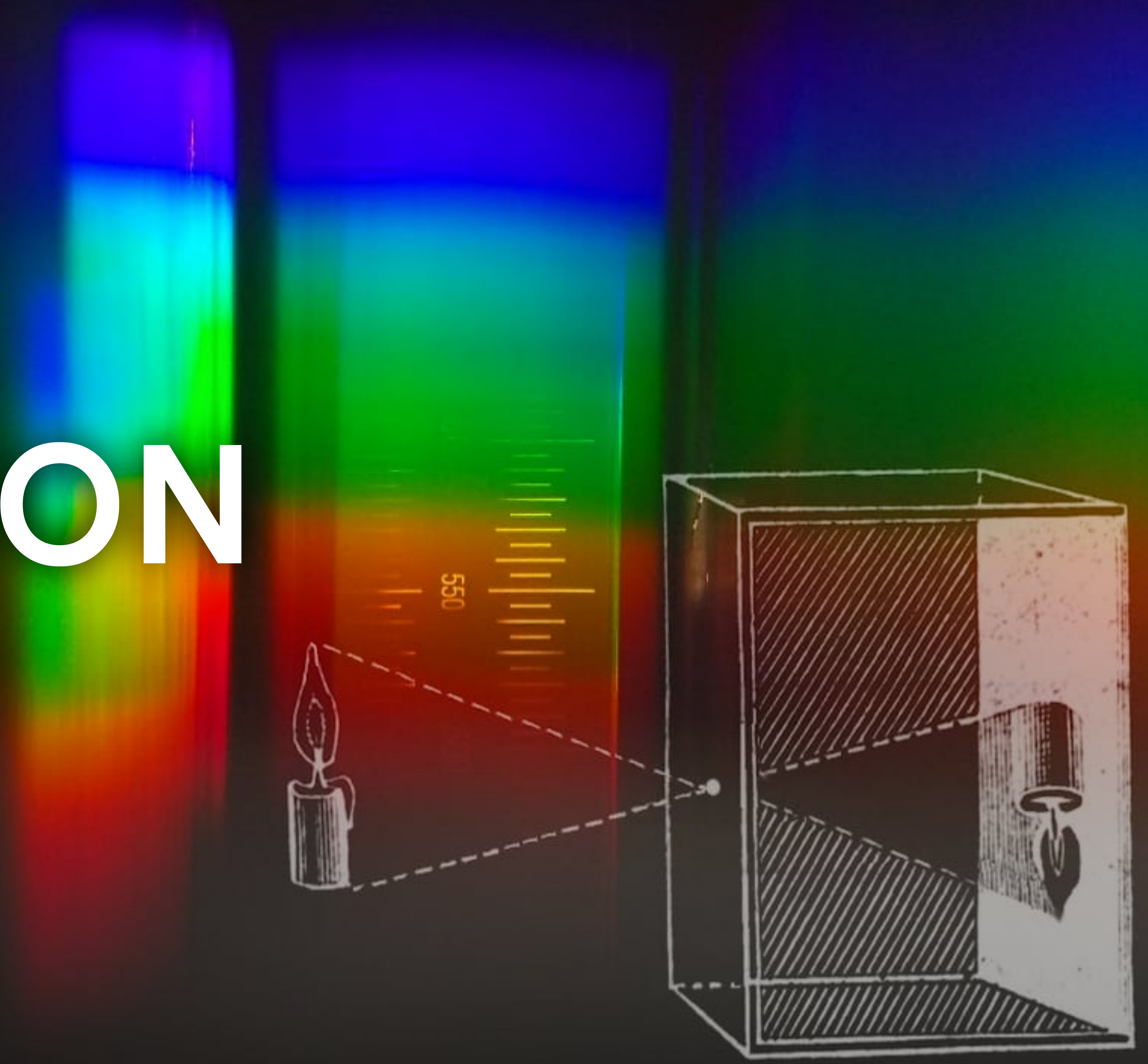
INVERSE GRAPHICS

MODULE 1:
Fundamentals of Image Formation

IMAGE FORMATION



6.S980

Update on Enrollment

- **For-credit (40):** Will randomly sample from waitlist tonight to fill free spots. Notifications will be sent out tomorrow morning.
- **Listeners (20):** Will randomly sample from listener sign-ups. Notifications will be sent out tomorrow morning.
- Asked for a bigger room to accommodate more listeners, but none available :/
- Slides, recordings, assignments, and everything else will be made available publicly - so everyone has the same access as the “official” listeners!
- My goal for next year: Bigger room, course instead of seminar -> better accreditation.

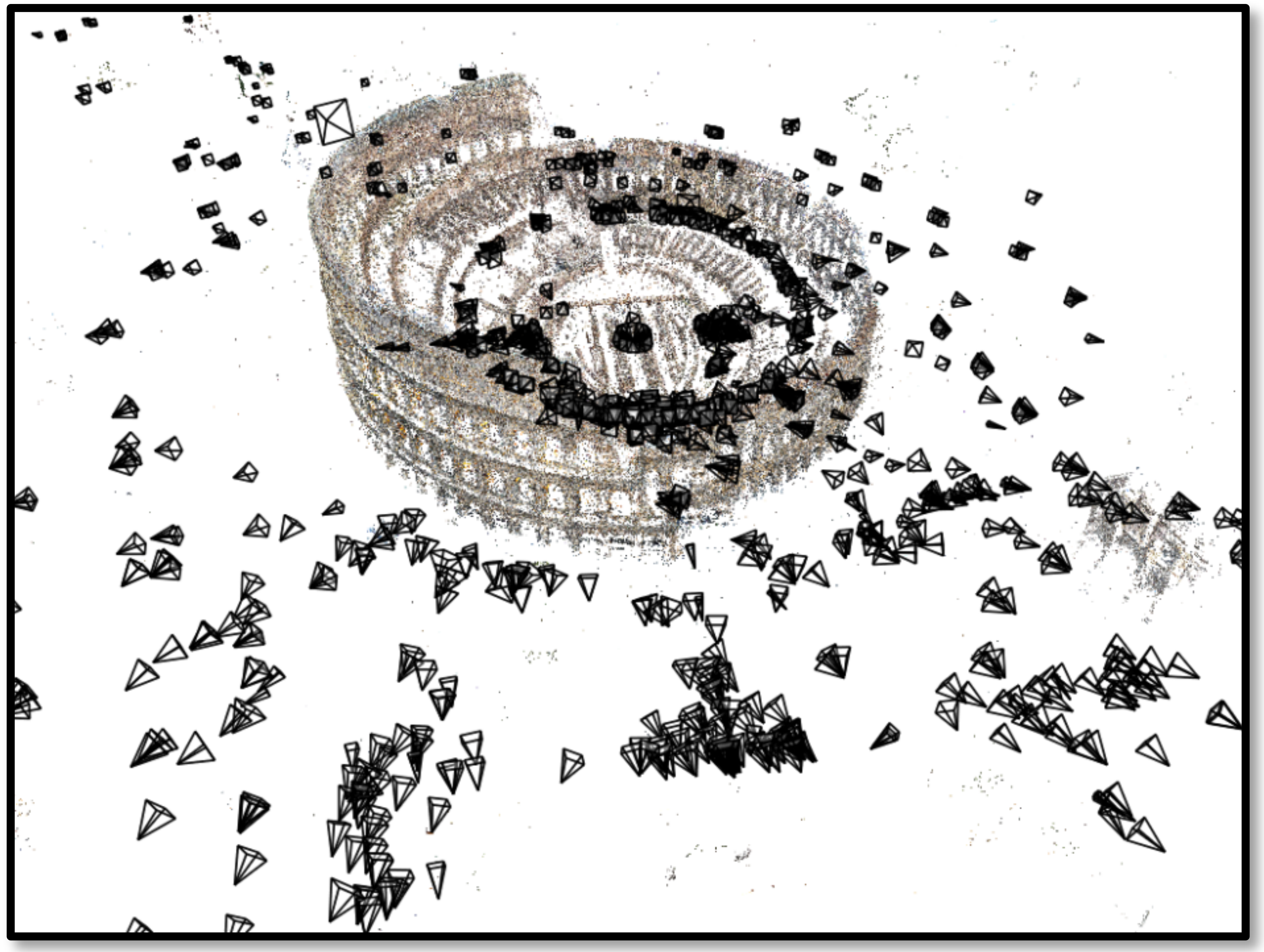
Other Admin Things

- Find calendar to subscribe to on course website.
- First lecture recording posted.
- Slides posted.
- First assignment posted later today!
  Note: Very long, but not much code to write at all!

Course Project Clarifications

- **Teams of 1-3.**
- **Counts for 30% of final grade.**
  - Proposal (5%)
  - 3-minute Mid-term Presentation (5%)
  - Final Report + Presentation (20%)
  - Late days for proposal, but not for final report.
- **See course website for up-to-date deadlines. Summary:**
  - Proposal due Oct 18.
  - Project Mid-term Update Presentation (3 min per team) in Nov. 3rd lecture.
  - Final project presentations (~5-10 minutes per team) on Dec 8th and Dec 13th.

Module 1: Image Formation and Multi-View Geometry



Why?	We want to understand 3D world only from 2D observations (images). For that, we need to have a mathematical understanding of how they are connected.
What you'll learn.	Mathematical model of cameras. Reconstruct camera poses, approximate geometry, and camera parameters from 2D images of a scene.

Image credits: *Noah Snavely*

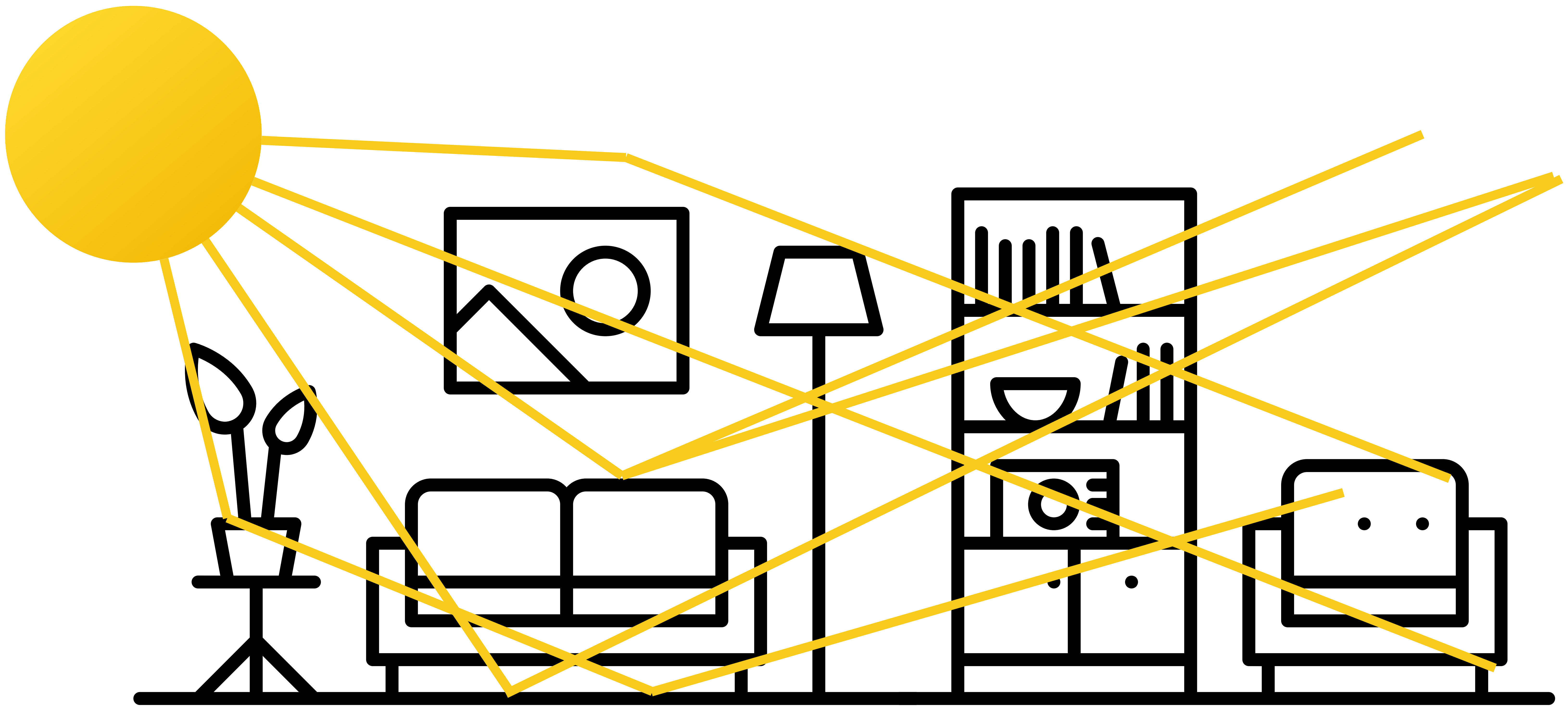
Some Slides adapted from…

- CMU 16-889: Learning for 3D Vision
  Prof. Shubham Tulsiani
- CMU 16-385: Computer Vision
  Prof. Kris Kitani
- MIT 6.819/6.869: Advances in Computer Vision,
  Profs. Bill Freeman, Phillip Isola, Antonio Torralba

What is a 3D scene?

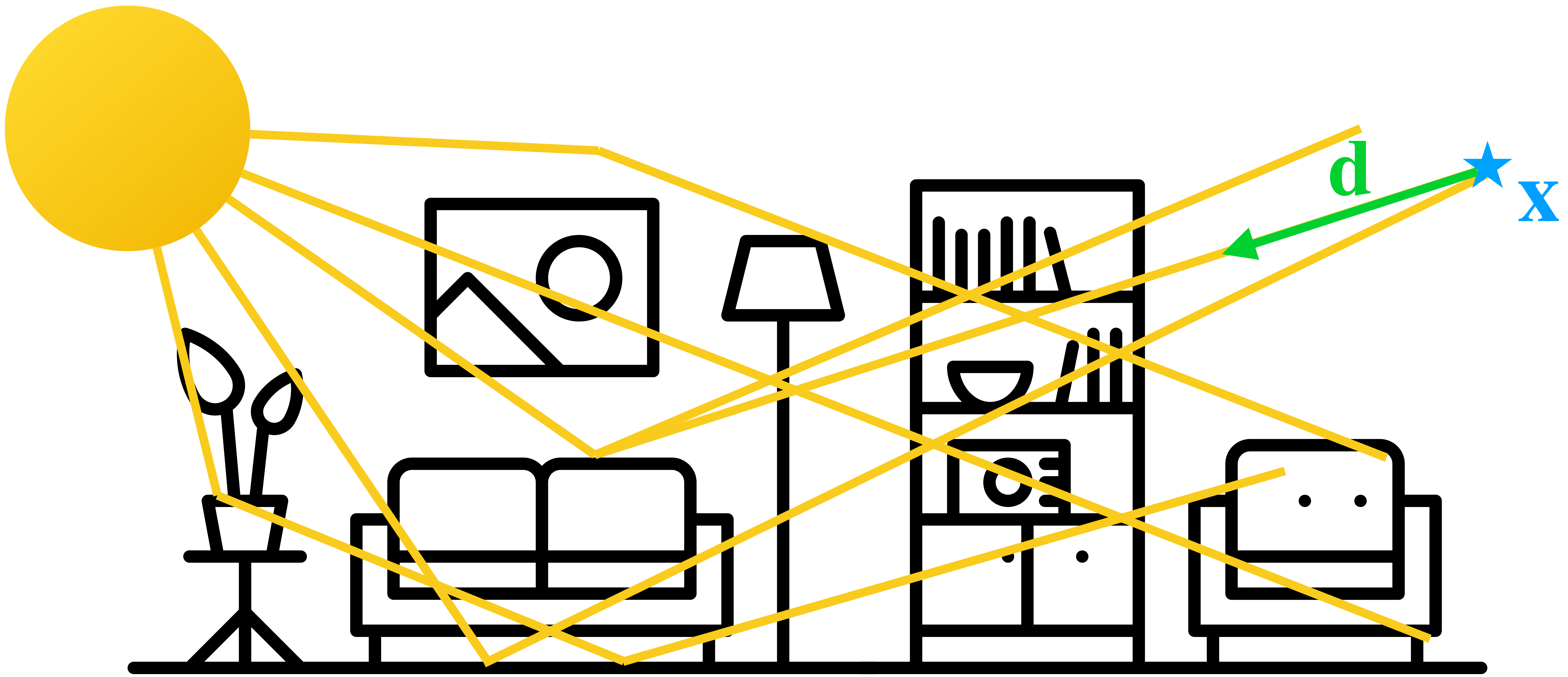
materials, light sources, 3D shape, color, weight, density, friction coefficients, etc

How do we observe scenes?



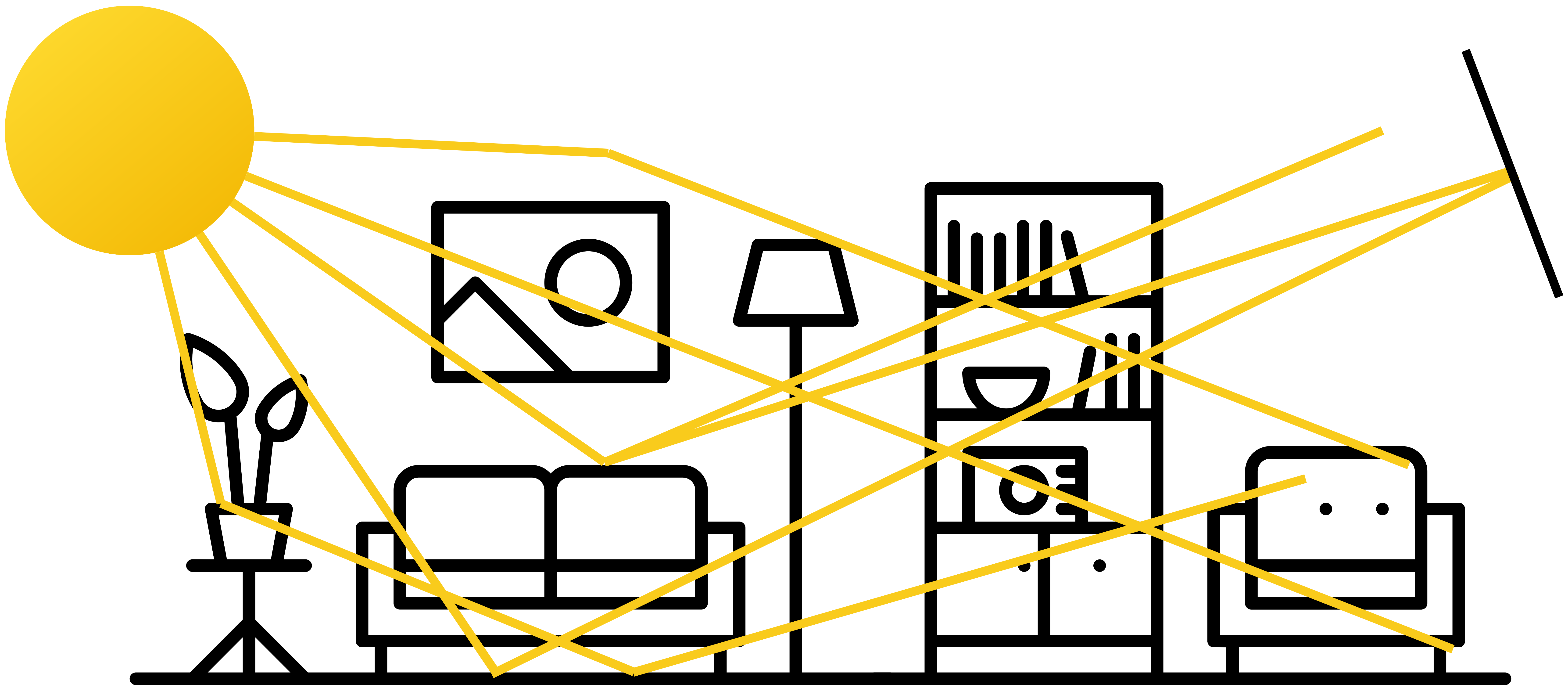
An eye (or a camera) observes a subset of all the light rays in a scene.

The Light Field: 3D coordinate plus ray direction is mapped to the color of that ray.

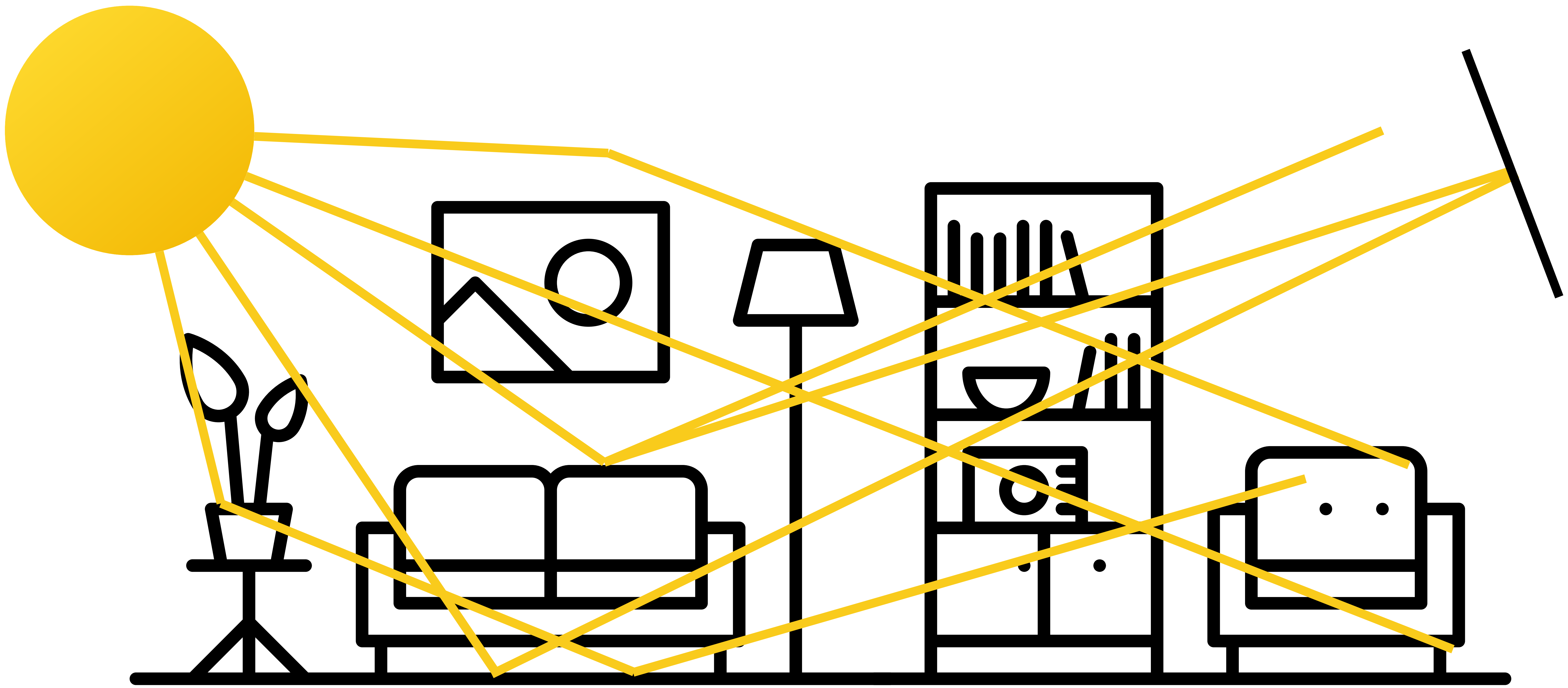


$$LF : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3, \quad LF(\mathbf{x}, \mathbf{d}) = \mathbf{c}$$

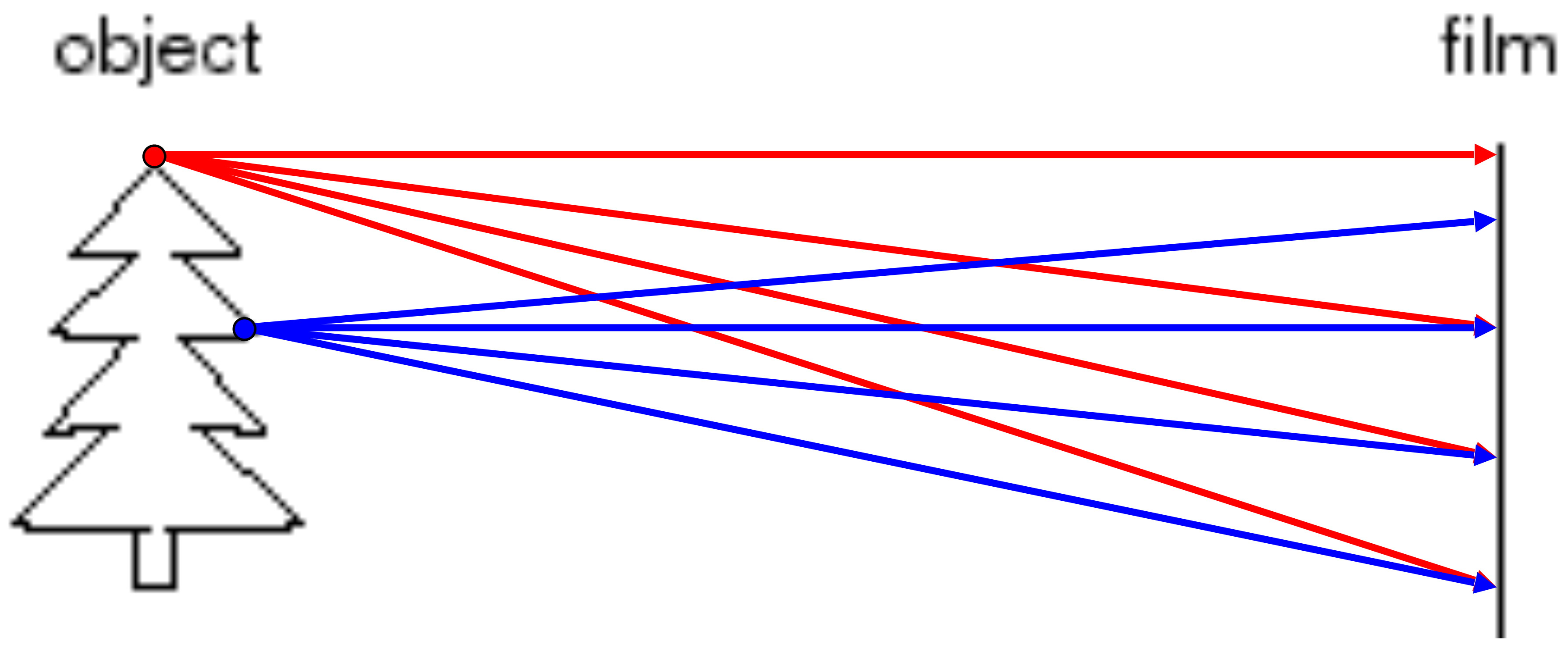
Why don’t we get an image if we hold up a piece of paper?



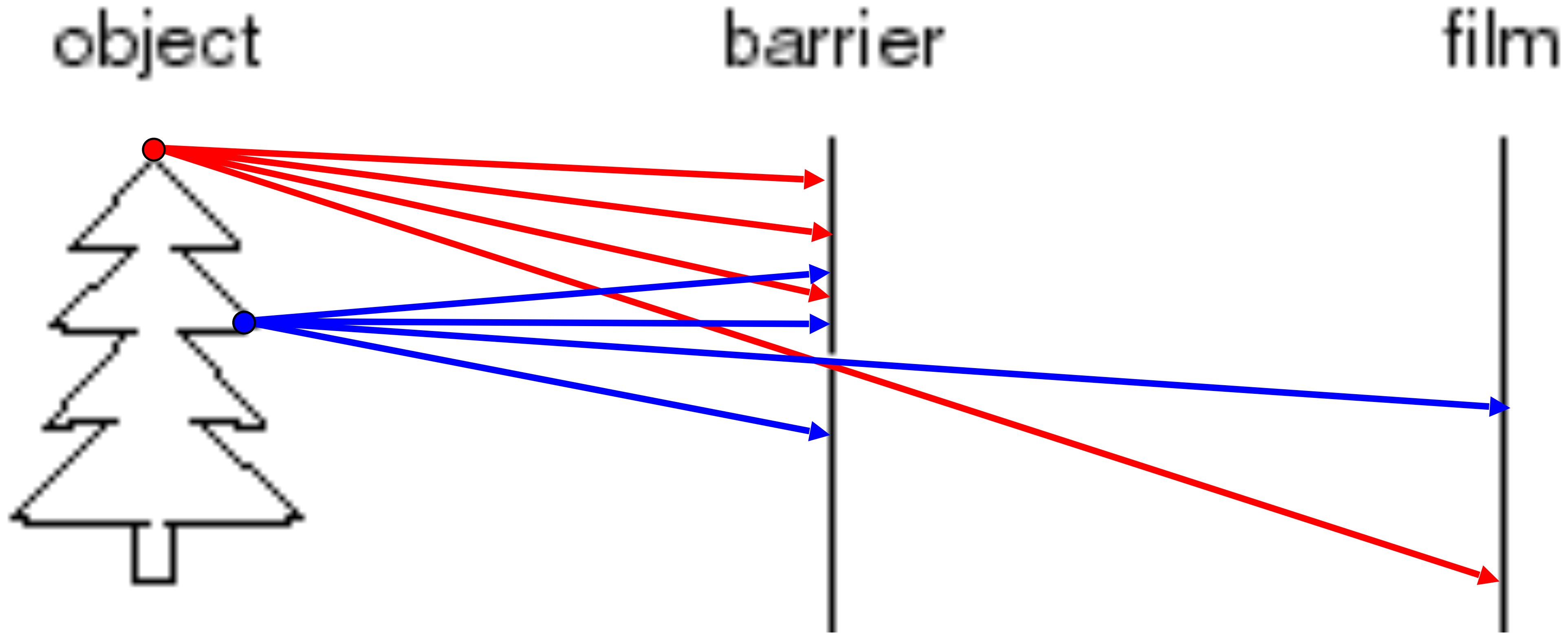
Every “pixel” on the paper is the average of all the rays in the scene.



Let’s make a Camera!



Idea 1: Put piece of film in front of an object



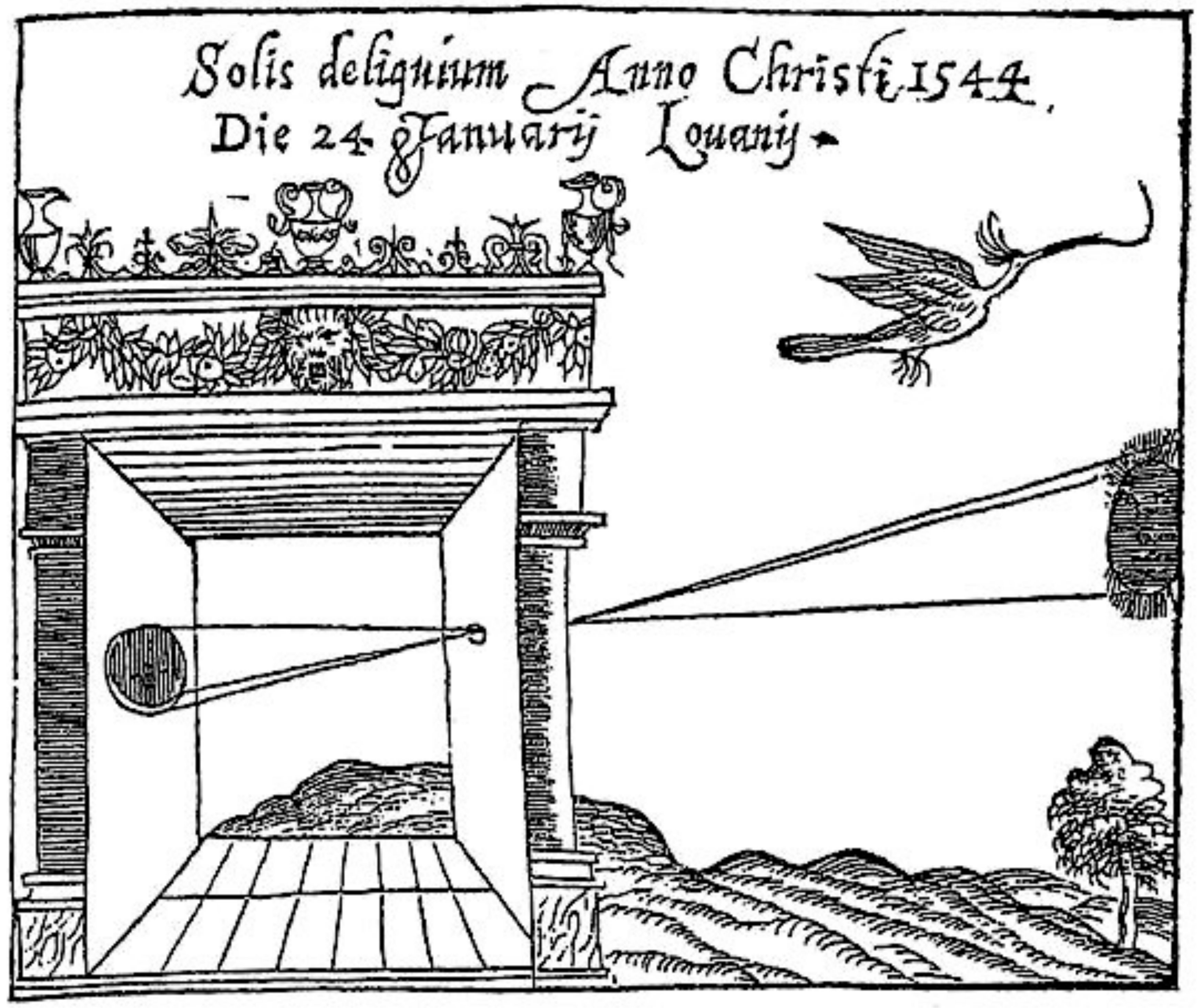
Add a barrier to block most rays.

Slide credit: Steve Seitz

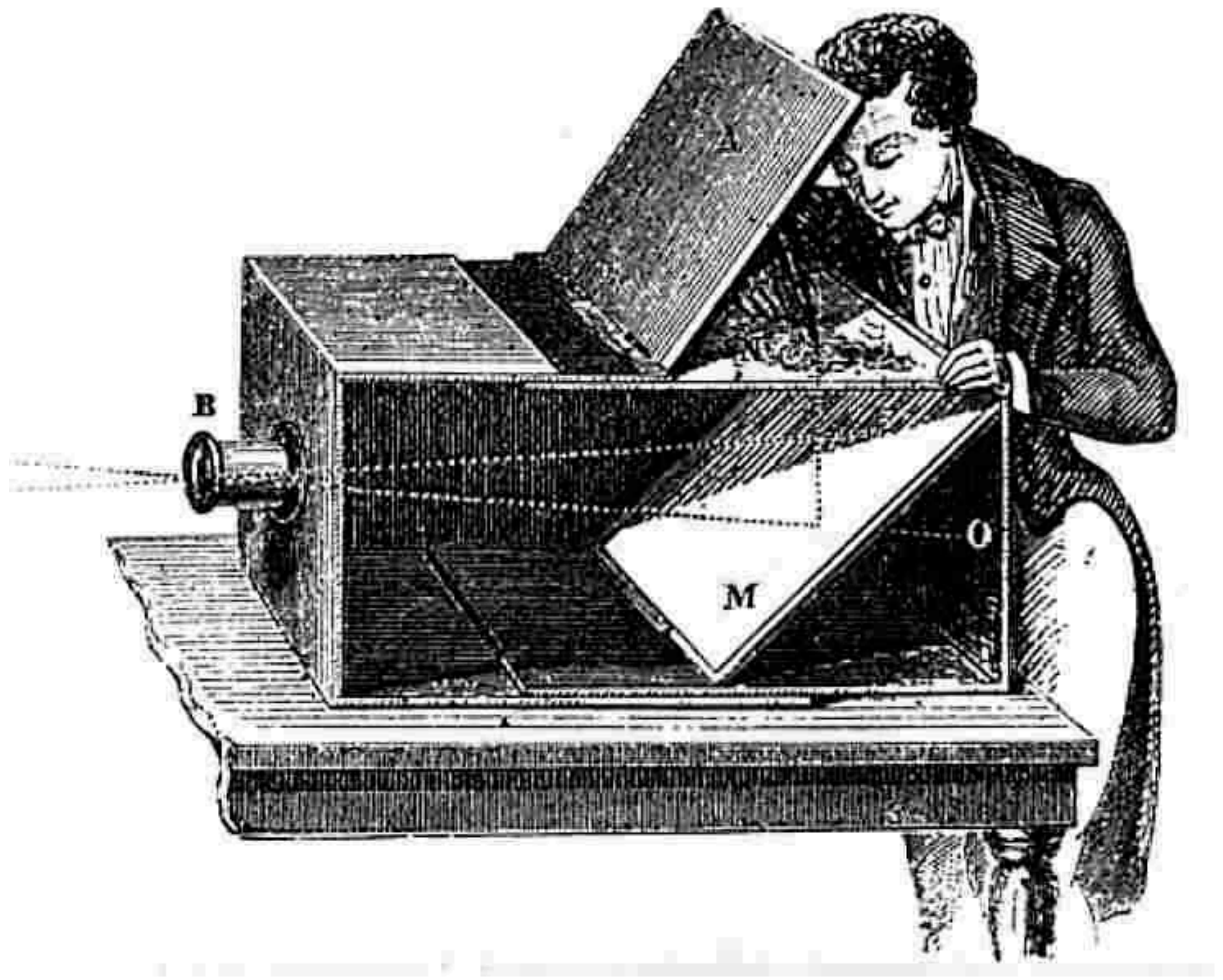
Camera Obscura

- Basic principle known to Mozi (470-390 BCE), Aristotle (384-322 BCE)

- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)



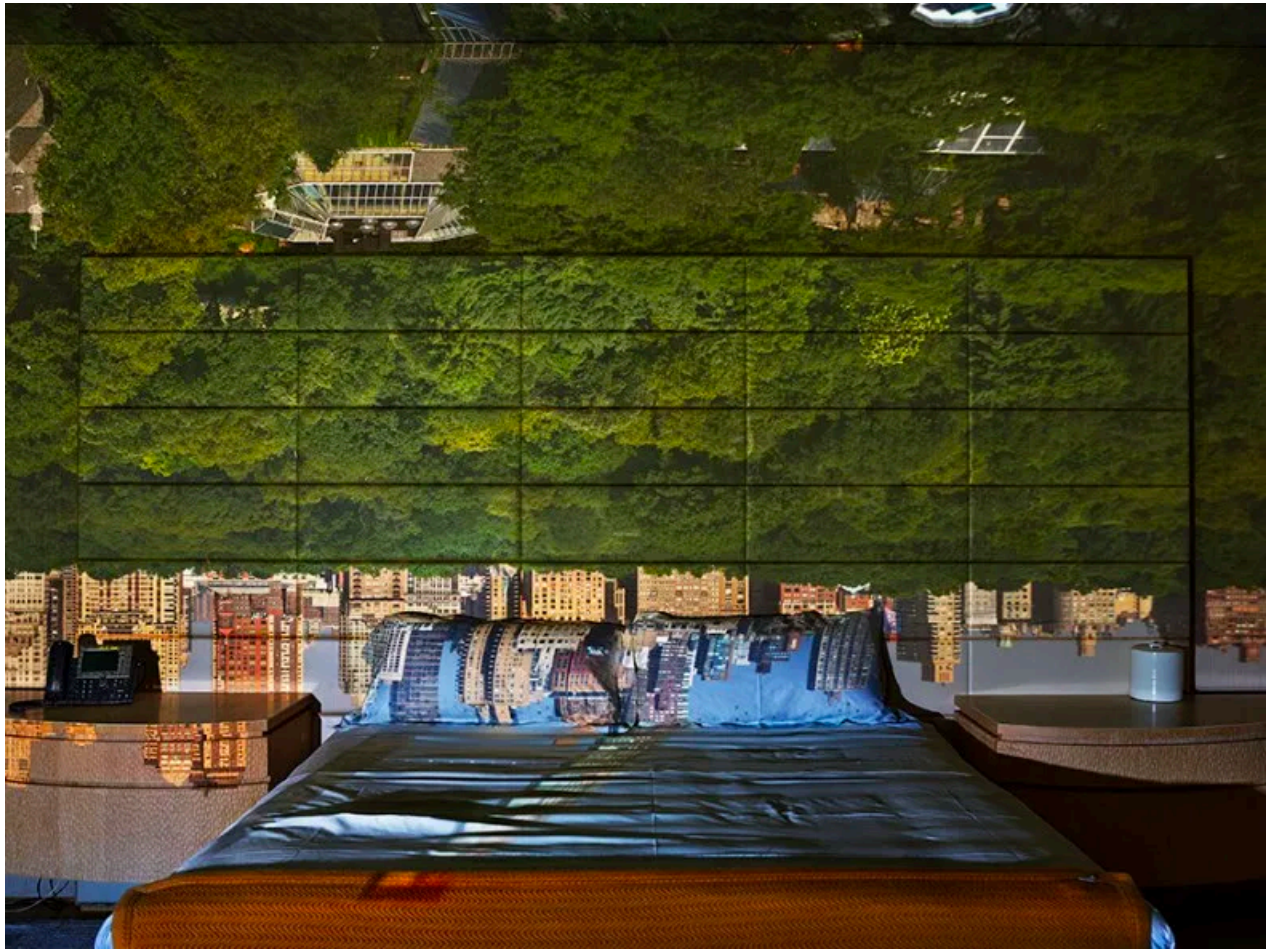
Gemma Frisius, 1558



Slide credit: Lana Lazebnik

Image source

Camera Obscura (*Dark Room*)



Camera Obscura: View of Central Park Looking West in Bedroom. Summer, 2018

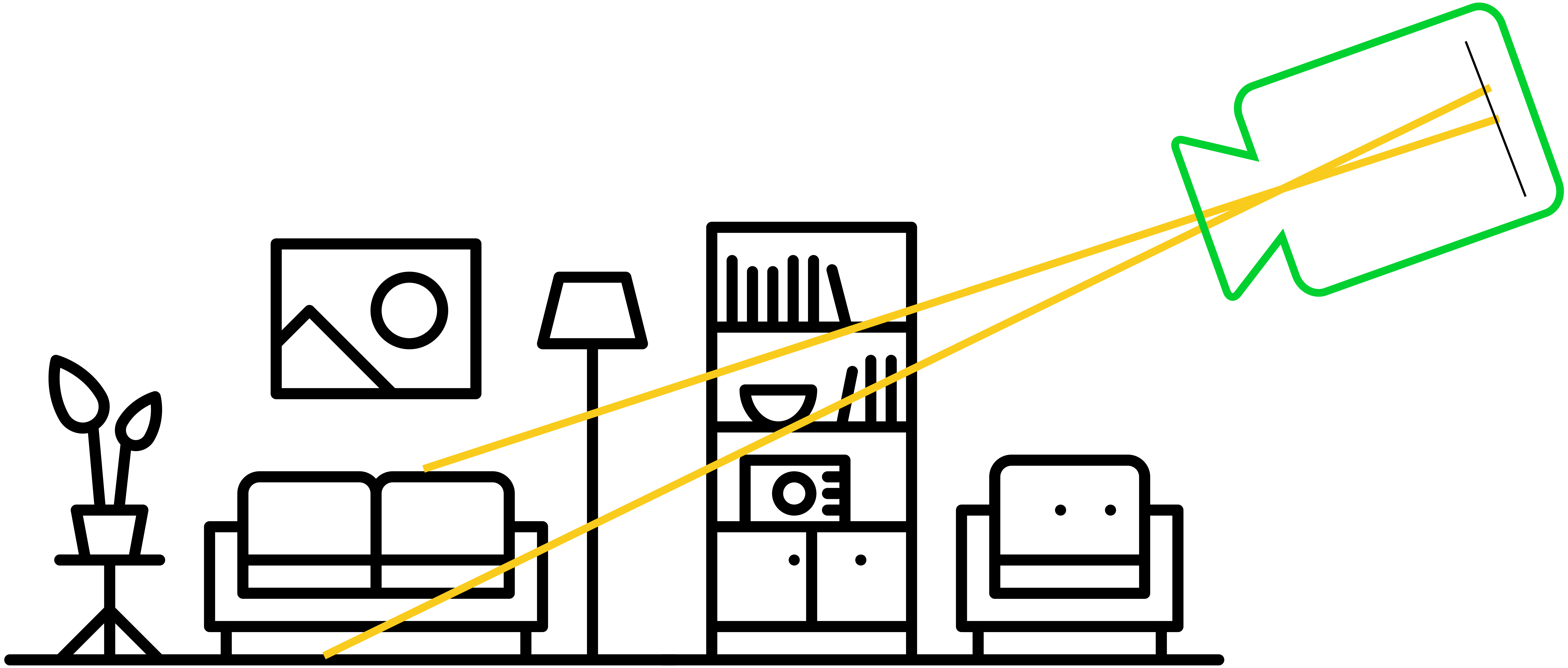
After scouting rooms and reserving one for at least a day, Morell masks the windows except for the aperture. He controls three elements: the size of the hole, with a smaller one yielding a sharper but dimmer image; the length of the exposure, usually eight hours; and the distance from the hole to the surface on which the outside image falls and which he will photograph. He used 4 x 5 and 8 x 10 view cameras and lenses ranging from 75 to 150 mm.

After he's done inside, it gets harder. "I leave the room and I am constantly checking the weather, I'm hoping the maid reads my note not to come in, I'm worrying that the sun will hit the plastic masking and it will fall down, or that I didn't trigger the lens."

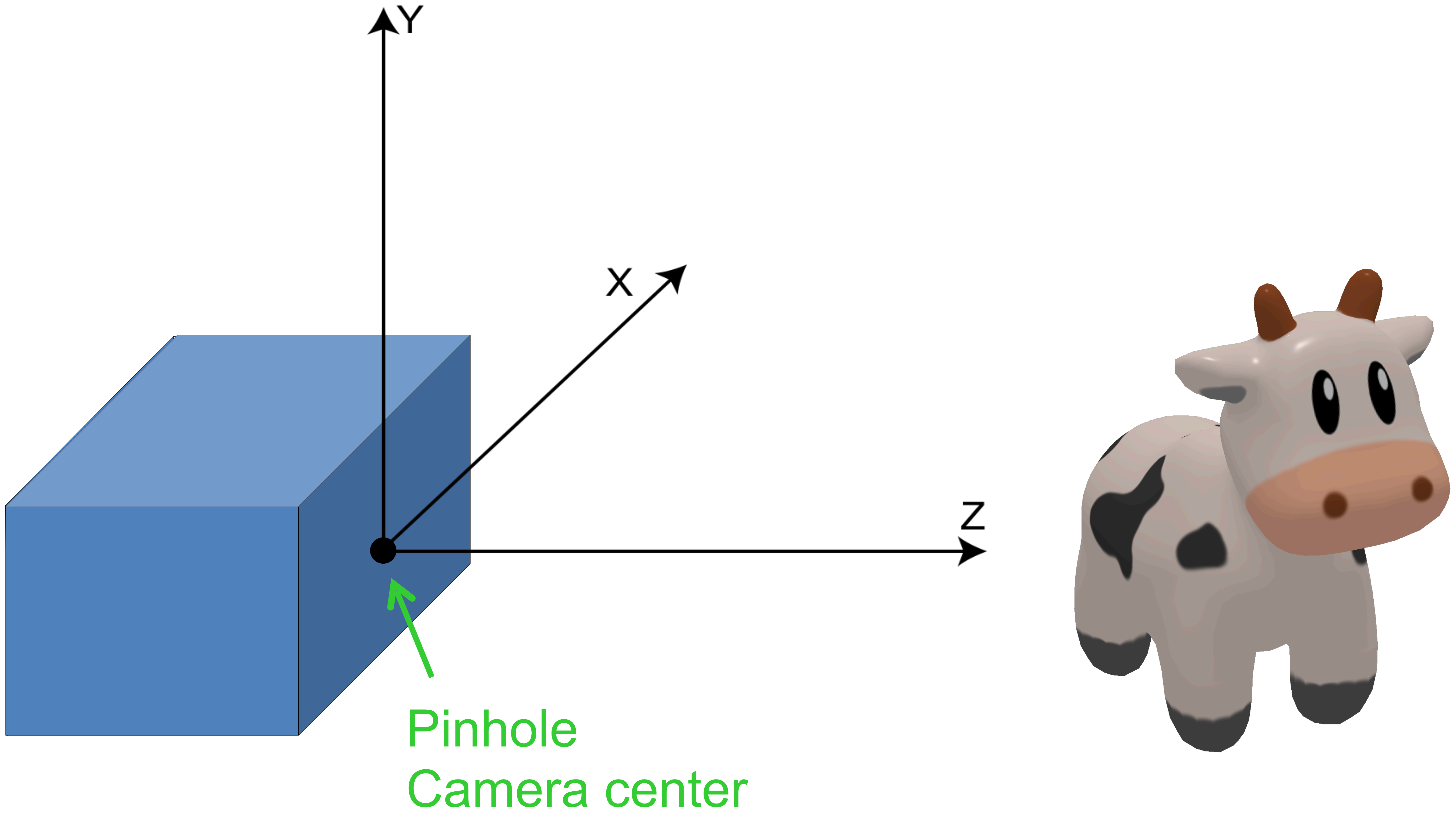
http://www.abelardomorell.net/project/camera-obscura/

Slide credit: Lana Lazebnik

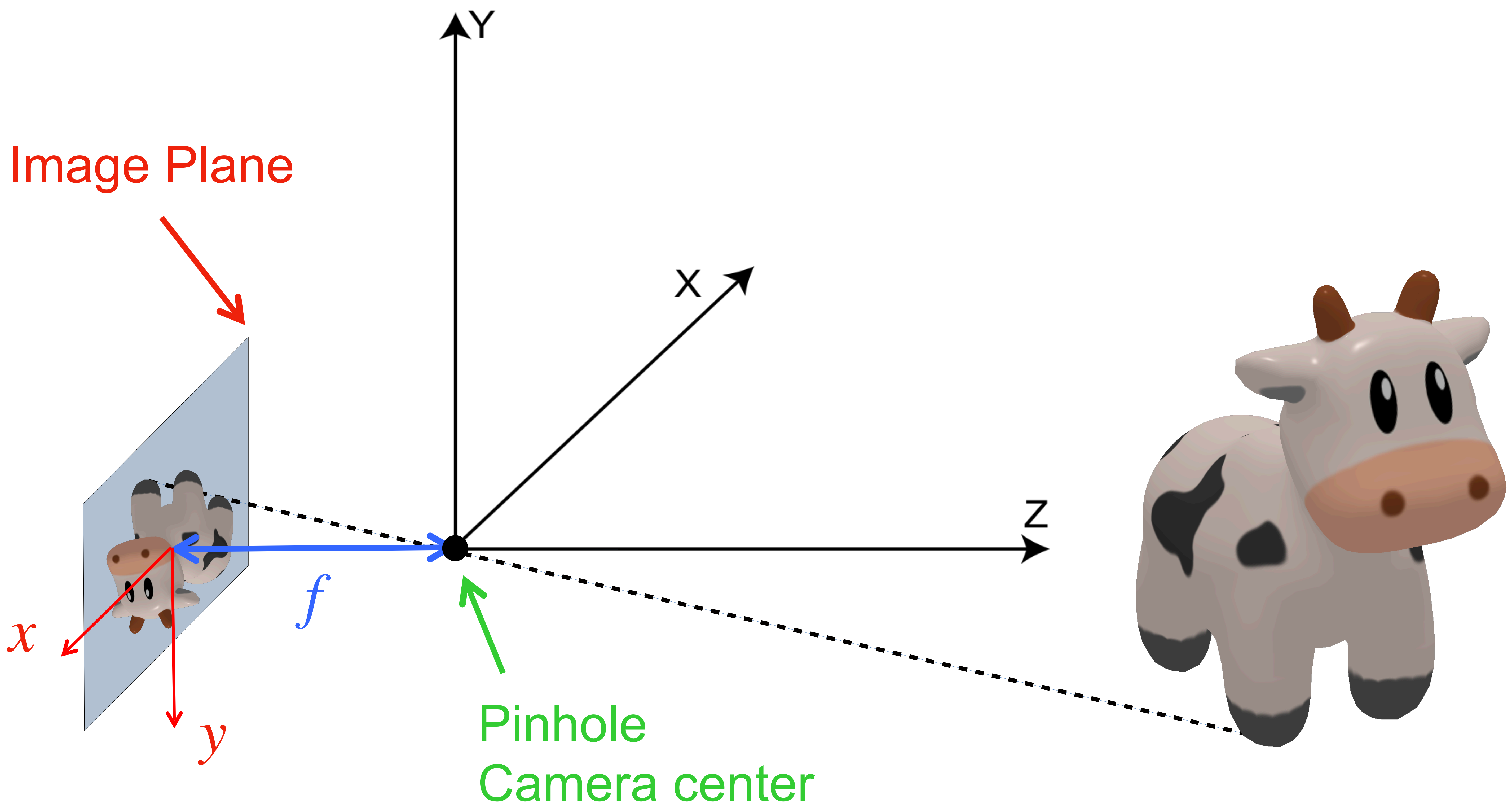
Today: Model the camera, describe the rays.



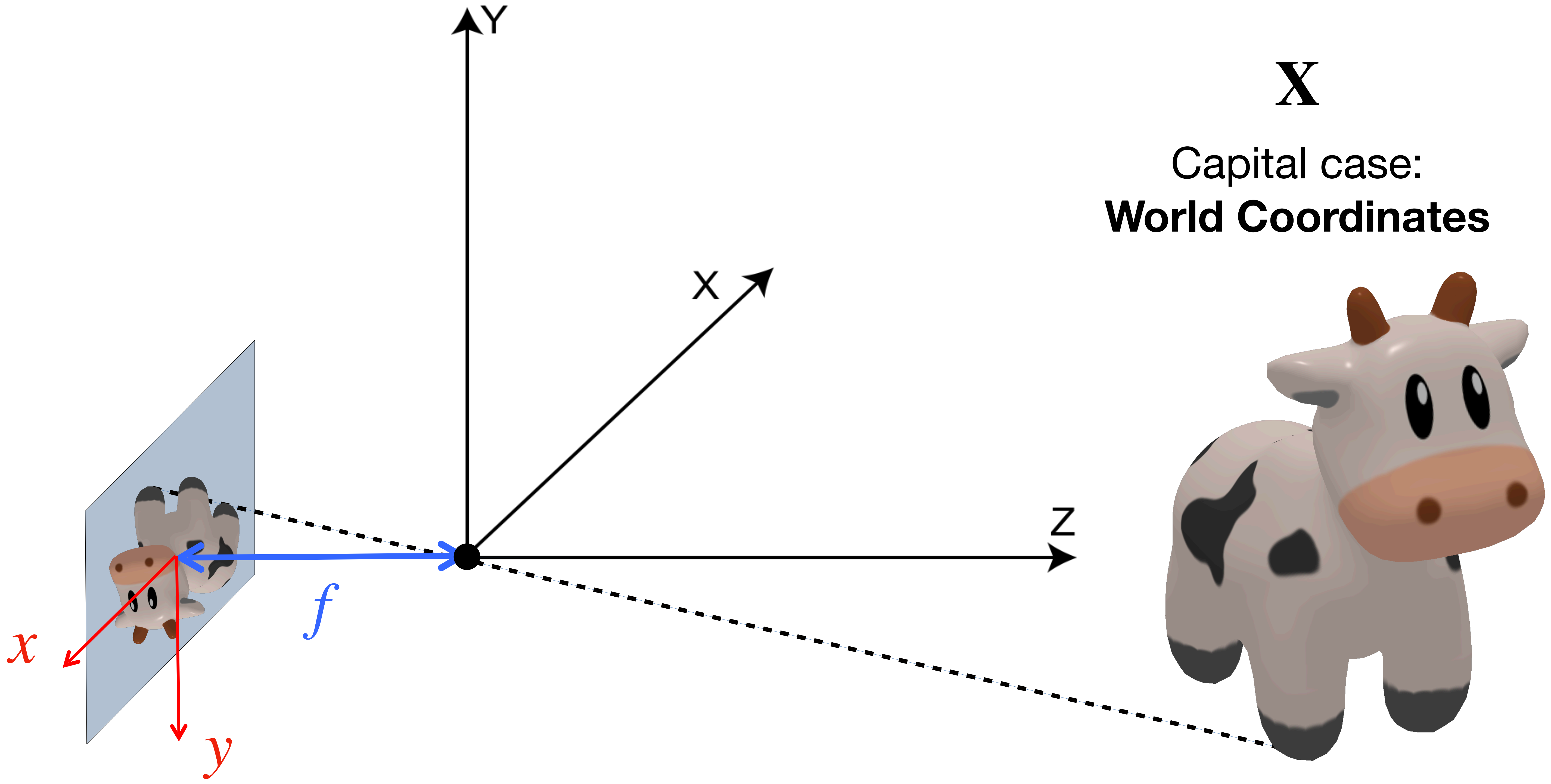
Perspective projection



Perspective projection



Perspective projection



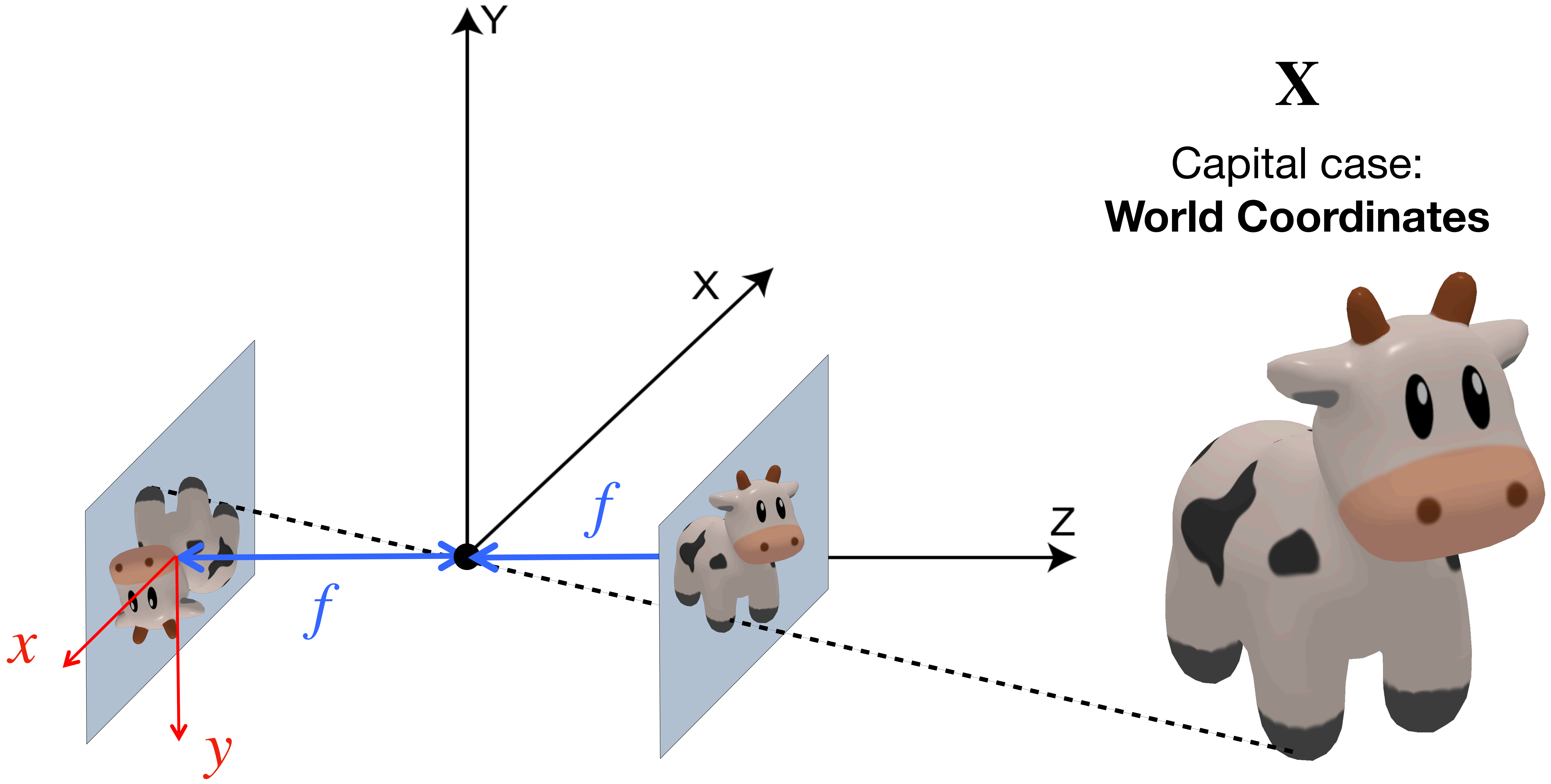
$\mathbf{X}$

Capital case:
World Coordinates

$\mathbf{x}$

Lowercase:
Pixel / Image Coordinates

Perspective projection



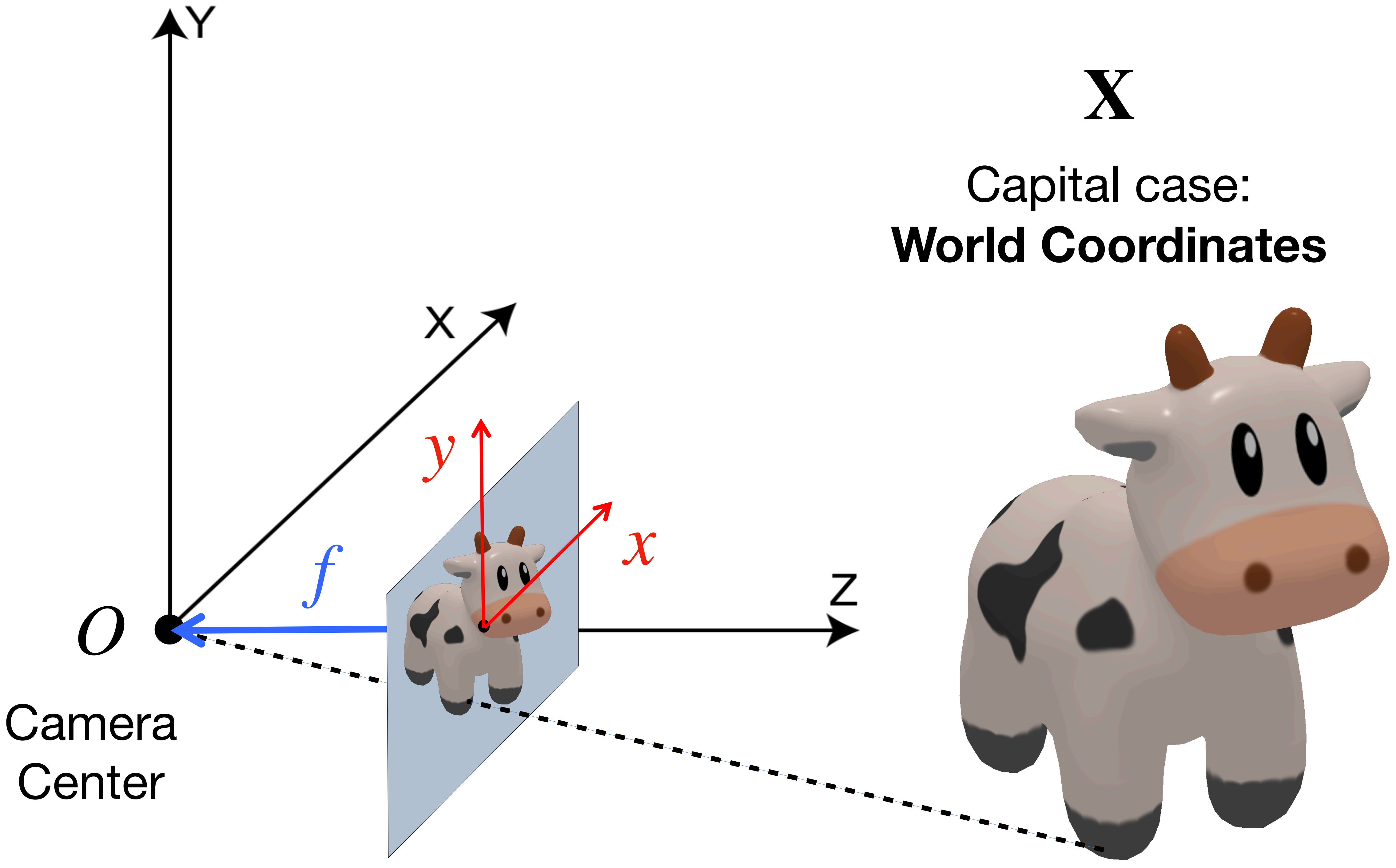
$\mathbf{X}$

Capital case:
World Coordinates

$\mathbf{x}$

Lowercase:
Pixel / Image Coordinates

Perspective projection

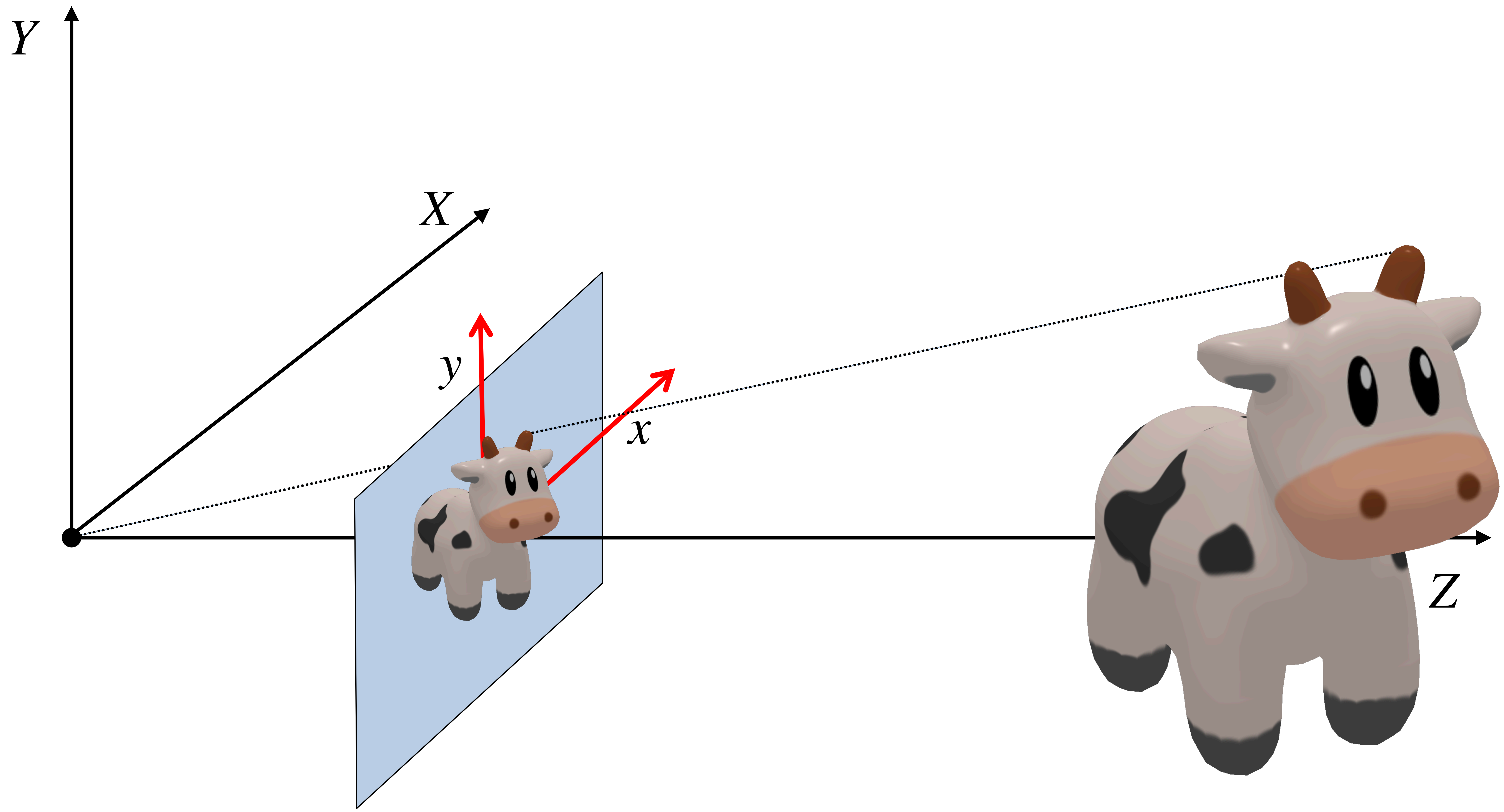


$\mathbf{x}$

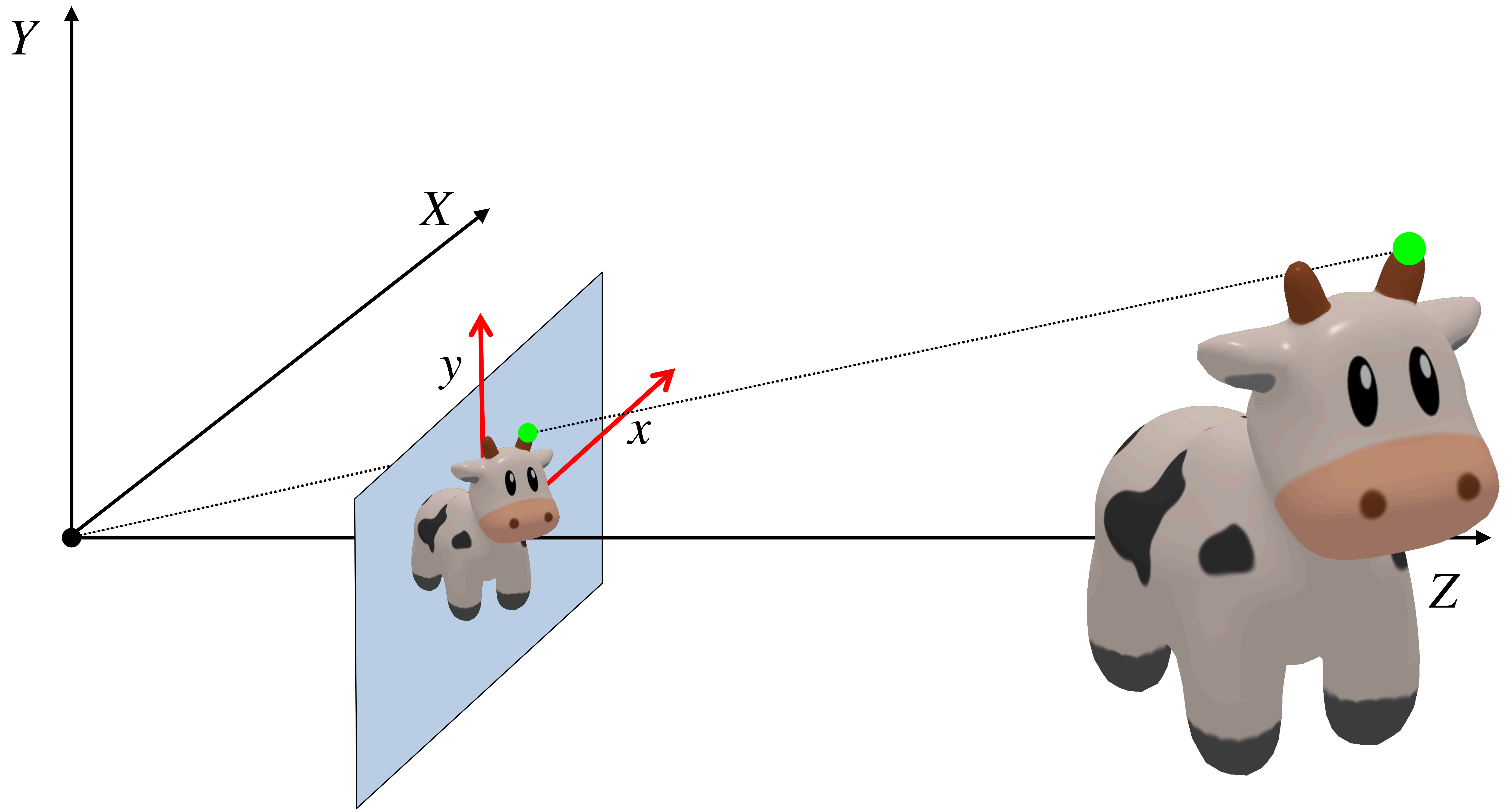
Lowercase:

Pixel / Image Coordinates

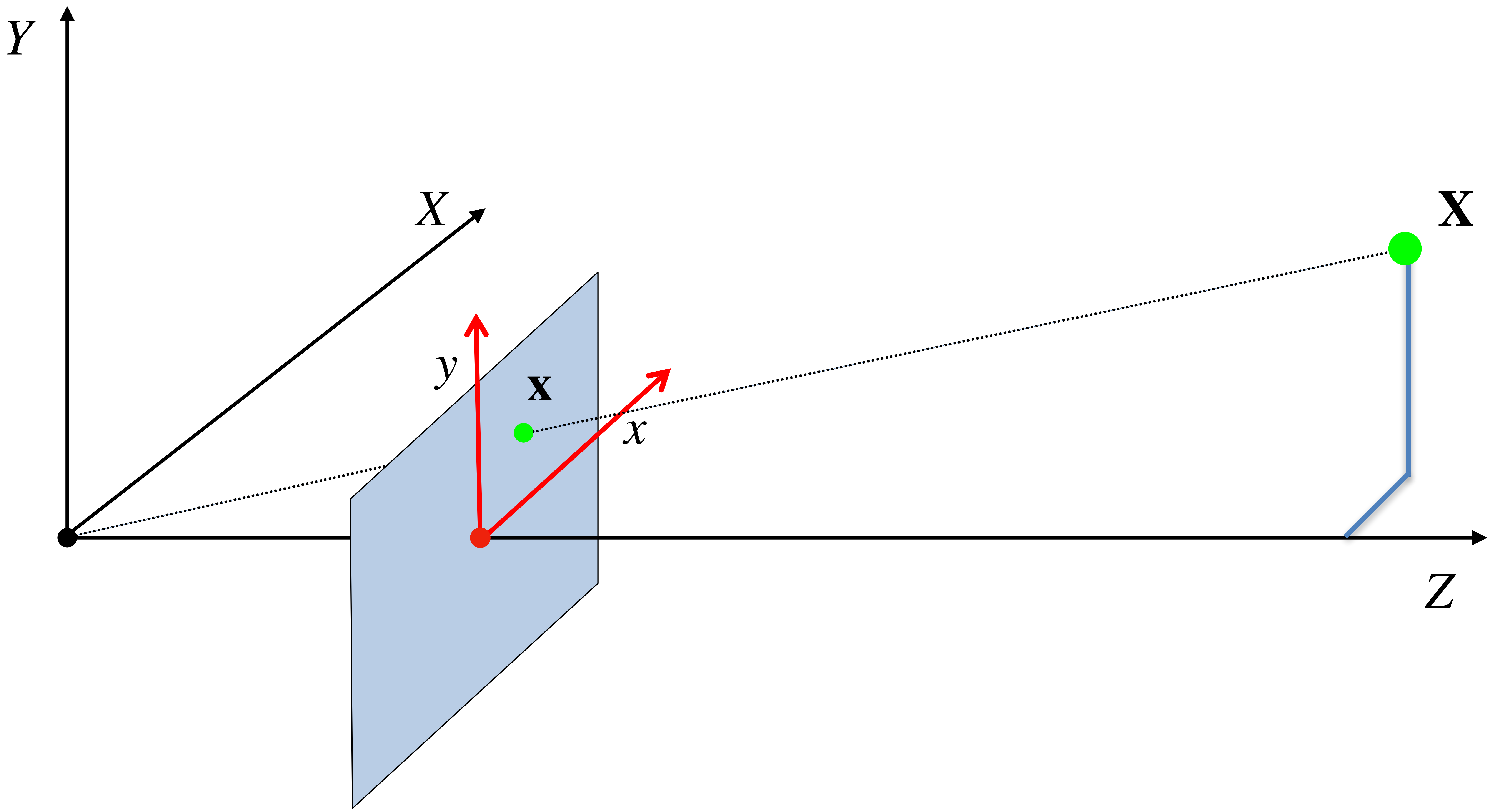
Perspective projection



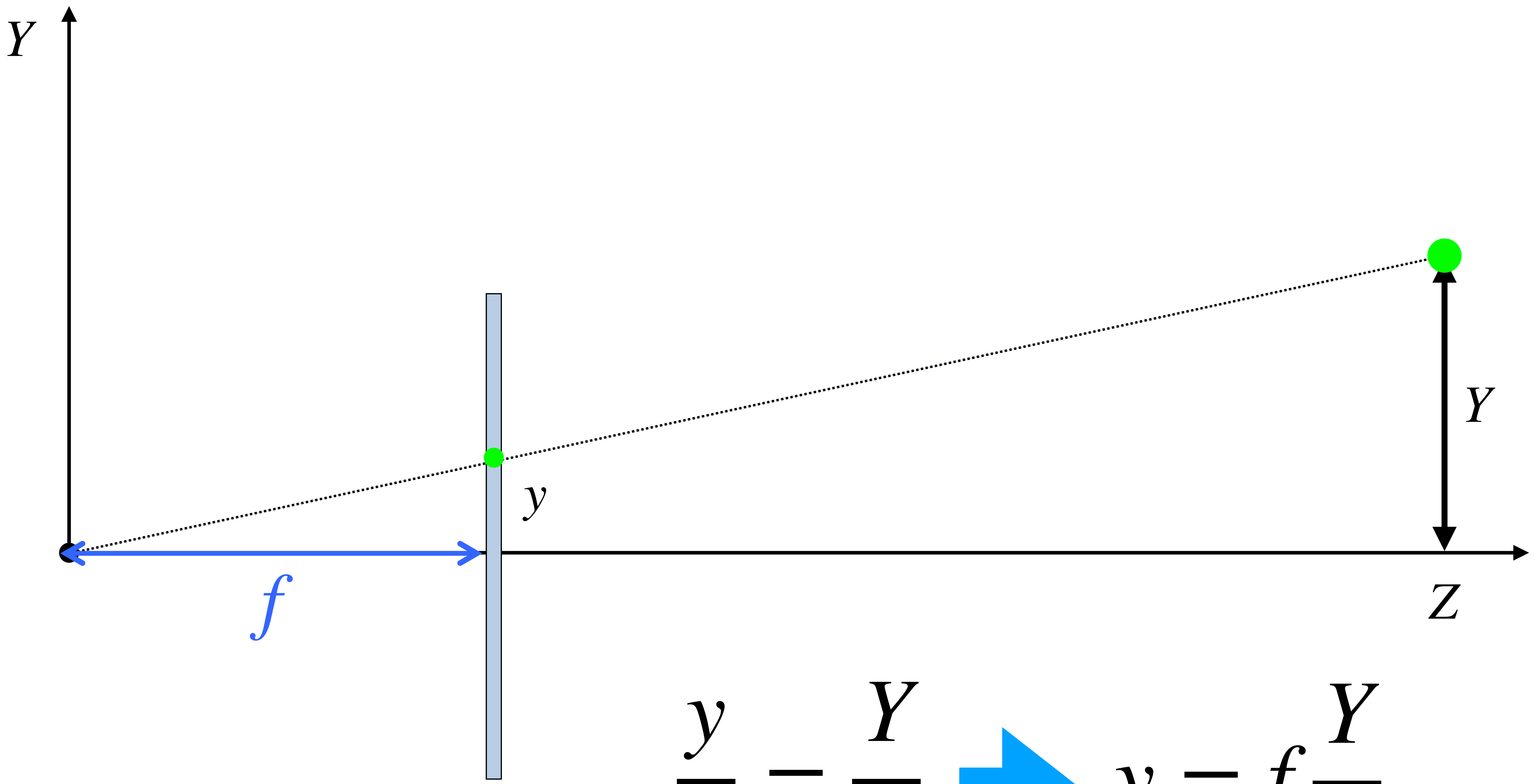
Perspective projection



Perspective projection

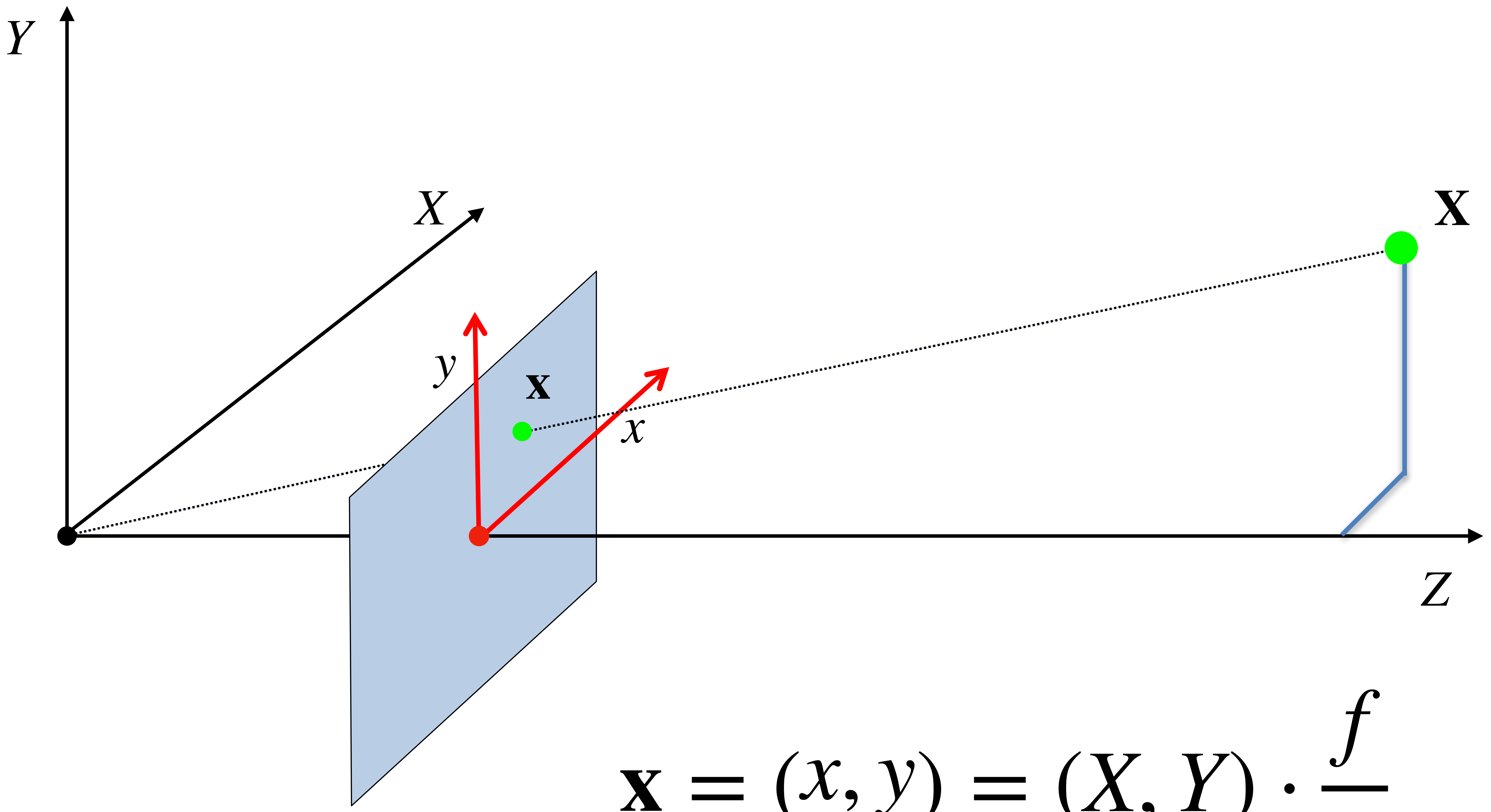


Perspective projection



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

Perspective projection



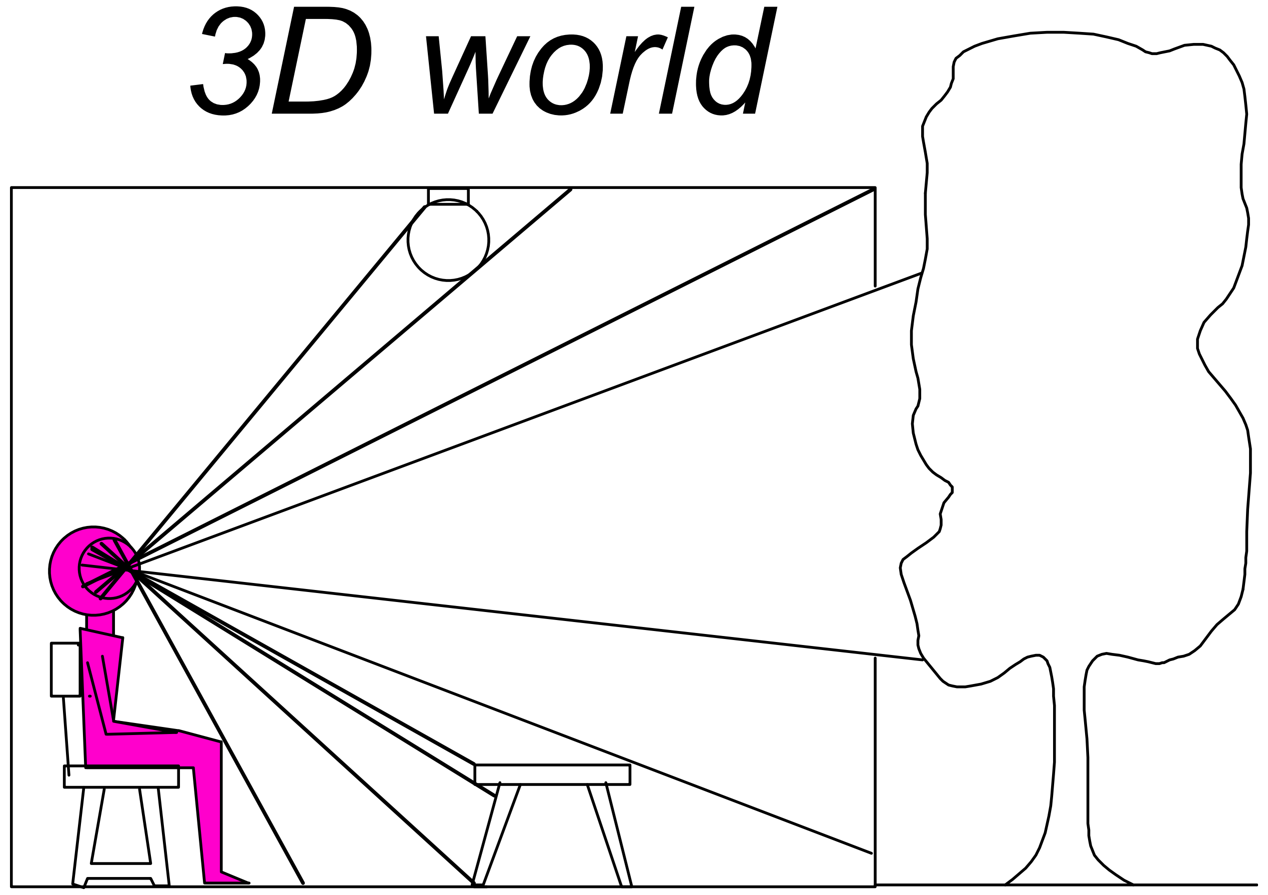
$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{f}{Z}$$

Image / Pixel Coordinates

World Coordinates

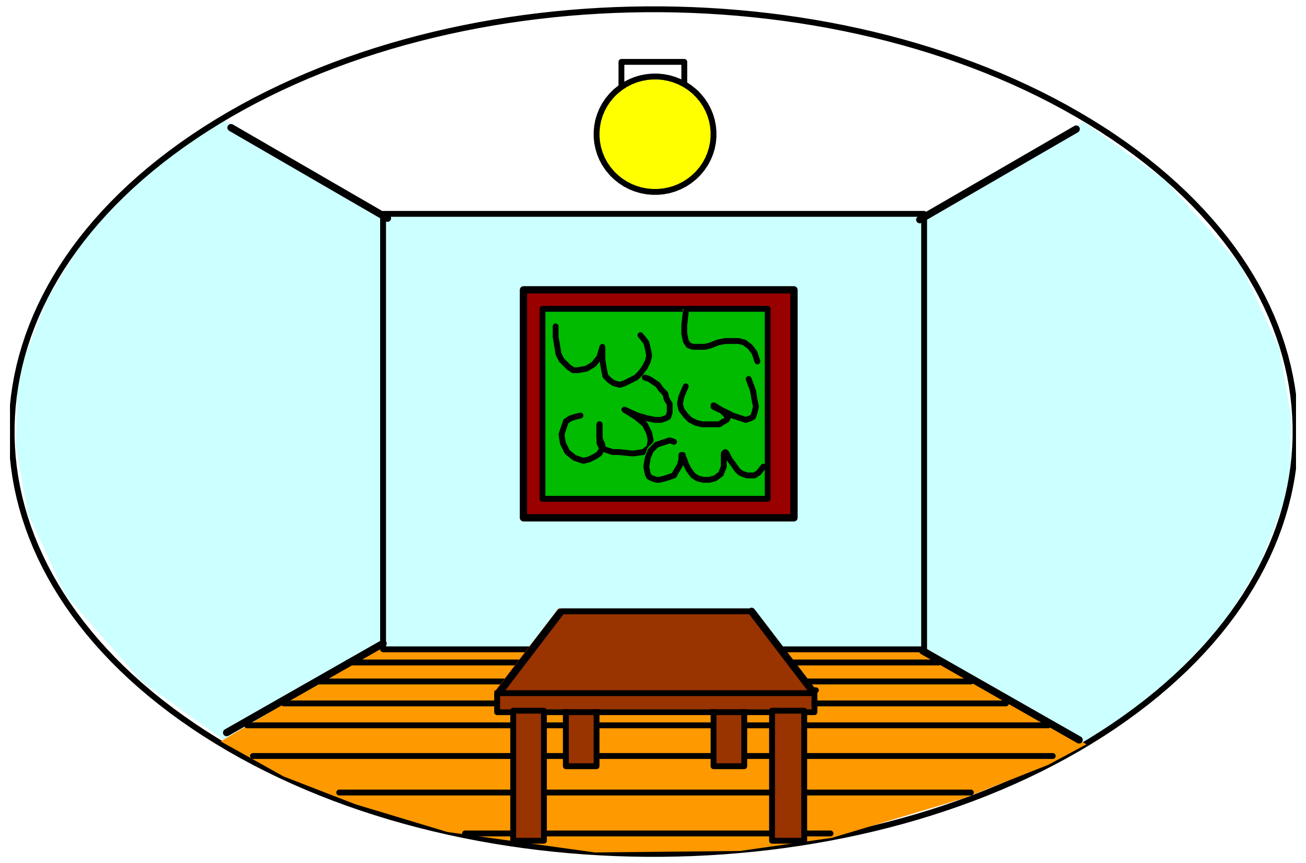
Perspective Projection

3D world



Point of observation

2D image



What properties of the world are preserved?

- Straight lines, incidence

What properties are not preserved?

- Angles, lengths

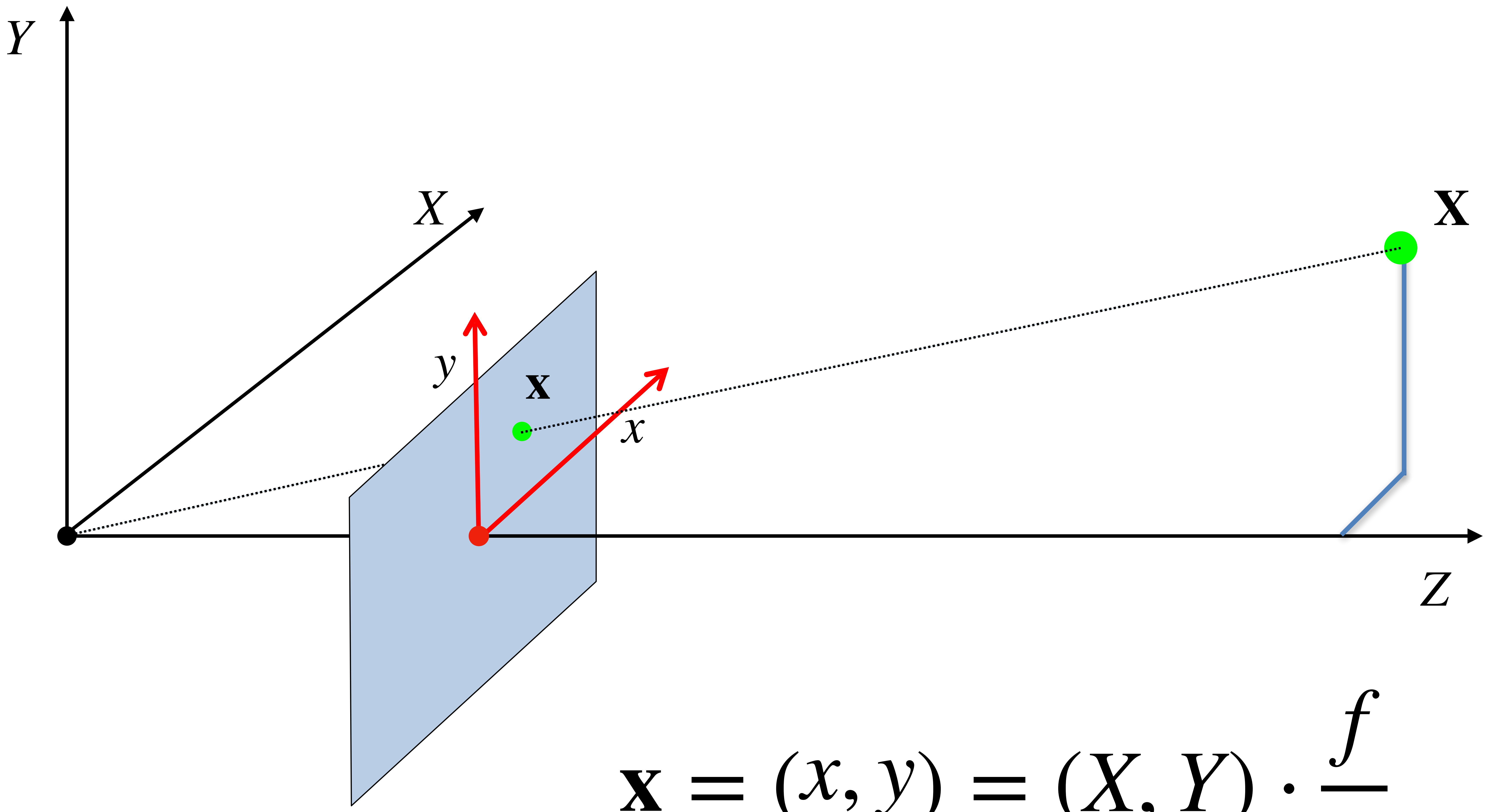
Perspective Projection

Video source: [Camera Obscura & World of Illusions]

Slide credit: CMU 16-889: Learning for 3D Vision, Prof. Shubham Tulsiani

Questions?

Perspective projection

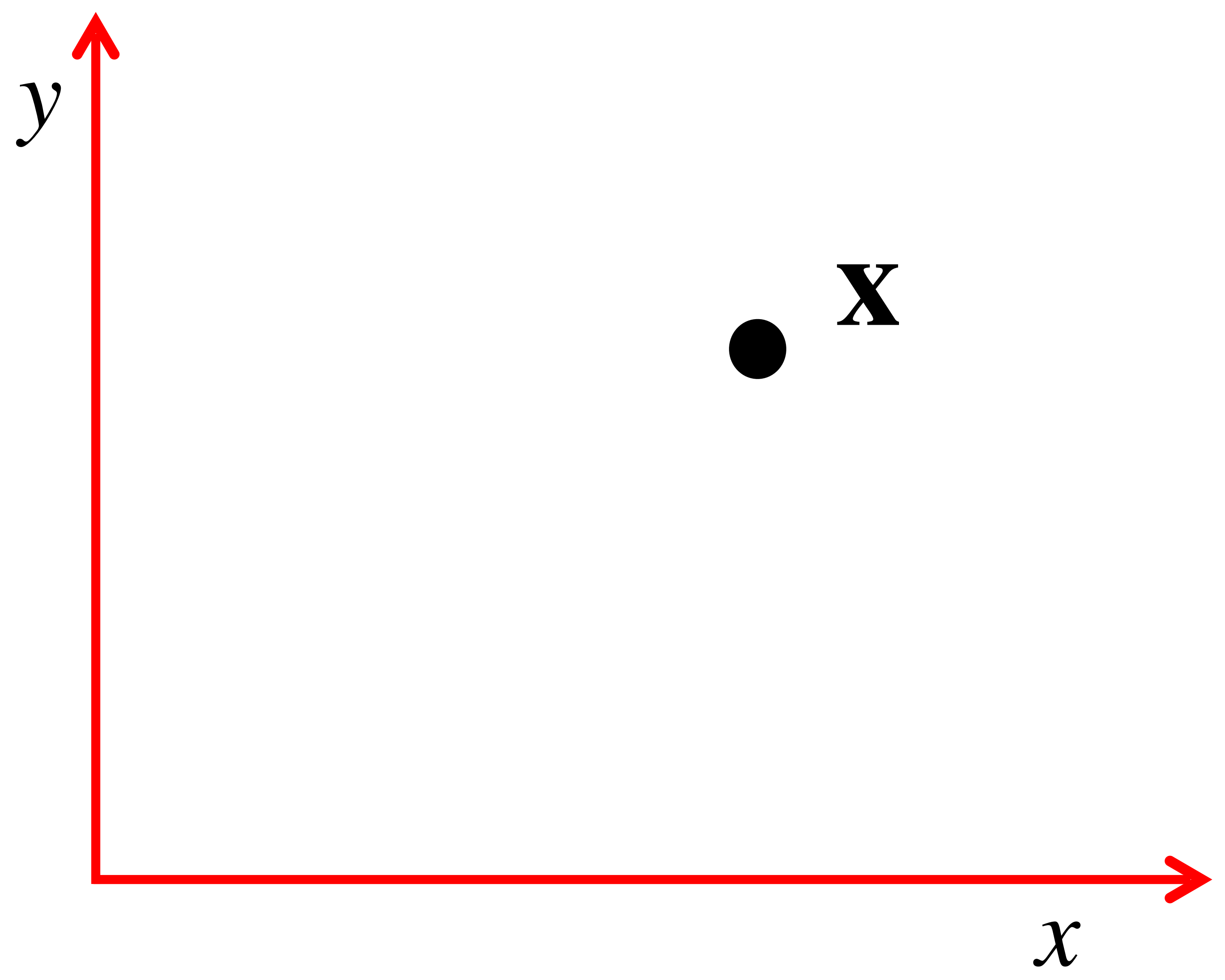


$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{f}{Z}$$

This is awkward… Not a linear operator.

Homogeneous coordinates

2D



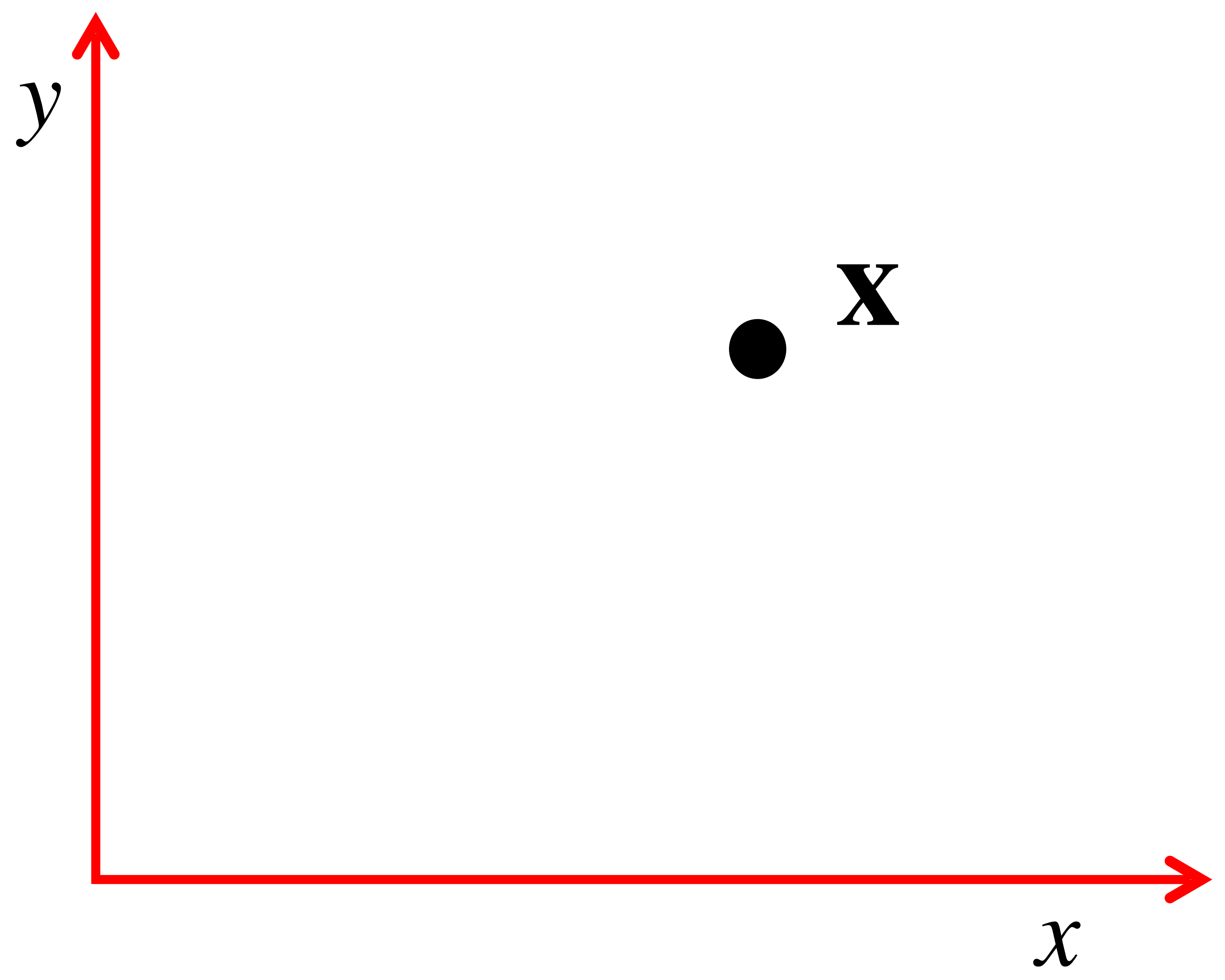
$$\mathbf{x} = (x, y) \quad \Rightarrow \quad \tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

heterogeneous coordinates

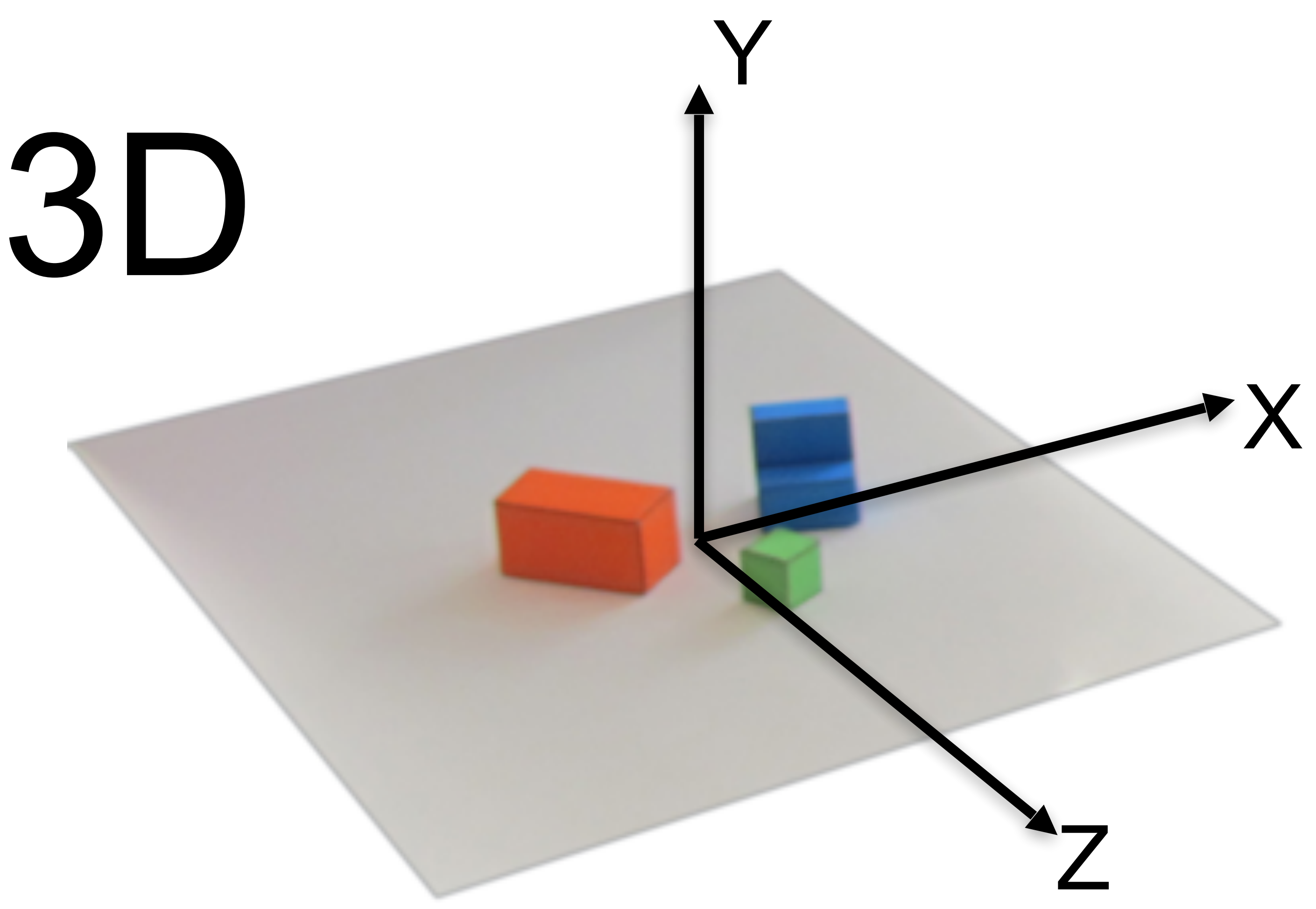
homogeneous coordinates

Homogeneous coordinates

2D



3D



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

heterogeneous coordinates

homogeneous coordinates

$$\mathbf{X} = (X, Y, Z) \Rightarrow \tilde{\mathbf{X}} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

From heterogeneous to homogeneous:

$$\mathbf{x} = (x, y) \quad \Rightarrow \quad \tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \cdot 1 \end{bmatrix}$$

From homogeneous to heterogeneous:

$$\tilde{\mathbf{x}} = \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} \quad \Rightarrow \quad \mathbf{x} = \left(x/w, y/w\right)$$

Homogeneous coordinates

2D

$\mathbb{R}^2$ $\mathbb{P}^2$

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

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

3D

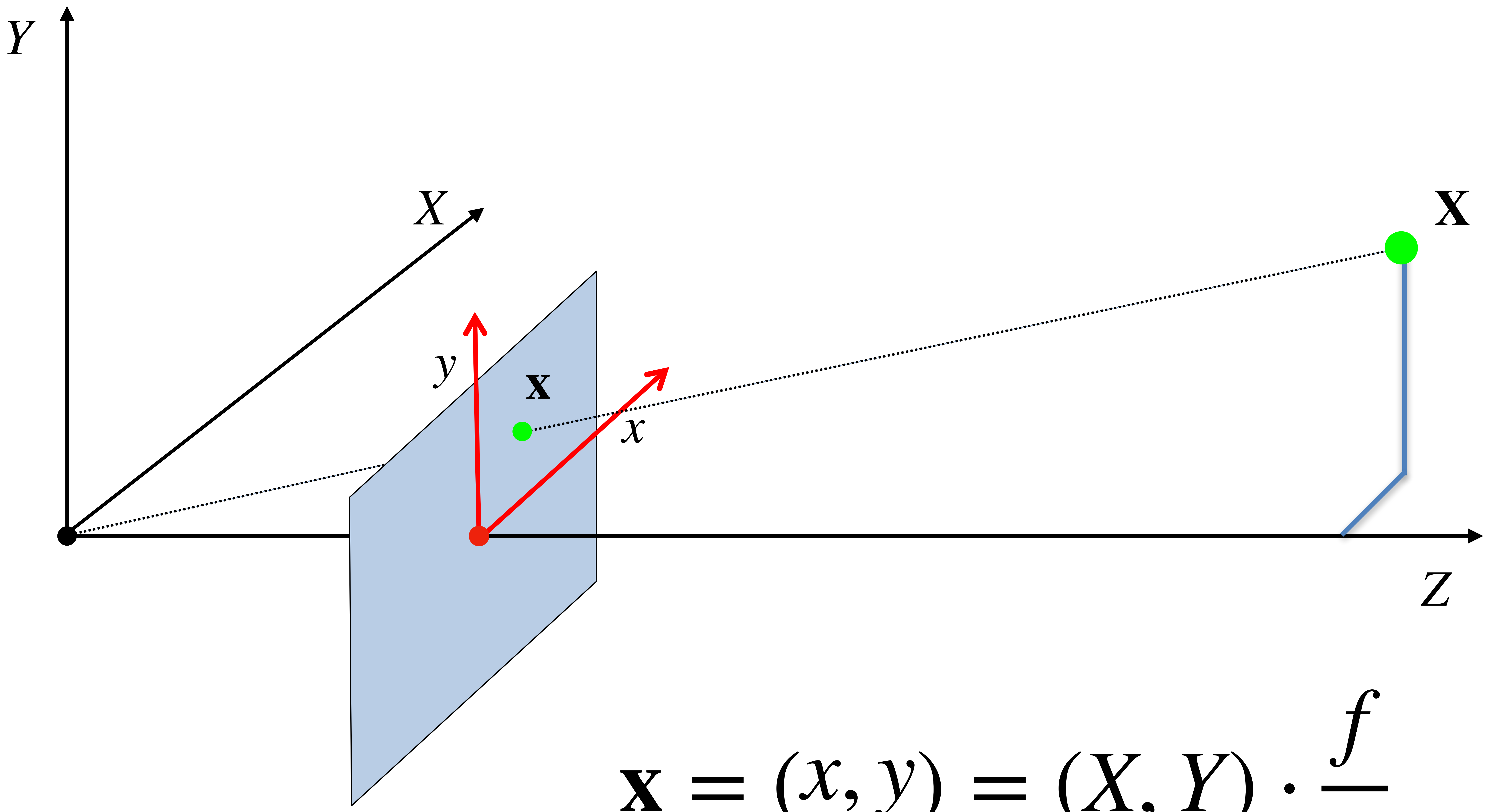
$\mathbb{R}^3$ $\mathbb{P}^3$

$$(X, Y, Z) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \Rightarrow (X/W, Y/W, Z/W)$$

Questions?

Perspective projection



$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{f}{Z}$$

This is awkward… Not a linear operator.

Perspective projection

Heterogeneous coordinates

$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{1}{Z}$$

Image / Pixel Coordinates

World Coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = ? \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Image / Pixel Coordinates

World Coordinates

Perspective projection

Heterogeneous coordinates

$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{1}{Z}$$

Image / Pixel Coordinates

World Coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Image / Pixel Coordinates

World coords.

Perspective projection

Heterogeneous coordinates

$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{1}{Z}$$

Image / Pixel Coordinates

World Coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Image / Pixel Coordinates

Projection Matrix

World coords.

Perspective projection

Heterogeneous coordinates

$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{1}{Z}$$

Image / Pixel Coordinates

World Coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

Image / Pixel Coordinates

Projection Matrix

World coords.

Perspective projection

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

Image / Pixel Coordinates

Projection Matrix

World coords.

Perspective projection

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

Image / Pixel Coordinates

Projection Matrix

World coords.

$$\begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix} \Rightarrow \left(f\frac{X}{Z},\ \ f\frac{Y}{Z} \right)$$

Perspective projection

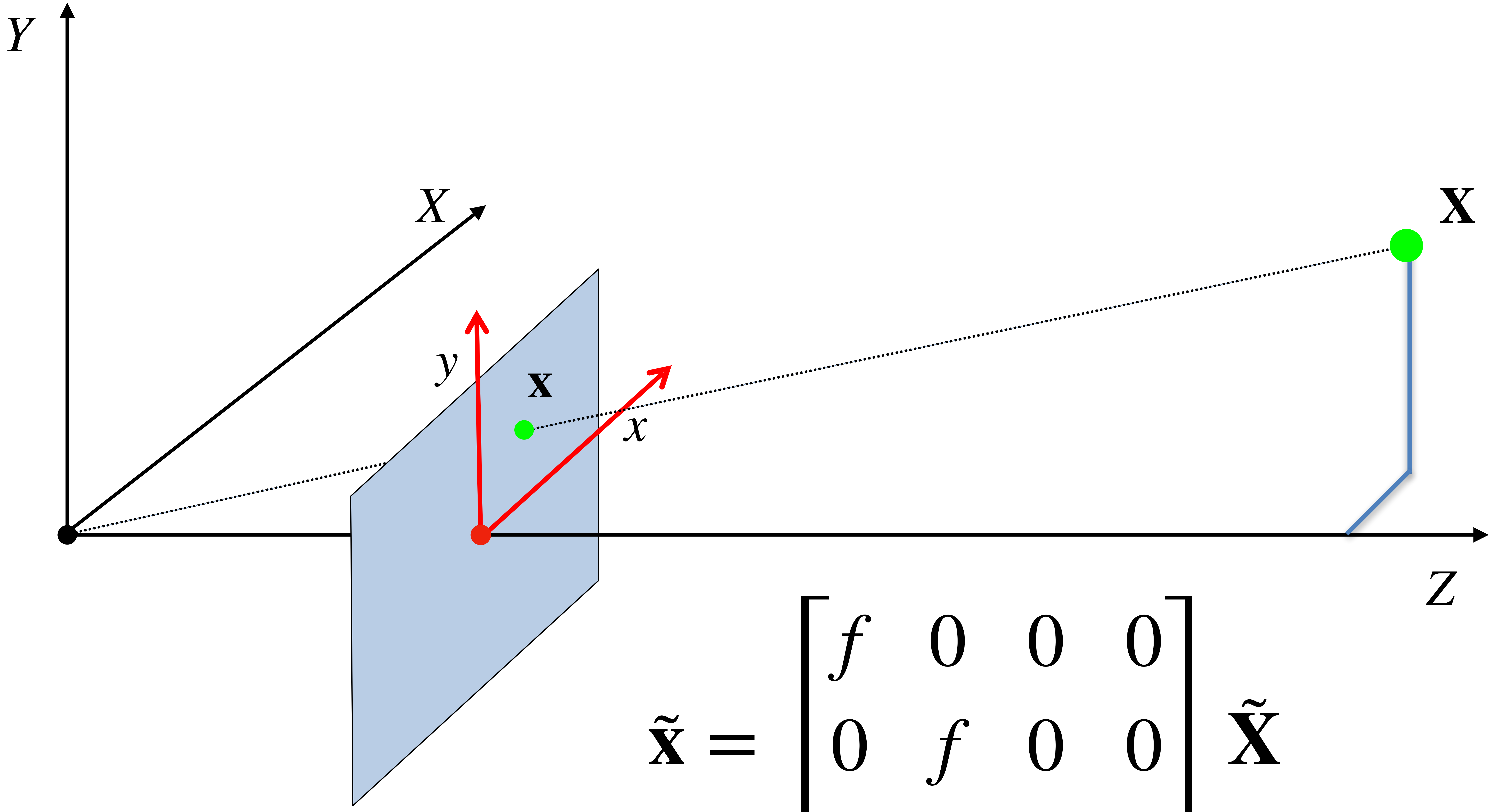
$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix} \Rightarrow \left(f\frac{X}{Z}, \quad f\frac{Y}{Z} \right)$$

$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} f \cdot X \\ f \cdot Y \\ Z \end{bmatrix} \Rightarrow \left(f\frac{X}{Z}, \quad f\frac{Y}{Z} \right)$$

Perspective projection

$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}$$

Perspective projection



$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}$$

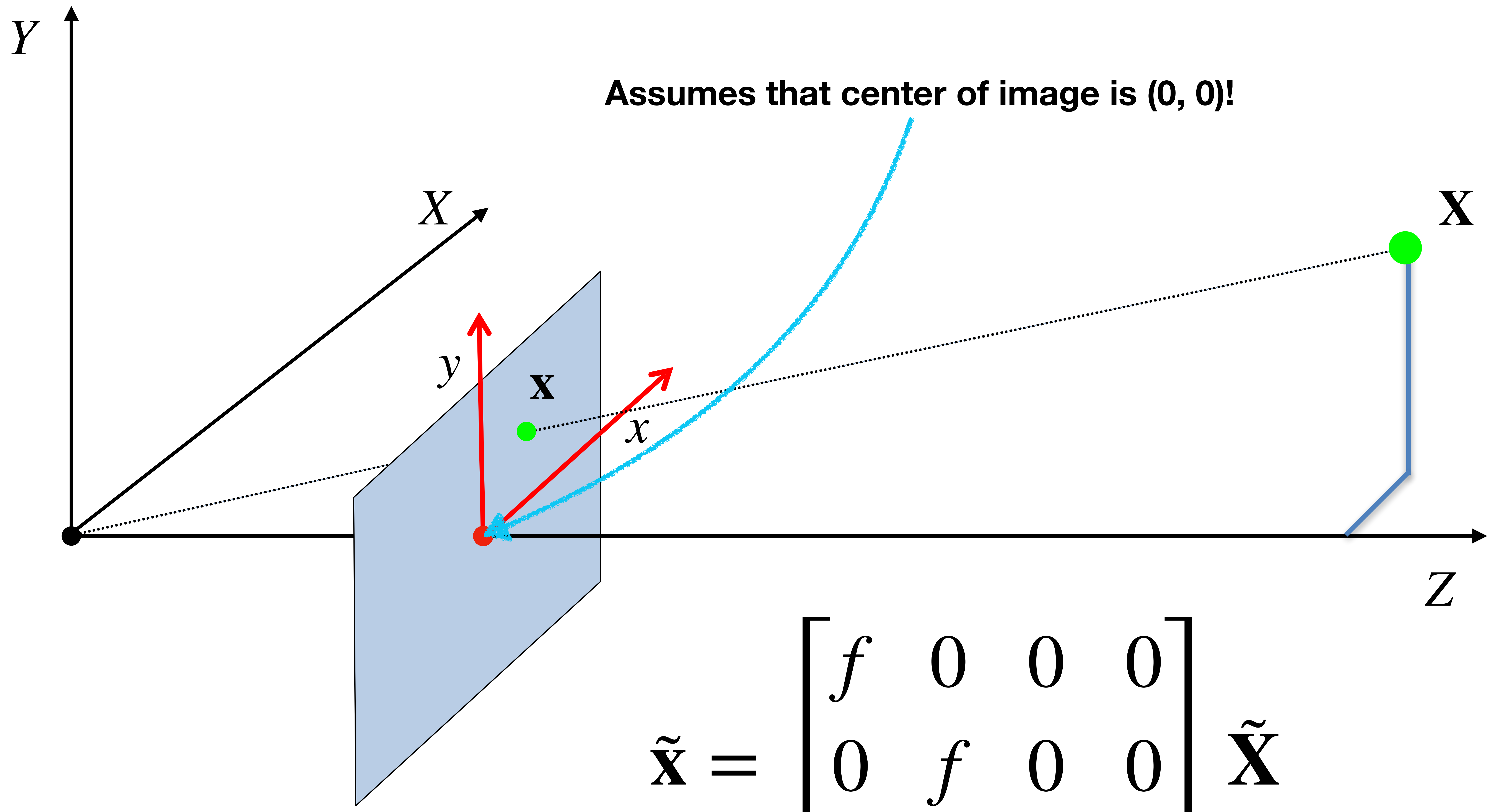
Image / Pixel coords.

World coords.

Projection Matrix

Questions?

Perspective projection



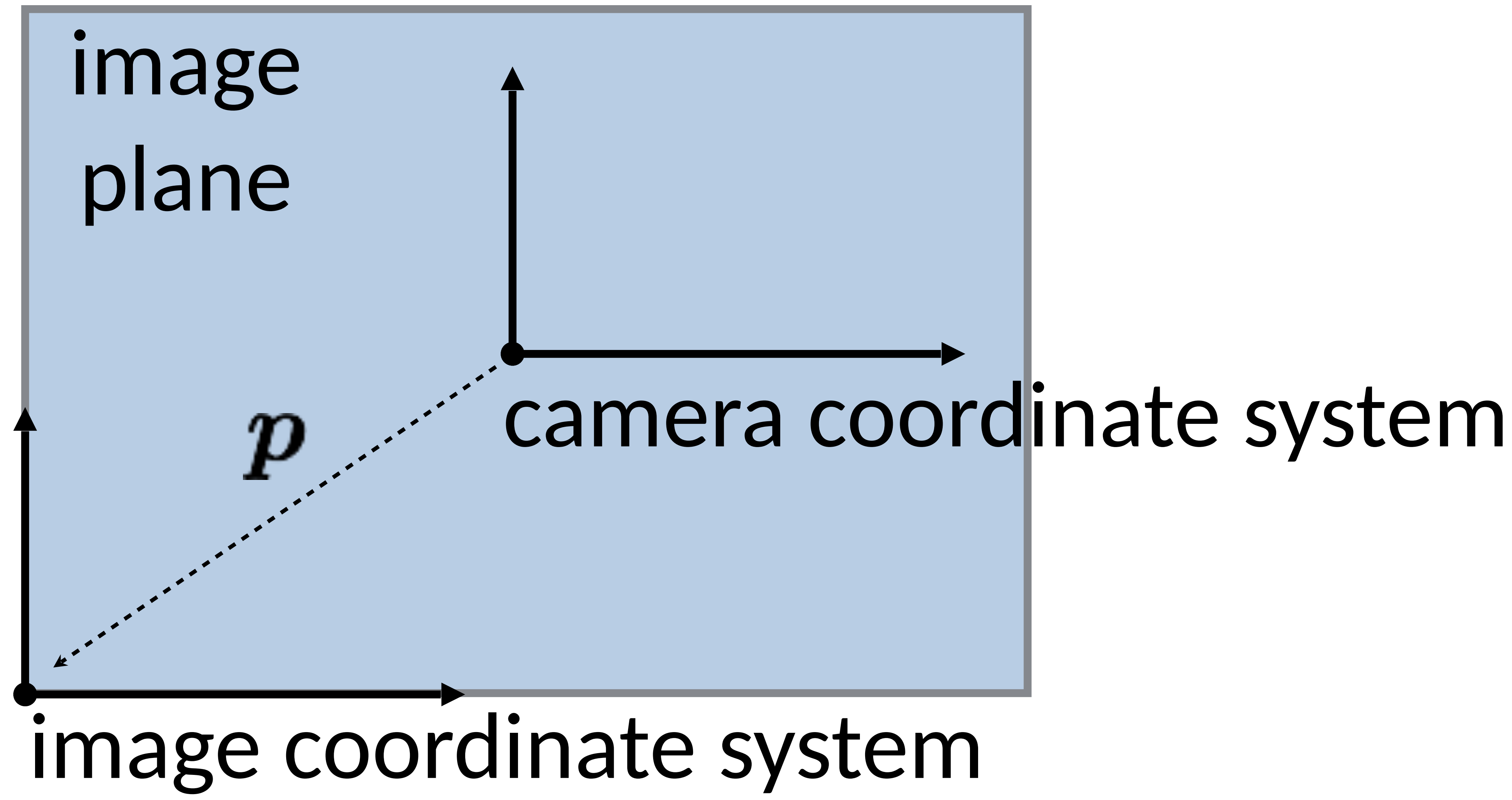
$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}$$

Image / Pixel coords.

Projection Matrix

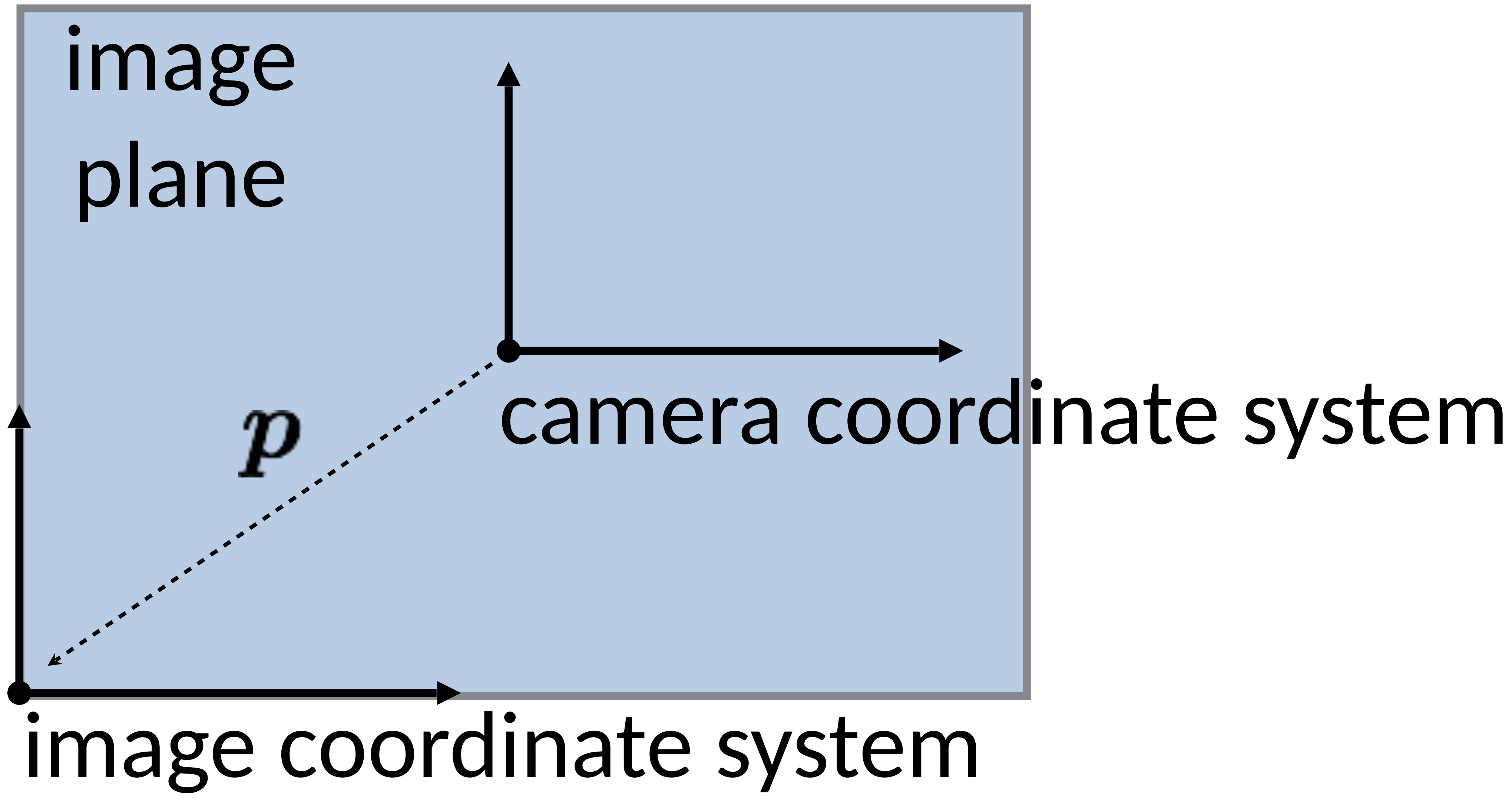
World coords.

More General Case: Arbitrary Image Centre



Slide credit: CMU 16-385 (Yannis, Kris)

More General Case: Arbitrary Image Centre



How does the projection matrix change?

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

shift vector transforming camera origin to image origin

Slide credit: CMU 16-385 (Yannis, Kris)

Decomposing the Projection Matrix

We can decompose the projection matrix like this:

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

What does each part of the matrix represent?

Slide credit: CMU 16-385 (Yannis, Kris)

Decomposing the Projection Matrix

We can decompose the projection matrix like this:

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

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

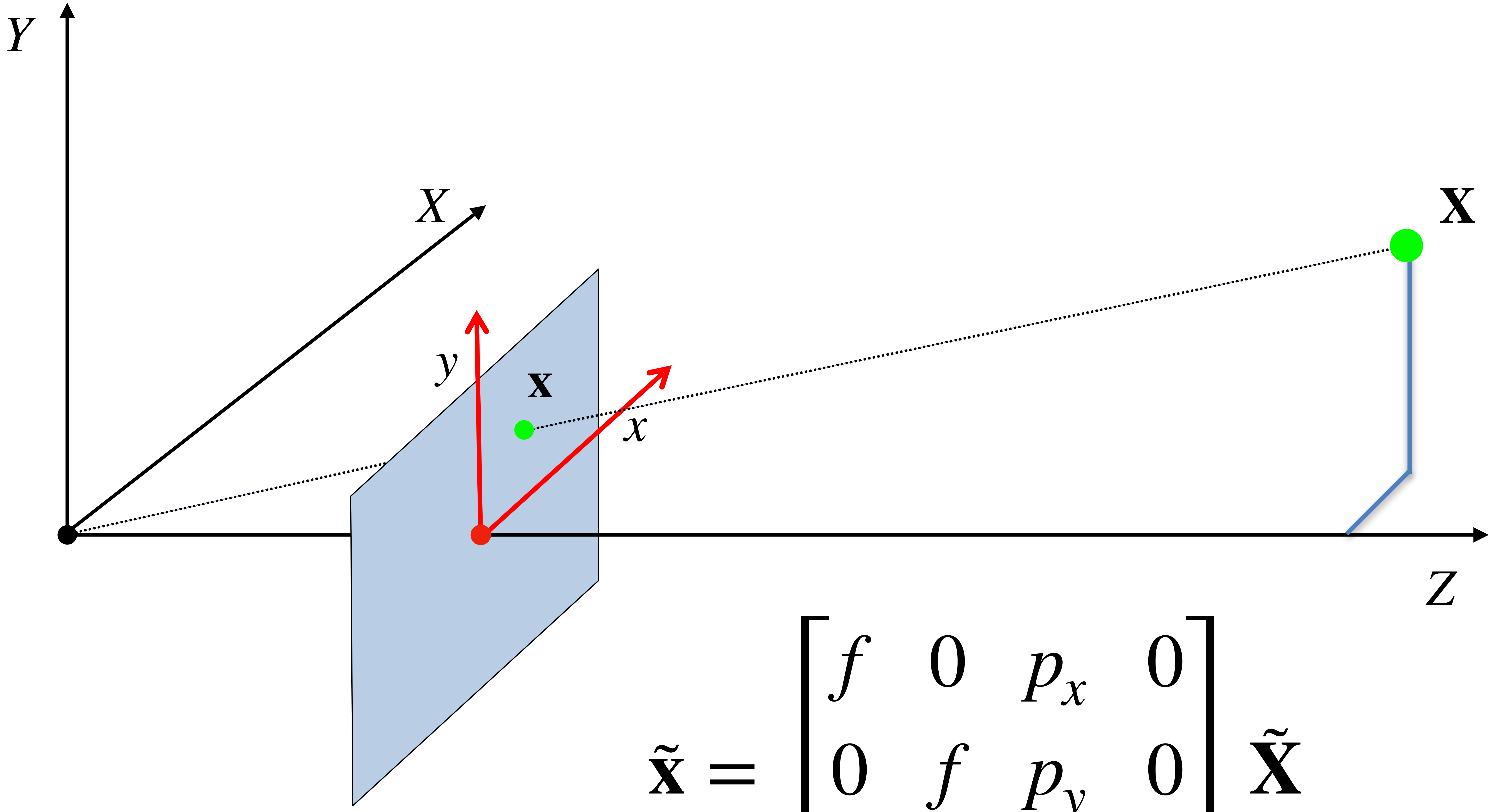
(homogeneous) perspective projection from 3D to 2D, assuming image plane at z = 1 and shared camera/image origin

Also written as: $\mathbf{K}\left[\mathbf{I} \mid \mathbf{0}\right]$

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

Slide credit: CMU 16-385 (Yannis, Kris)

Perspective projection



$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}$$

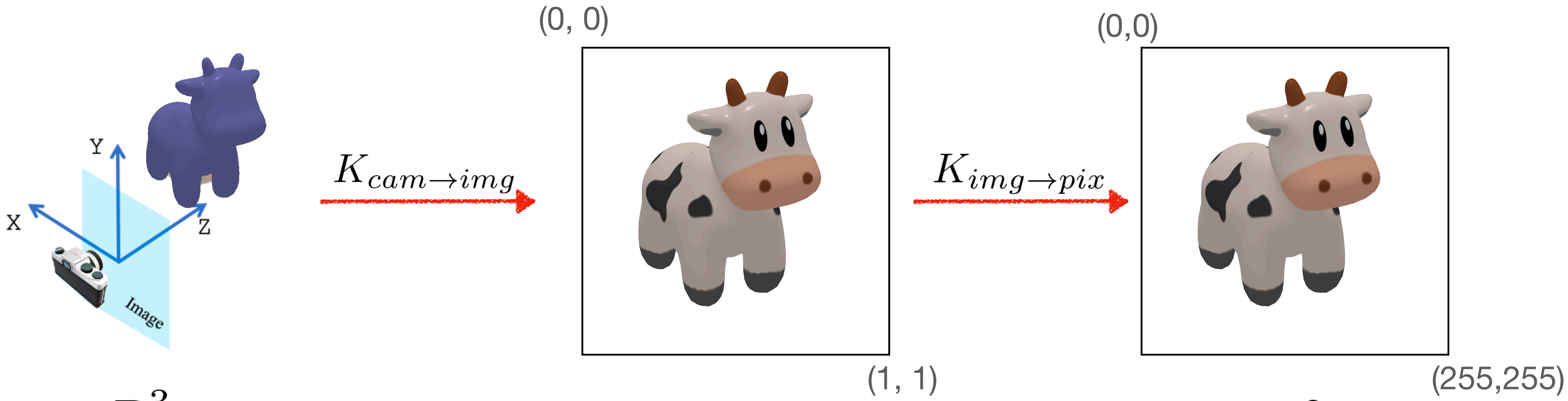
Image / Pixel coords.

Projection Matrix

World coords.

In practice: Decoupling Projection from Image Size

$$K \equiv K_{img \to pix} \; K_{cam \to img}$$



$\mathbb{P}^3$

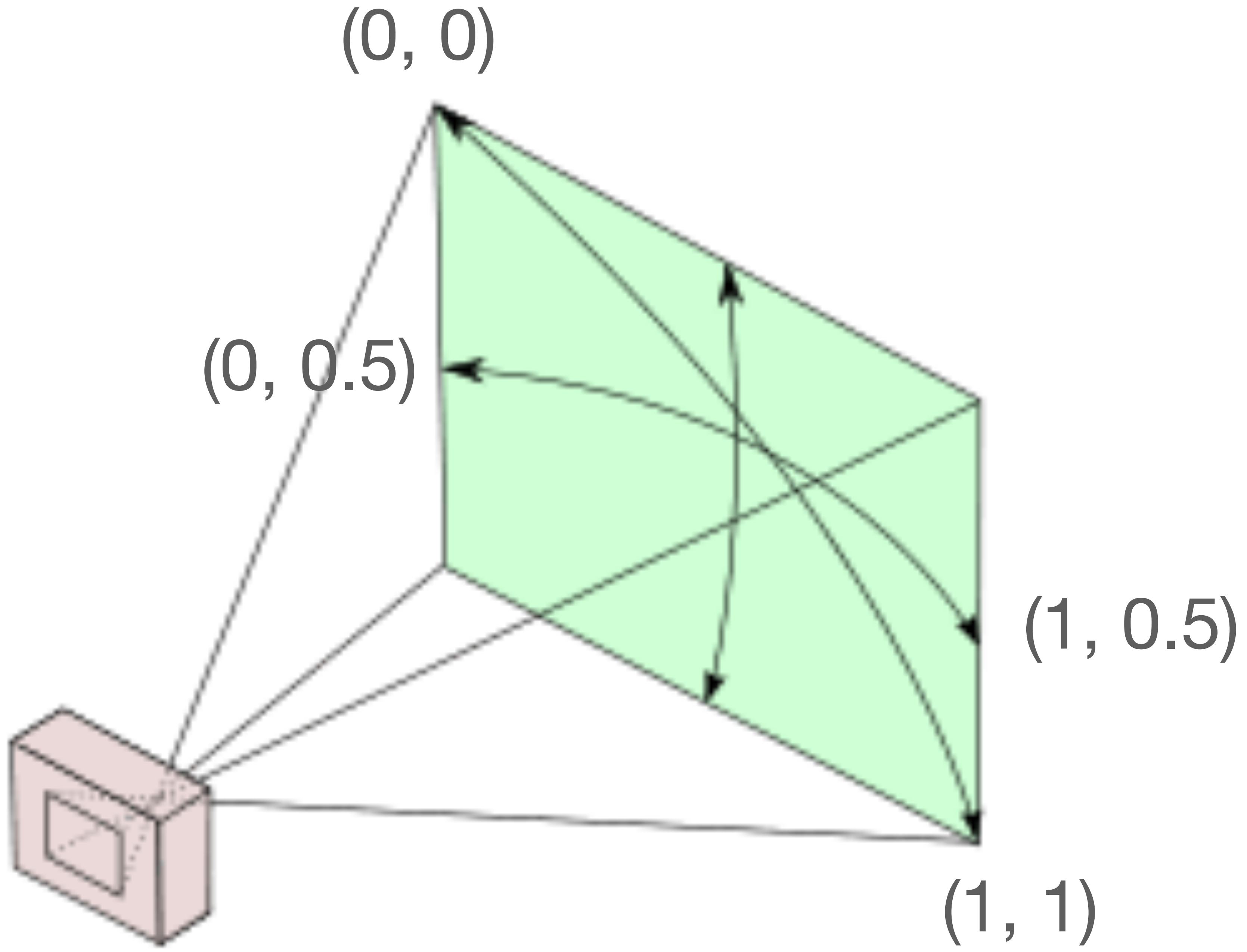
3D in Camera Frame

Normalized Image Frame (here OpenCV convention)

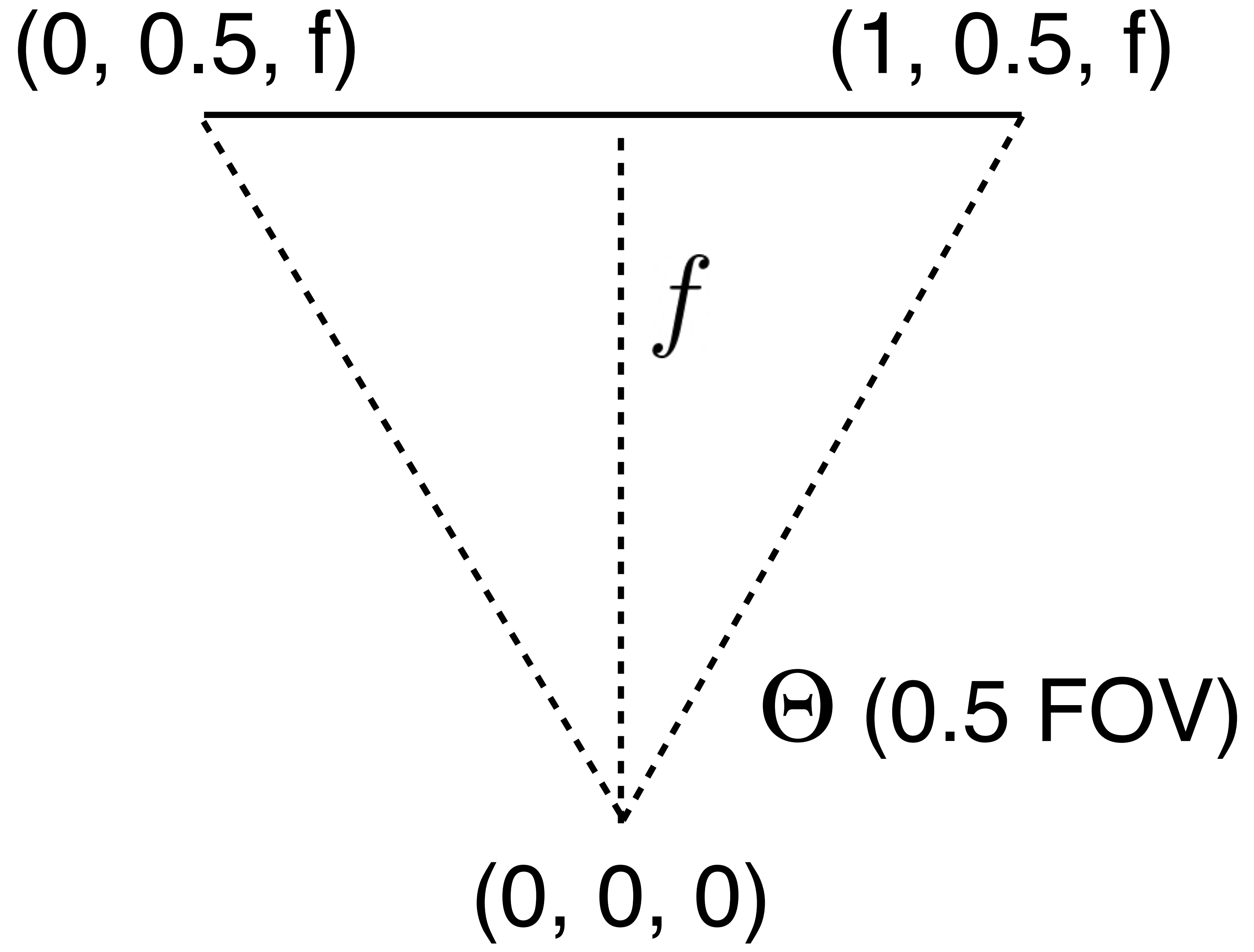
$\mathbb{P}^2$

Exercise: (Horizontal) Field of view = 60 degrees

$$K_{cam \rightarrow img}$$



$$\begin{bmatrix} \frac{1}{2\tan(\Theta)} & 0 & 0.5 & 0 \\ 0 & \frac{1}{2\tan(\Theta)} & 0.5 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Slide credit: CMU 16-889: Learning for 3D Vision, Prof. Shubham Tulsiani

Exercise: Cropping an Image

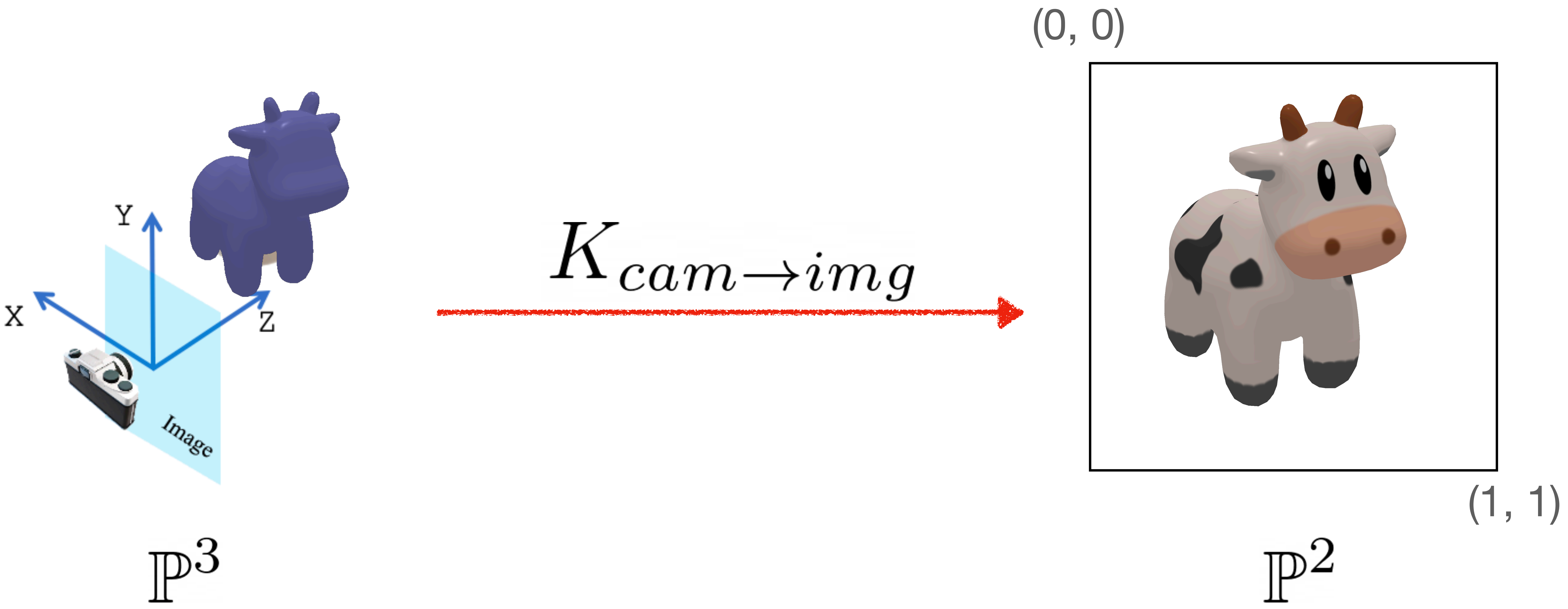
$$K_{cam\rightarrow img}$$

$$K'_{cam\rightarrow img}$$

$$K'_{cam\rightarrow img} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} K_{cam\rightarrow img}$$

Slide credit: CMU 16-889: Learning for 3D Vision, Prof. Shubham Tulsiani

In practice: Decoupling Projection from Image Size

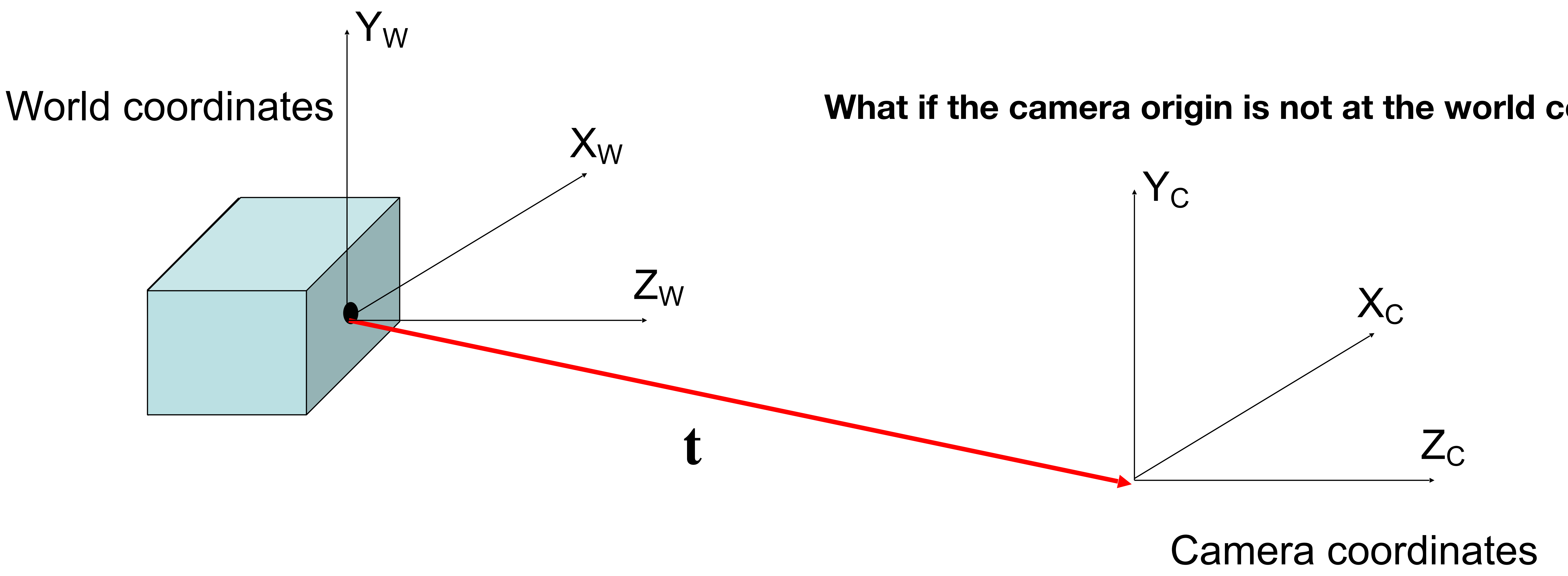


Helps (conceptually and in implementation) to reason in normalized image coordinates

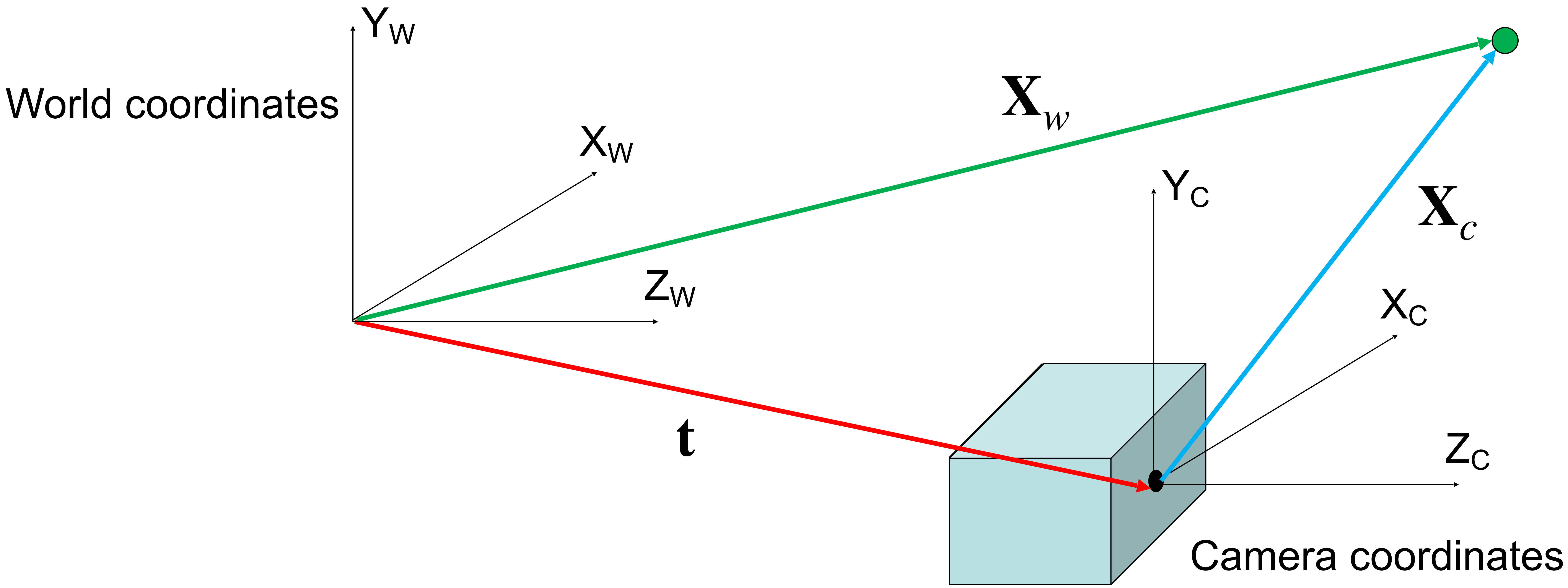
Questions?

Camera parameters

What if the camera origin is not at the world coordinates origin?



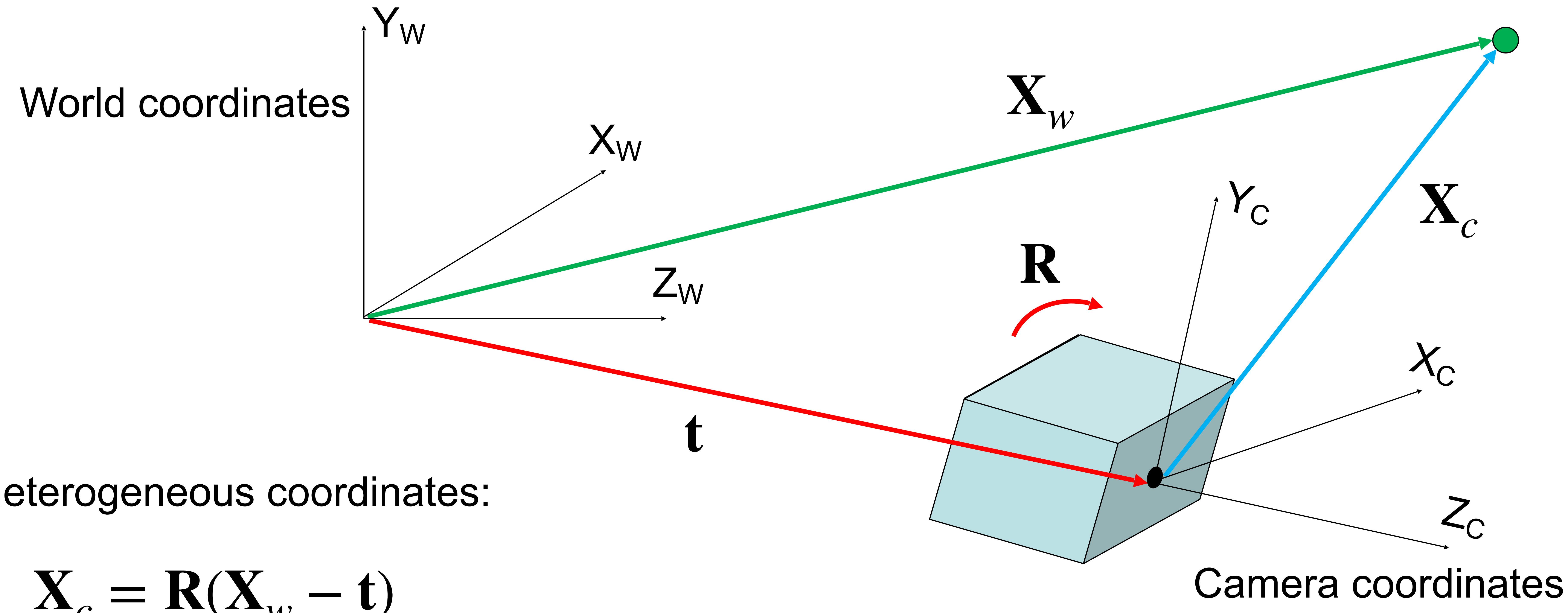
Camera parameters



In heterogeneous coordinates:

$$\mathbf{X}_c = \mathbf{X}_w - \mathbf{t}$$

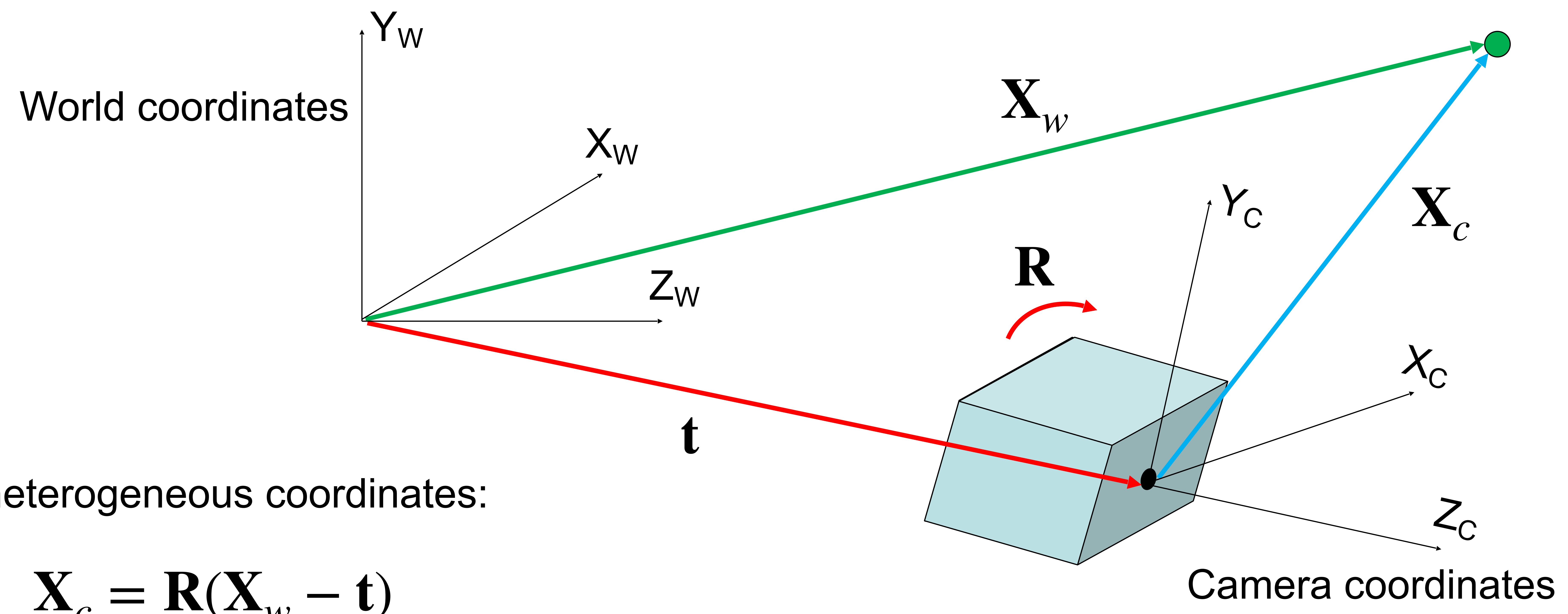
Camera parameters



In heterogeneous coordinates:

$$\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{t})$$

Camera parameters



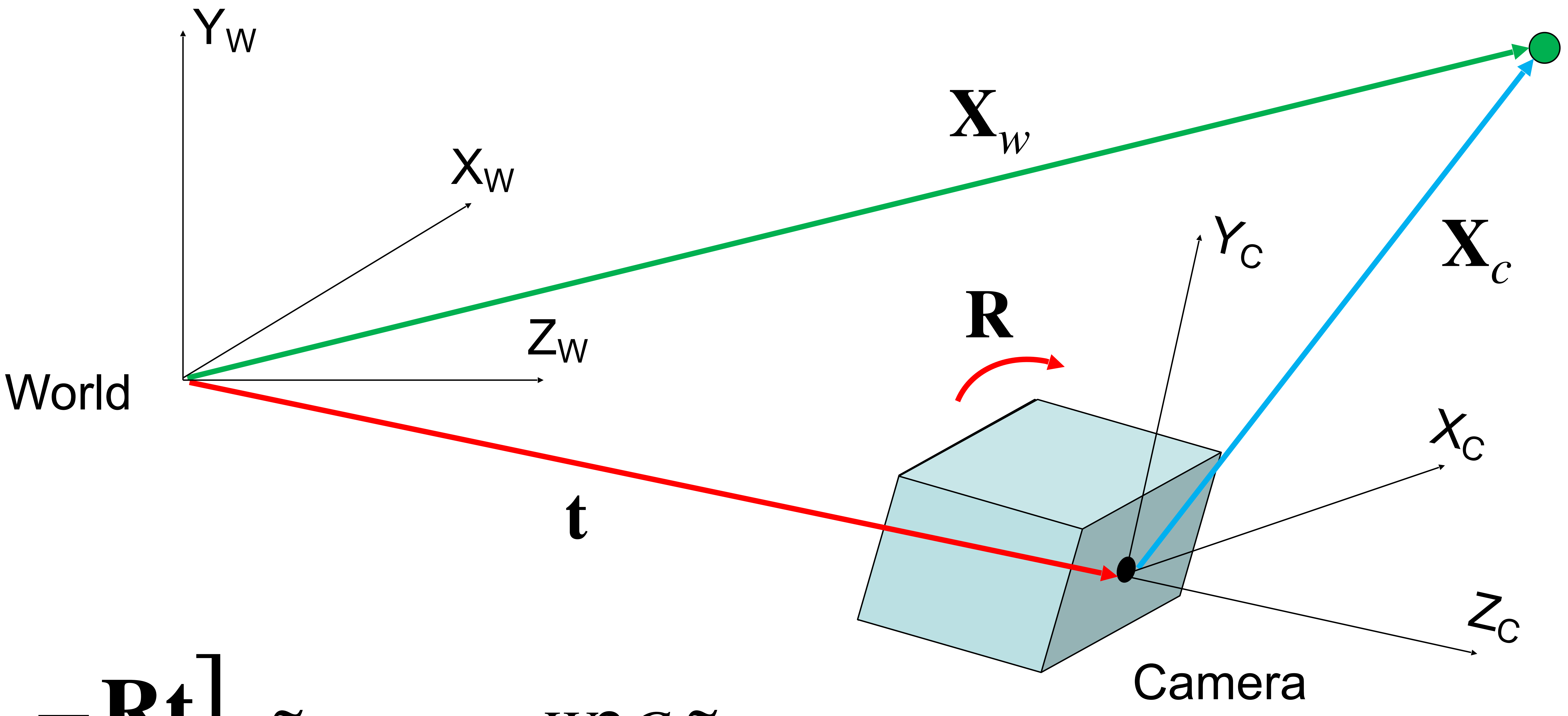
In heterogeneous coordinates:

$$\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{t})$$

In homogeneous coordinates:

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

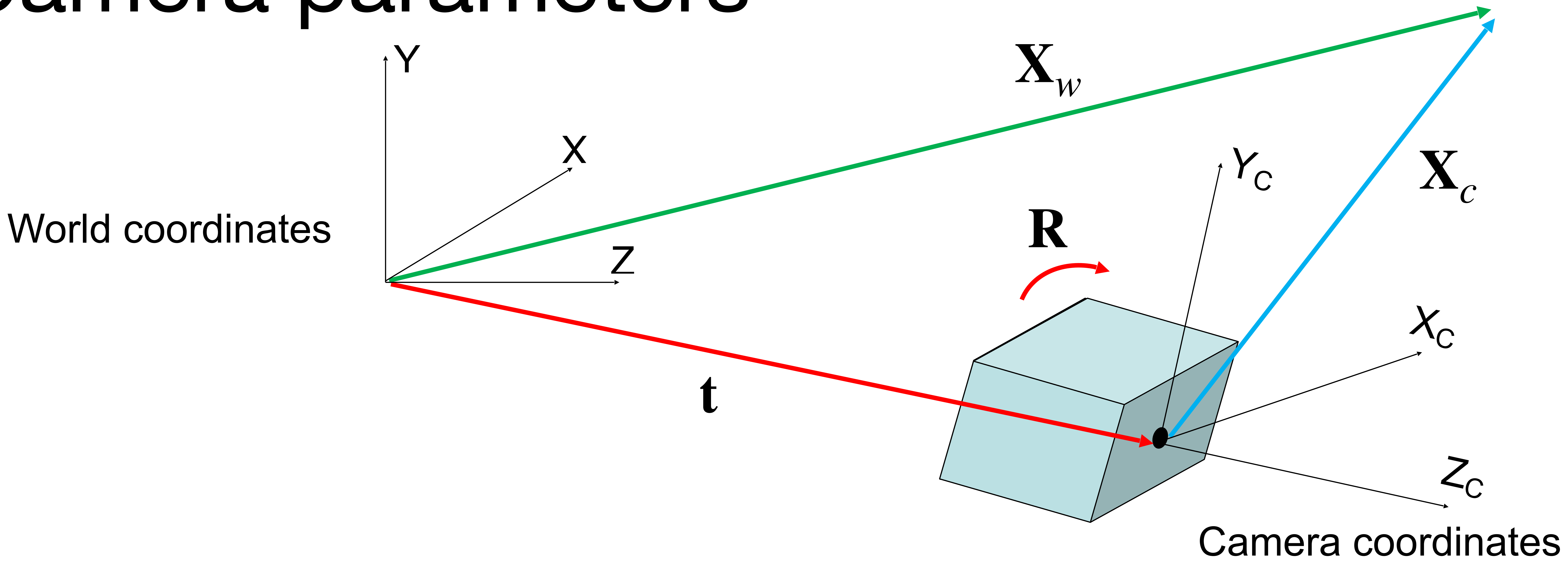
Cam2World vs. World2Cam extrinsic parameters



$$\tilde{\mathbf{X}}_c = \begin{bmatrix} \mathbf{R} & -\mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w = \mathbf{C}^{W2C} \tilde{\mathbf{X}}_w$$

$$\tilde{\mathbf{X}}_w = \begin{bmatrix} \mathbf{R}^T & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_c = (\mathbf{C}^{W2C})^{-1} \tilde{\mathbf{X}}_c = \mathbf{C}^{C2W} \tilde{\mathbf{X}}_c$$

Camera parameters



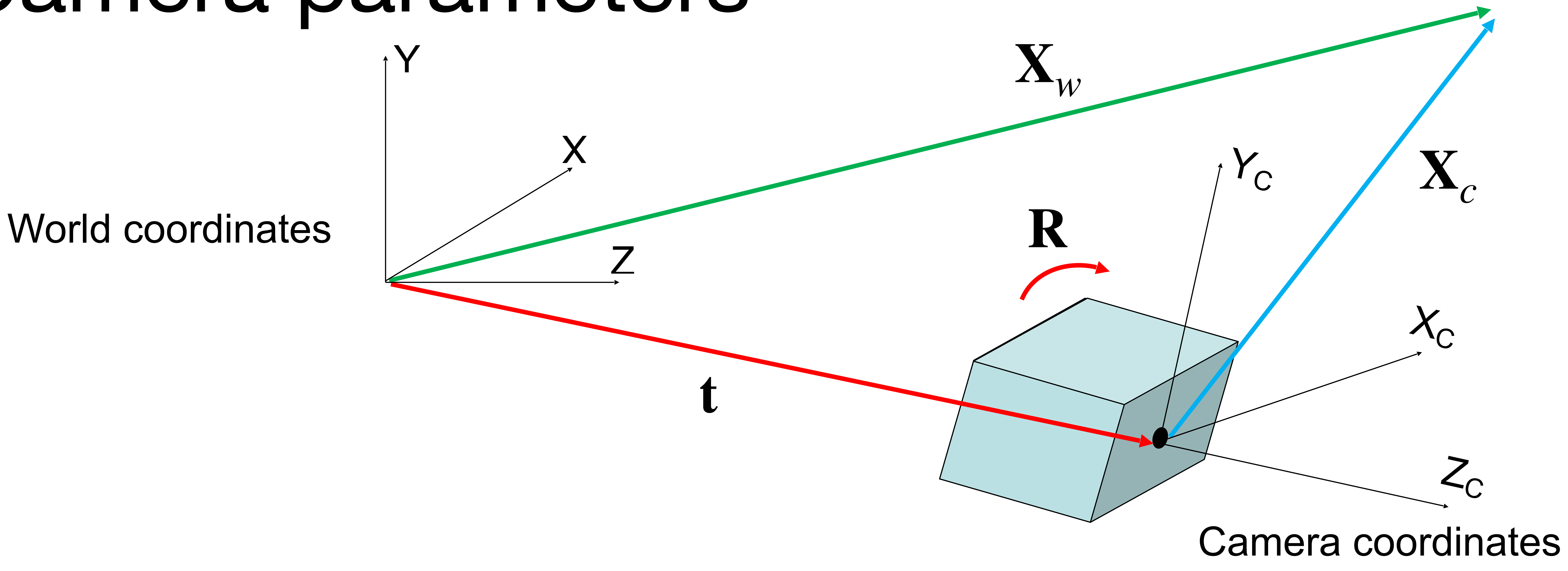
World coordinates to camera coordinates

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

Camera coordinates to image coordinates

$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}_c$$

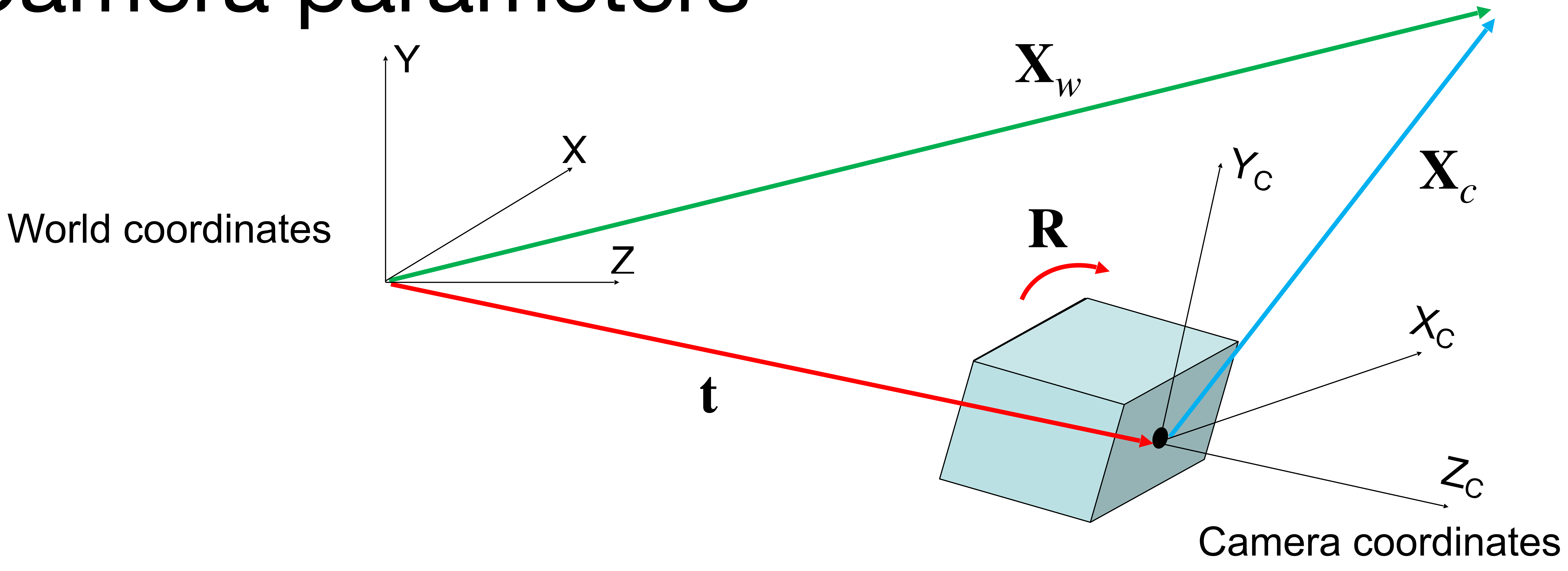
Camera parameters



World coordinates to image coordinates

$$\tilde{\mathbf{x}} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \tilde{\mathbf{X}}_w$$

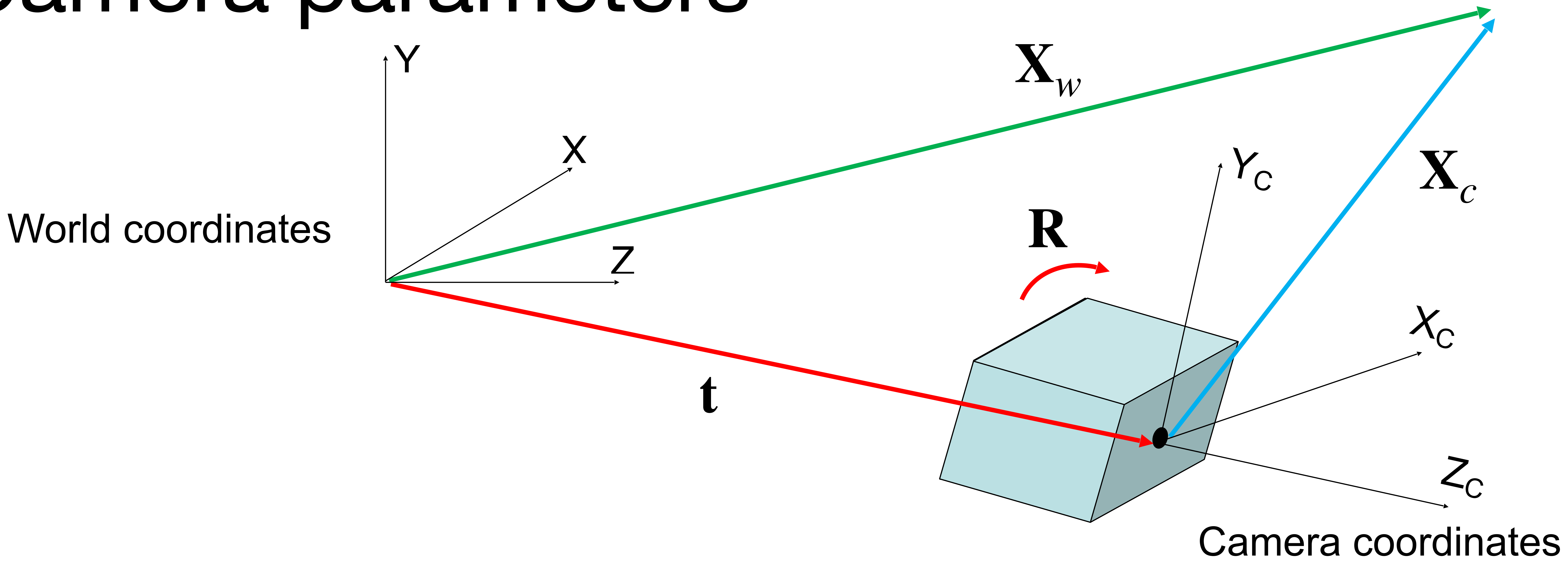
Camera parameters



World coordinates to image coordinates

$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w$$

Camera parameters



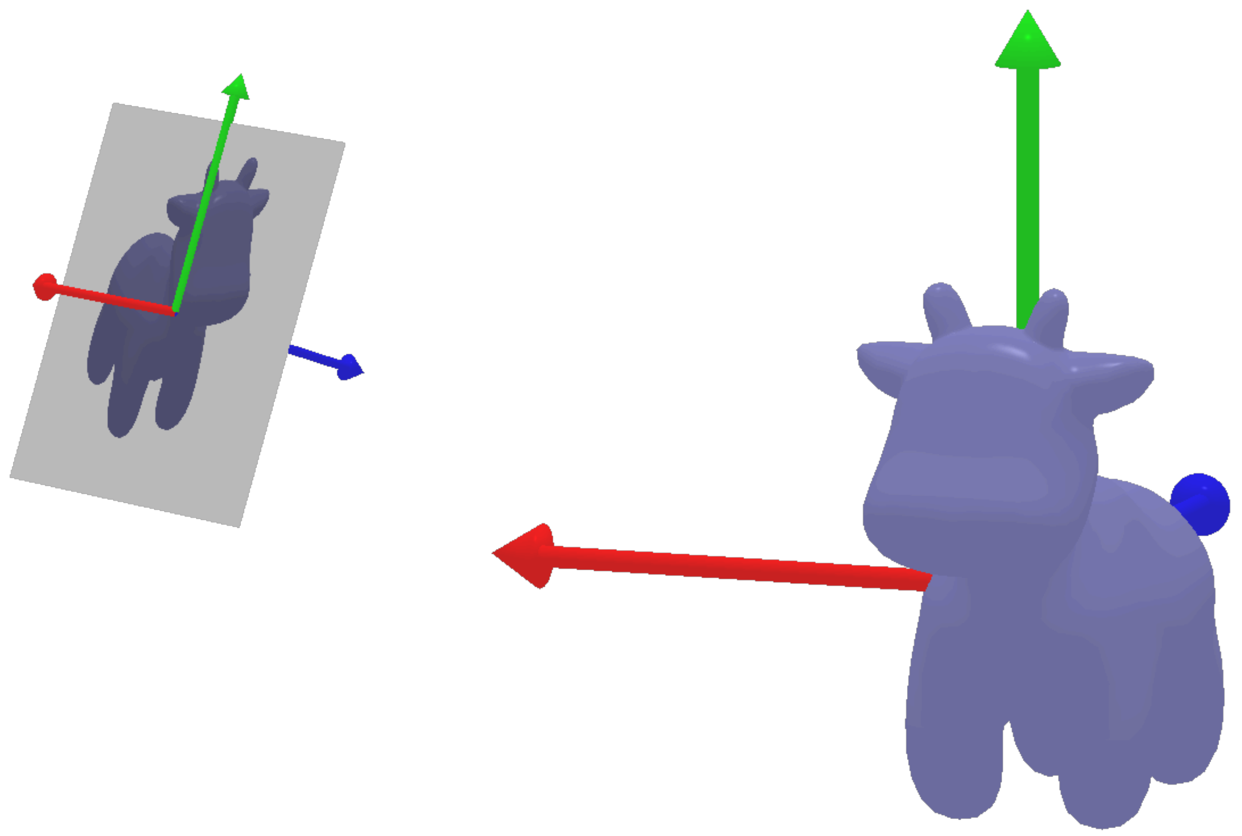
World coordinates to image coordinates

$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w = \mathbf{P}\tilde{\mathbf{X}}_w$$

Intrinsic parameters **Extrinsic** parameters

World2Camera Transformations: Exercise

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



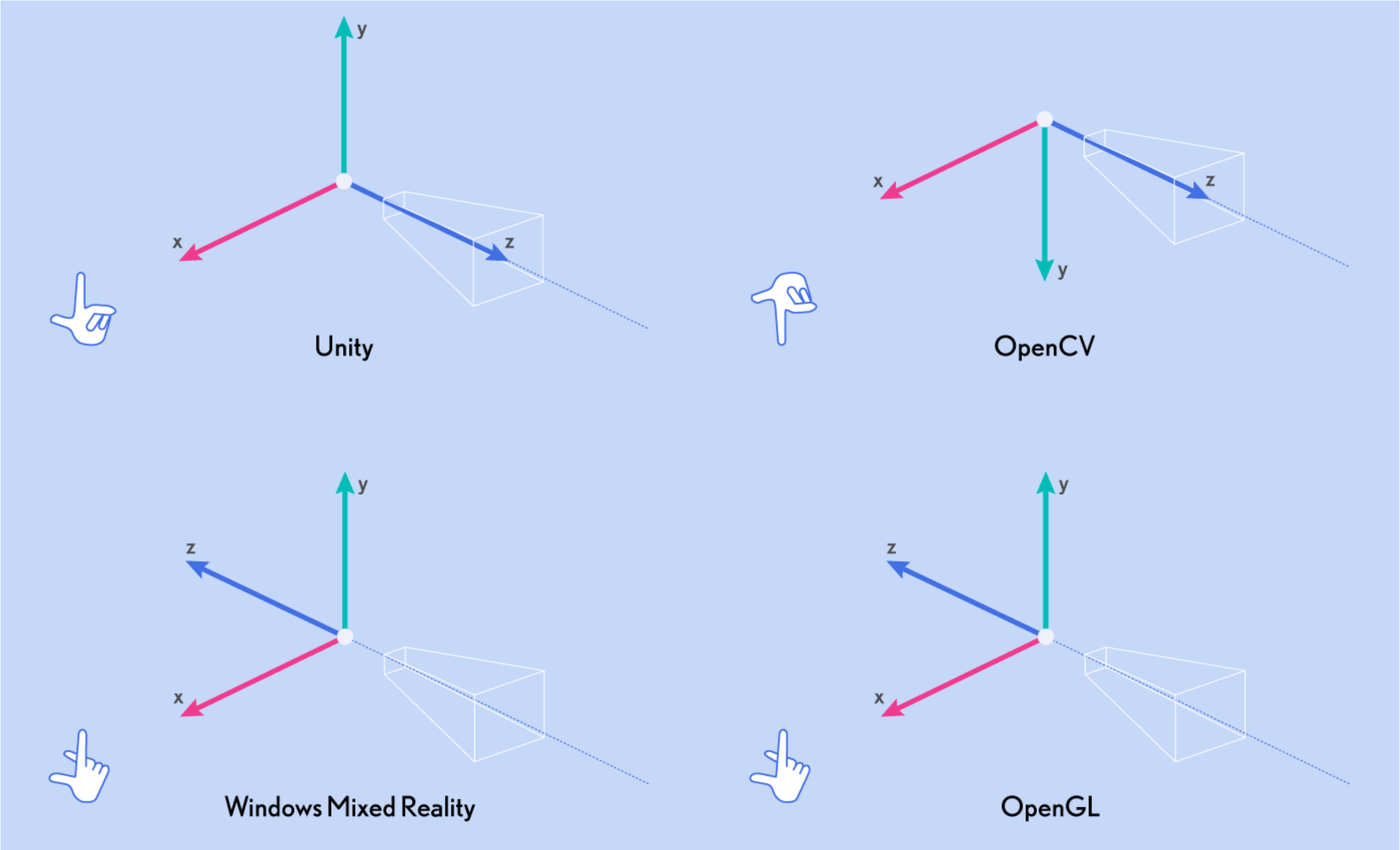
What are R, t for an upright (world and camera y-directions align) origin-facing camera 2m away from origin located at (0,0,-2)?

$$R = \mathbf{I}; \quad \mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

What are R, t for an upright (world and camera y-directions align) origin-facing camera 2m away from origin located at (-2,0,0)?

$$R = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}; \quad \mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Beware of Conventions



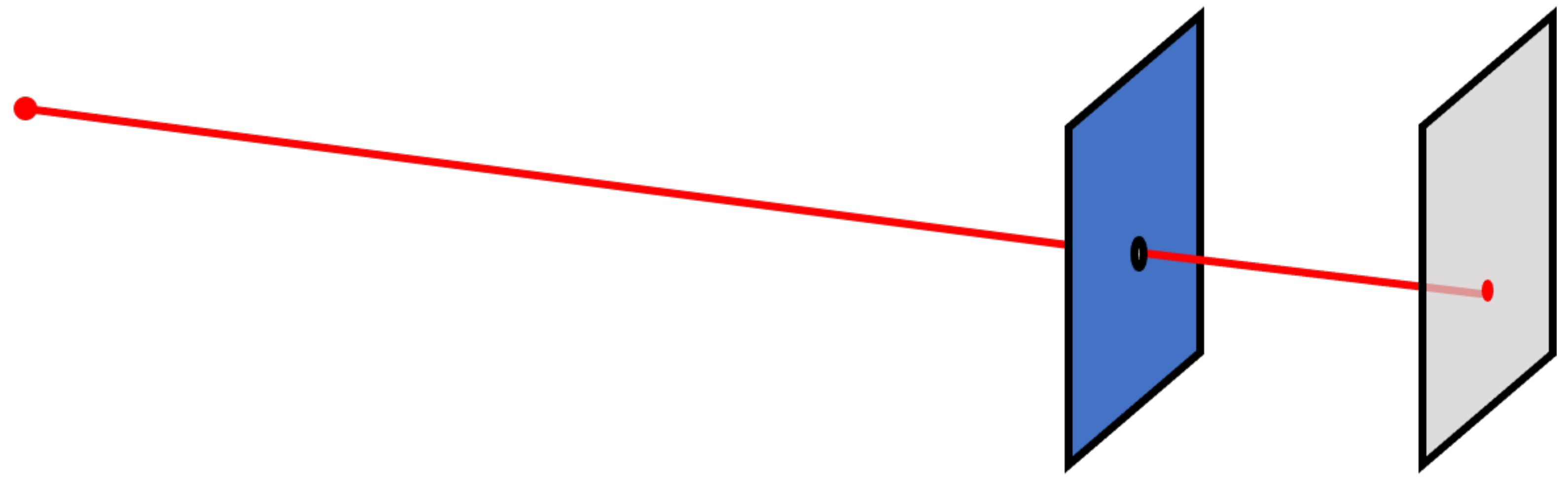
X=left or right? Y=up or down? etc..

If you externally obtain transformation matrices (e.g. using someone else's code), make sure of convention compatibility (one of the *most common* sources of bugs!)

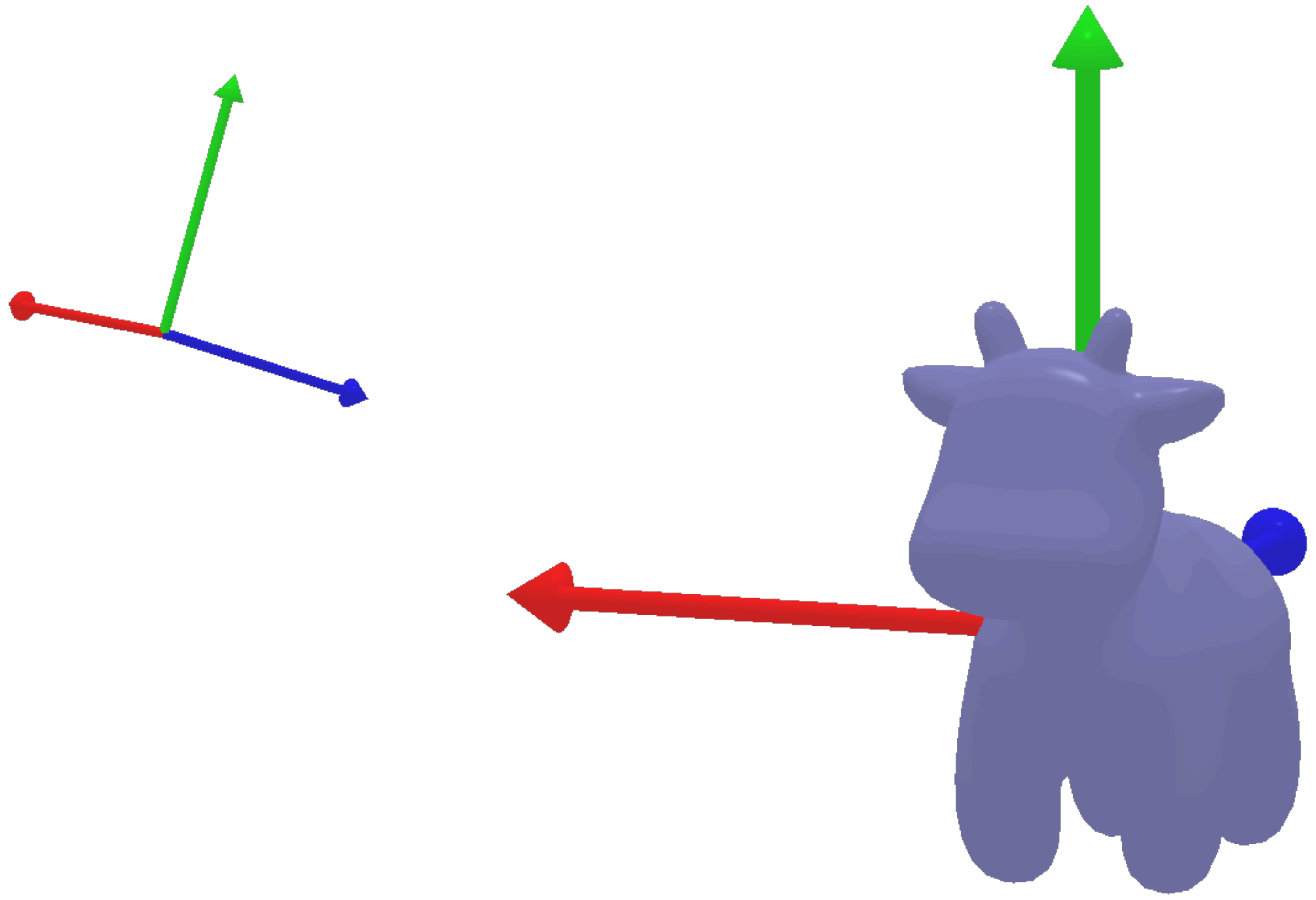
read https://pytorch3d.org/docs/cameras

Slide credit: Shubham Tulsiani. Image courtesy: Christoph Krautz

Summary



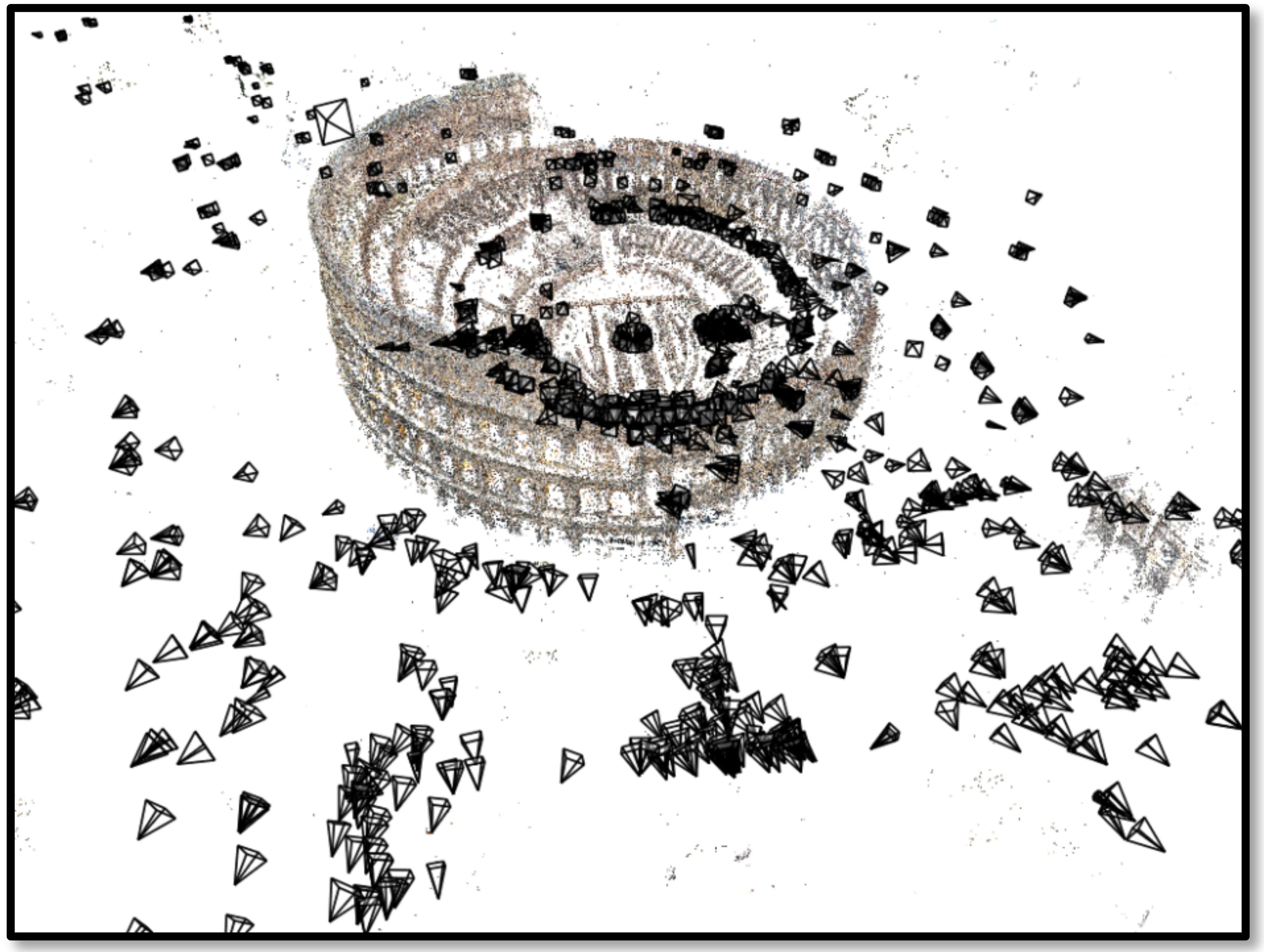
Projection — associating rays to points in a plane



Camera Transformation — from world to camera coordinates

Questions?

Module 1: Image Formation and Multi-View Geometry



Why?	**We want to understand 3D world only from 2D observations (images). For that, we need to have a mathematical understanding of how they are connected.**
What you'll learn.	**Mathematical model of cameras. Reconstruct camera poses, approximate geometry, and camera parameters from 2D images of a scene.**

Image credits: *Noah Snavely*