The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

Spring 2024

INTRODUCTION TO COMPUTER VISION

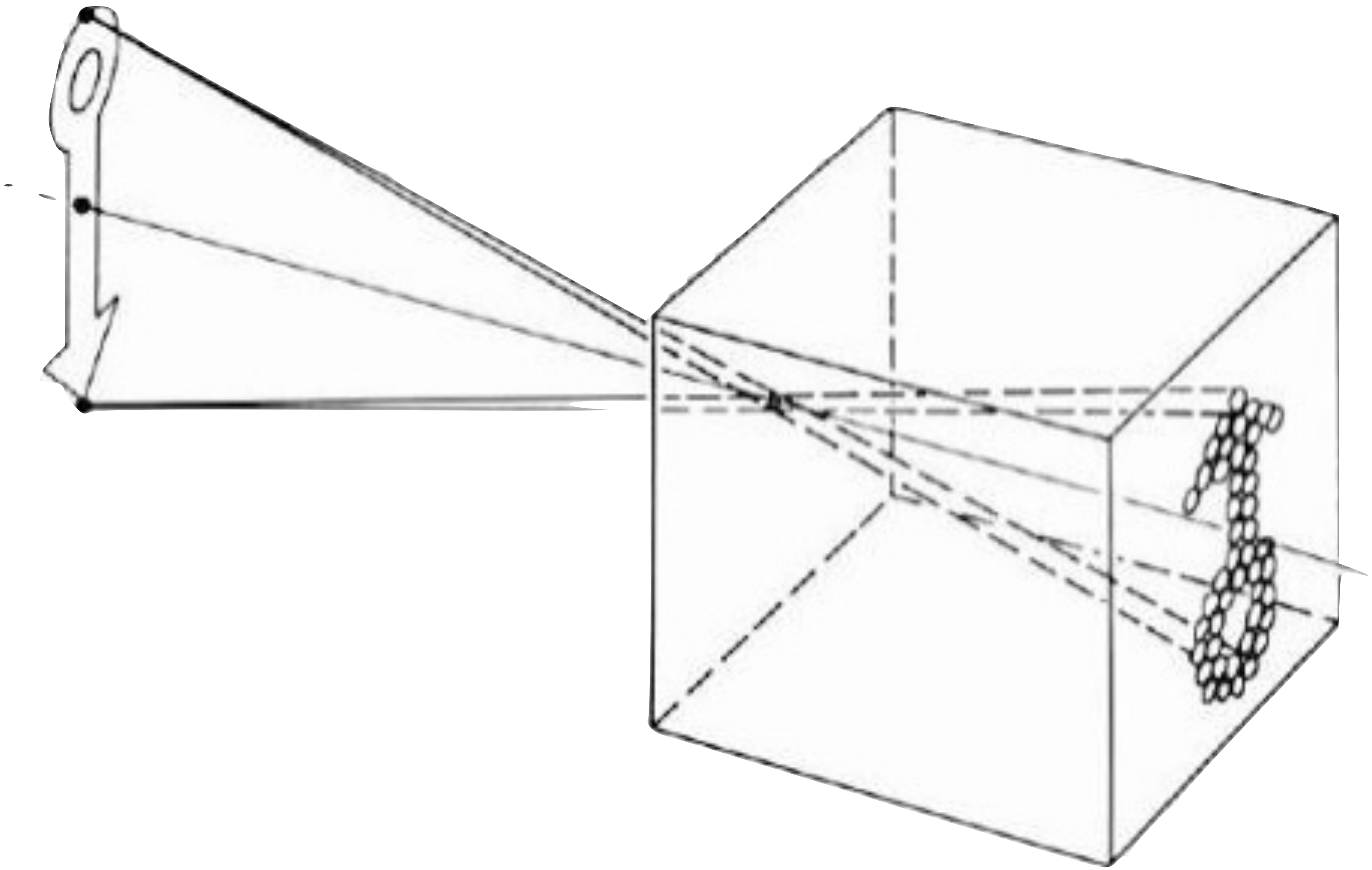
Atlas Wang
Associate Professor, The University of Texas at Austin

Visual Informatics Group@UT Austin
https://vita-group.github.io/

Camera Model

Pinhole and Lens

Pinhole camera a.k.a. camera obscura

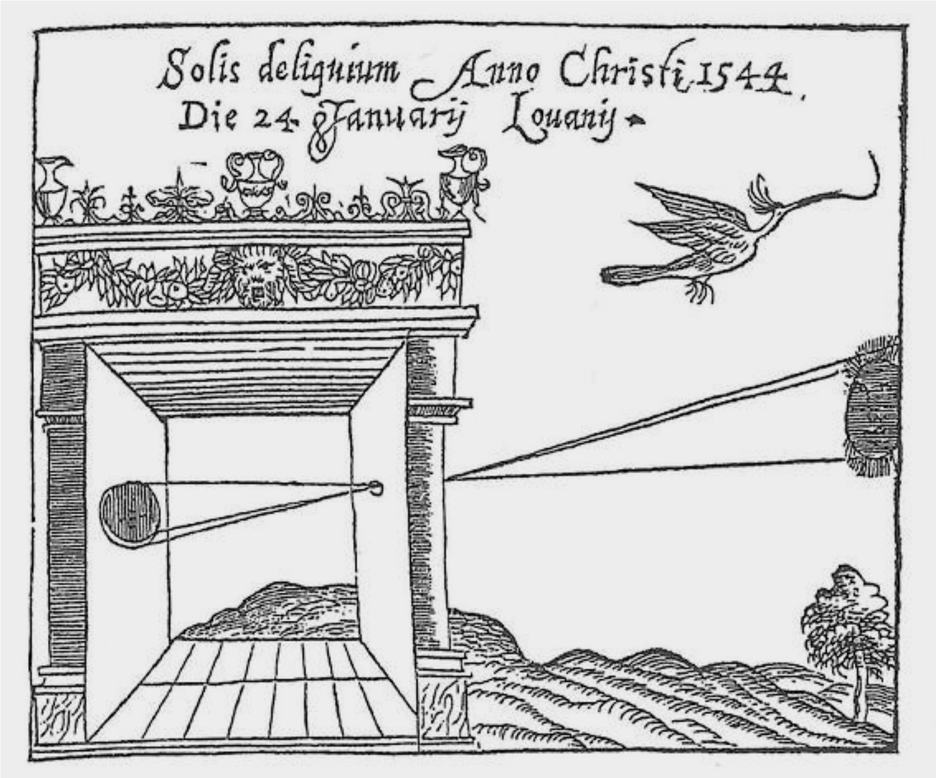


Pinhole camera a.k.a. camera obscura

First mention …

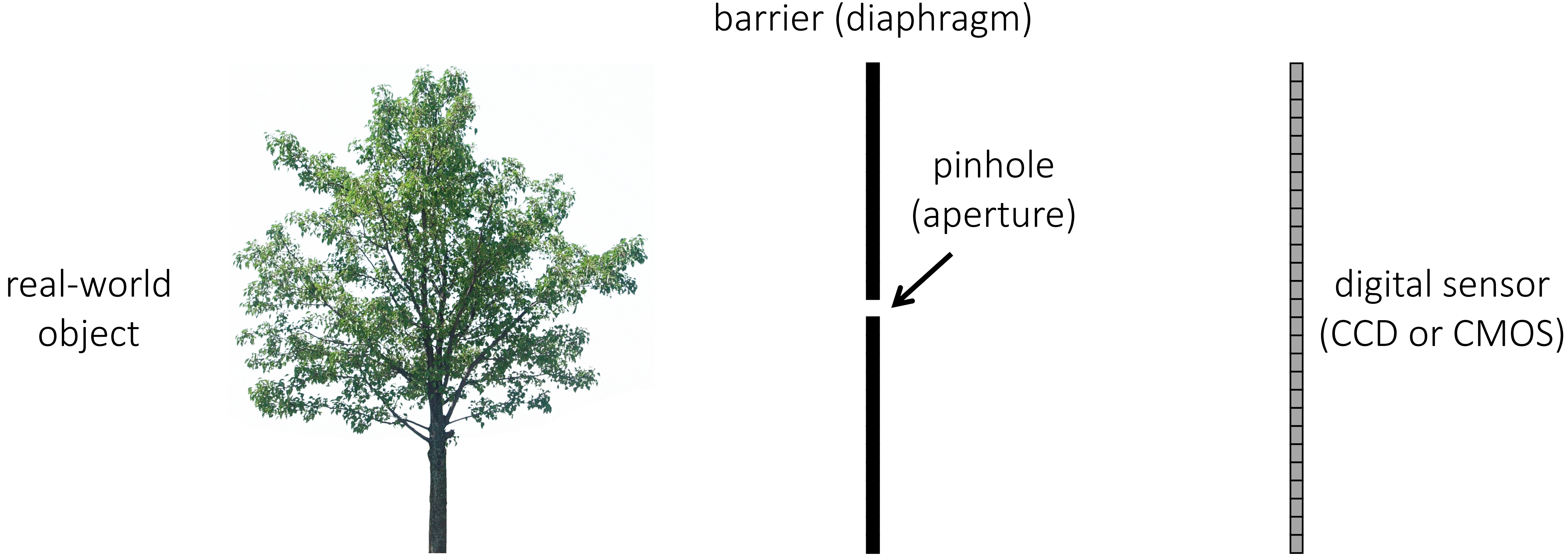
Chinese philosopher Mozi
(470 to 390 BC)

First camera …

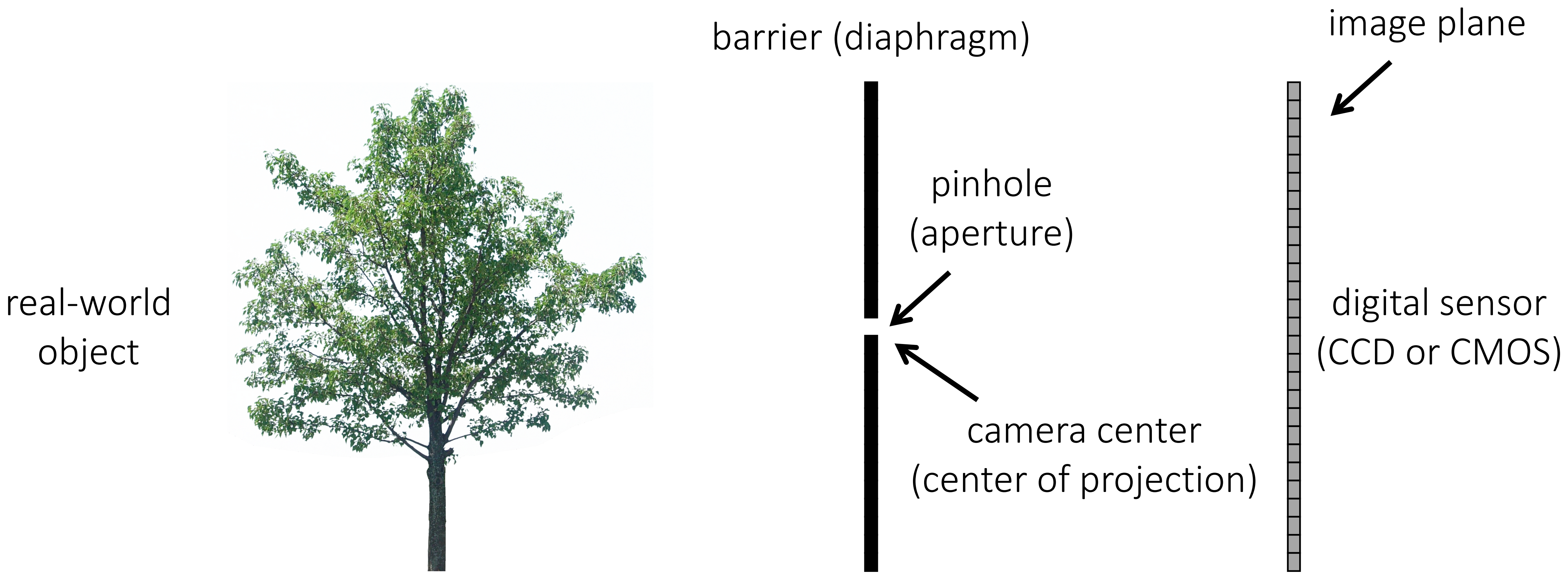


Greek philosopher Aristotle
(384 to 322 BC)

Pinhole camera terms



Pinhole camera terms



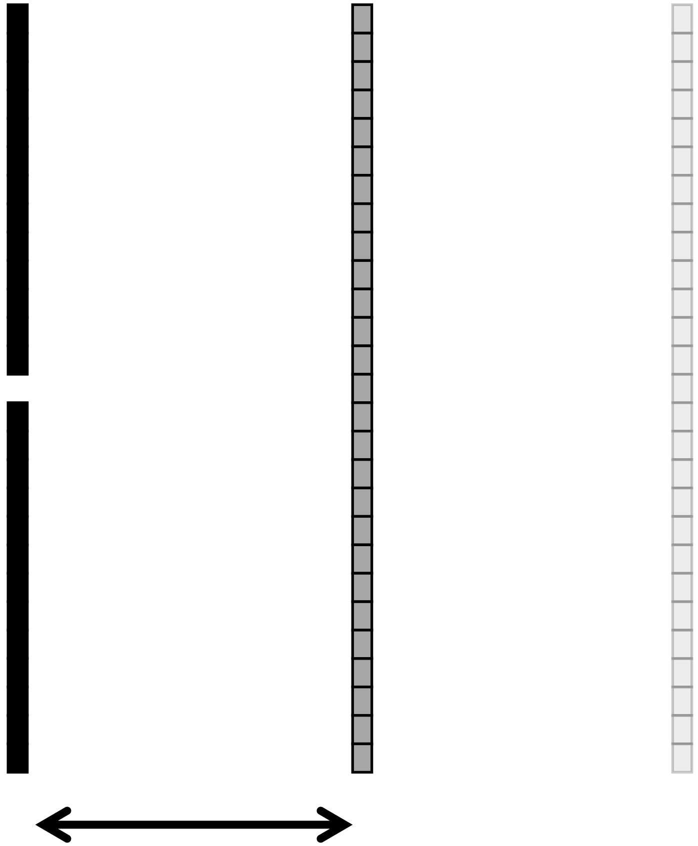
Focal length

real-world
object

Focal length

What happens as we change the focal length?

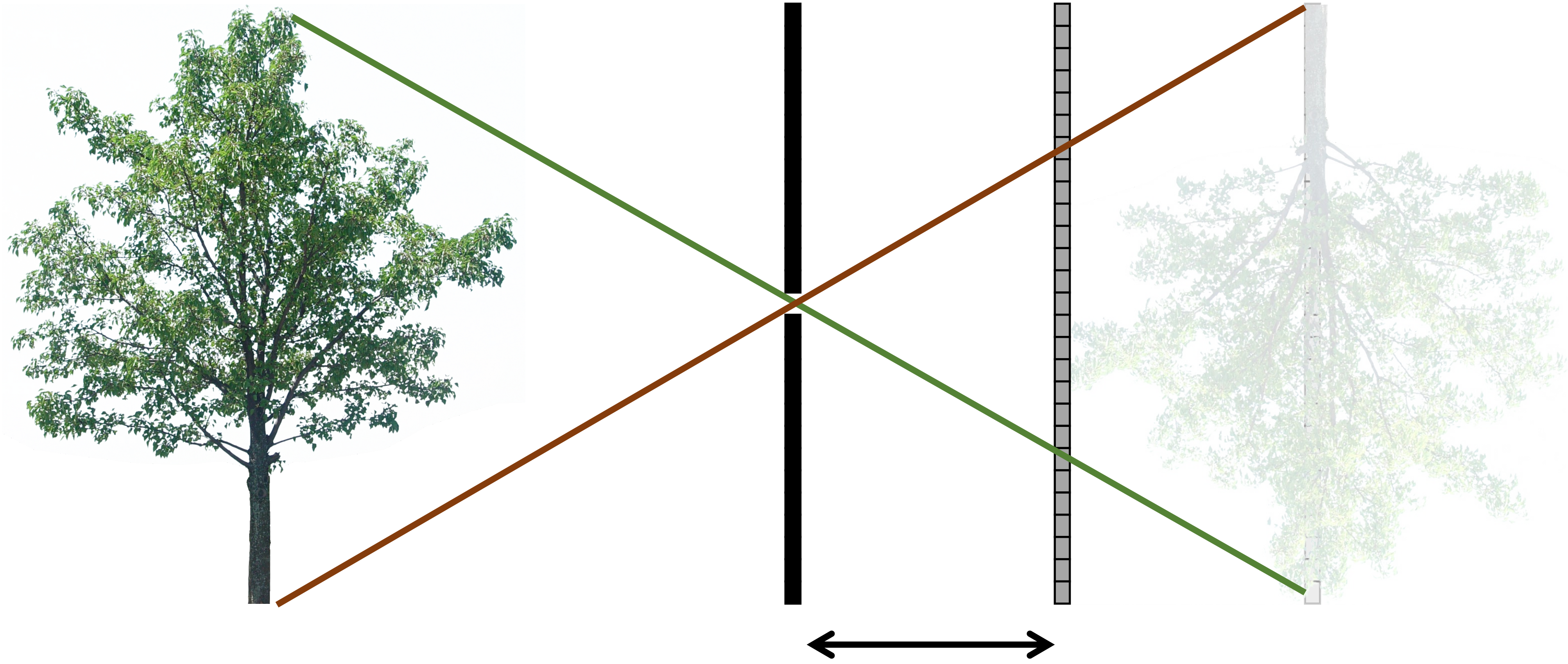
real-world
object



Focal length

What happens as we change the focal length?

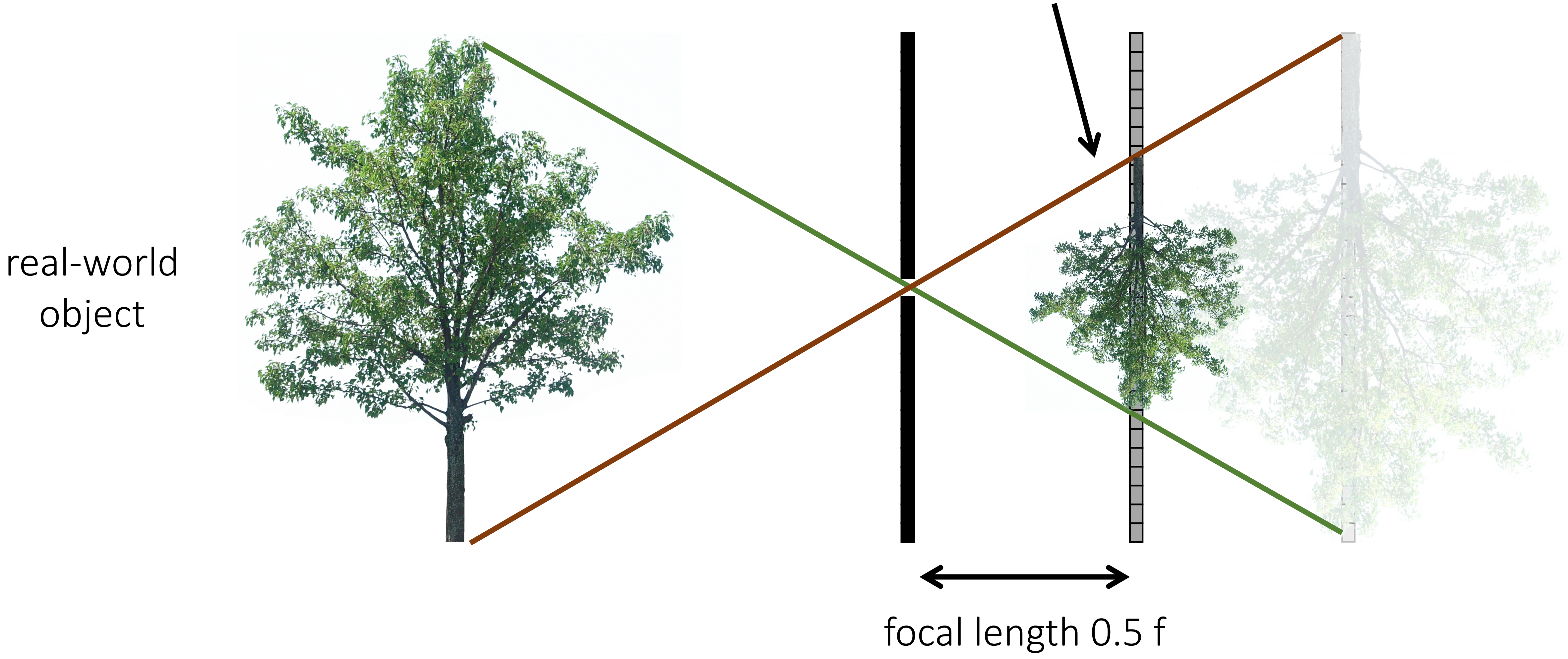
real-world object



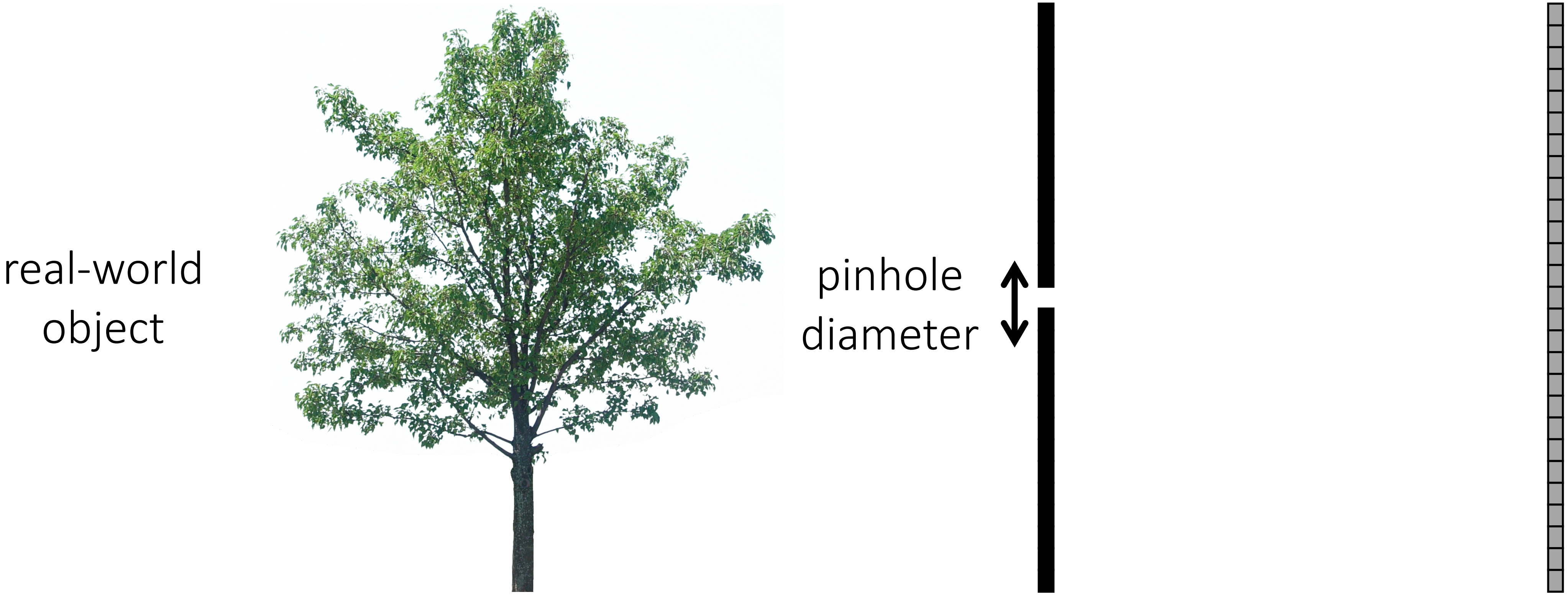
focal length 0.5 f

Focal length

What happens as we change the focal length?



Pinhole size

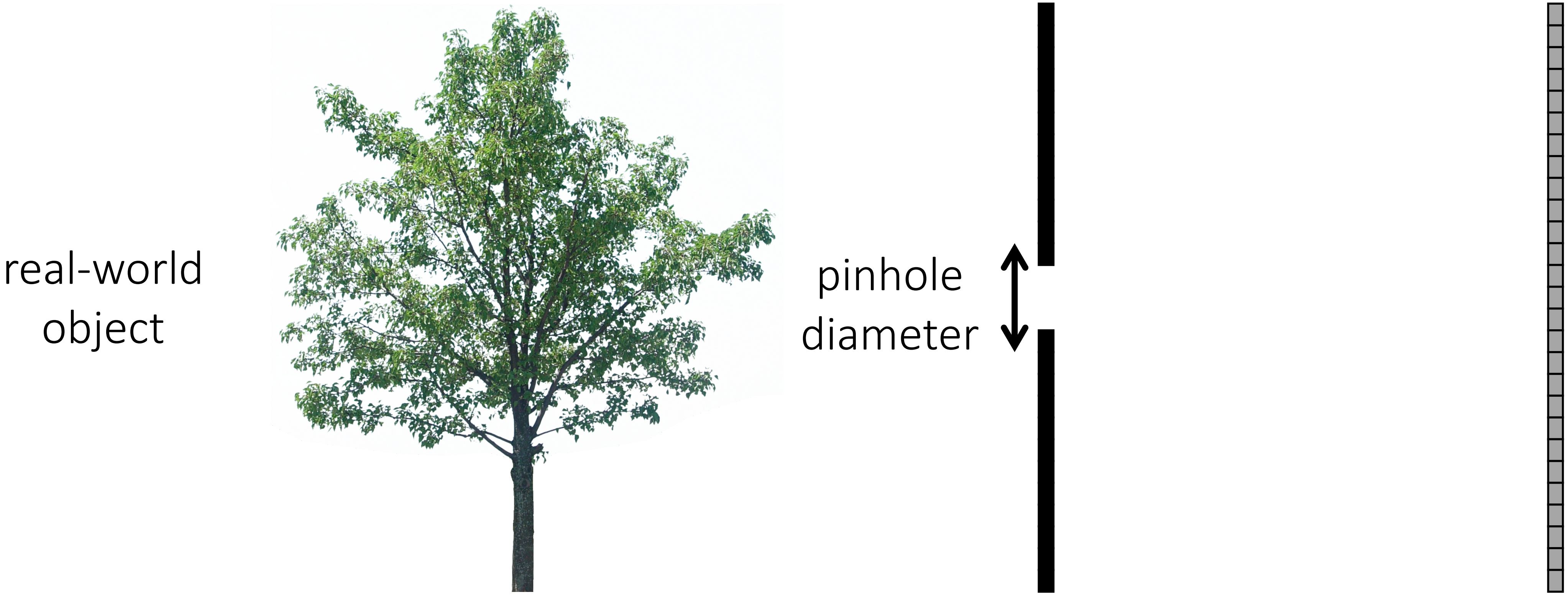


Ideal pinhole has infinitesimally small size

- In practice that is impossible.

Pinhole size

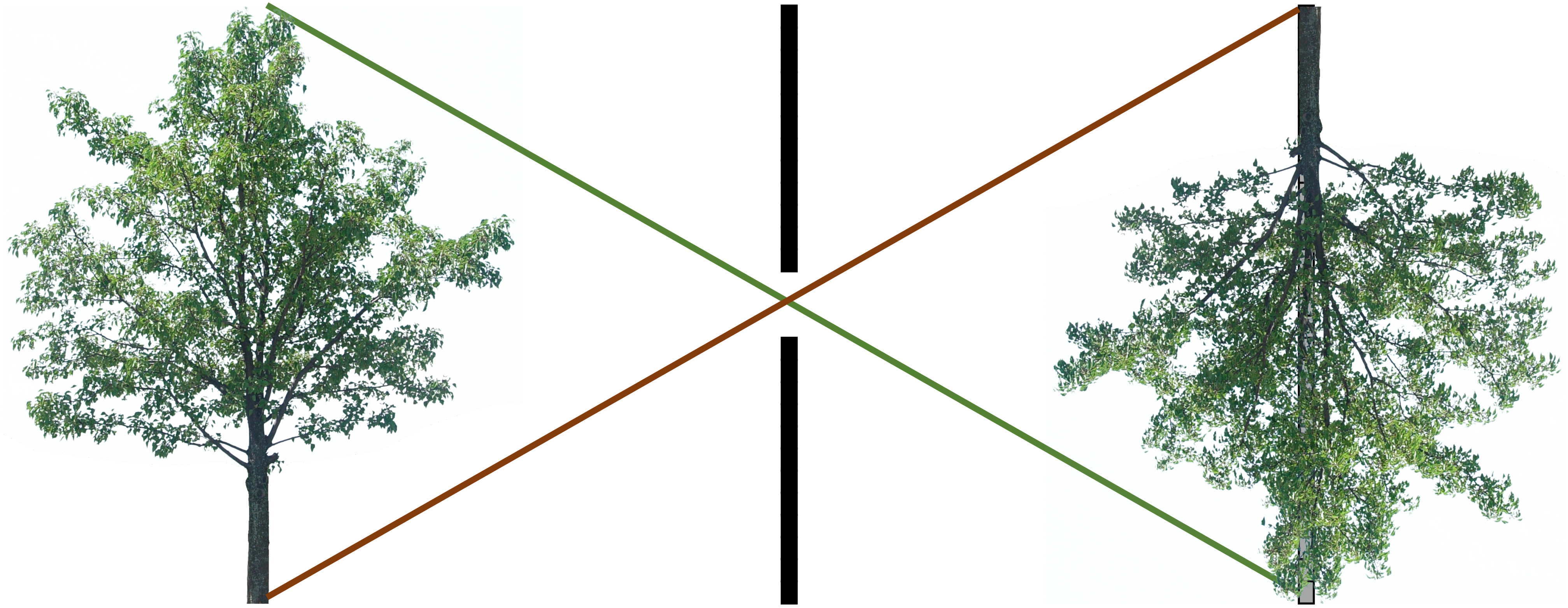
What happens as we change the pinhole diameter?



Pinhole size

What happens as we change the pinhole diameter?

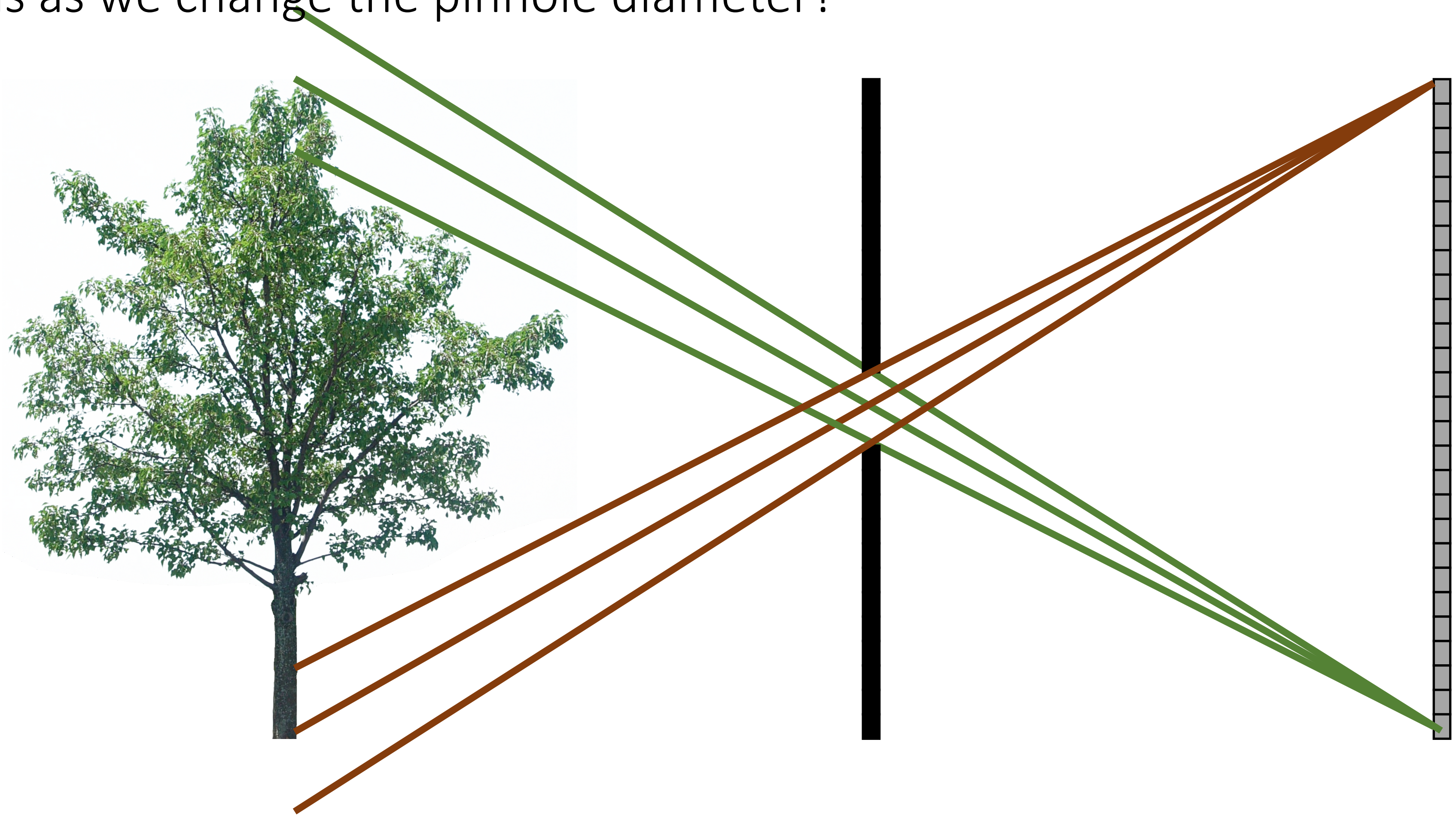
real-world object



Pinhole size

What happens as we change the pinhole diameter?

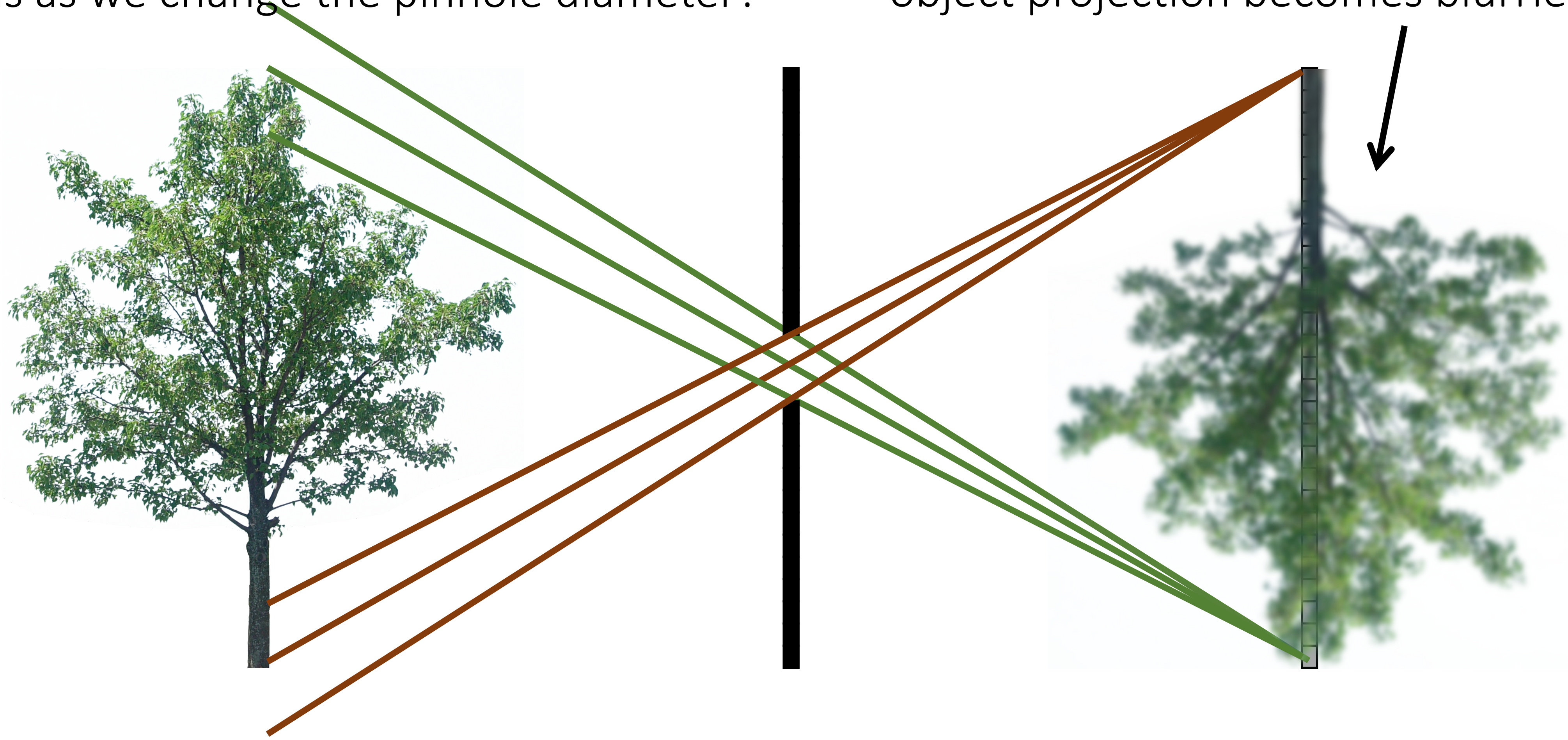
real-world
object



Pinhole size

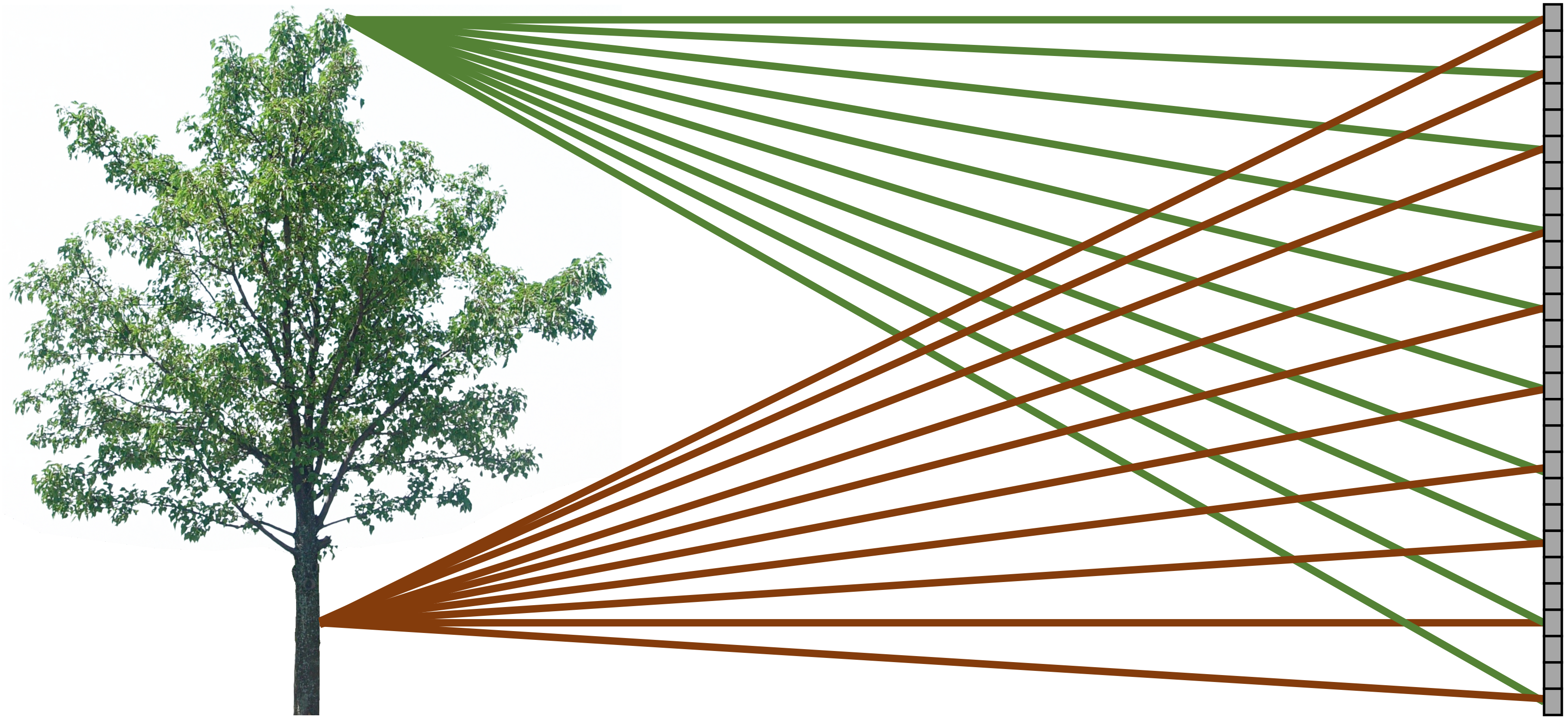
What happens as we change the pinhole diameter?

object projection becomes blurrier



Extreme Case: Bare-sensor imaging

real-world object



digital sensor (CCD or CMOS)

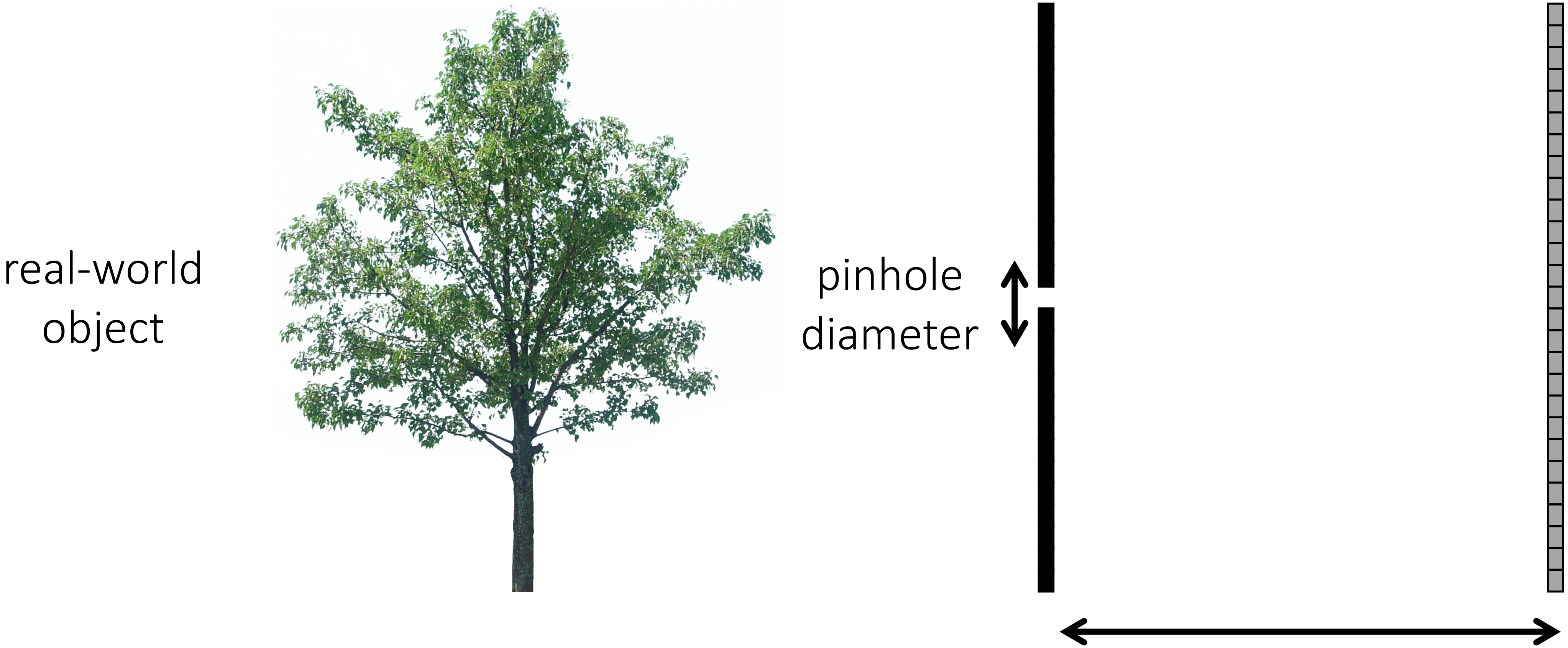
All scene points contribute to all sensor pixels

What does the image on the sensor look like?

Extreme Case: Bare-sensor imaging

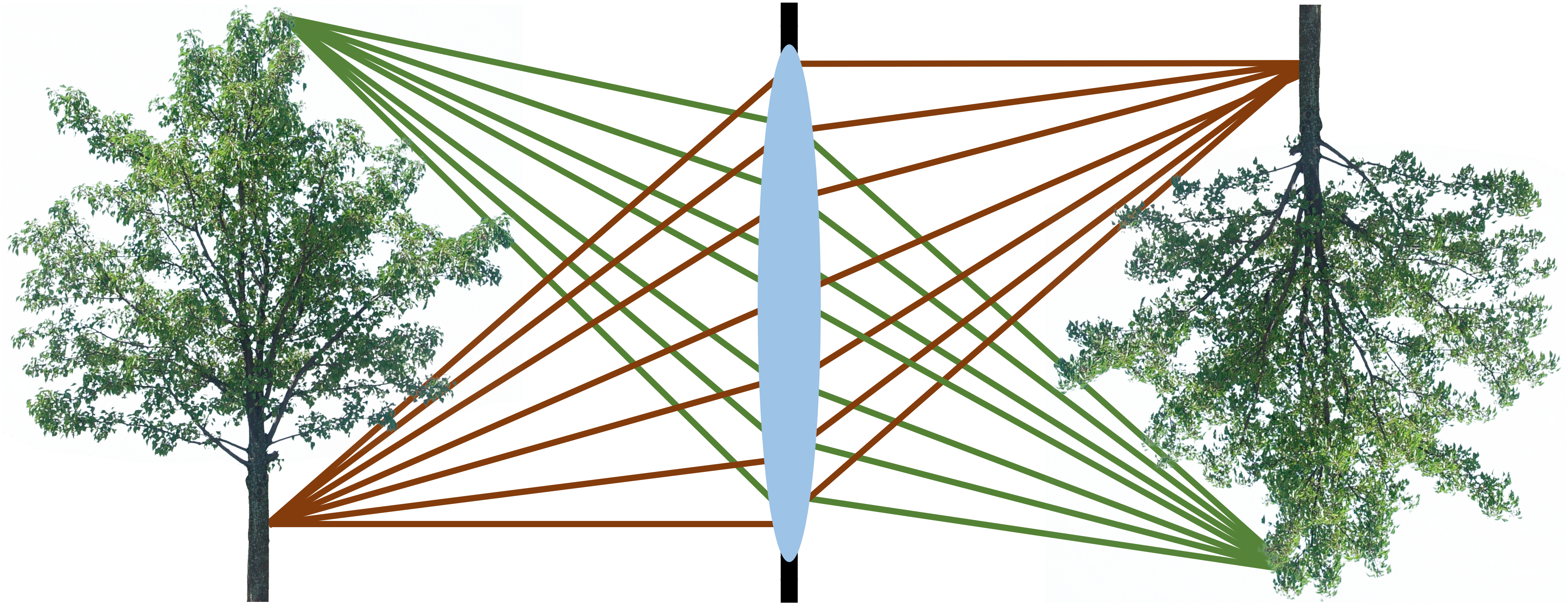
All scene points contribute to all sensor pixels

What about light efficiency?



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

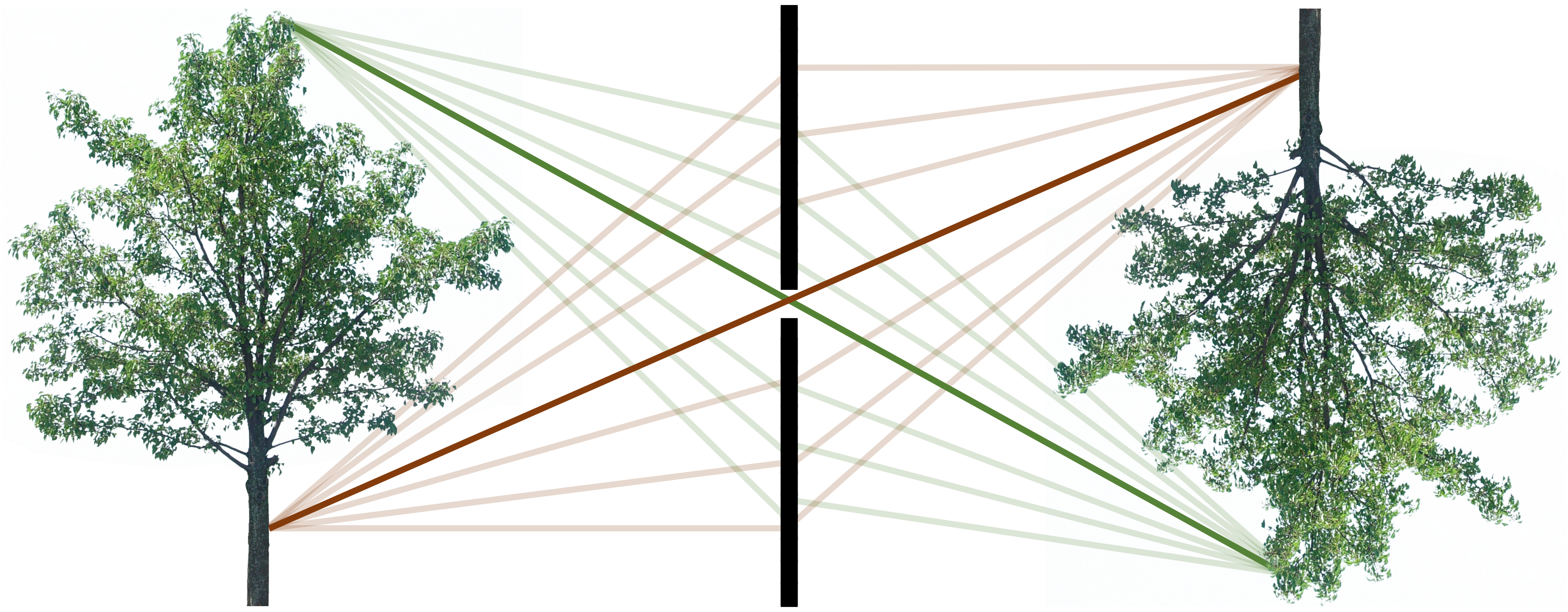
The lens camera



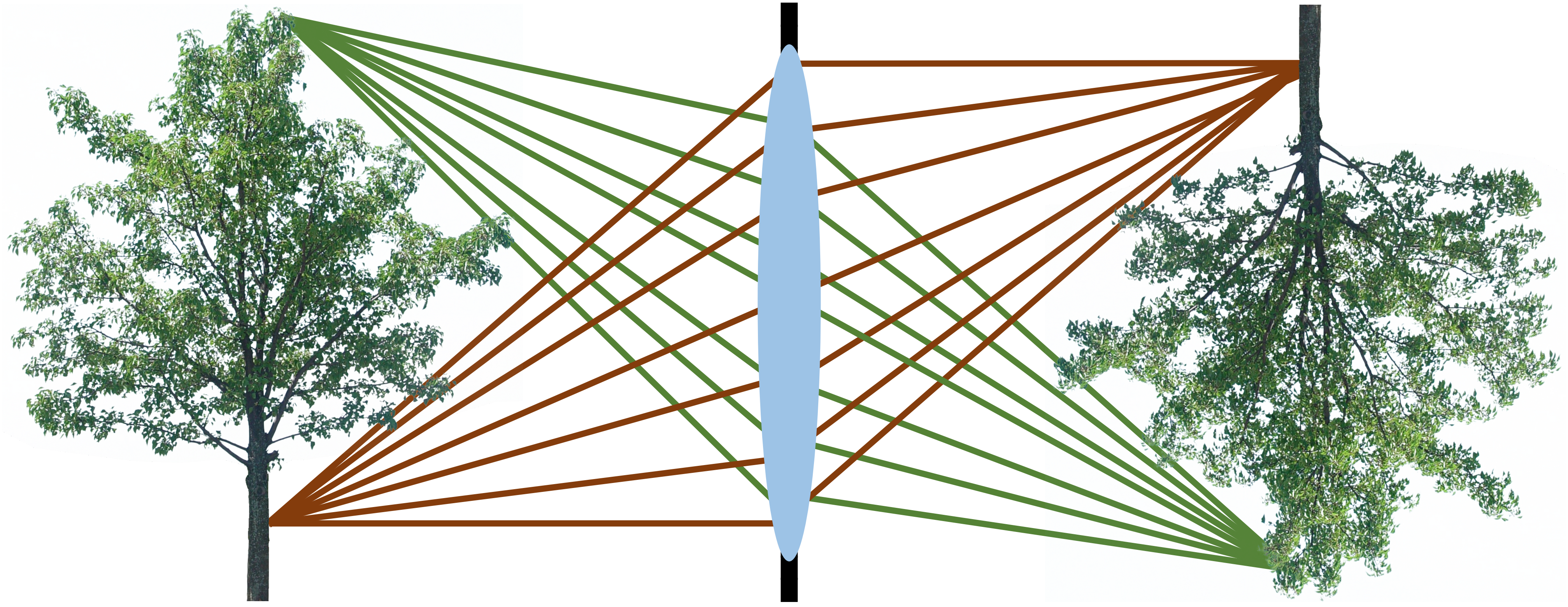
Lenses map “bundles” of rays from points on the scene to the sensor.

How does this mapping work exactly?

The pinhole camera



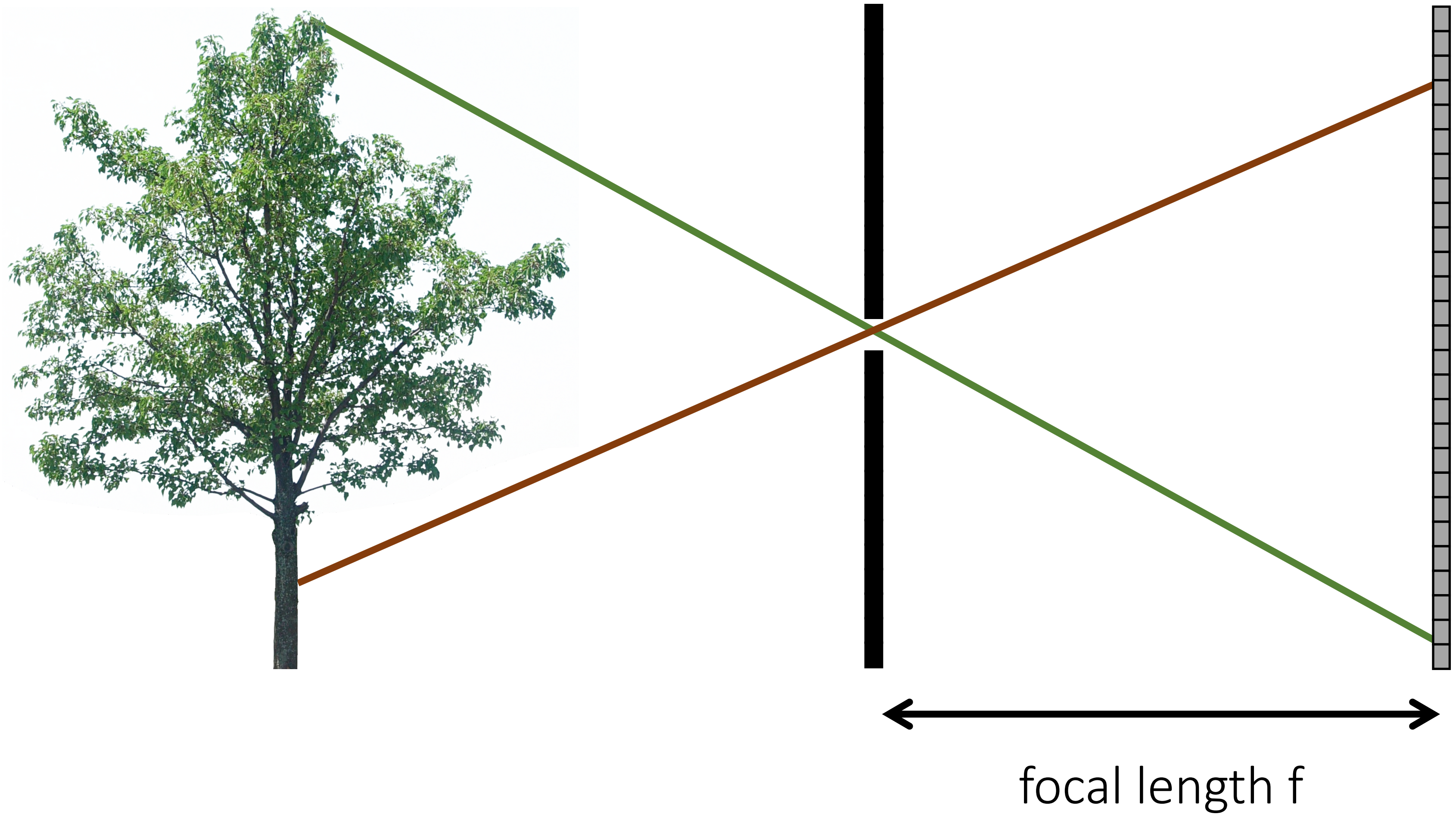
The lens camera



Central rays propagate in the same way for both models!

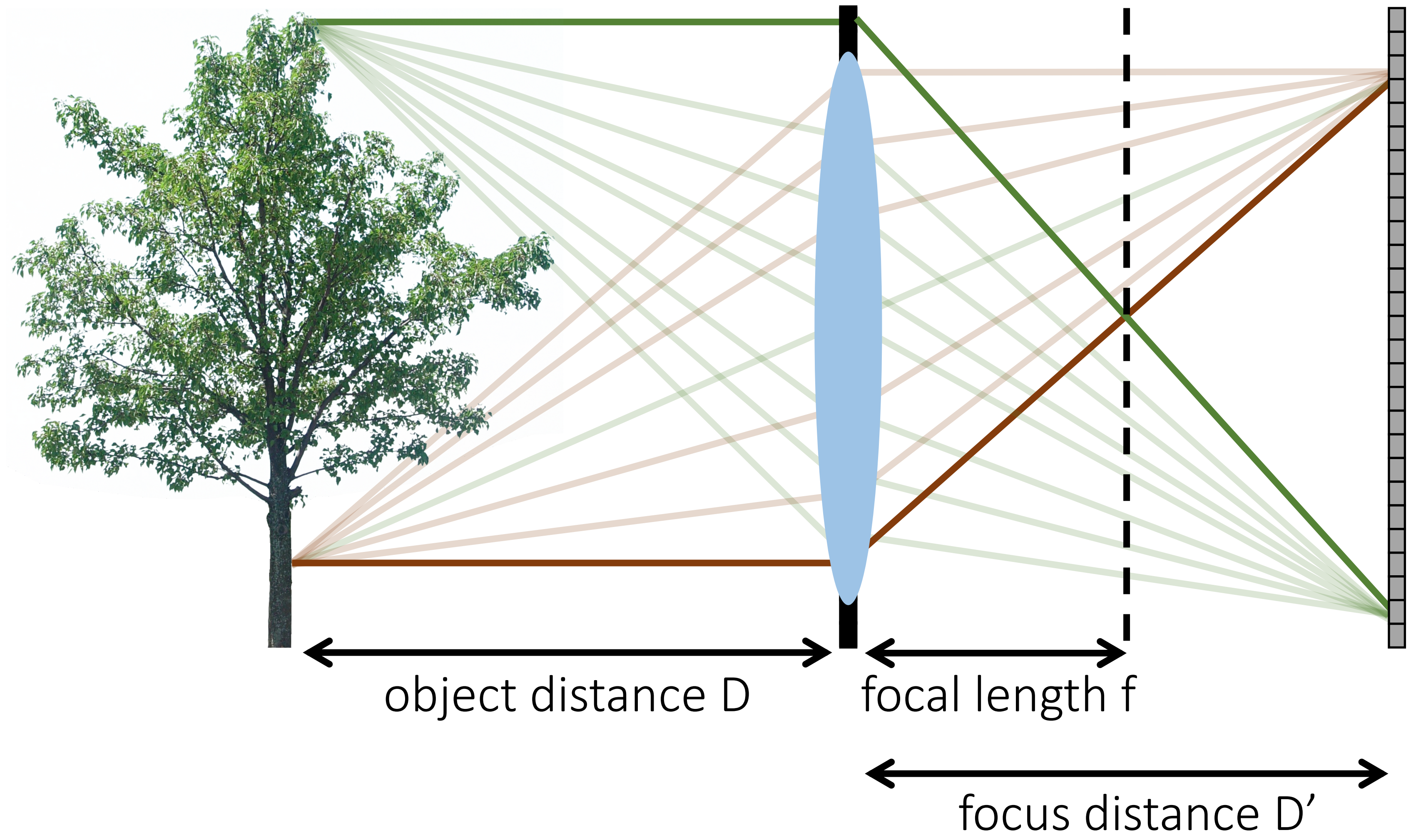
Important difference: focal length

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

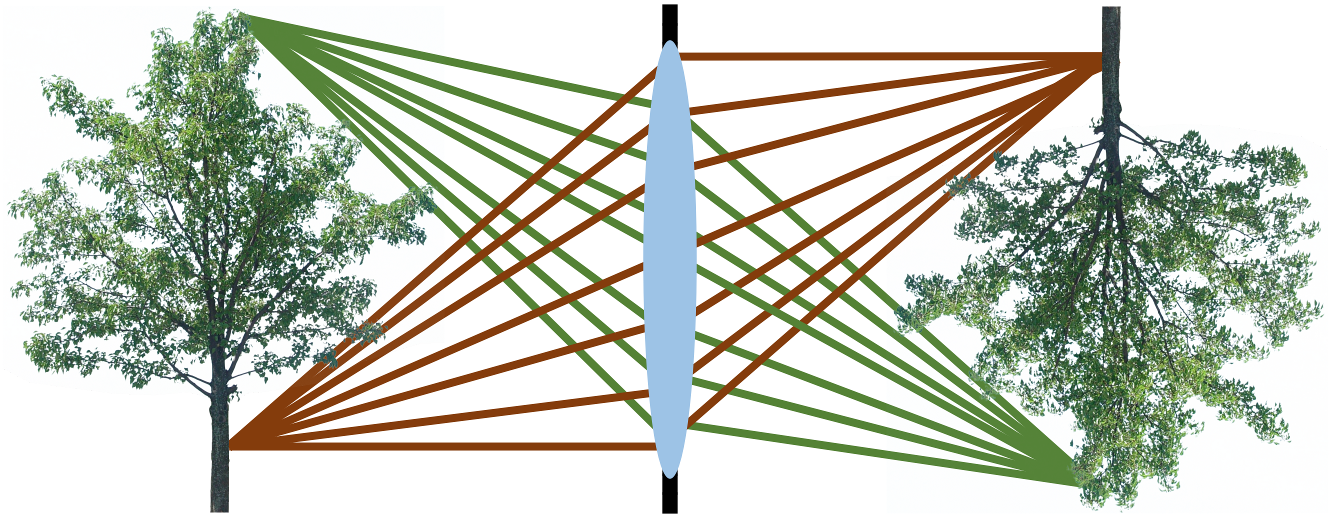


Important difference: focal length

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

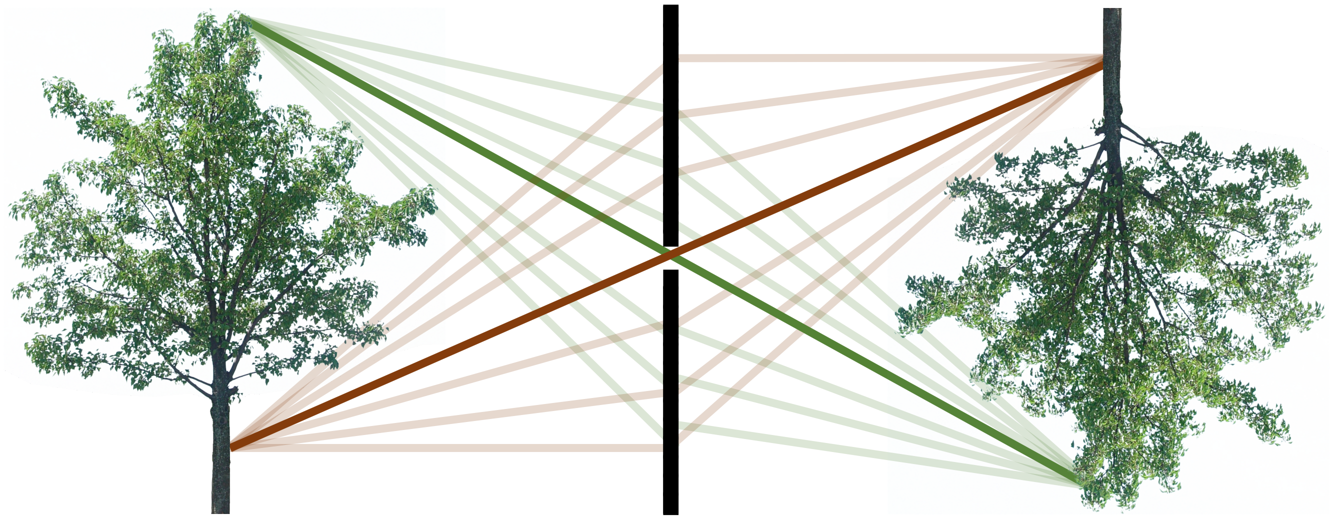


Describing both lens and pinhole cameras



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

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



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.

Camera Matrix

The camera as a coordinate transformation

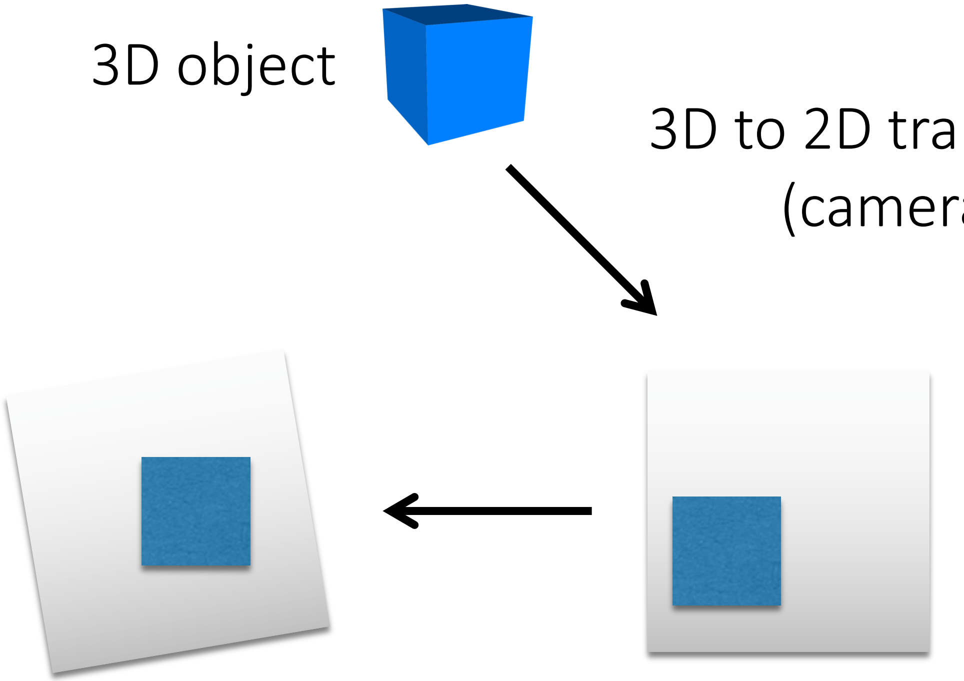
The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image



The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image

homogeneous coordinates

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

2D image point — camera matrix — 3D world point

What are the dimensions of each variable?

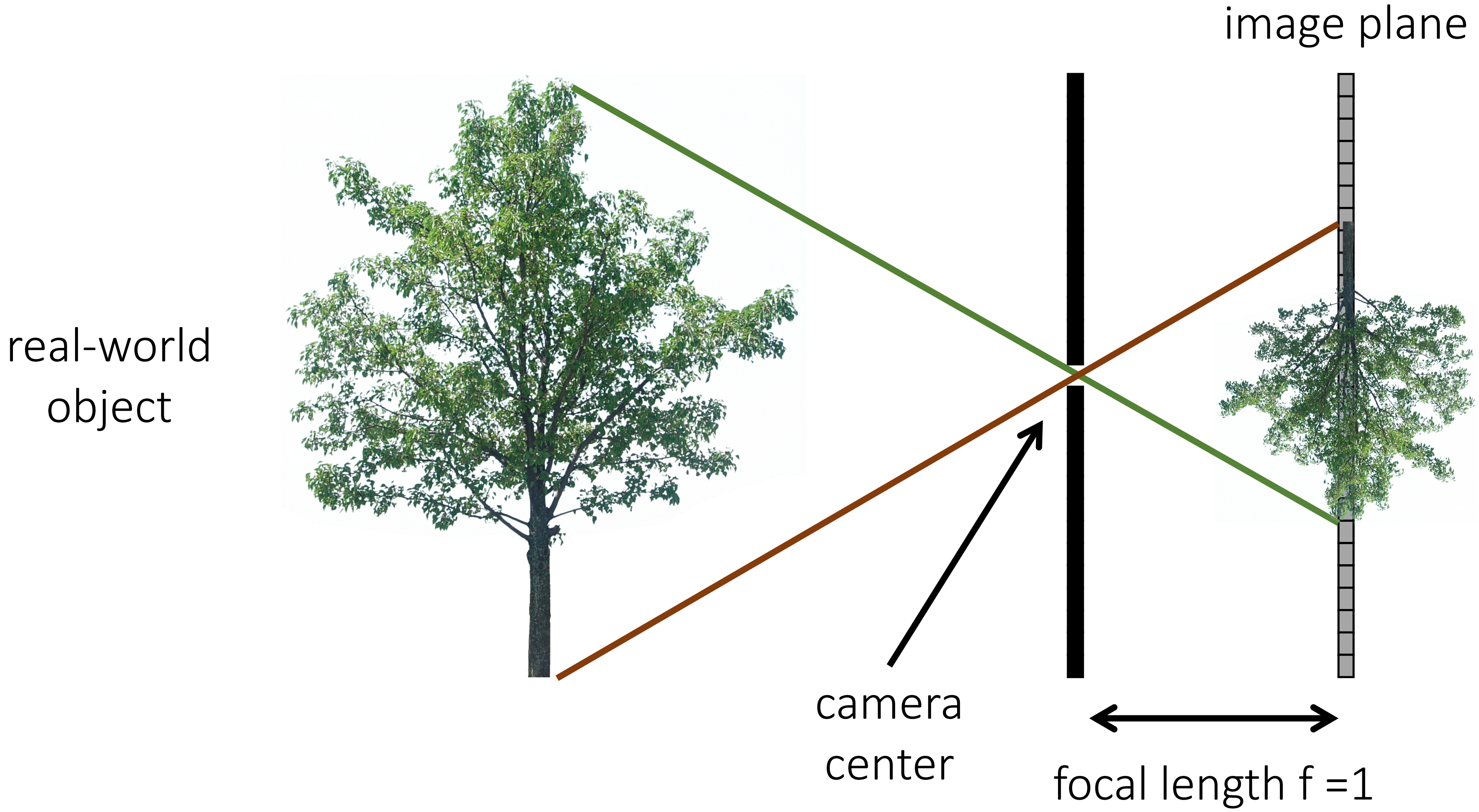
The camera as a coordinate transformation

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

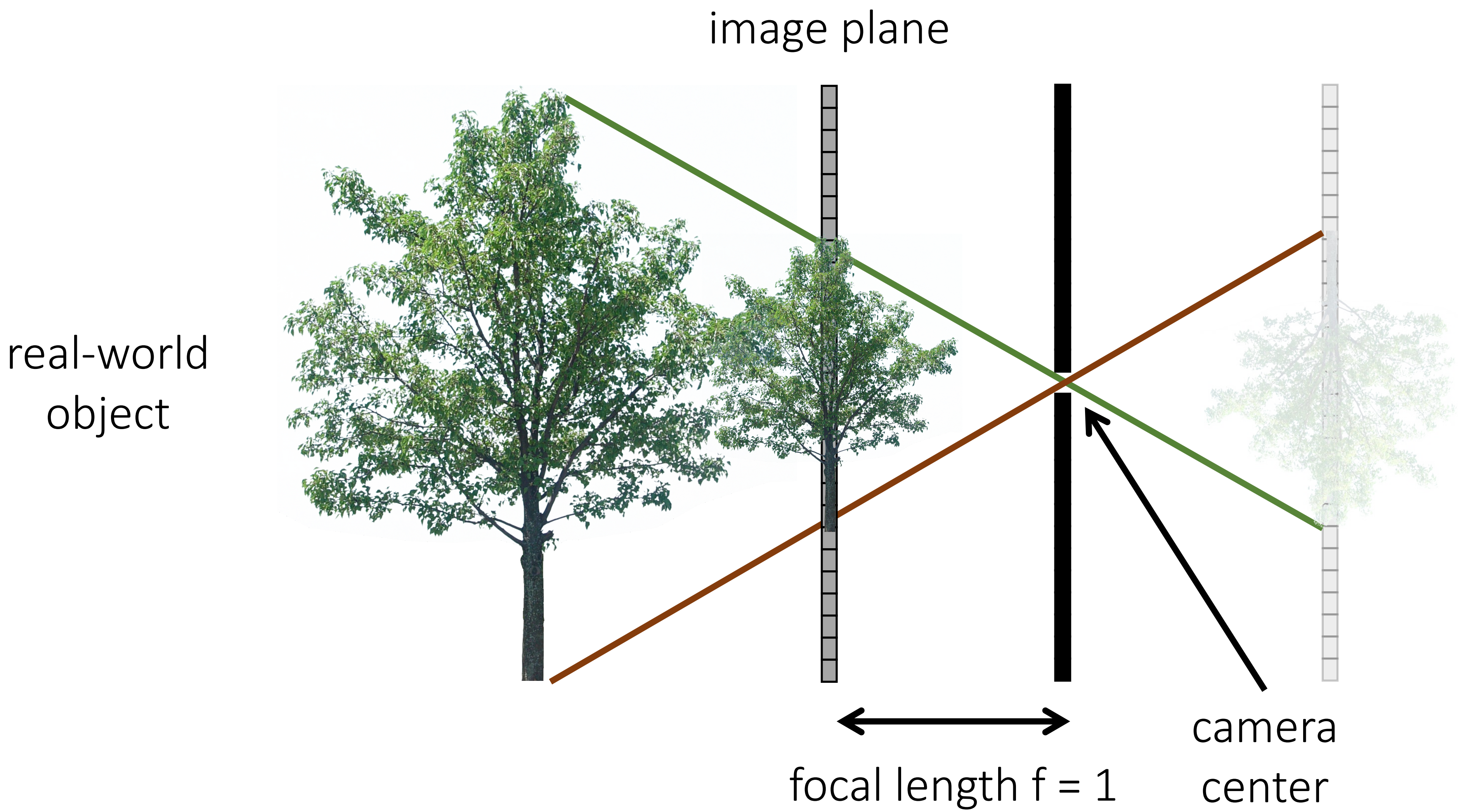
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image coordinates 3 x 1	camera matrix 3 x 4	homogeneous world coordinates 4 x 1

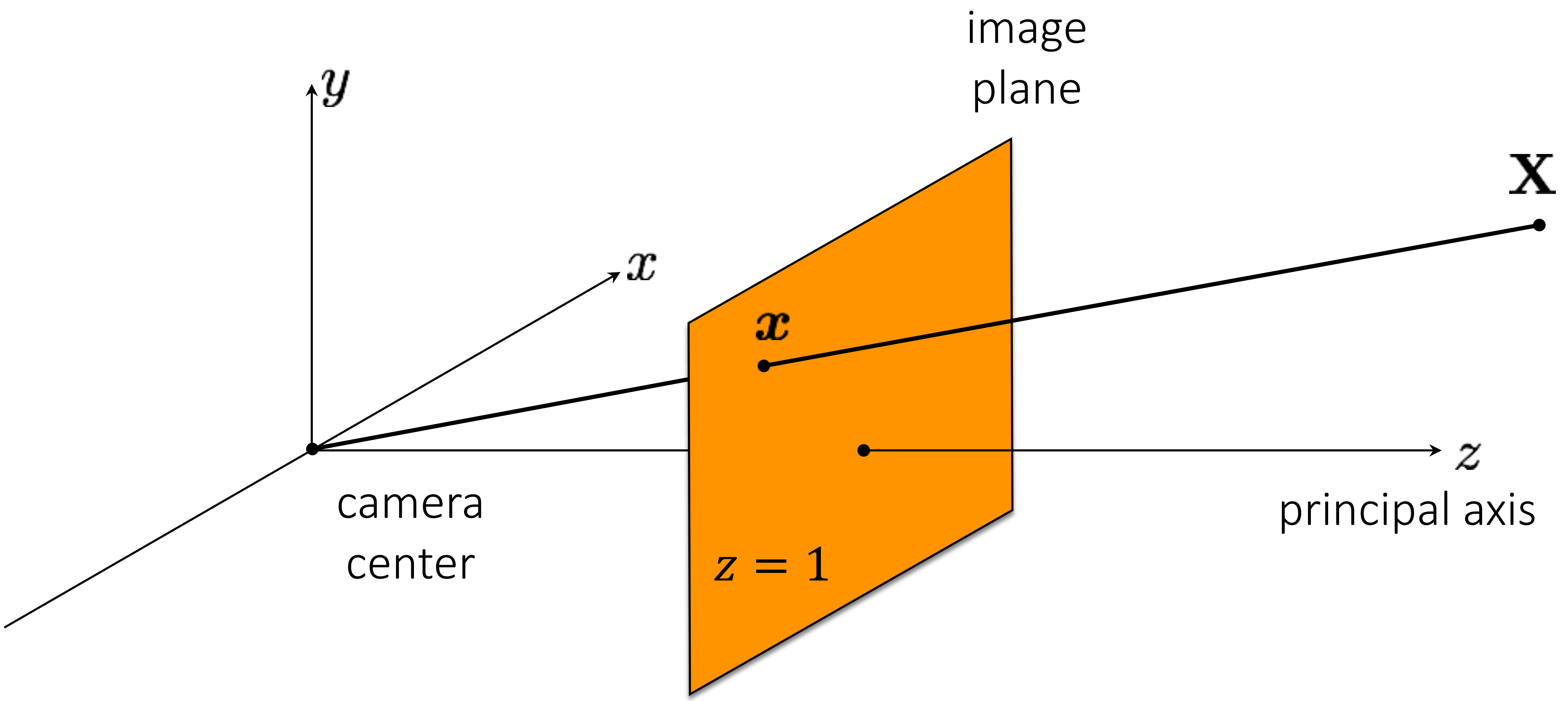
The pinhole camera



The (rearranged) pinhole camera

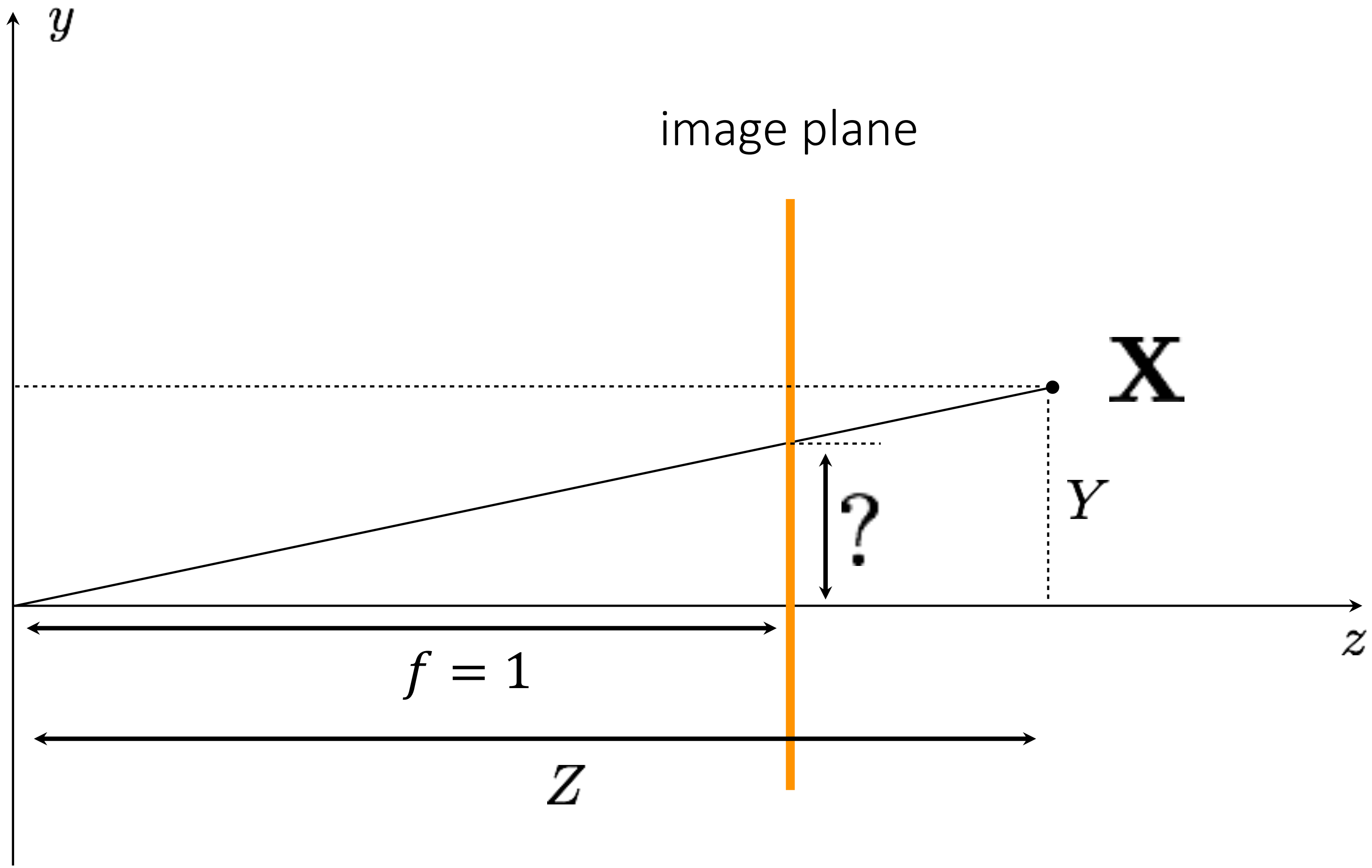


The (rearranged) pinhole camera



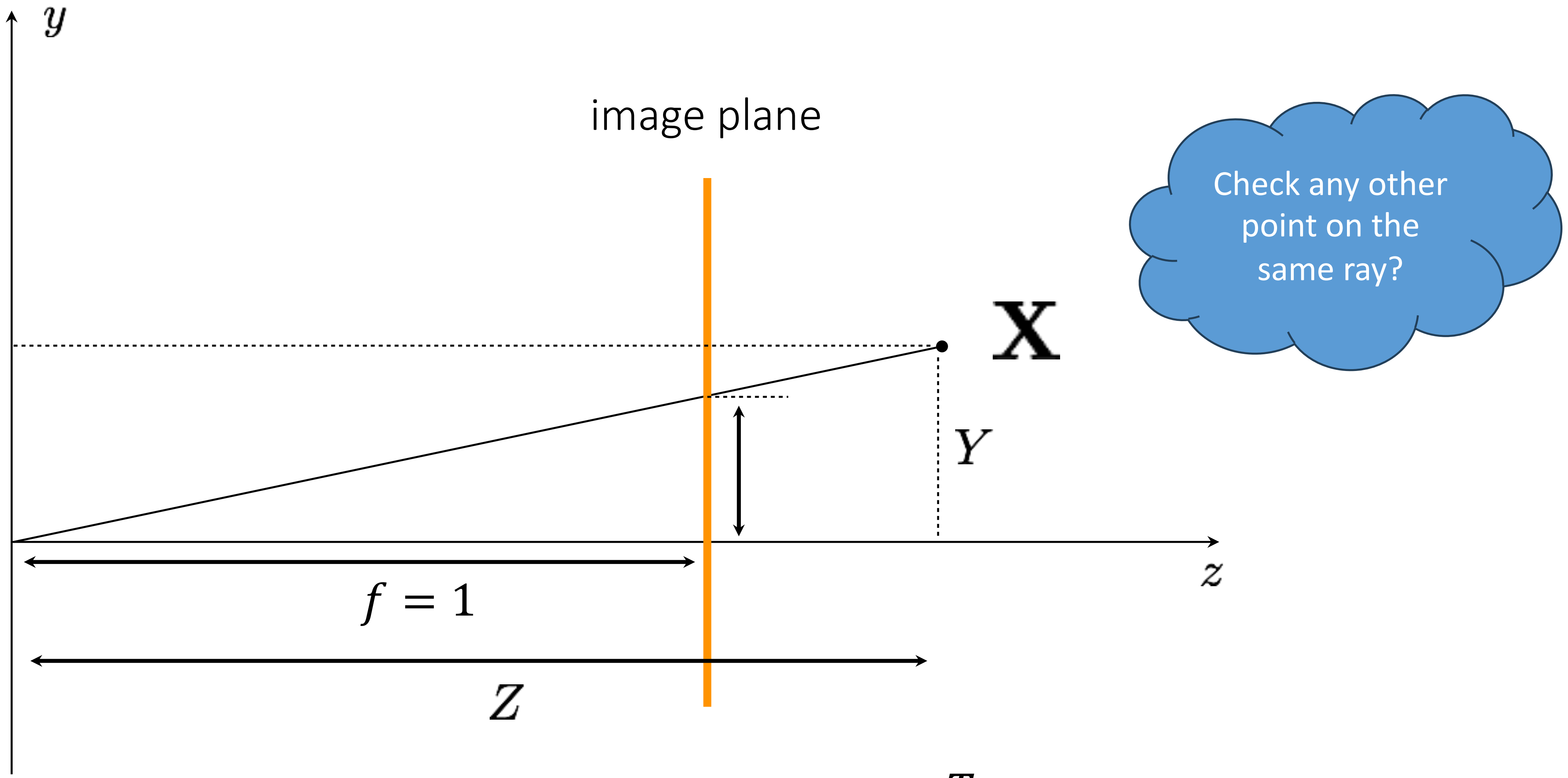
What is the equation for image coordinate **x** in terms of **X**?

The 2D view of the (rearranged) pinhole camera



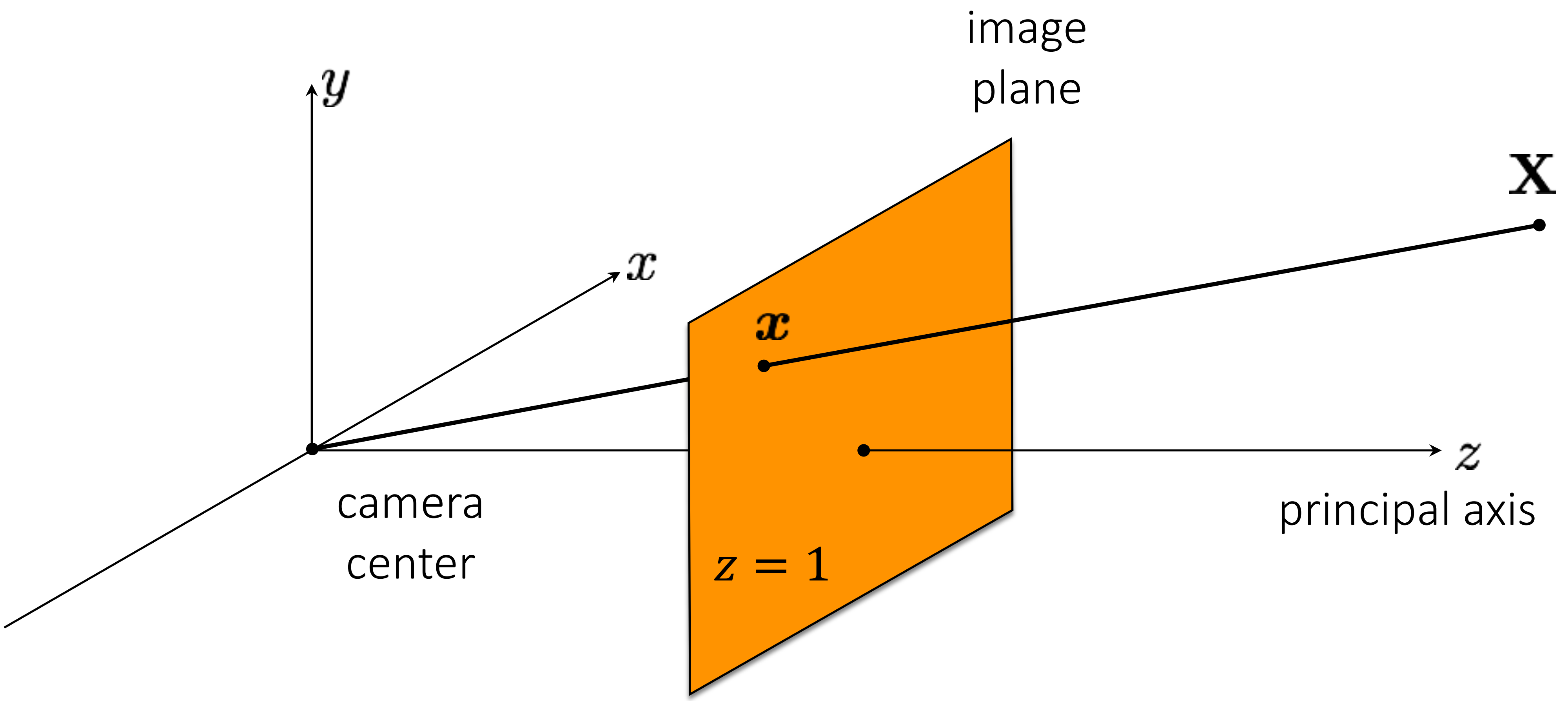
What is the equation for image coordinate **x** in terms of X?

The 2D view of the (rearranged) pinhole camera



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

The (rearranged) pinhole camera



What is the camera matrix **P** for a pinhole camera? Can this be linear transform?

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

Homogeneous Coordinates

Given a point $\mathbf{p}$ in $\mathbb{R}^2$, represented as $P = (p_1, p_2)$, i.e., the vector $\mathbf{p} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^\mathsf{T}$ its homogeneous representation (using homogeneous coordinates) is

$$\tilde{\mathbf{p}} = \begin{bmatrix} \tilde{p}_1 & \tilde{p}_2 & \tilde{p}_3 \end{bmatrix}^\mathsf{T}; \quad \text{with } \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \text{ not allowed}$$

The vector representation is obtained dividing the first n homogeneous components by the $(n+1)$-th, that is often called scale.

$$p_1 = \tilde{p}_1/\tilde{p}_3; \quad p_2 = \tilde{p}_2/\tilde{p}_3$$

Physical meaning: distance between object plane center and camera center!

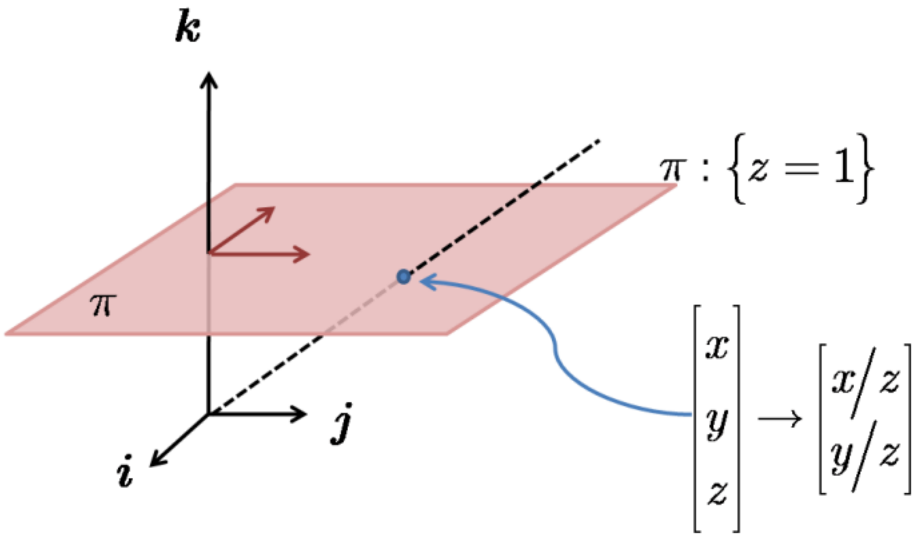


Figure: Geometric interpretation of homogeneous coordinates.

Take-Away:

- All points, on the same projection ray, are mapped to the same homogeneous coordinate!
- It simplifies many equations in projective geometry! Let's see next page…

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?

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

Normal coordinates:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$$

Homogenous coordinates:

$$\begin{bmatrix} x \\ y \\ z(=1) \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

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

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

Normal coordinates:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}$$

Homogenous coordinates:

$$\begin{bmatrix} x \\ y \\ z(=1) \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The pinhole camera matrix

Relationship from similar triangles:

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

General camera model *in homogeneous coordinates*:

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

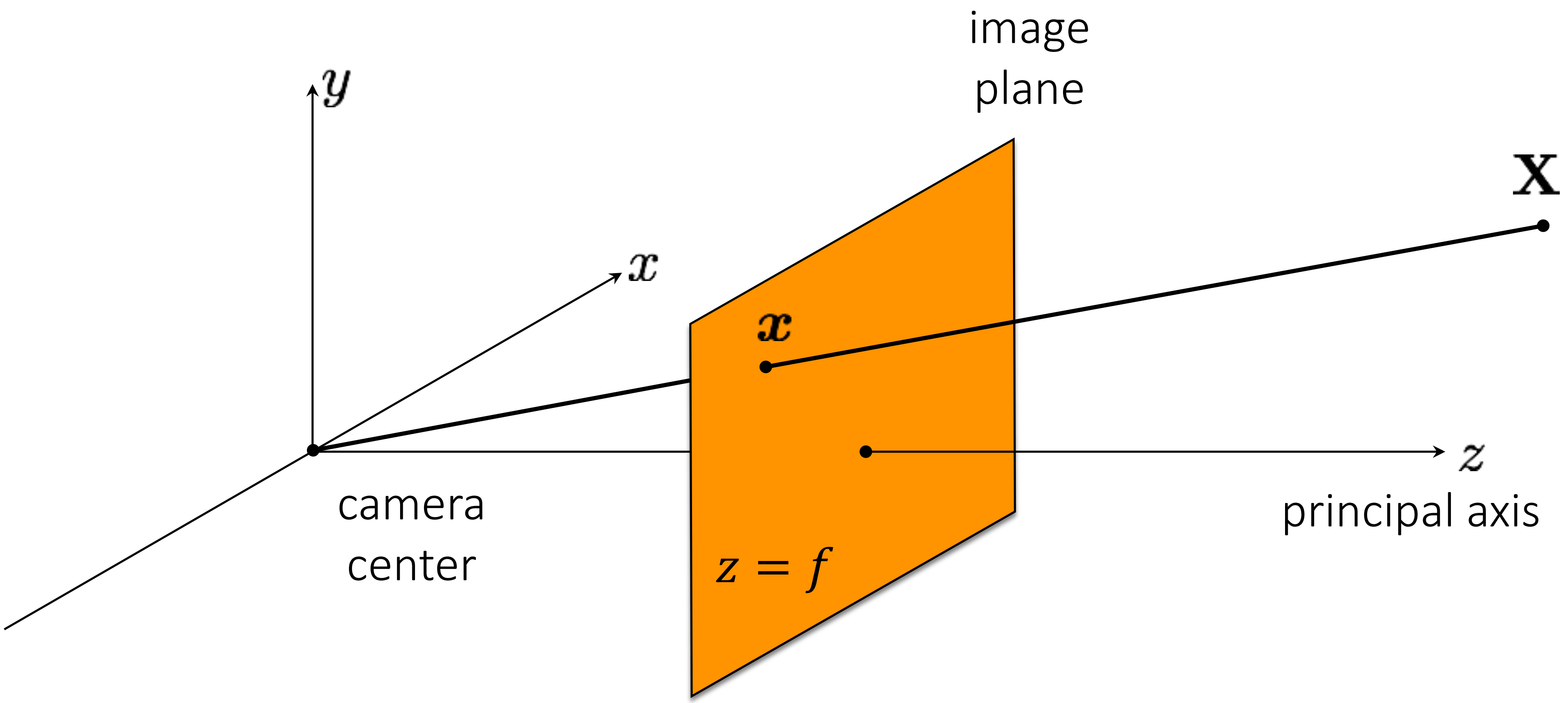
What does the pinhole camera projection look like?

The perspective projection matrix

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

alternative way to write the same thing

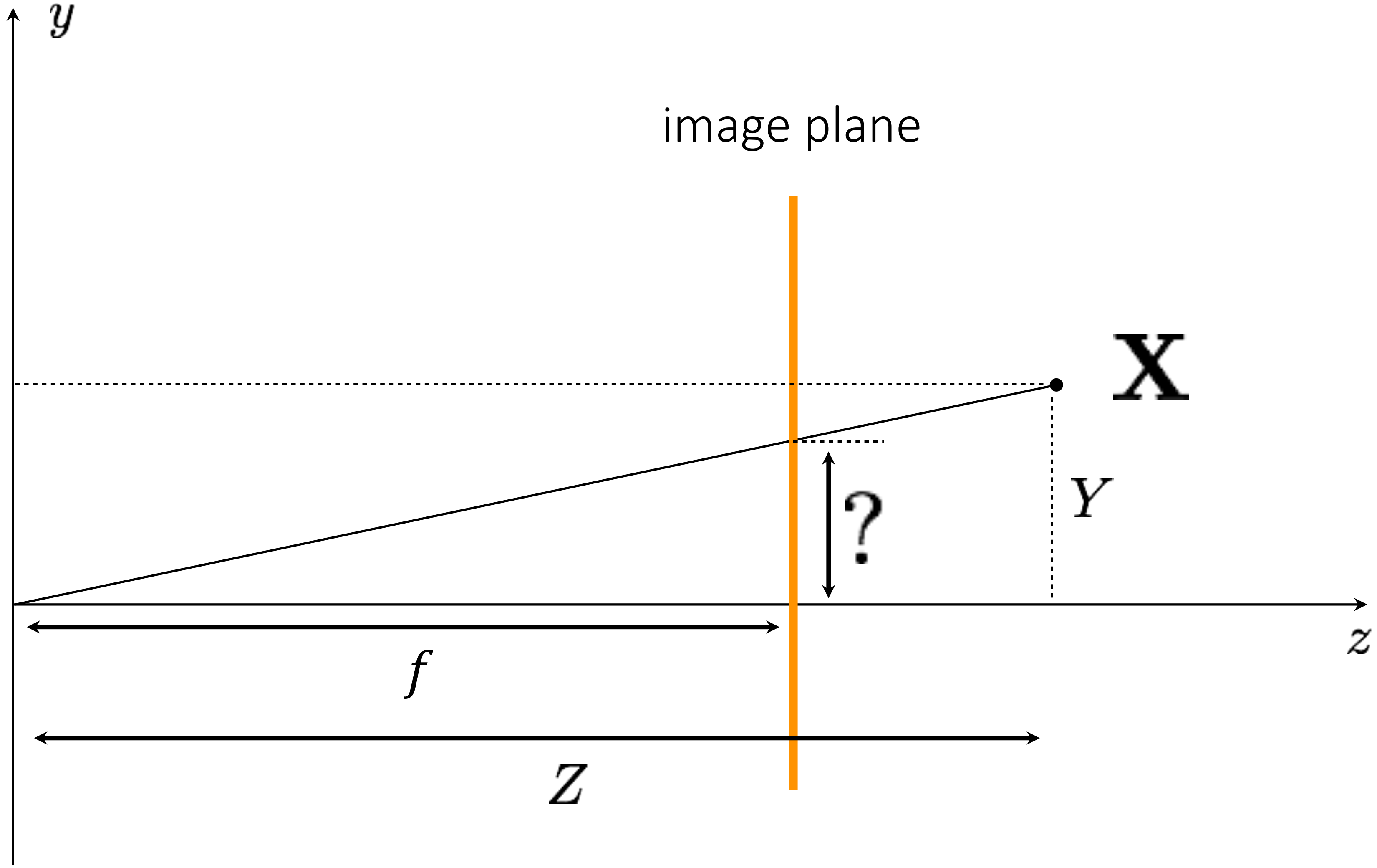
More general case: arbitrary focal length



What is the camera matrix **P** for a pinhole camera?

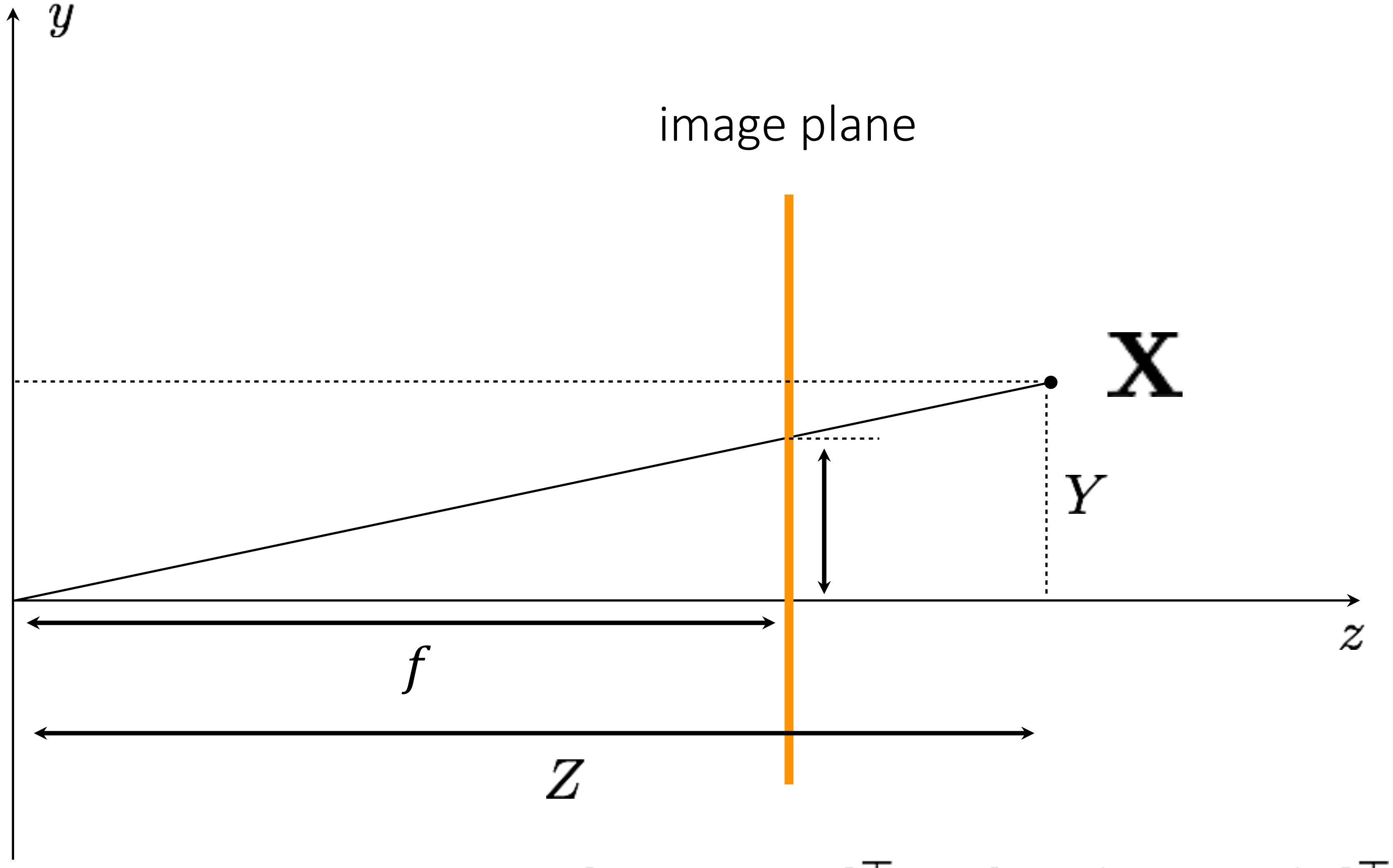
$$\boldsymbol{x} = \mathbf{PX}$$

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



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

The pinhole camera matrix for arbitrary focal length

Relationship from similar triangles:

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

General camera model *in homogeneous coordinates*:

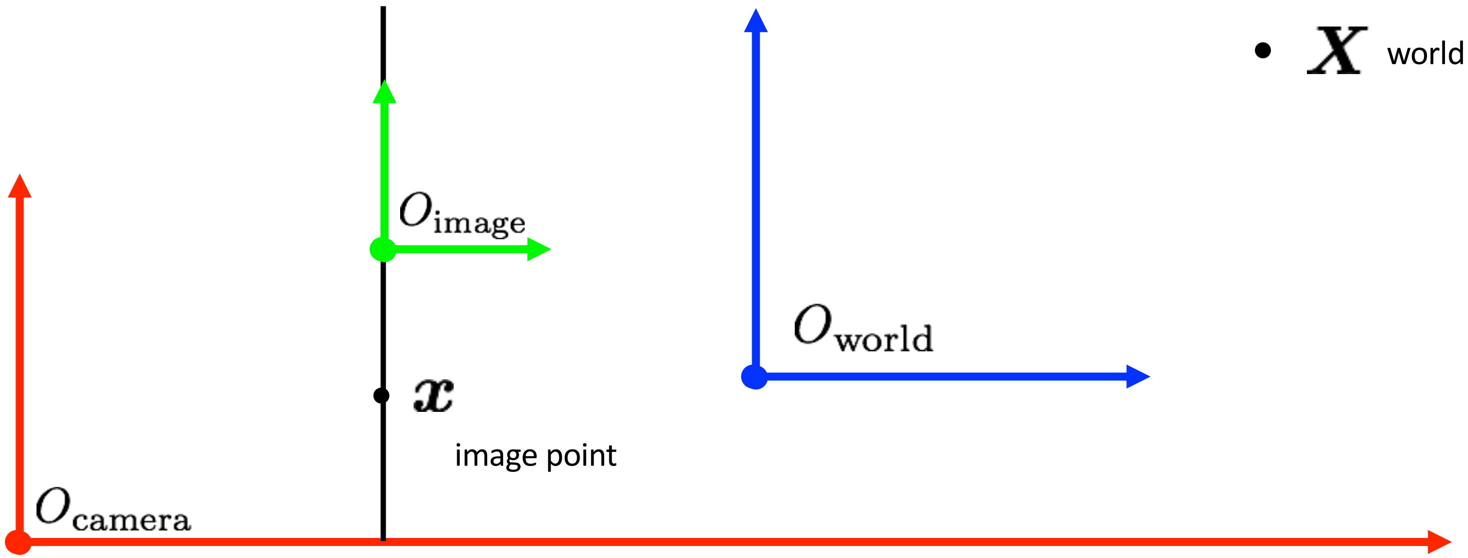
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

What does the pinhole camera projection look like?

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

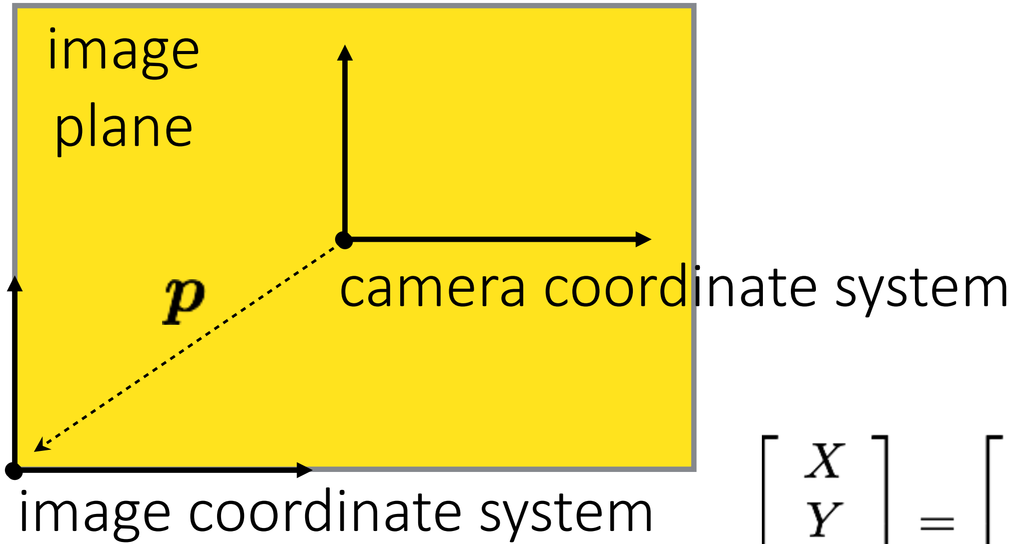
Generalizing the camera matrix

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



Generalizing the camera matrix

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



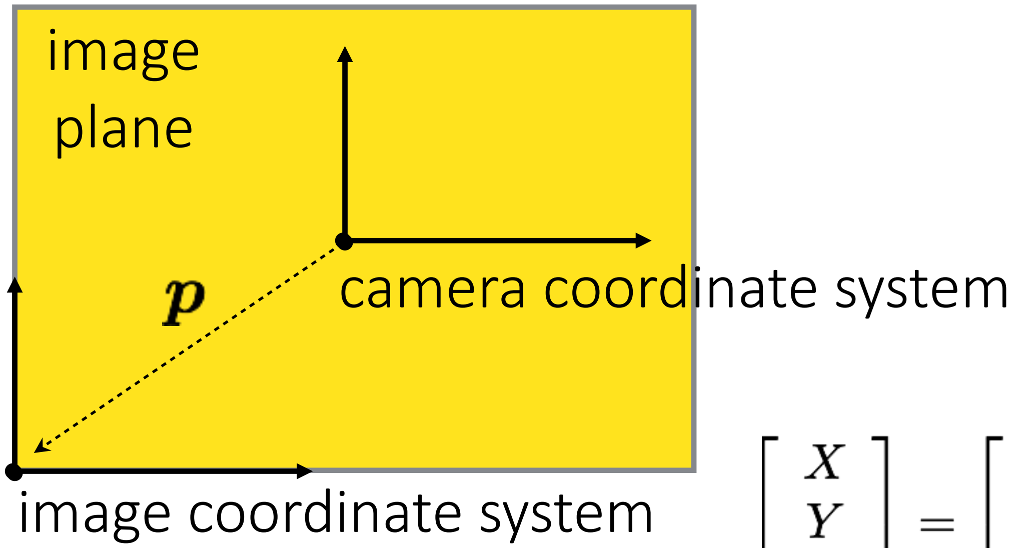
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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:



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

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:

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

What does each part of the matrix represent?

Camera matrix decomposition

We can decompose the camera matrix like this:

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

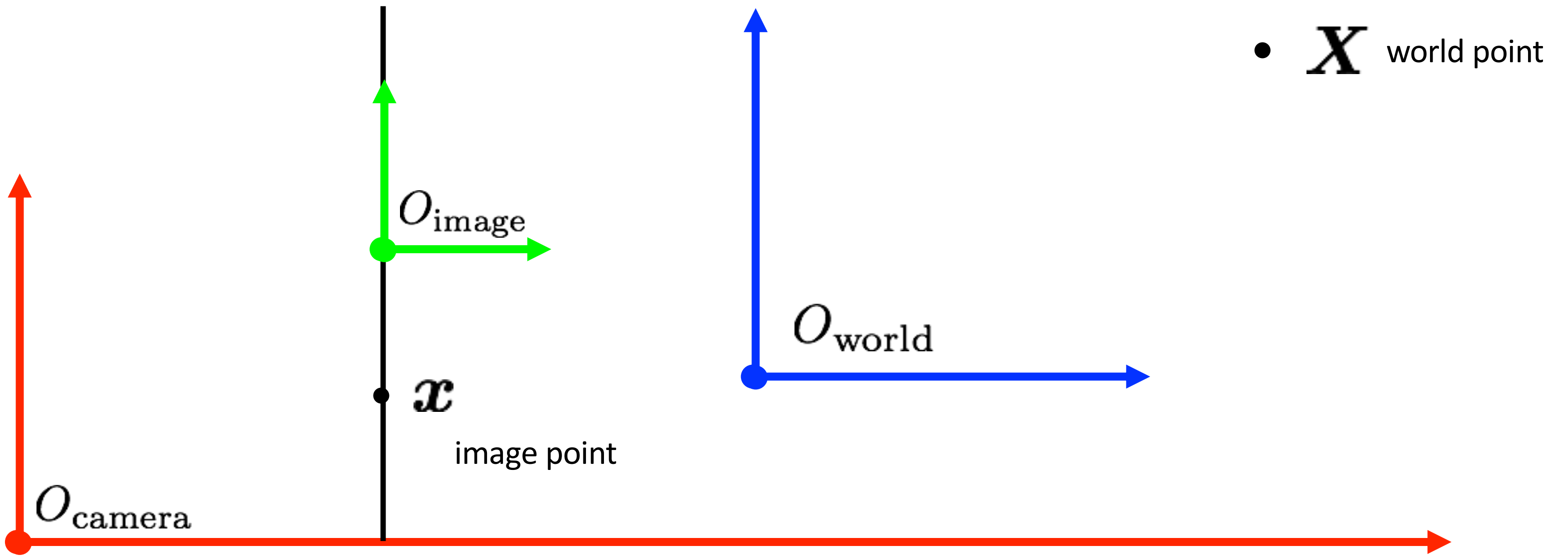
(homogeneous) transformation from 2D to 2D, accounting for not unit focal length and origin shift

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

Also written as: $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ where $\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$

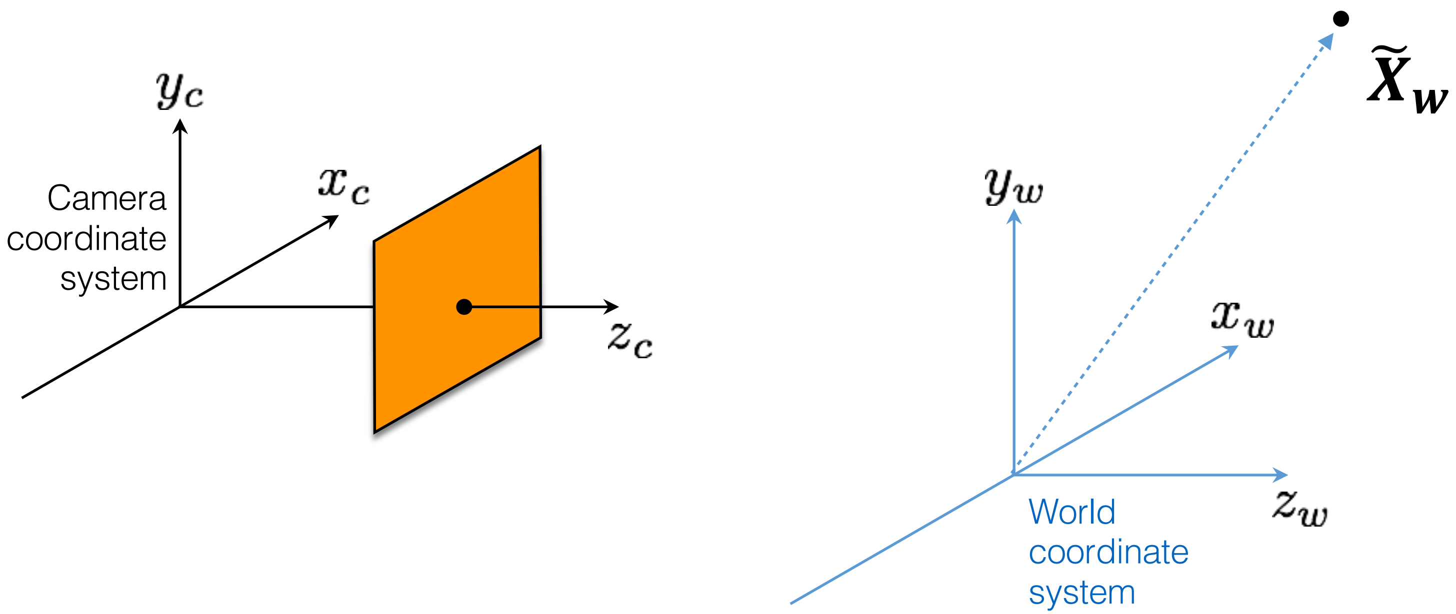
Generalizing the camera matrix

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



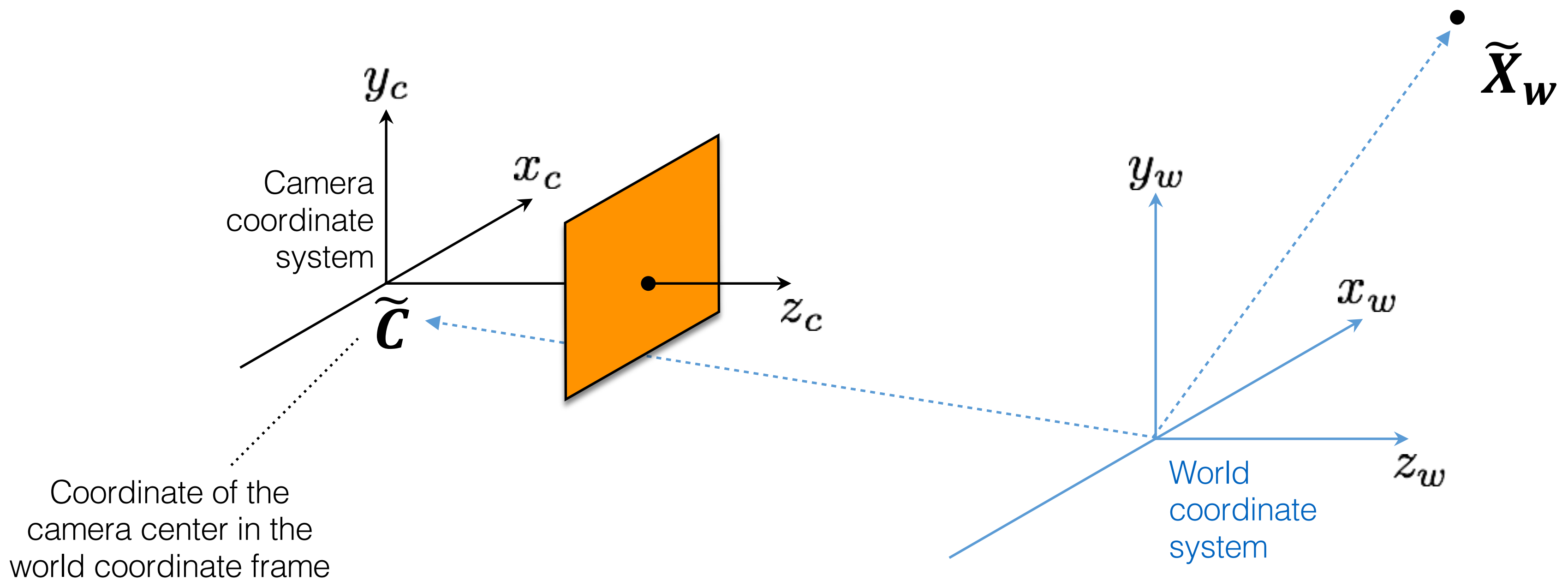
We need to know the transformations between them.

World-to-camera coordinate system transformation

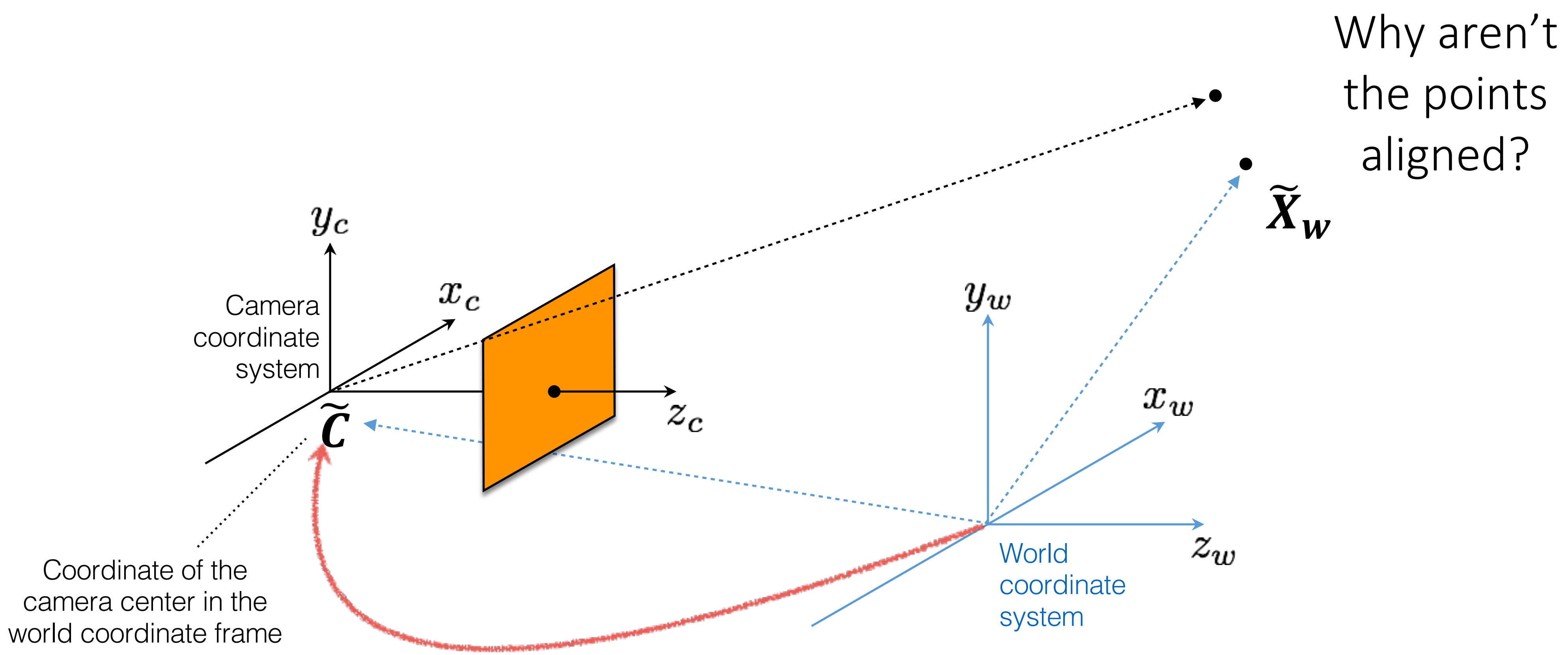


tilde means *heterogeneous* coordinates

World-to-camera coordinate system transformation



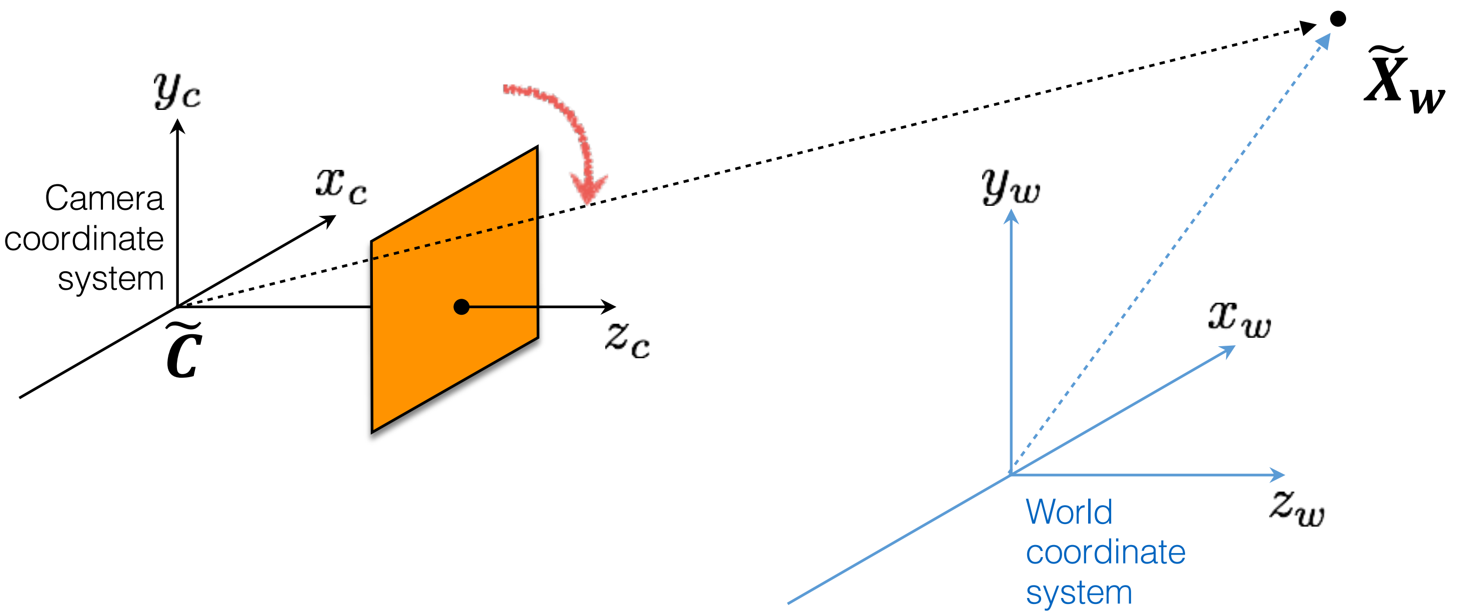
World-to-camera coordinate system transformation



$$(\tilde{\boldsymbol{X}}_{\boldsymbol{w}} - \tilde{\boldsymbol{C}})$$

translate

World-to-camera coordinate system transformation



points now coincide

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

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: (pay attention to R and C dimension!)

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

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

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

Or:

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

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

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	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?

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

3D world points to 2D image points, in homogeneous coordinates

Quiz

The camera matrix relates what two quantities?

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

3D world points to 2D image points, in homogeneous coordinates

The camera matrix can be decomposed into?

Quiz

The camera matrix relates what two quantities?

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

3D world points to 2D image points, in homogeneous coordinates

The camera matrix can be decomposed into?

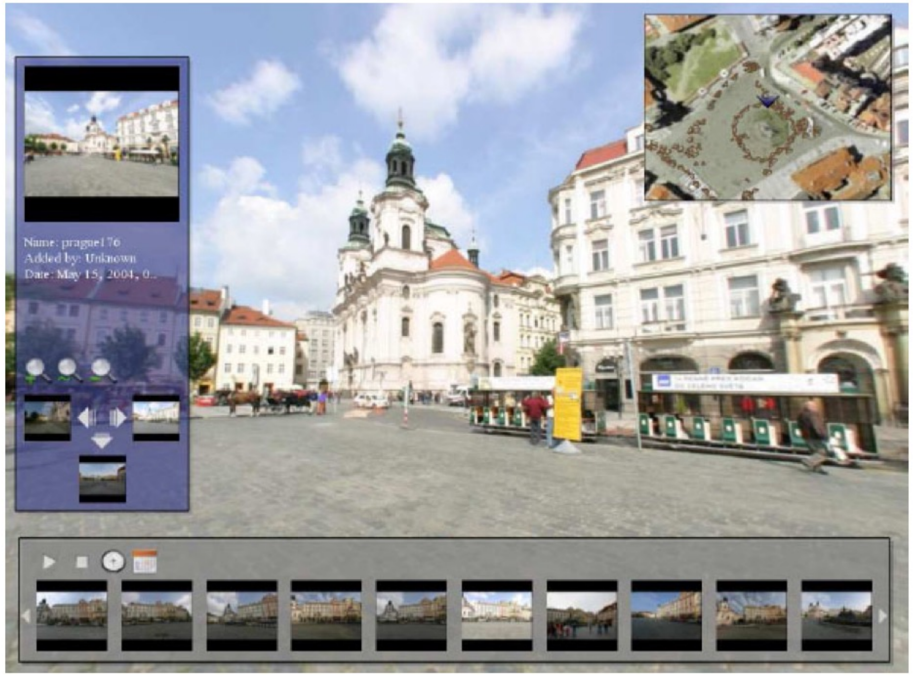
$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

intrinsic and extrinsic parameters

Geometric camera calibration

(a.k.a. camera pose estimation)

Pose Estimation



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

Geometric camera calibration

Given a set of matched points

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

point in 3D space

point in the image

and camera model

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

projection model

parameters

Camera matrix

Find the (pose) estimate of

$$\mathbf{P}$$

Mapping between 3D point and image points

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

What are the unknowns?

Mapping between 3D point and image points

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

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

Heterogeneous coordinates

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

(non-linear relation between coordinates)

How can we make these relations linear?

How can we make these relations linear?

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

Make them linear with algebraic manipulation…

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

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

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

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

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

How do we proceed?

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

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

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

How do we proceed?

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

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

In matrix form …

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

For N points …

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

How do we solve this system?

Solve for camera matrix by

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

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

SVD!

Solve for camera matrix by

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

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

Solution **x** is the column of **V** corresponding to smallest singular value of

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

Solve for camera matrix by

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

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

Equivalently, solution ***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

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

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

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

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

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

Find the camera center **C**

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

How do we compute the camera center from this?

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

Decomposition of the Camera Matrix

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

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

Find the camera center **C**

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

SVD of P!

c *is the Eigenvector corresponding to smallest Eigenvalue*

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

Decomposition of the Camera Matrix

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

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

Find the camera center **C**

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

SVD of P!

***c** is the Eigenvector corresponding to smallest Eigenvalue*

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

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

Any useful properties of K and R we can use?

Decomposition of the Camera Matrix

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

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

Find the camera center **C**

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

SVD of P!

***c** is the Eigenvector corresponding to smallest Eigenvalue*

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

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

right upper triangle

orthogonal

How do we find K and R?

Decomposition of the Camera Matrix

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

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

Find the camera center **C**

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

SVD of P!

***c** is the Eigenvector corresponding to smallest Eigenvalue*

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

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

QR decomposition

Geometric camera calibration (Tsai's algorithm!)

Given a set of matched points

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

point in 3D space　　point in the image

Where do we get these matched points from?

and camera model

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

projection model　　parameters　　Camera matrix

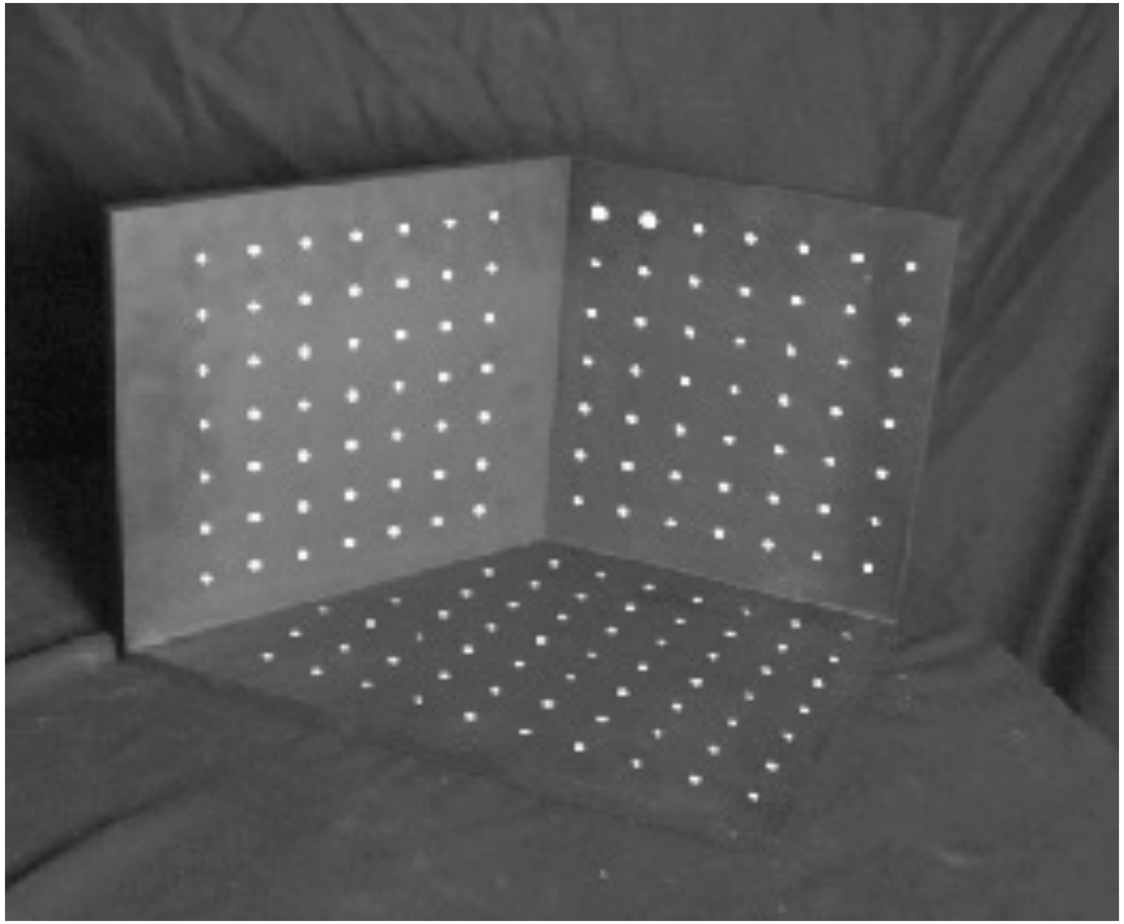
Find the (pose) estimate of

$$\mathbf{P}$$

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:

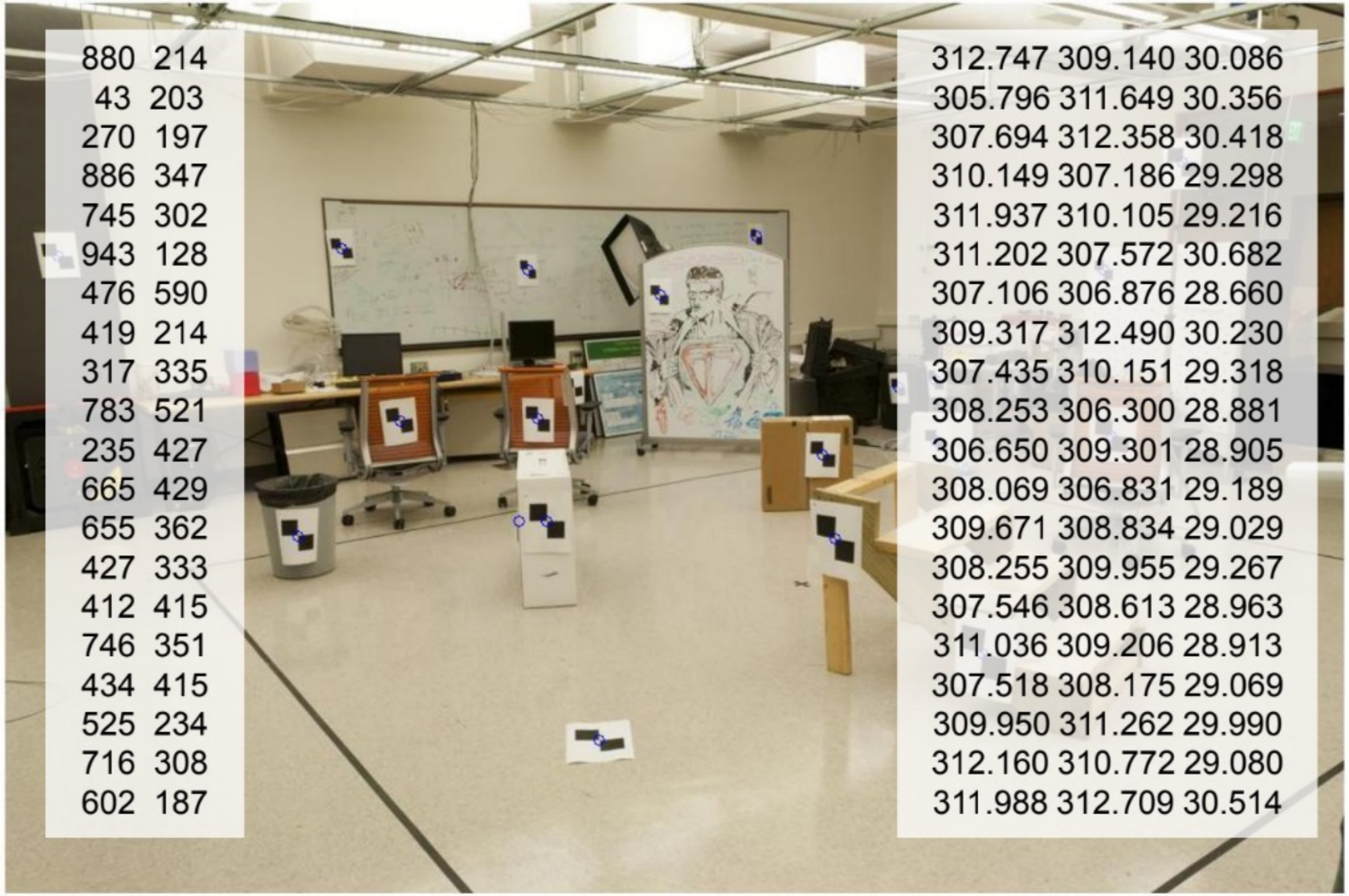
- Doesn’t model radial distortion.
- Hard to impose constraints (e.g., known *f*).
- Doesn’t minimize the correct error function.

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

Known 2d image coords

Known 3d world locations



880	214
43	203
270	197
886	347
745	302
943	128
476	590
419	214
317	335
783	521
235	427
665	429
655	362
427	333
412	415
746	351
434	415
525	234
716	308
602	187

312.747	309.140	30.086
305.796	311.649	30.356
307.694	312.358	30.418
310.149	307.186	29.298
311.937	310.105	29.216
311.202	307.572	30.682
307.106	306.876	28.660
309.317	312.490	30.230
307.435	310.151	29.318
308.253	306.300	28.881
306.650	309.301	28.905
308.069	306.831	29.189
309.671	308.834	29.029
308.255	309.955	29.267
307.546	308.613	28.963
311.036	309.206	28.913
307.518	308.175	29.069
309.950	311.262	29.990
312.160	310.772	29.080
311.988	312.709	30.514

Known 2d image cords (px)

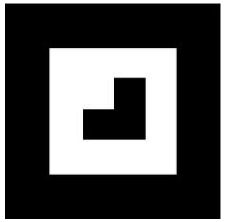
Known 3d world locations (m)

M

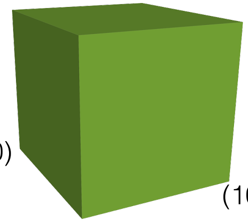
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Unknown Camera Parameters

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 **P**

3. Project 3D content to image plane using **P**

More Advance Calibration using Multiple Views….

https://grandvisual.com/work/pepsi-max-bus-shelter/ (London, 2014)

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering