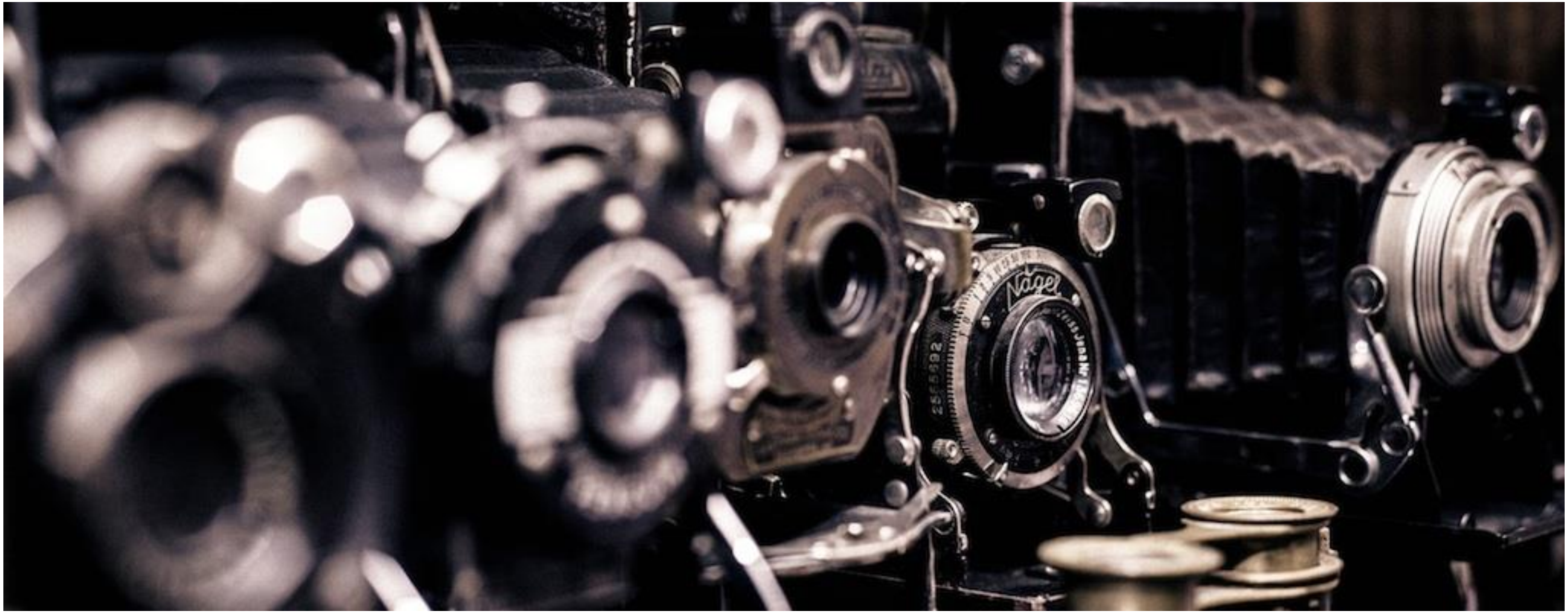
Geometric camera models



16-385 Computer Vision
Fall 2024, Lecture 9

http://16385.courses.cs.cmu.edu/

Overview of today’s lecture

- Some motivational imaging experiments.
- Pinhole camera.
- Accidental pinholes.
- Camera matrix.

Slide credits

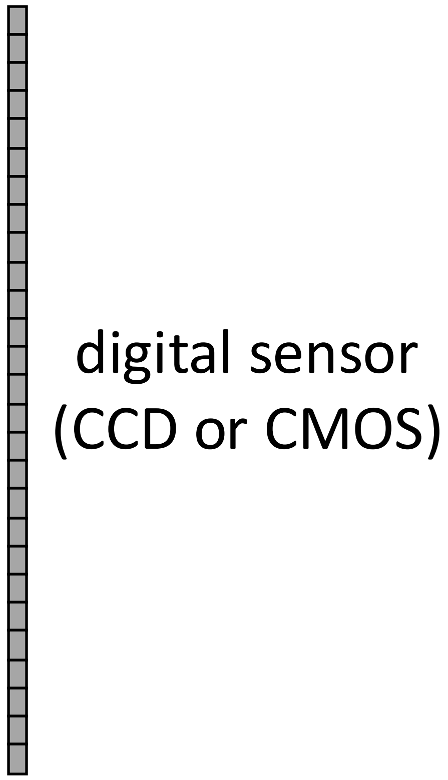
Most of these slides were adapted from:

- Matt O’Toole (16-385, Fall 2024)
- Kris Kitani (15-463, Fall 2016).

- Fredo Durand (MIT).

Some motivational imaging experiments

Let’s say we have a sensor…



… and an object we like to photograph

real-world
object

digital sensor
(CCD or CMOS)

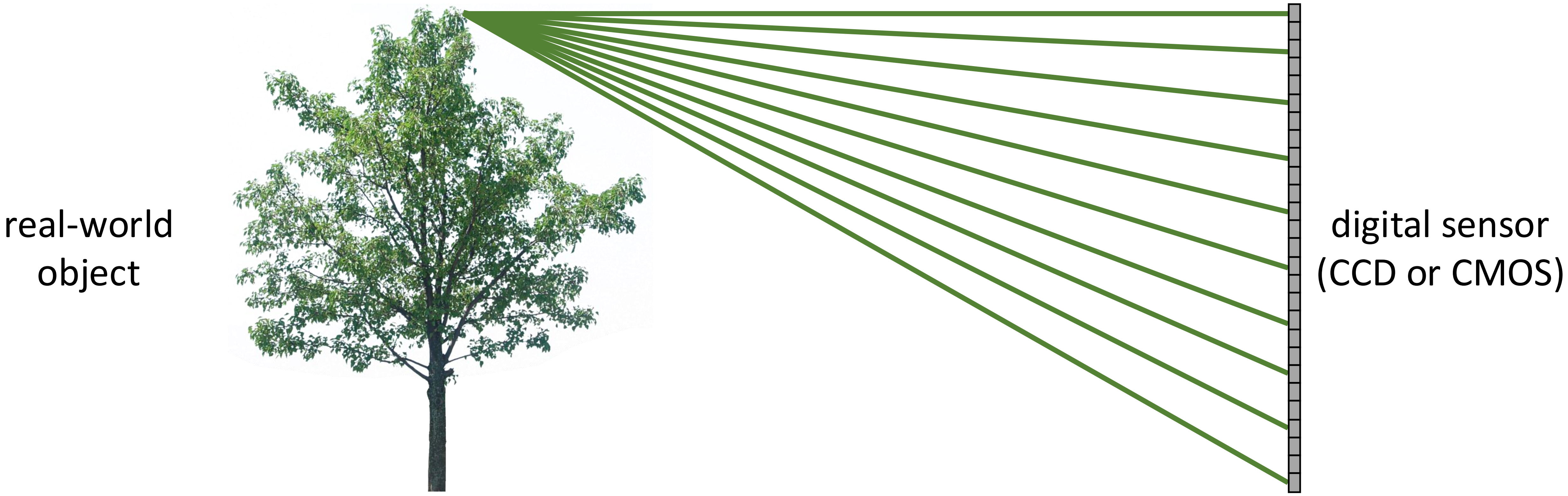
What would an image taken like this look like?

Bare-sensor imaging

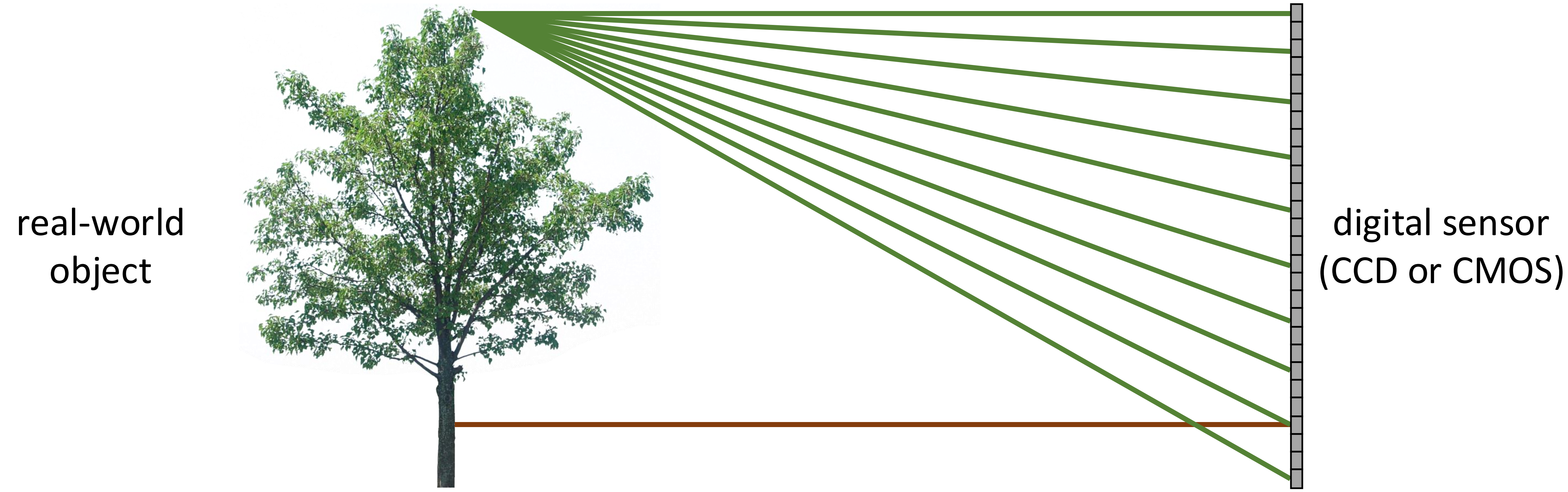
real-world object

digital sensor (CCD or CMOS)

Bare-sensor imaging



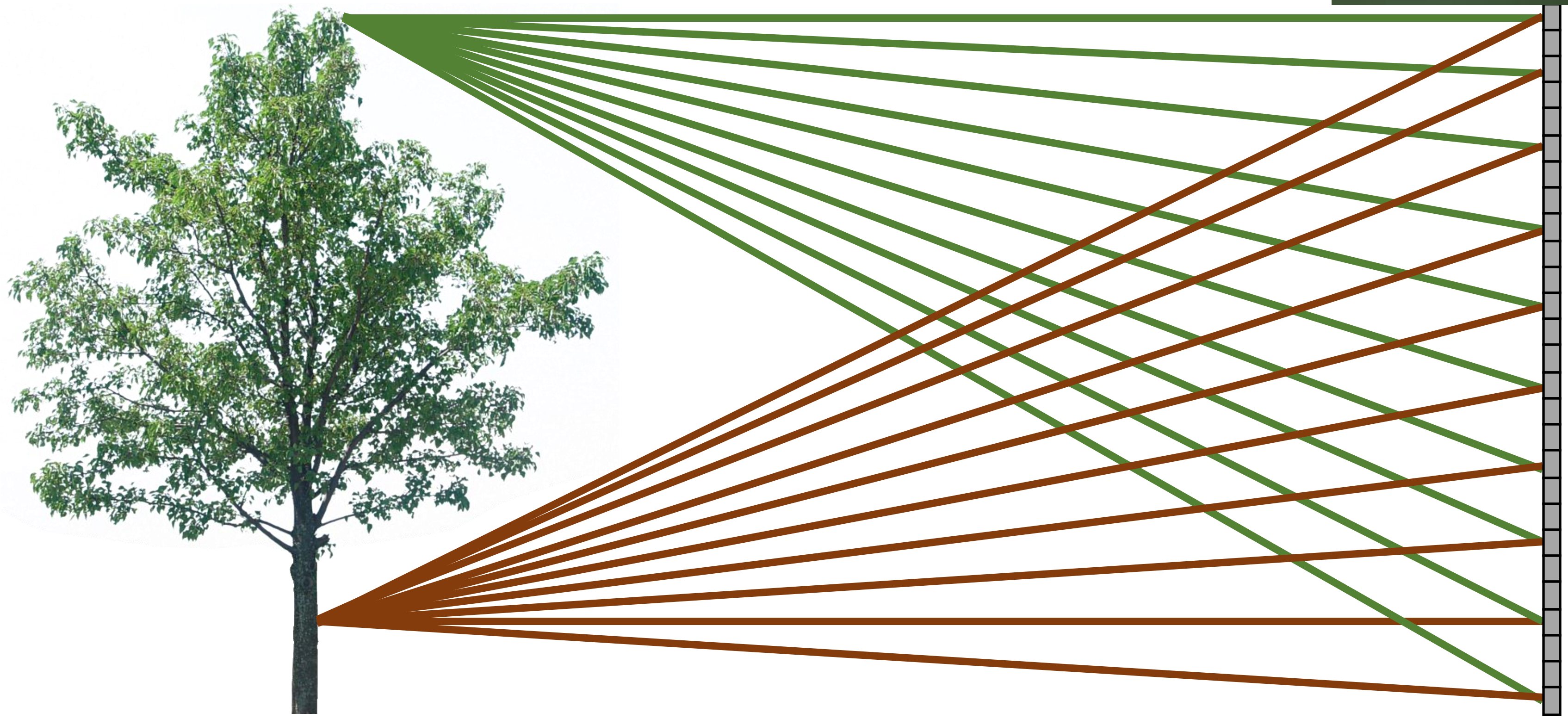
Bare-sensor imaging



Bare-sensor imaging

How do we fix this?

real-world
object

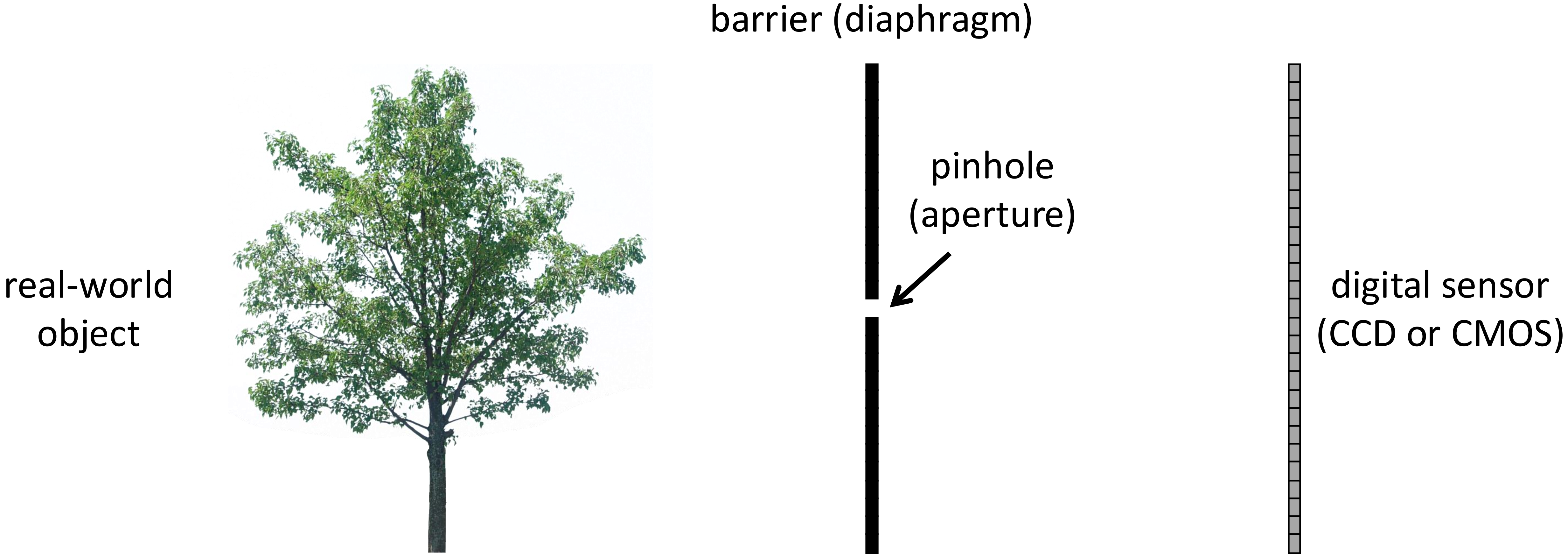


digital sensor
(CCD or CMOS)

All scene points contribute to all sensor pixels

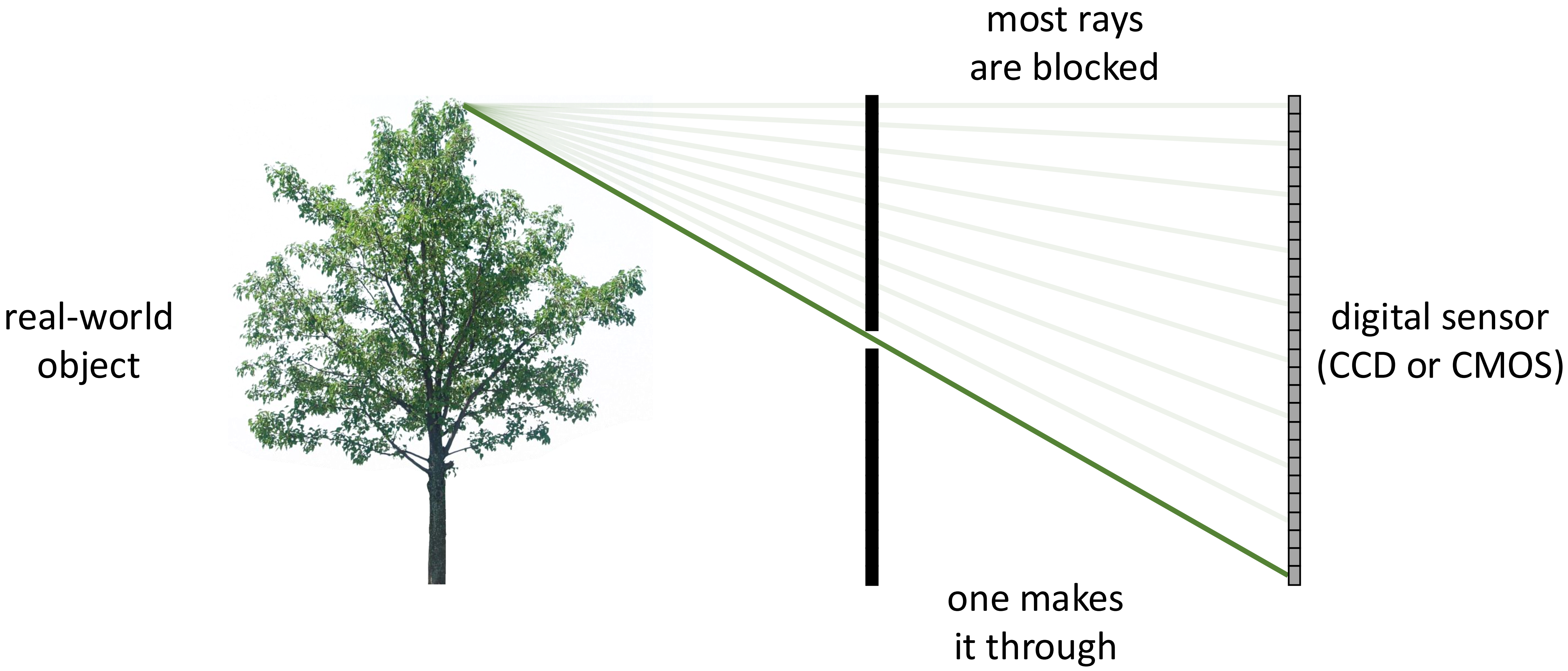
What does the
image on the
sensor look like?

Let’s add something to this scene

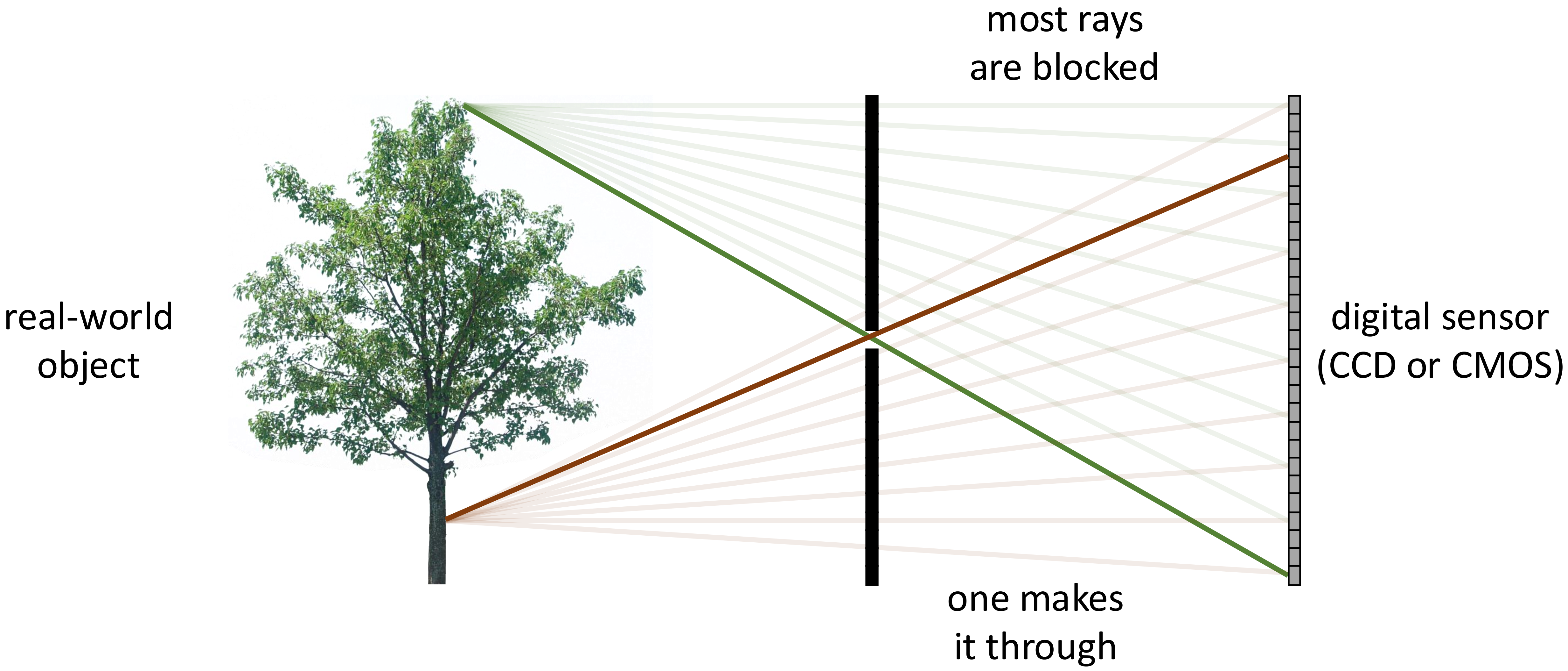


What would an image taken like this look like?

Pinhole imaging

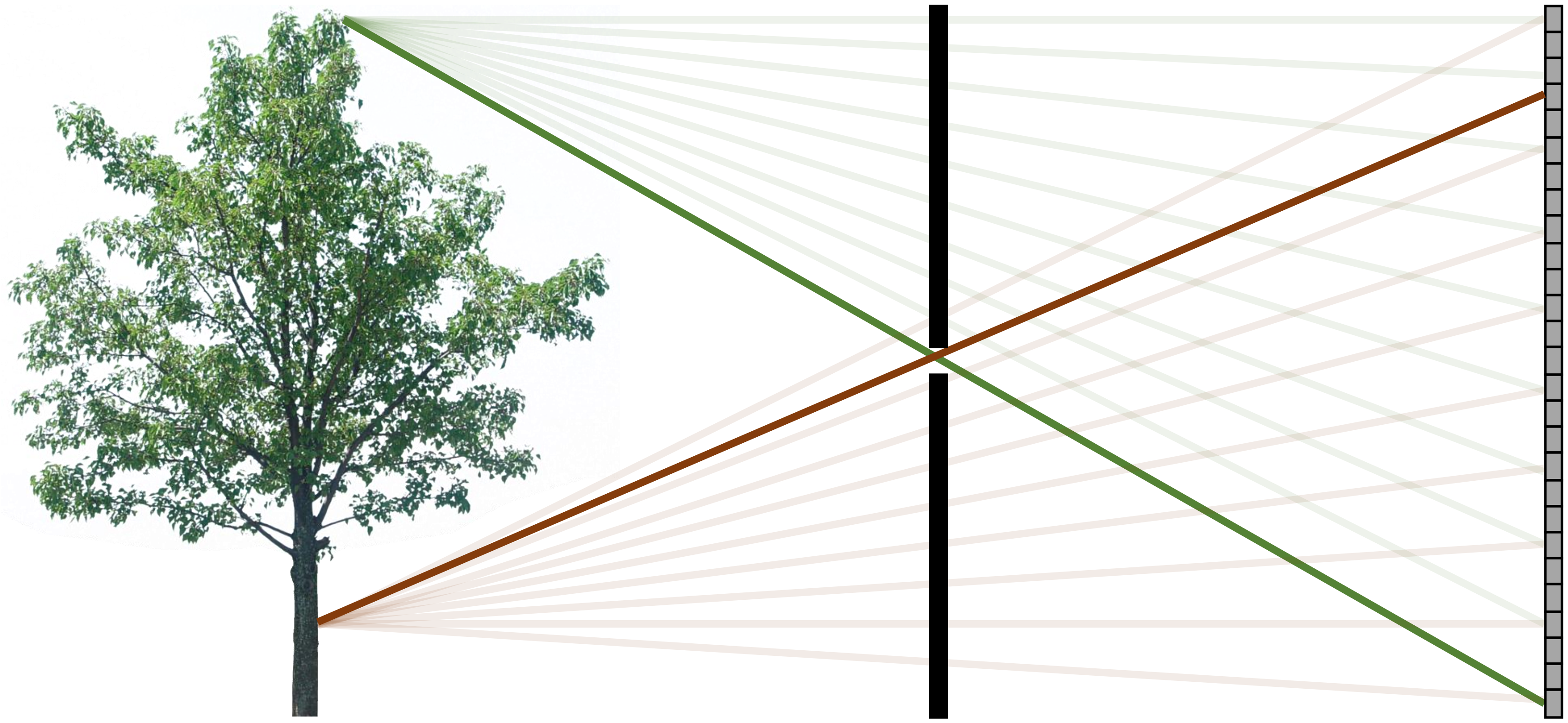


Pinhole imaging



Pinhole imaging

real-world object



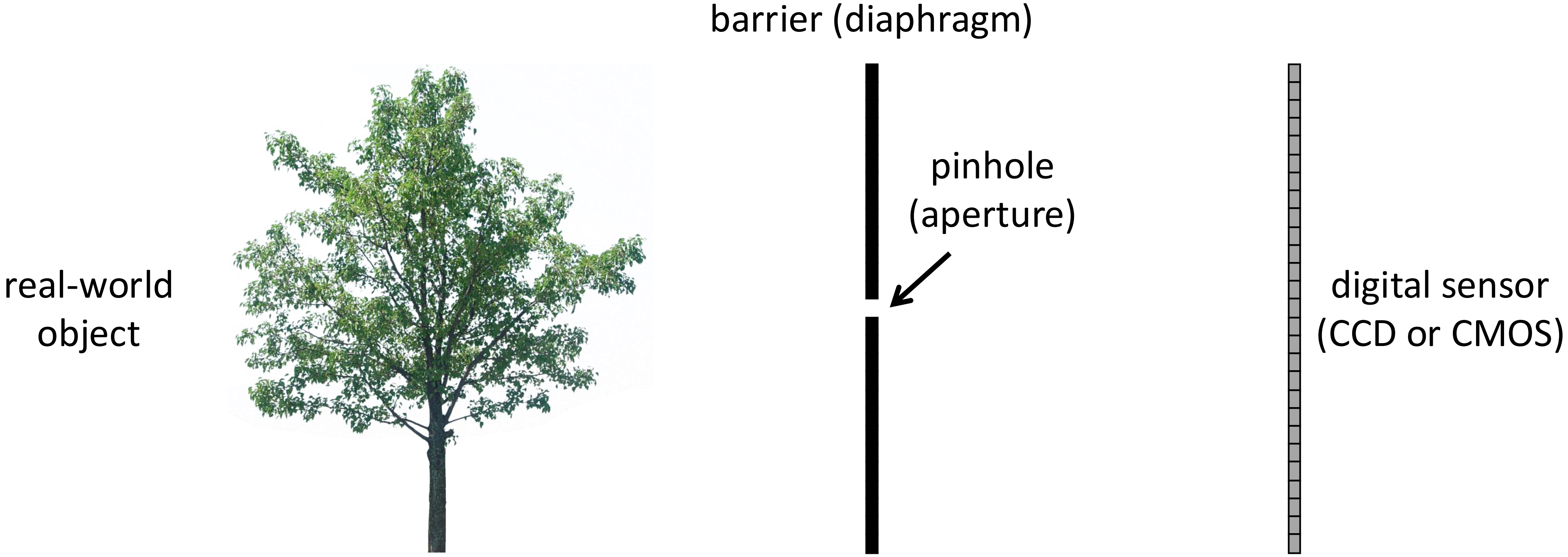
digital sensor (CCD or CMOS)

Each scene point contributes to only one sensor pixel

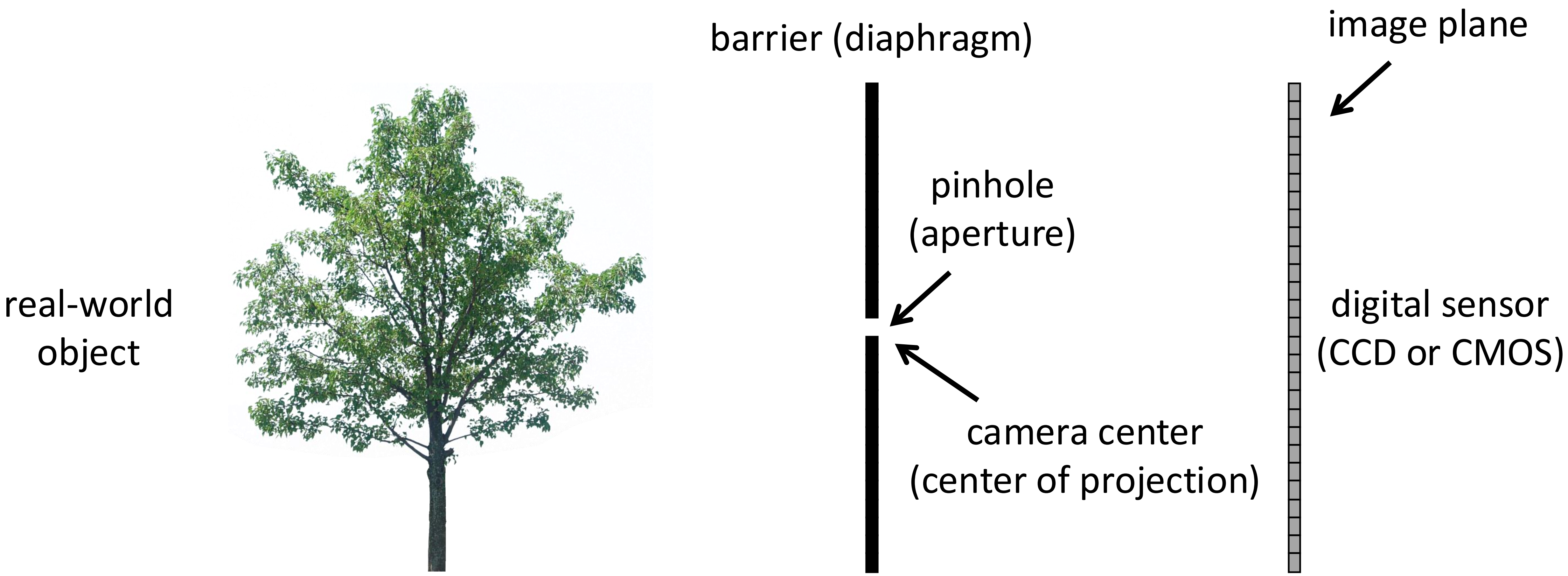
What does the image on the sensor look like?

Pinhole imaging

Pinhole camera terms

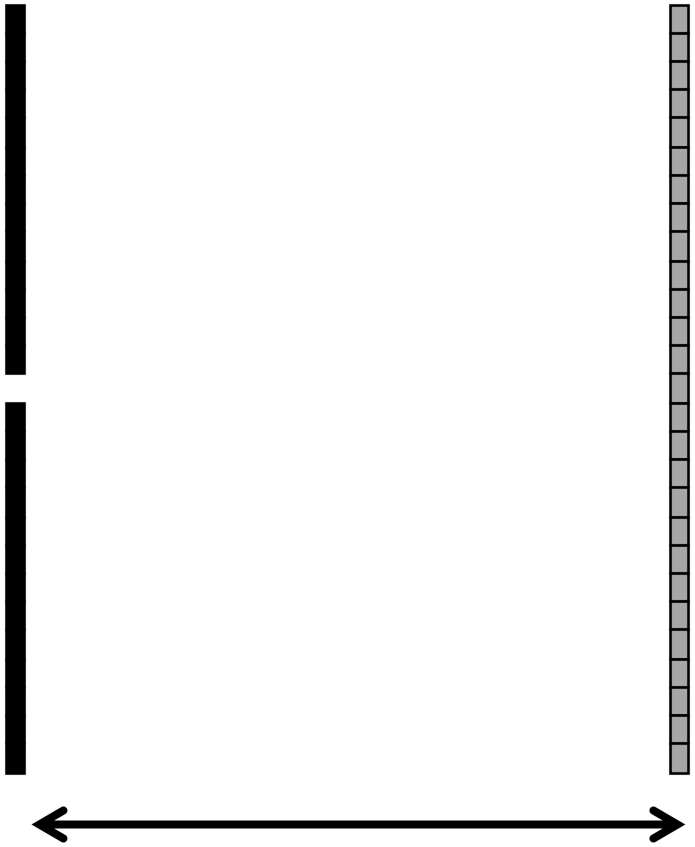


Pinhole camera terms



Focal length

real-world
object

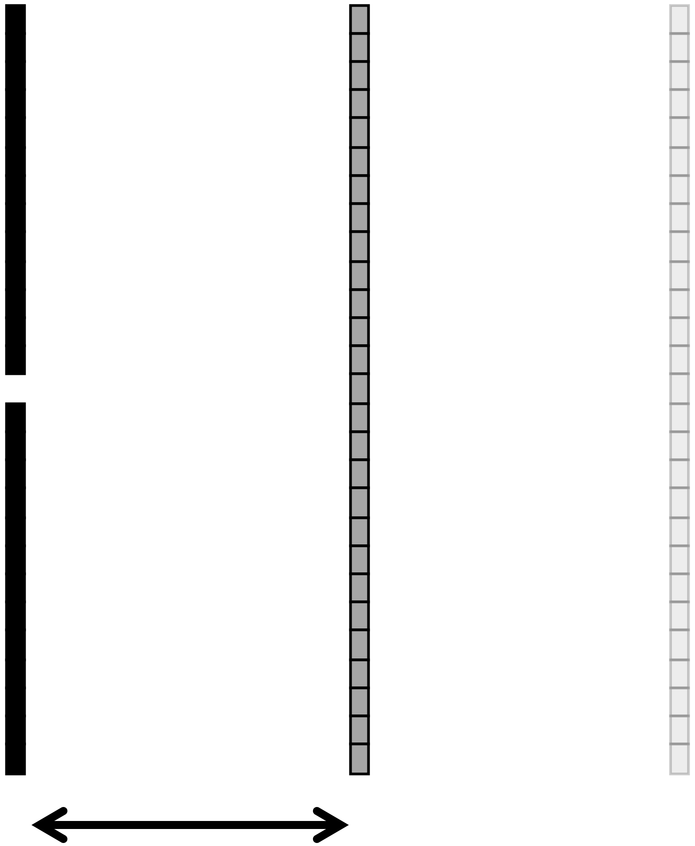


focal length f

Focal length

What happens as we change the focal length?

real-world
object

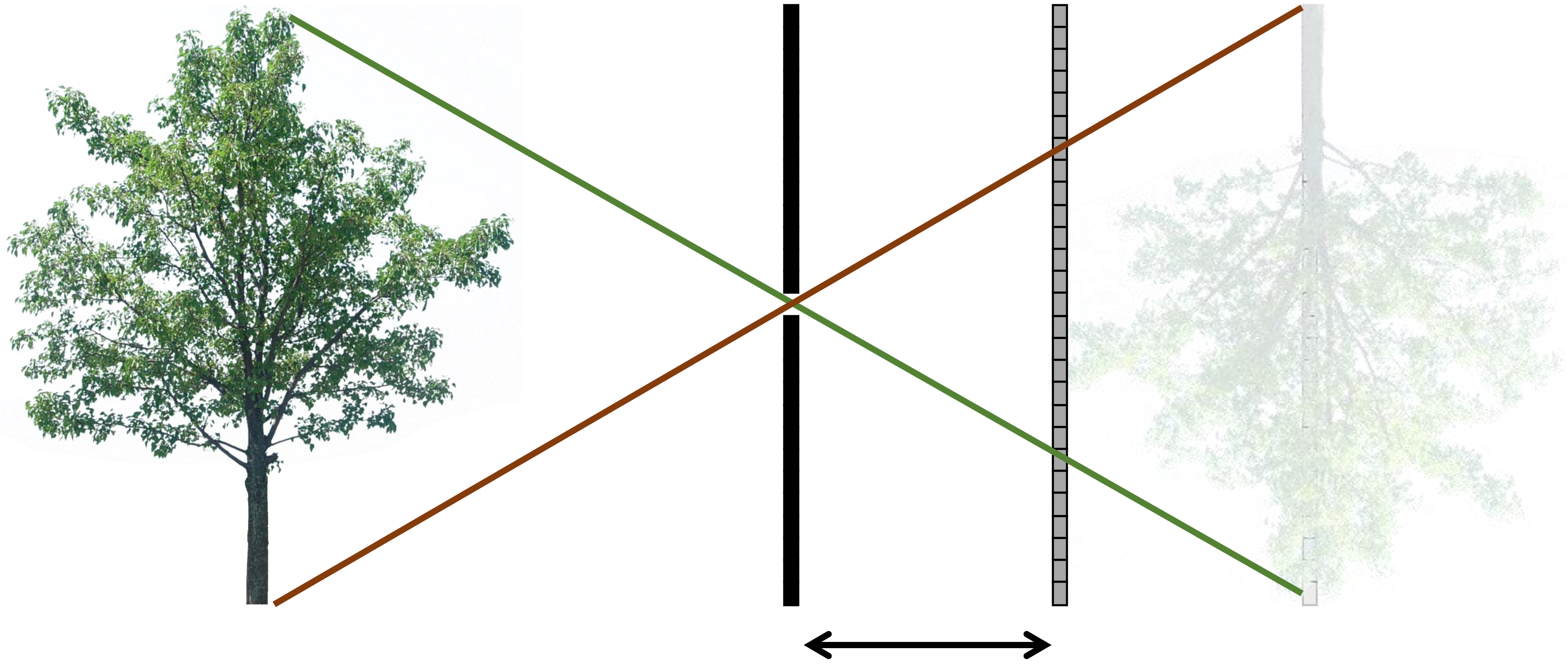


focal length 0.5 f

Focal length

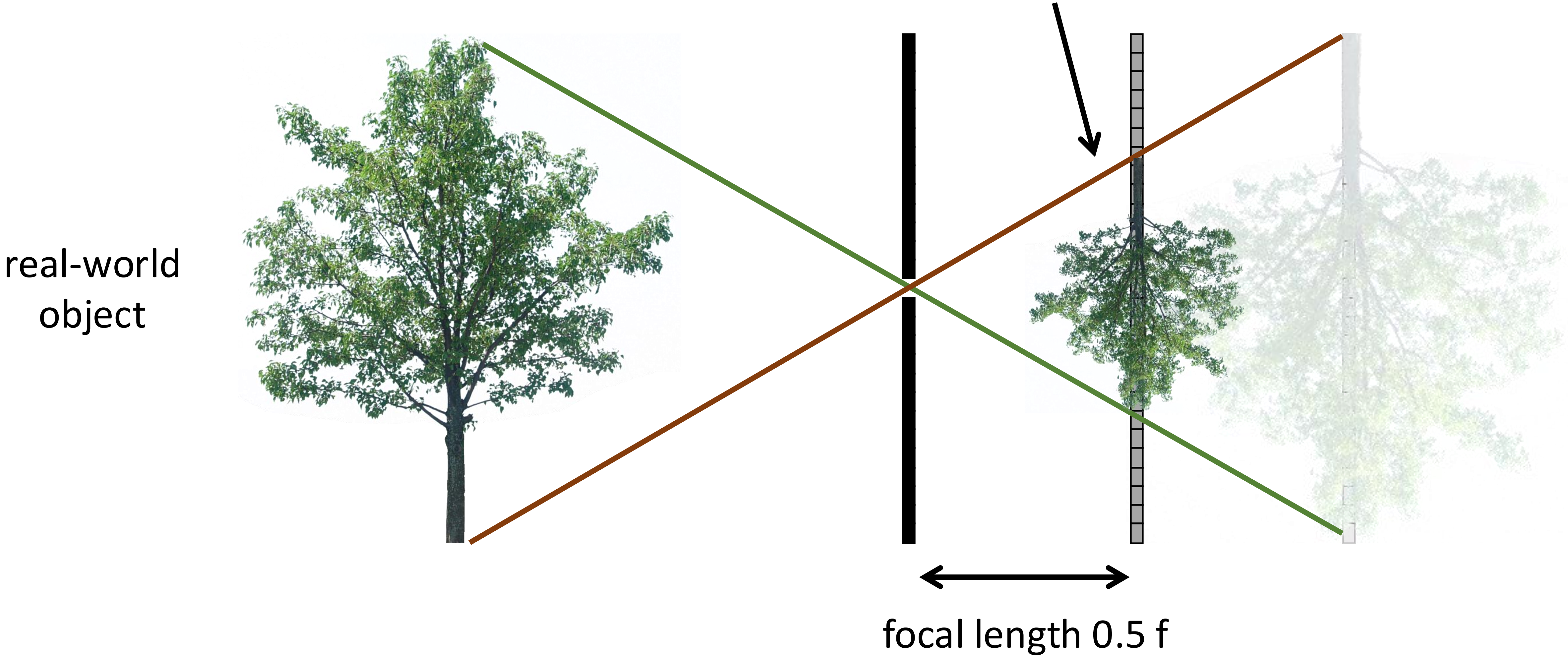
What happens as we change the focal length?

real-world
object



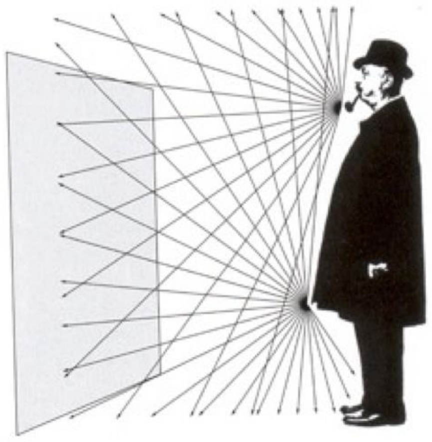
Focal length

What happens as we change the focal length?

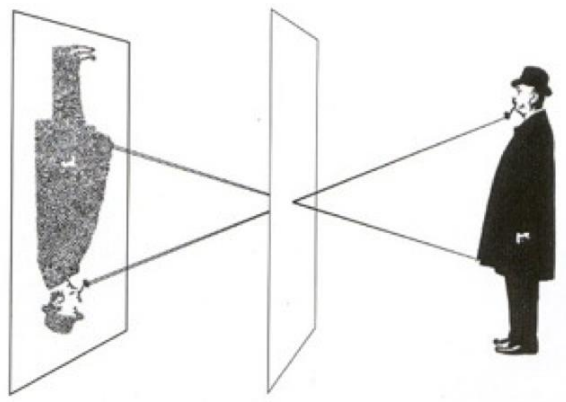


What happens as we change the size of the pinhole?

Pinhole optics



With a large pinhole, images will be blurry

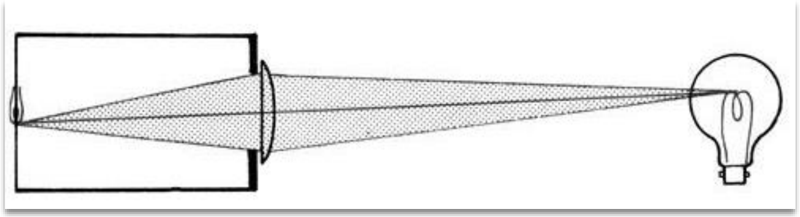


- With a very small pinhole, not much light will get through
- We will need to keep the pinhole open a long time to get enough light for the image (camera exposure)

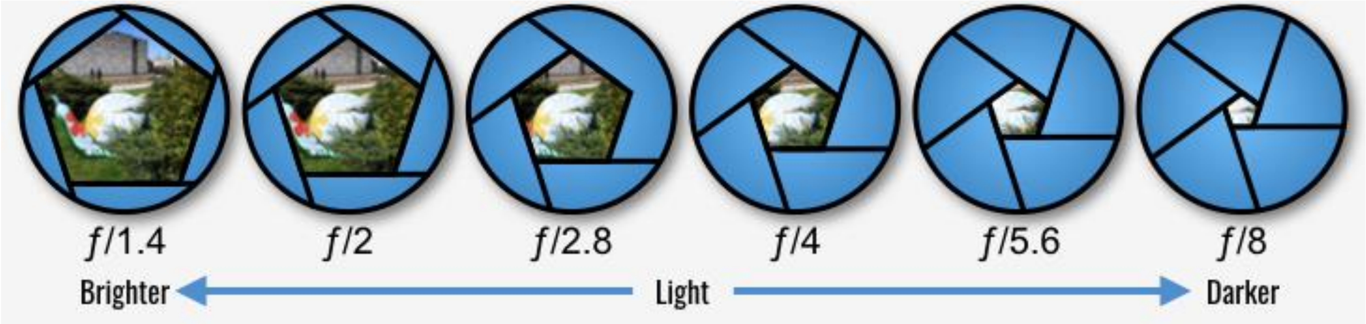
How do we get a sharp image with a small exposure time?

Depth of Field

Use a lens to grab more photons
Allows for a larger pinhole; lens focuses the light
from a particular depth



Most cameras can adjust the size of the pinhole ("aperture"):

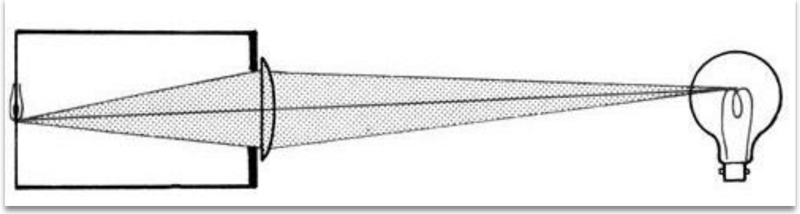


But why would we ever want a smaller pinhole if we can just use a larger pinhole and a lens?

Cons of a larger pinhole:
- Depth of field
- Radial Distortion

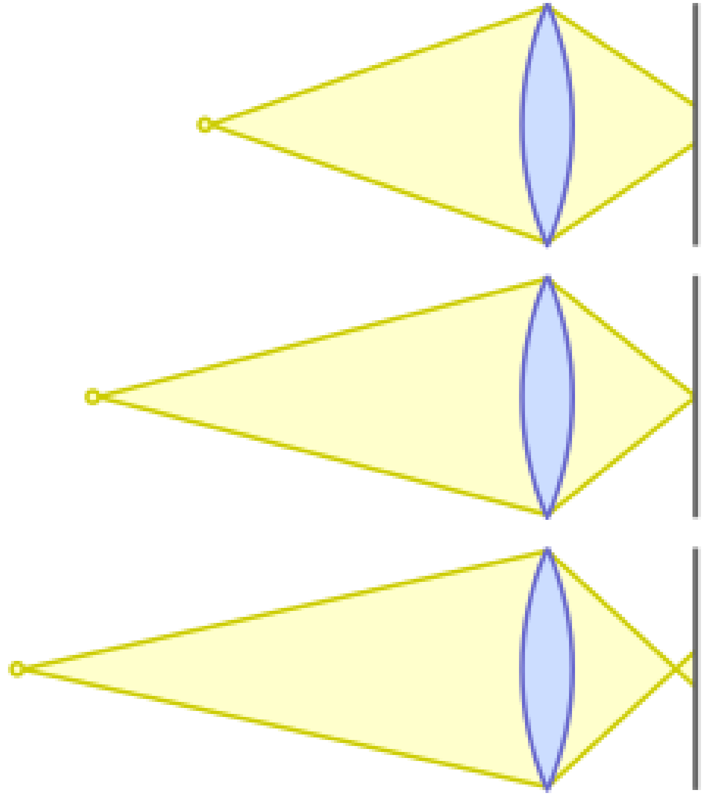
Depth of Field

Use a lens to grab more photons
Allows for a larger pinhole; lens focuses the light
from a particular depth



With a lens, objects outside of a particular depth will be blurred:

World point Lens Image plane

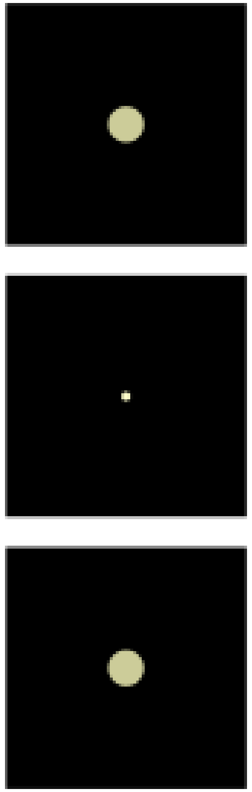


Focus at a
"circle of confusion"

Focus at a
single point

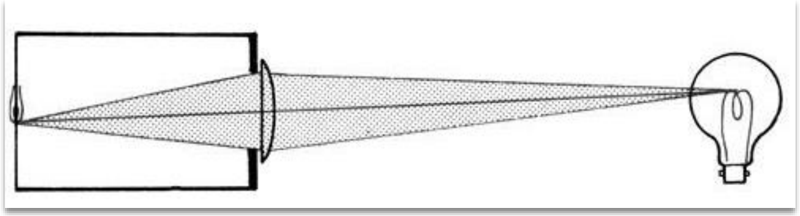
Focus at a
"circle of confusion"

Camera image



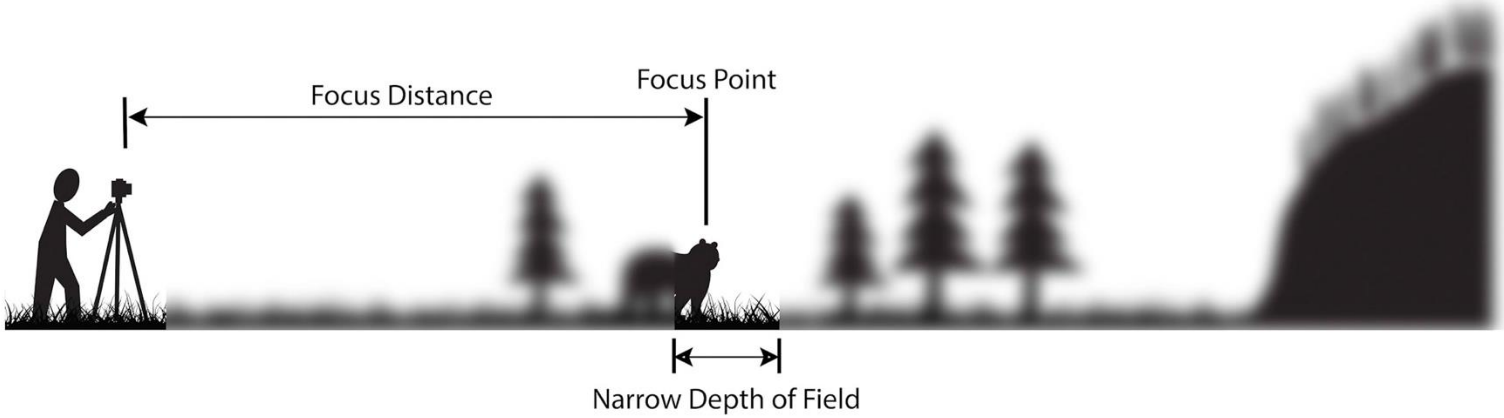
Depth of Field

Use a lens to grab more photons
Allows for a larger pinhole; lens focuses the light
from a particular depth



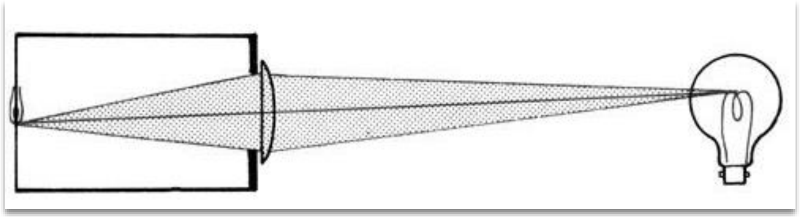
Objects outside the particular depth will be blurred (limited “depth-of-field”)

Larger
aperture

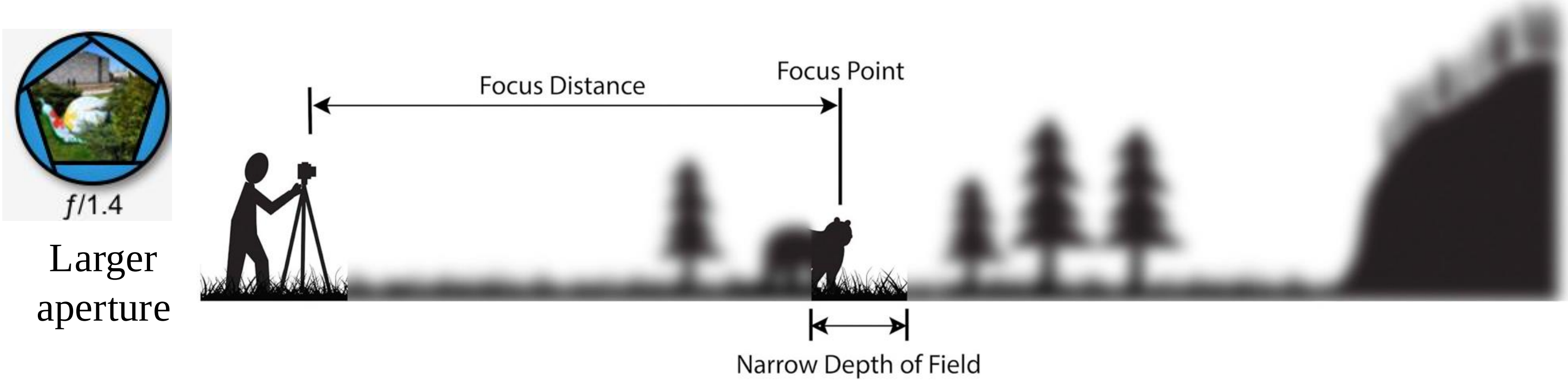


Depth of Field

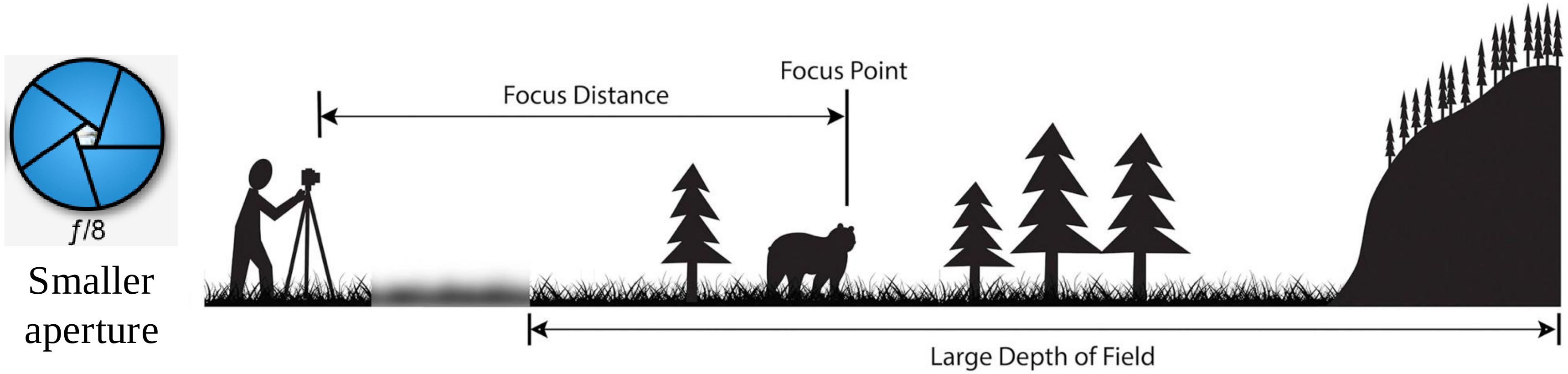
Use a lens to grab more photons
Allows for a larger pinhole; lens focuses the light
from a particular depth



Objects outside the particular depth will be blurred (limited "depth-of-field")



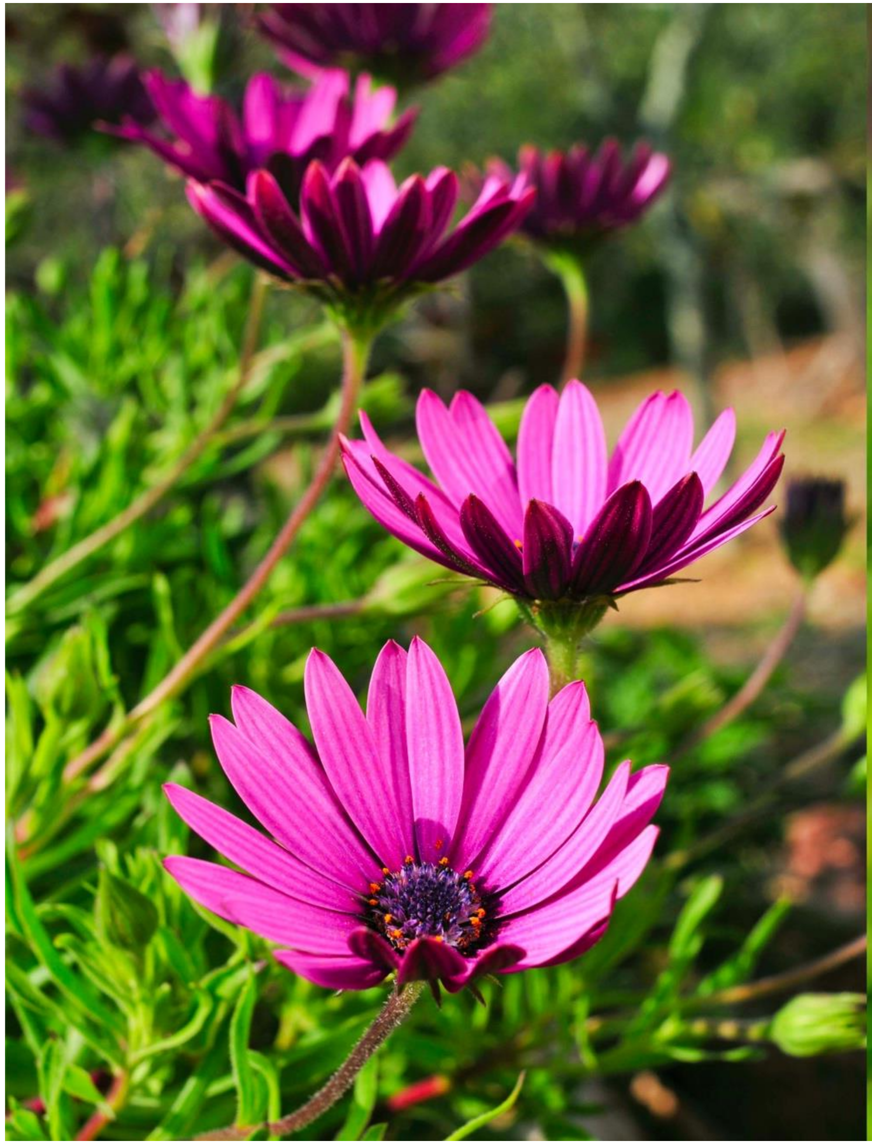
Larger
aperture



Smaller
aperture

Small Aperture
Large Depth of Field

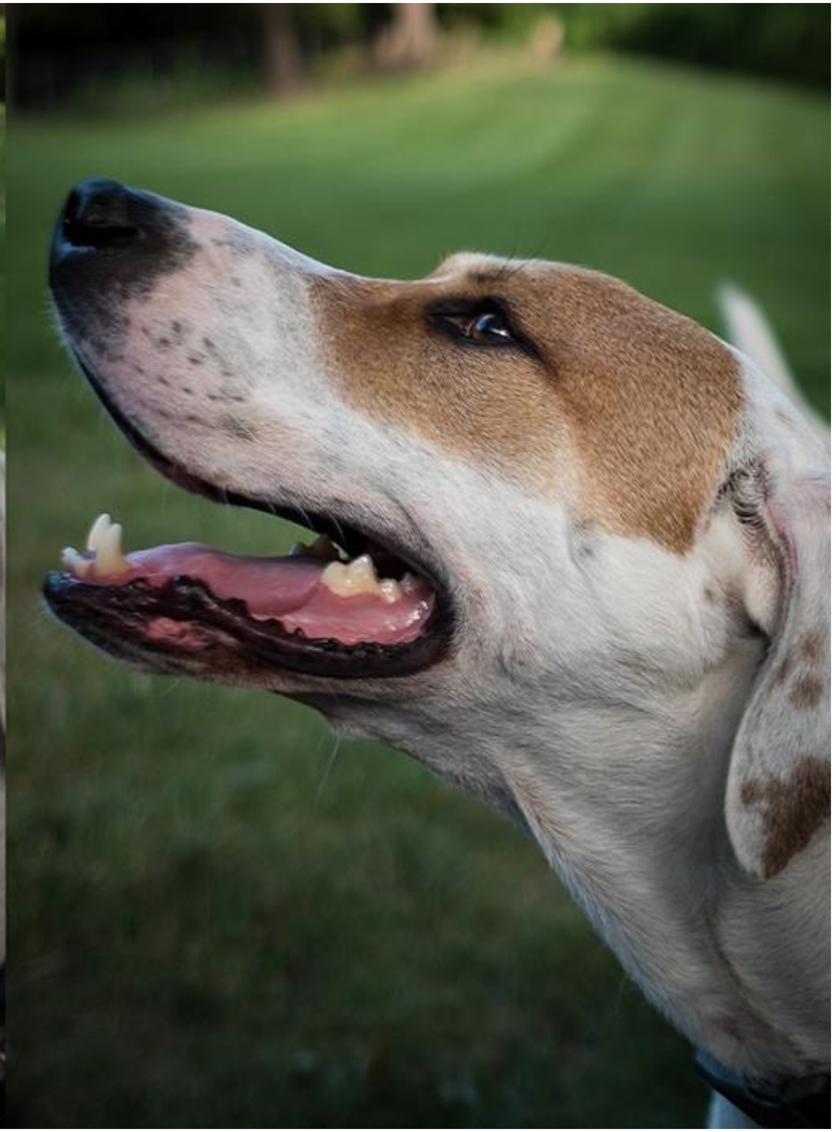
Large Aperture
Narrow Depth of Field



Small Aperture
Large Depth of Field

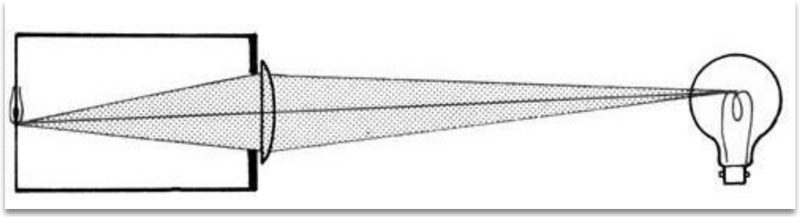
Large Aperture
Narrow Depth of Field

Small Aperture
Large Depth of Field



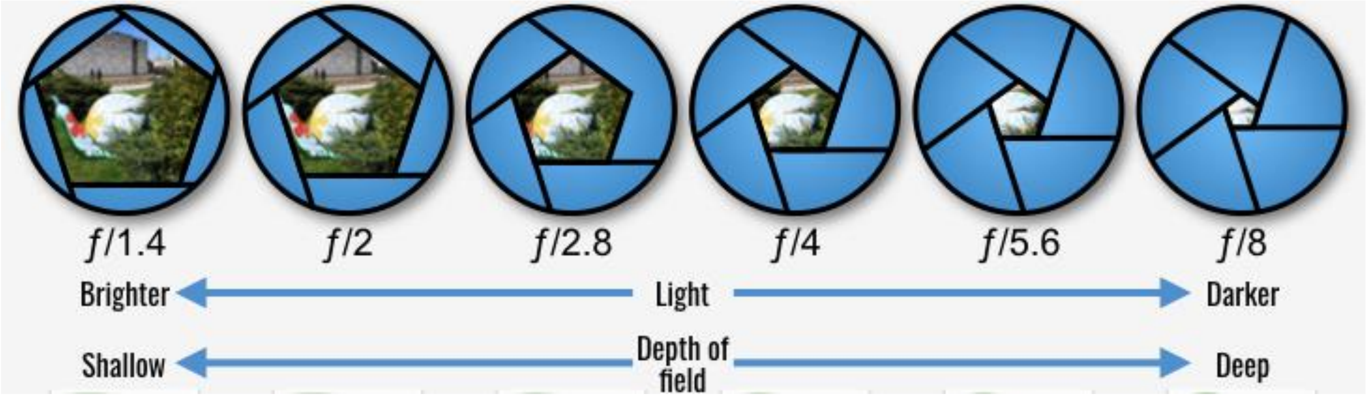
Large Aperture
Narrow Depth of Field

Depth of Field



Use a lens to grab more photons
Allows for a larger pinhole; lens focuses the light
from a particular depth

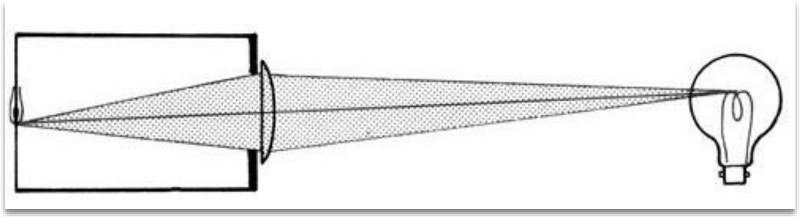
Larger aperture -> larger depth of field



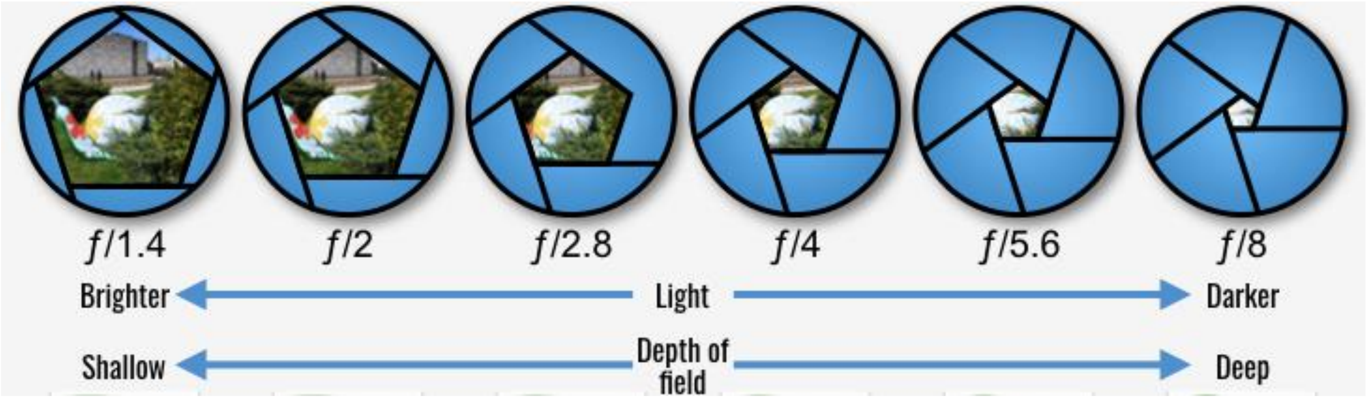
Which aperture is best for computer vision?

Depth of Field

Use a lens to grab more photons
Allows for a larger pinhole; lens focuses the light
from a particular depth



Larger aperture -> larger depth of field



Less in focus -> Bad for computer vision

Brightest -> Good for computer vision

More in focus -> Good for computer vision

Darkest -> Bad for computer vision (Low Signal-to-Noise Ratio)

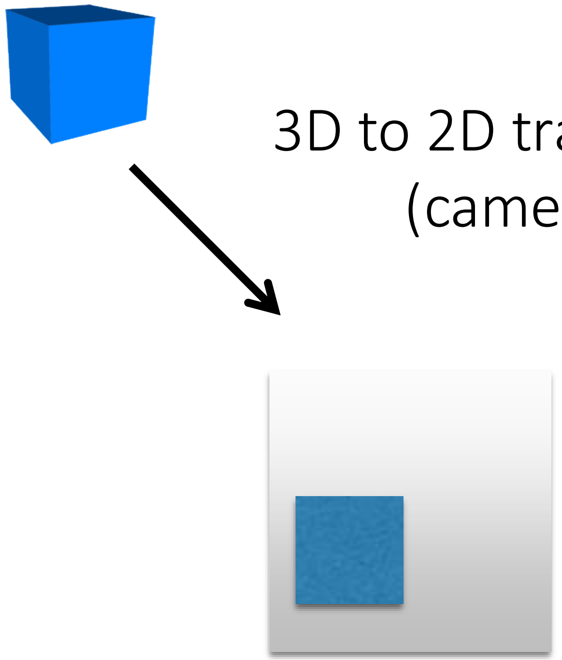
Camera Matrix

A camera is a mapping from:

the 3D world

to:

a 2D image



The camera as a coordinate transformation

$$x = KX$$

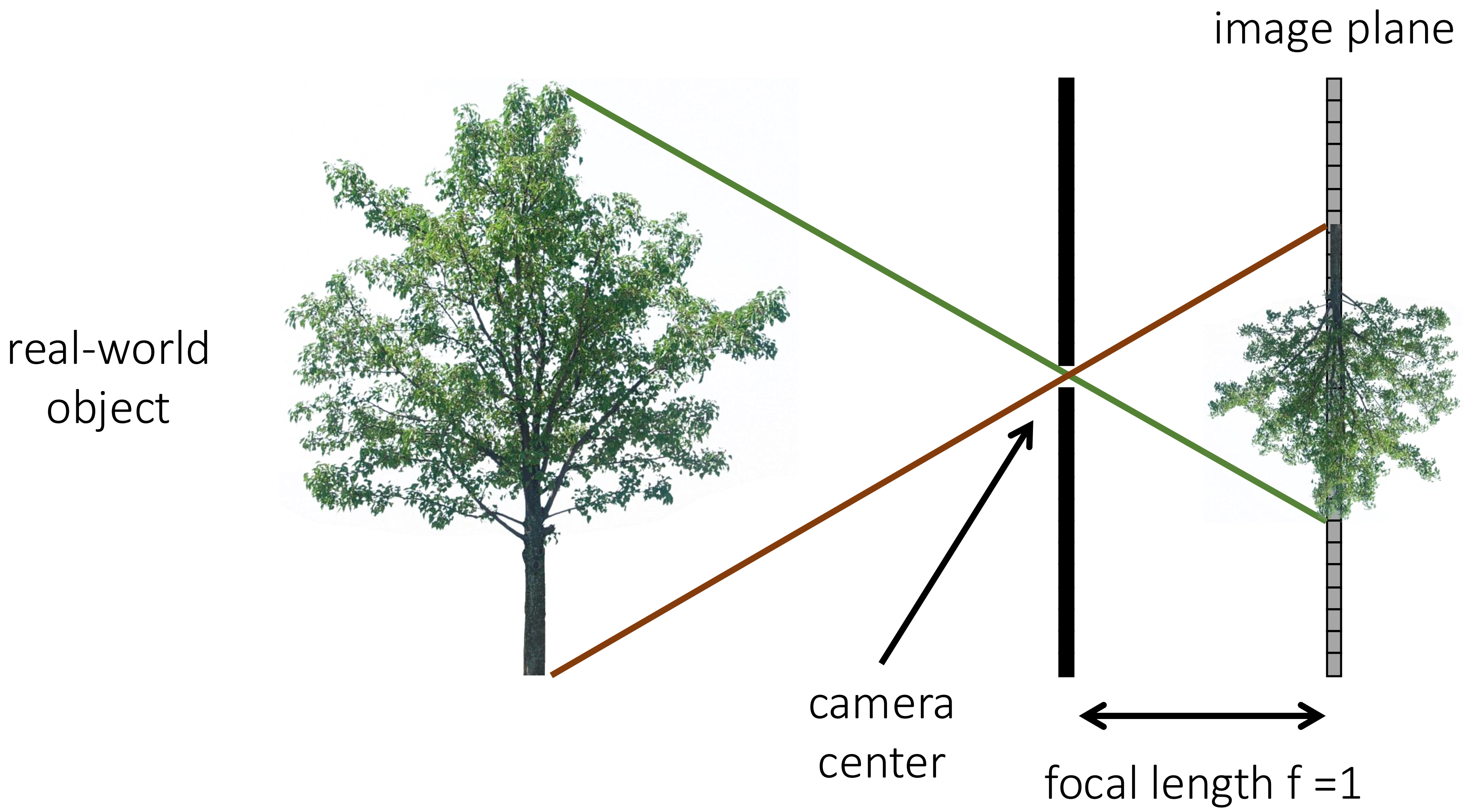
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

homogeneous
image coordinates
3 x 1

camera
matrix
3 x 3

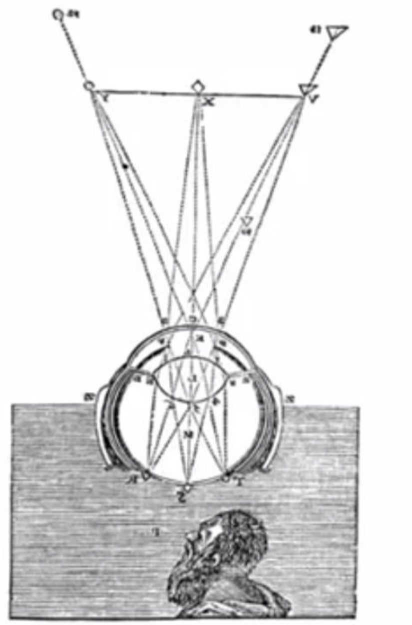
world coordinates
3 x 1

The pinhole camera



Annoying detail: image inversion

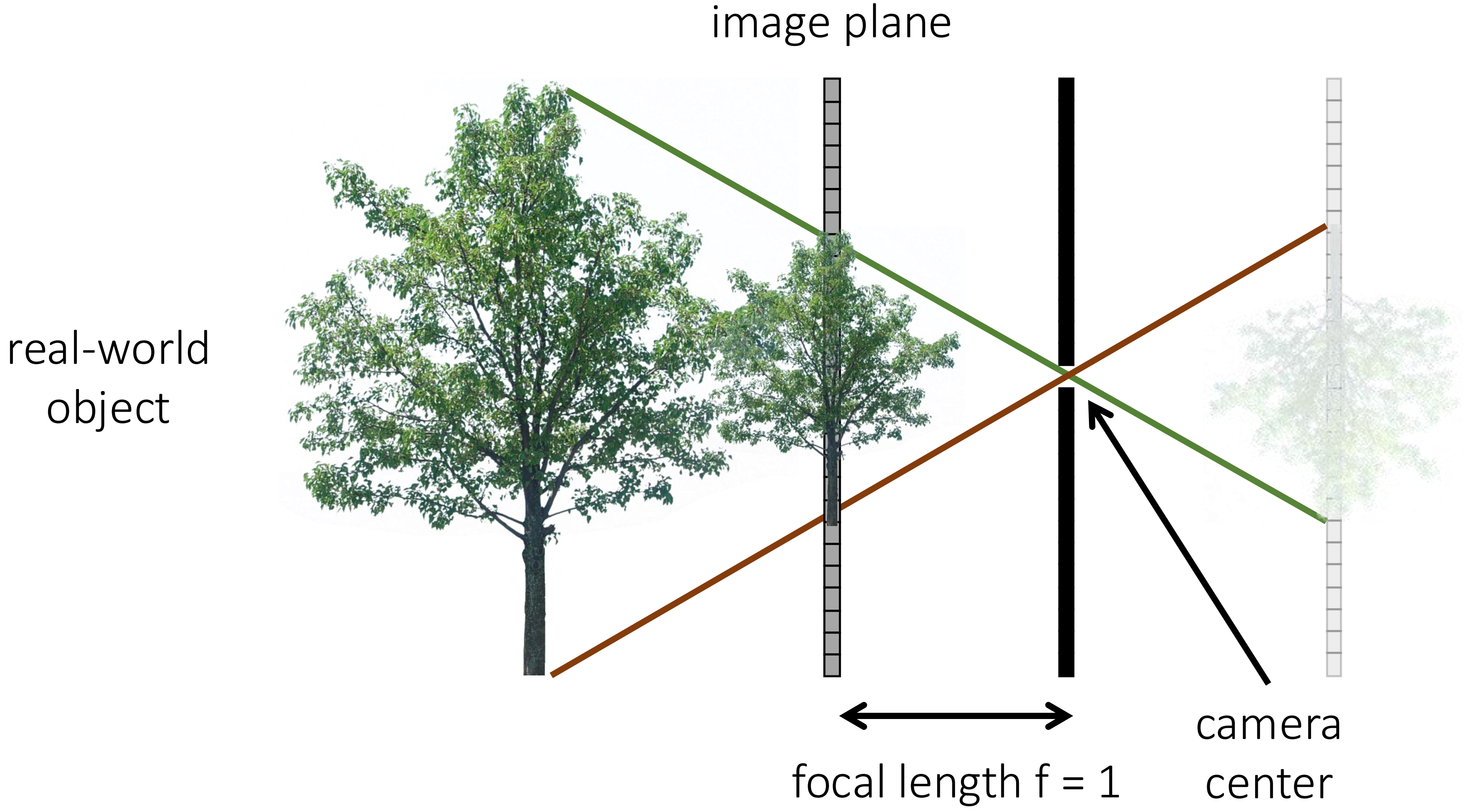
Why don't we see an upside-down world?



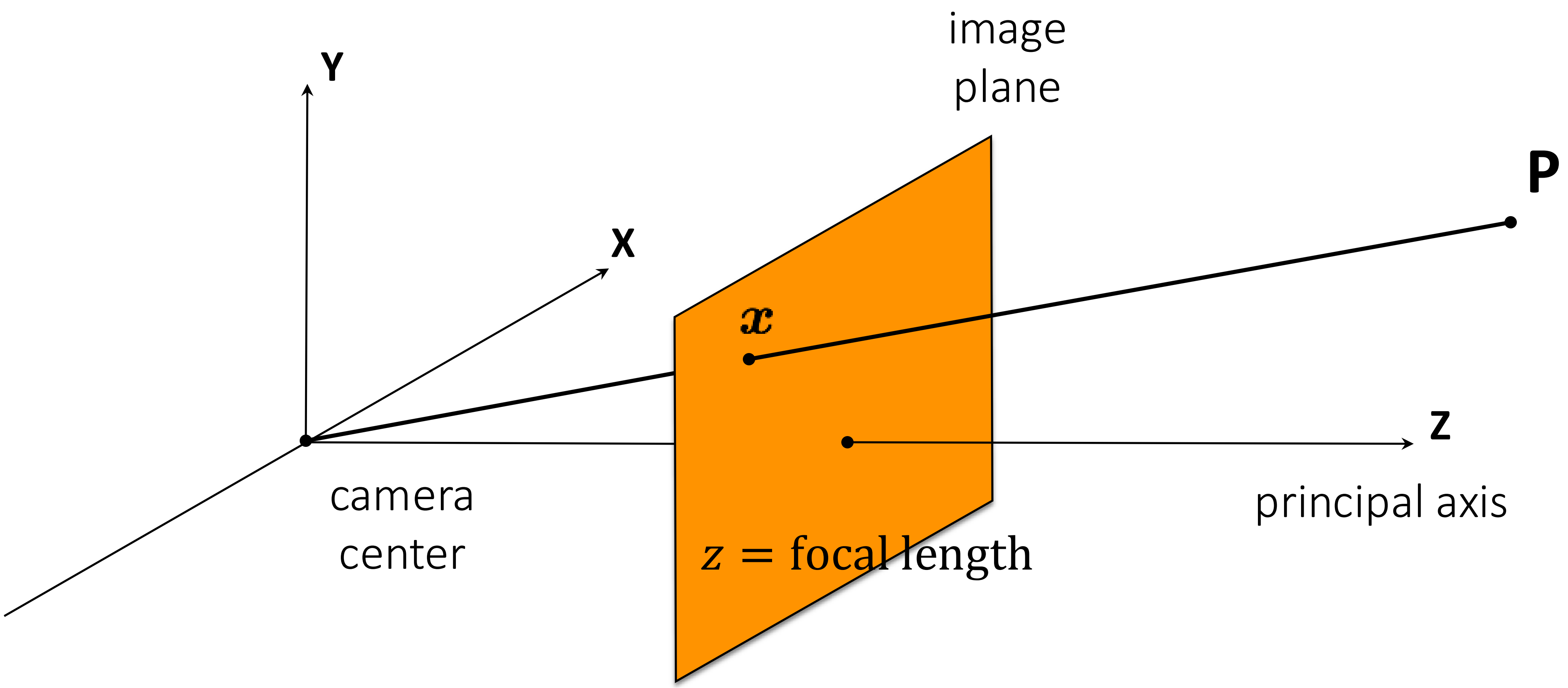
From Descartes(1937), La Dioptrique

This question perplexed folks for a while.
But software / your brain can simply invert the image.

The (rearranged) pinhole camera

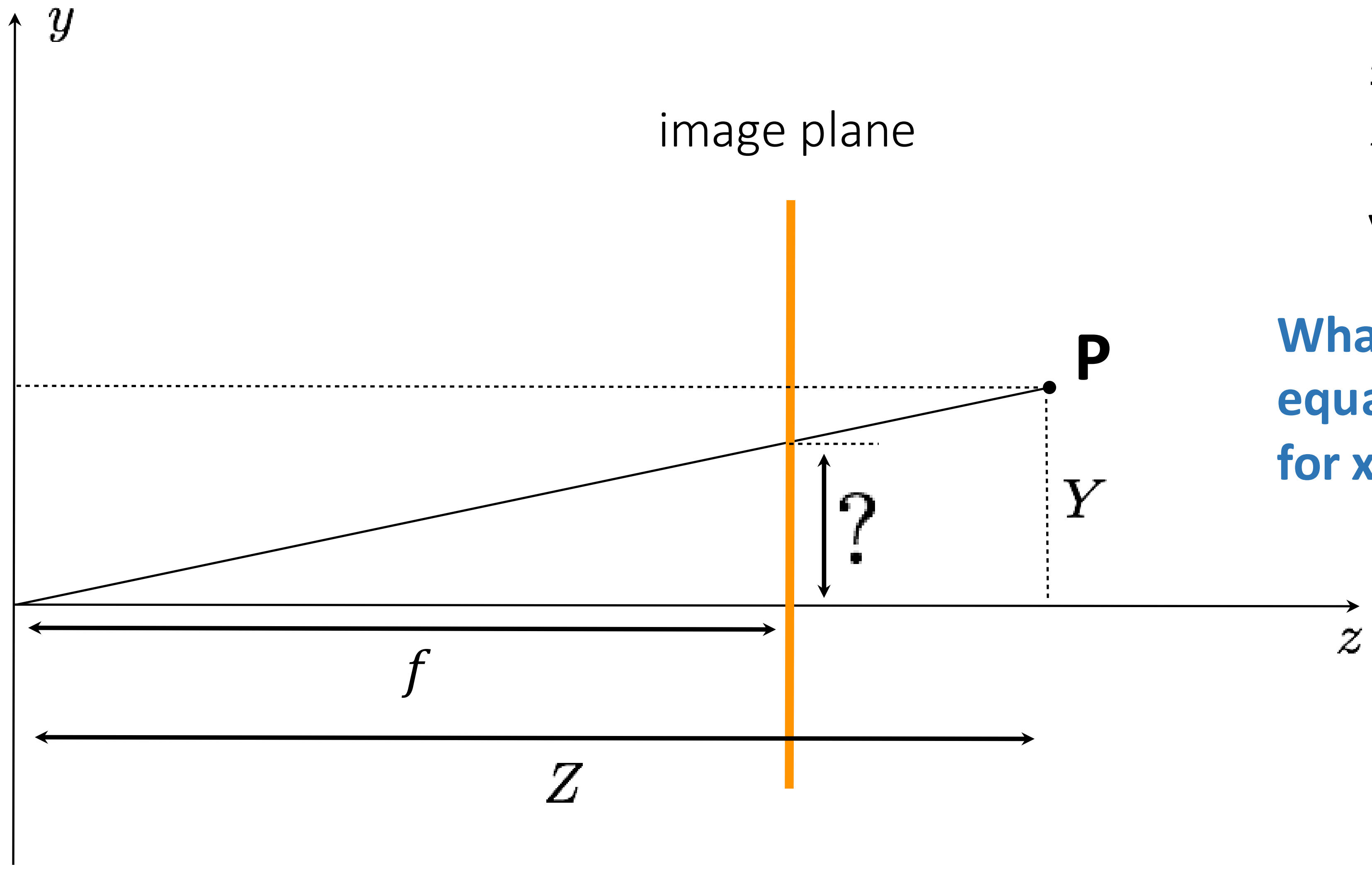


The (rearranged) pinhole camera



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

The 2D view of the (rearranged) pinhole camera



$$\frac{y}{f} = \frac{Y}{Z}$$

$$\text{y} = f\,\frac{Y}{Z}$$

What would the equation look like for x?

$$\text{x} = f\,\frac{X}{Z}$$

What is the equation for image coordinate y in terms of f, Z, Y?

The pinhole camera matrix for arbitrary focal length

Relationship from similar triangles:

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

General camera model *in homogeneous coordinates*:

$$\begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

What does the pinhole camera projection look like?

The pinhole camera matrix for arbitrary focal length

Relationship from similar triangles:

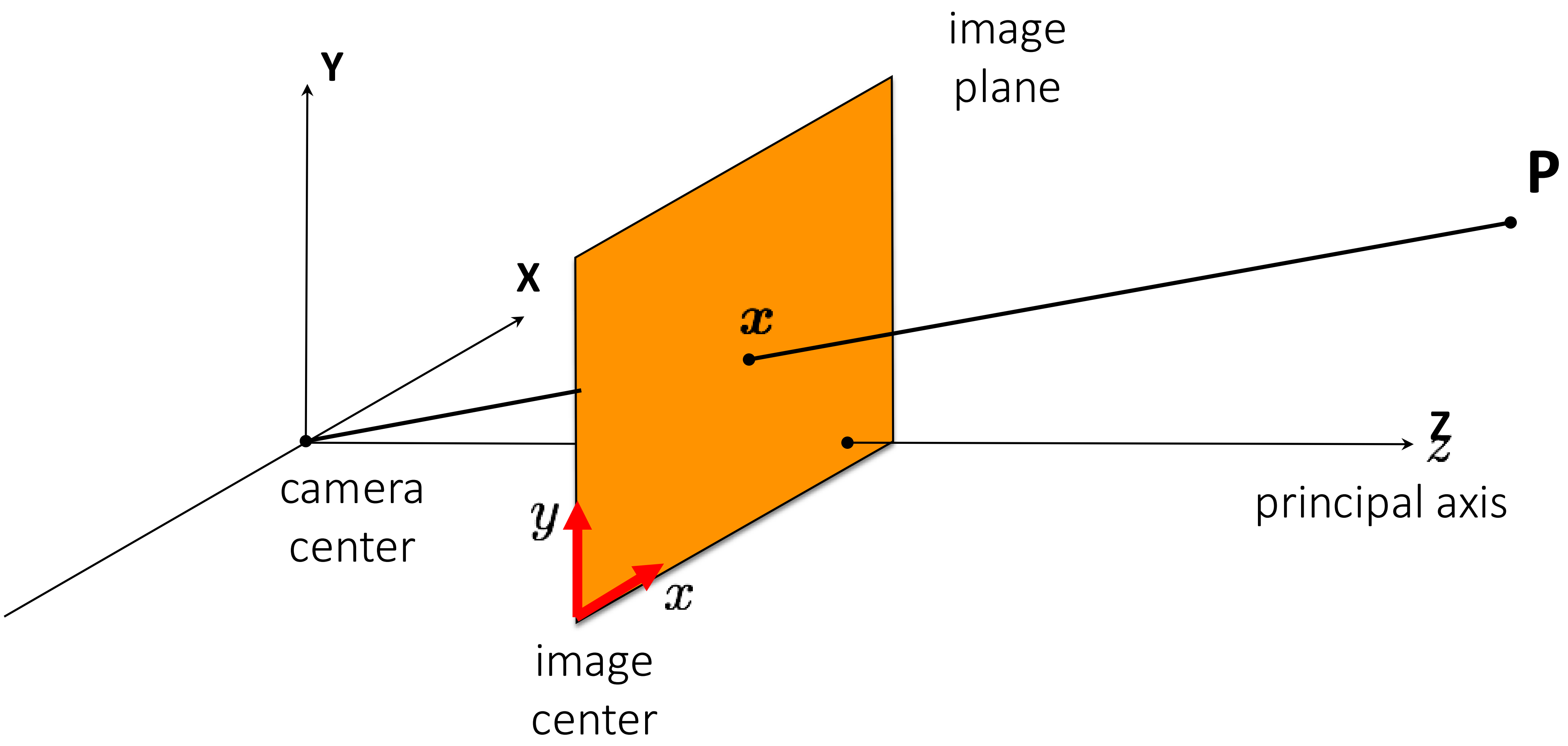
$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

General camera model *in homogeneous coordinates*:

$$\begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

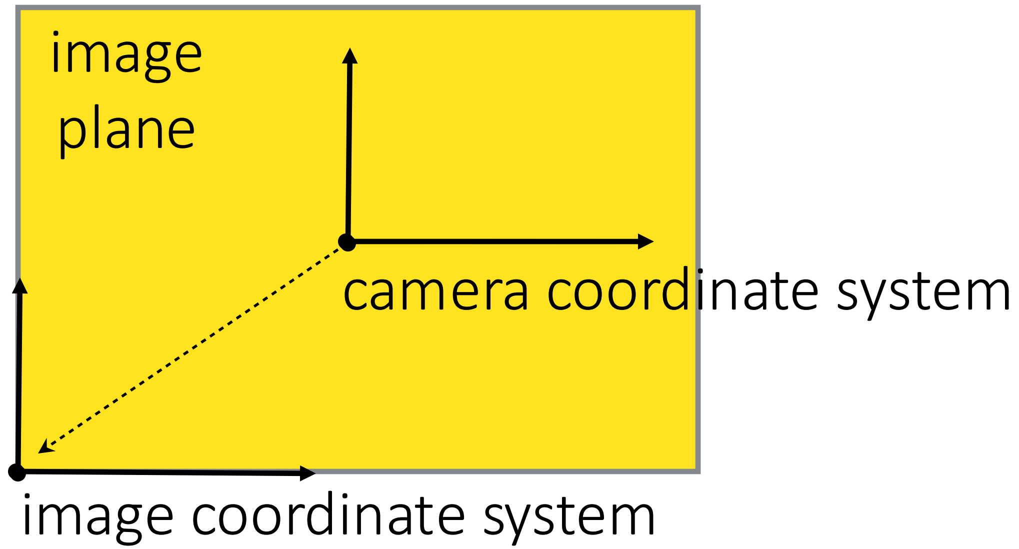
Generalizing the camera matrix

In general, the camera and image have *different* centers:



Generalizing the camera matrix

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

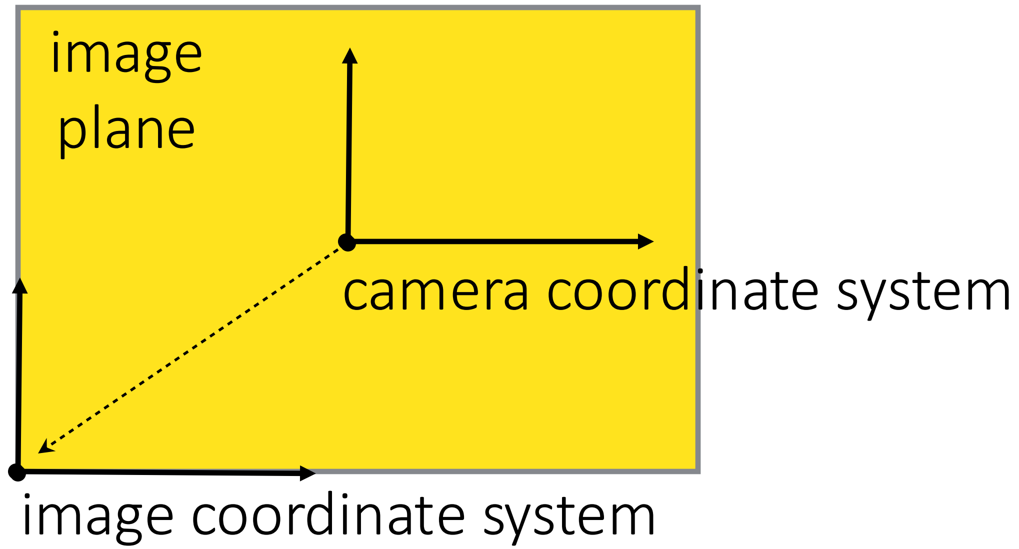


How does the camera matrix change?

$$\begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Generalizing the camera matrix

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



How does the camera matrix change?

shift vector transforming camera origin to image origin

$$\begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Fancier intrinsics

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

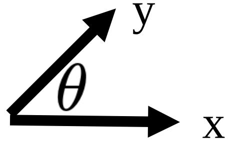
Camera <u>instrinsic</u> matrix K

Now let’s capture more details about how cameras work:

$$\left.\begin{aligned} x_s &= s_x x \\ y_s &= s_y y \end{aligned}\right\} \text{non-square pixels}$$

$$\left.\begin{aligned} x' &= x_s + o_x \\ y' &= y_s + o_y \end{aligned}\right\} \text{shifted origin}$$

$$x\text{”} = x' + s_\theta y'$$



skewed image axes

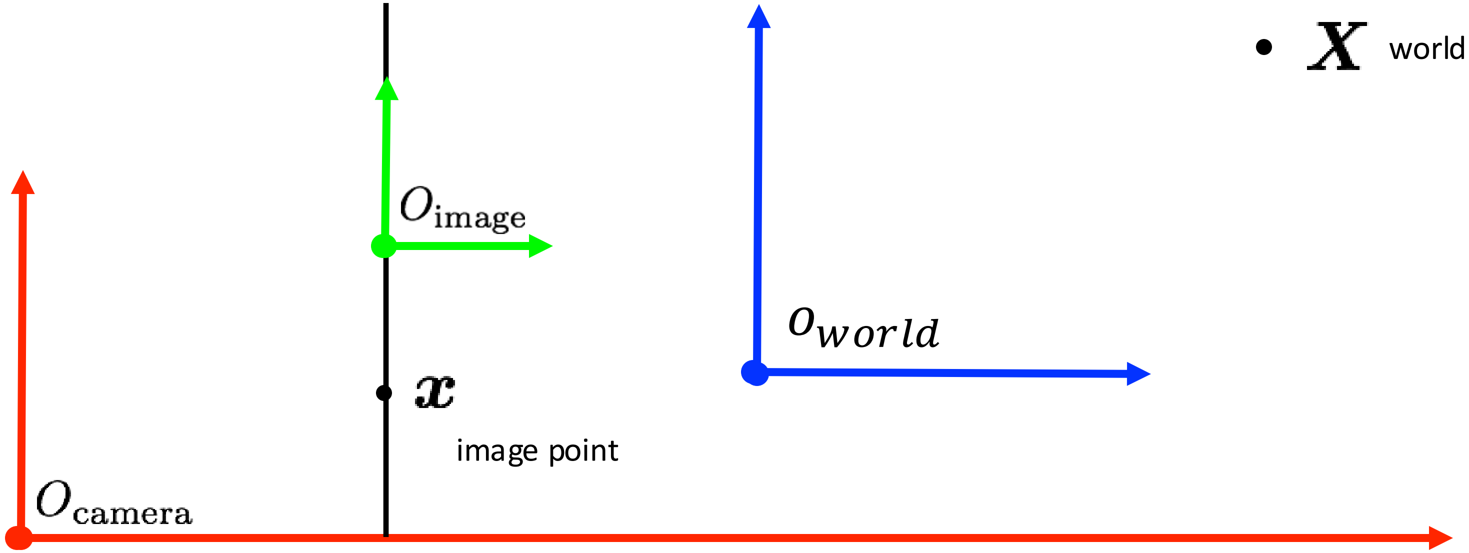
$$K = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

To “calibrate” a camera, I need to compute this matrix K for a given camera
(tells me how to project from world to image coordinates)

(obtain simpler intrinsics by setting s_x, s_y = f and s_θ, o_x, o_y = 0)

Generalizing the camera matrix

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



We need to know the transformations between them.

3D rigid-body transformations

3D translations

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} X + t_x \\ Y + t_y \\ Z + t_z \end{bmatrix}$$

3D rotations

$$R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

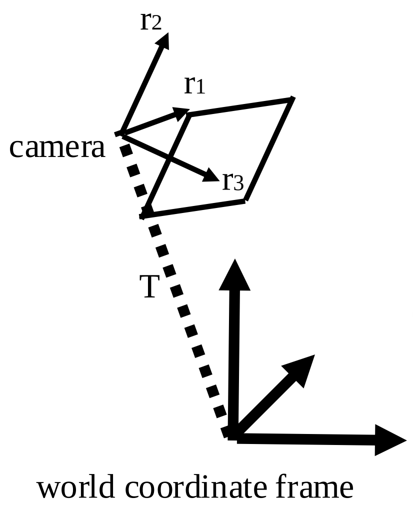
$$R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

“homogenous” world coordinates

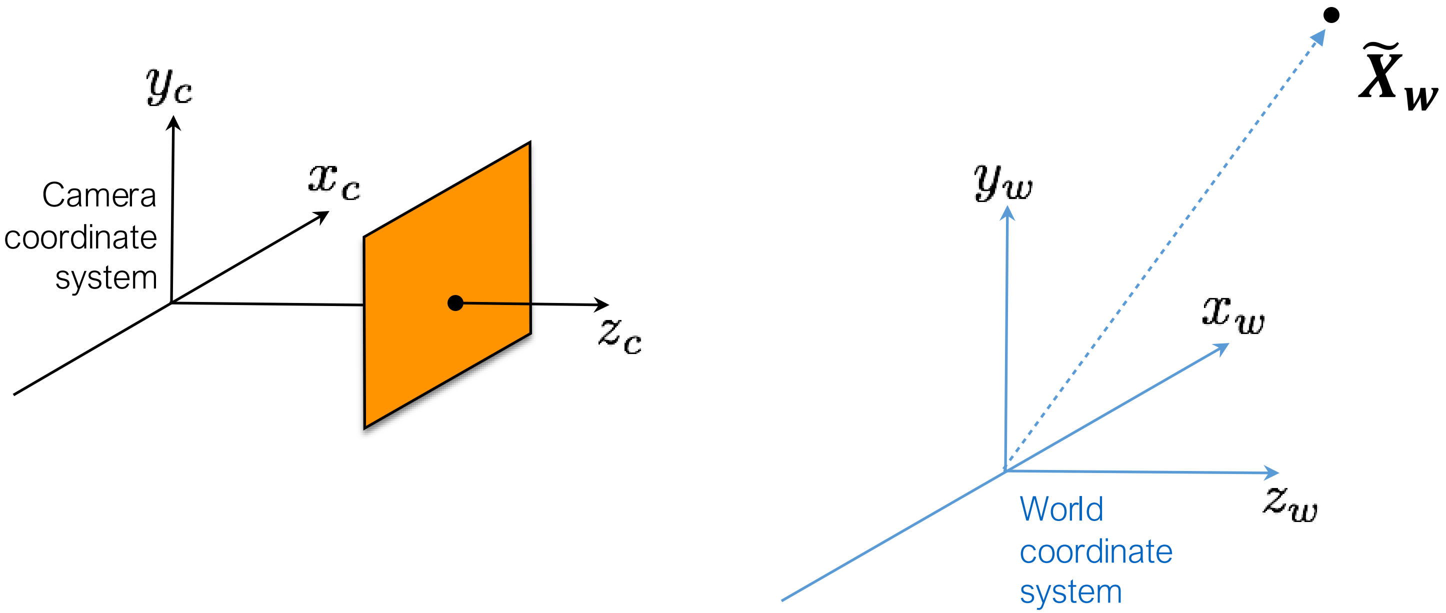
Alternative perspective for rigid transformation:

Think of a camera moving through world coordinate frame

$$R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

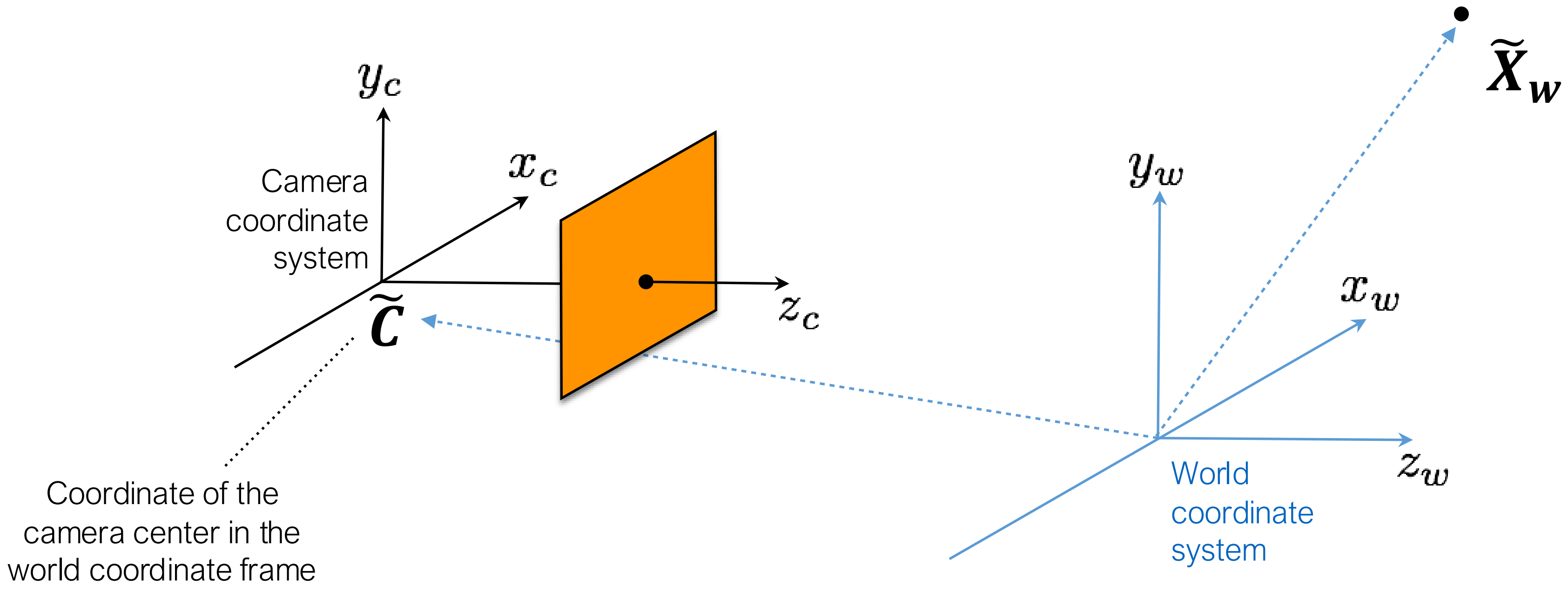


World-to-camera coordinate system transformation

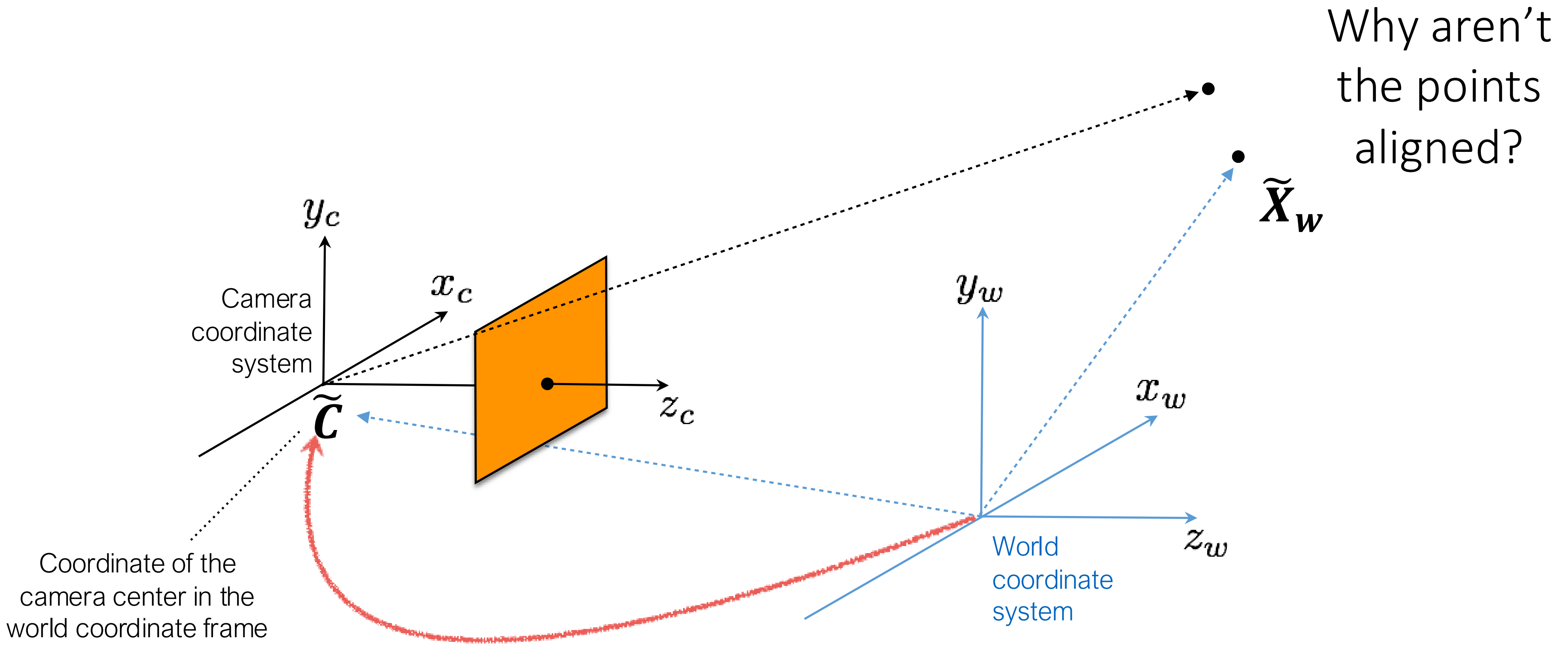


tilde means *heterogeneous* coordinates

World-to-camera coordinate system transformation



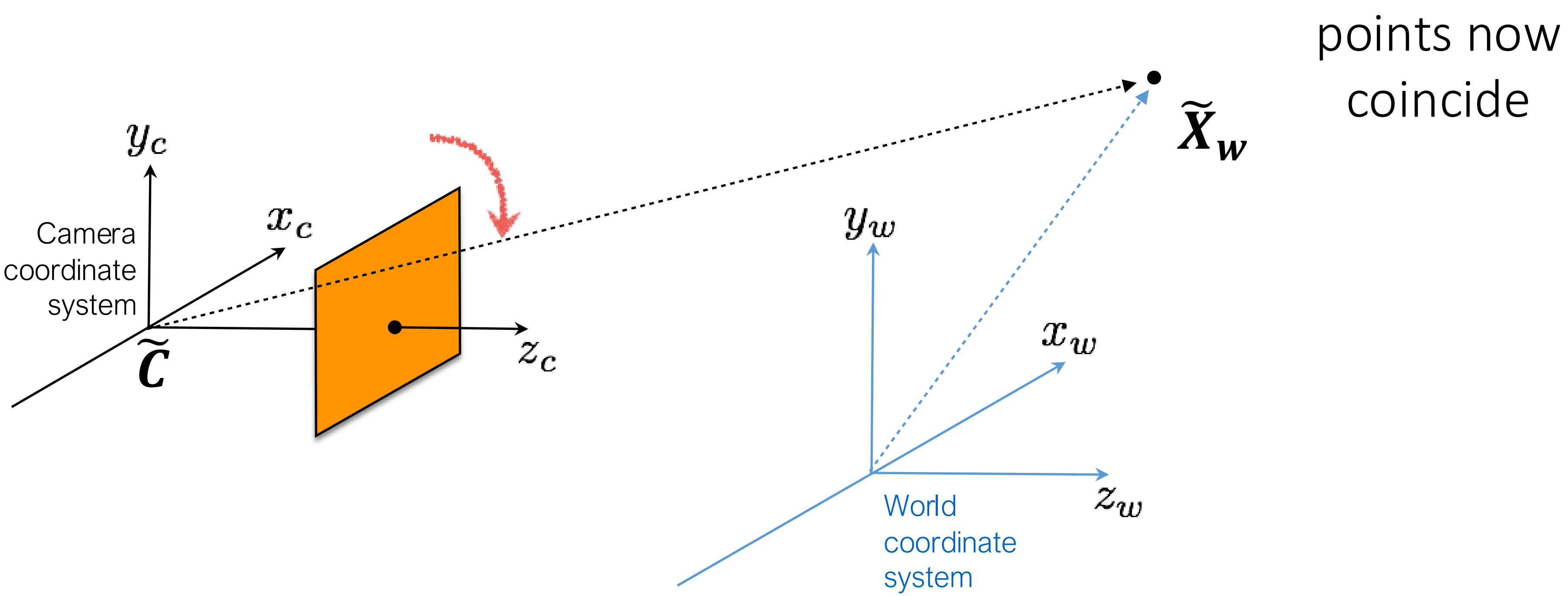
World-to-camera coordinate system transformation



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

translate

World-to-camera coordinate system transformation



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

rotate translate

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

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

How do we write this transformation in homogeneous coordinates?

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

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

In homogeneous coordinates, we have:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

or

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

Incorporating the transform in the camera matrix

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

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

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:

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

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{I} \quad | \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}$$

intrinsic parameters (3 x 3): correspond to camera internals (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 (world-to-camera transformation)

Putting it all together

We can write everything into a single projection:

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

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\mathbf{R} \;\middle|\; -\mathbf{RC} \right]$$

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

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

General pinhole camera matrix

We can decompose the camera matrix like this:

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

(translate first then rotate)

Another way to write the mapping:

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

where $\mathbf{t} = -\mathbf{RC}$

(rotate first then translate)

General pinhole camera matrix

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

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

intrinsic parameters

extrinsic parameters

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

3D rotation

3D translation

Recap

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

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

? ? ? ?

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
intrinsics

3x3
3D rotation

3x3
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 intrinsics	3x3 3D rotation	3x3 identity	3x1 3D translation

Quiz

The camera matrix relates what two quantities?

Quiz

The camera matrix relates what two quantities?

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

homogeneous 3D points to 2D image points

Quiz

The camera matrix relates what two quantities?

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

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?

Quiz

The camera matrix relates what two quantities?

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

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.

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

More general camera matrices

CCD camera: pixels may not be square.

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

How many degrees of freedom?

More general camera matrices

CCD camera: pixels may not be square.

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

How many degrees of freedom?

10 DOF

More general camera matrices

Finite projective camera: sensor be skewed.

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

How many degrees of freedom?

More general camera matrices

Finite projective camera: sensor be skewed.

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

How many degrees of freedom?

11 DOF