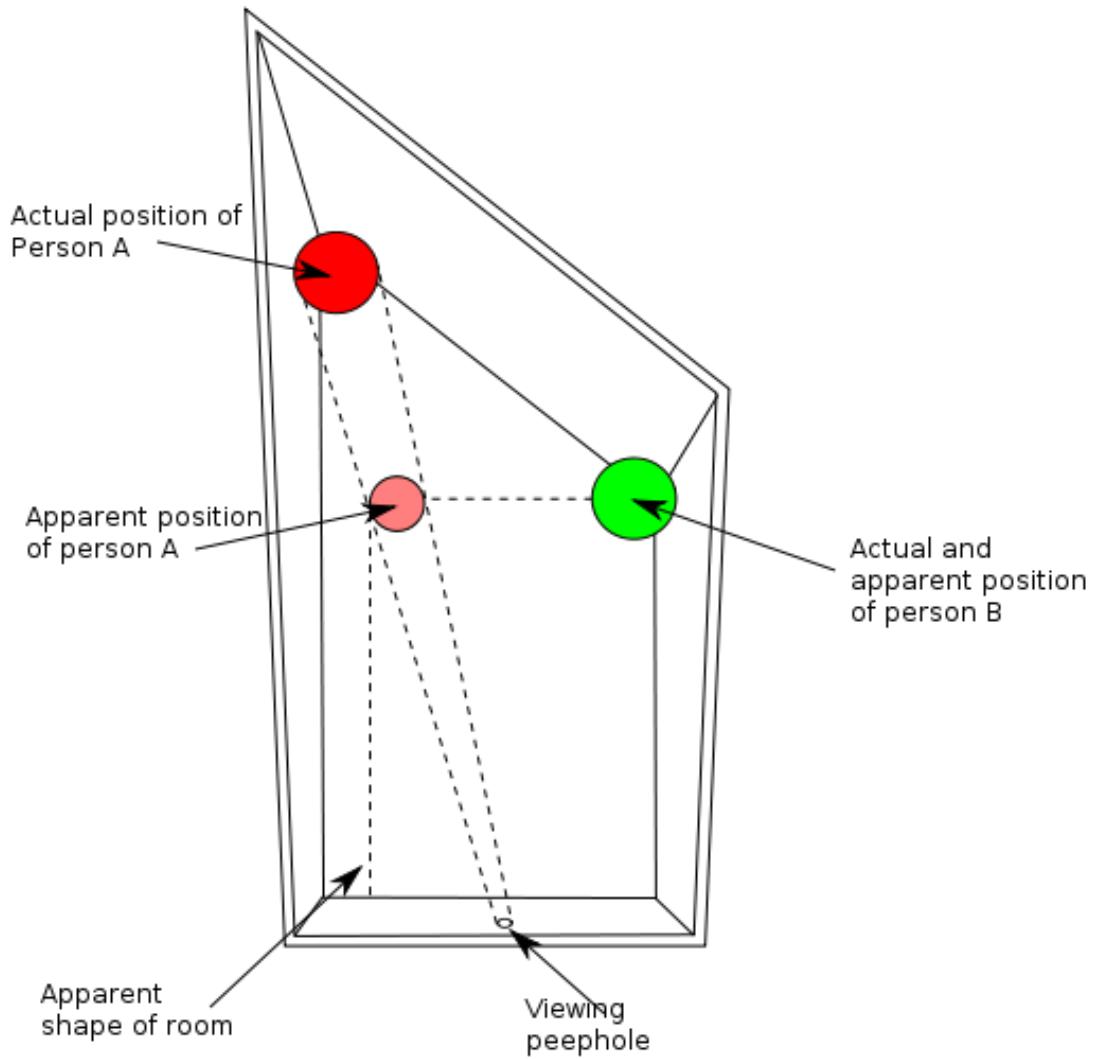
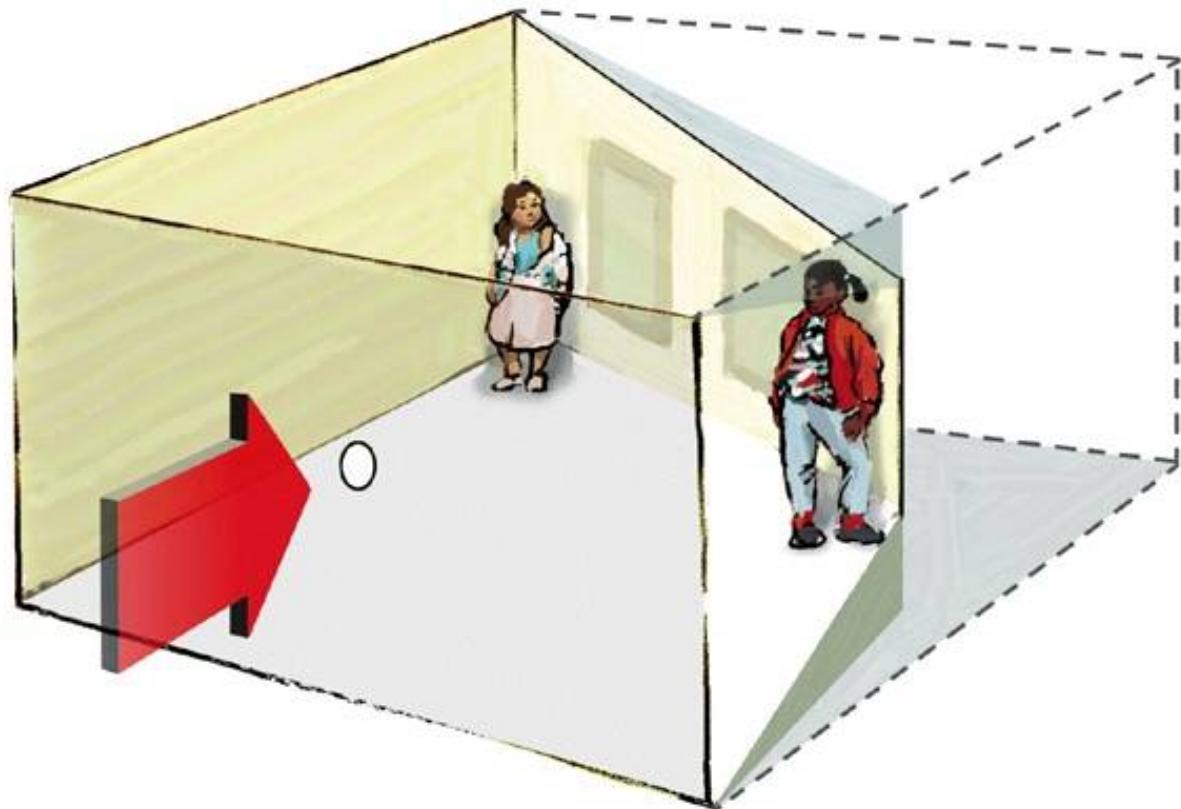
Forced perspective



The Ames room illusion



The Ames room illusion

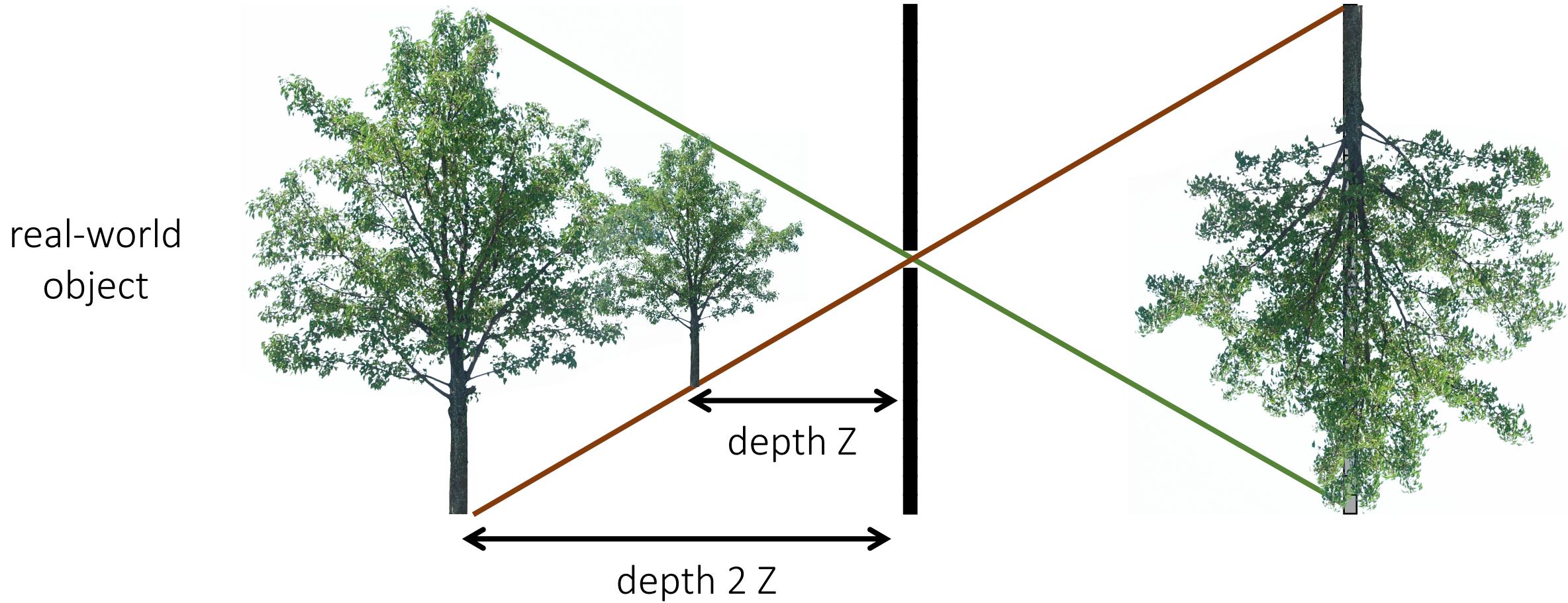


The arrow illusion

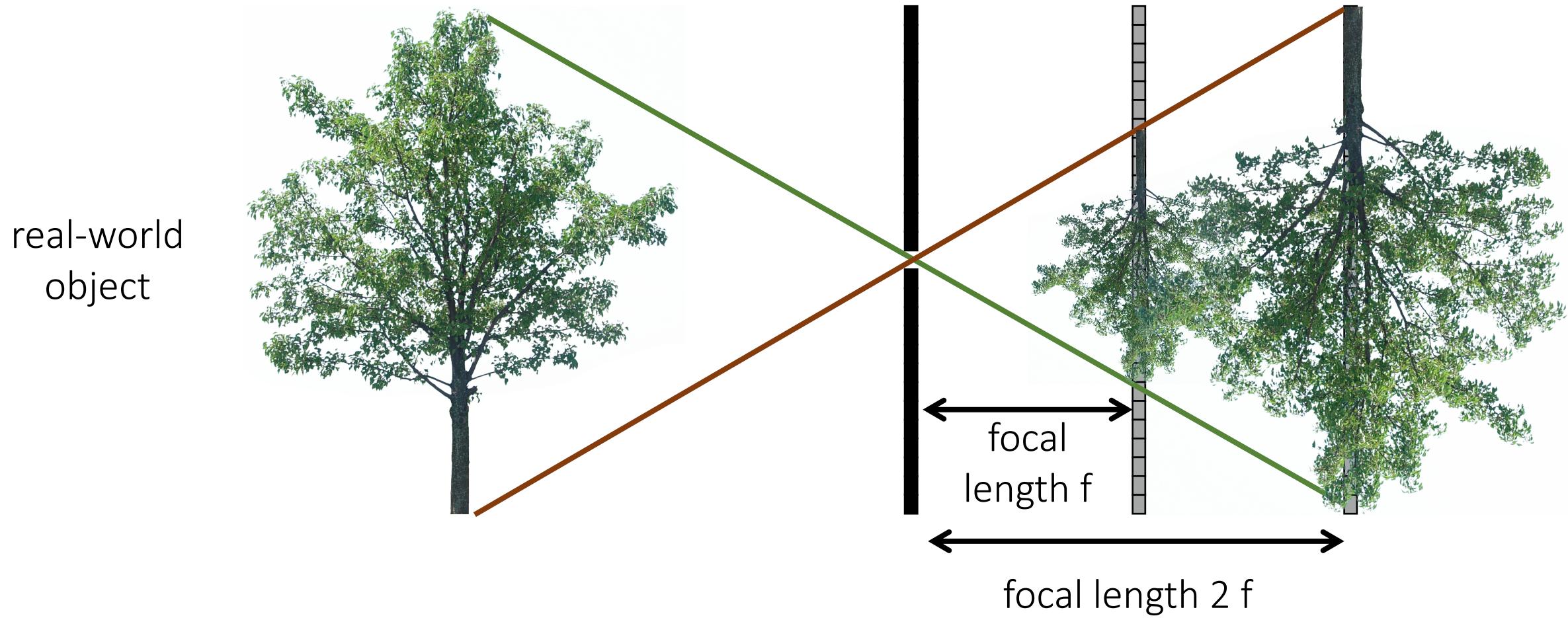


Magnification depends on depth

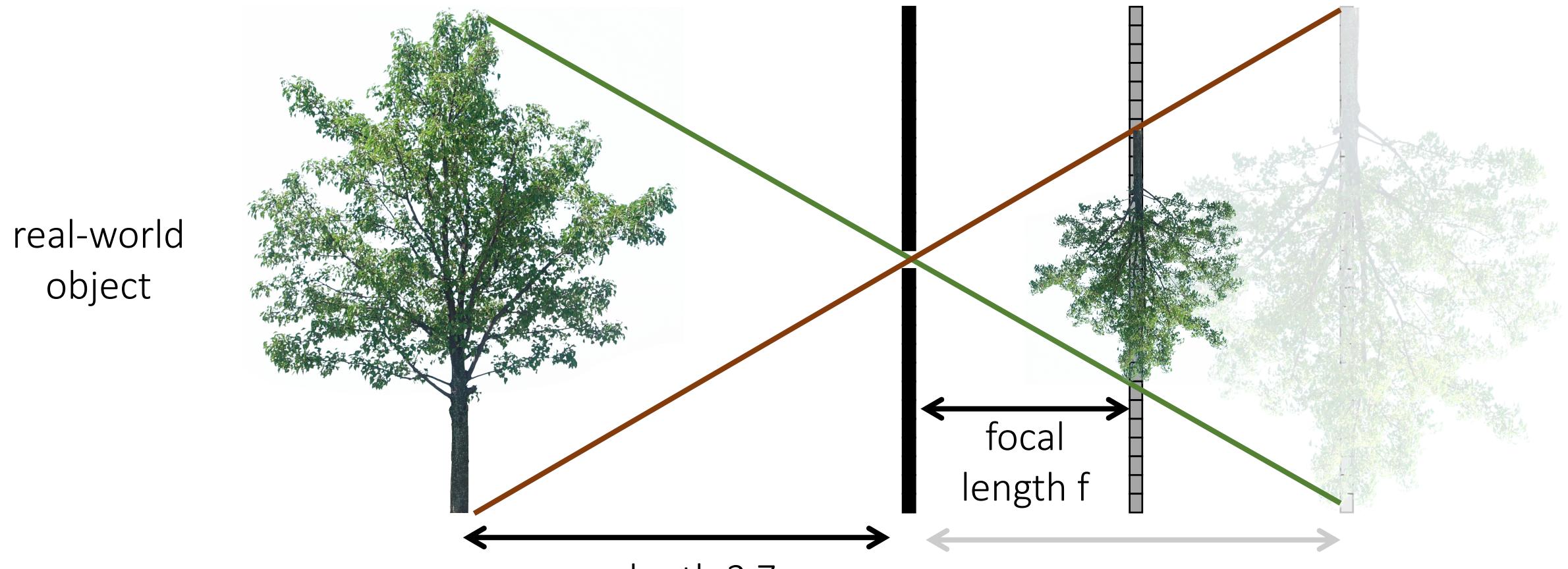
What happens as we change the focal length?



Magnification depends on focal length



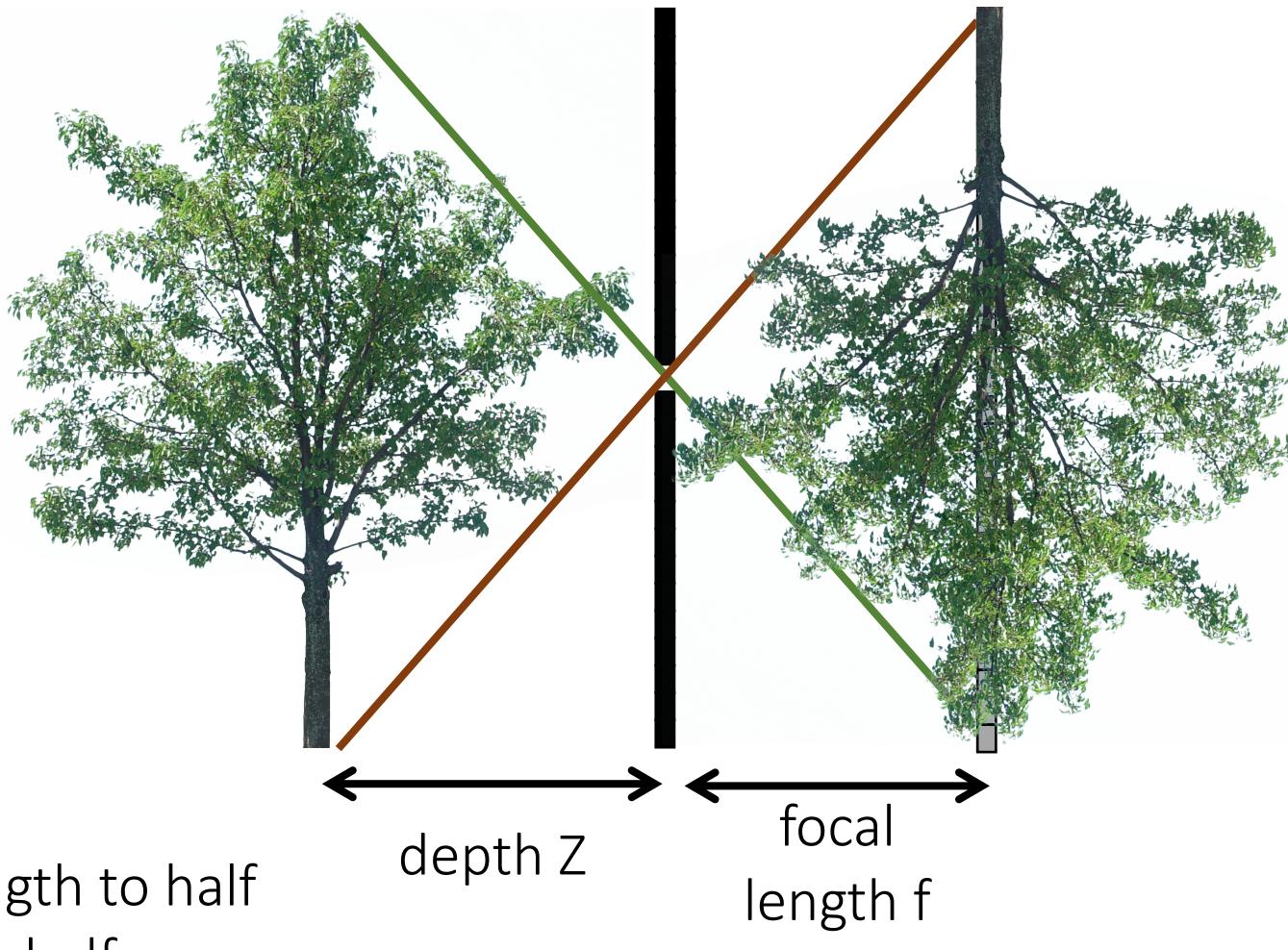
What if...



1. Set focal length to half

What if...

real-world
object



1. Set focal length to half
2. Set depth to half

Is this the same image as
the one I had at focal
length $2f$ and distance $2Z$?

Perspective distortion



long focal length



mid focal length



short focal length

Perspective distortion



Vertigo effect

Named after Alfred Hitchcock's movie

- also known as “dolly zoom”



Vertigo effect

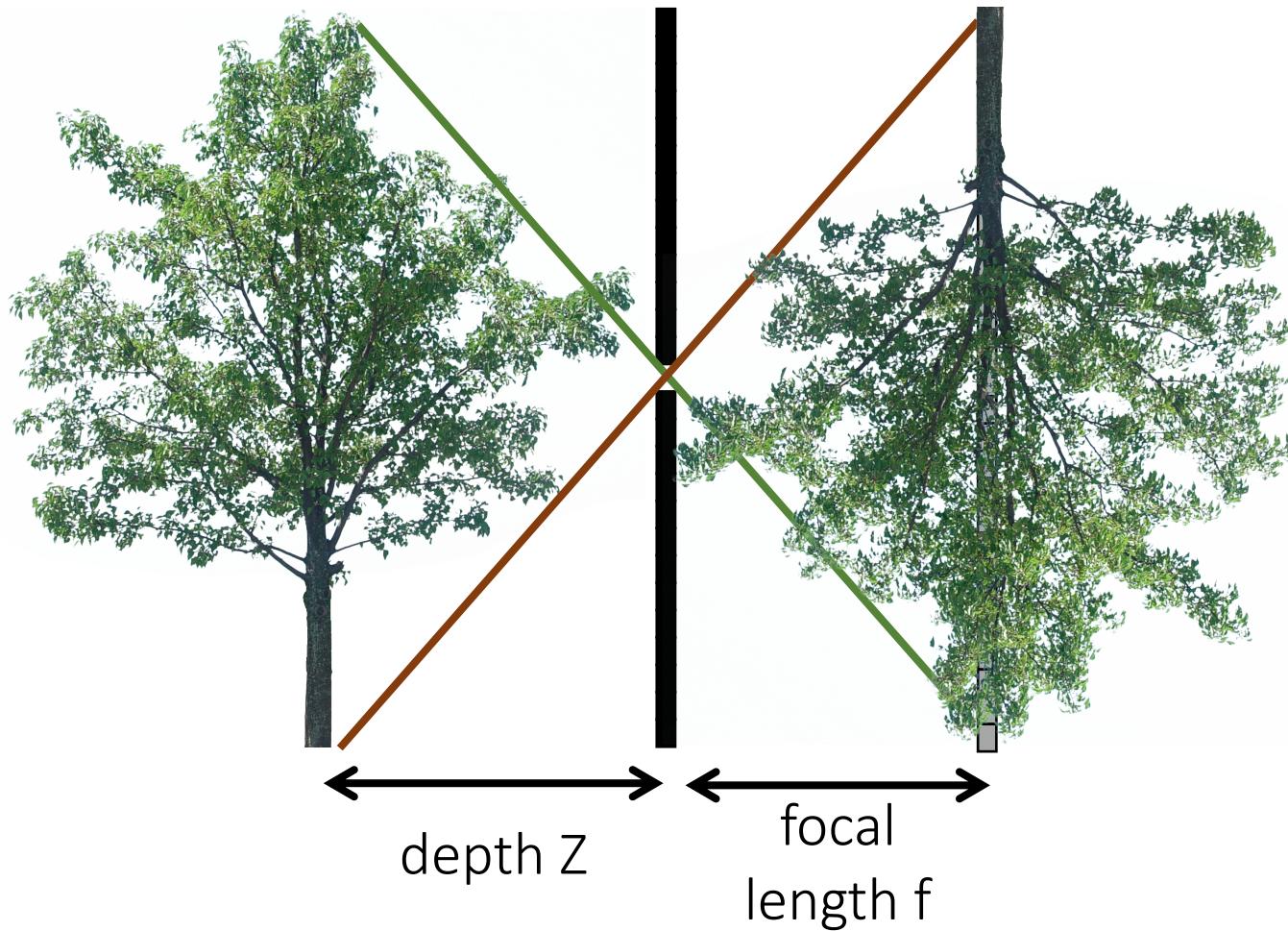


How would you
create this effect?

Other camera models

What if...

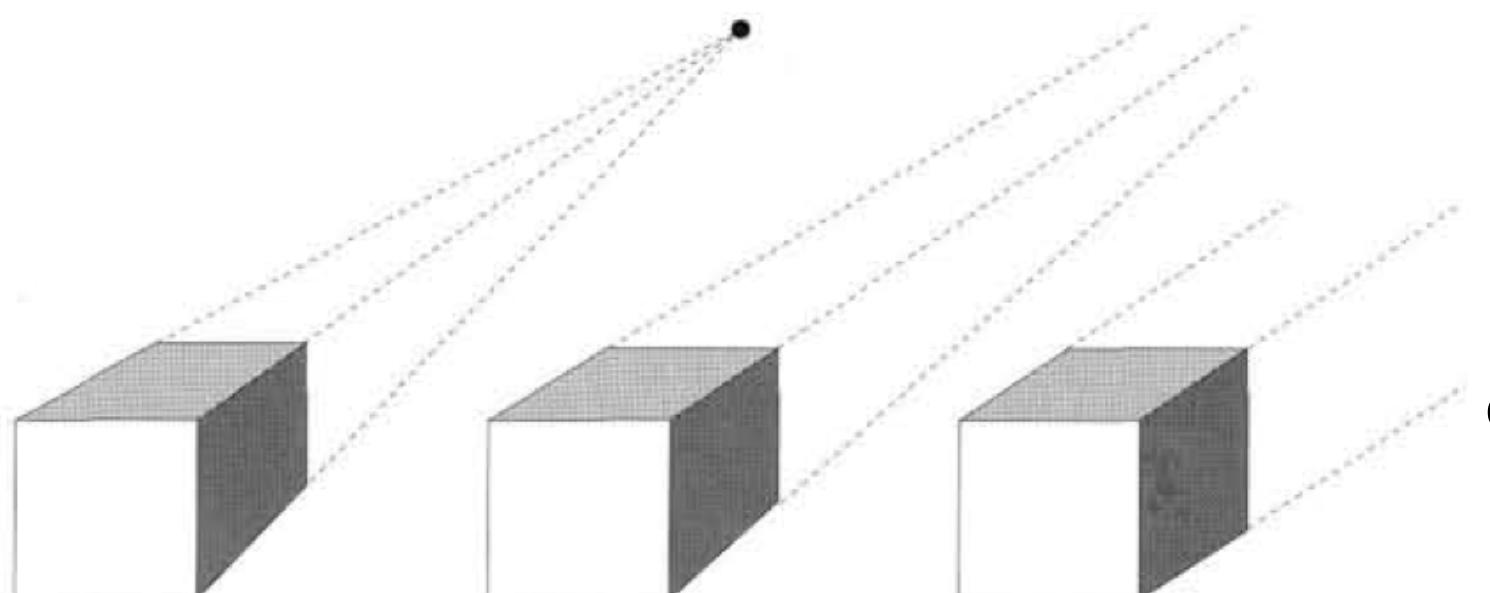
real-world
object



... 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

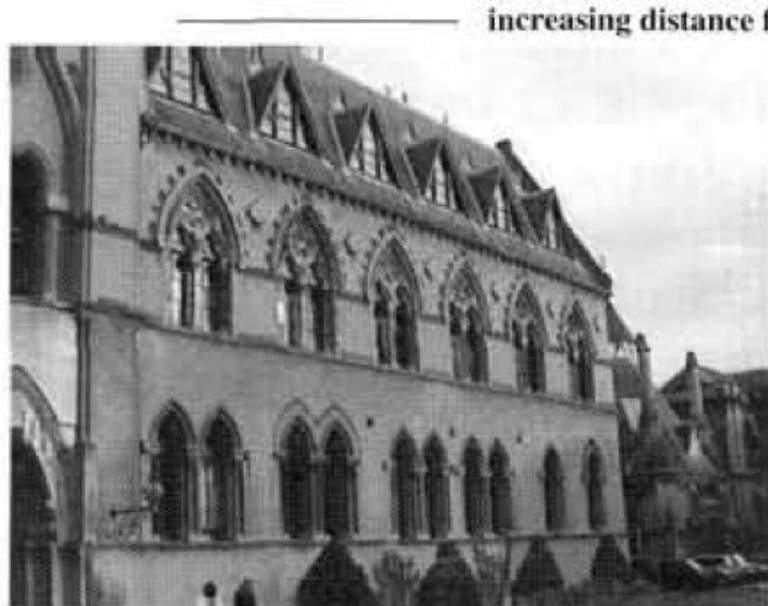


perspective

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

weak perspective

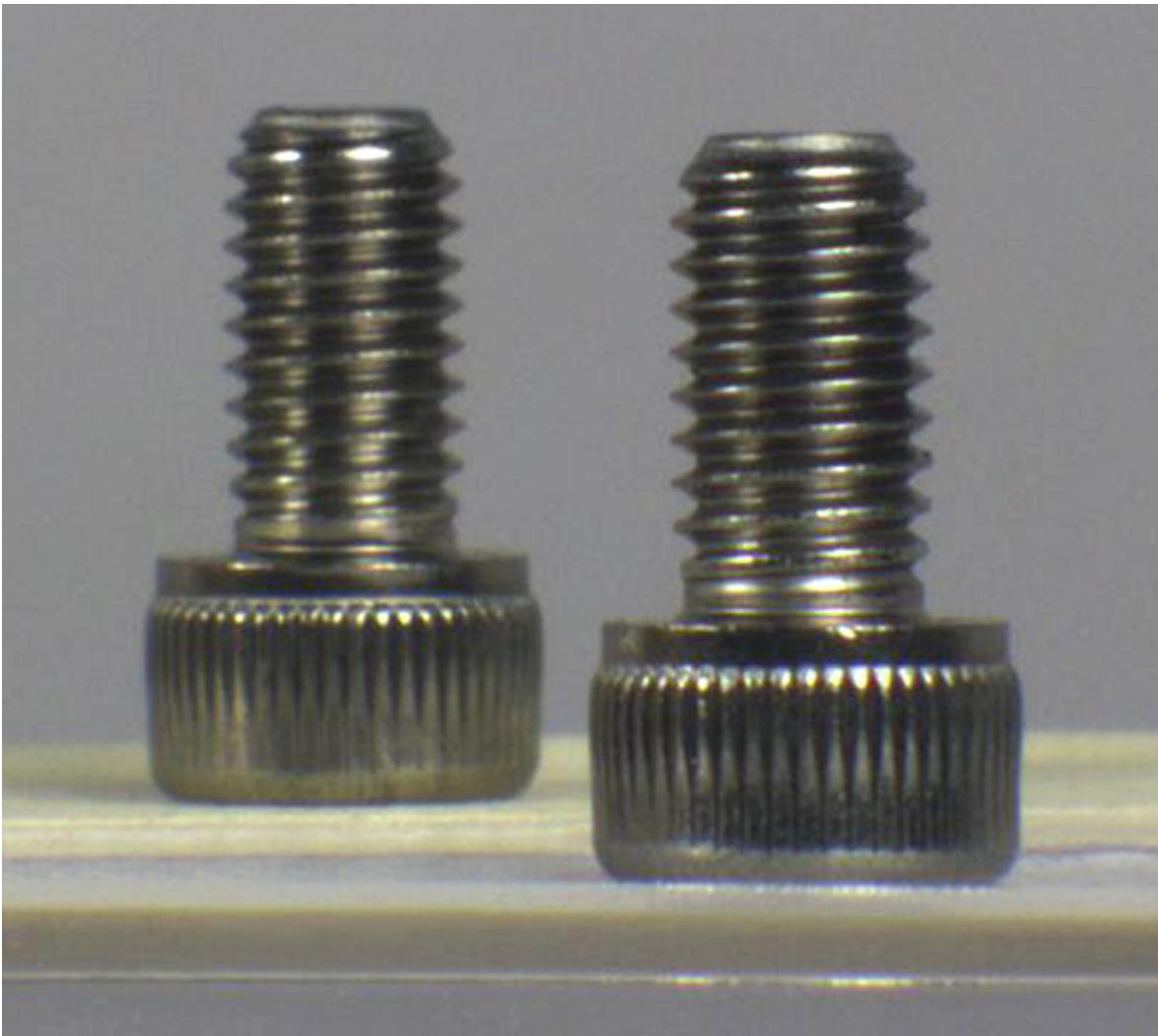
increasing focal length →



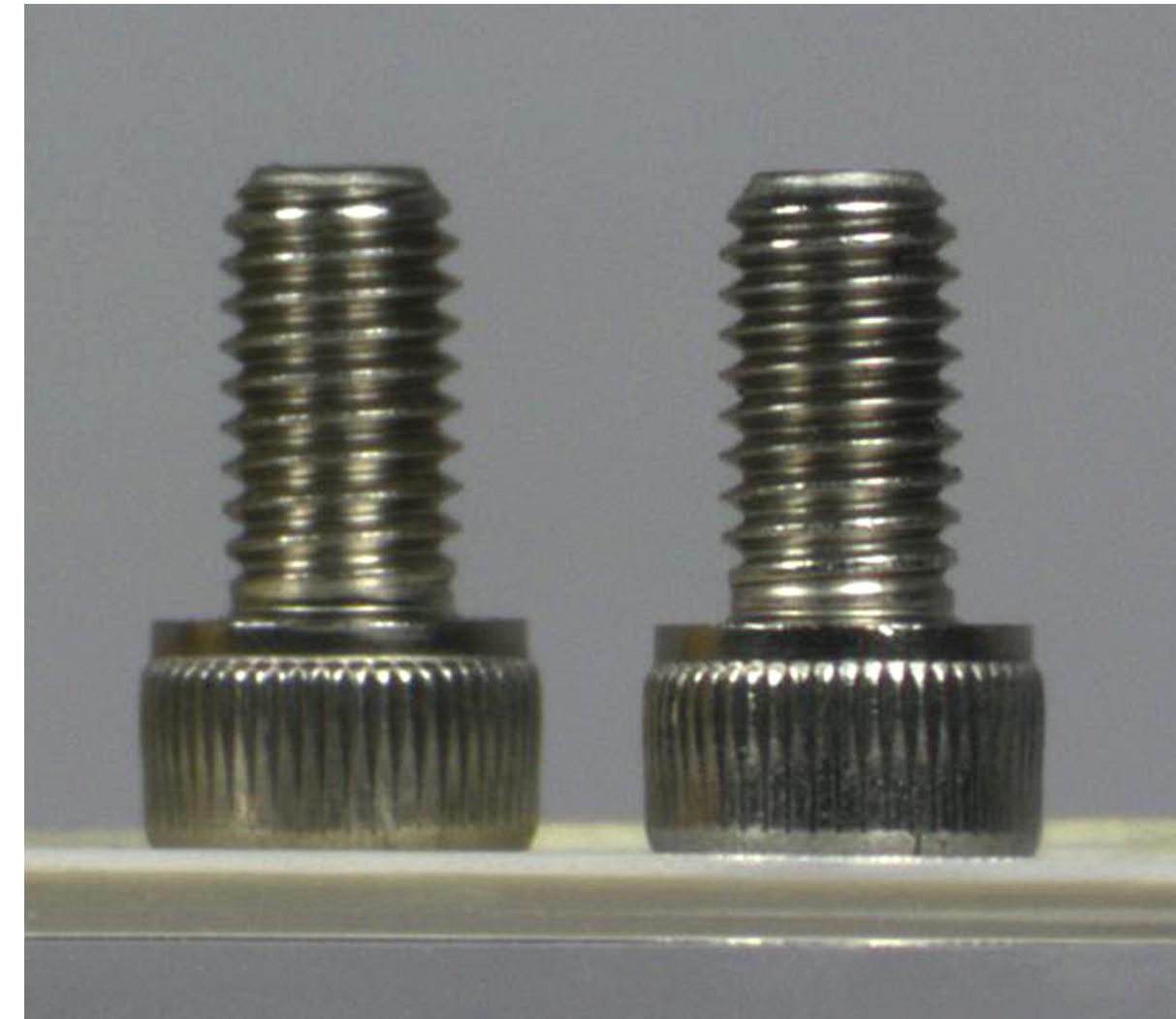
increasing distance from camera →



Different cameras

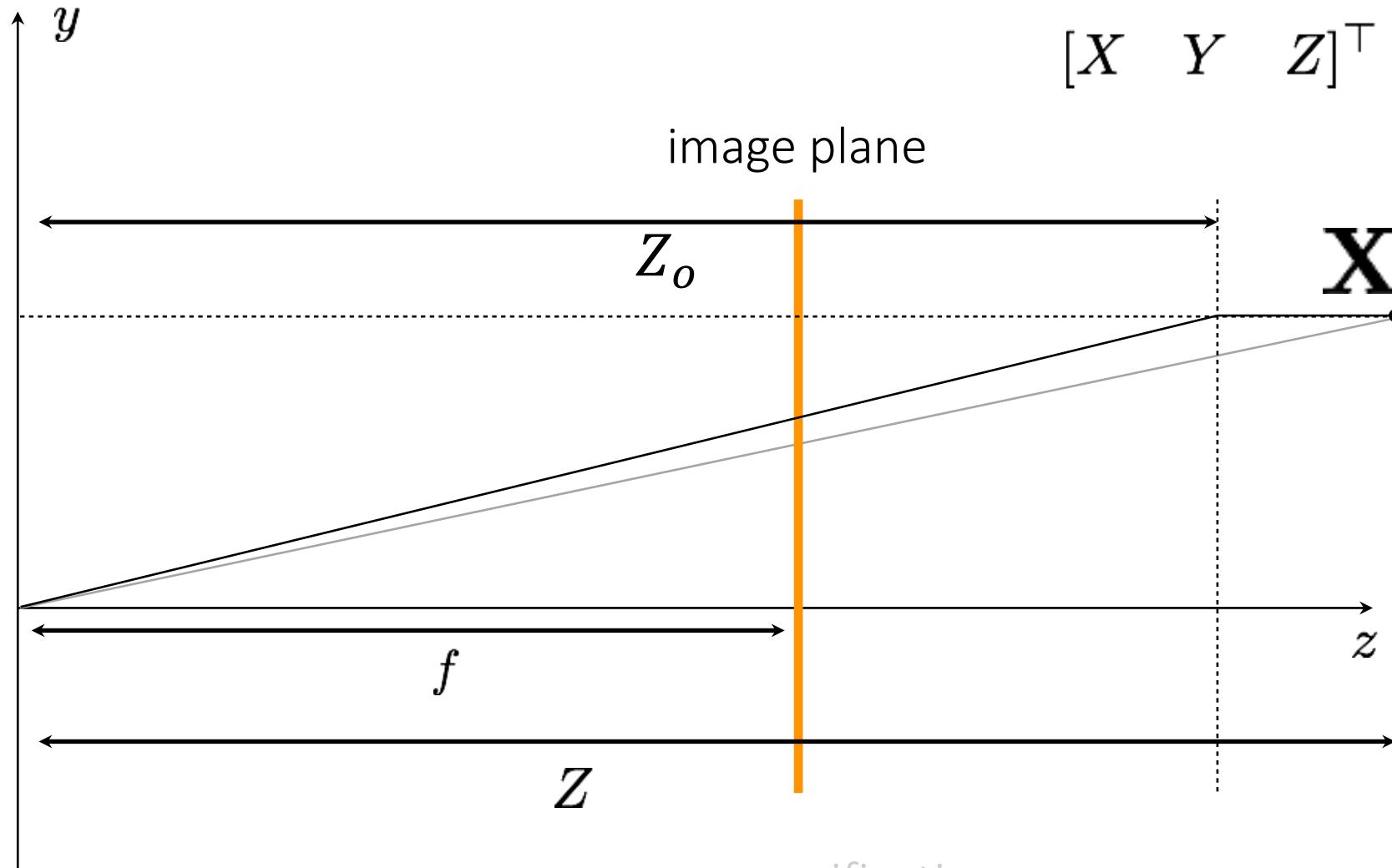


perspective camera



weak perspective camera

Weak perspective vs perspective camera



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

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

magnification
changes with depth

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

Comparing camera matrices

Let's assume that the world and camera coordinate systems are the same.

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

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

- What would the matrix of the weak perspective camera look like?

Comparing camera matrices

Let's assume that the world and camera coordinate systems are the same.

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

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

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

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

Comparing camera matrices

Let's assume that the world and camera coordinate systems are the same.

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

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

where we now have the more general intrinsic matrix

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

In both cameras, we can incorporate extrinsic parameters same as we did before.

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.

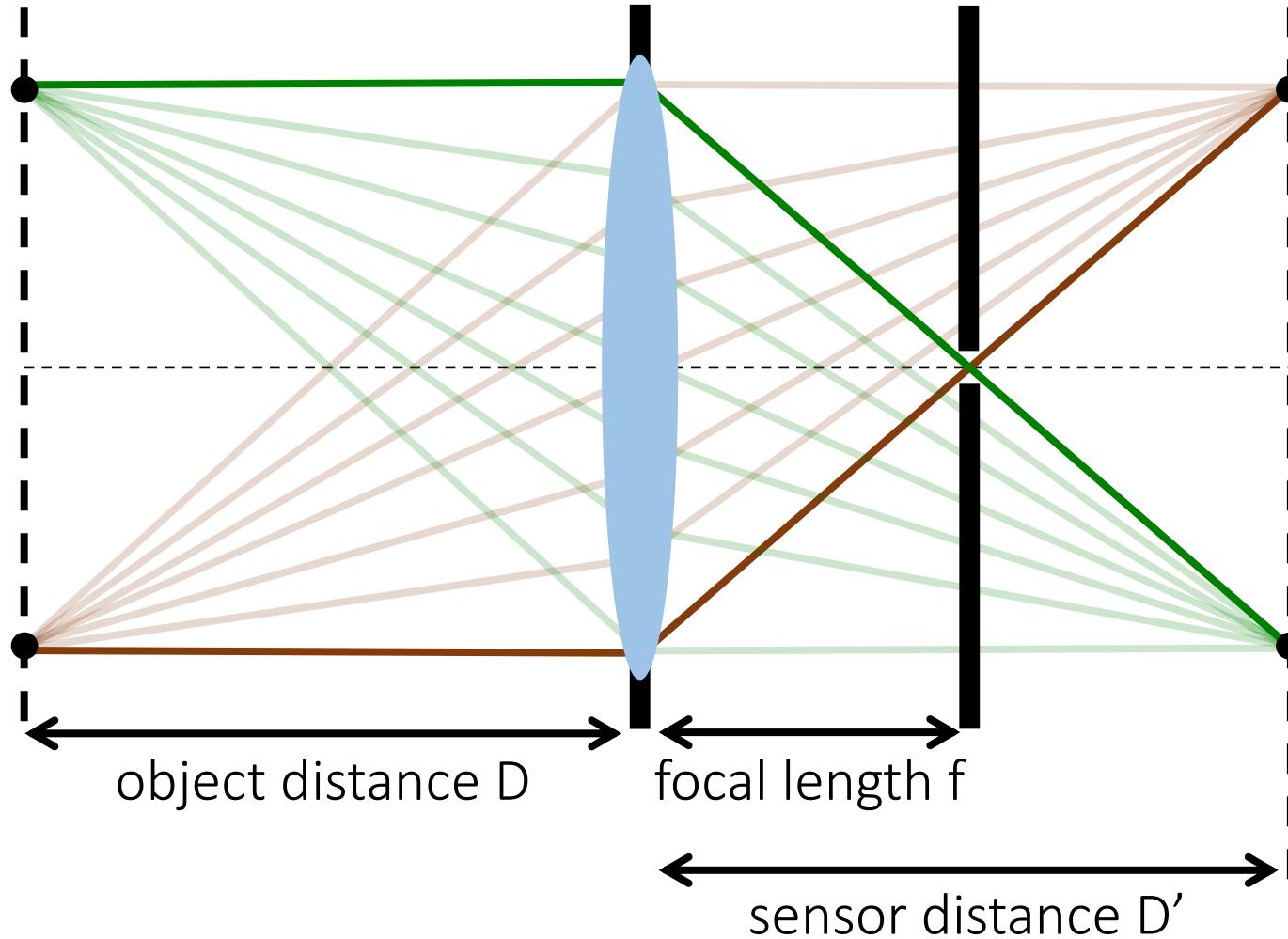


Weak perspective projection applies to the mountains.

When can we assume a weak perspective camera?

2. When we use a telecentric lens.

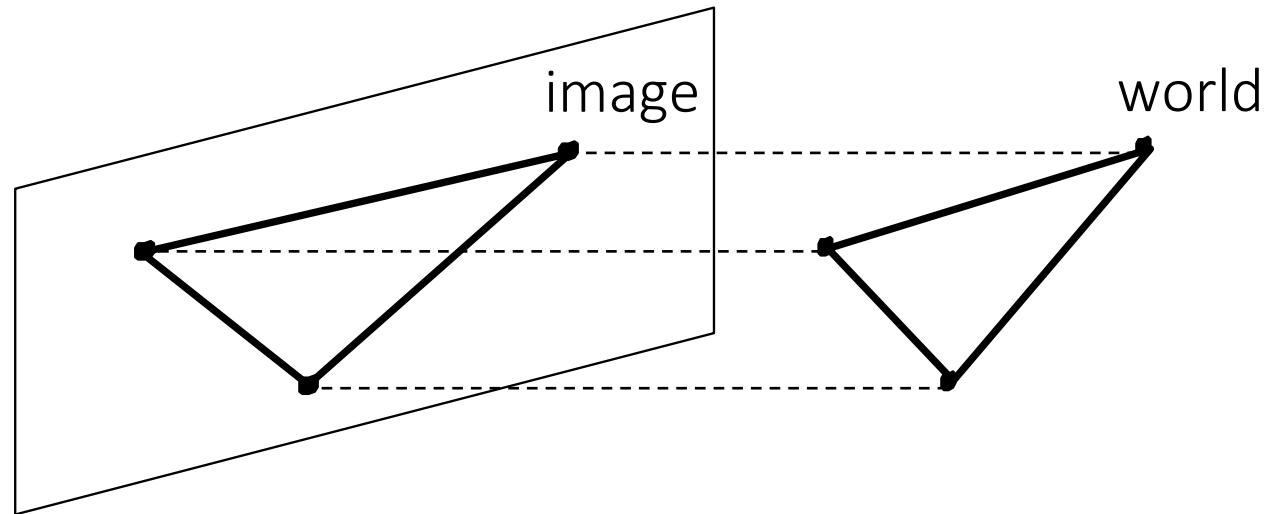
Place a pinhole at focal length, so that only rays parallel to primary ray pass through.



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.

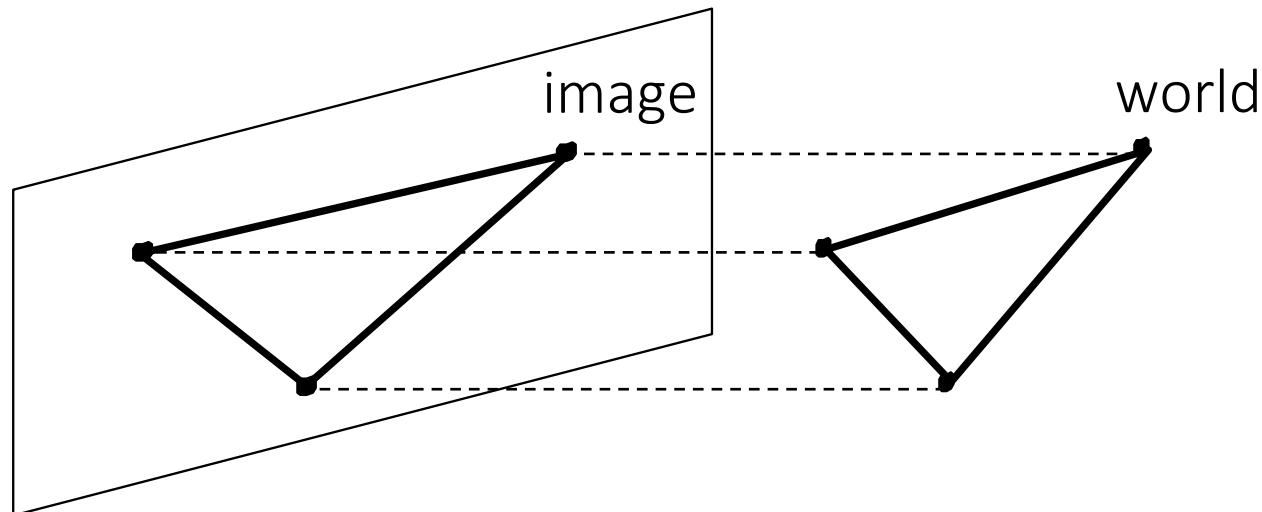


What is the camera matrix in this case?

Orthographic camera

Special case of weak perspective camera where:

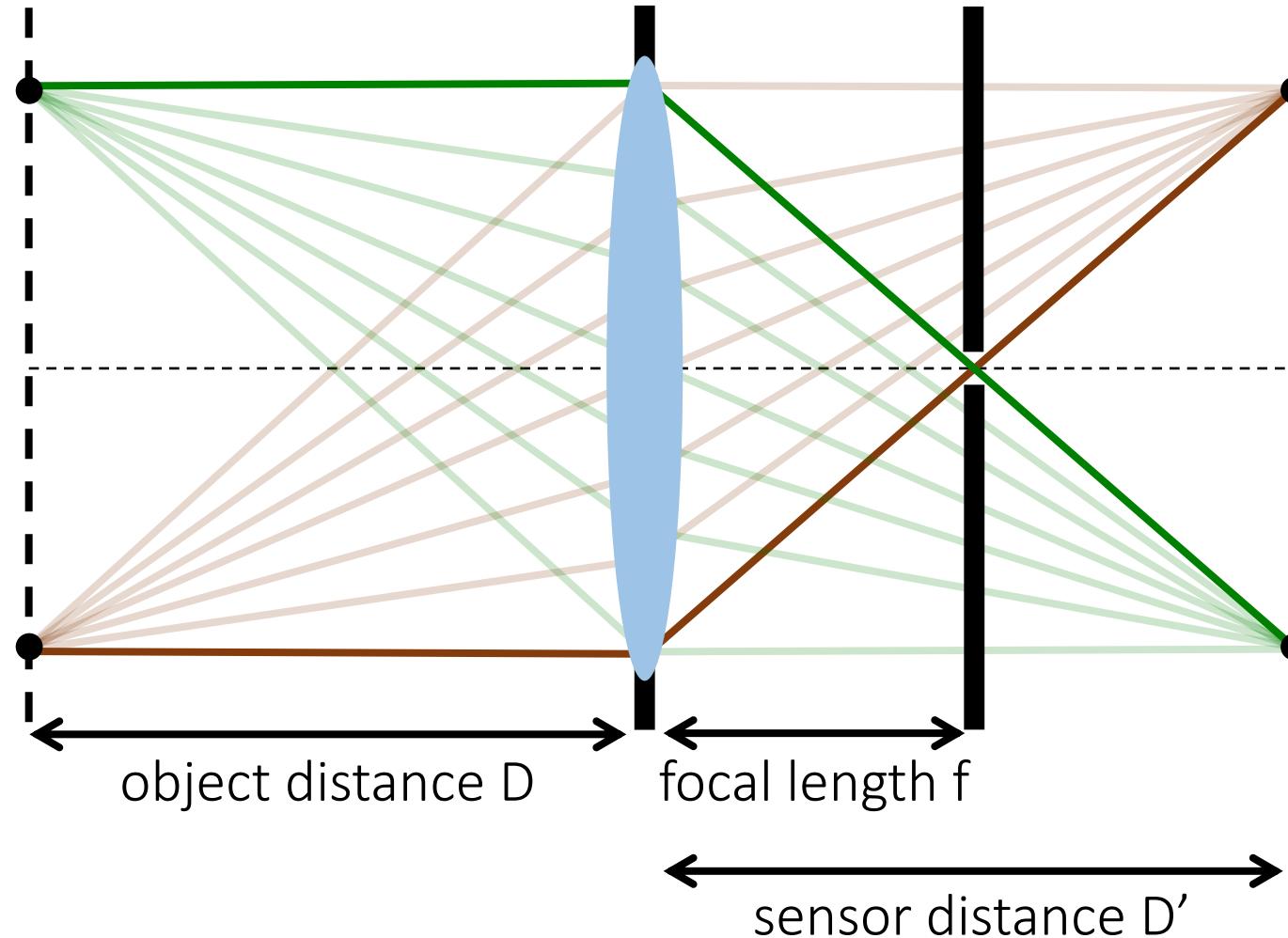
- constant magnification is equal to 1.
- there is no shift between camera and image origins.
- the world and camera coordinate systems are the same.



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

Orthographic projection using a telecentric lens

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

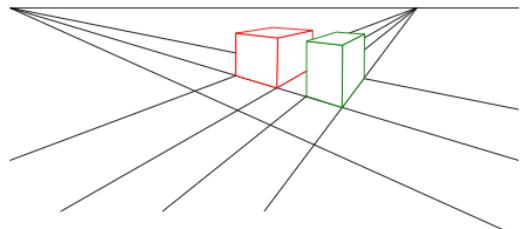


We set the sensor distance as:

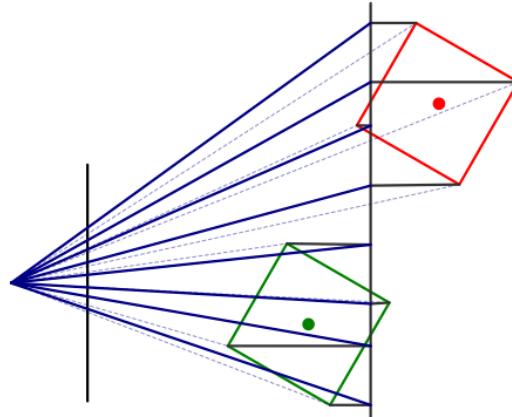
$$D' = 2f$$

in order to achieve unit magnification.

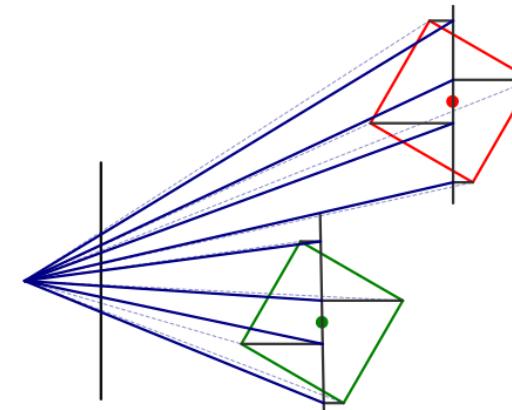
Many other types of cameras



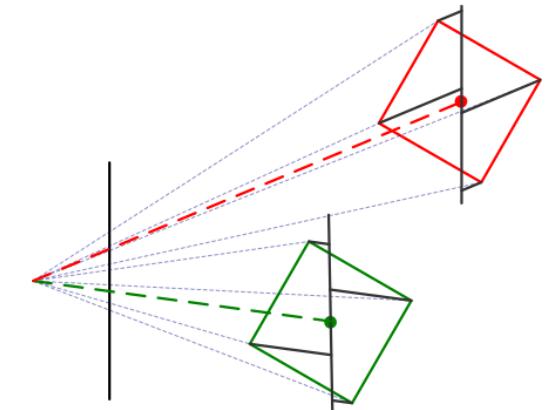
(a) 3D view



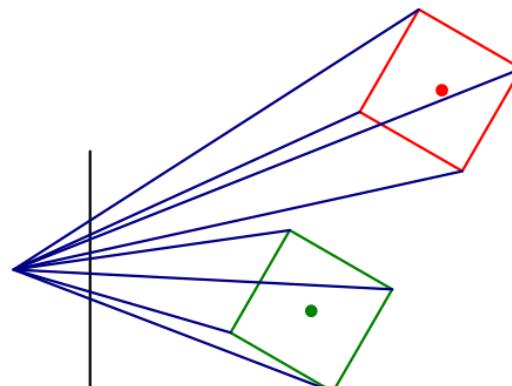
(b) orthography



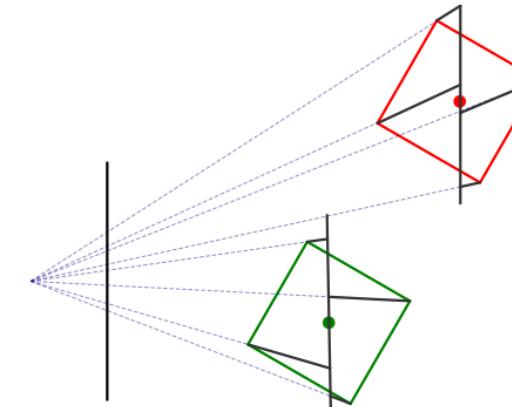
(c) scaled orthography



(d) para-perspective



(e) perspective

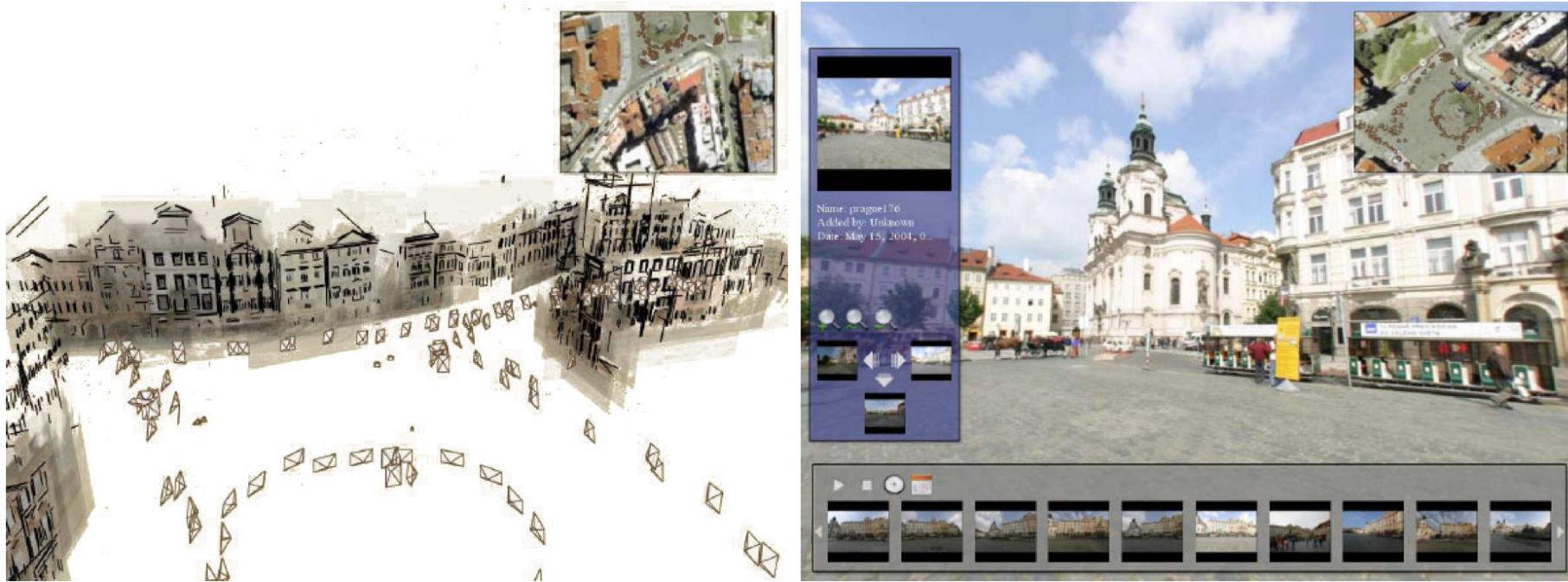


(f) object-centered

Geometric camera calibration

	Structure (scene geometry)	Motion (camera geometry)	Measurements
Camera Calibration (a.k.a. Pose Estimation)	known	estimate	3D to 2D correspondences
Triangulation	estimate	known	2D to 2D coorespondences
Reconstruction	estimate	estimate	2D to 2D coorespondences

Pose Estimation



Given a single image,
estimate the exact position of the photographer

Geometric camera calibration

Given a set of matched points

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

point in 3D
space point in the
image

and camera model

$$\mathbf{x} = \mathbf{f}(\mathbf{X}; \mathbf{p}) = \mathbf{P}\mathbf{X}$$

projection
model parameters Camera
matrix

Find the (pose) estimate of

P

We'll use a **perspective** camera
model for pose estimation

Same setup as homography estimation
(slightly different derivation here)

Mapping between 3D point and image points

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

What are the unknowns?

Mapping between 3D point and image points

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

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

Heterogeneous coordinates

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

(non-linear relation between coordinates)

How can we make these relations linear?

How can we make these relations linear?

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

Make them linear with algebraic manipulation...

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

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

Now we can setup a system of linear equations
with multiple point correspondences

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

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

How do we proceed?

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

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

In matrix form ...

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

How do we proceed?

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

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

In matrix form ...

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

For N points ...

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

*How do we solve
this system?*

Solve for camera matrix by

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|^2 \text{ subject to } \|\mathbf{x}\|^2 = 1$$

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

SVD!

Solve for camera matrix by

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|^2 \text{ subject to } \|\mathbf{x}\|^2 = 1$$

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

Solution \mathbf{x} is the column of \mathbf{V}
corresponding to smallest singular
value of

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

Solve for camera matrix by

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|^2 \text{ subject to } \|\mathbf{x}\|^2 = 1$$

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

Equivalently, solution \mathbf{x} is the
Eigenvector corresponding to
smallest Eigenvalue of

$$\mathbf{A}^\top \mathbf{A}$$

Now we have:

$$\mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix}$$

Are we done?

Almost there ...

$$\mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix}$$

How do you get the intrinsic and extrinsic parameters from the projection matrix?

Decomposition of the Camera Matrix

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

Decomposition of the Camera Matrix

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

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

Decomposition of the Camera Matrix

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

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

Decomposition of the Camera Matrix

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

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

Find the camera center **C**

What is the projection of the camera center?

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

Decomposition of the Camera Matrix

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

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

Find the camera center \mathbf{c}

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

How do we compute the camera center from this?

Find intrinsic \mathbf{K} and rotation \mathbf{R}

Decomposition of the Camera Matrix

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

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

Find the camera center \mathbf{c}

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

SVD of \mathbf{P} !

\mathbf{c} is the Eigenvector corresponding to
smallest Eigenvalue

Find intrinsic \mathbf{K} and rotation \mathbf{R}

Decomposition of the Camera Matrix

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

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

Find the camera center \mathbf{c}

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

SVD of P!

\mathbf{c} is the Eigenvector corresponding to
smallest Eigenvalue

Find intrinsic \mathbf{K} and rotation \mathbf{R}

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

*Any useful properties of K
and R we can use?*

Decomposition of the Camera Matrix

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

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

Find the camera center \mathbf{c}

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

SVD of P!

\mathbf{c} is the Eigenvector corresponding to
smallest Eigenvalue

Find intrinsic \mathbf{K} and rotation \mathbf{R}

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

right upper triangle orthogonal

*How do we find K
and R?*

Decomposition of the Camera Matrix

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

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

Find the camera center \mathbf{c}

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

SVD of P!

\mathbf{c} is the Eigenvector corresponding to
smallest Eigenvalue

Find intrinsic \mathbf{K} and rotation \mathbf{R}

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

QR decomposition

Geometric camera calibration

Given a set of matched points

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

point in 3D
space point in the
image

*Where do we get these
matched points from?*

and camera model

$$\mathbf{x} = \mathbf{f}(\mathbf{X}; \mathbf{p}) = \mathbf{P}\mathbf{X}$$

projection
model parameters Camera
matrix

Find the (pose) estimate of

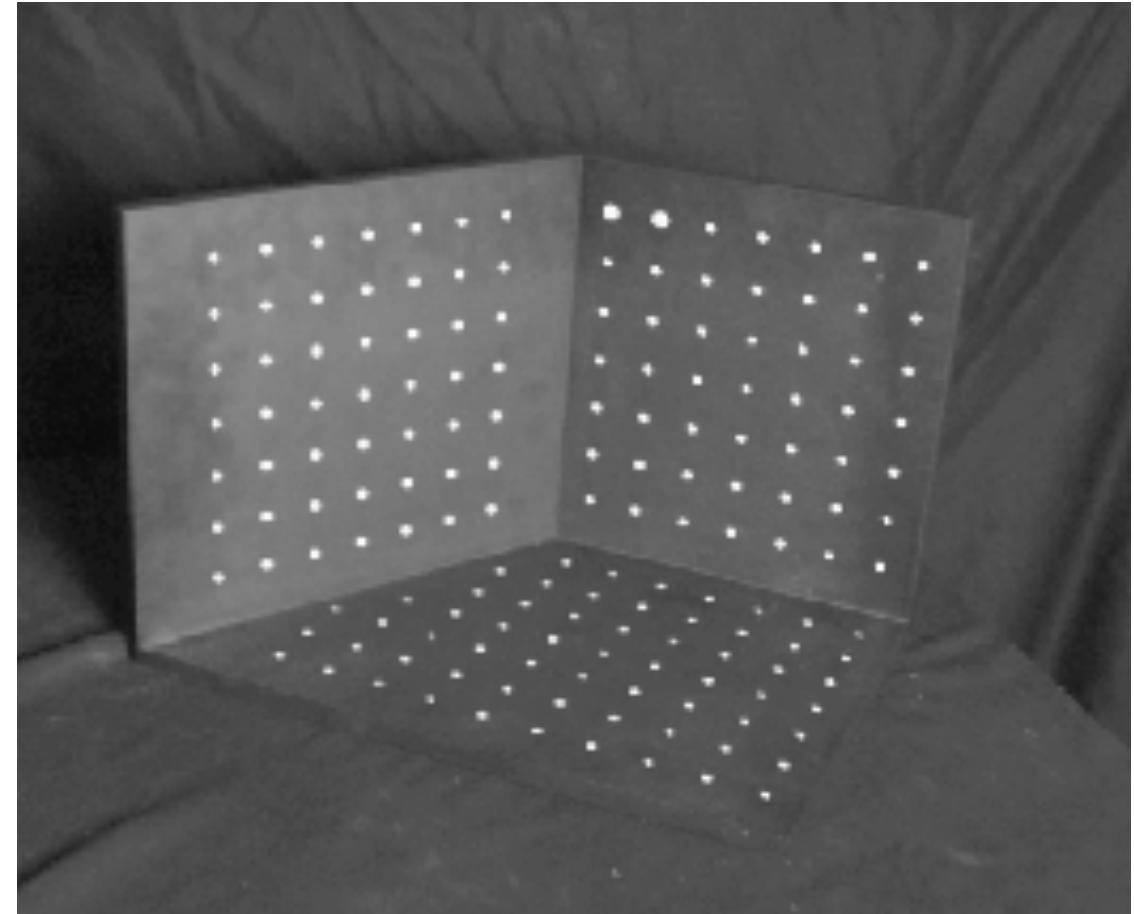
P

We'll use a **perspective** camera
model for pose estimation

Calibration using a reference object

Place a known object in the scene:

- identify correspondences between image and scene
- compute mapping from scene to image



Issues:

- must know geometry very accurately
- must know 3D->2D correspondence

Geometric camera calibration

Advantages:

- Very simple to formulate.
- Analytical solution.

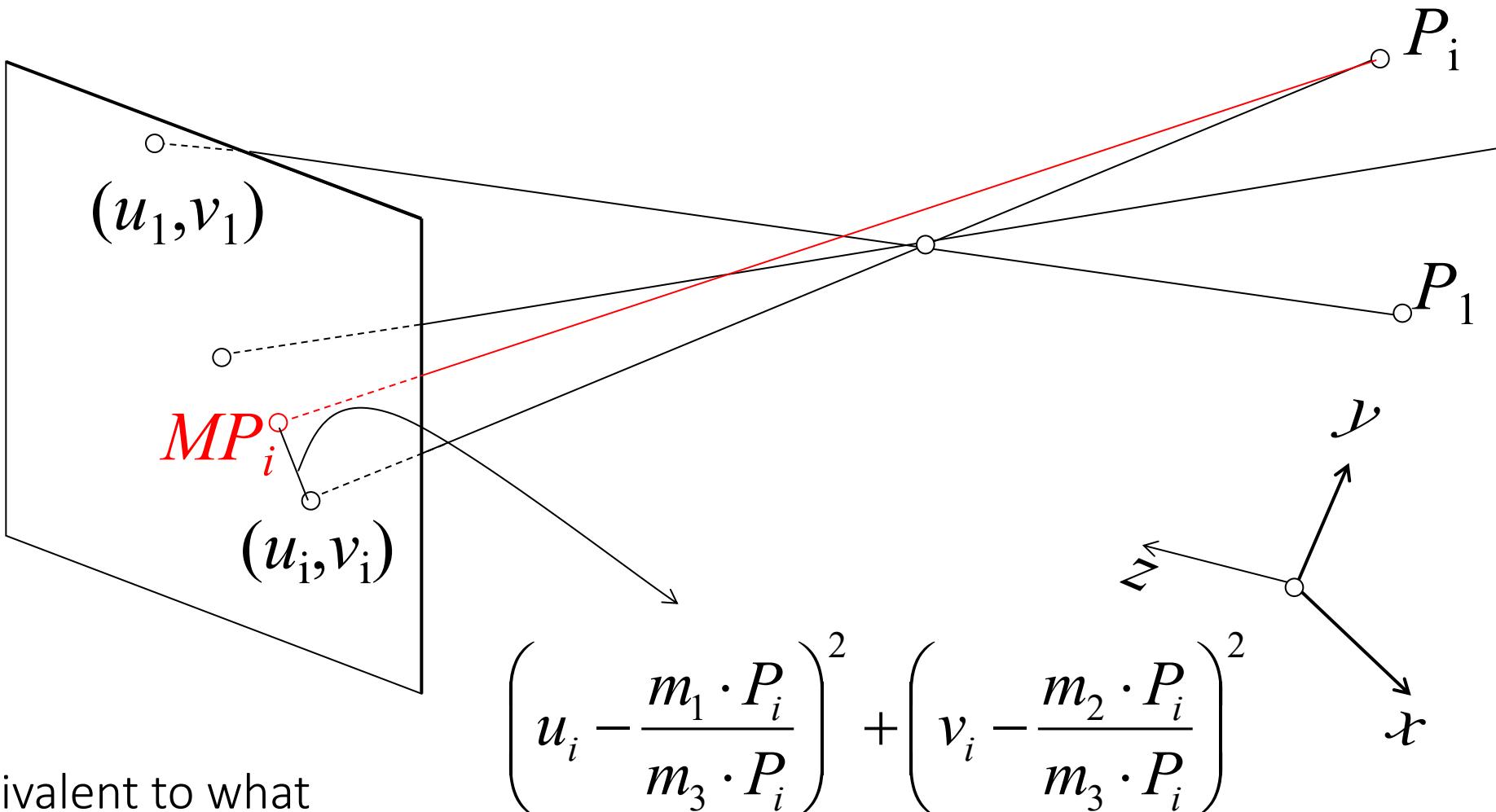
Disadvantages:

- Doesn't model radial distortion.
- Hard to impose constraints (e.g., known f).
- Doesn't minimize the correct error function.

For these reasons, *nonlinear methods* are preferred

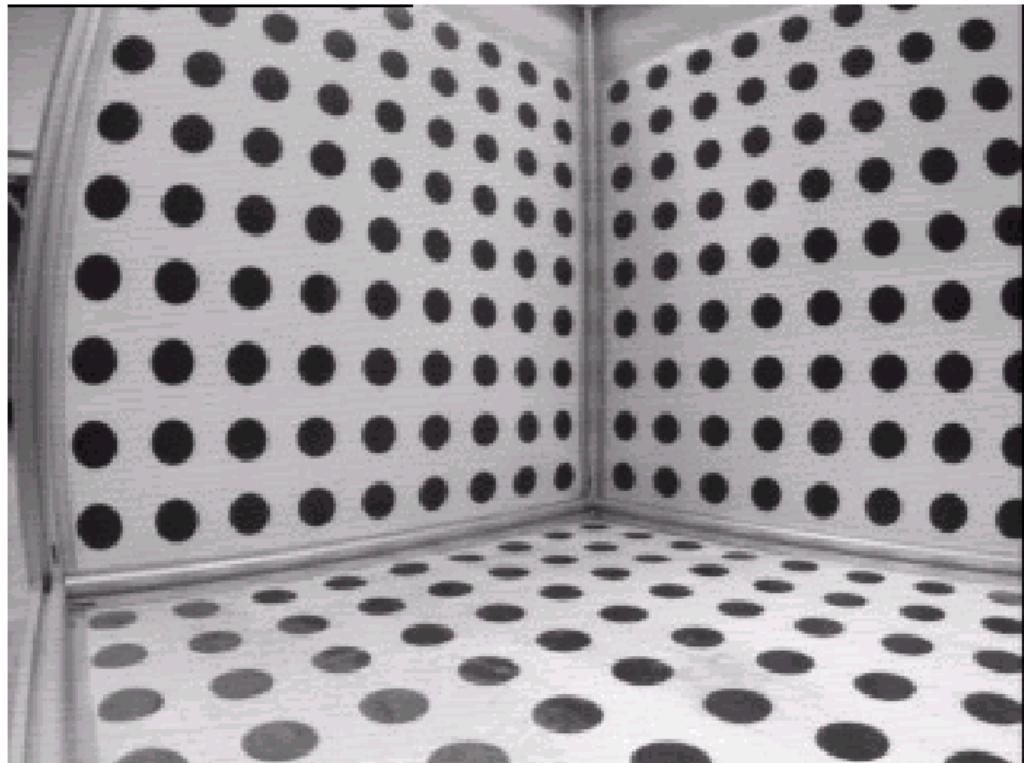
- Define error function E between projected 3D points and image positions
 - E is nonlinear function of intrinsics, extrinsics, radial distortion
- Minimize E using nonlinear optimization techniques

Minimizing reprojection error

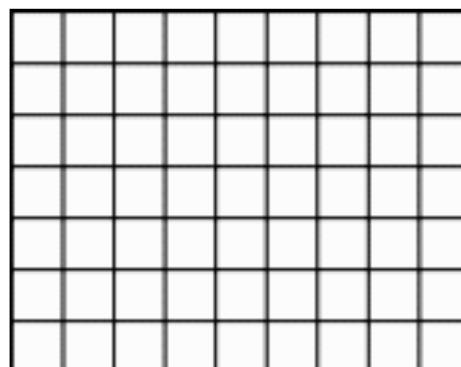


Is this equivalent to what
we were doing previously?

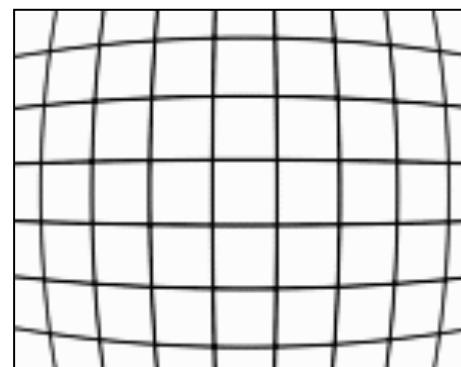
Radial distortion



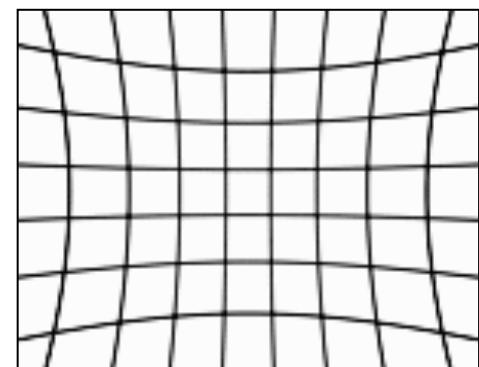
What causes this distortion?



no distortion

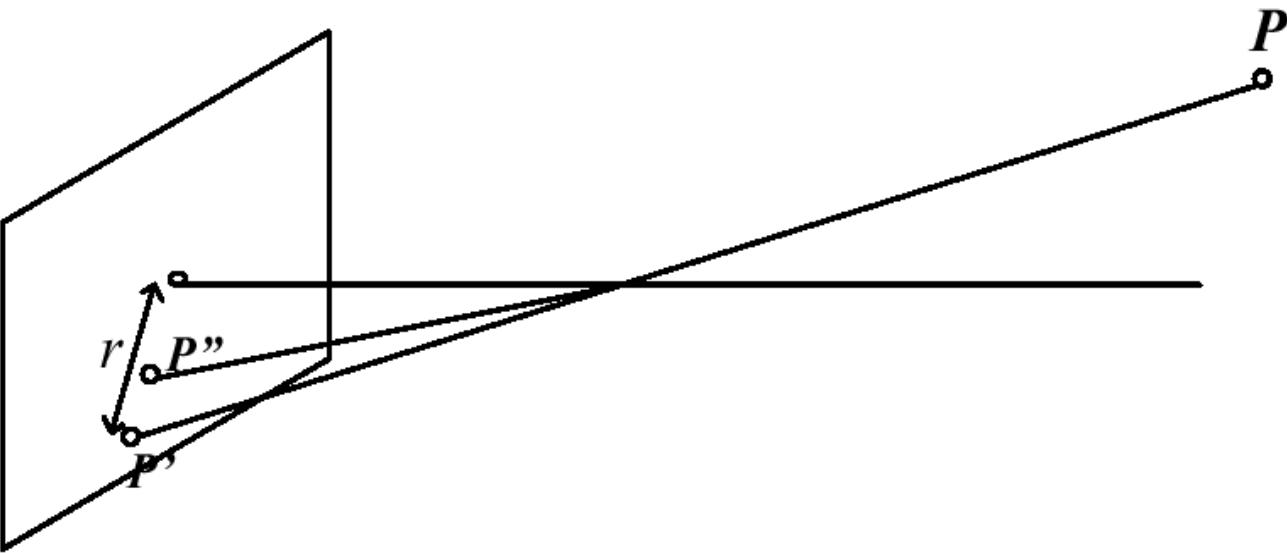


barrel distortion



pincushion distortion

Radial distortion model



Ideal:

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

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

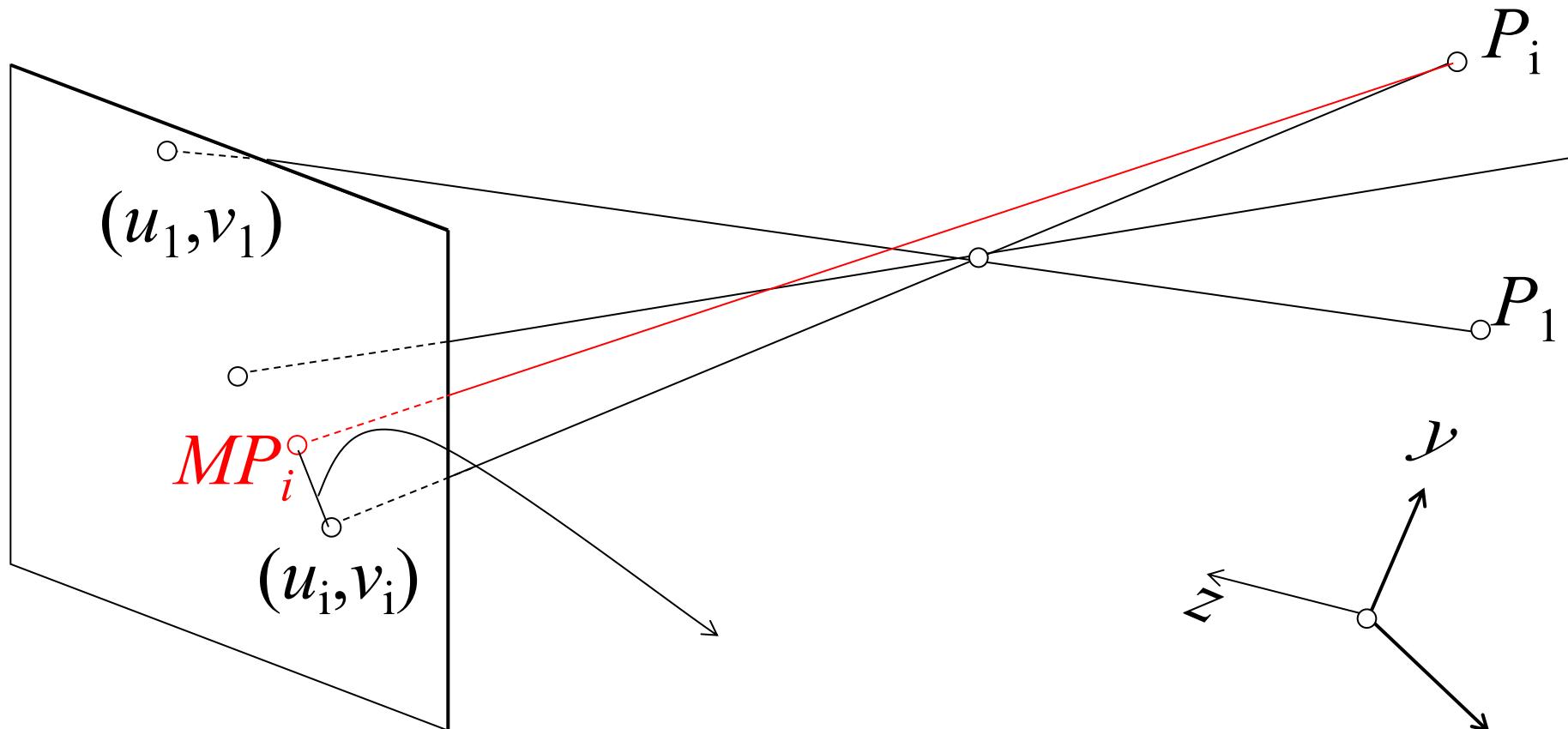
Distorted:

$$x'' = \frac{1}{\lambda} x'$$

$$y'' = \frac{1}{\lambda} y'$$

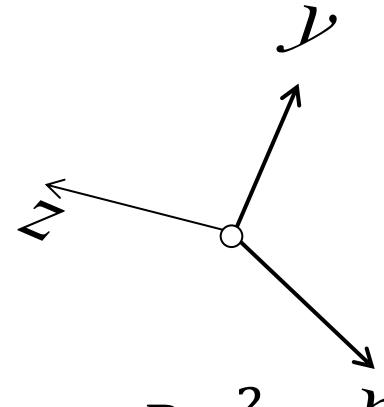
$$\lambda = 1 + k_1 r^2 + k_2 r^4 + \dots$$

Minimizing reprojection error with radial distortion



Add distortions to
reprojection error:

$$\left(u_i - \frac{1}{\lambda} \frac{m_1 \cdot P_i}{m_3 \cdot P_i} \right)^2 + \left(v_i - \frac{1}{\lambda} \frac{m_2 \cdot P_i}{m_3 \cdot P_i} \right)^2$$



Correcting radial distortion

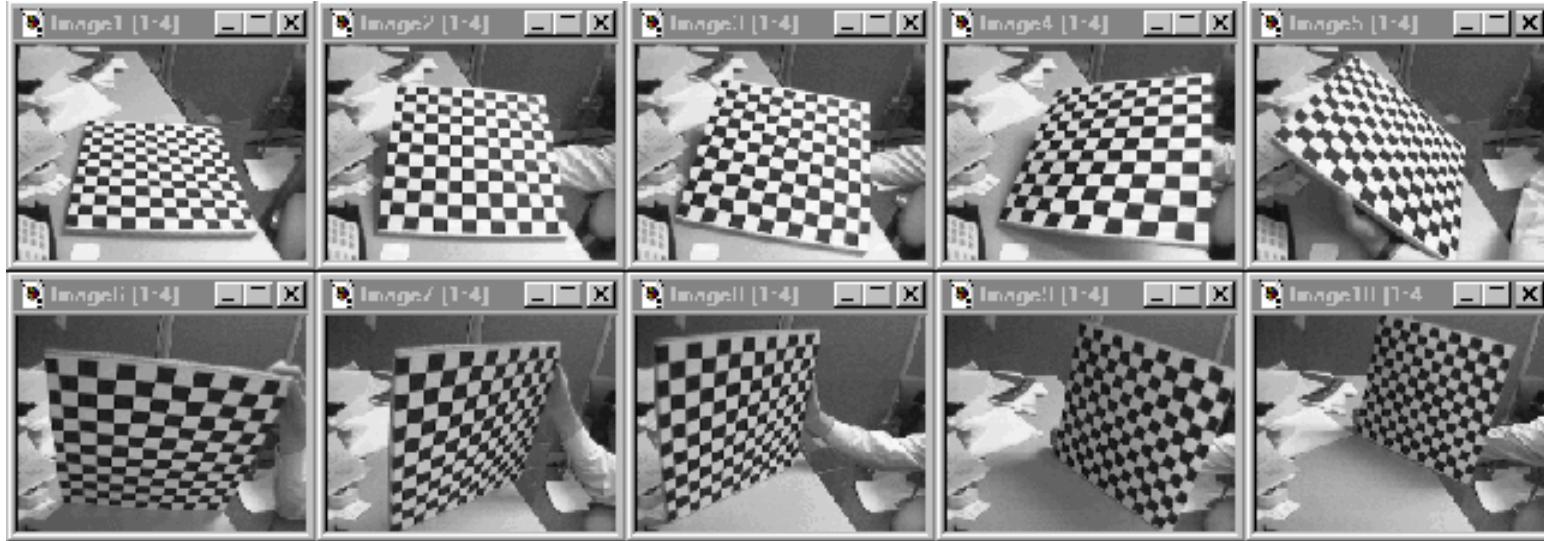


before



after

Alternative: Multi-plane calibration

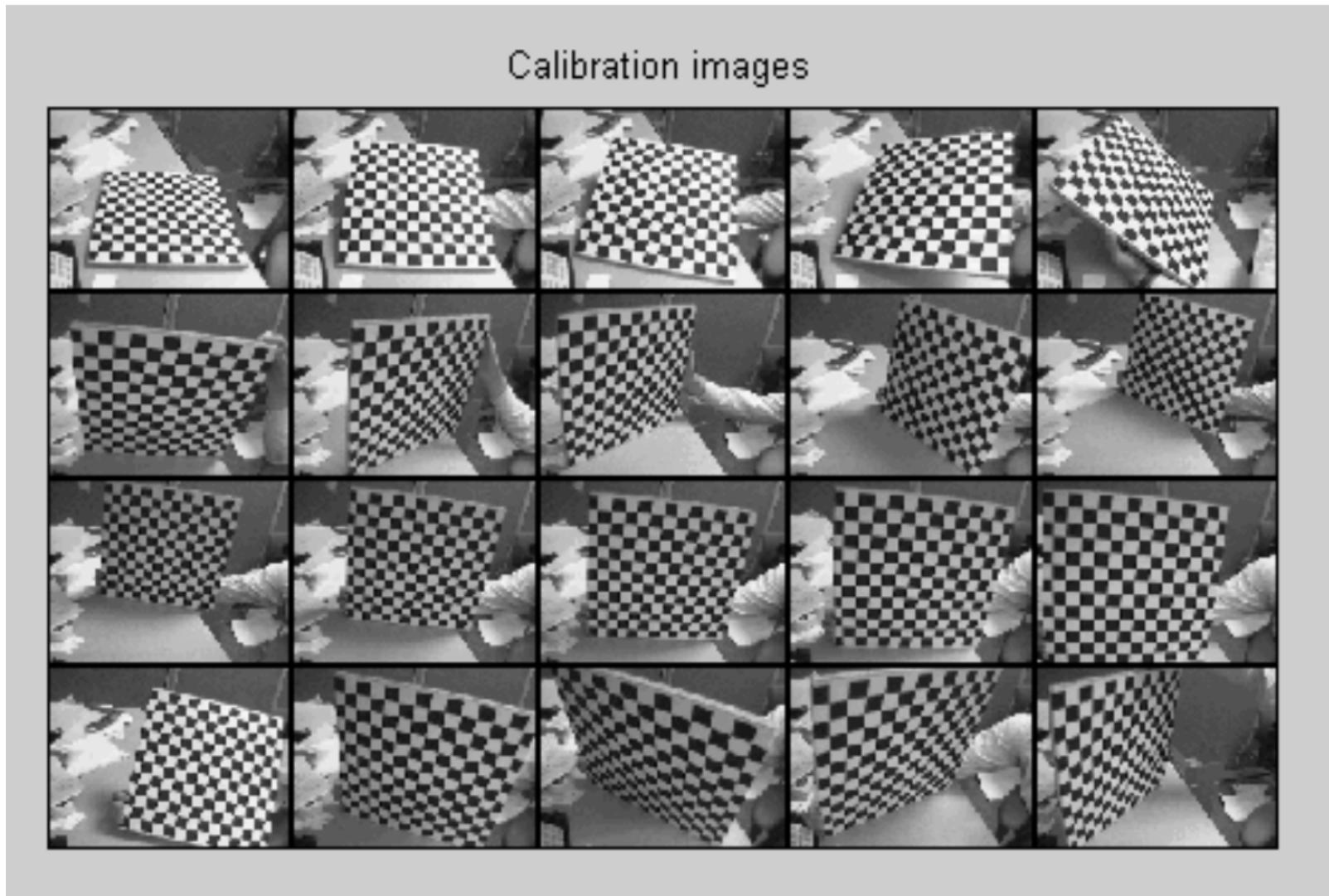


Advantages:

- Only requires a plane
- Don't have to know positions/orientations
- Great code available online!
 - Matlab version: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
 - Also available on OpenCV.

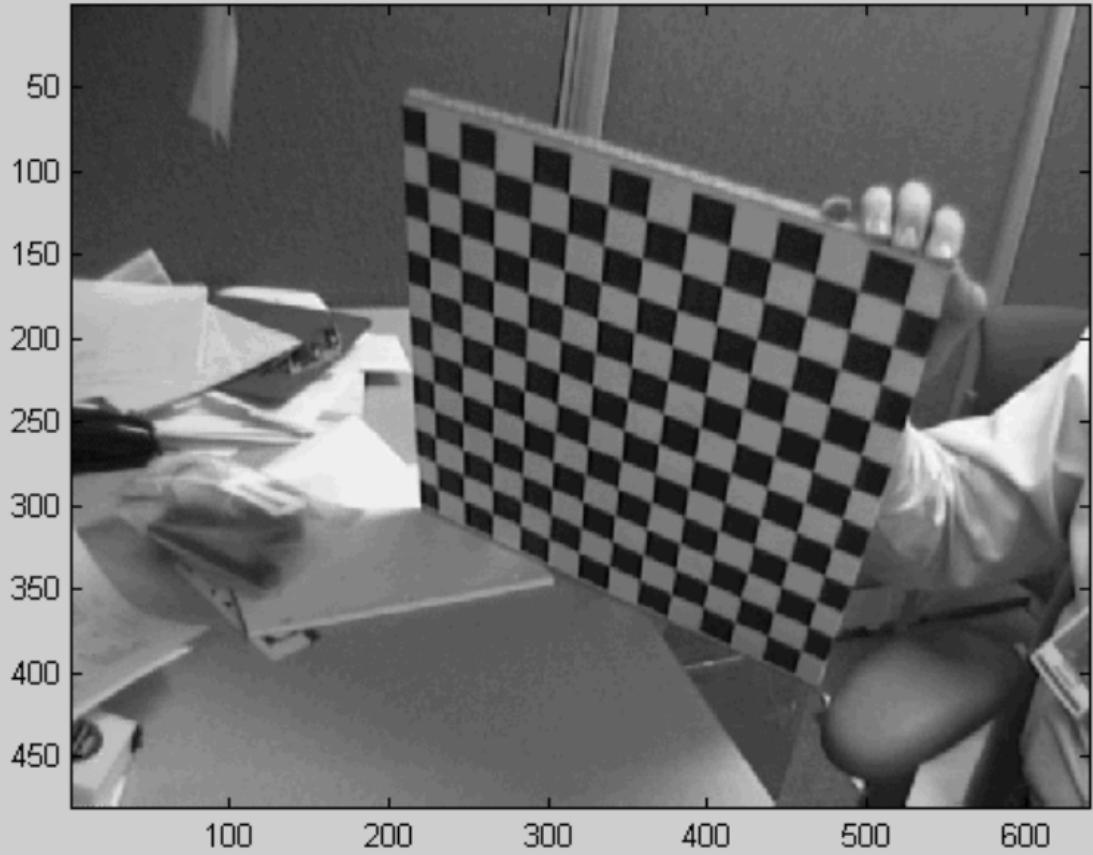
Disadvantage: Need to solve non-linear optimization problem.

Step-by-step demonstration

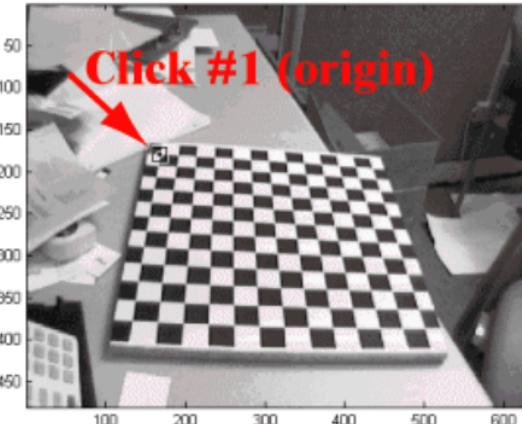


Step-by-step demonstration

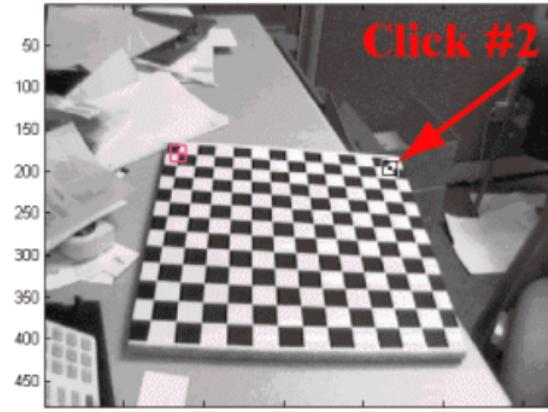
Click on the four extreme corners of the rectangular pattern...



Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



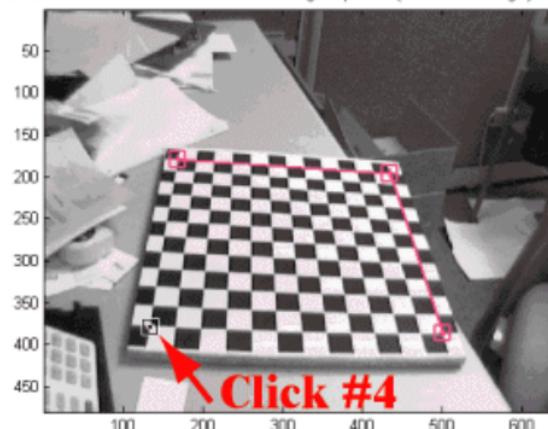
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



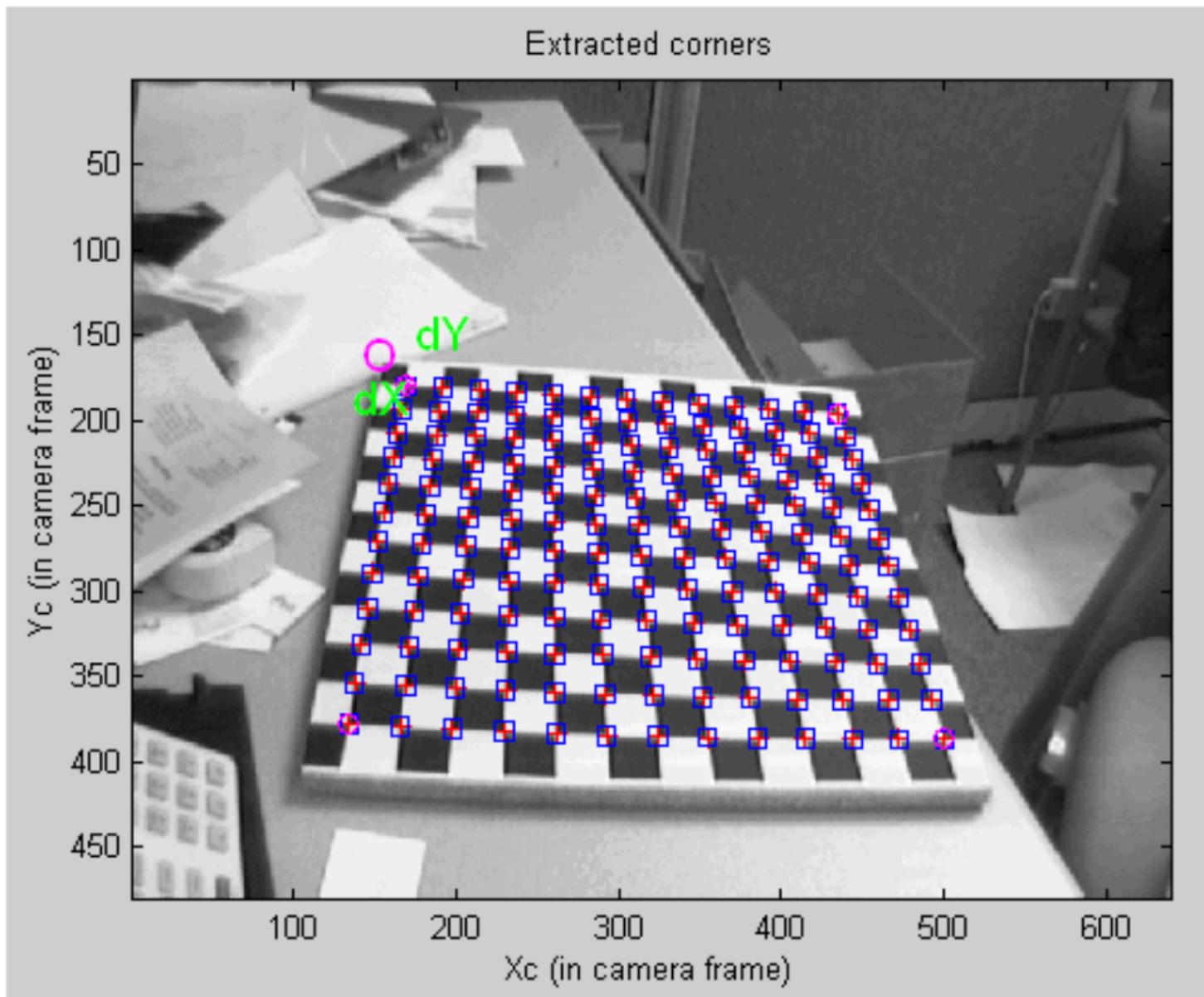
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



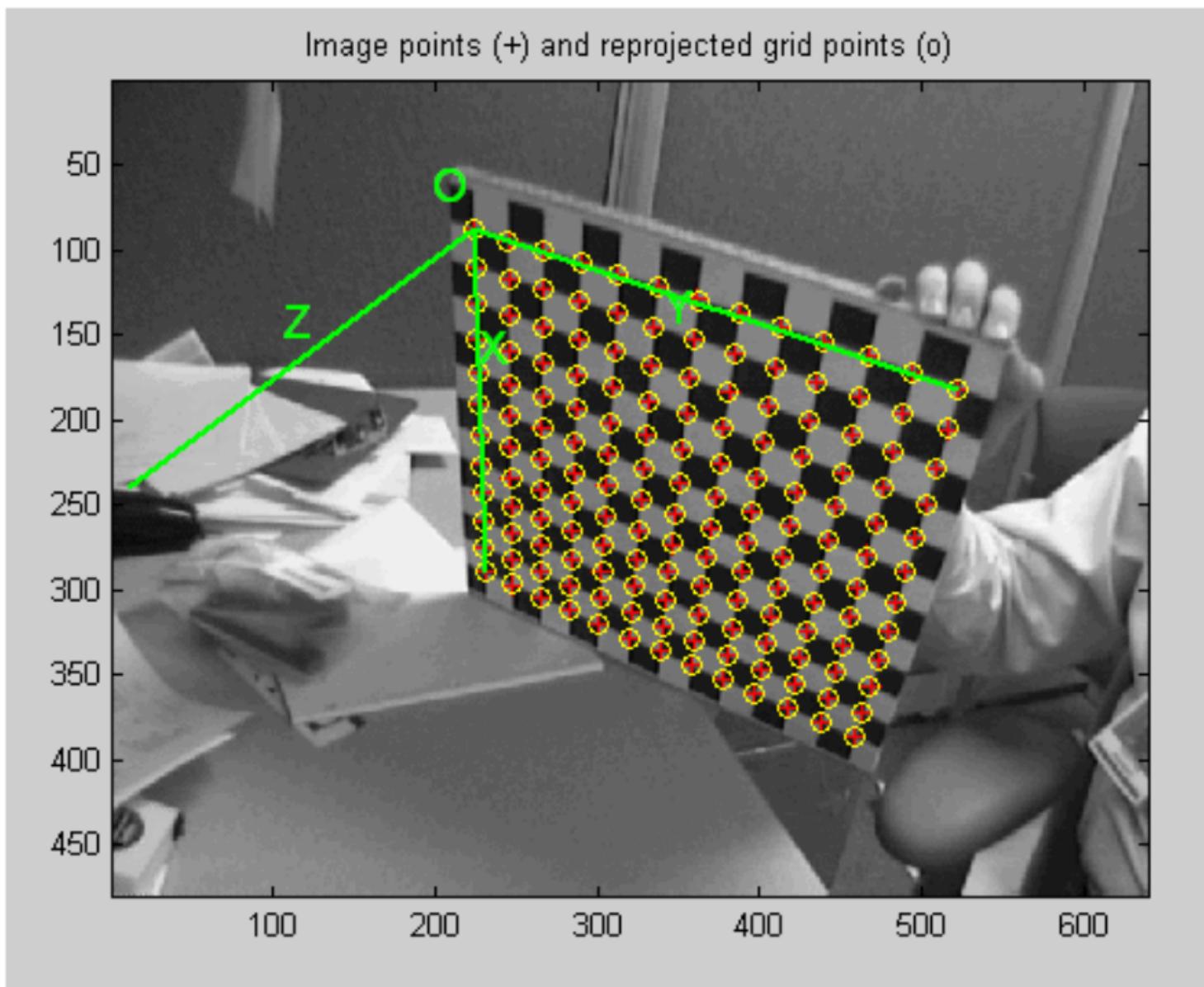
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



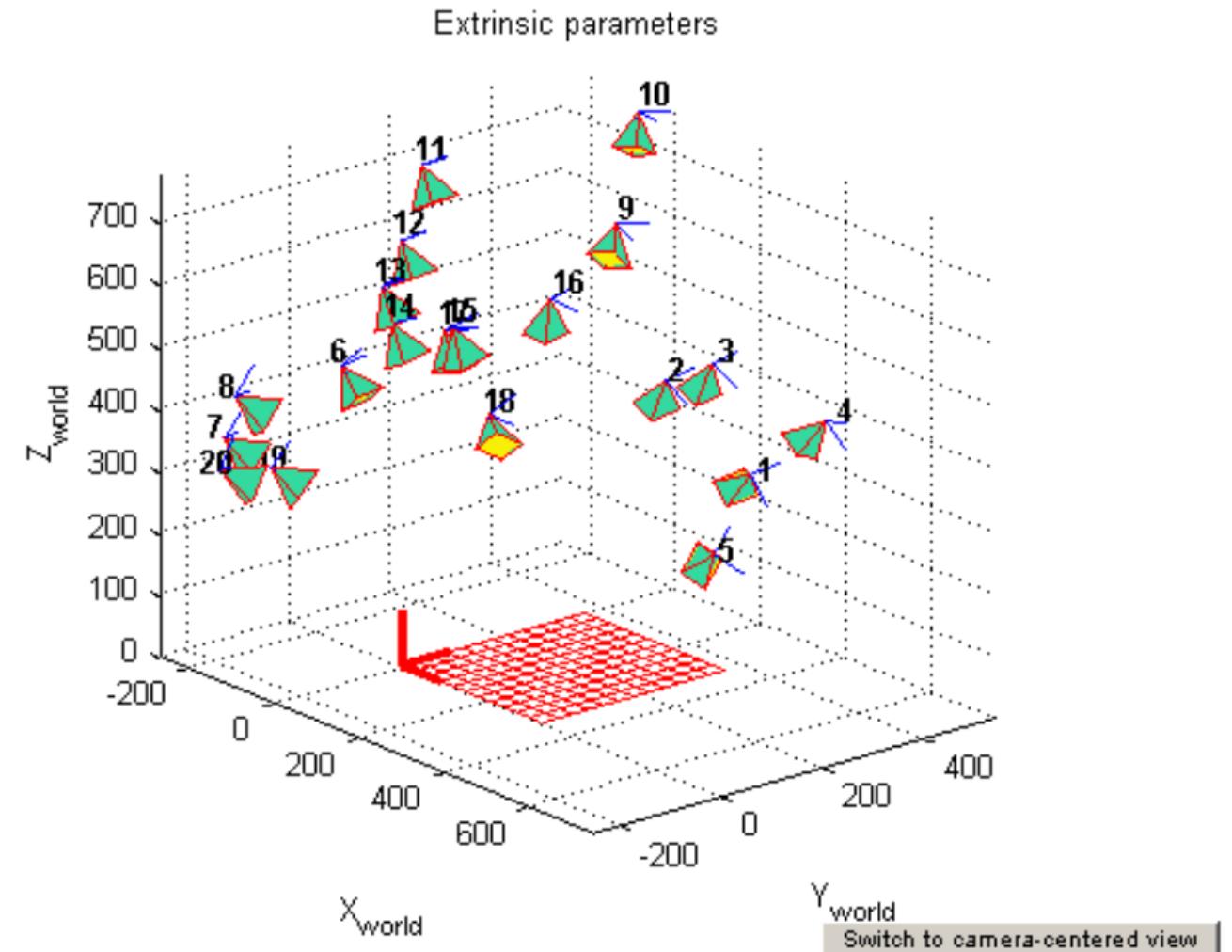
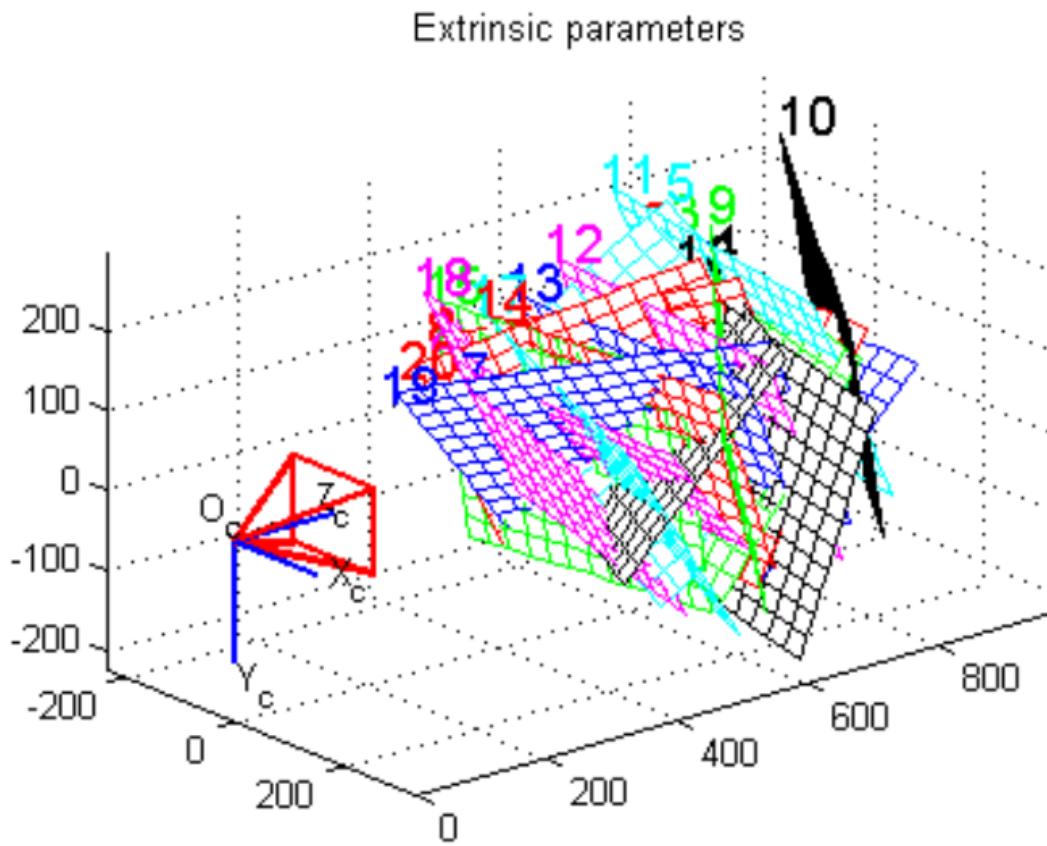
Step-by-step demonstration



Step-by-step demonstration



Step-by-step demonstration



Switch to camera-centered view

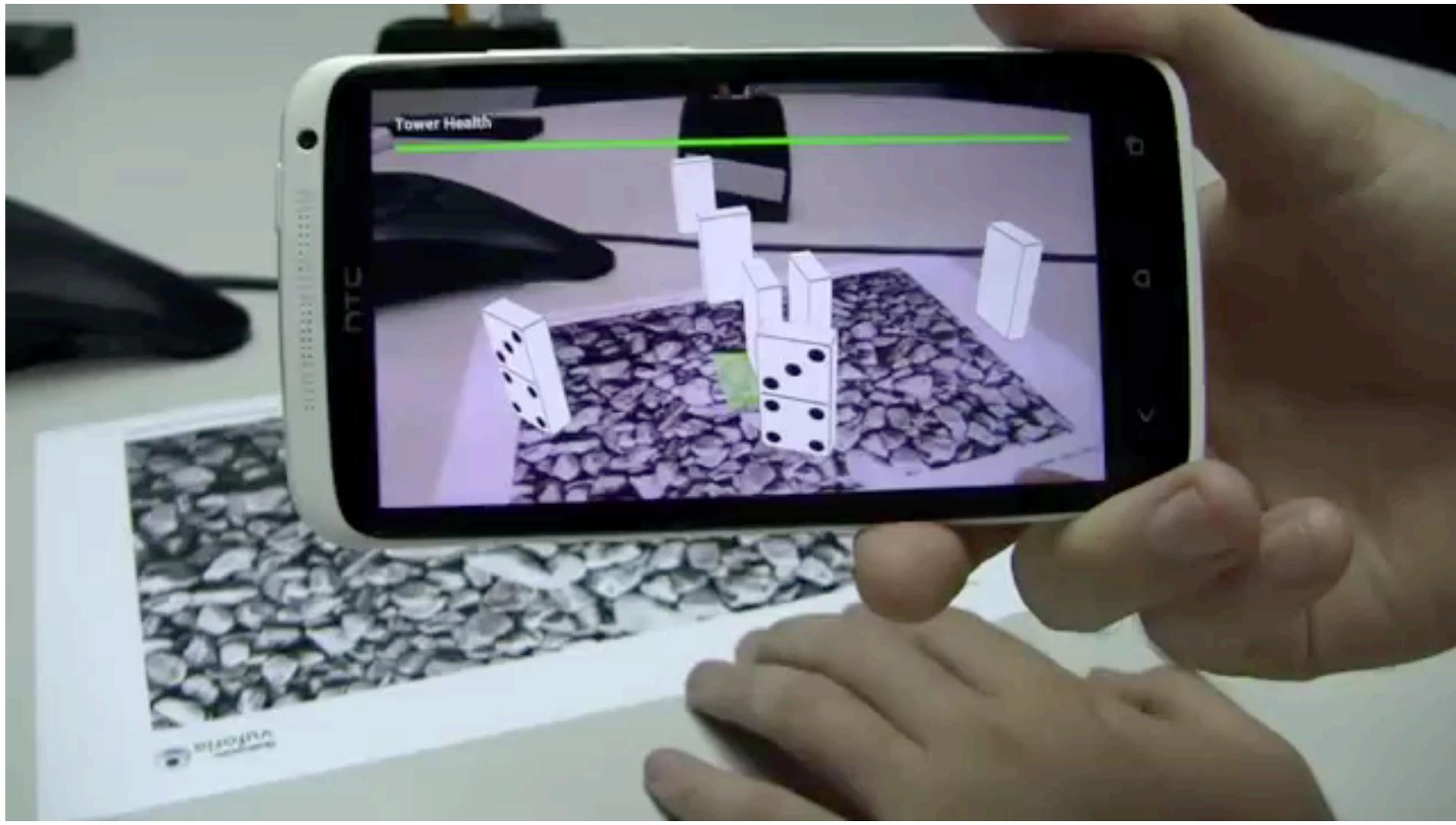
What does it mean to “calibrate a camera”?

What does it mean to “calibrate a camera”?

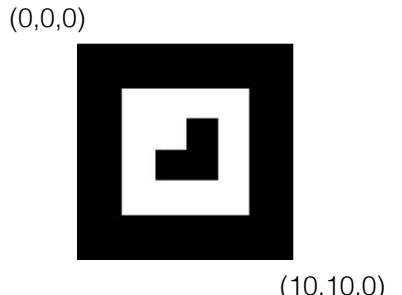
Many different ways to calibrate a camera:

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

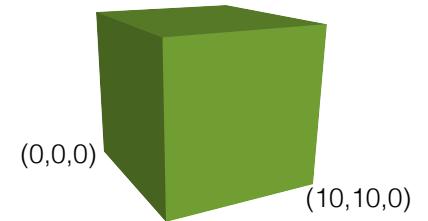
We'll briefly discuss radiometric and color calibration in later lectures. For the rest, see 15-463/663/862.



3D locations of planar marker features are known in advance



3D content prepared in advance



Simple AR program

1. Compute point correspondences (2D and AR tag)
2. Estimate the pose of the camera \mathbf{P}
3. Project 3D content to image plane using \mathbf{P}





PEPSI MAX
PRESENTS

References

Basic reading:

- Szeliski textbook, Section 2.1.5, 6.2.

Additional reading:

- Hartley and Zisserman, “Multiple View Geometry in Computer Vision,” Cambridge University Press 2004.
chapter 6 of this book has a very thorough treatment of camera models.
- Torralba and Freeman, “Accidental Pinhole and Pinspeck Cameras,” CVPR 2012.
the eponymous paper discussed in the slides.