

Introduction to Computer Vision

Kaveh Fathian

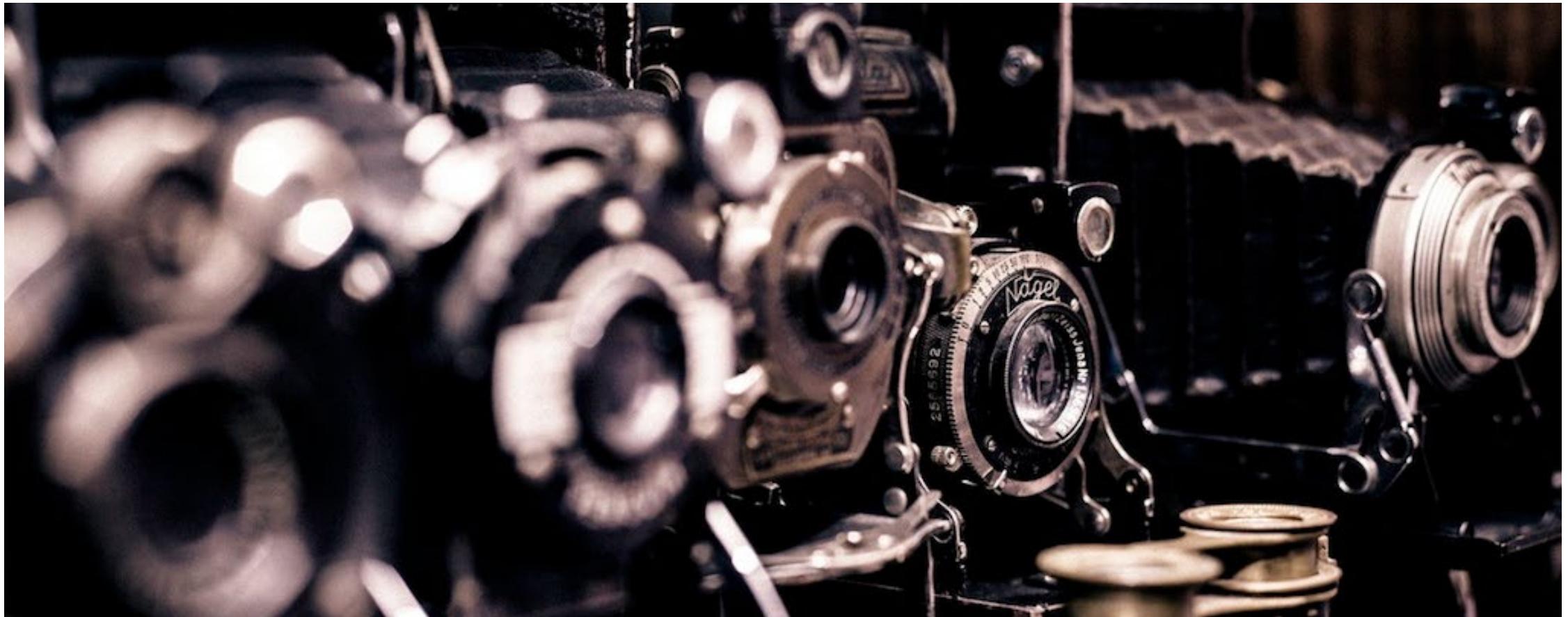
Assistant Professor

Computer Science Department

Colorado School of Mines

Lecture 10

Camera Geometry



Camera Geometry

Learning outcomes:

- Pinhole Camera Model
- Transformations
- Homogenous Coordinates
- Distortions
- Intrinsic and Extrinsic Matrices

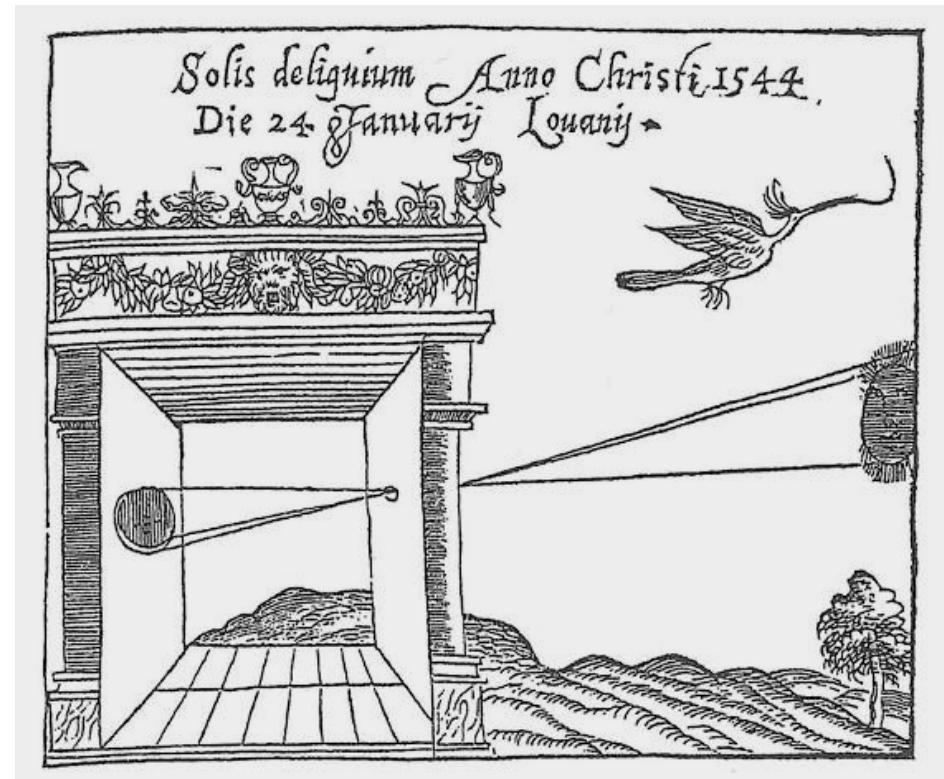
Camera Obscura

First mention ...



Chinese philosopher Mozi
(470 to 390 BC)

First camera ...



Greek philosopher Aristotle
(384 to 322 BC)

First Photograph

Oldest surviving photograph

- Took 8 hours on pewter plate (with some reports claiming it was several days)

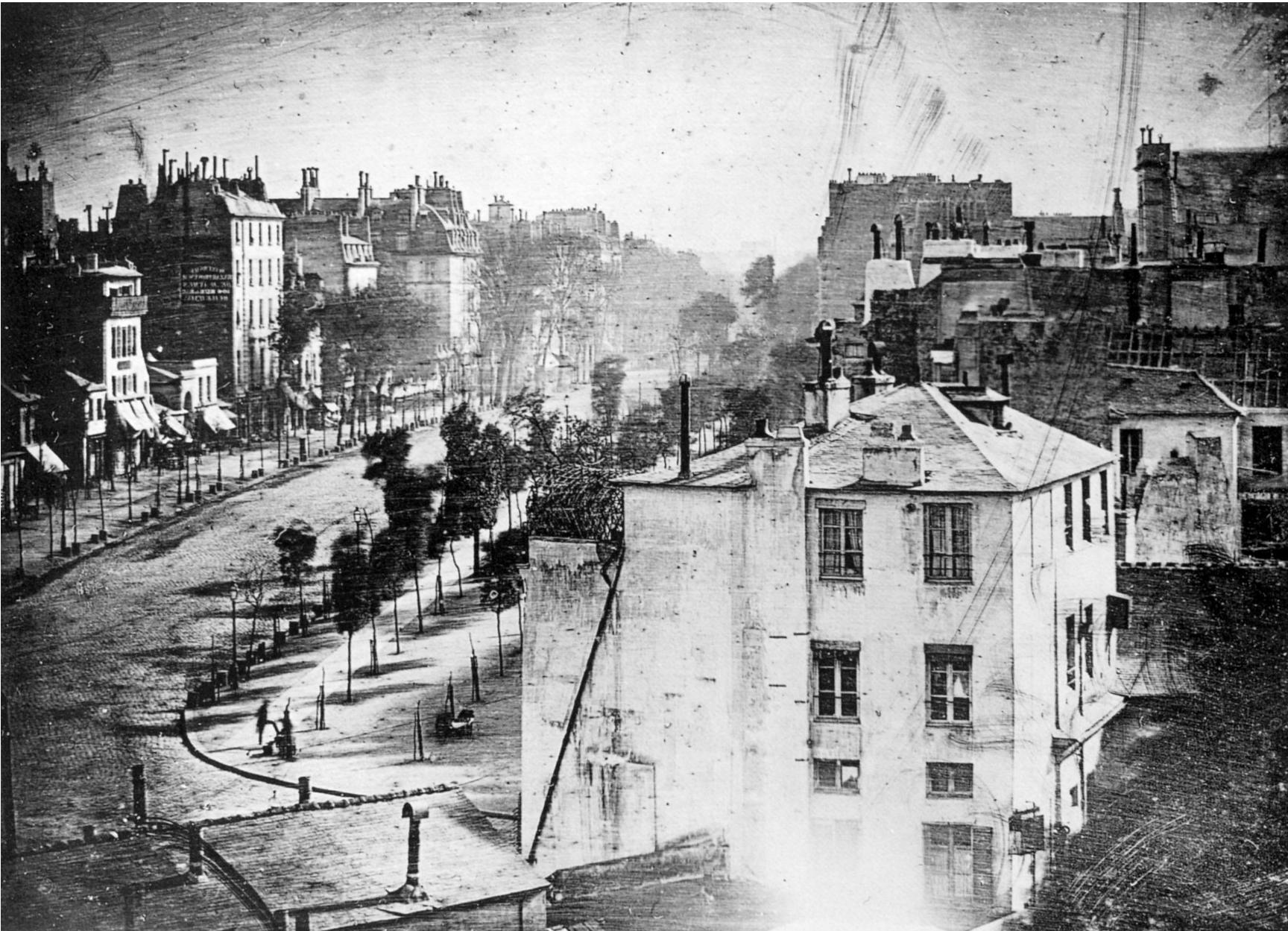


Joseph Niepce, 1826



Stored at UT Austin

First Photograph with People



Boulevard du Temple, Louis
Daguerre, 1838.Wikipedia.

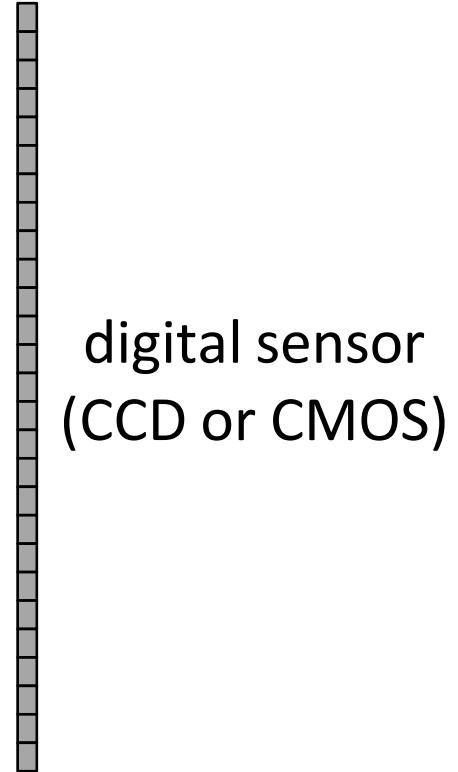
First Digital Camera

- First prototype by Kodak engineer Steven Sasson in 1975
- Captured black & white images
- Resolution of 0.01 megapixels
- Images stored on a cassette tape
- Bulky device weighed ~8 pounds
- Took 23 seconds to capture a single image



Pinhole Camera Model

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

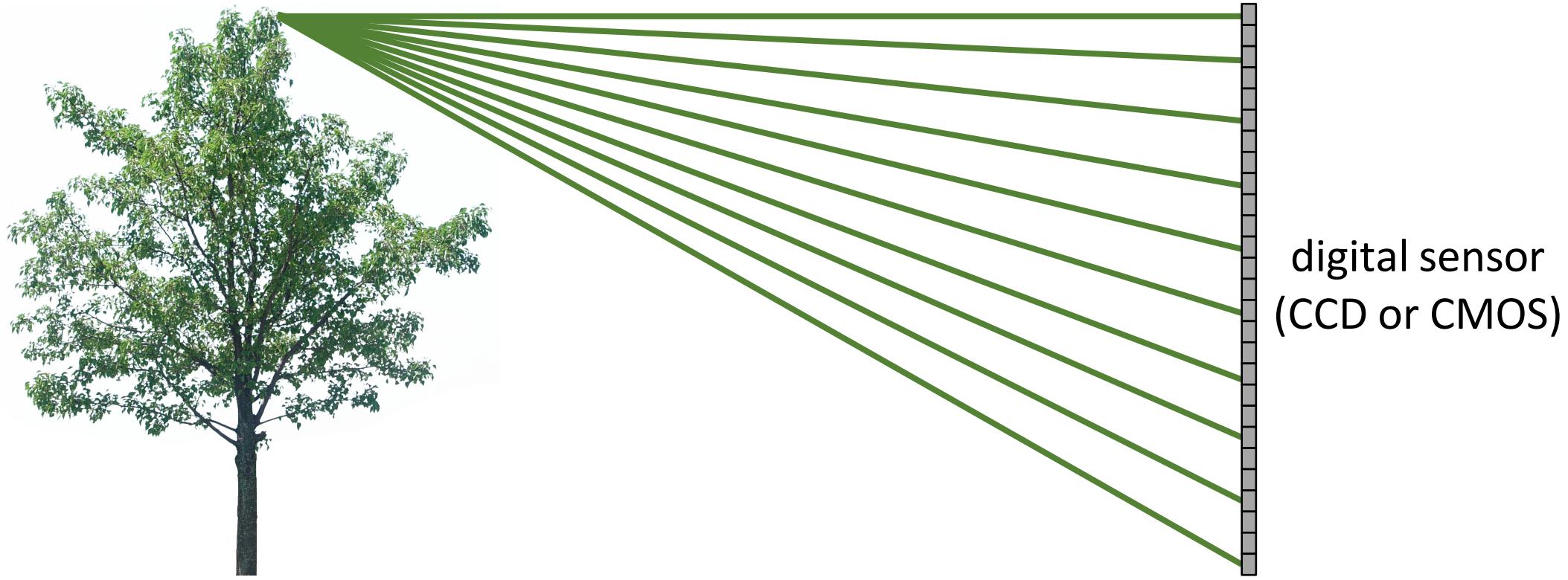


digital sensor
(CCD or CMOS)



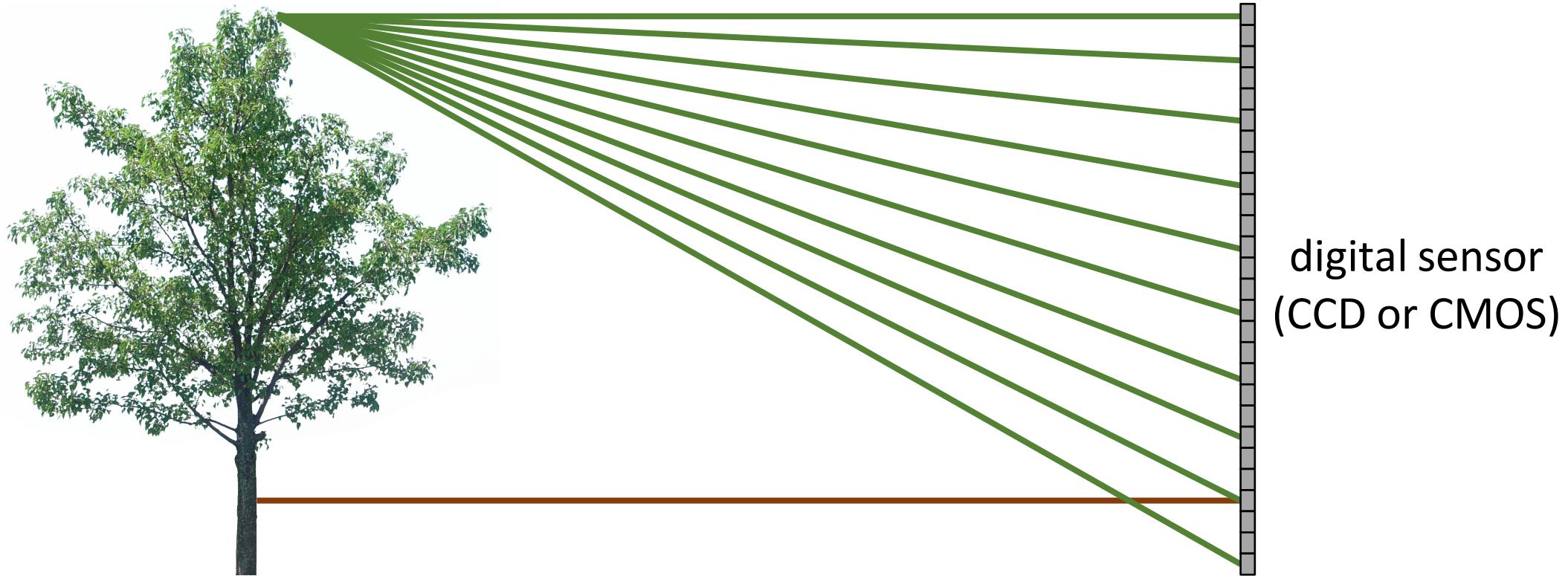
Bare-sensor imaging

real-world
object



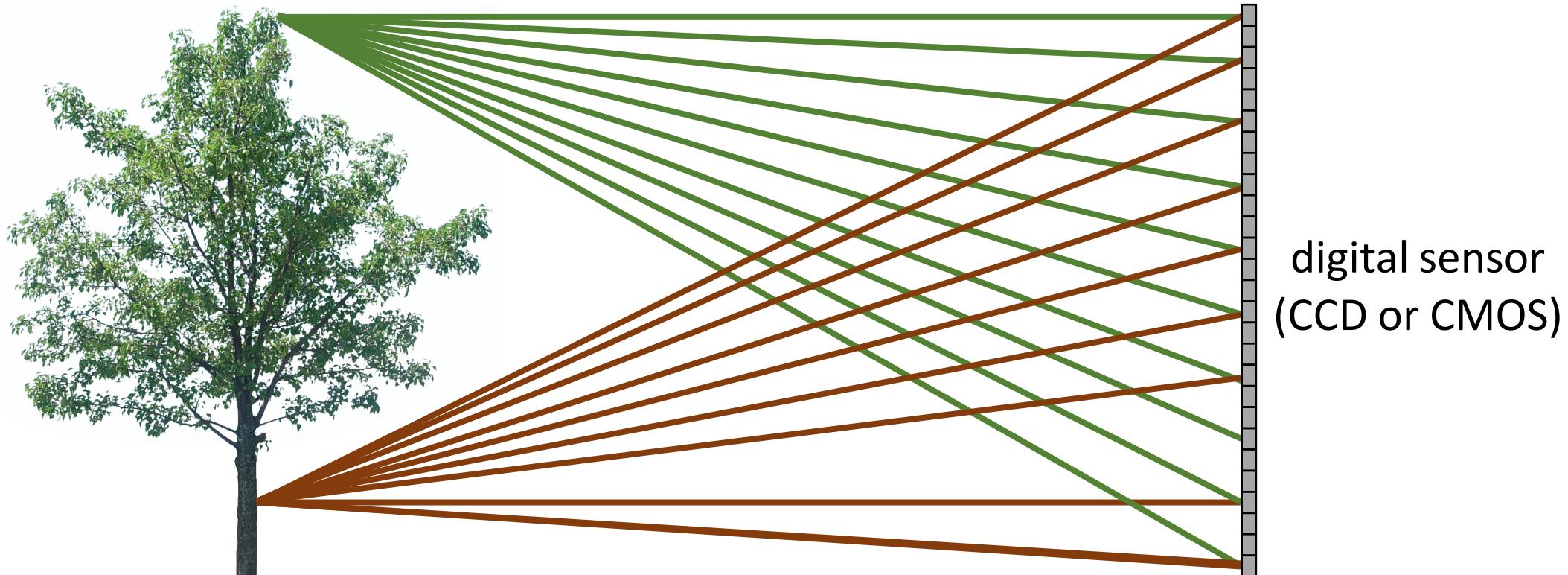
Bare-sensor imaging

real-world
object



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



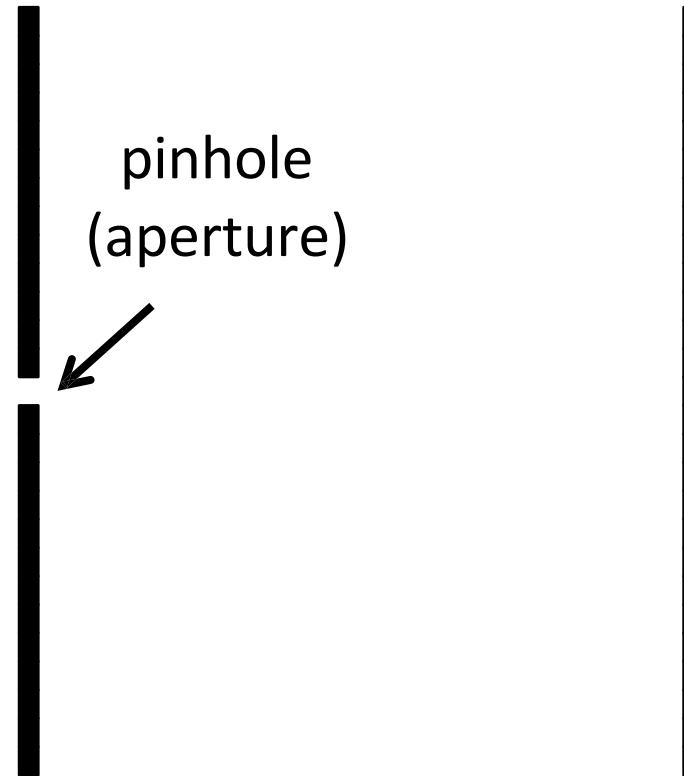
All scene points contribute to all sensor pixels

Let's add something to this scene

real-world
object



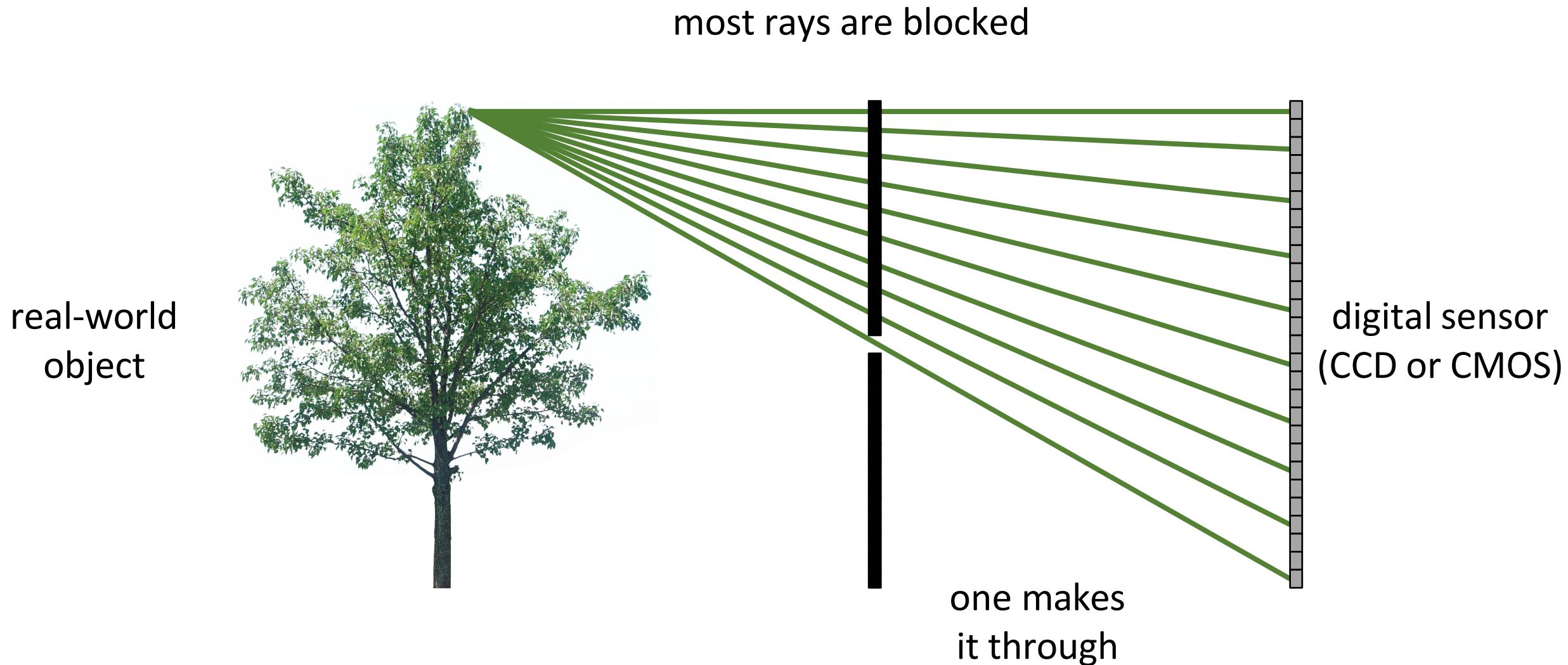
barrier (diaphragm)



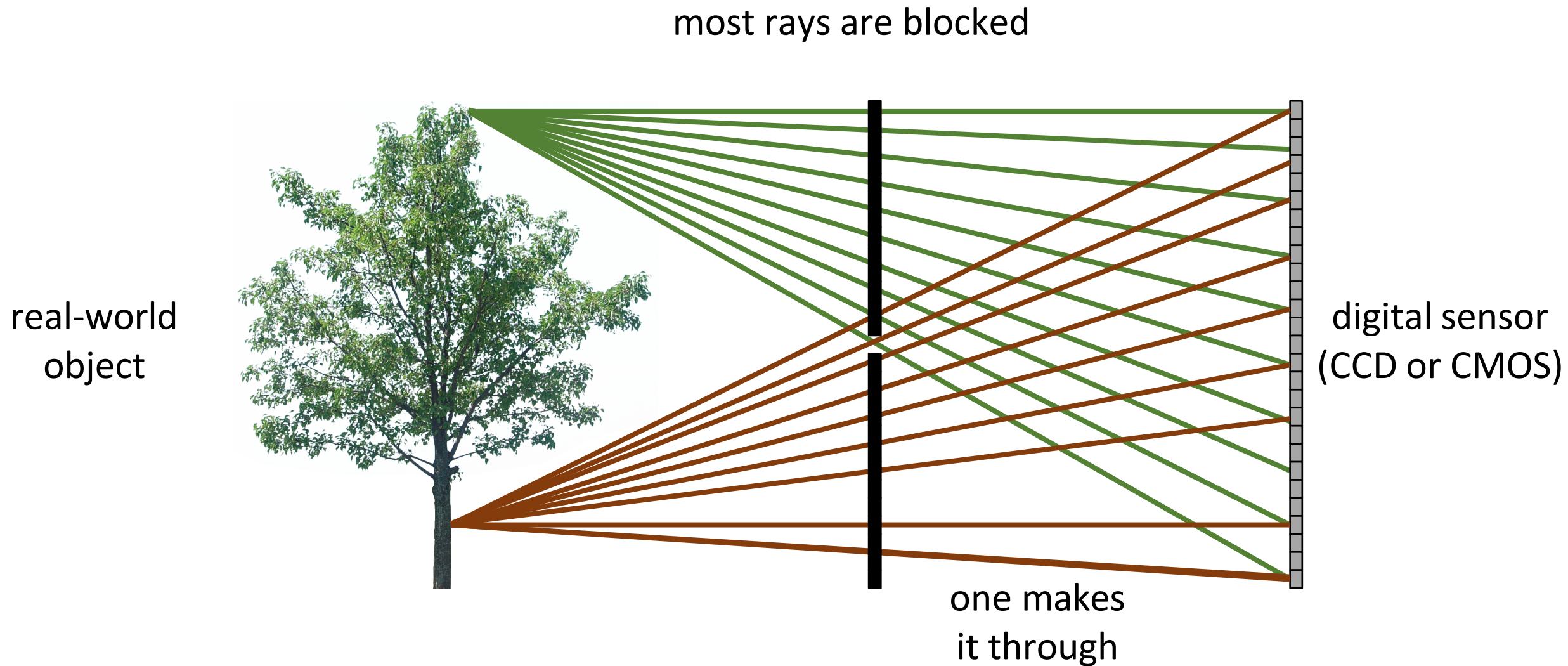
digital sensor
(CCD or CMOS)

What would an image taken like this look like?

Pinhole imaging

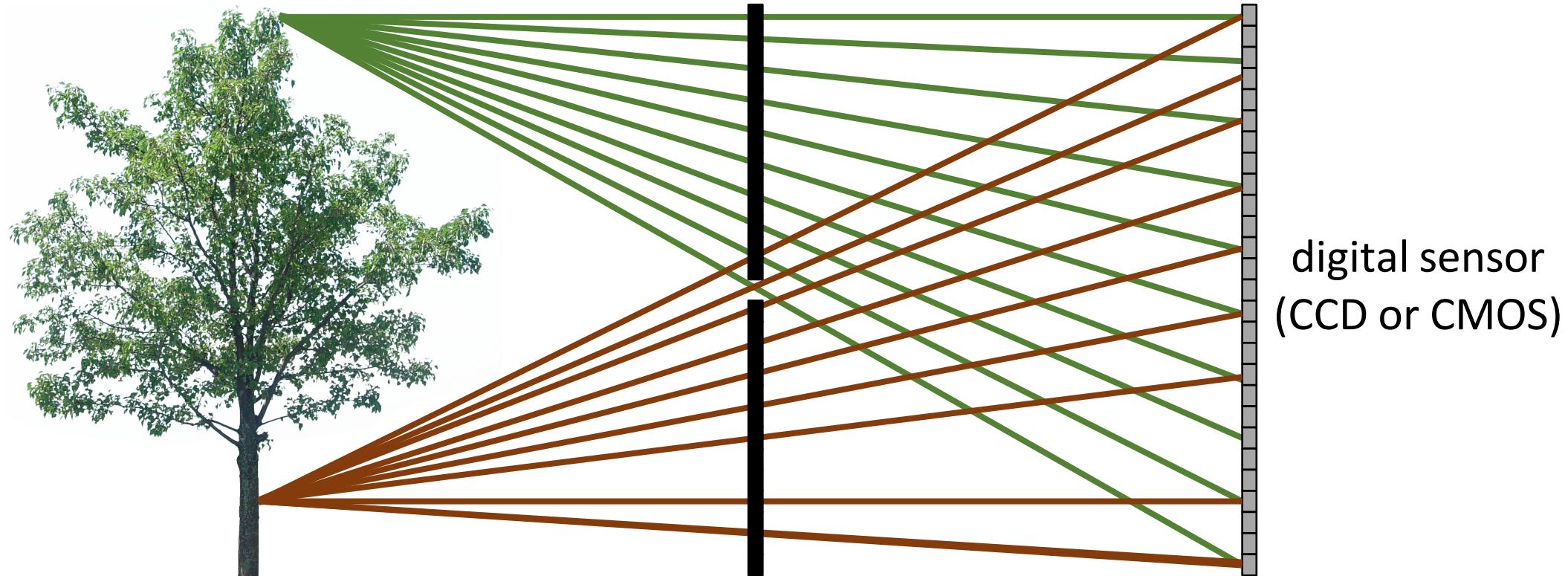


Pinhole imaging



Pinhole imaging

real-world
object

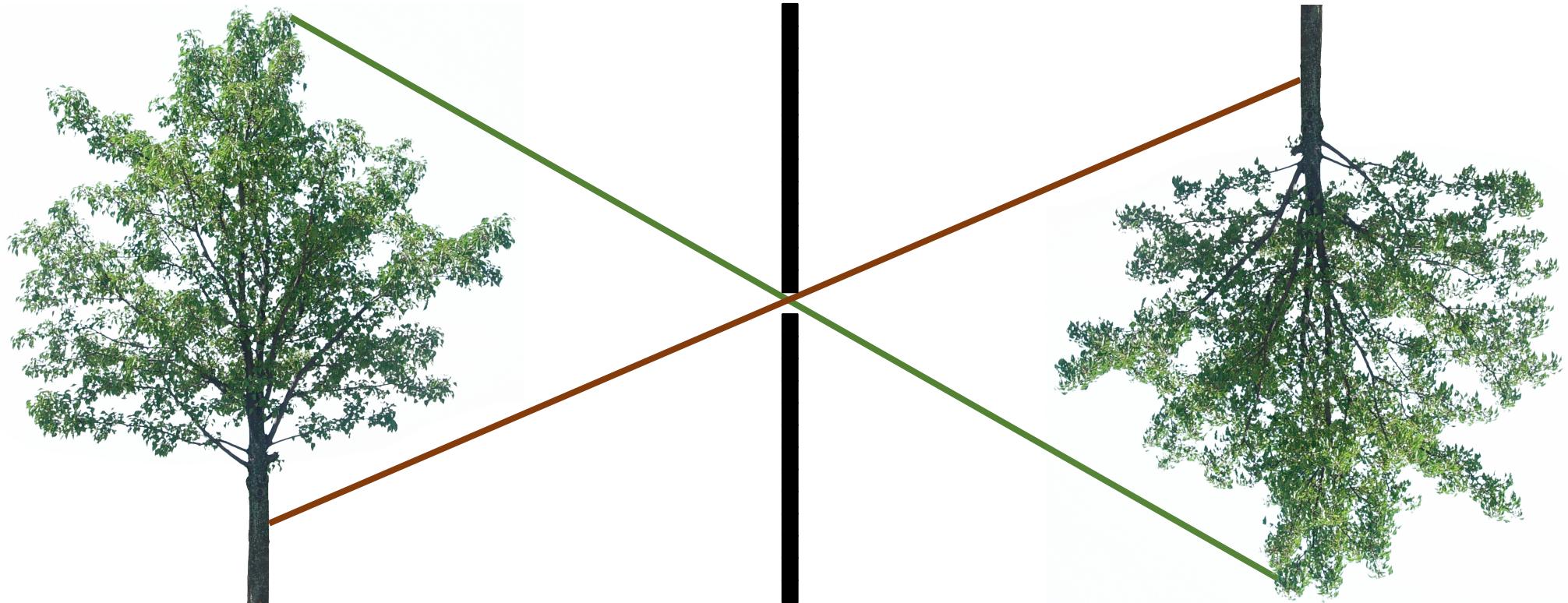


Each scene point contributes to only one sensor pixel

What does the
image on the
sensor look like?

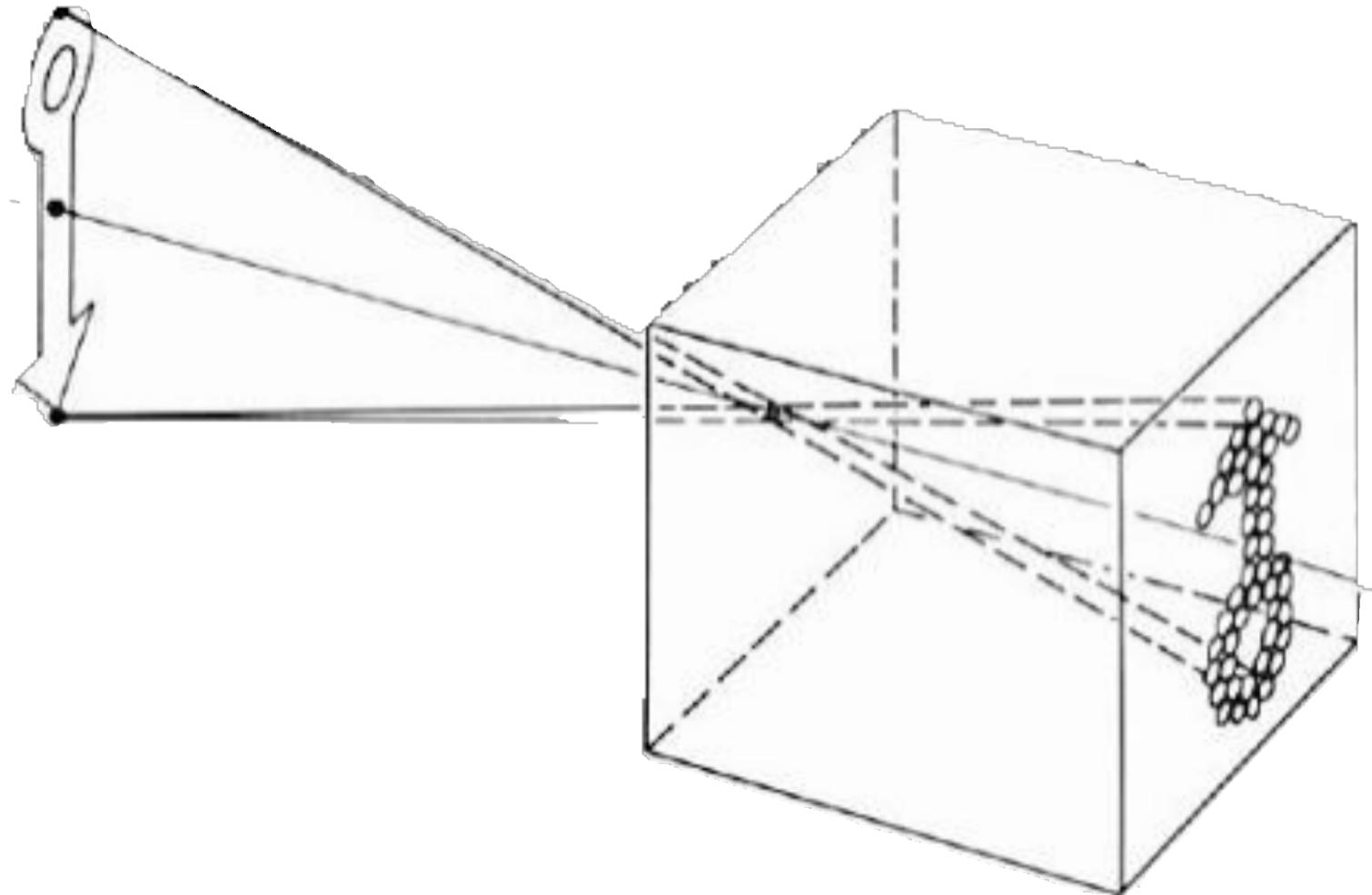
Pinhole imaging

real-world
object



copy of real-world object
(inverted and scaled)

Pinhole camera a.k.a. camera obscura

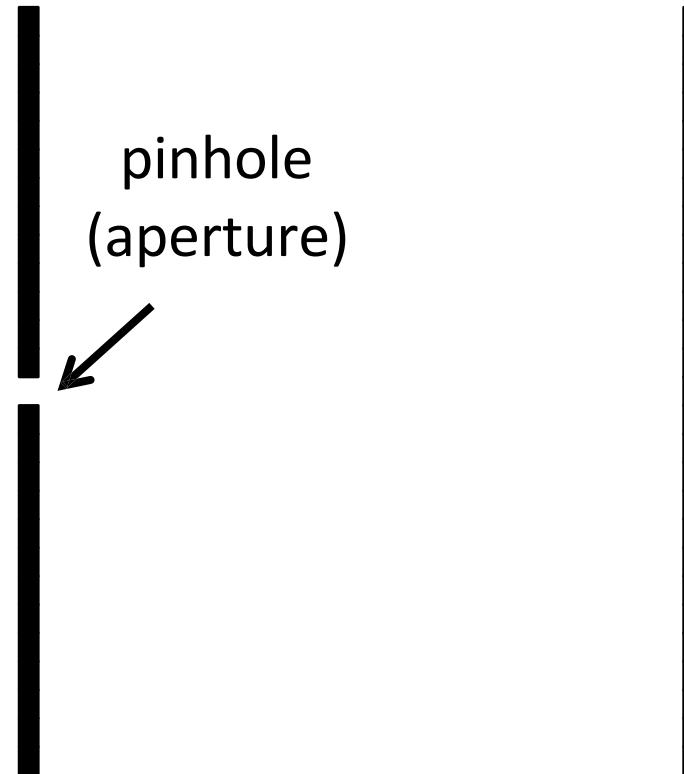


Pinhole camera terms

real-world
object



barrier (diaphragm)



digital sensor
(CCD or CMOS)

Pinhole camera terms

real-world
object



barrier (diaphragm)

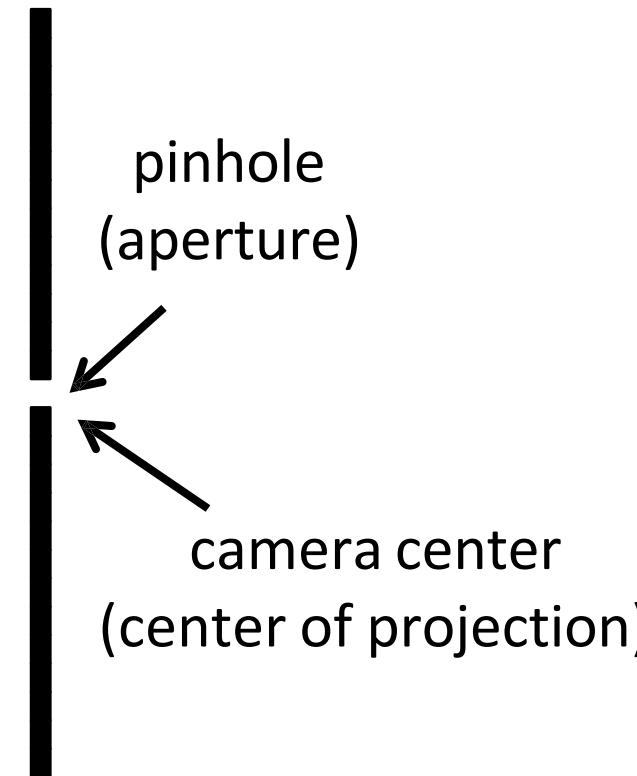
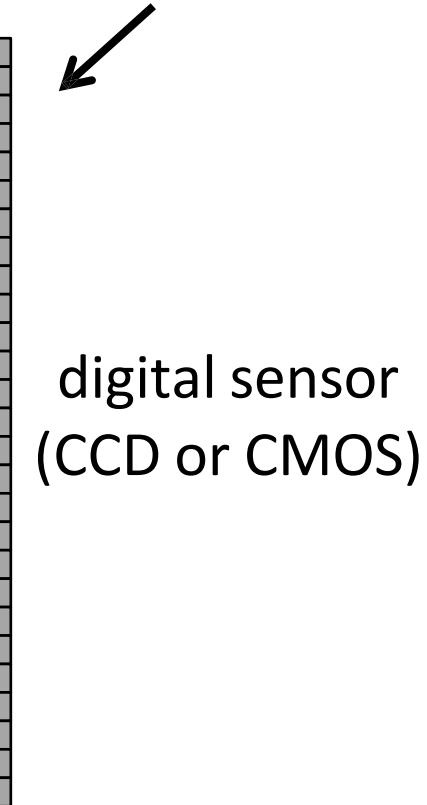
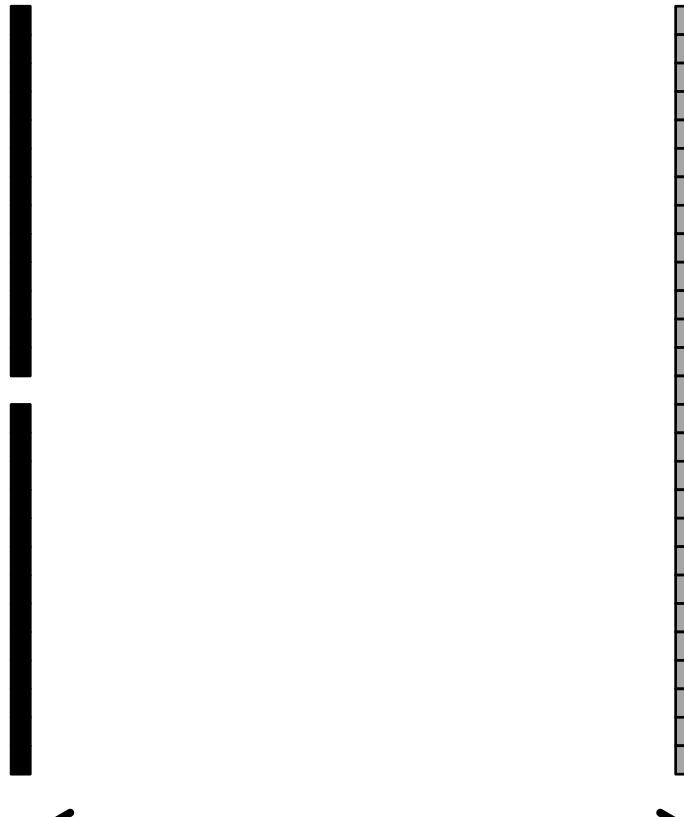


image plane



Focal length

real-world
object

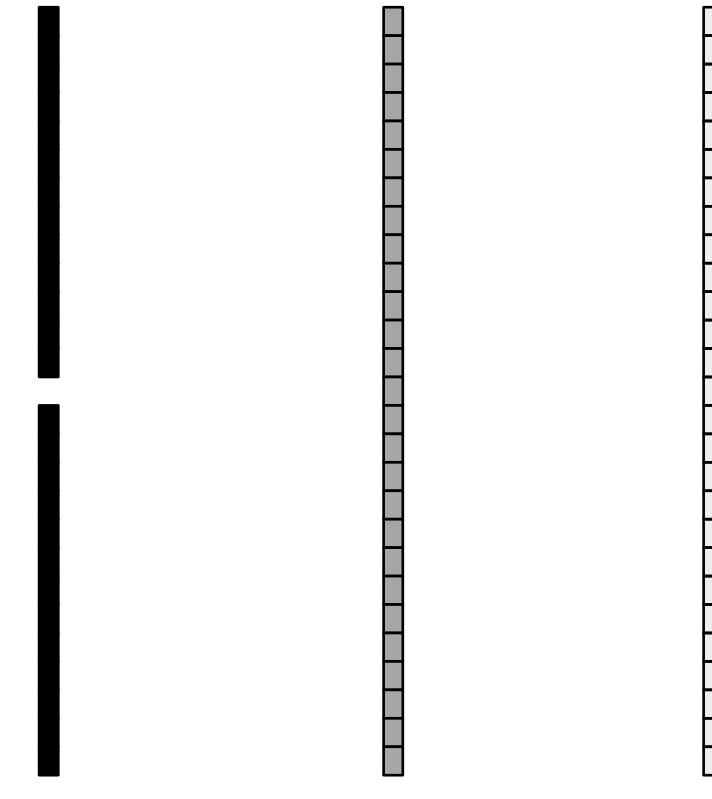


focal length f

Focal length

What happens as we change the focal length?

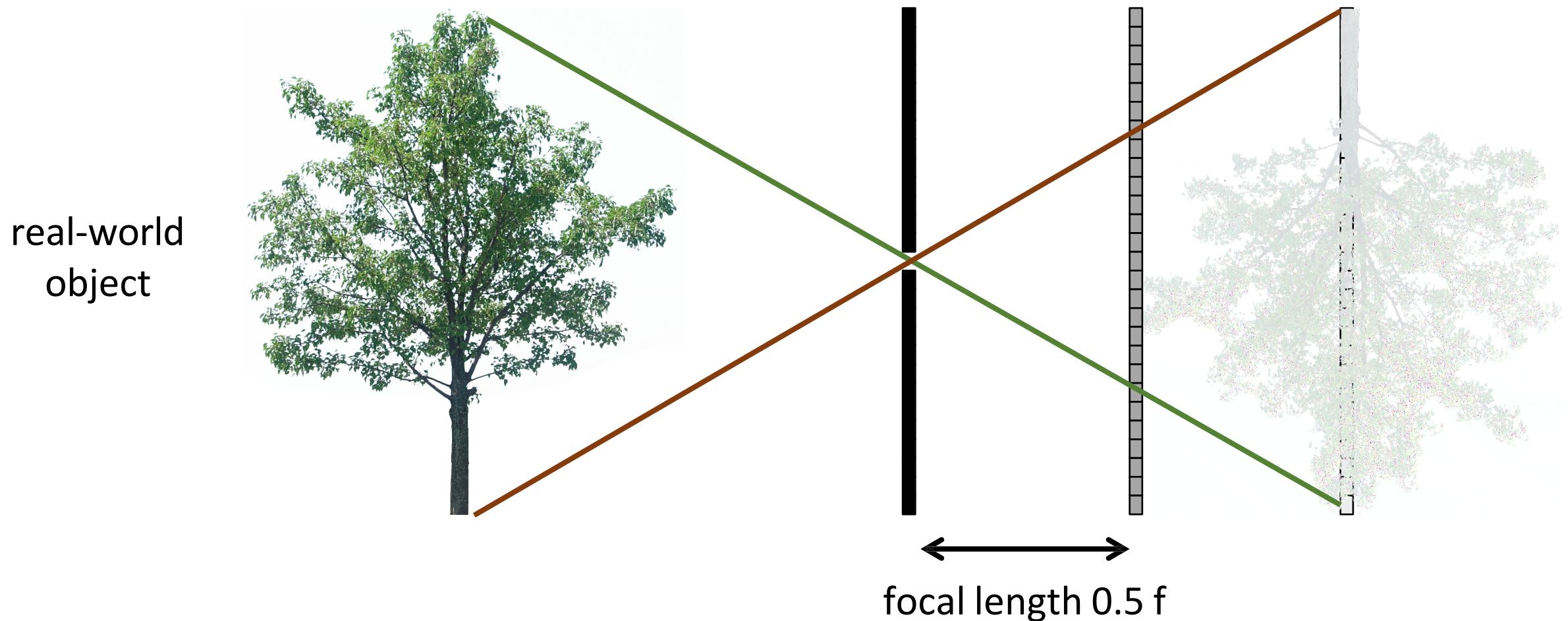
real-world
object



focal length $0.5 f$

Focal length

What happens as we change the focal length?

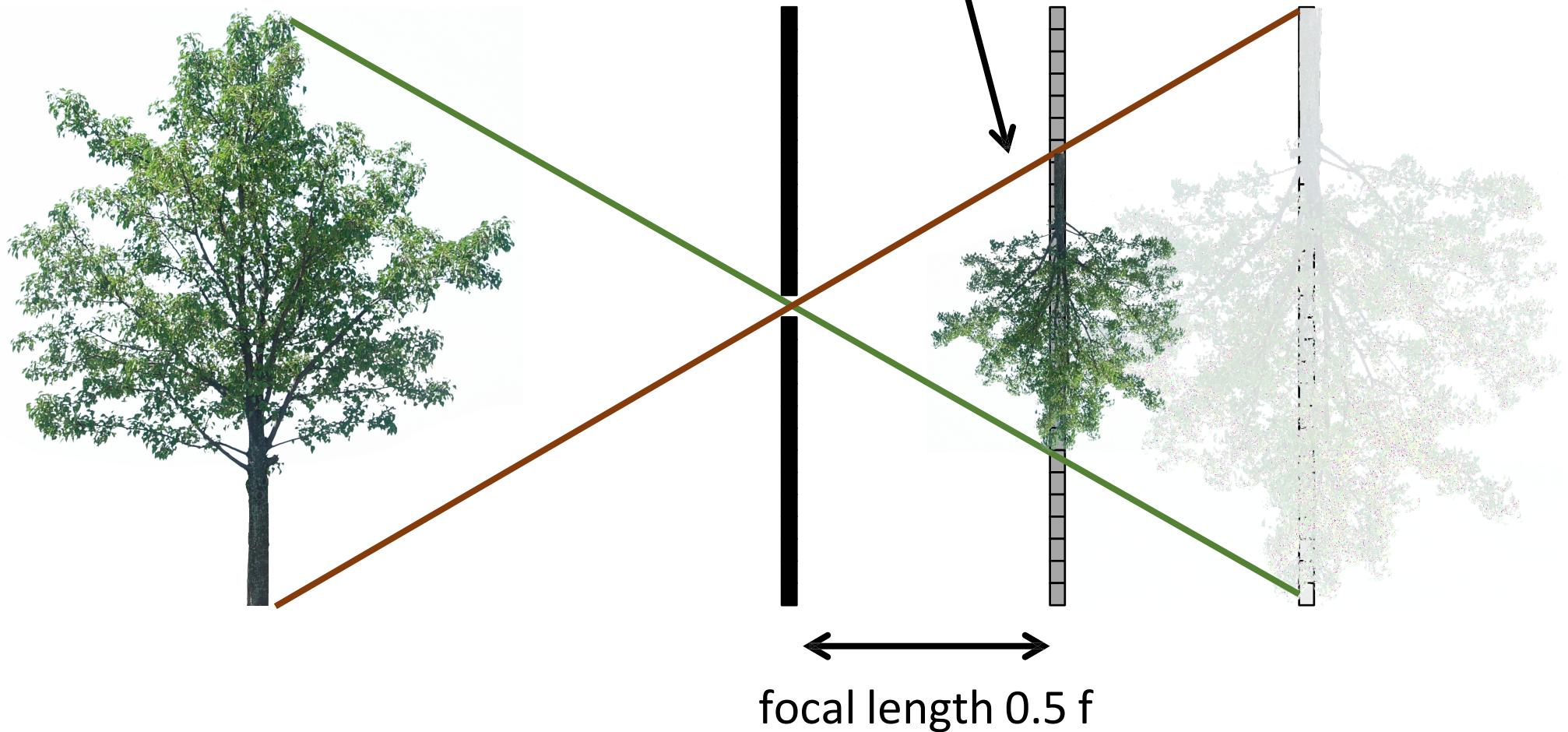


Focal length

What happens as we change the focal length?

object projection is half the size

real-world
object

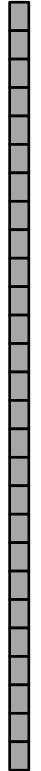


Pinhole size

real-world
object



pinhole
diameter



Ideal pinhole has infinitesimally small size

- In practice that is impossible.

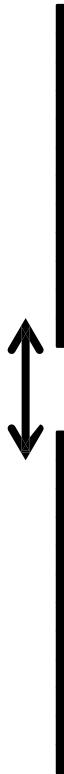
Pinhole size

What happens as we change the pinhole diameter?

real-world
object



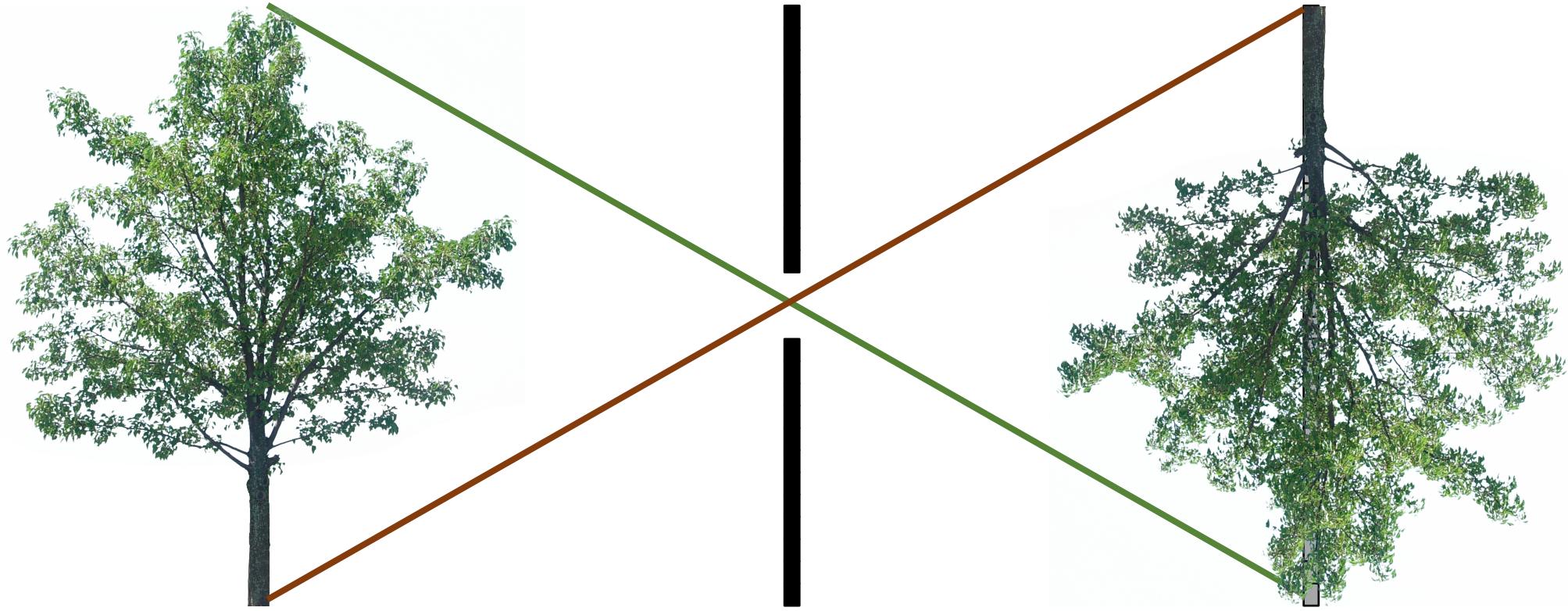
pinhole
diameter



Pinhole size

What happens as we change the pinhole diameter?

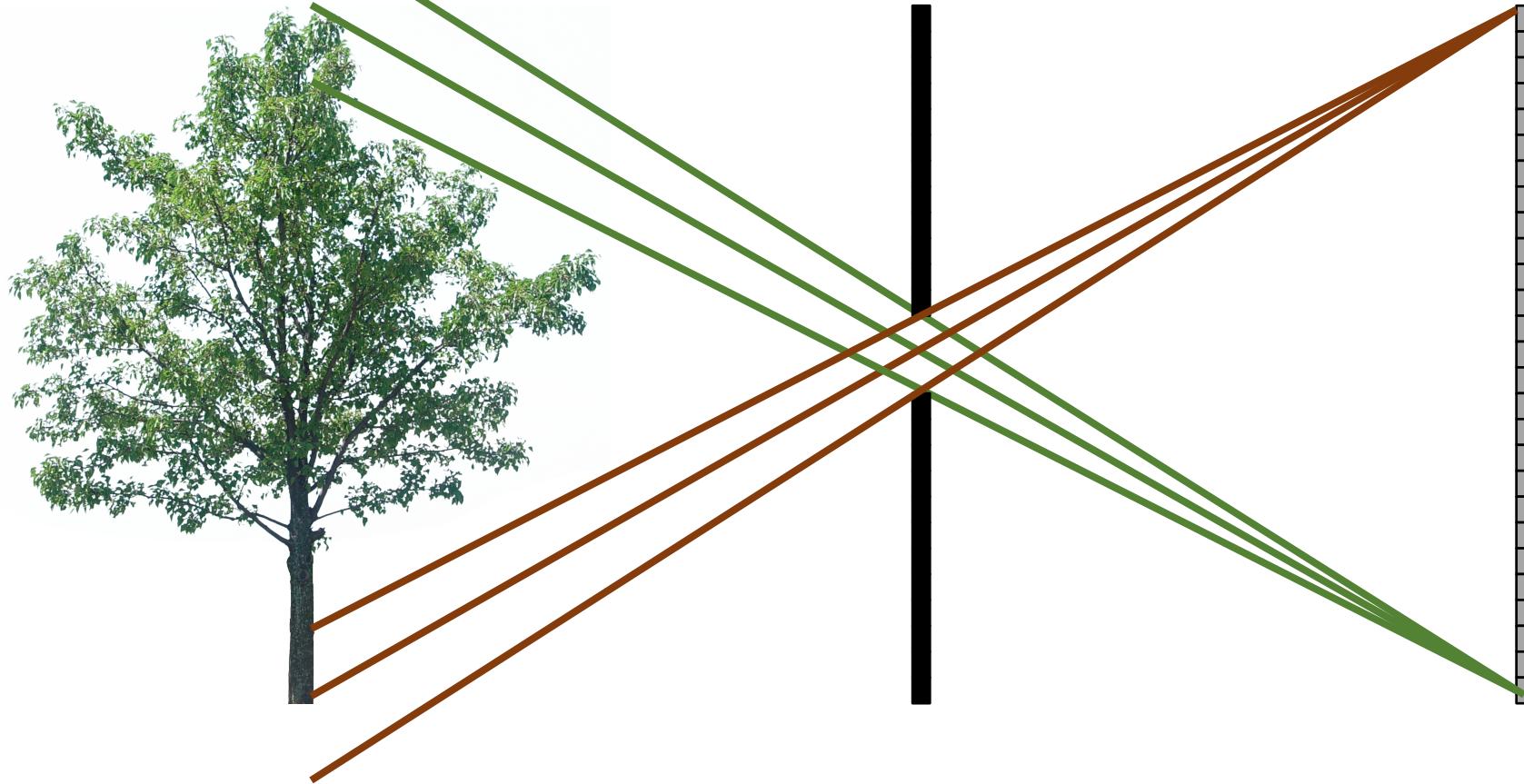
real-world
object



Pinhole size

What happens as we change the pinhole diameter?

real-world
object

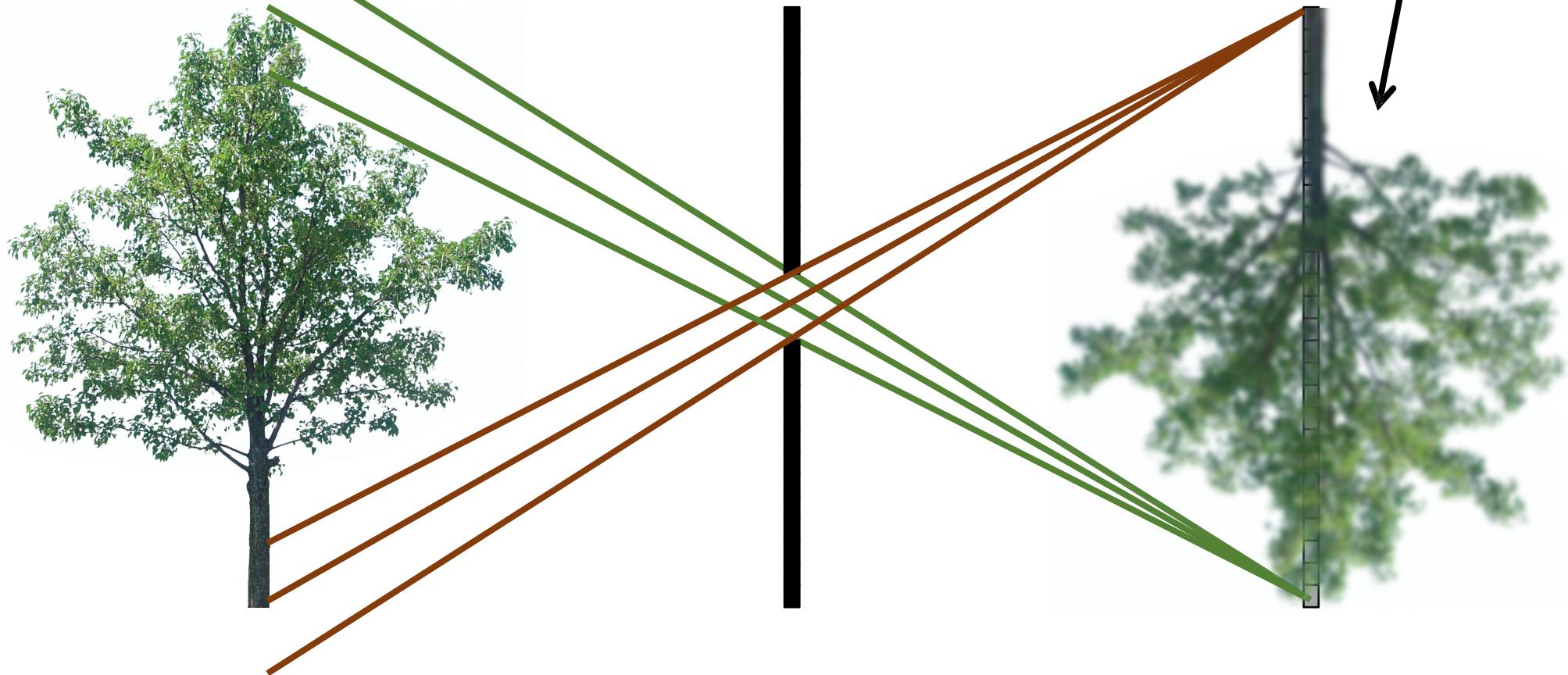


Pinhole size

What happens as we change the pinhole diameter?

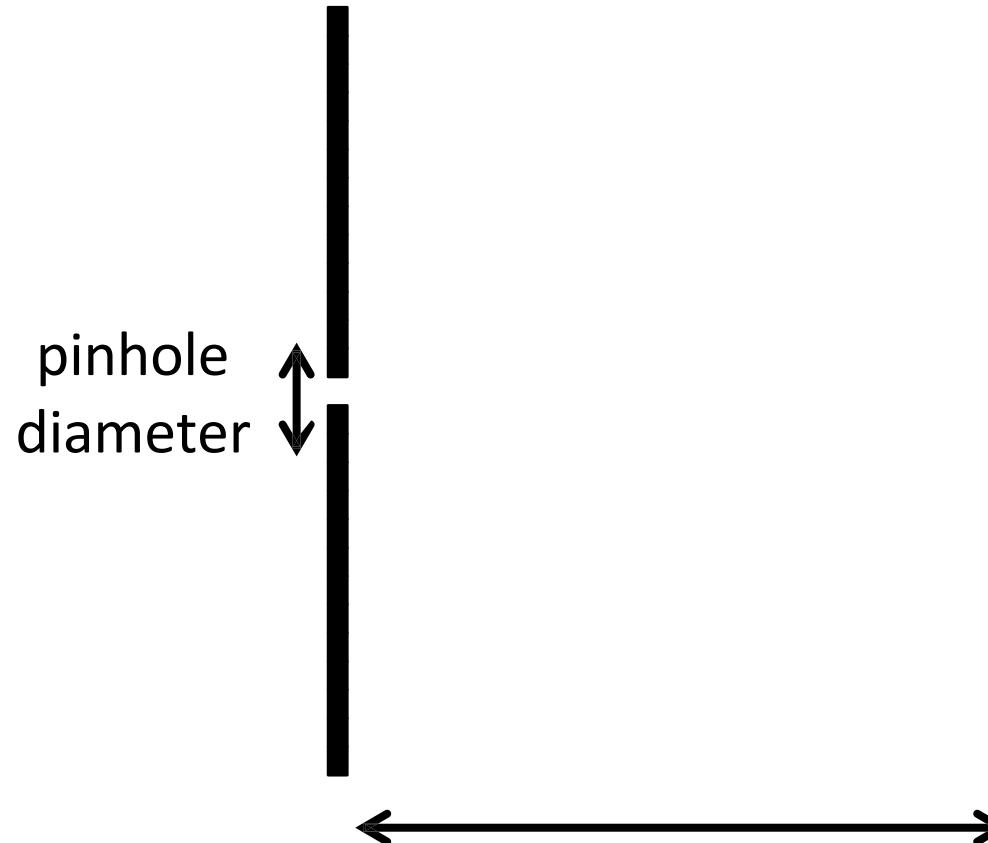
real-world
object

object projection becomes
blurrier



What about light efficiency?

real-world
object

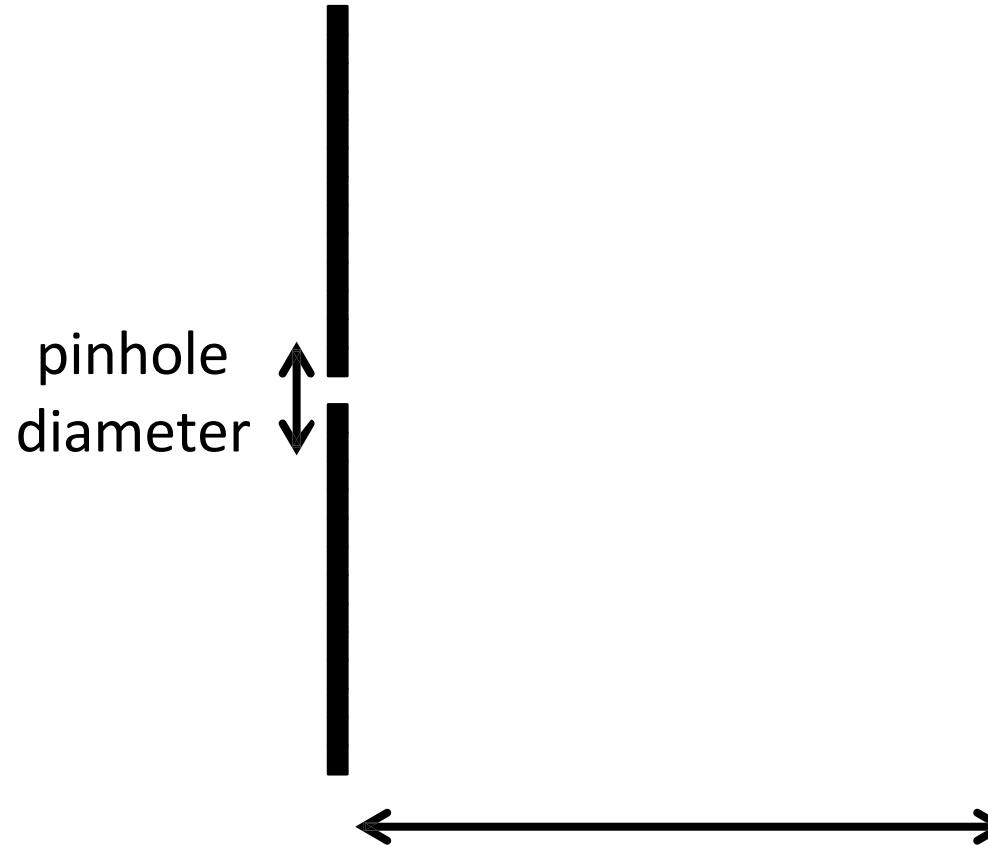


- What is the effect of doubling the pinhole diameter?
- What is the effect of doubling the focal length?

focal length f

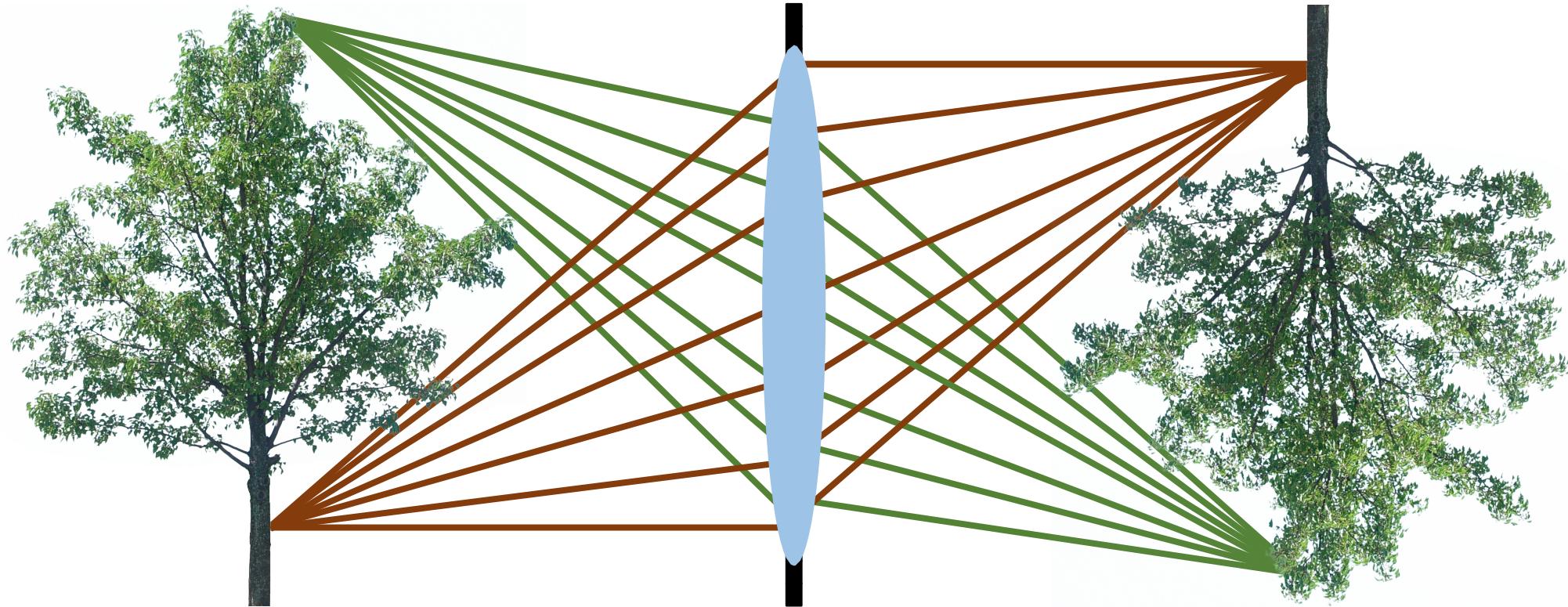
What about light efficiency?

real-world
object



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

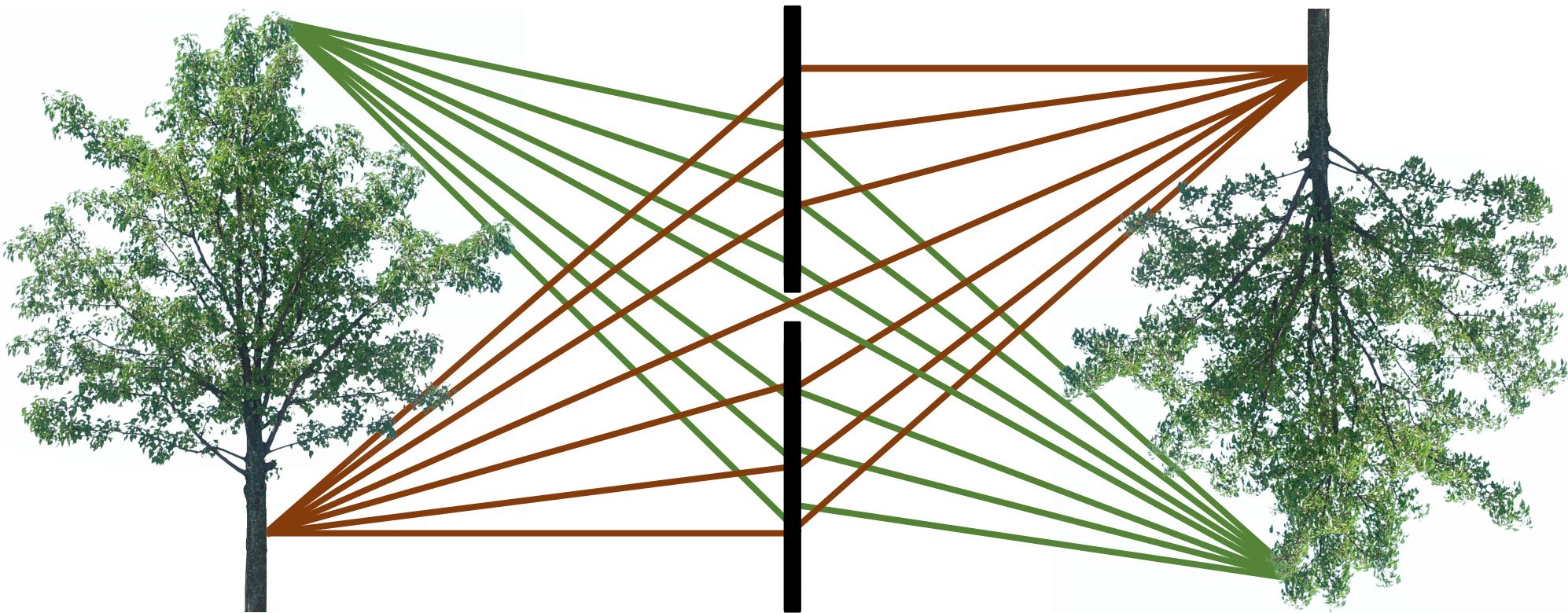
The lens camera



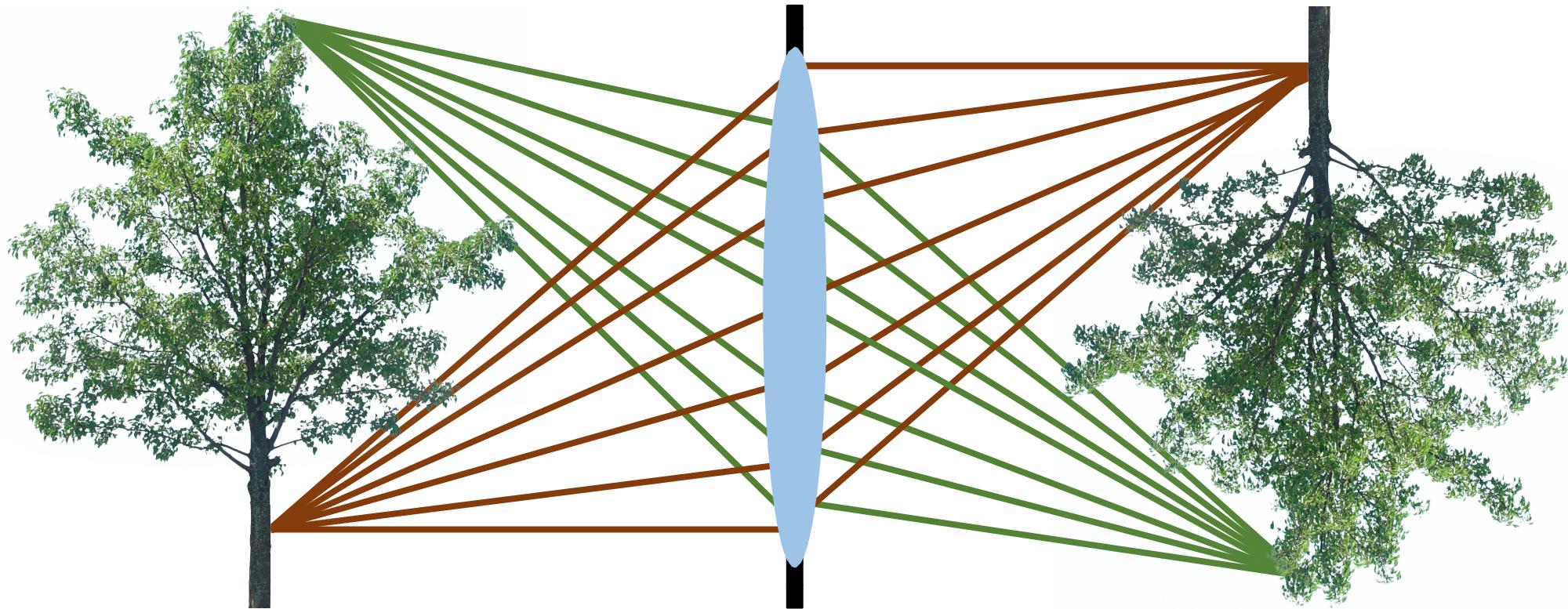
Lenses map “bundles” of rays from points on the scene to the sensor.

How does this mapping work exactly?

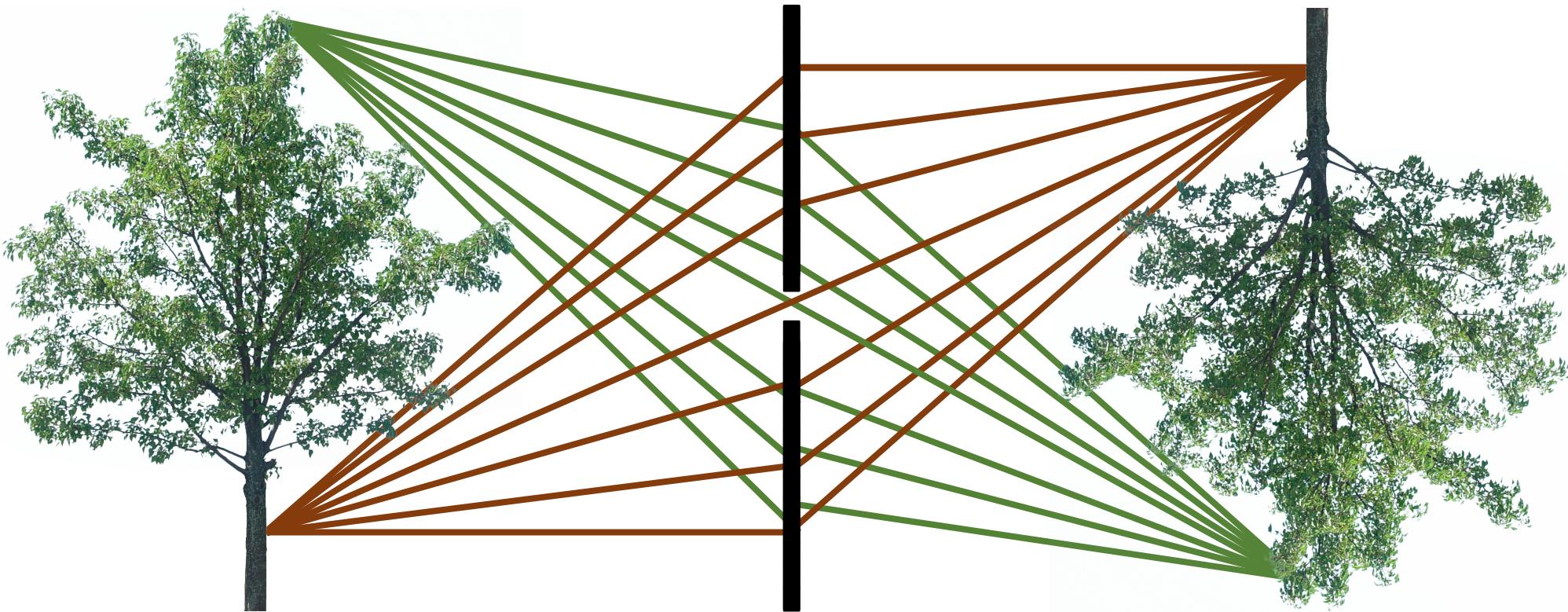
The pinhole camera



The lens camera

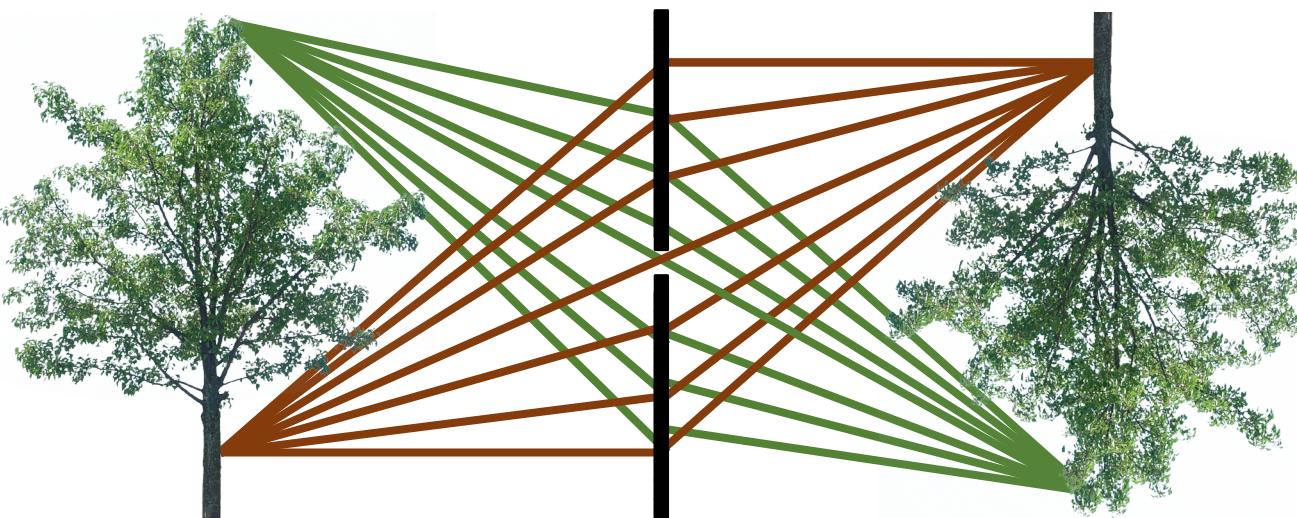
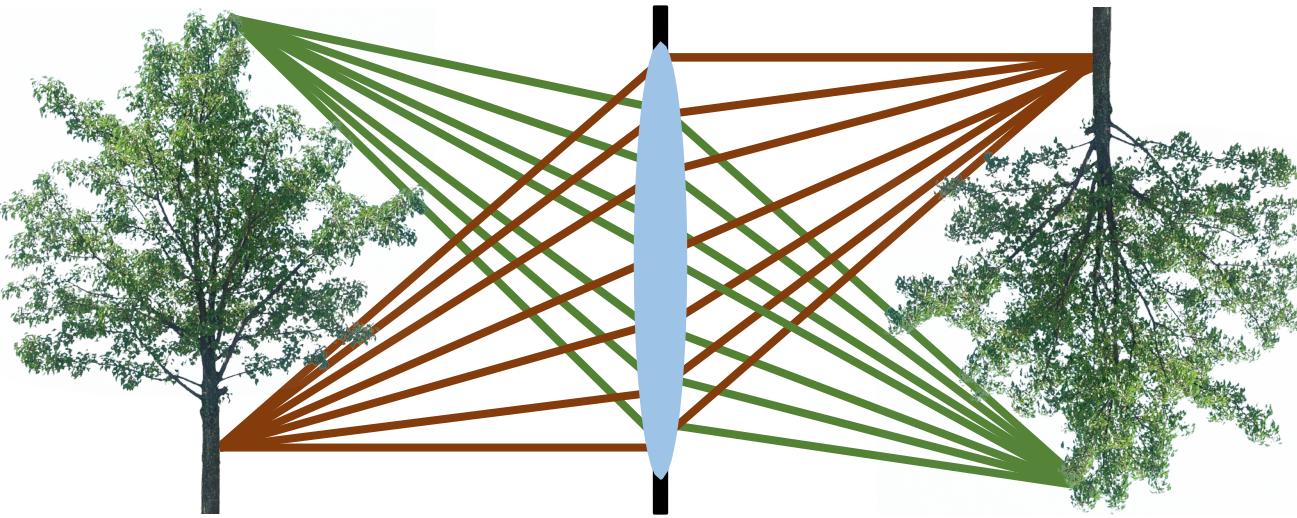


The pinhole camera



Central rays propagate in the same way for both models!

Describing both lens and pinhole cameras

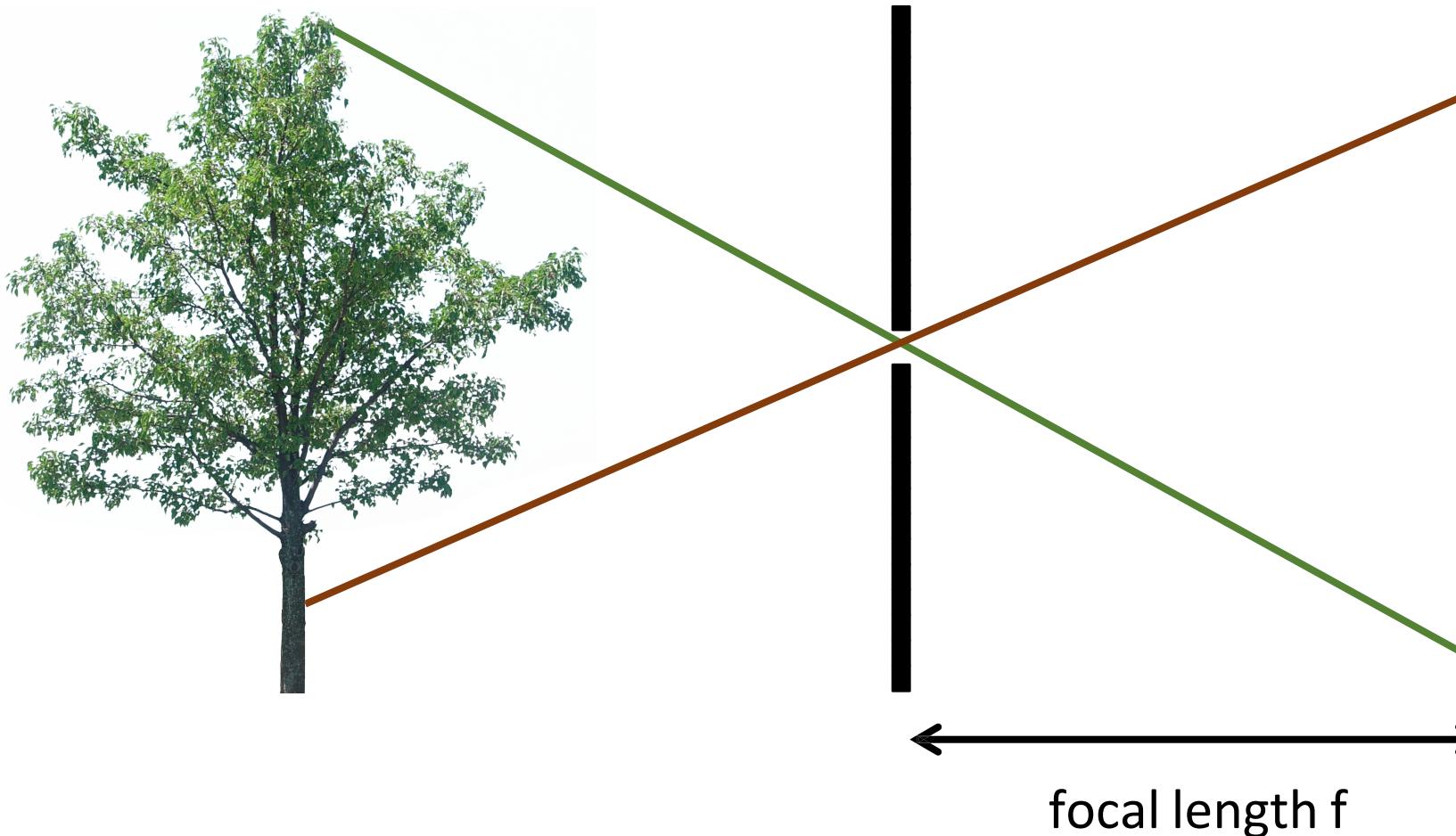


We can derive properties and descriptions that hold for both camera models if:

- We use only central rays.
- We assume the lens camera is in focus.

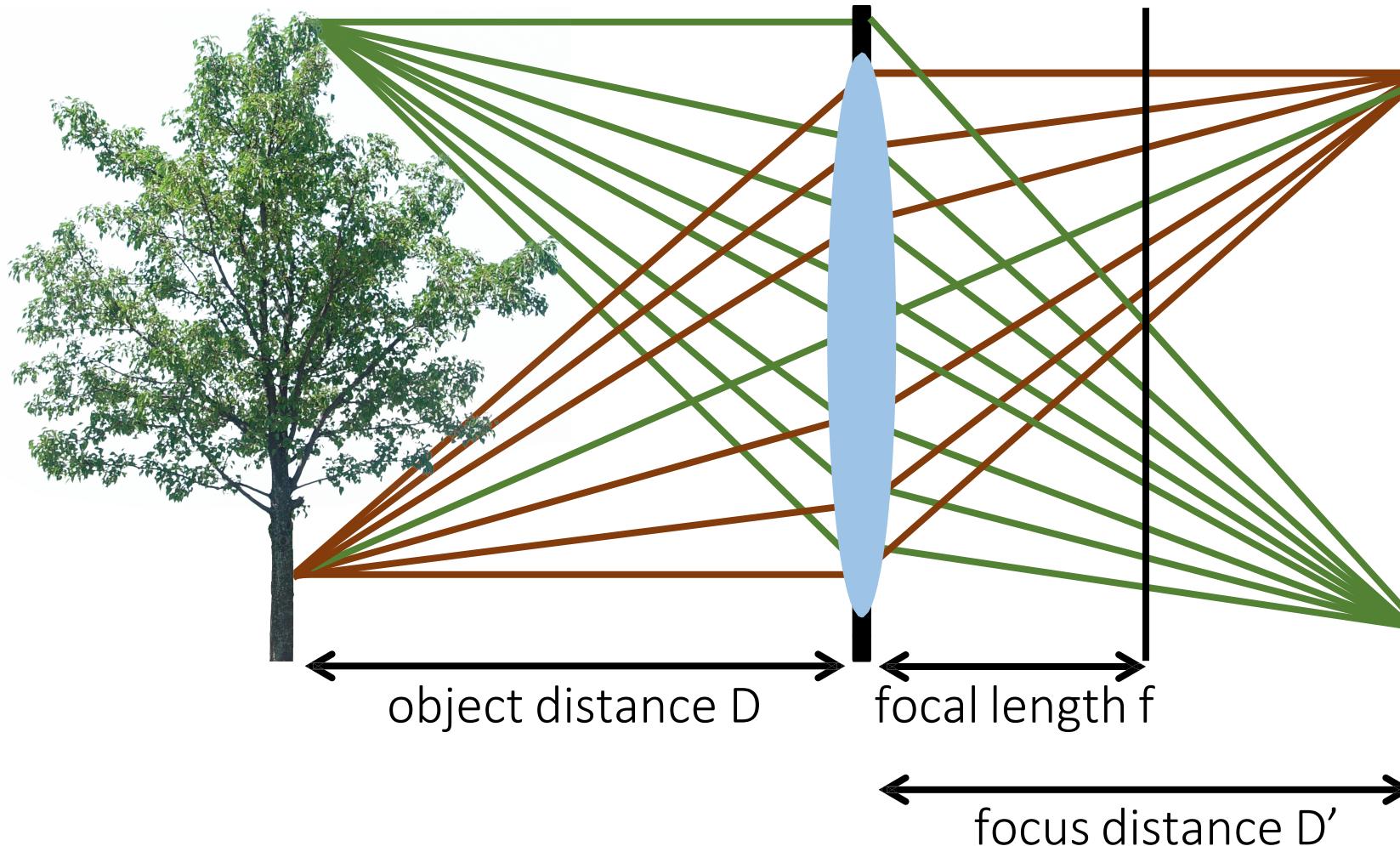
Important difference: focal length

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

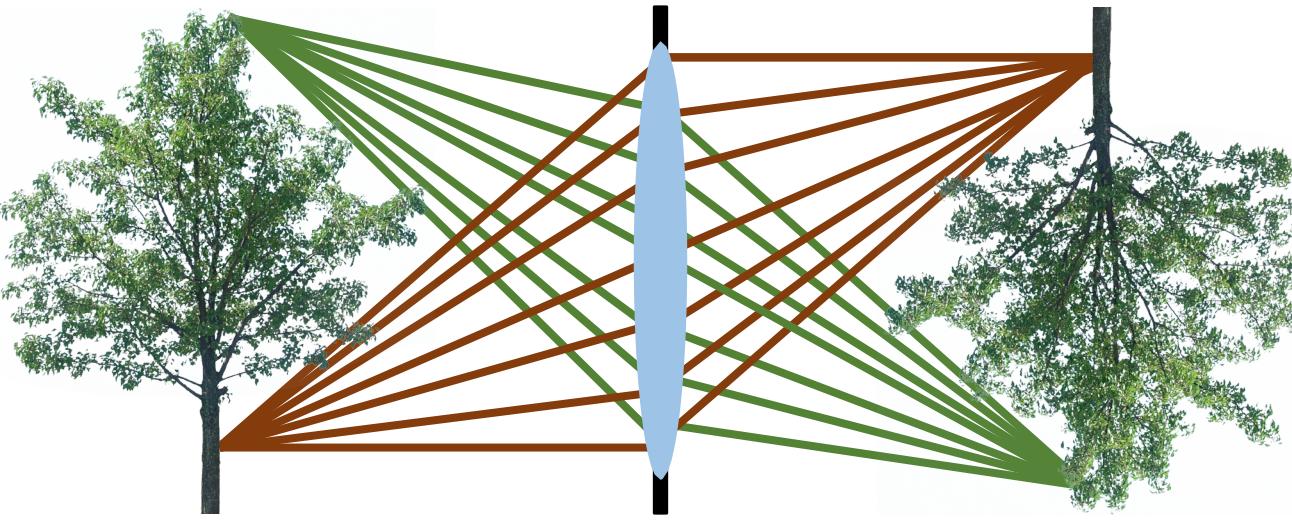


Important difference: focal length

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

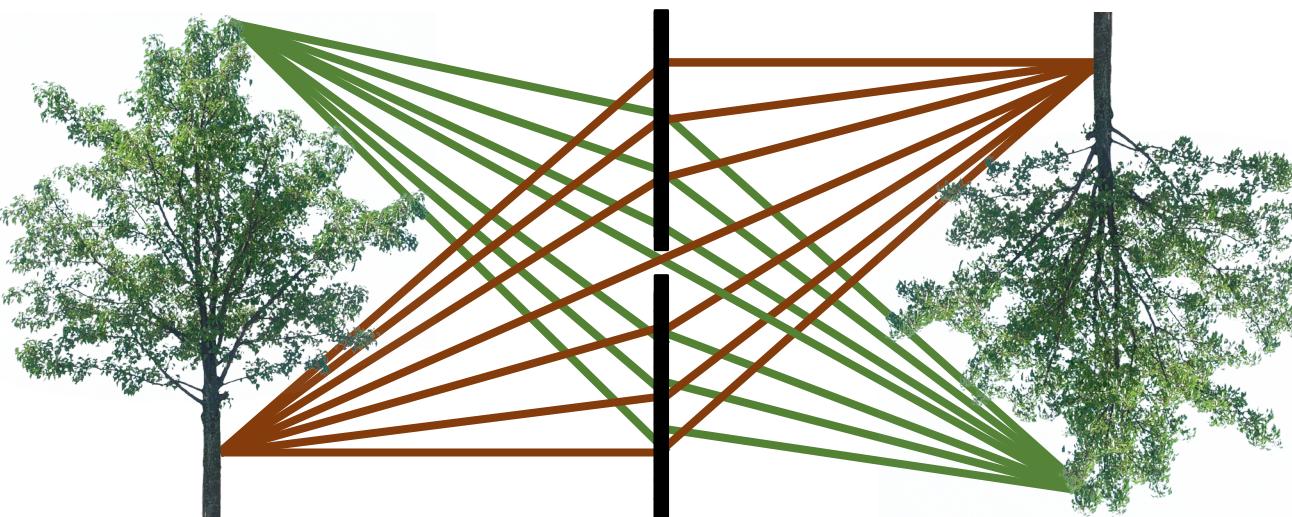


Describing both lens and pinhole cameras



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.

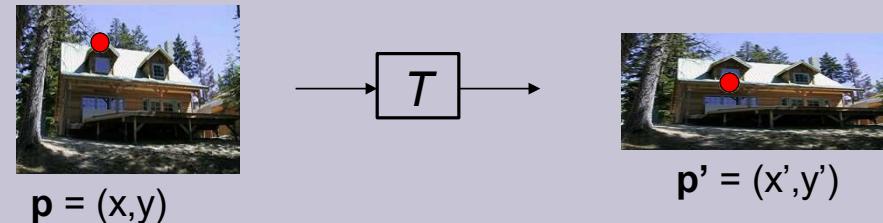
- In this course, we use it to refer to the aperture-sensor distance, as in the pinhole camera case.

Parametric Transformations

Parametric (Global) Transformations

Definition

A transformation T is a coordinate-changing operation $p' = T(p)$



What does it mean that T is global?

- T is the same for any point p

For linear transformations, we can represent T as a matrix

$$p' = \mathbf{T}p$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix}$$

Common Transformations



Original

Transformed



Translation



Rotation



Scaling



Affine



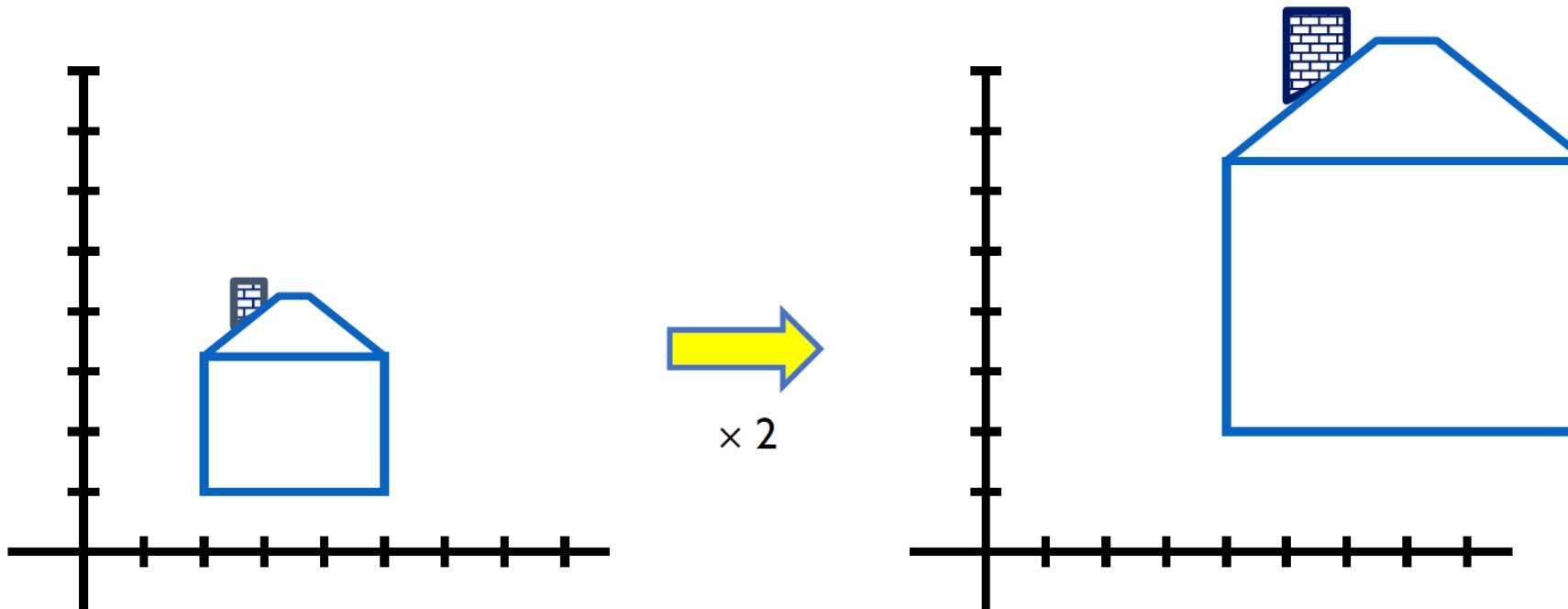
Perspective

Scaling

Definition

Scaling a coordinate means multiplying each of its components by a scalar.

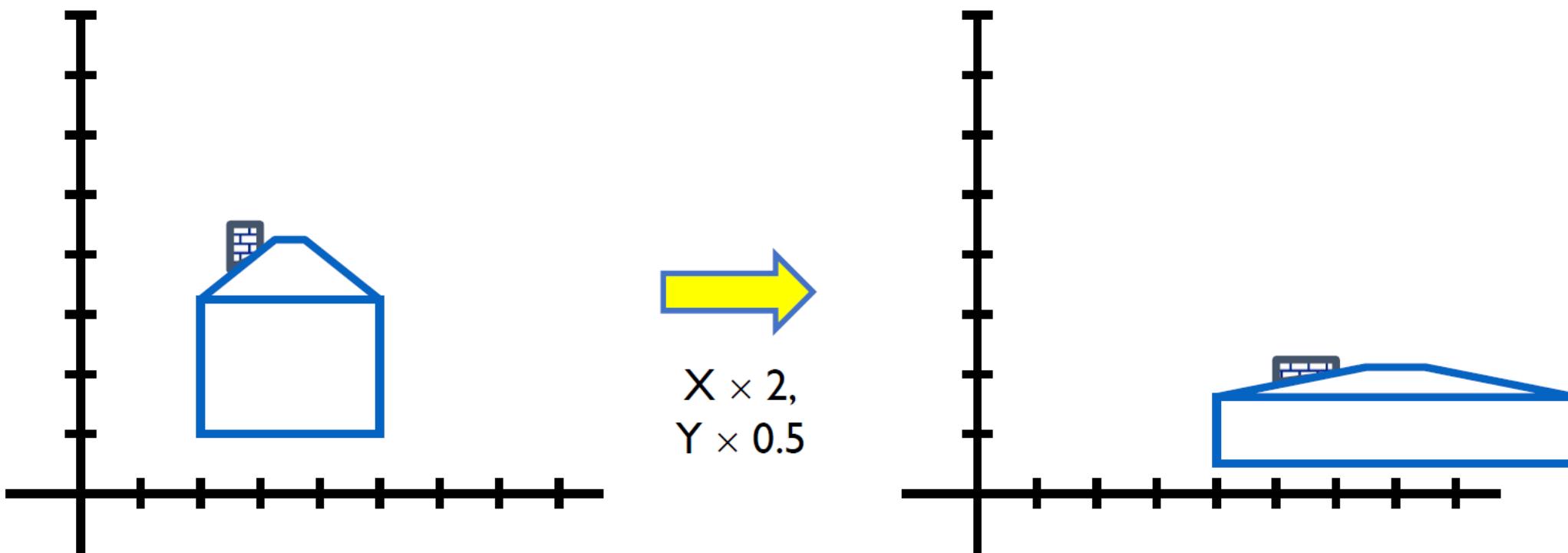
Uniform scaling means this scalar is the same for all components.



Scaling

Definition

Non-uniform scaling: different scalars per component



Scaling

- Scaling operation:

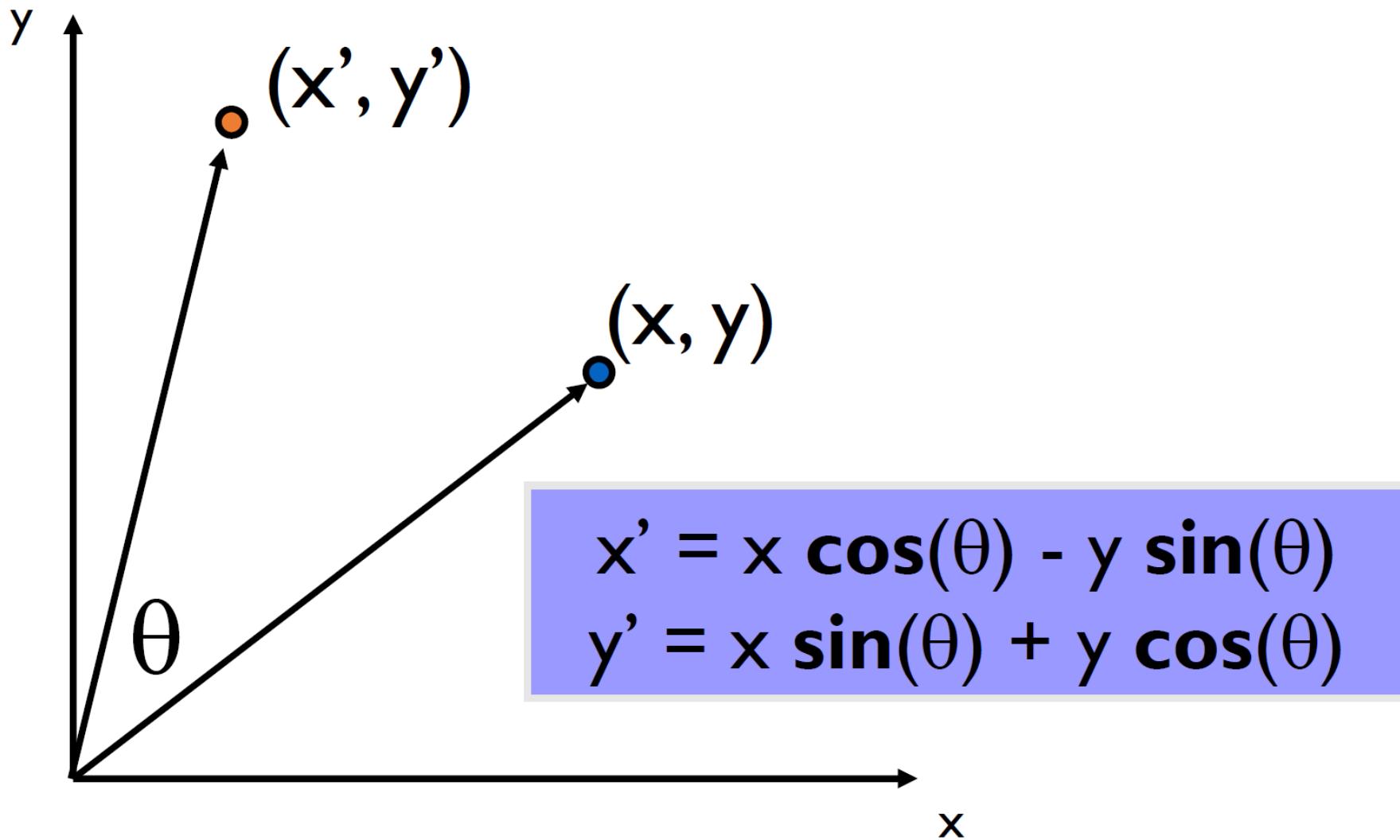
$$x' = ax$$

$$y' = by$$

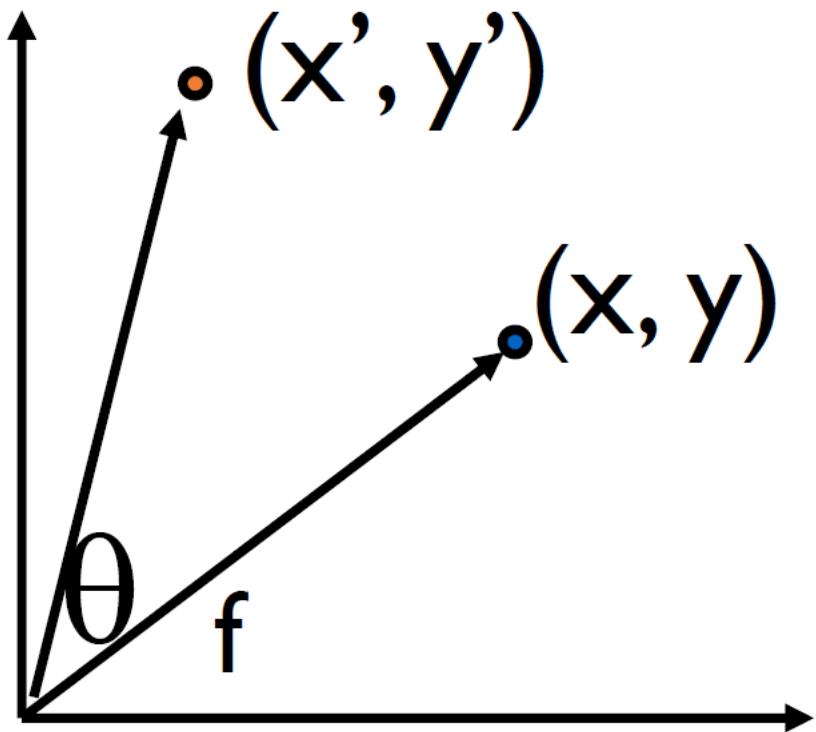
- Or, in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\text{scaling matrix } S} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Rotation



Additional Slide: 2D Rotation



Polar coordinates...

$$x = r \cos(f)$$

$$y = r \sin(f)$$

$$x' = r \cos(f + \theta)$$

$$y' = r \sin(f + \theta)$$

Trig Identity...

$$x' = r \cos(f) \cos(\theta) - r \sin(f) \sin(\theta)$$

$$y' = r \sin(f) \cos(\theta) + r \cos(f) \sin(\theta)$$

Substitute...

$$x' = x \cos(\theta) - y \sin(\theta)$$

$$y' = x \sin(\theta) + y \cos(\theta)$$

2D Rotation

- This is easy to capture in matrix form:

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

- What is the inverse transformation?
 - Rotation by $-\theta$
 - For rotation matrices $R^{-1} = R^T$

Basic 2D Transformations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \alpha_x \\ \alpha_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine



Affine is any combination of translation, scale, rotation, shear

Affine Transformations

- Affine transformations are combinations of
 - Linear transformations, and
 - Translations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

or

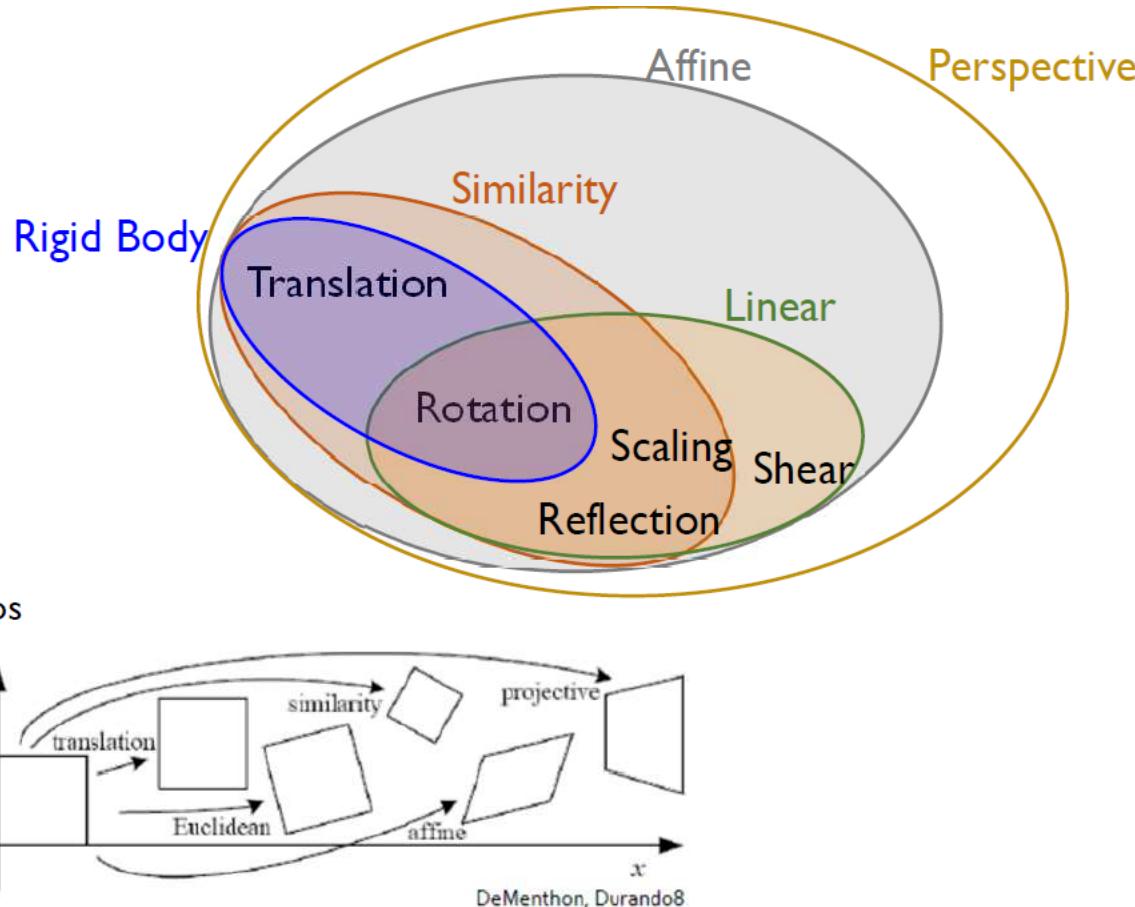
- Properties of affine transformations:
 - Lines map to lines
 - Parallel lines remain parallel
 - Ratios are preserved
 - Closed under composition

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

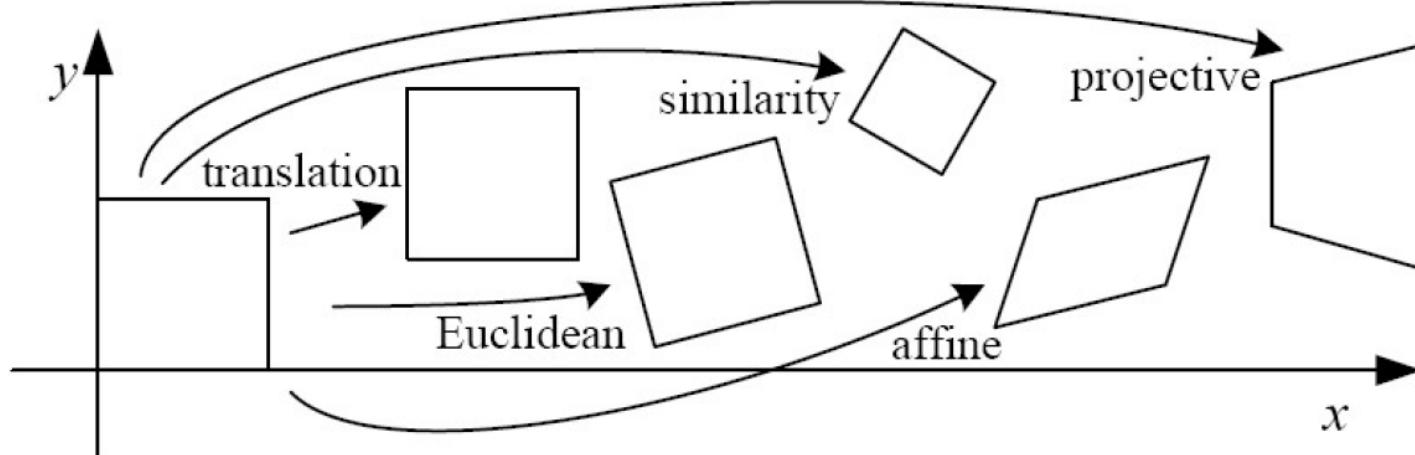
2D Transformations: Properties Preserved

Transformations can be classified based on the properties they preserve:

- **Rigid Body**
 - angles, lengths, areas
- **Similarity**
 - angles, length ratios
- **Linear**
 - linear combination
- **Affine**
 - parallel lines, length ratios, area ratios
- **Perspective**
 - collinearity, cross-ratio
- ...



2D Transformations



Szeliski 2.1

Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	2	orientation + ...	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	3	lengths + ...	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	4	angles + ...	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2 \times 3}$	6	parallelism + ...	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3 \times 3}$	8	straight lines	

'Homography'

Cameras are Projective Transformation Machines

Projective Transformation

3D World

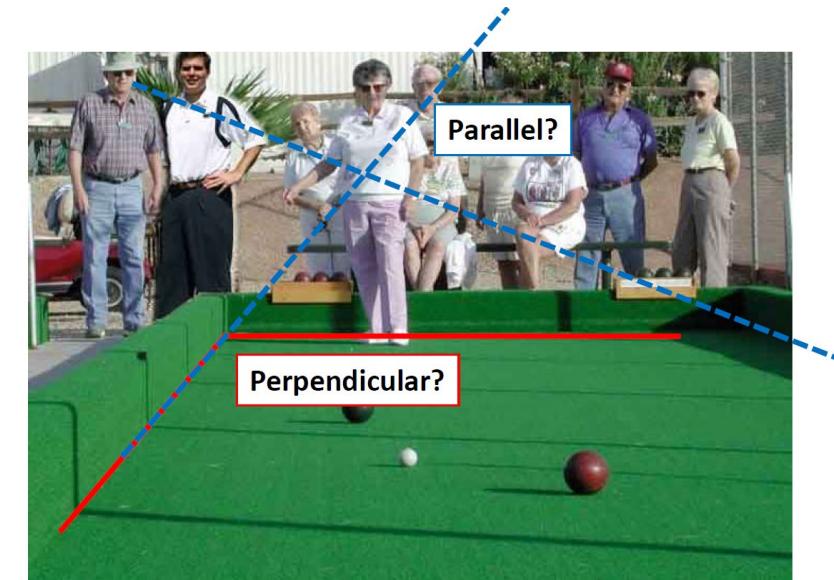


2D Image



Projective Transformations

- Projective transformations are combos of
 - Affine transformations, and
 - Projective warps
- Properties of projective transformations:
 - Lines map to lines
 - Parallel lines do not necessarily remain parallel
 - Ratios are not preserved
 - Closed under composition
 - Models change of basis
 - Projective matrix is defined up to a scale (8 degrees of freedom)



$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Homogeneous Coordinates

Homogeneous Coordinates

- Introduced by August Ferdinand Möbius
- An alternative coordinate system for projective geometry
 - Compare to Cartesian coordinates
- Simplify formulas
 - Make translation a linear transformation
 - Facilitate matrix-vector formulas



Wikipedia

Homogeneous Coordinates

Converting to homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D (image) coordinates

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

3D (scene) coordinates

Converting from homogeneous coordinates

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

2D (image) coordinates

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

3D (scene) coordinates

Homogeneous Coordinates

Scale invariance in projection space

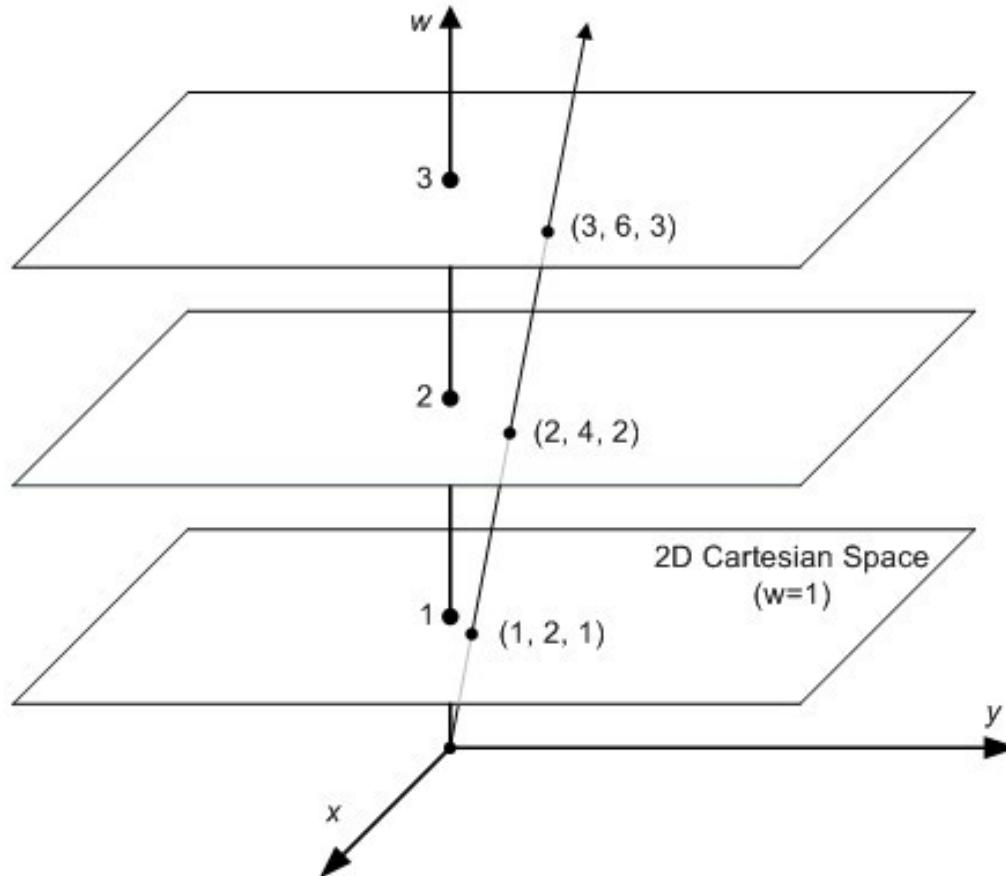
$$k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

Homogeneous Coordinates Cartesian Coordinates

e.g., we can uniformly scale the projective space, and it will still produce the same image \Rightarrow scale ambiguity

Homogeneous Coordinates

- Projective
- Point becomes a line



Song Ho Ahn

To homogeneous

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

From homogeneous

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

Basic 2D Transformations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \alpha_x \\ \alpha_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine



Affine is any combination of translation, scale, rotation, shear

Camera Matrix

Fundamental Equations of Computer Vision

1. Image Filtering

$$h[m,n] = \sum_{k,l} f[k,l] I[m+k, n+l]$$

2. Optical Flow

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

3. Camera Geometry

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \ \mathbf{t}] \mathbf{X} \quad x^T F x' = 0$$

4. Machine Learning

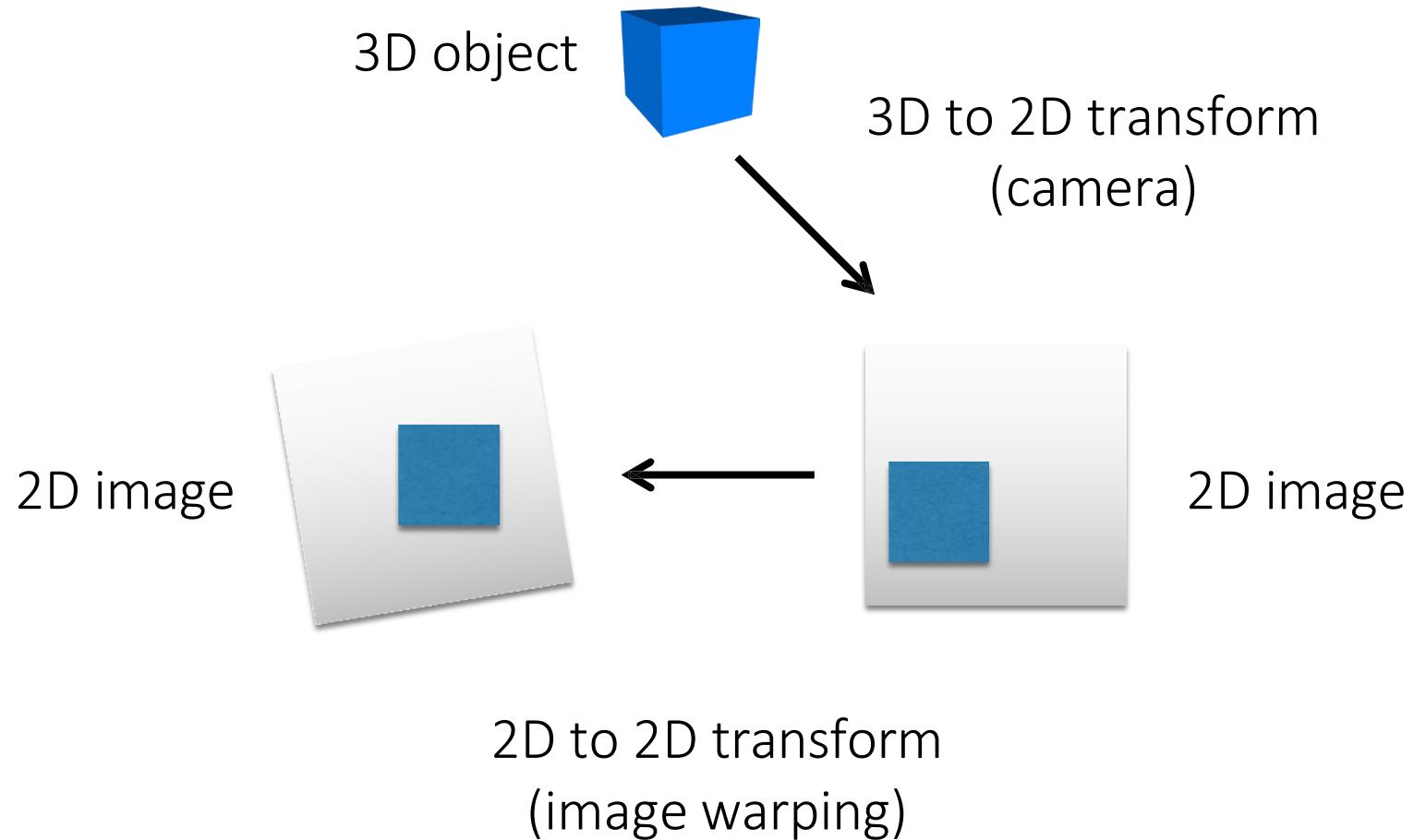
$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2. \quad y = \varphi(\sum_{i=1}^n w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

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:

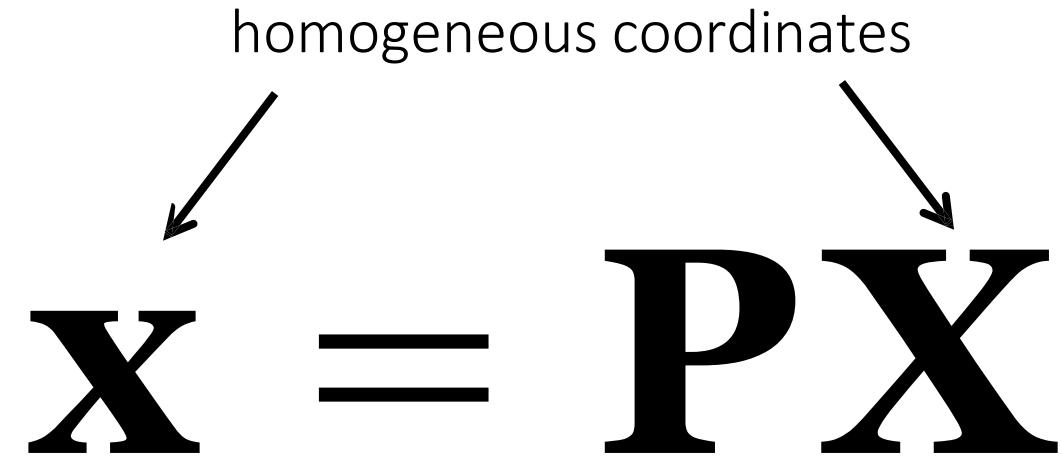
the 3D world

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

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

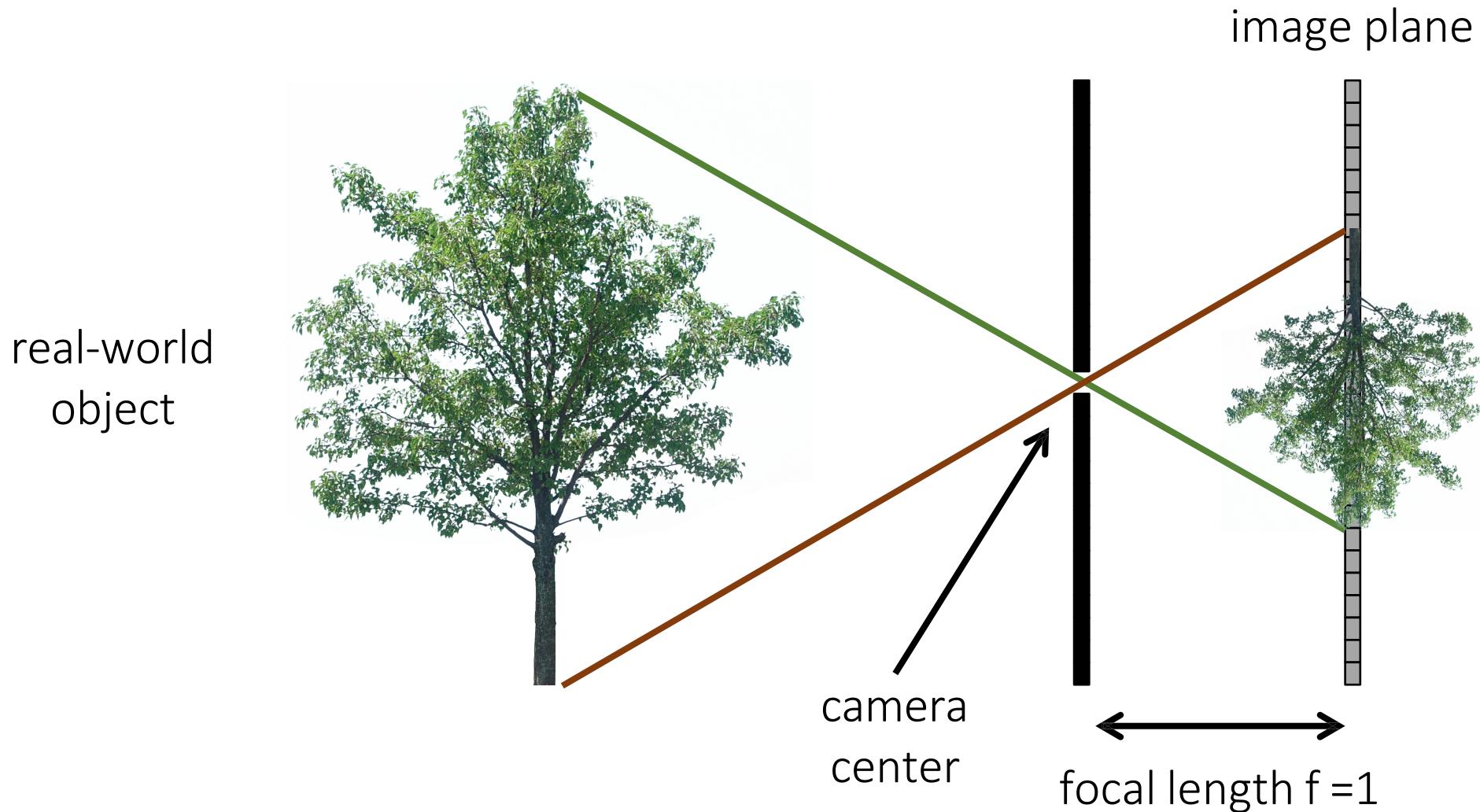
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous
image coordinates
 3×1

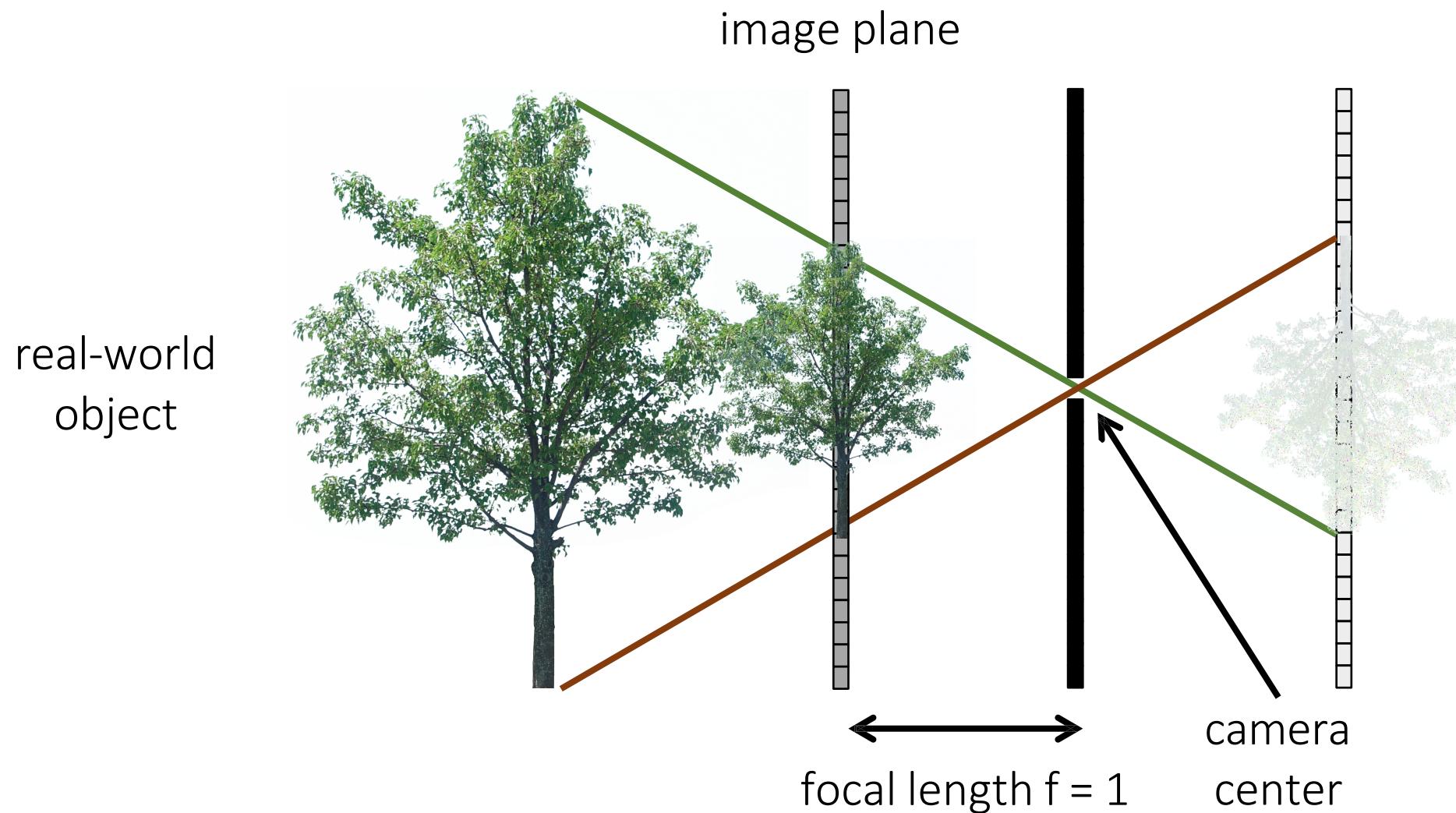
camera
matrix
 3×4

homogeneous
world coordinates
 4×1

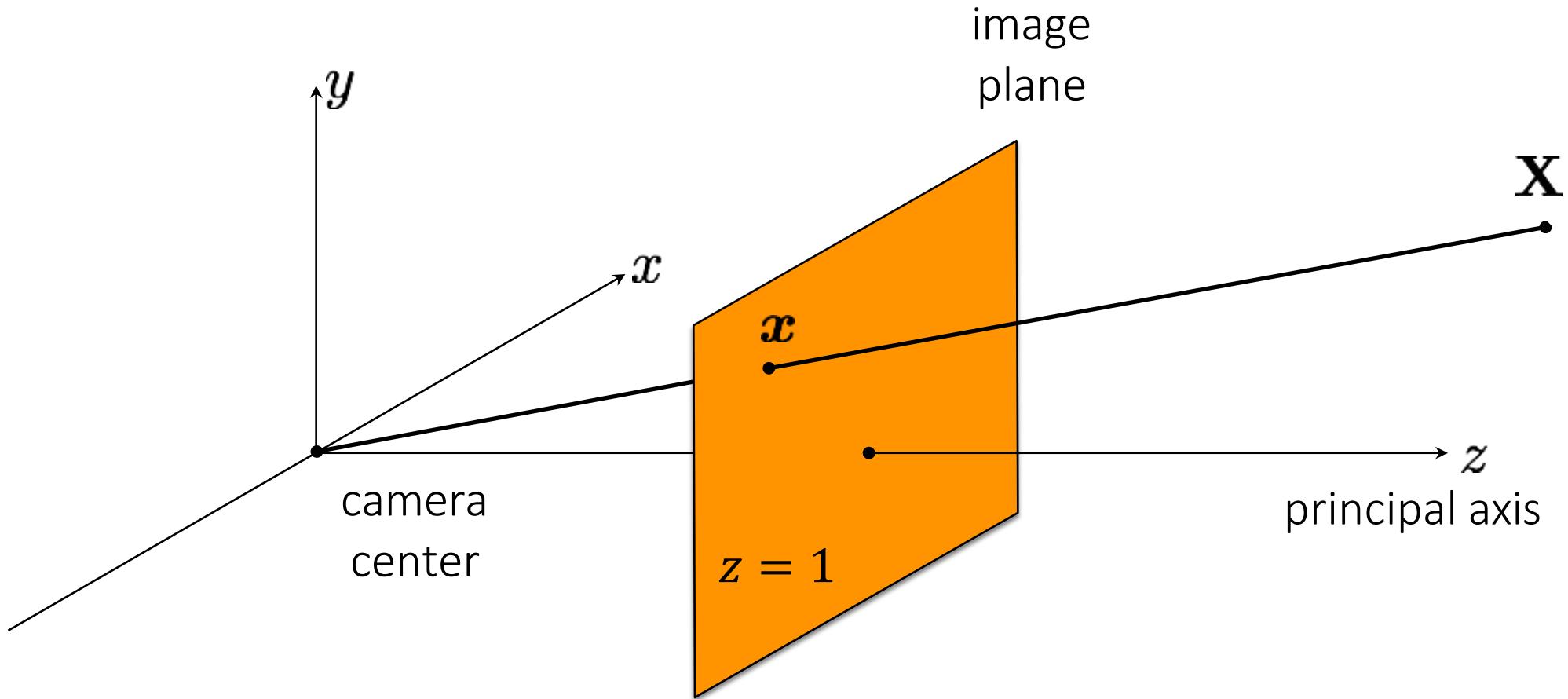
The pinhole camera



The (rearranged) pinhole camera

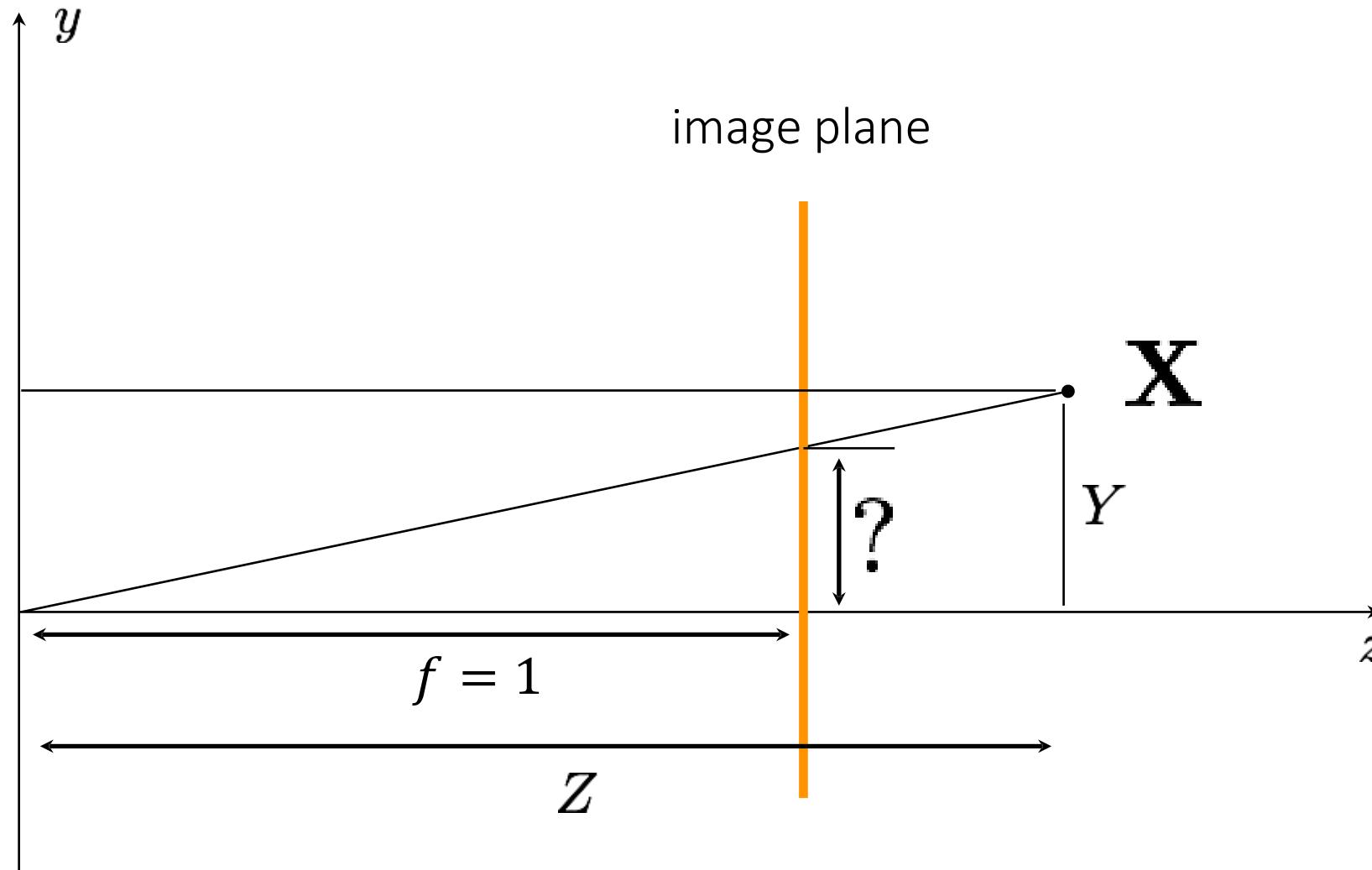


The (rearranged) pinhole camera



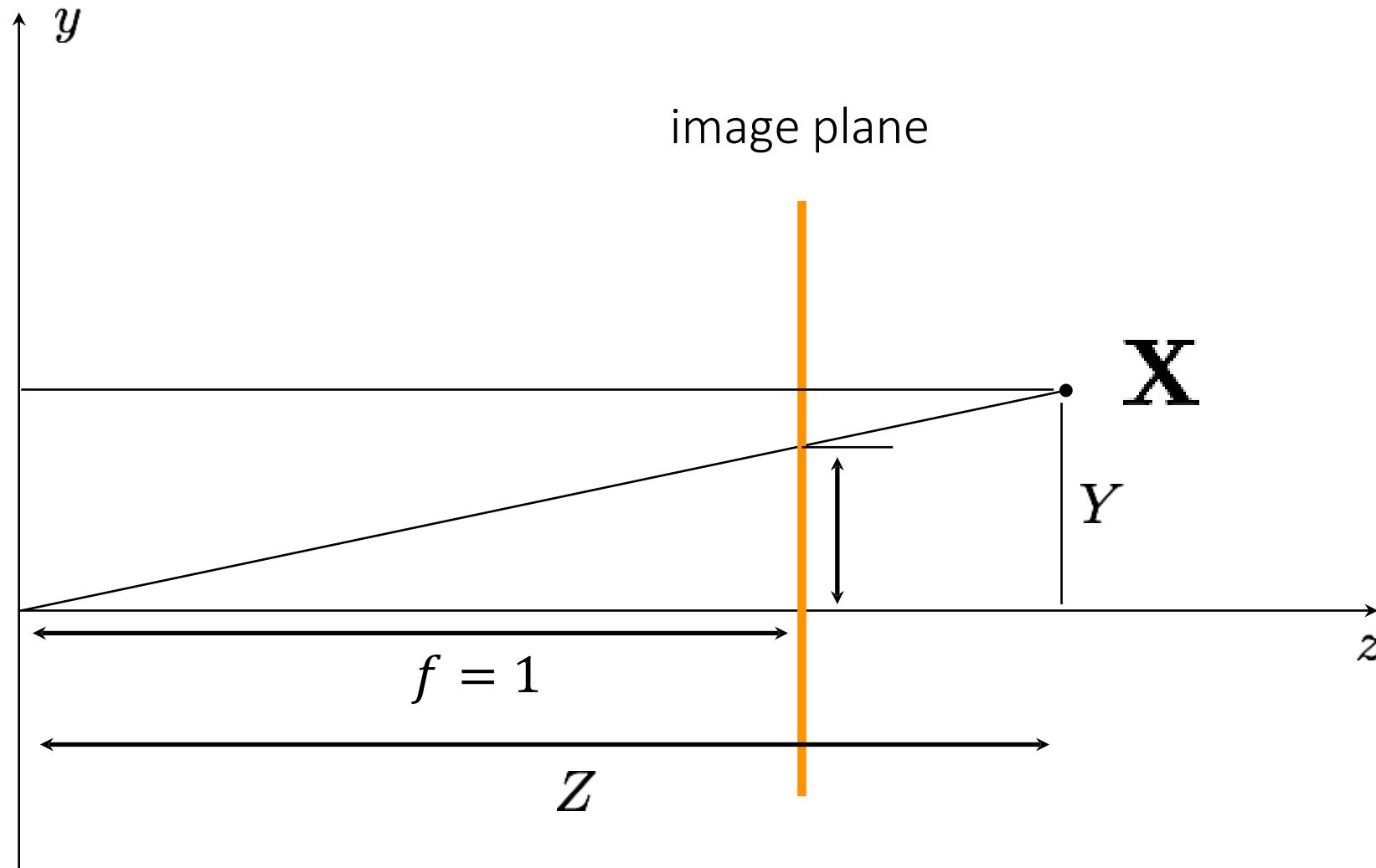
What is the equation for image coordinate x in terms of X ?

The 2D view of the (rearranged) pinhole camera



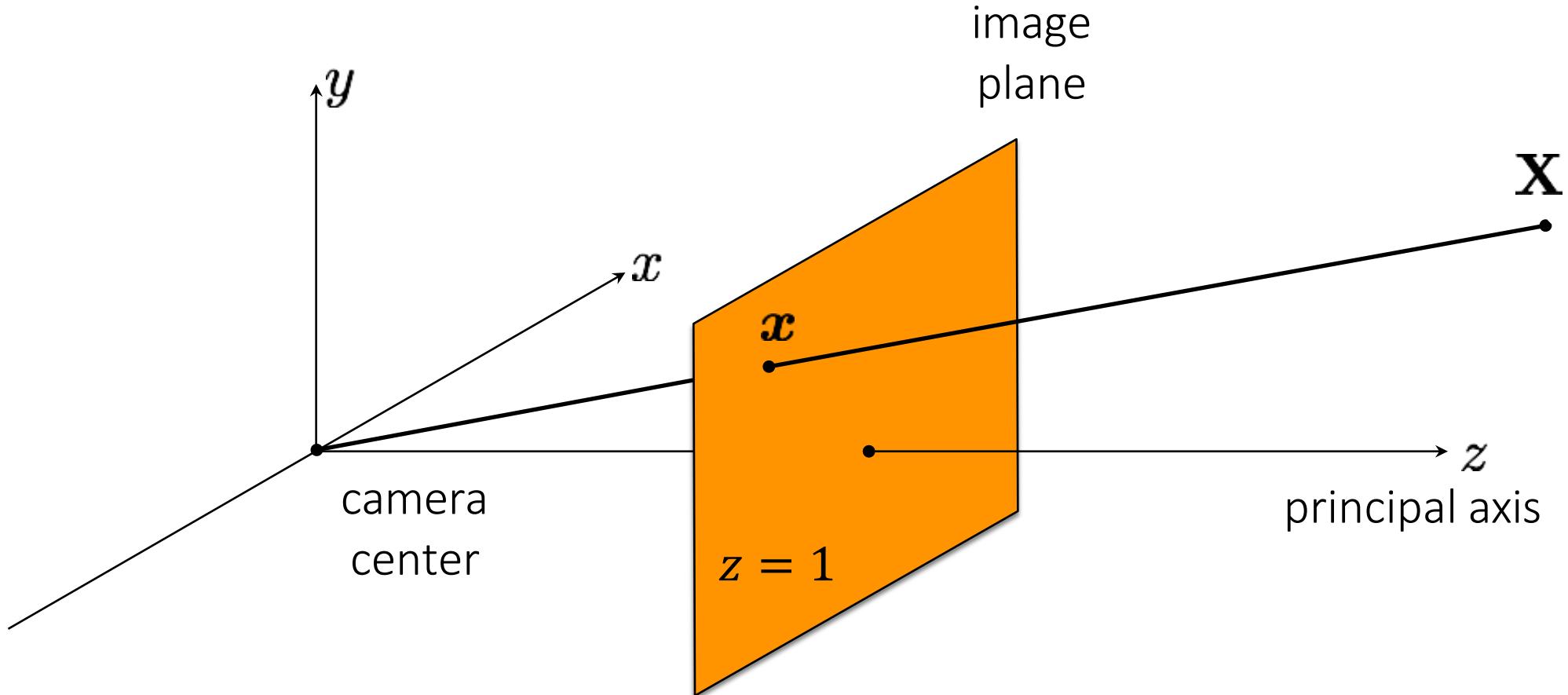
What is the equation for image coordinate x in terms of X ?

The 2D view of the (rearranged) pinhole camera



$$[X \quad Y \quad Z] \rightarrow [X/Z \quad Y/Z]$$

The (rearranged) pinhole camera



What is the camera matrix P for a pinhole camera?

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

The pinhole camera matrix

Relationship from similar triangles:

$$[X \quad Y \quad Z]^T \rightarrow [X/Z \quad Y/Z]$$

General camera model in homogeneous coordinates:

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

What does the pinhole camera projection look like?

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

The pinhole camera matrix

Relationship from similar triangles:

$$[X \quad Y \quad Z]^T \rightarrow [X/Z \quad Y/Z]$$

General camera model in homogeneous coordinates:

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

What does the pinhole camera projection look like?

The perspective
projection matrix

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

The pinhole camera matrix

Relationship from similar triangles:

$$[X \ Y \ Z]^T \rightarrow [X/Z \ Y/Z]$$

General camera model in homogeneous coordinates:

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

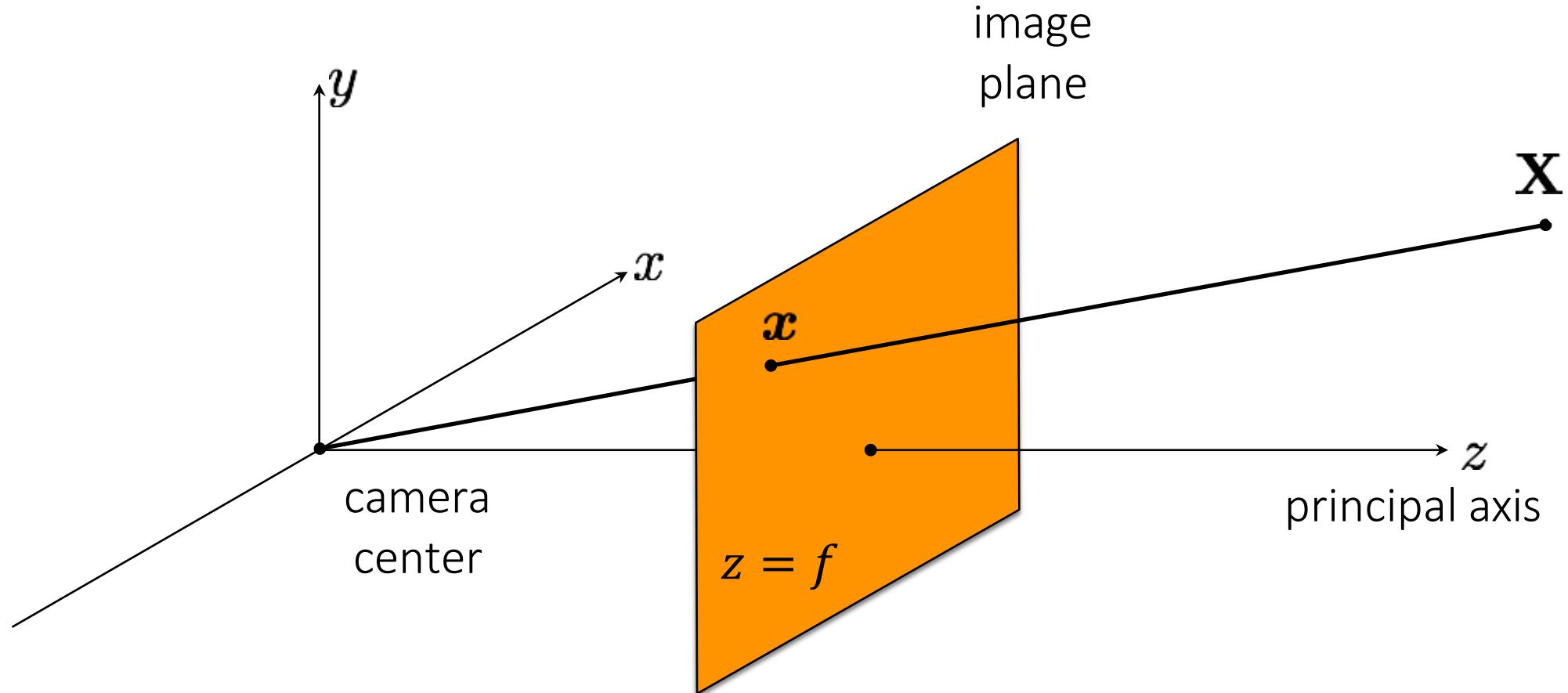
What does the pinhole camera projection look like?

The perspective
projection matrix

$$\mathbf{P} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right] = \mathbf{I} \quad | \quad \mathbf{0}$$

alternative way to write
the same thing

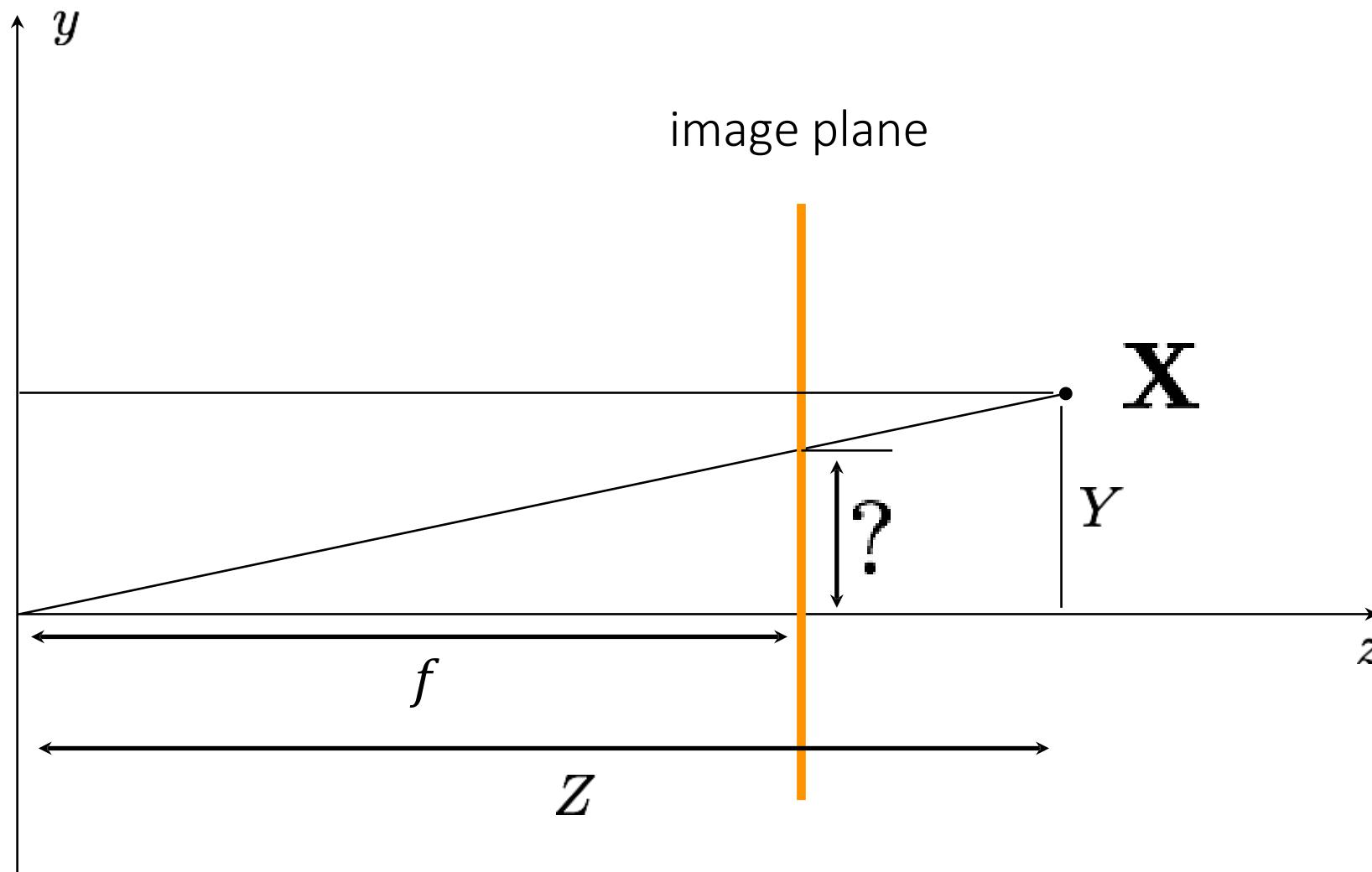
More general case: arbitrary focal length



What is the camera matrix P for a pinhole camera?

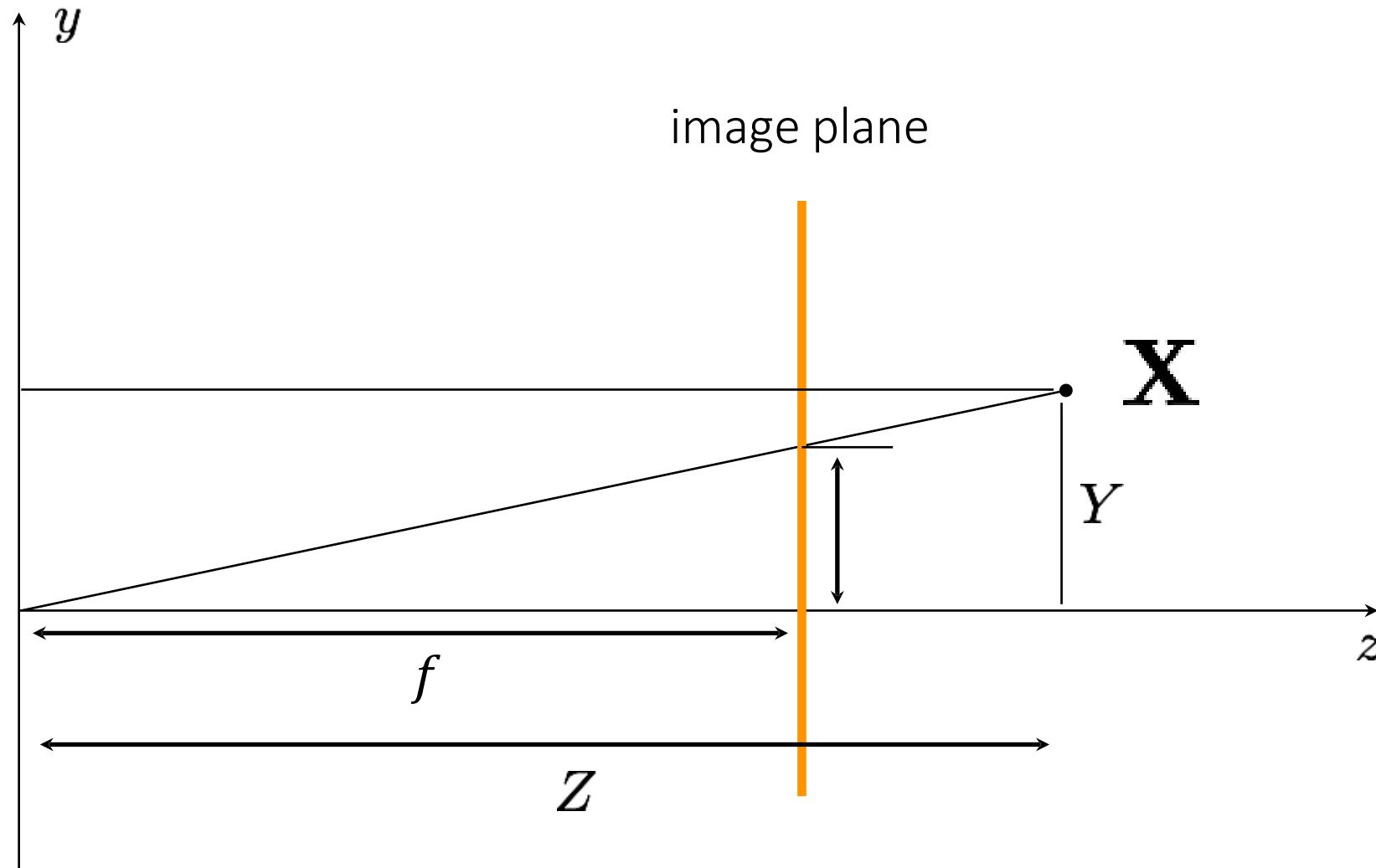
$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

More general (2D) case: arbitrary focal length



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

More general (2D) case: arbitrary focal length



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

The pinhole camera matrix for arbitrary focal length

Relationship from similar triangles:

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

General camera model *in homogeneous coordinates*:

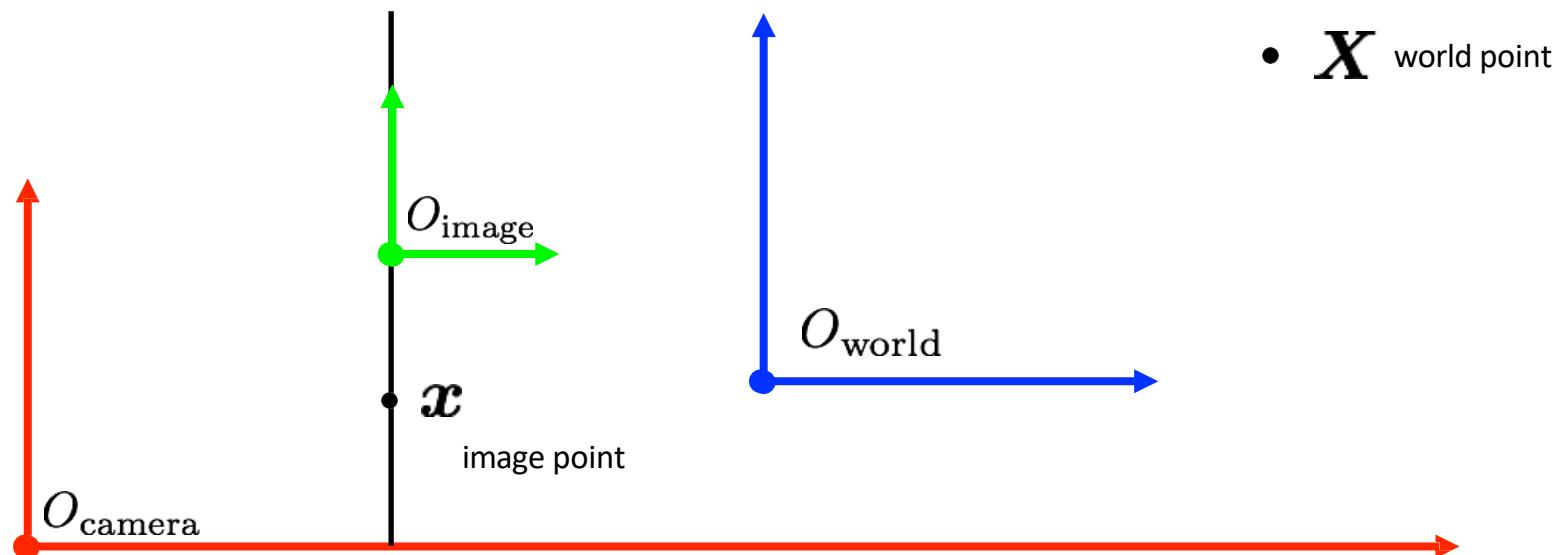
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

What does the pinhole camera projection look like?

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

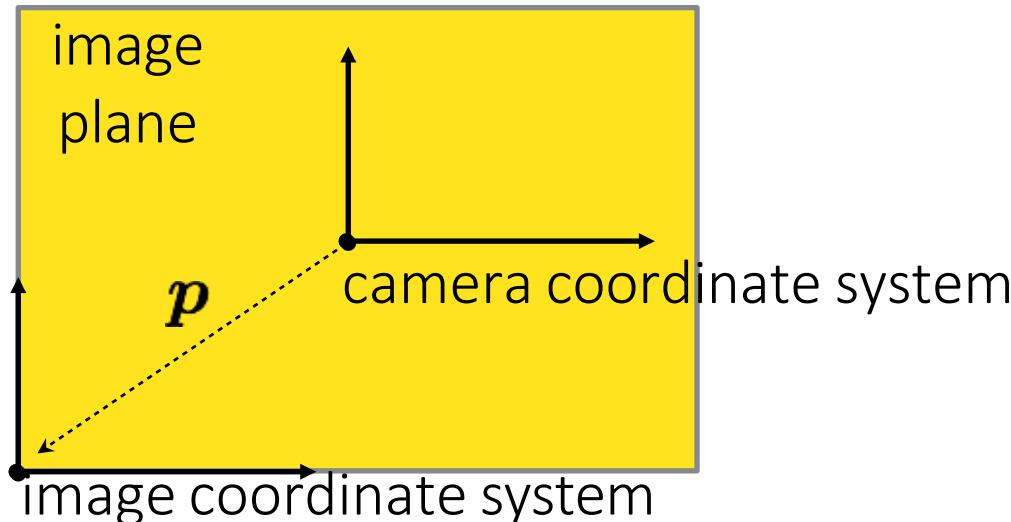
Generalizing the camera matrix

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



Generalizing the camera matrix

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

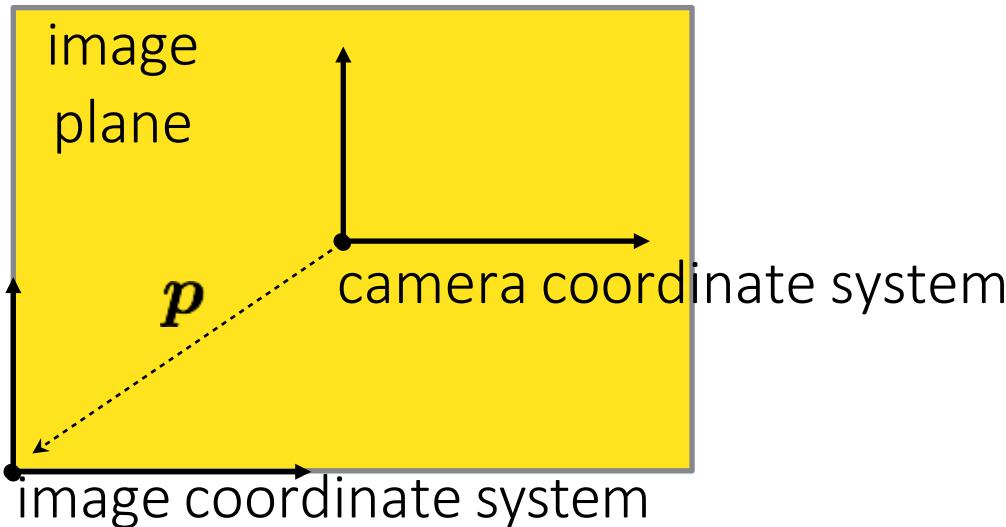


How does the camera matrix change?

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

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

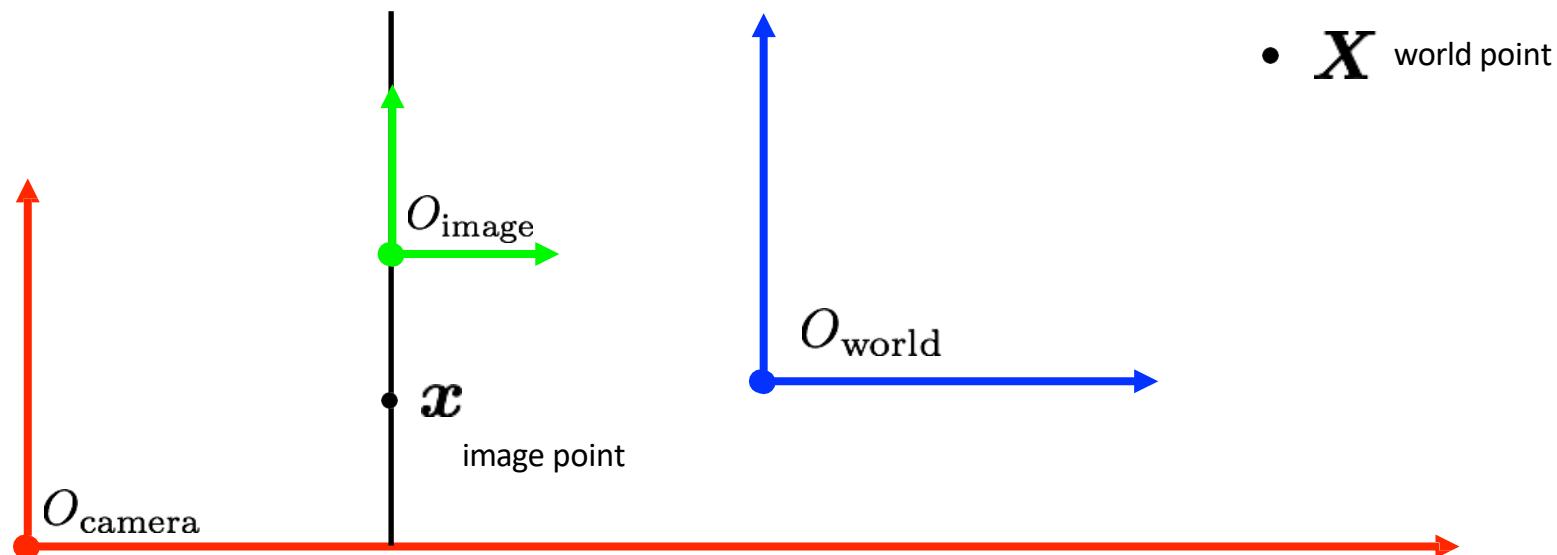
(homogeneous) perspective projection
from 3D to 2D, assuming image plane at
 $z = 1$ and shared camera/image origin

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

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

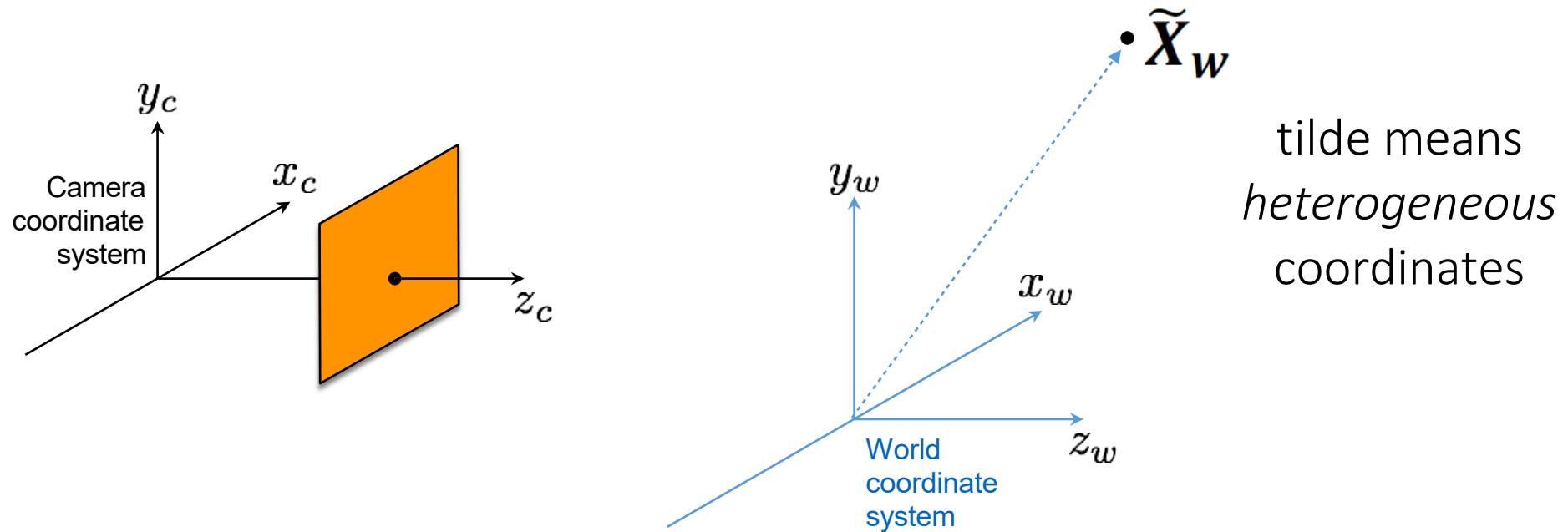
Generalizing the camera matrix

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

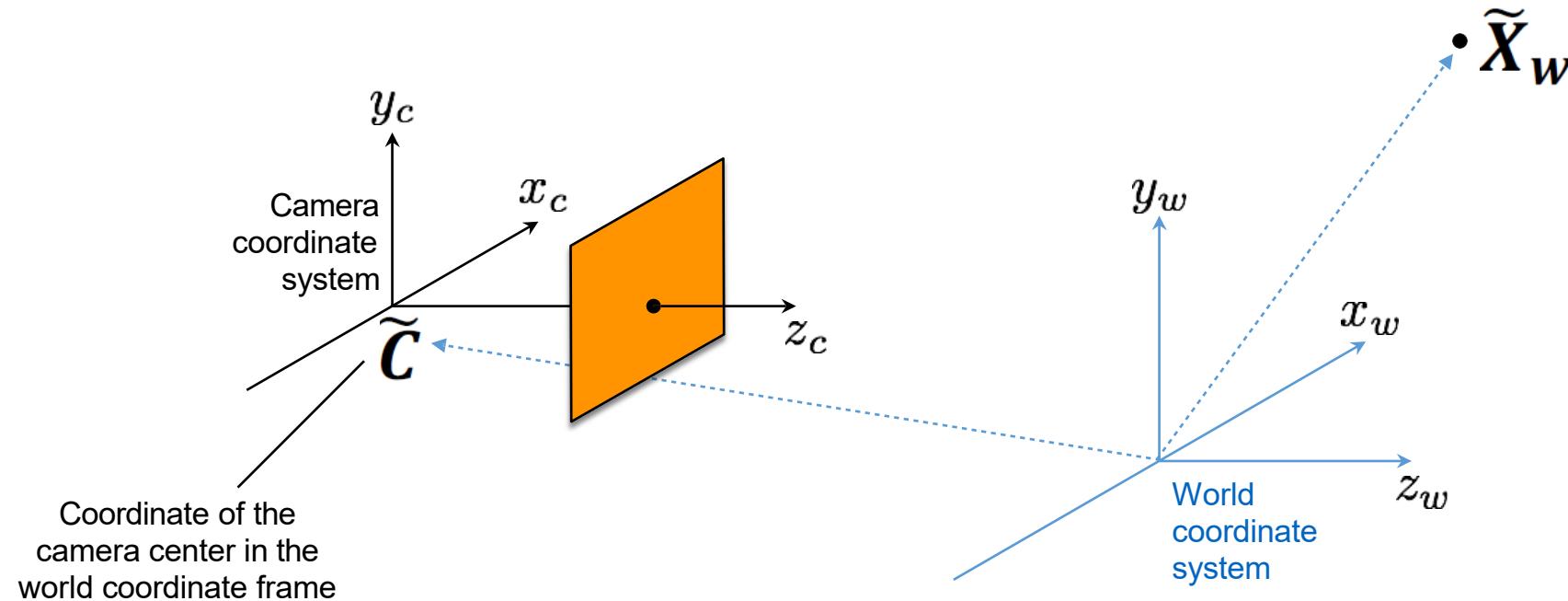


We need to know the transformations between them.

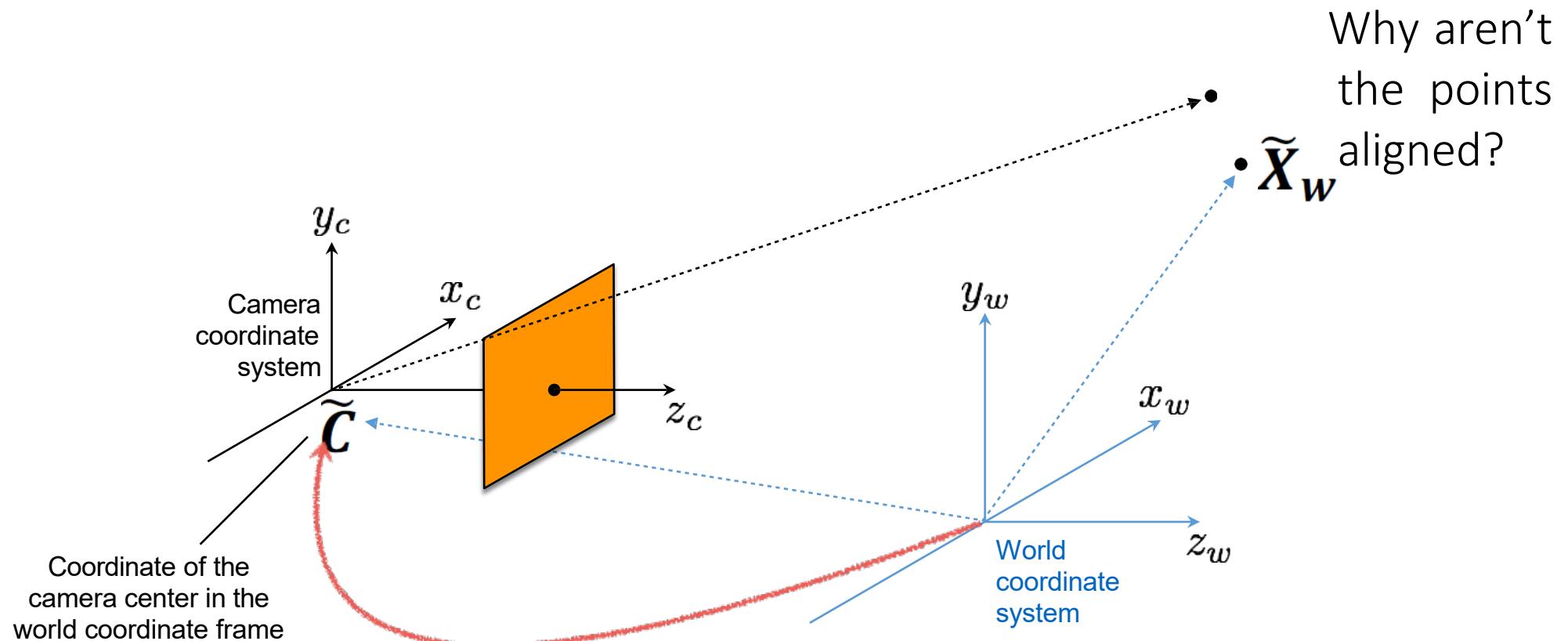
World-to-camera coordinate system transformation



World-to-camera coordinate system transformation



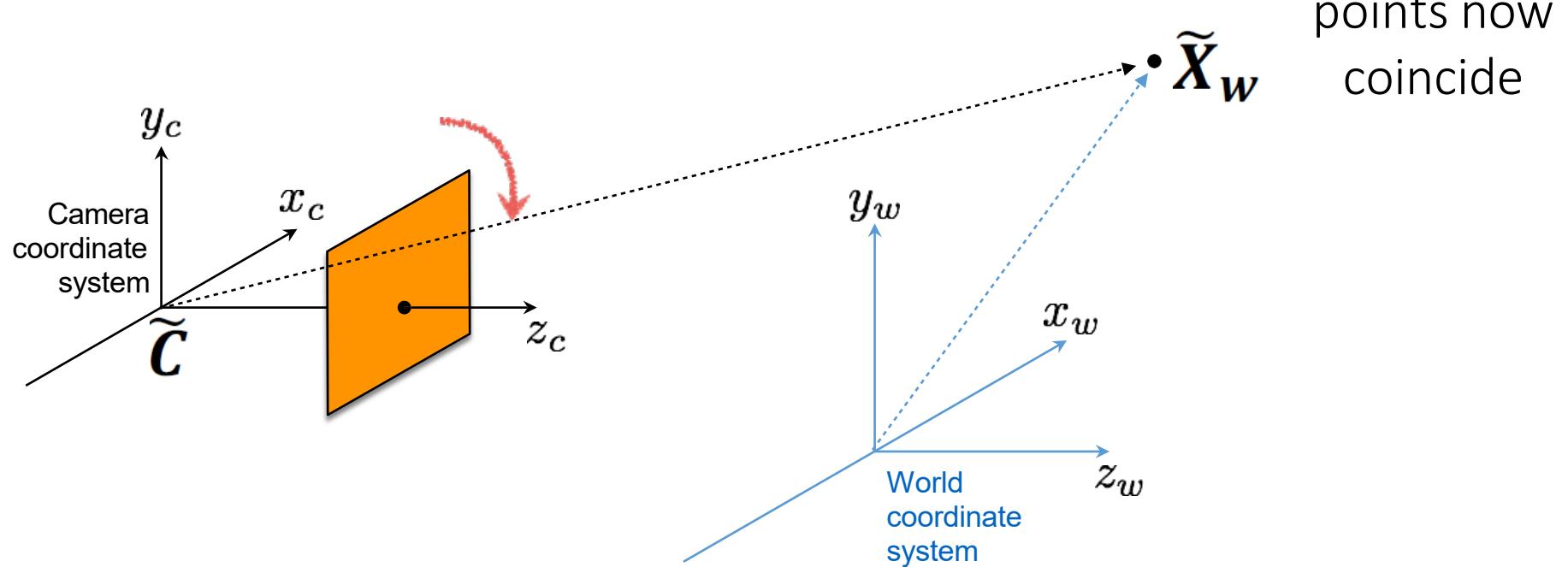
World-to-camera coordinate system transformation



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

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:

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

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:

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

intrinsic parameters (3×3):
correspond to camera
internals (image-to-image
transformation)

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

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

Putting it all together

We can write everything into a single projection:

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

The camera matrix now looks like:

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

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?

$$P = K R [I] - C$$

The diagram consists of a mathematical equation $P = K R [I] - C$. Below the equation, there are four question marks: one under K , one under R , one under the bracketed I , and one under C . Four arrows originate from these question marks and point directly to their respective terms in the equation above.

Recap

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

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

The diagram illustrates the components of the camera matrix \mathbf{P} . The term \mathbf{K} is labeled as a 3×3 matrix representing intrinsics. The term \mathbf{R} is labeled with a question mark, indicating its size or meaning is unknown. The term $[\mathbf{I}]$ is also labeled with a question mark. The term \mathbf{C} is also labeled with a question mark.

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.

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

More general camera matrices

CCD camera: pixels may not be square.

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

How many degrees of freedom?

More general camera matrices

CCD camera: pixels may not be square.

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

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.

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

How many degrees of freedom?

11 DOF