Lecture 15

Image Formation & Projective Geometry

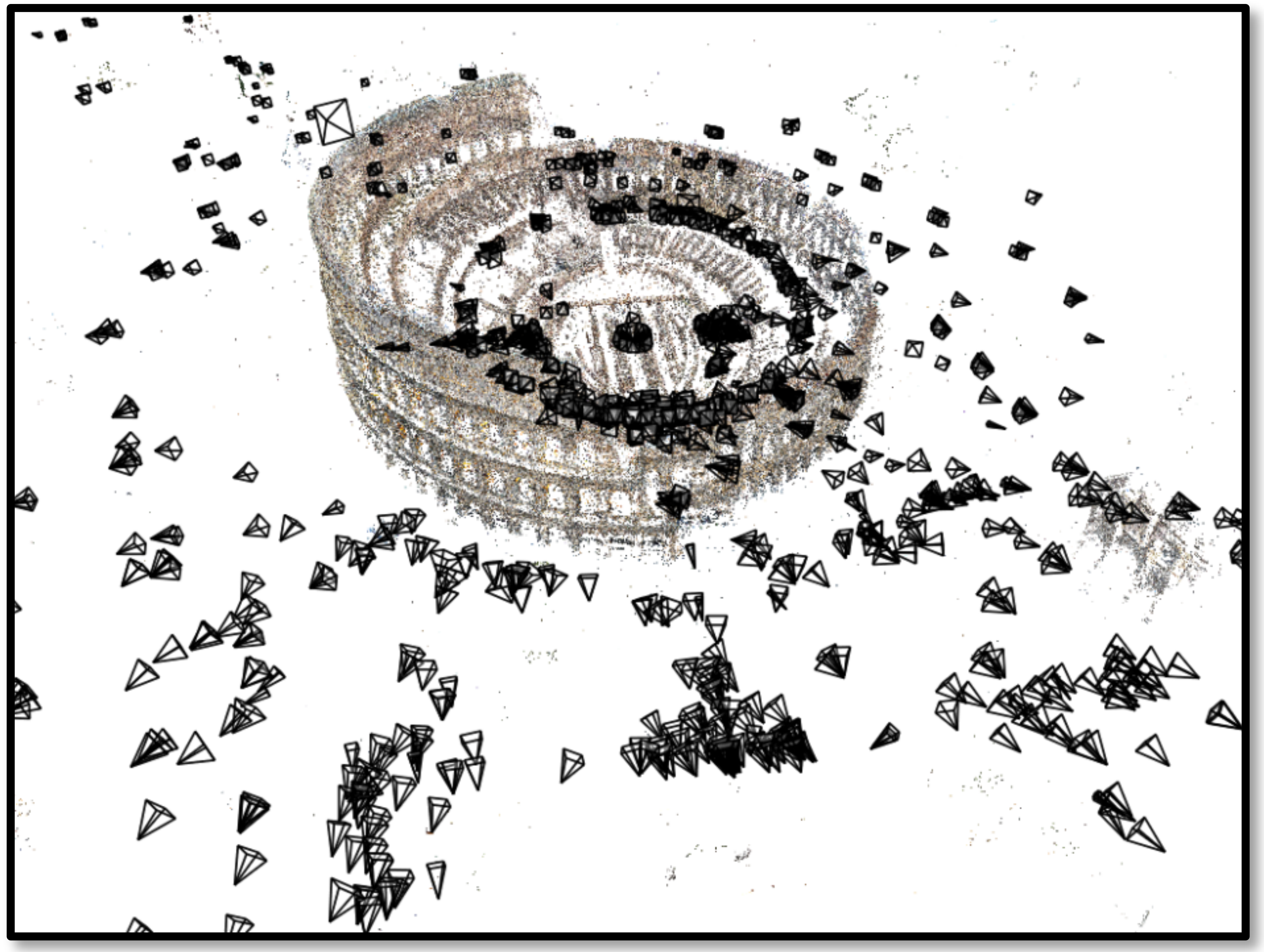
Spring 2023

Vincent Sitzmann, Bill Freeman, Mina Luković

Course Project Notes

- **If you don’t have access to external compute resources, don’t propose a compute-intensive project.**
- **For (very) small GPU workloads, you can use Google Colab.**
- **You can buy a bit more compute with a Google Colab Pro subscription (1 month = USD 10) - *still not enough* for compute-intensive projects, runs out very quickly**
  - We also offer a reimbursement of these USD 10 for those of you for whom this is a **prohibitive** expense - will post instructions on Piazza
- **We also offer USD 50 in Google Cloud credits for *everyone* - instructions on how to obtain these will be posted on Piazza**

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*

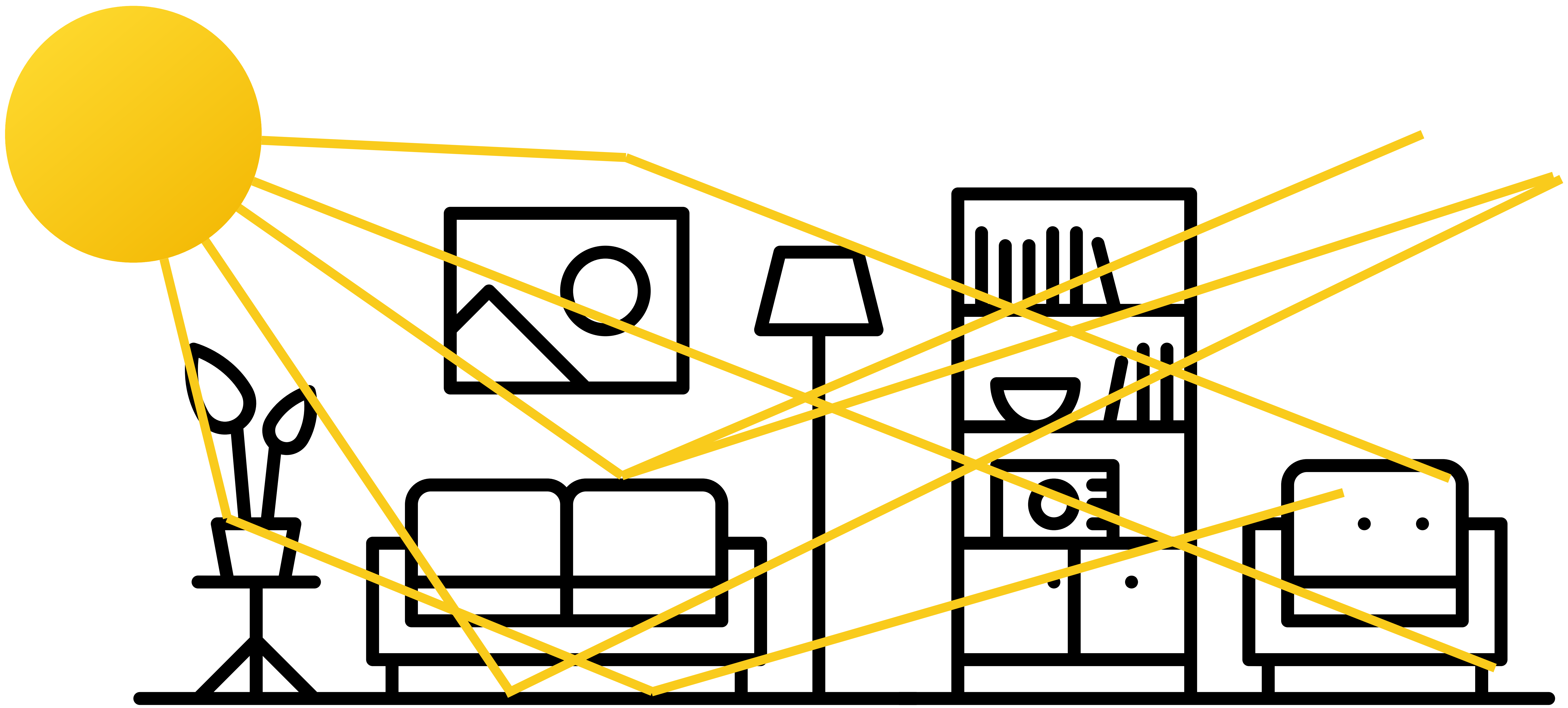
Some Slides adapted from…

- 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?

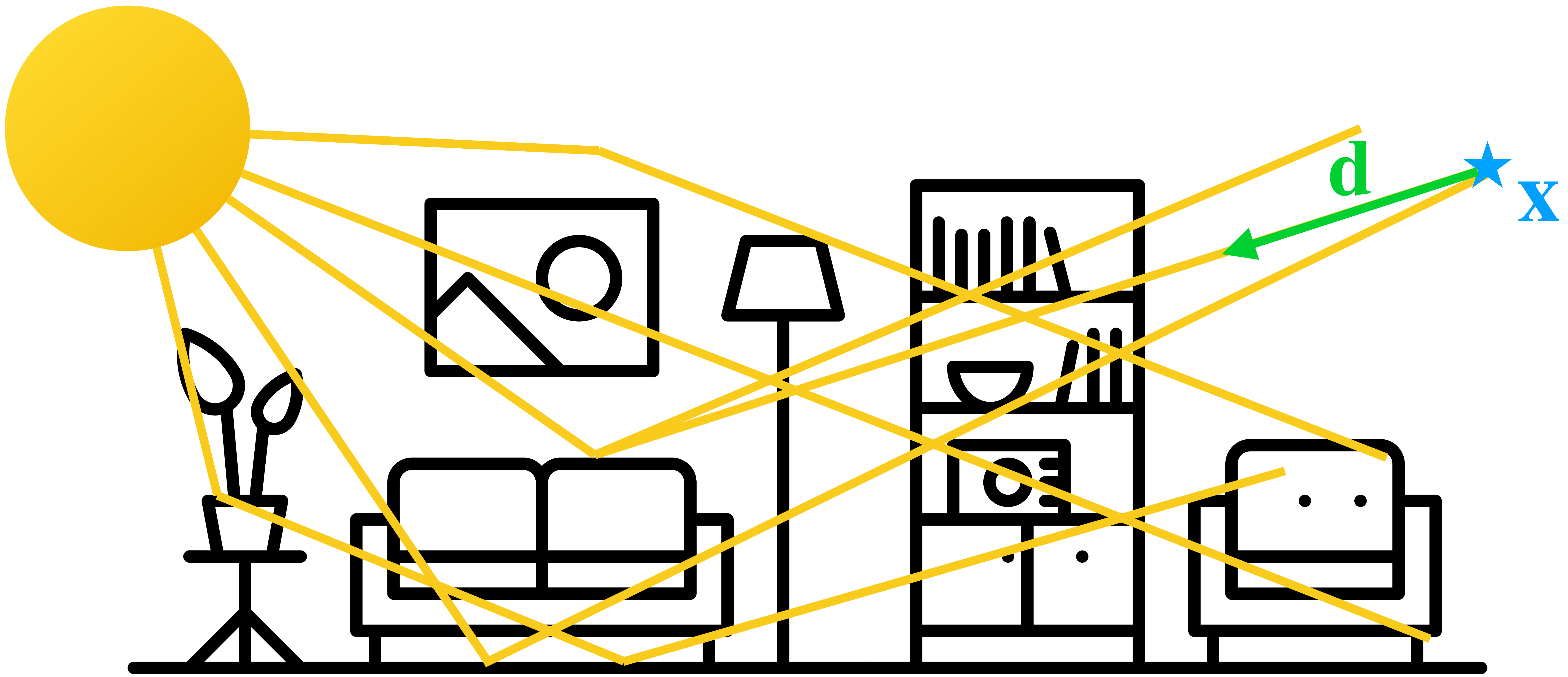
materials, light sources, 3D shape, color, weight, density, friction coefficients, etc

How do we observe scenes?



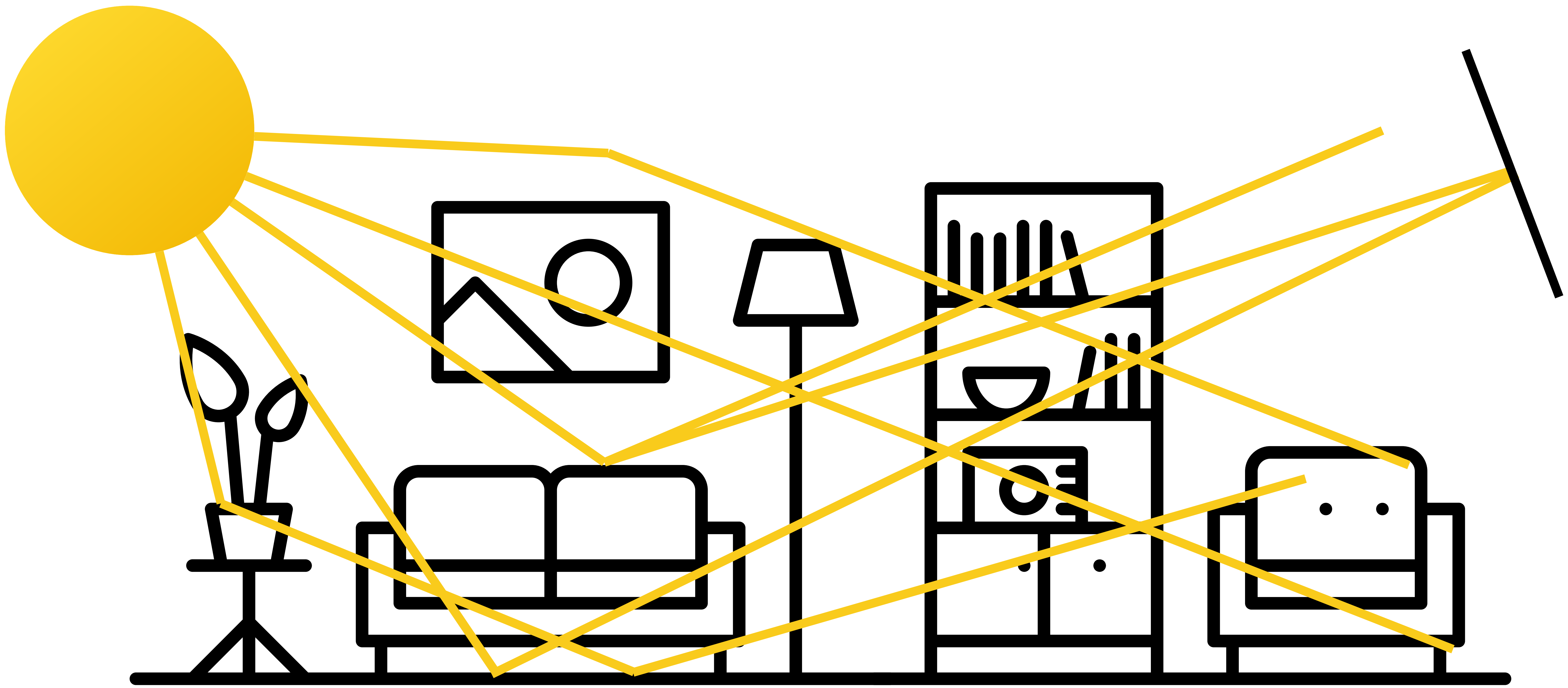
An eye (or a camera) observes a subset of all the light rays in a scene.

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

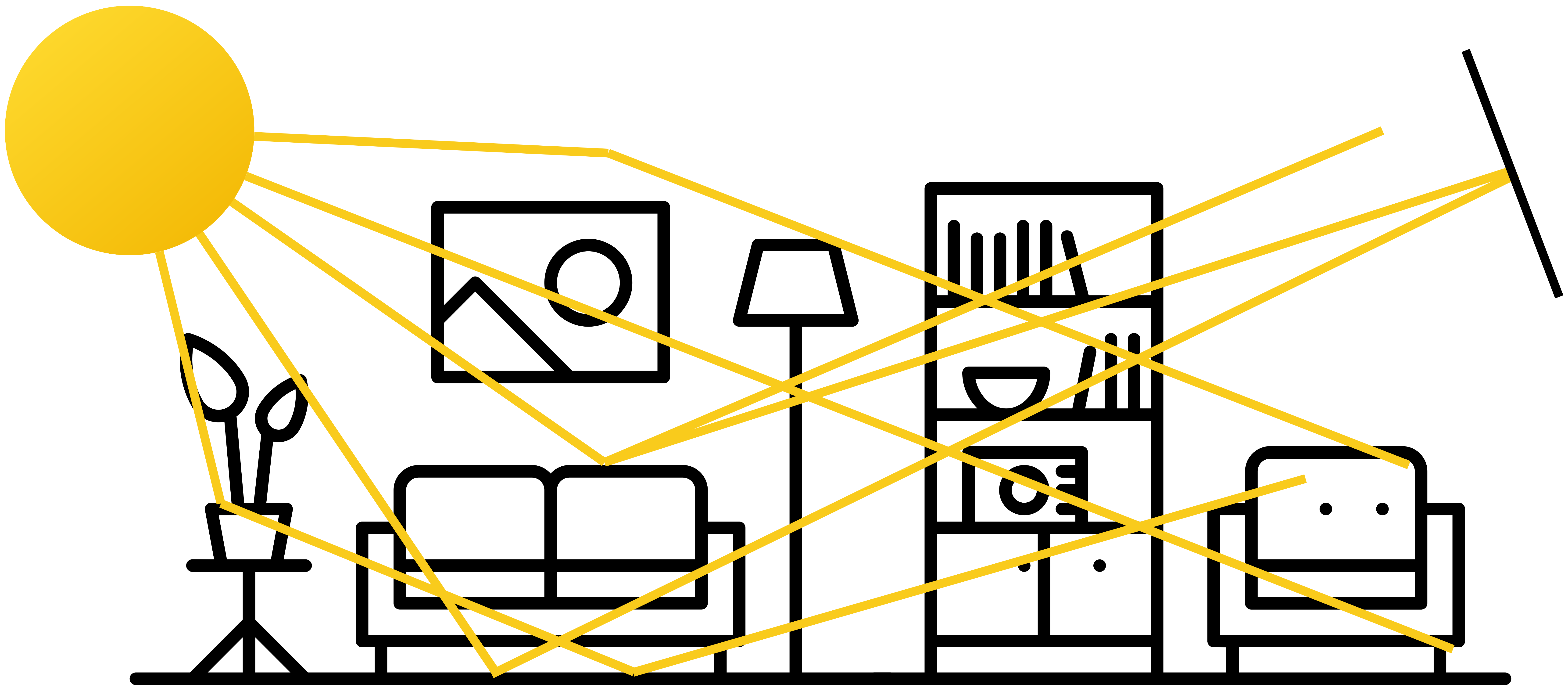


$$LF : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3, \quad LF(\mathbf{x}, \mathbf{d}) = \mathbf{c}$$

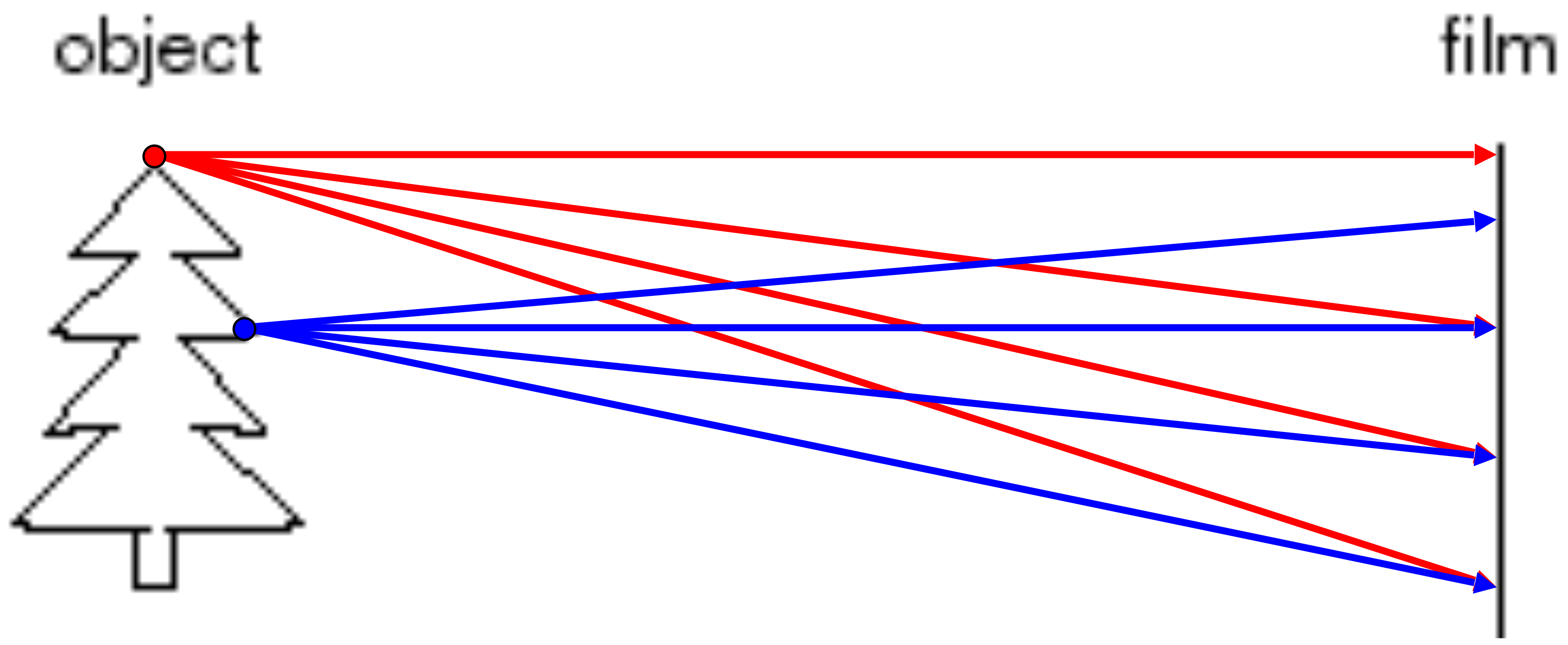
Why don’t we get an image if we hold up a piece of paper?



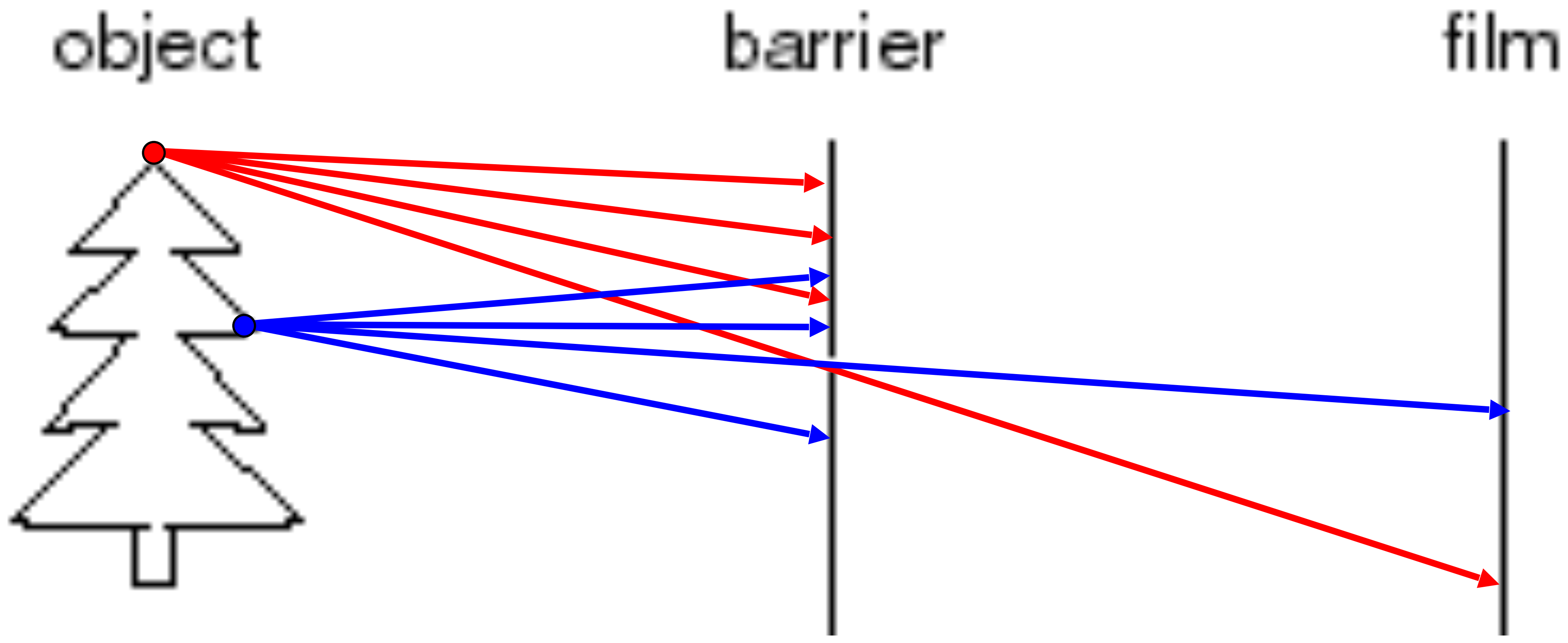
Every “pixel” on the paper is the average of all the rays in the scene.



Let’s make a Camera!



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

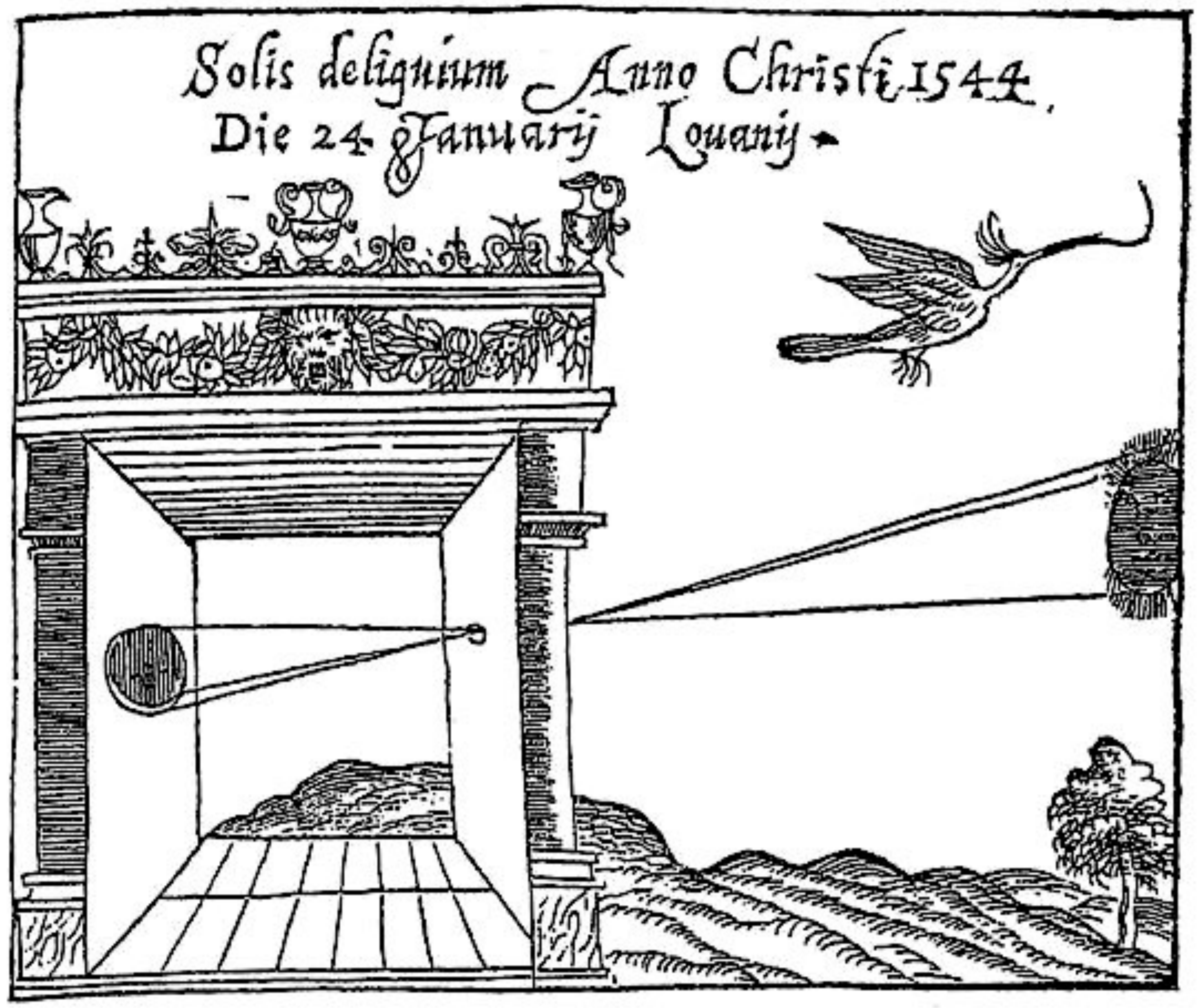


Add a barrier to block most rays.

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

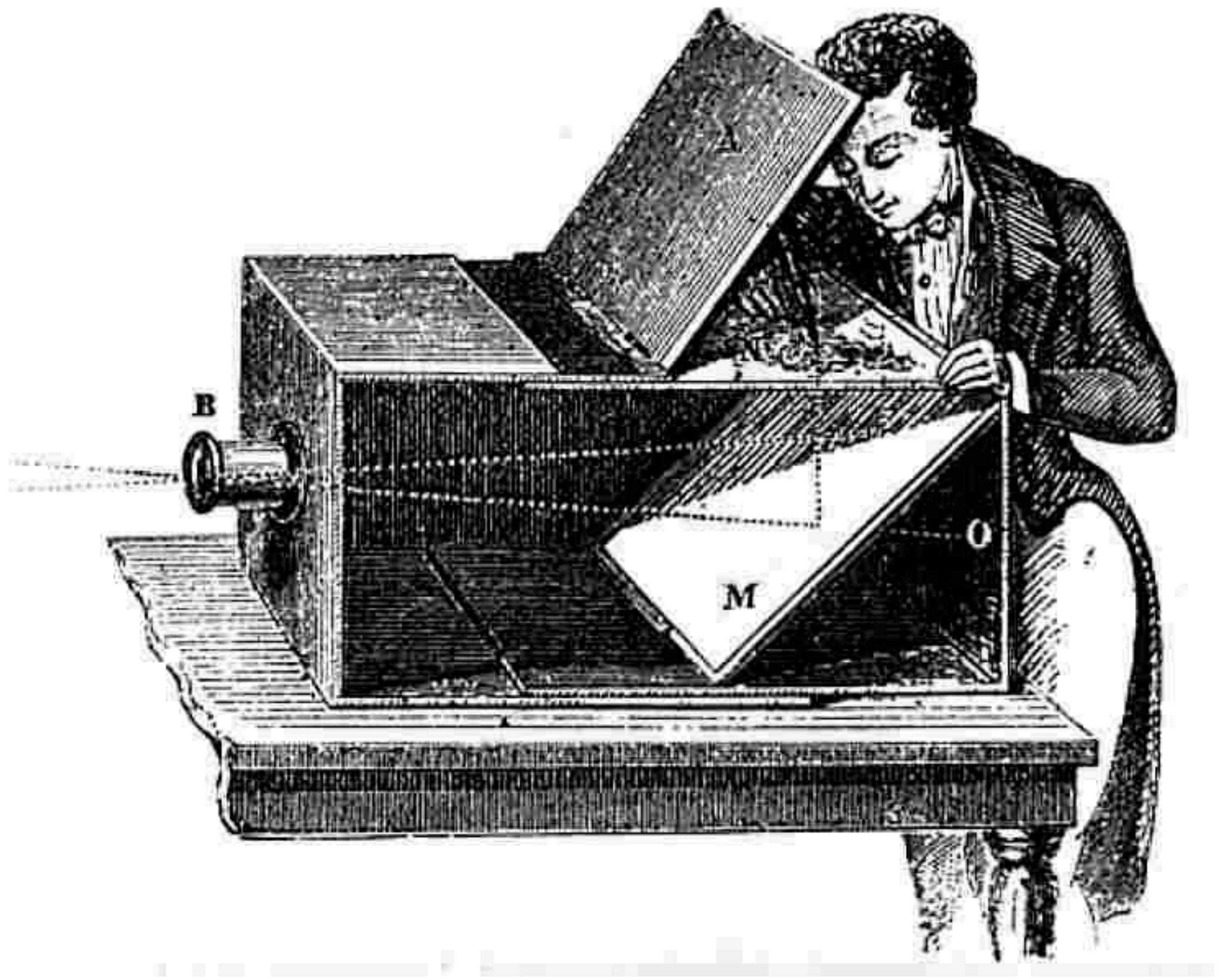
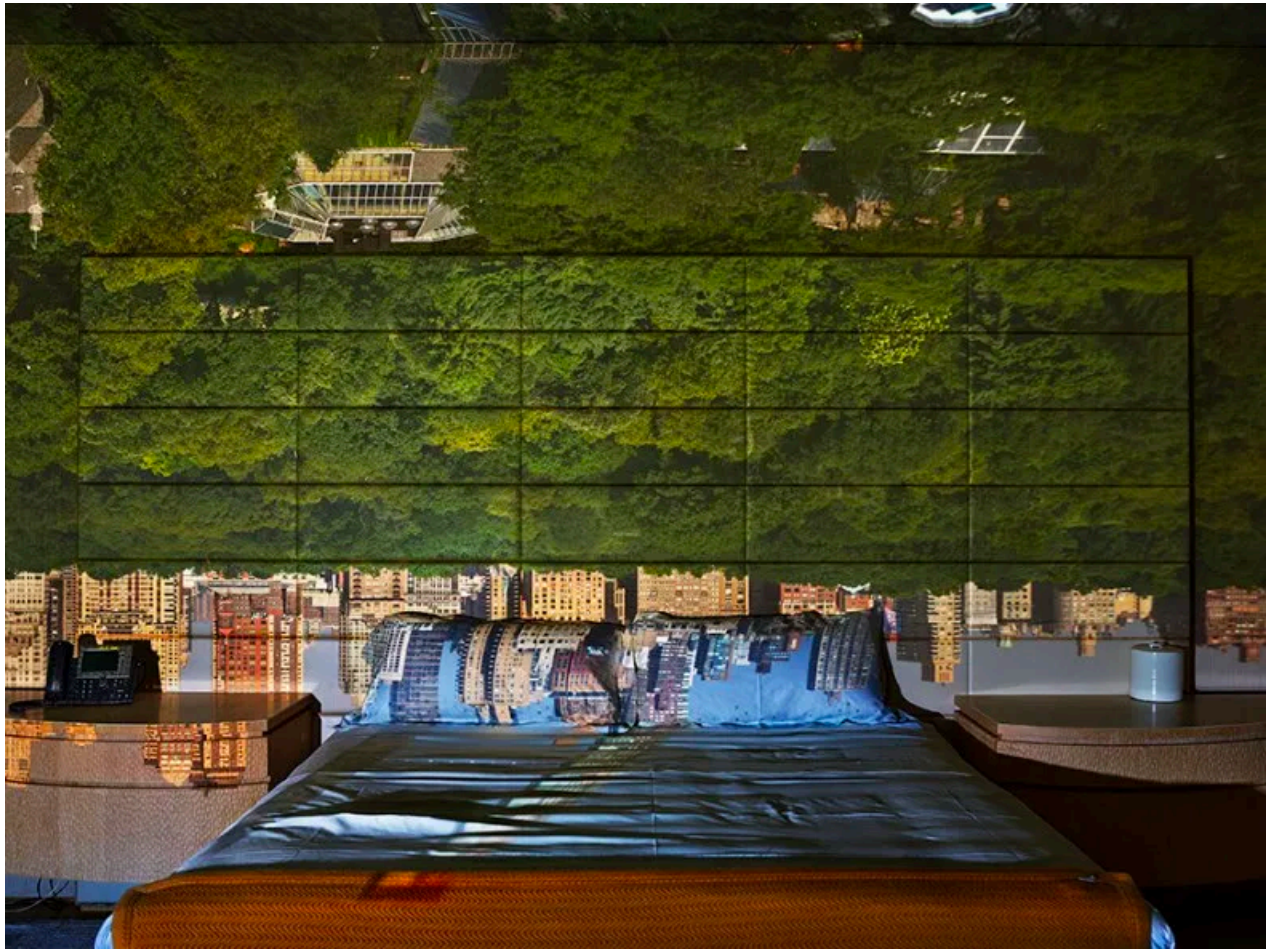


Image source

Camera Obscura (*Dark Room*)



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

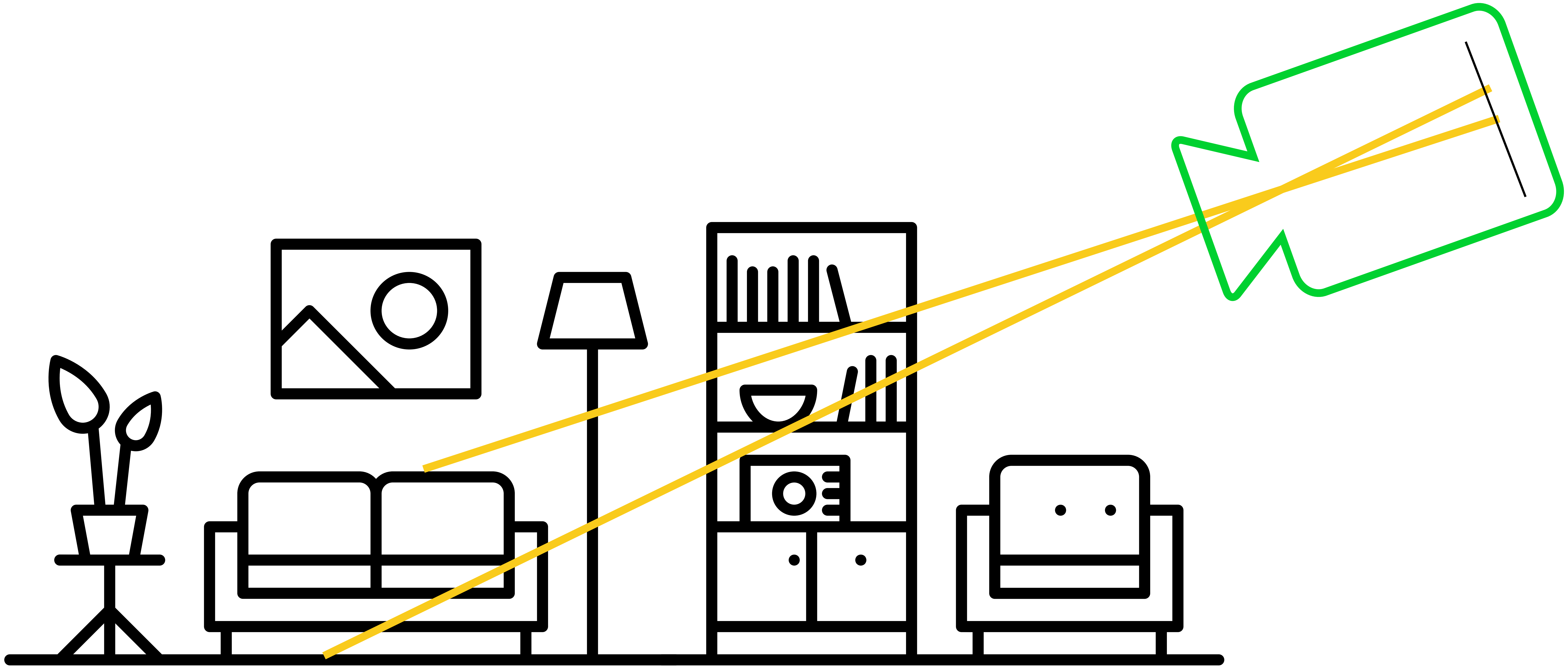
After scouting rooms and reserving one for at least a day, Morell masks the windows except for the aperture. He controls three elements: the size of the hole, with a smaller one yielding a sharper but dimmer image; the length of the exposure, usually eight hours; and the distance from the hole to the 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."

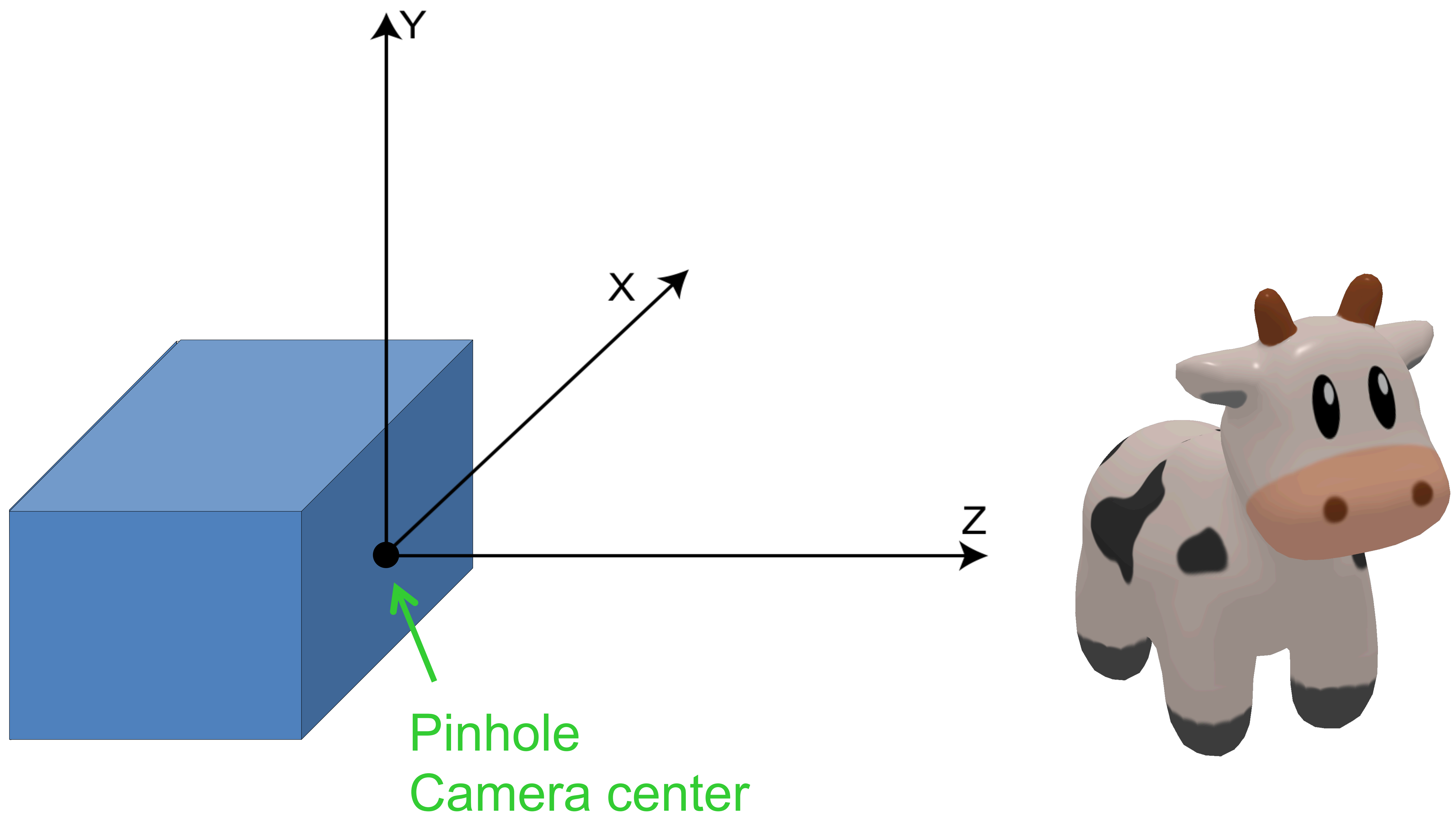
http://www.abelardomorell.net/project/camera-obscura/

Slide credit: Lana Lazebnik

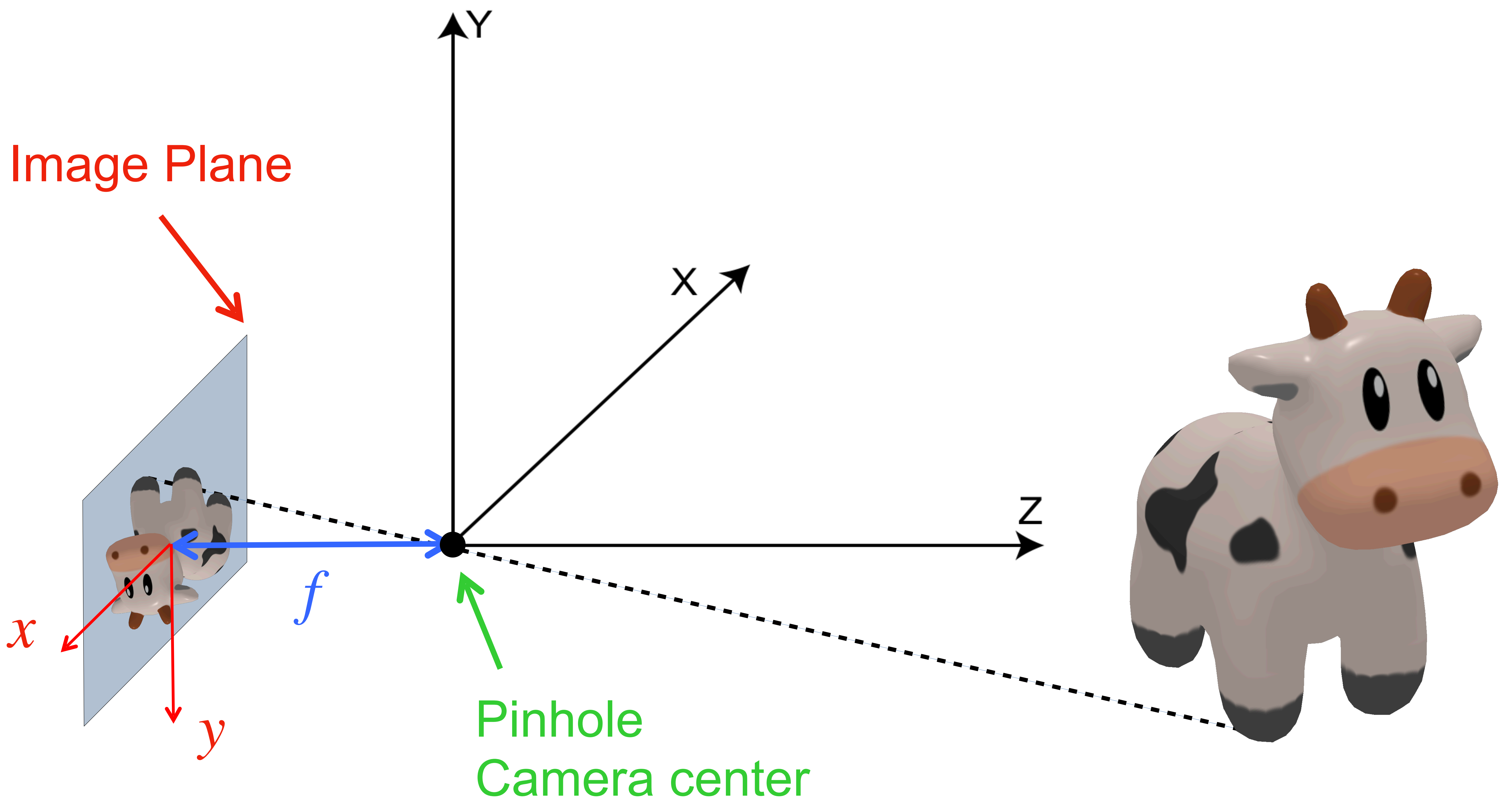
Today: Model the camera, describe the rays.



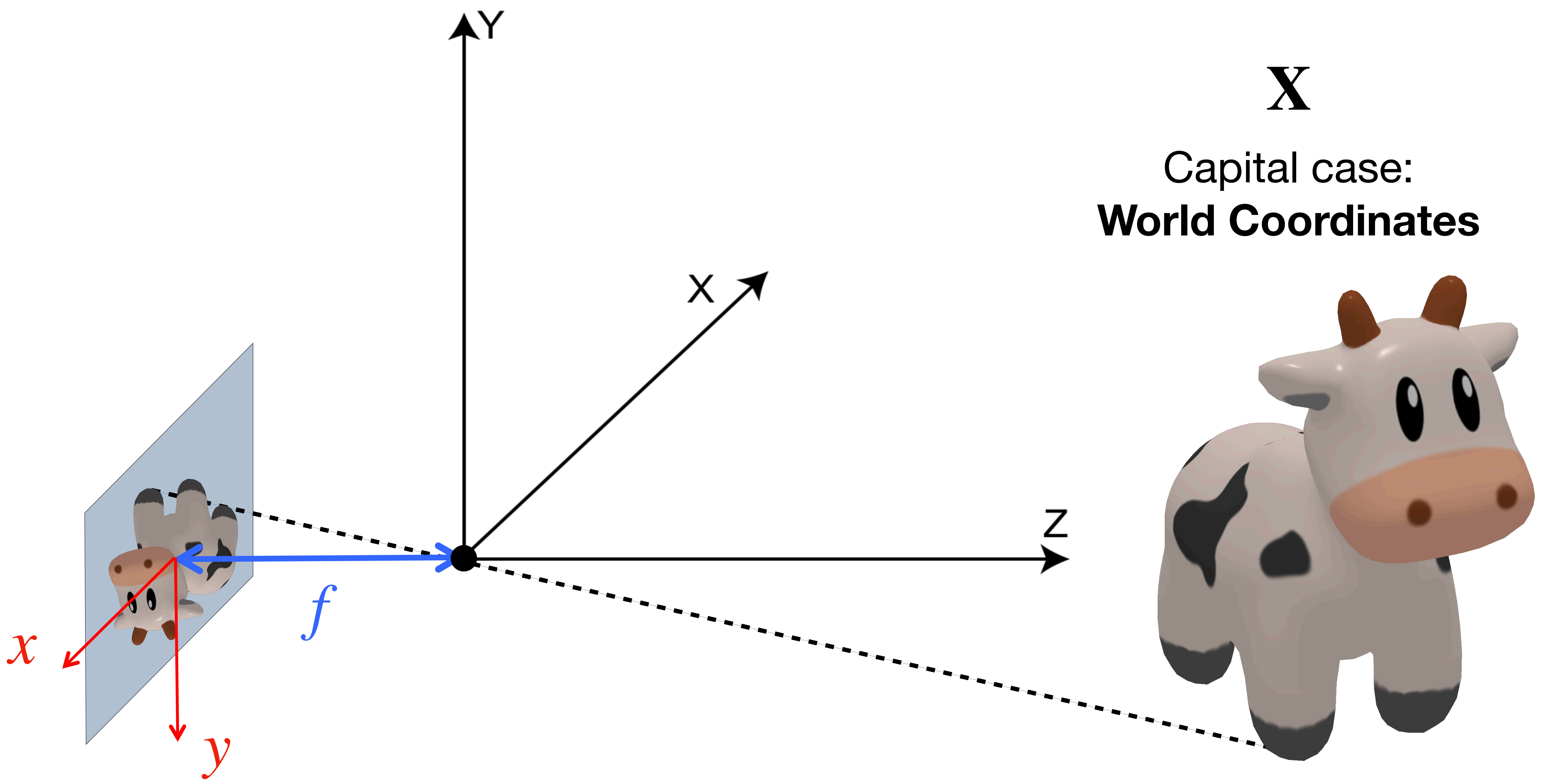
Perspective projection



Perspective projection



Perspective projection



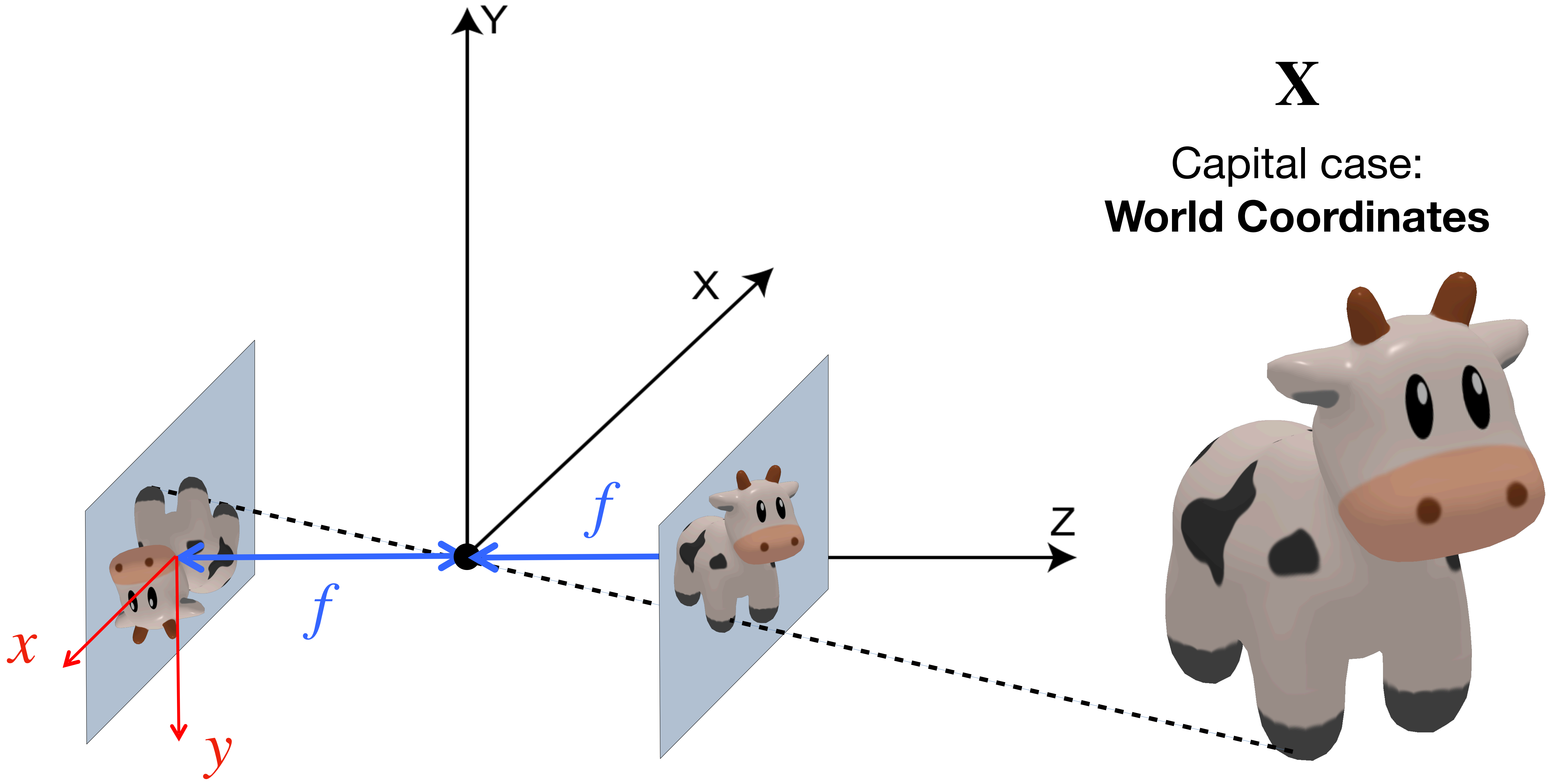
$\mathbf{X}$

Capital case:
World Coordinates

$\mathbf{x}$

Lowercase:
Pixel / Image Coordinates

Perspective projection

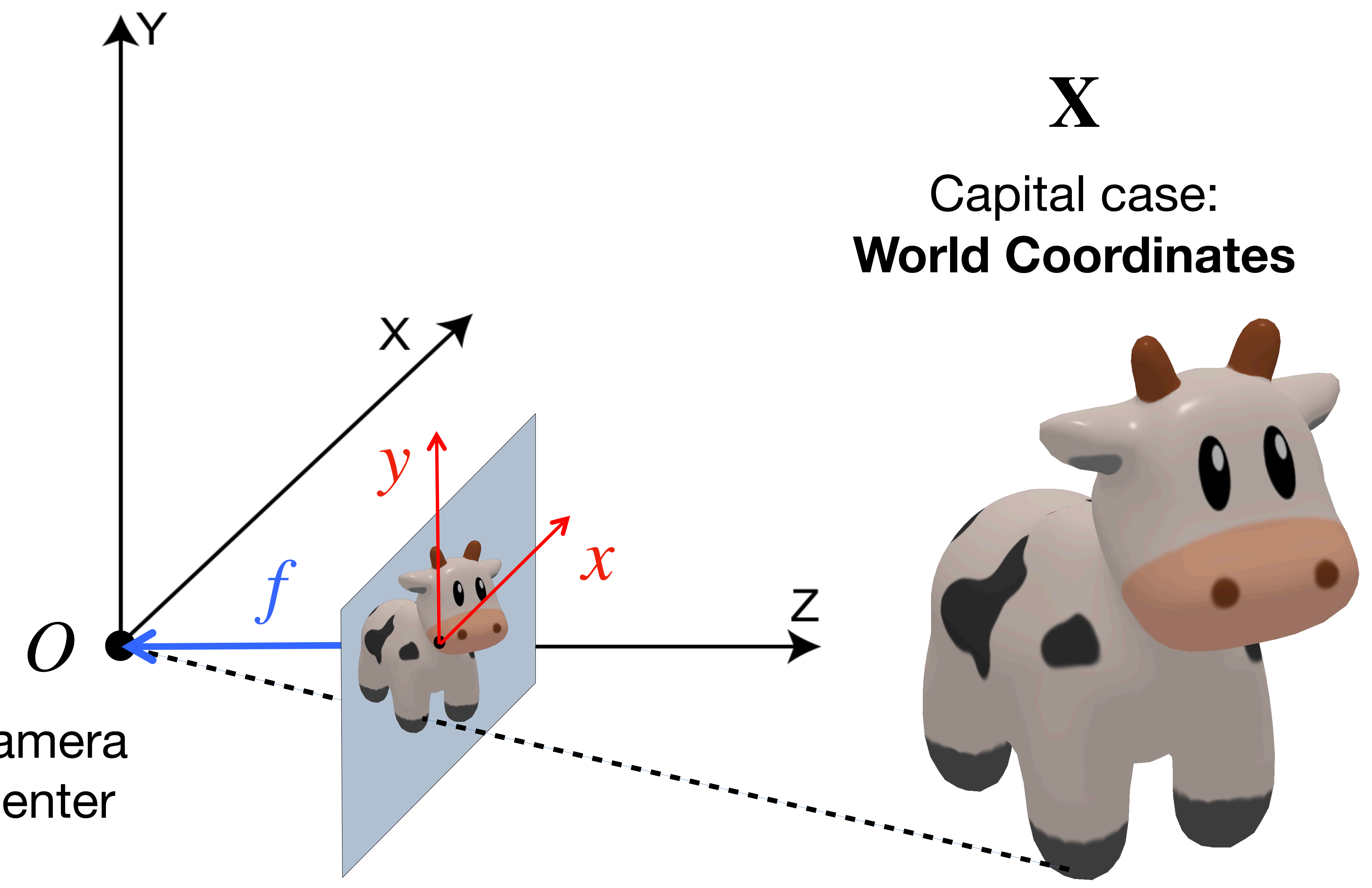


$\mathbf{x}$

Lowercase:

Pixel / Image Coordinates

Perspective projection



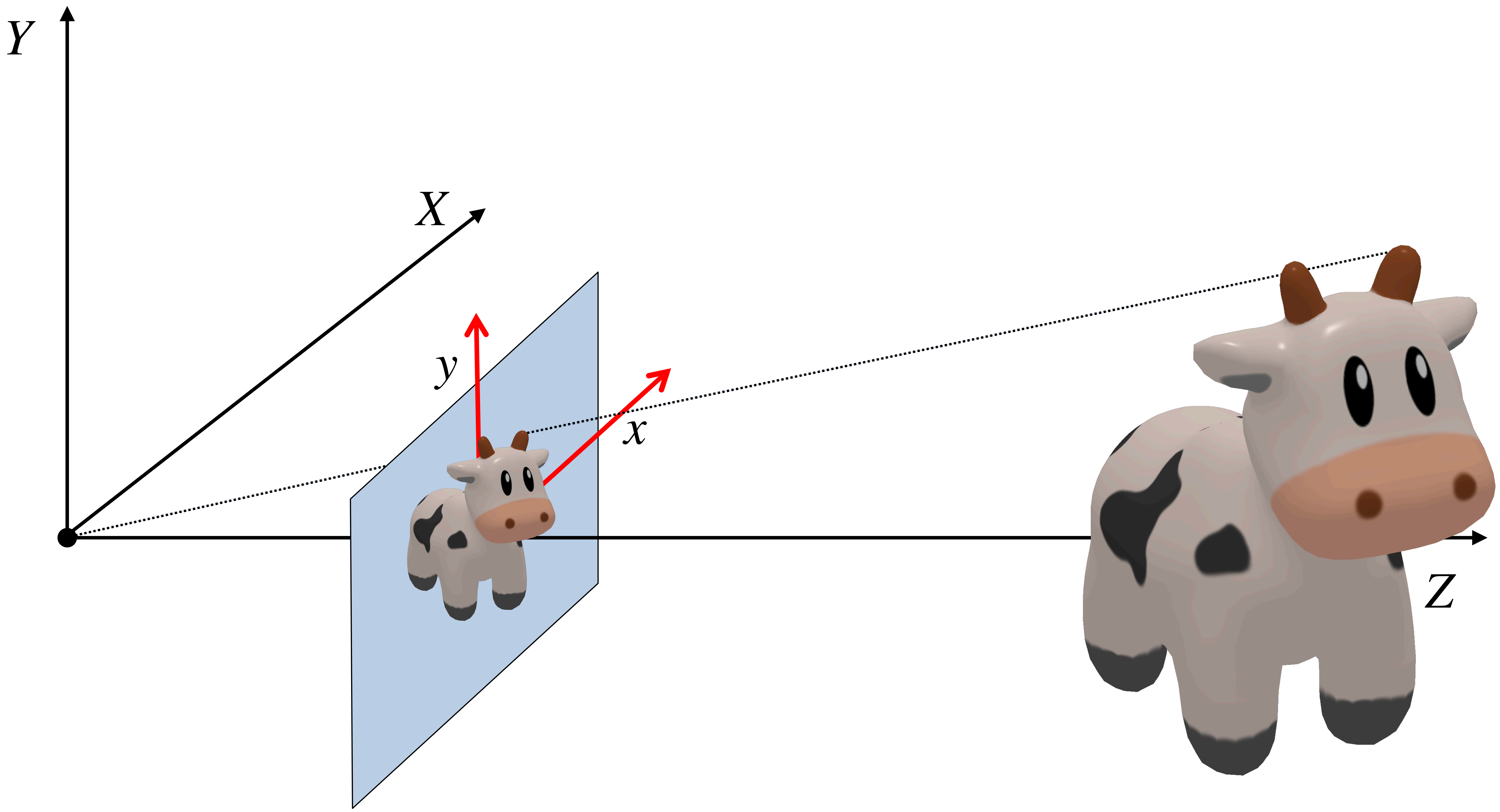
X

Capital case:
World Coordinates

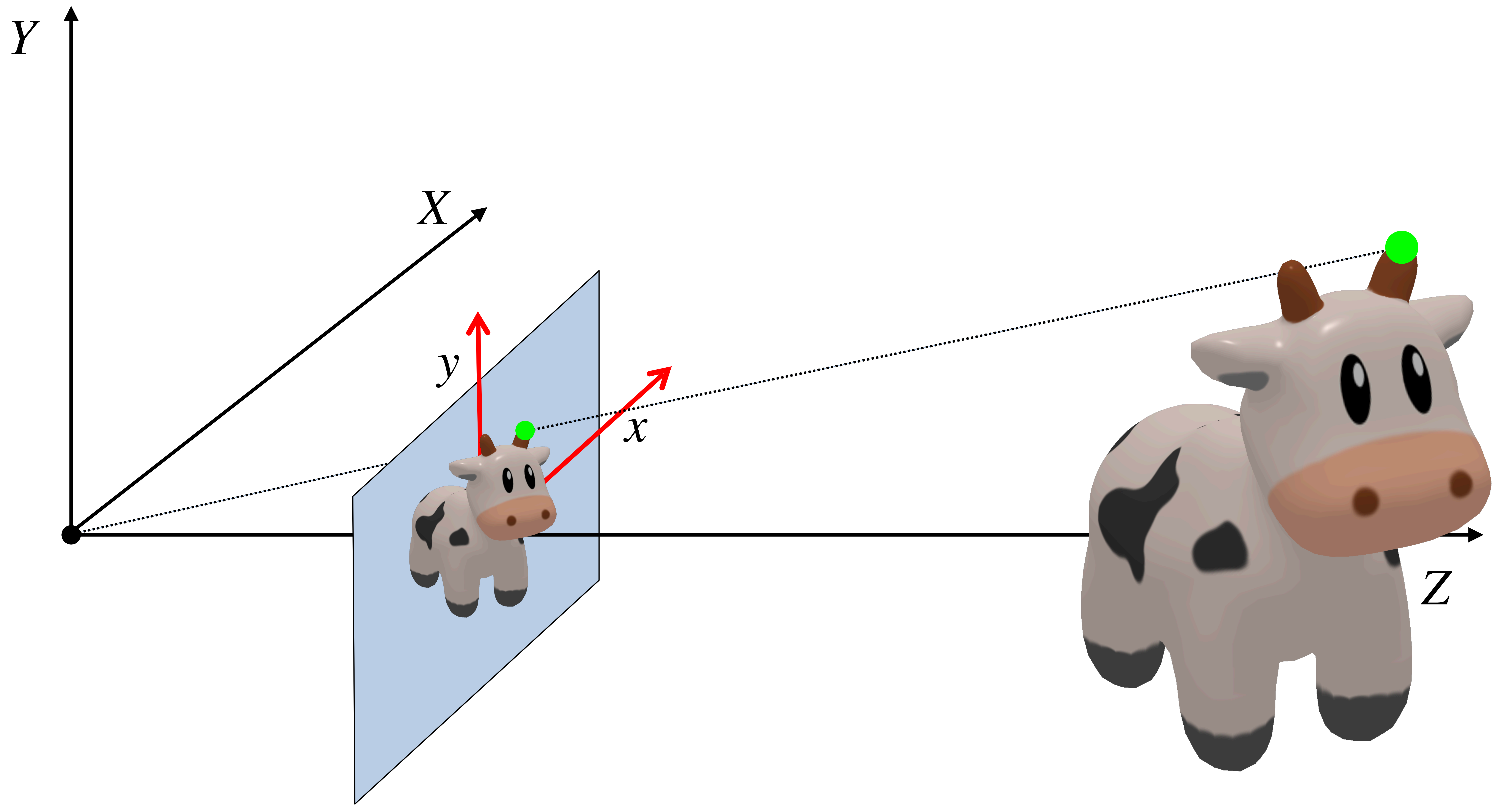
x

Lowercase:
Pixel / Image Coordinates

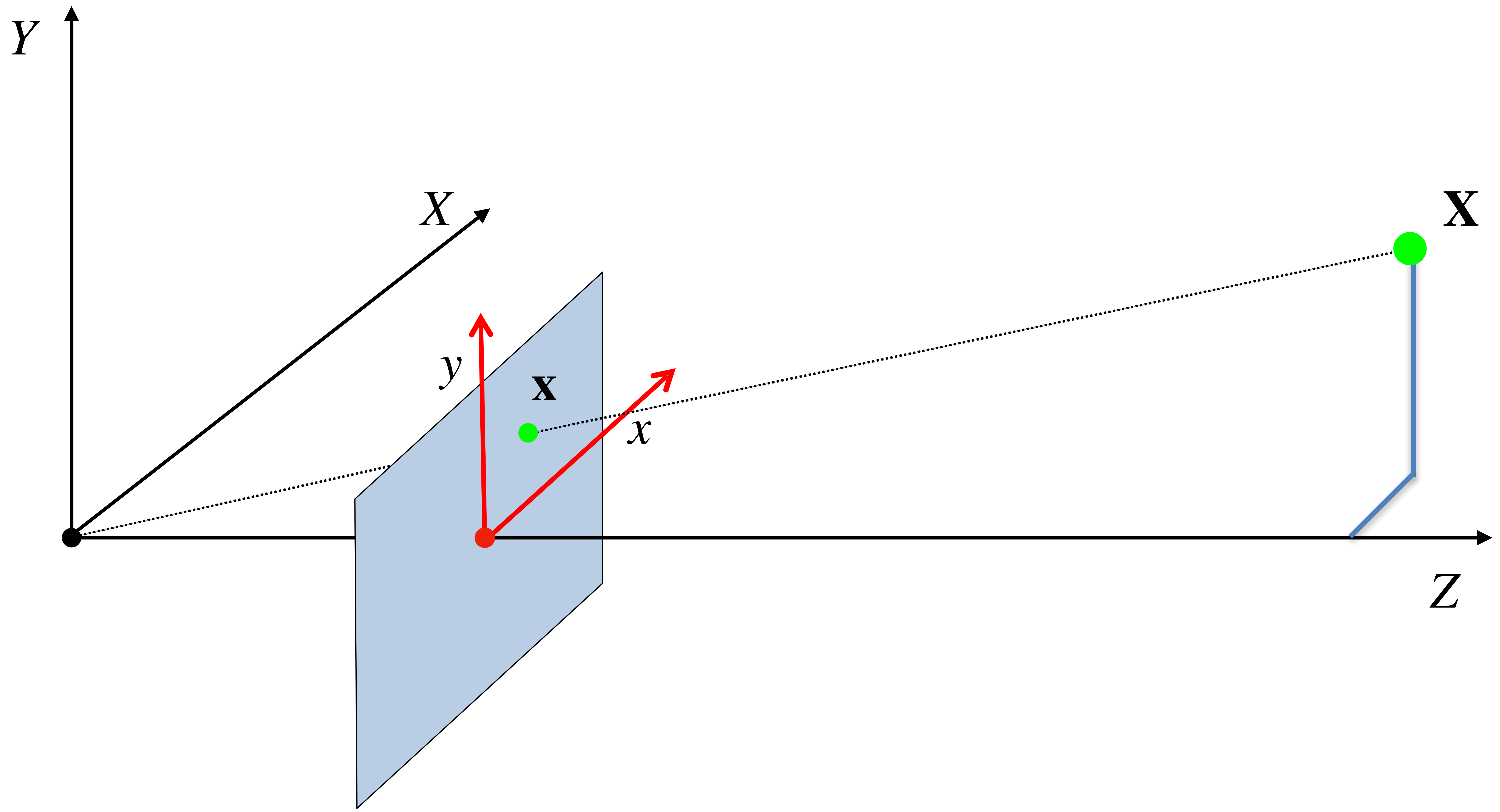
Perspective projection



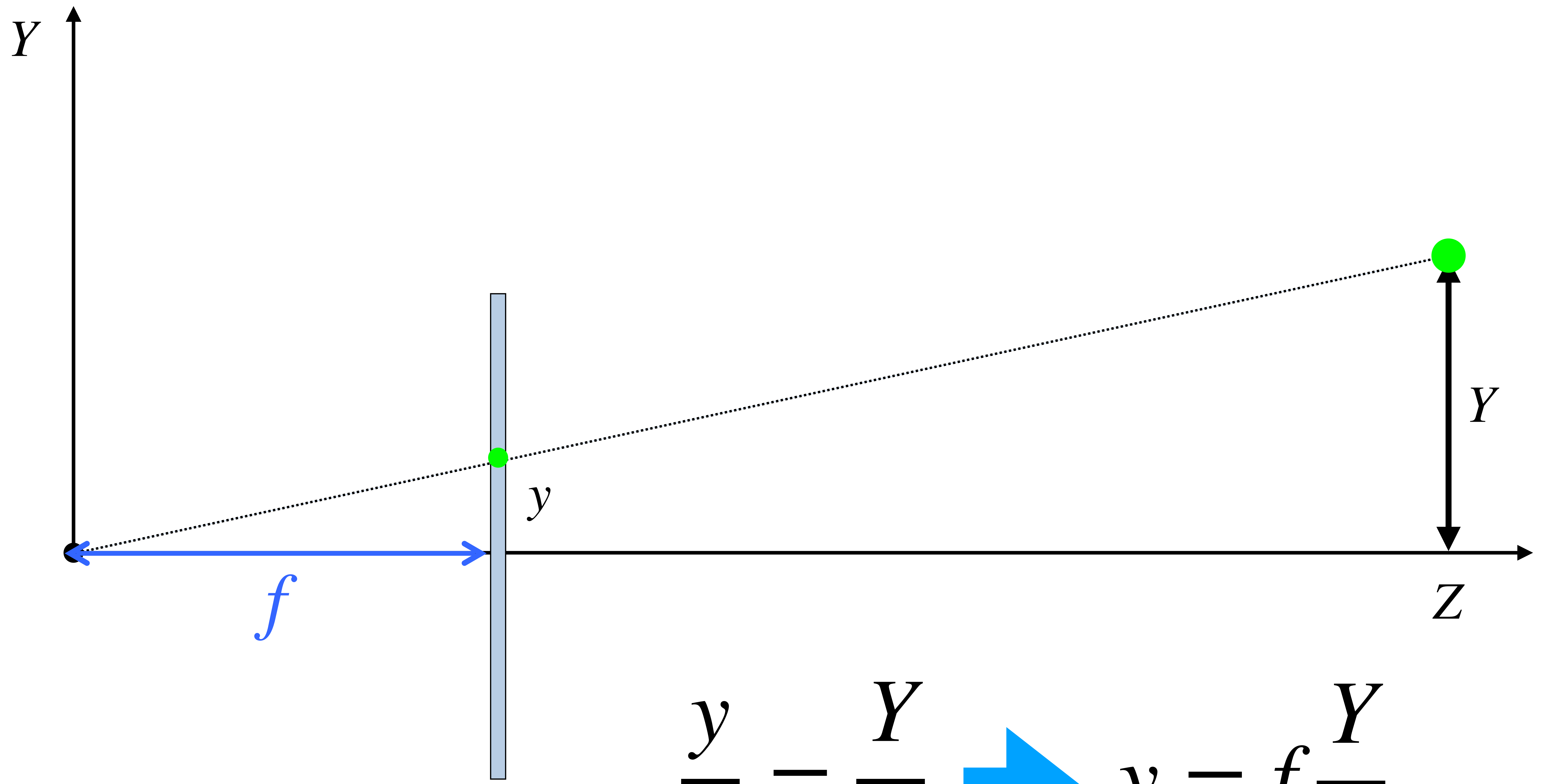
Perspective projection



Perspective projection

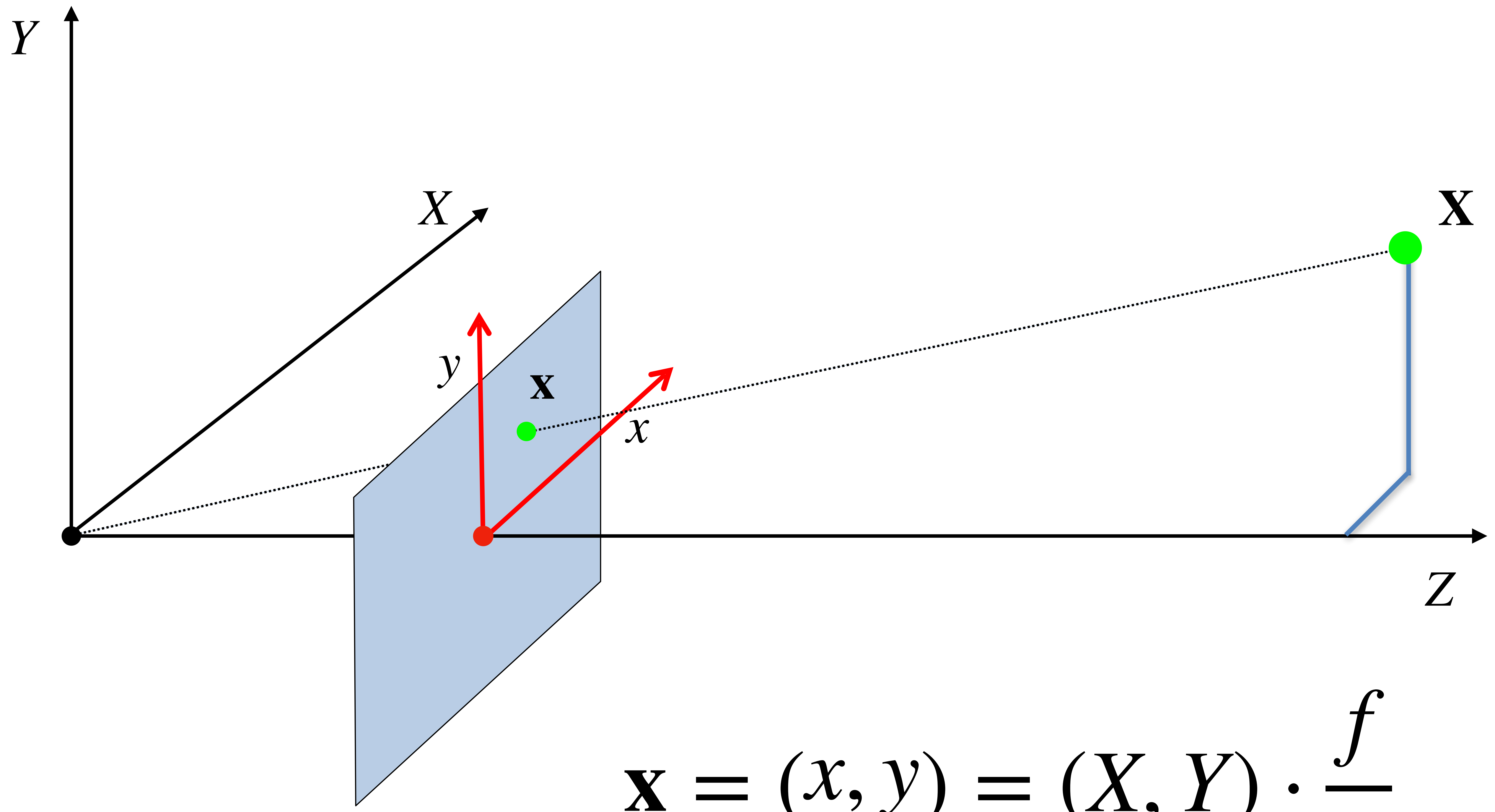


Perspective projection



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

Perspective projection



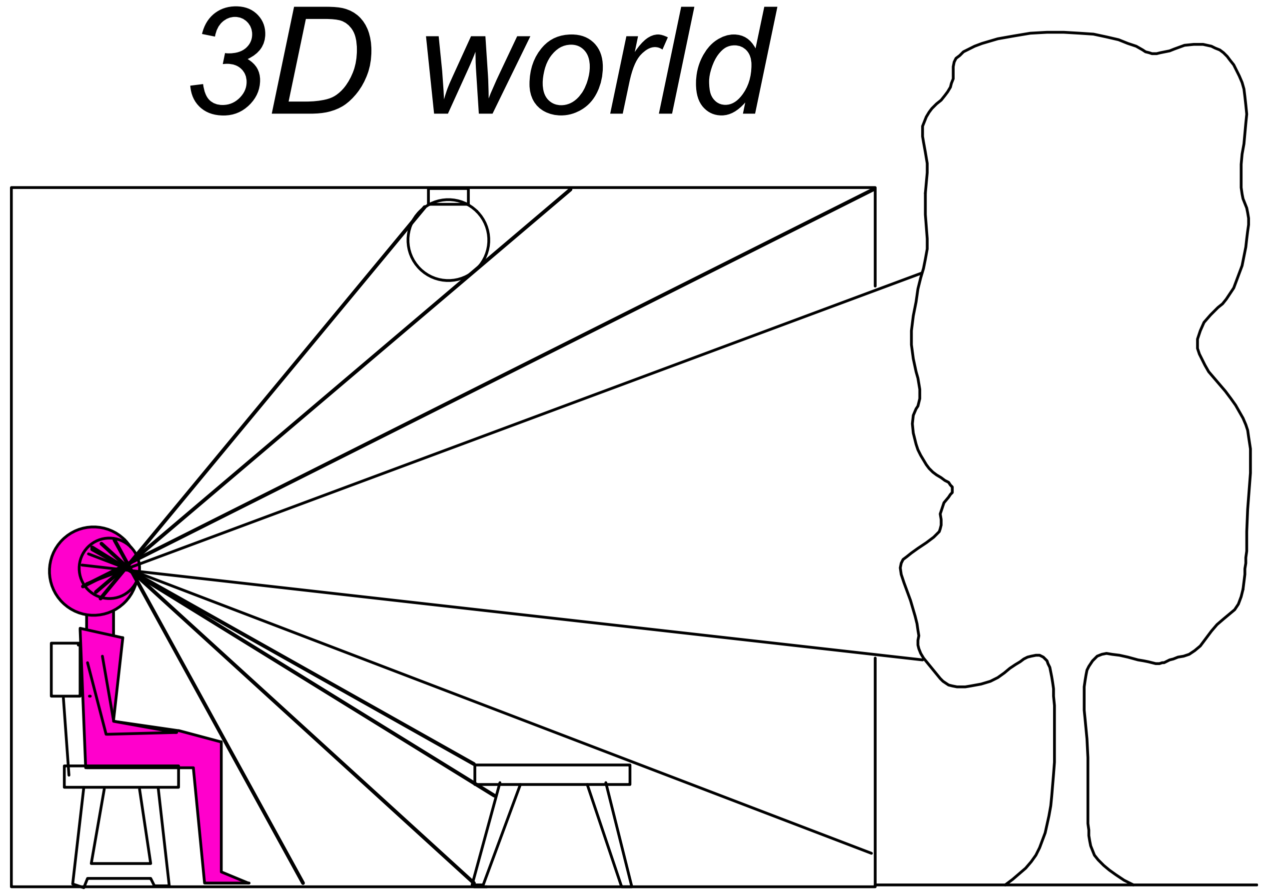
$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{f}{Z}$$

Image / Pixel Coordinates

World Coordinates

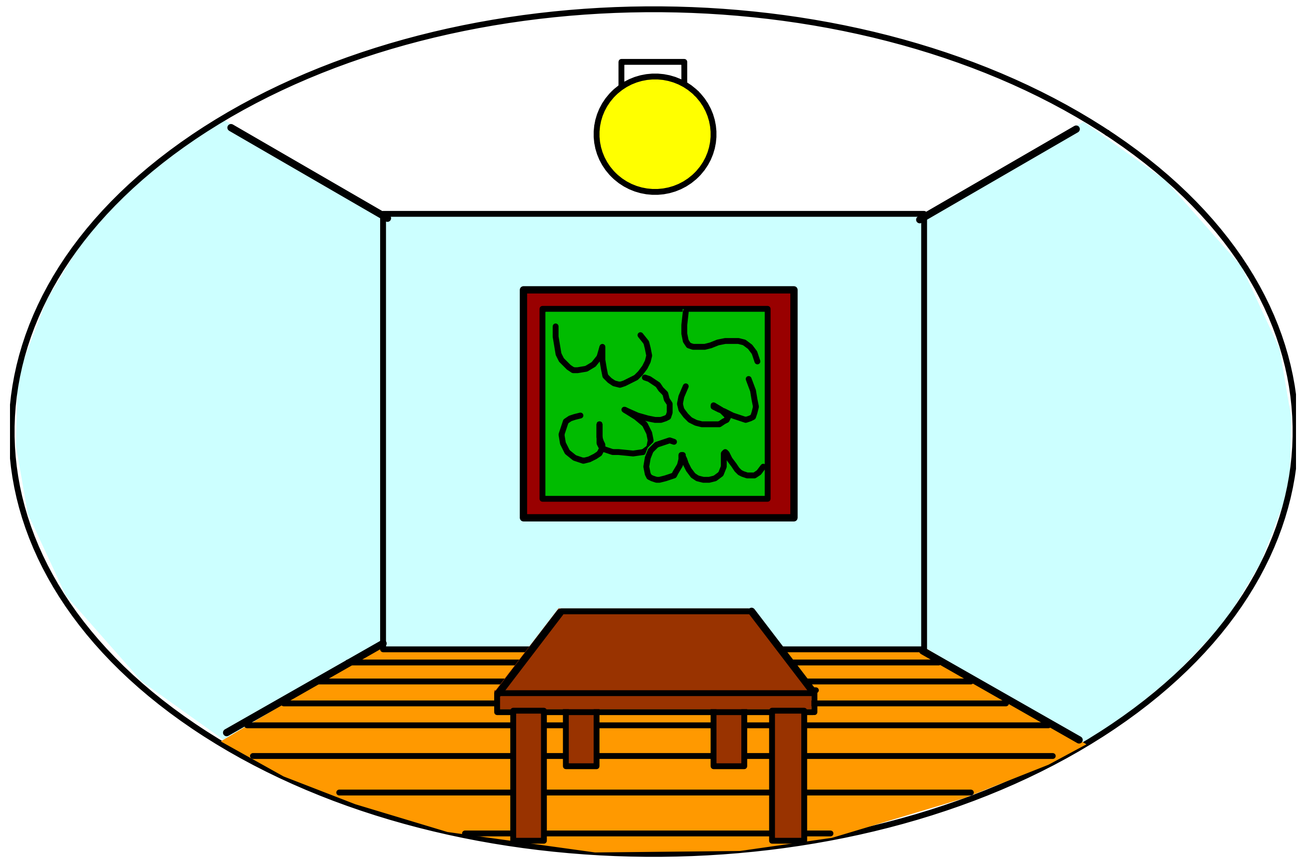
Perspective Projection

3D world



Point of observation

2D image



What properties of the world are preserved?

- Straight lines, incidence

What properties are not preserved?

- Angles, lengths

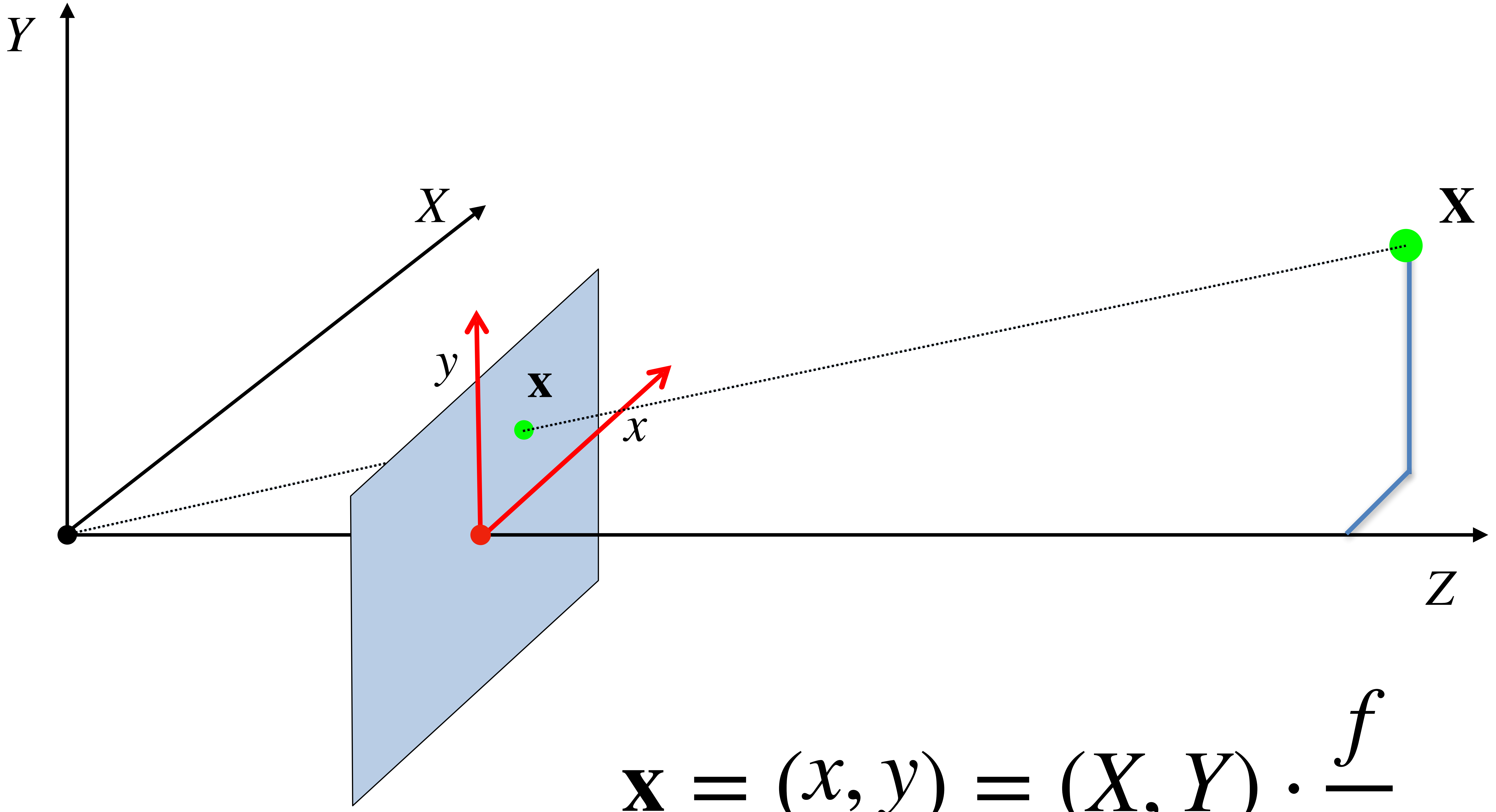
Perspective Projection

Video source: [Camera Obscura & World of Illusions]

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

Questions?

Perspective projection

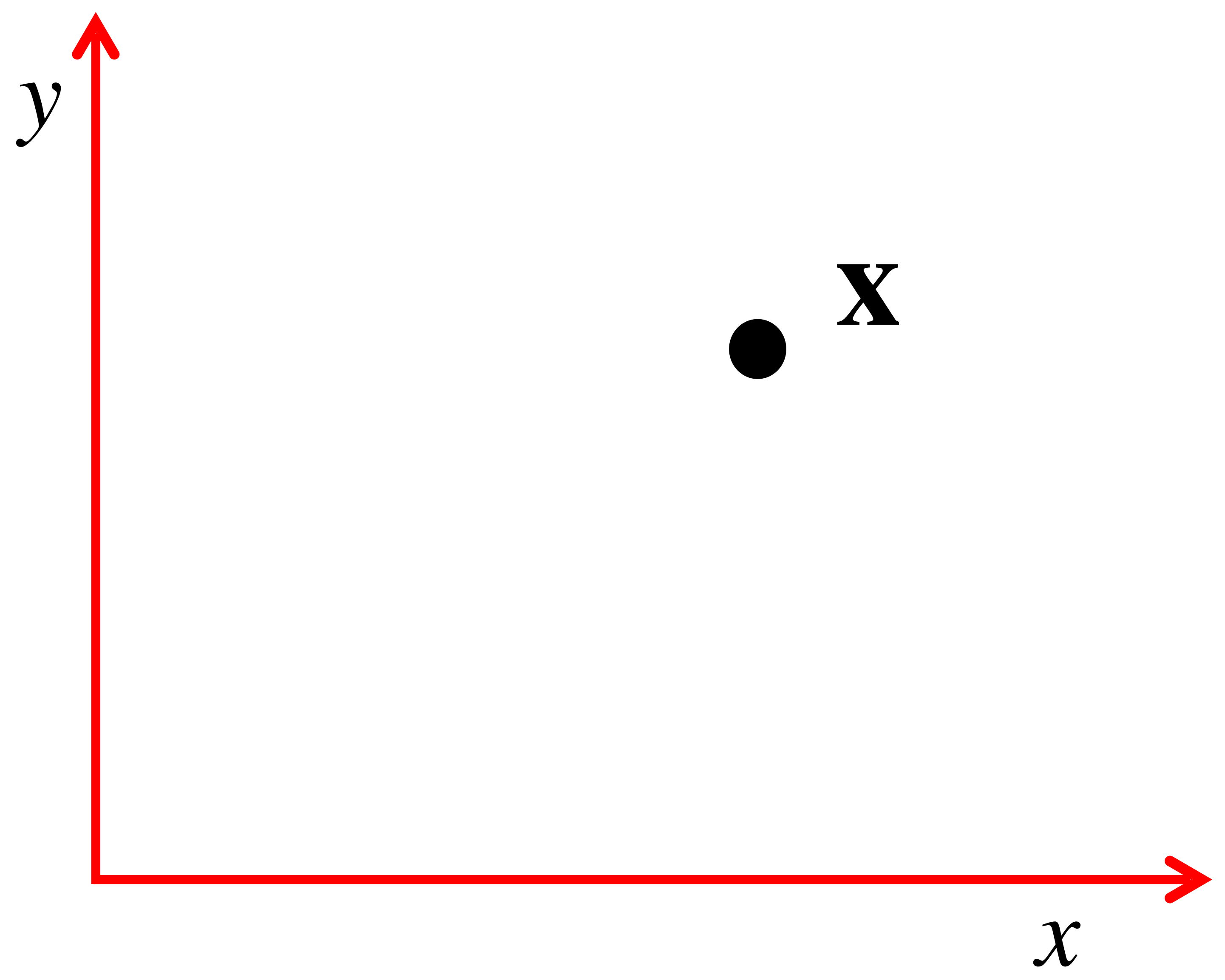


$$\mathbf{x} = (x, y) = (X, Y) \cdot \frac{f}{Z}$$

This is awkward... Not a linear operator.

Homogeneous coordinates

2D



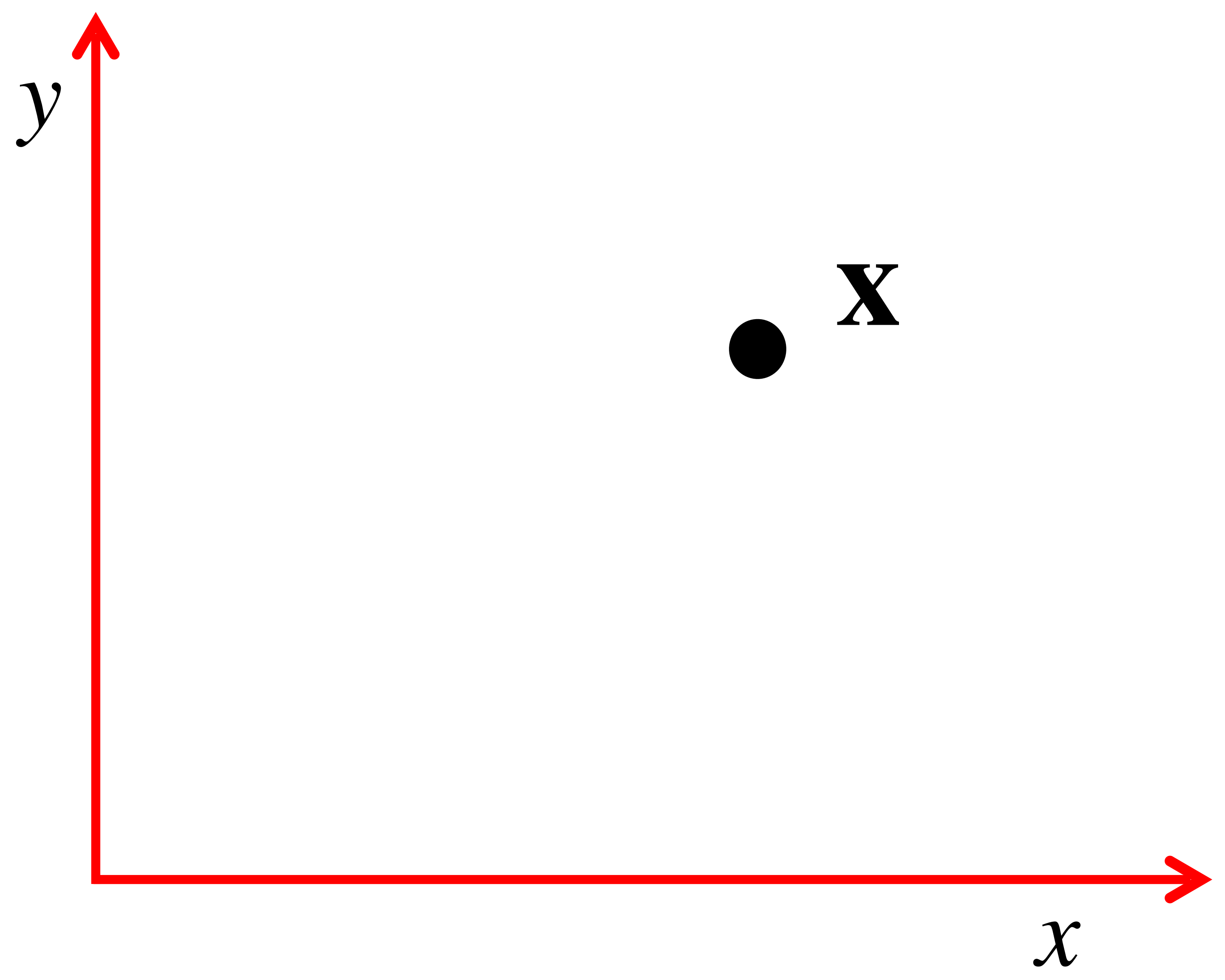
$$\mathbf{x} = (x, y) \quad \Rightarrow \quad \tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

heterogeneous coordinates

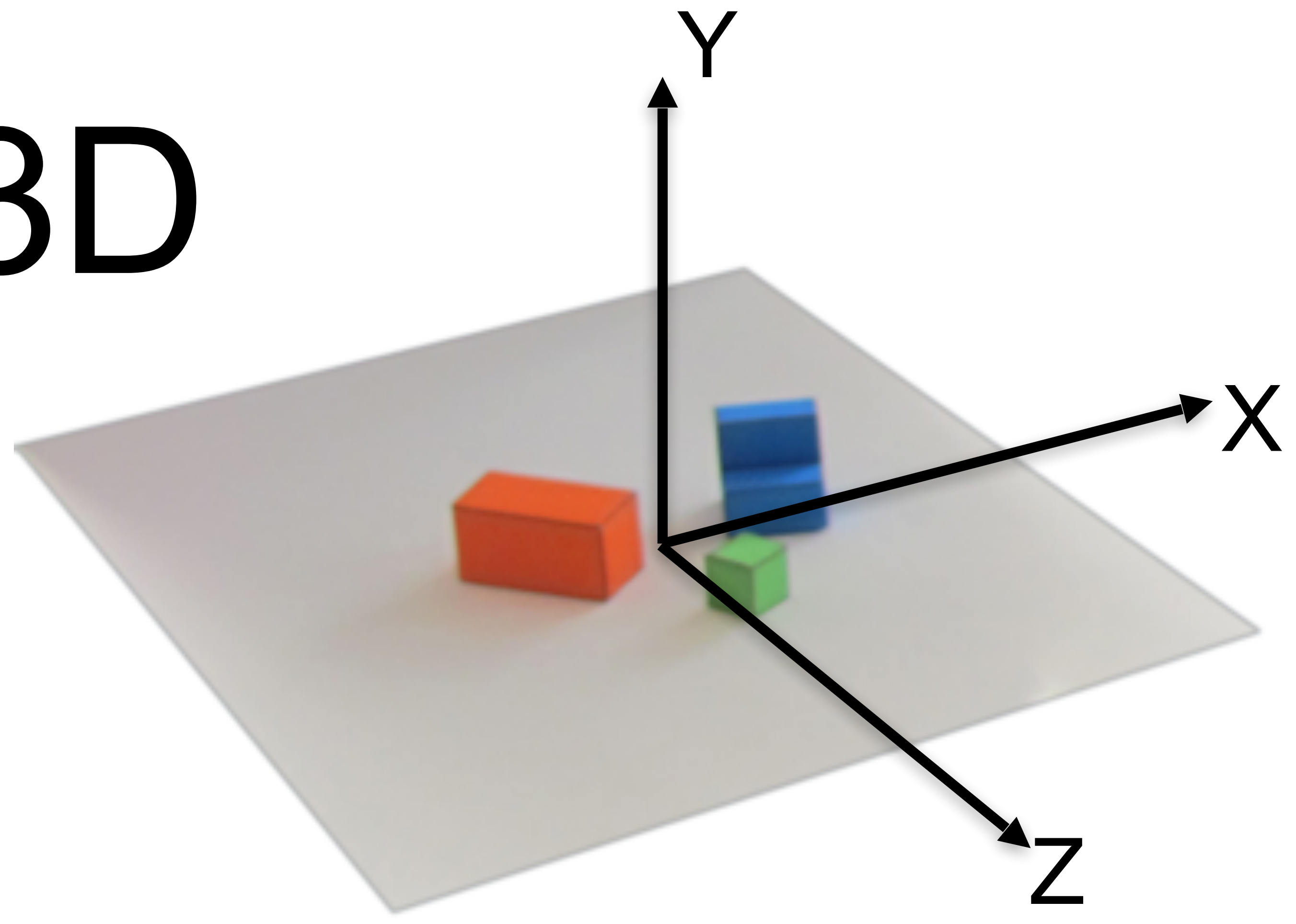
homogeneous coordinates

Homogeneous coordinates

2D



3D



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

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

heterogeneous coordinates

homogeneous coordinates

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:

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

Homogeneous coordinates

2D

$\mathbb{R}^2$ → $\mathbb{P}^2$

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

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

3D

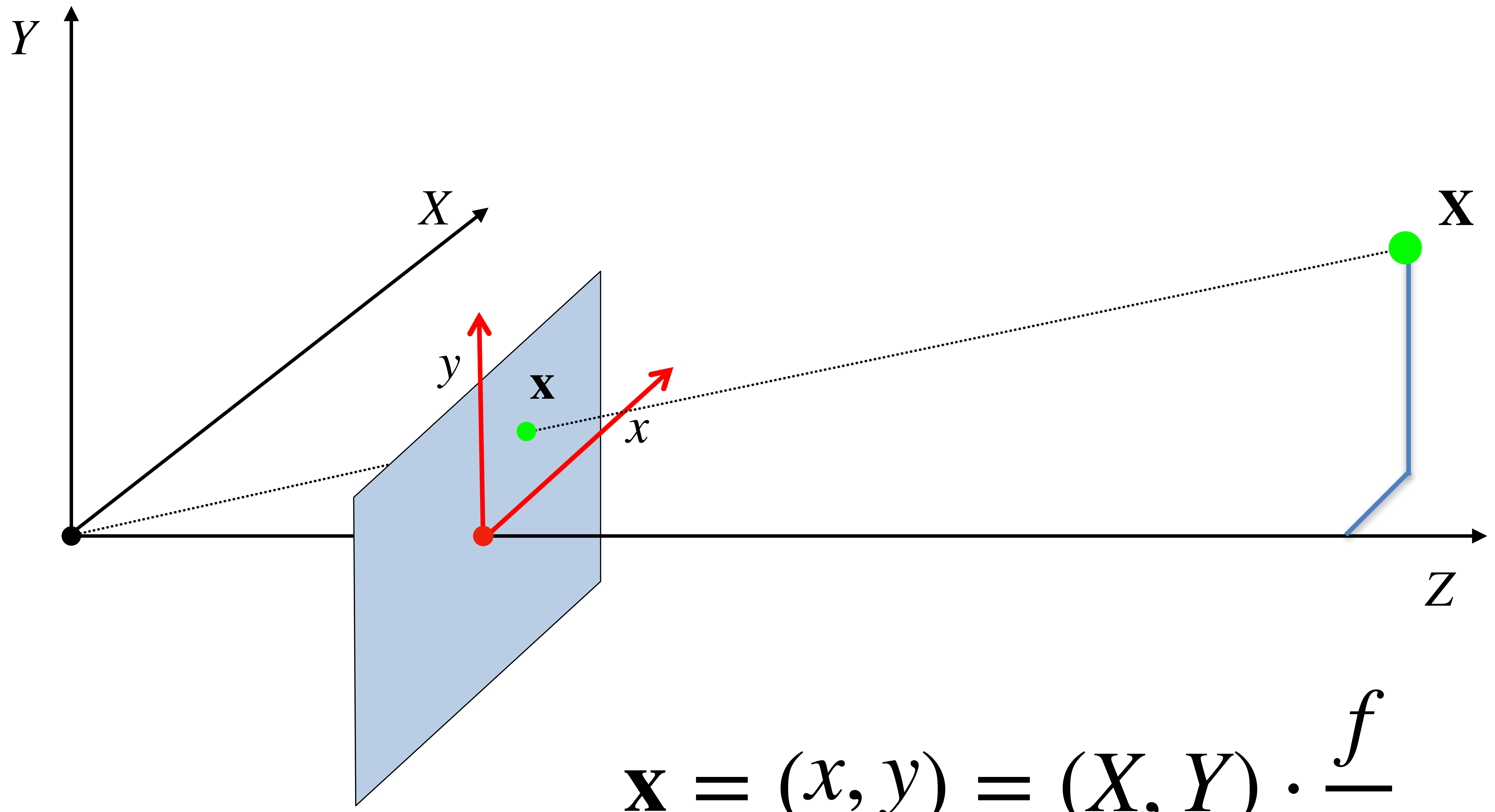
$\mathbb{R}^3$ → $\mathbb{P}^3$

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

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

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.

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

Perspective projection

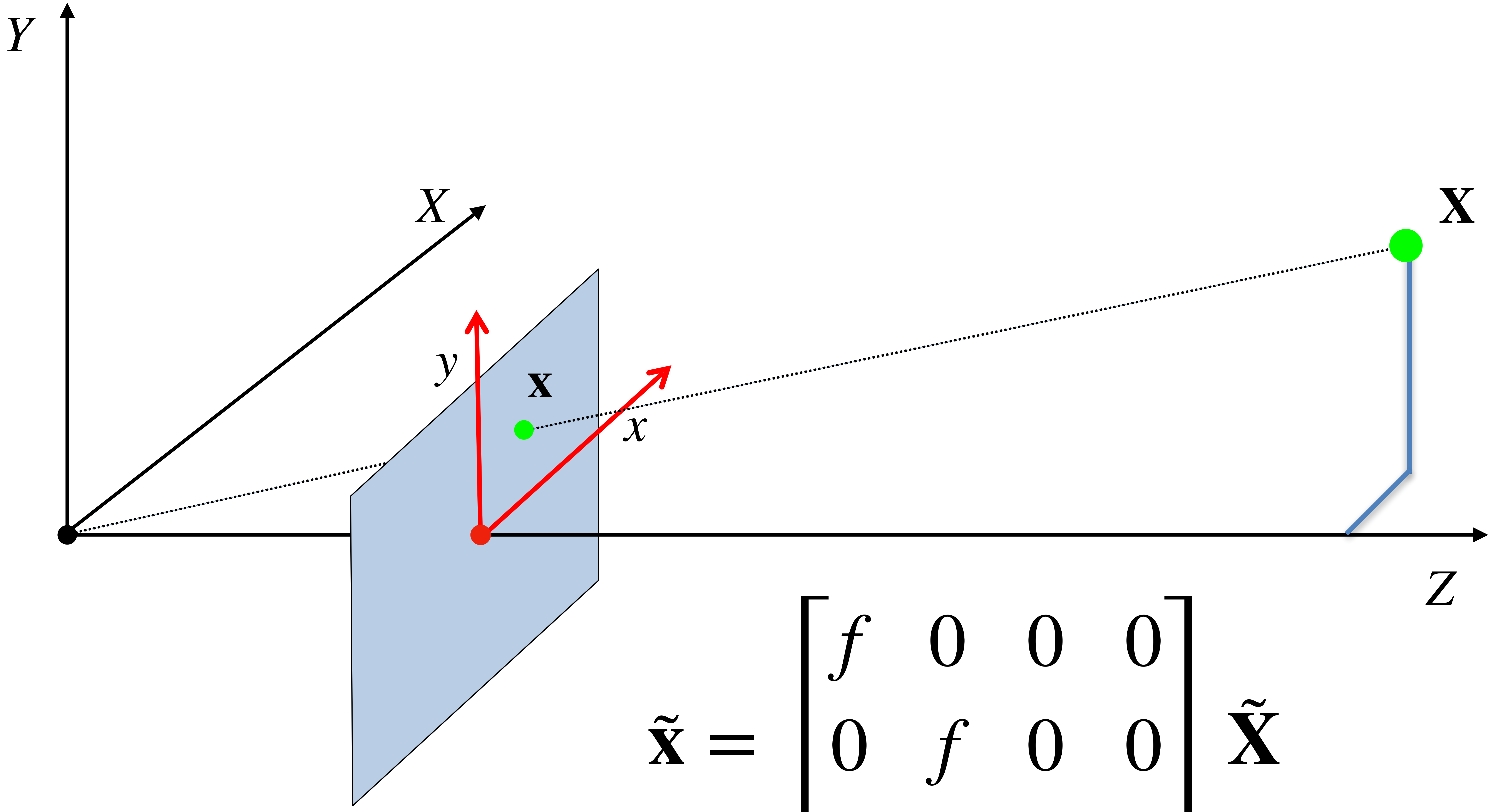
$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix} \Rightarrow \mathbf{x} = \left(f\frac{X}{Z}, \quad f\frac{Y}{Z} \right)$$

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

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



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

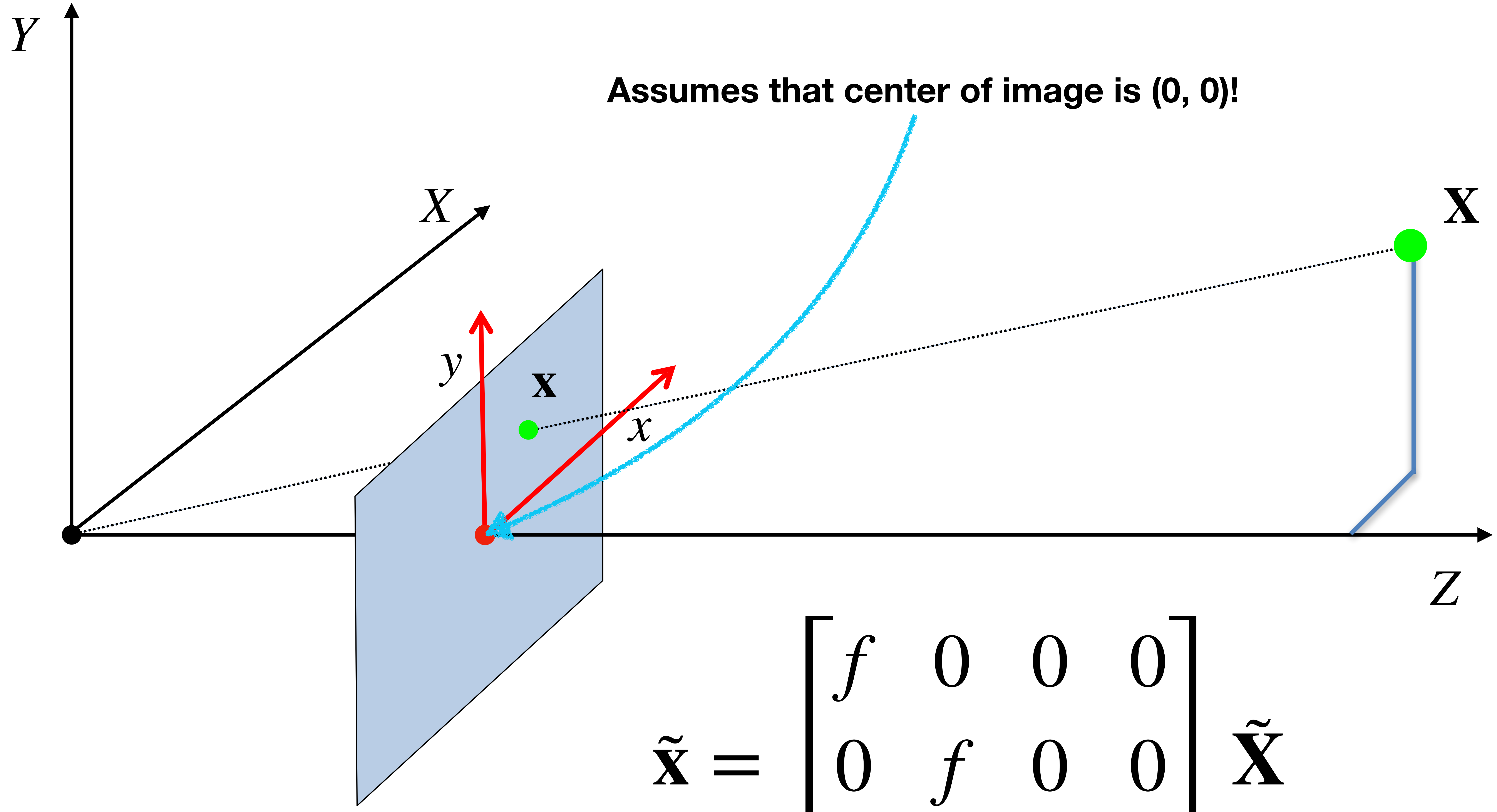
Image / Pixel coords.

World coords.

Projection Matrix

Questions?

Perspective projection



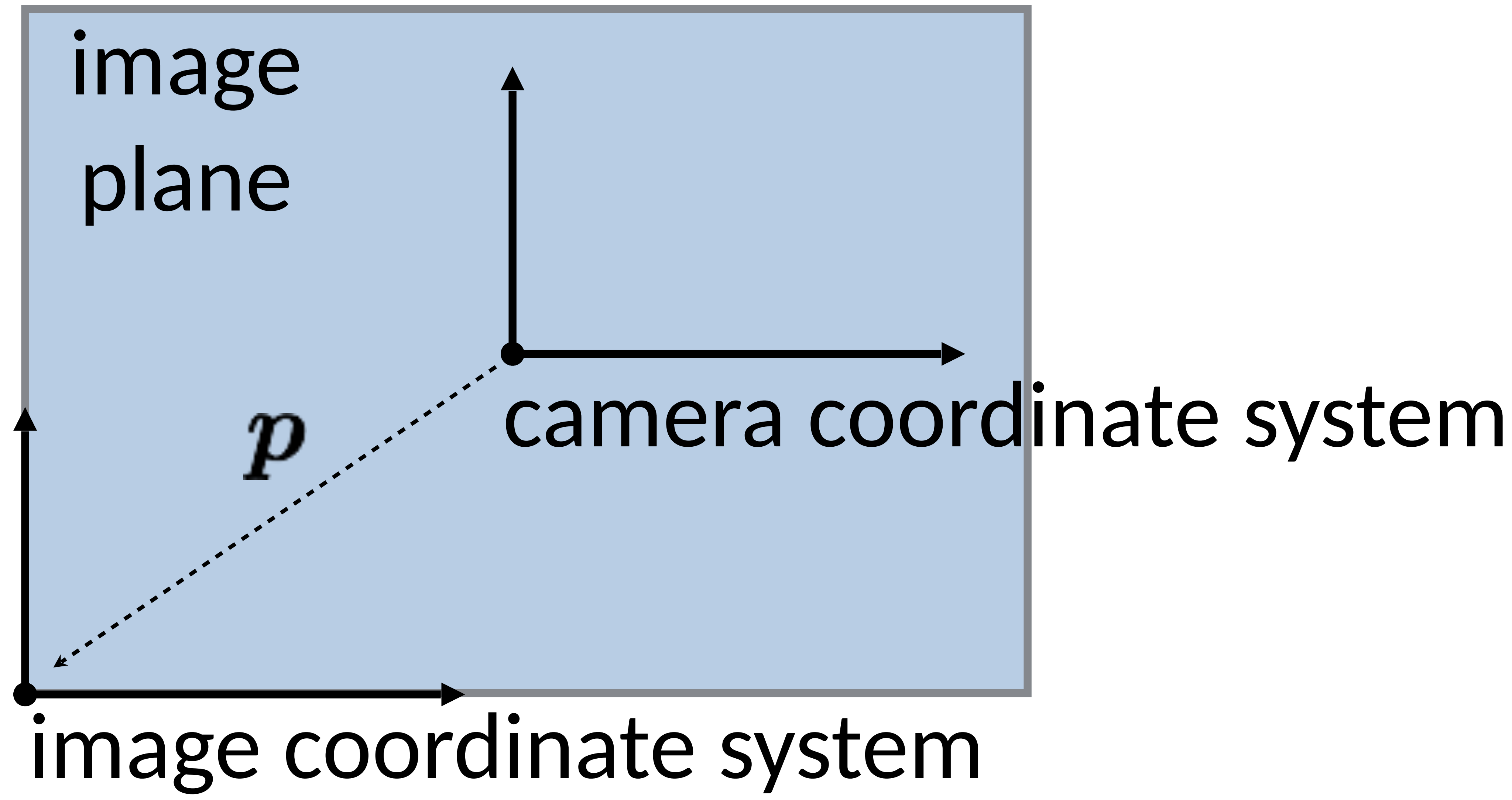
$$\tilde{\mathbf{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}$$

Image / Pixel coords.

Projection Matrix

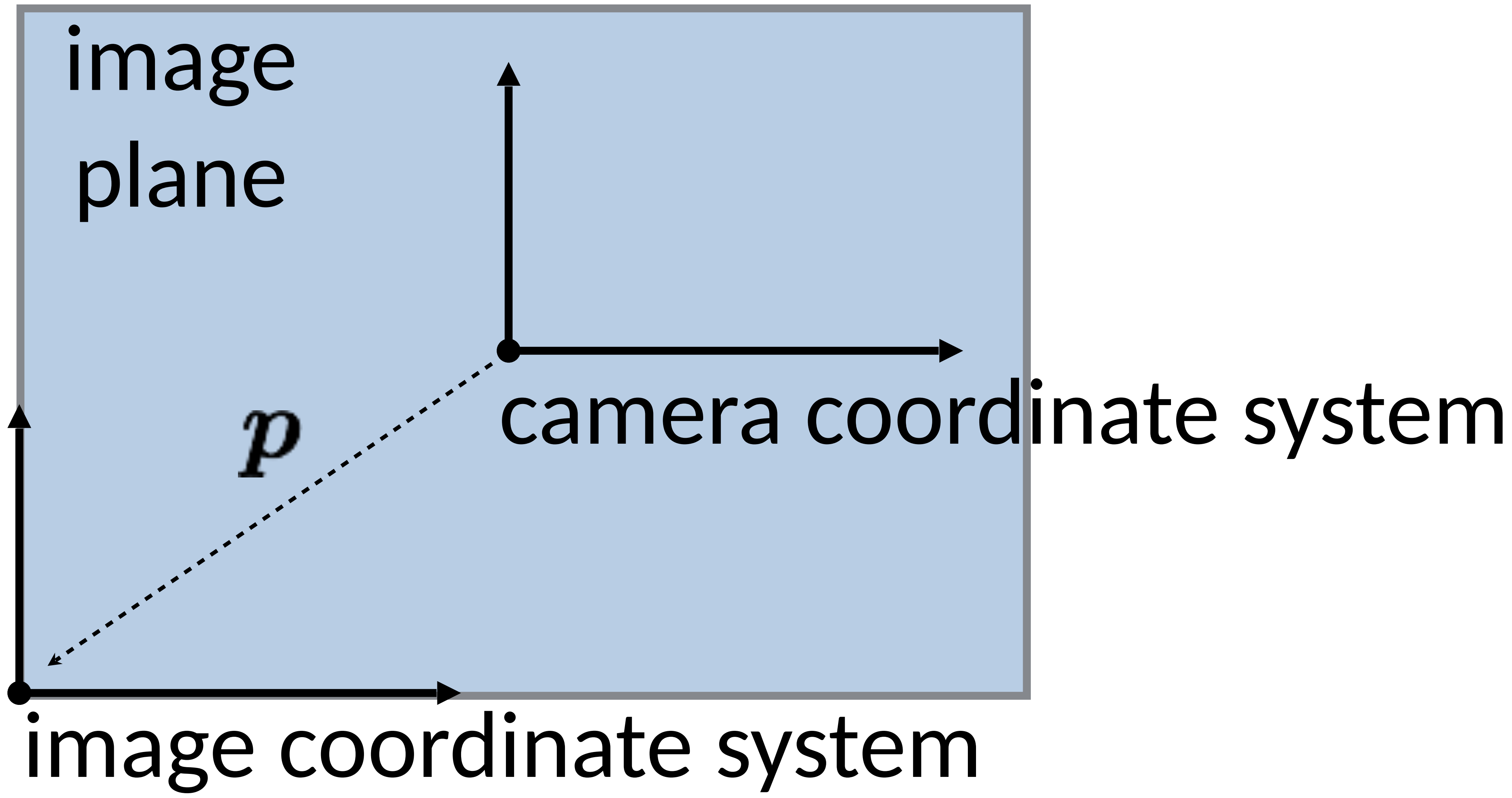
World coords.

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

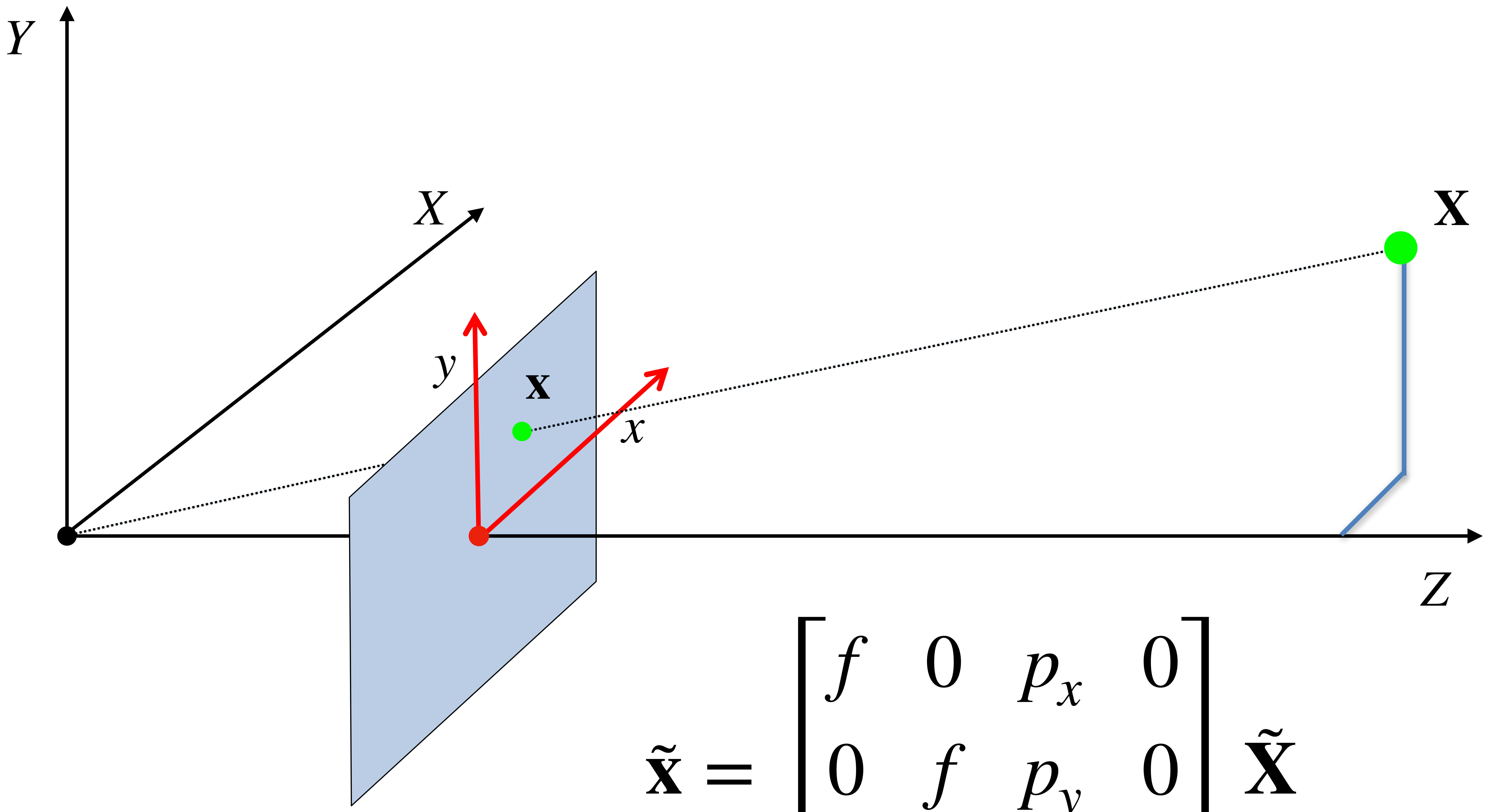
(homogeneous) perspective projection from 3D to 2D, assuming image plane at z = 1 and shared camera/image origin

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

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



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

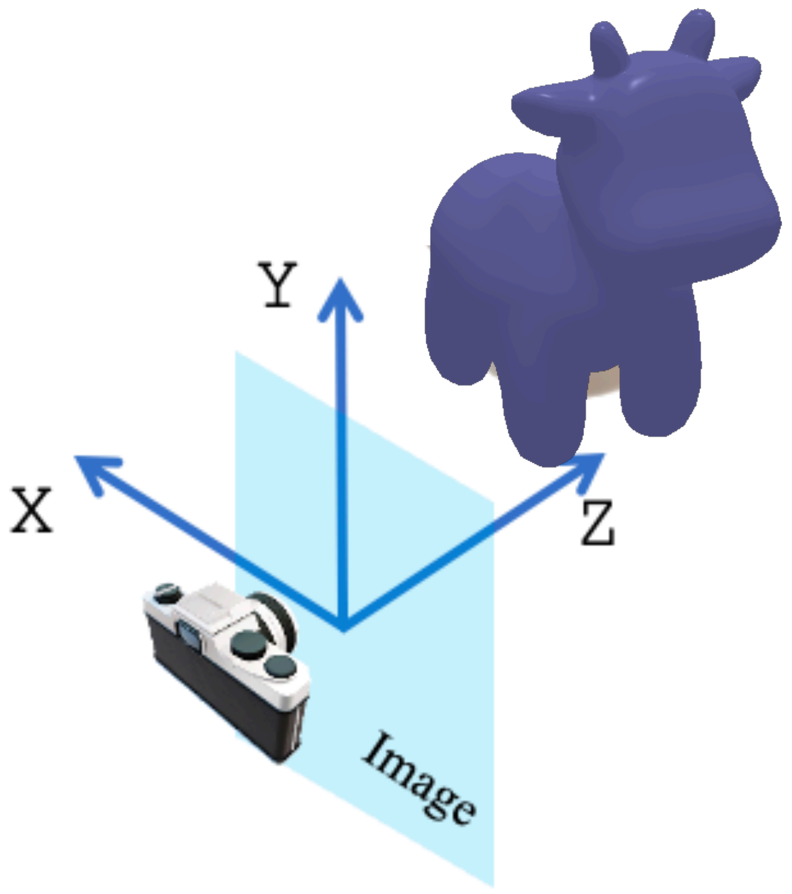
Image / Pixel coords.

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

$K_{cam \to img}$ →

(0, 0)

(1, 1)

Normalized Image Frame (here OpenCV convention)

$K_{img \to pix}$ →

(0,0)

(255,255)

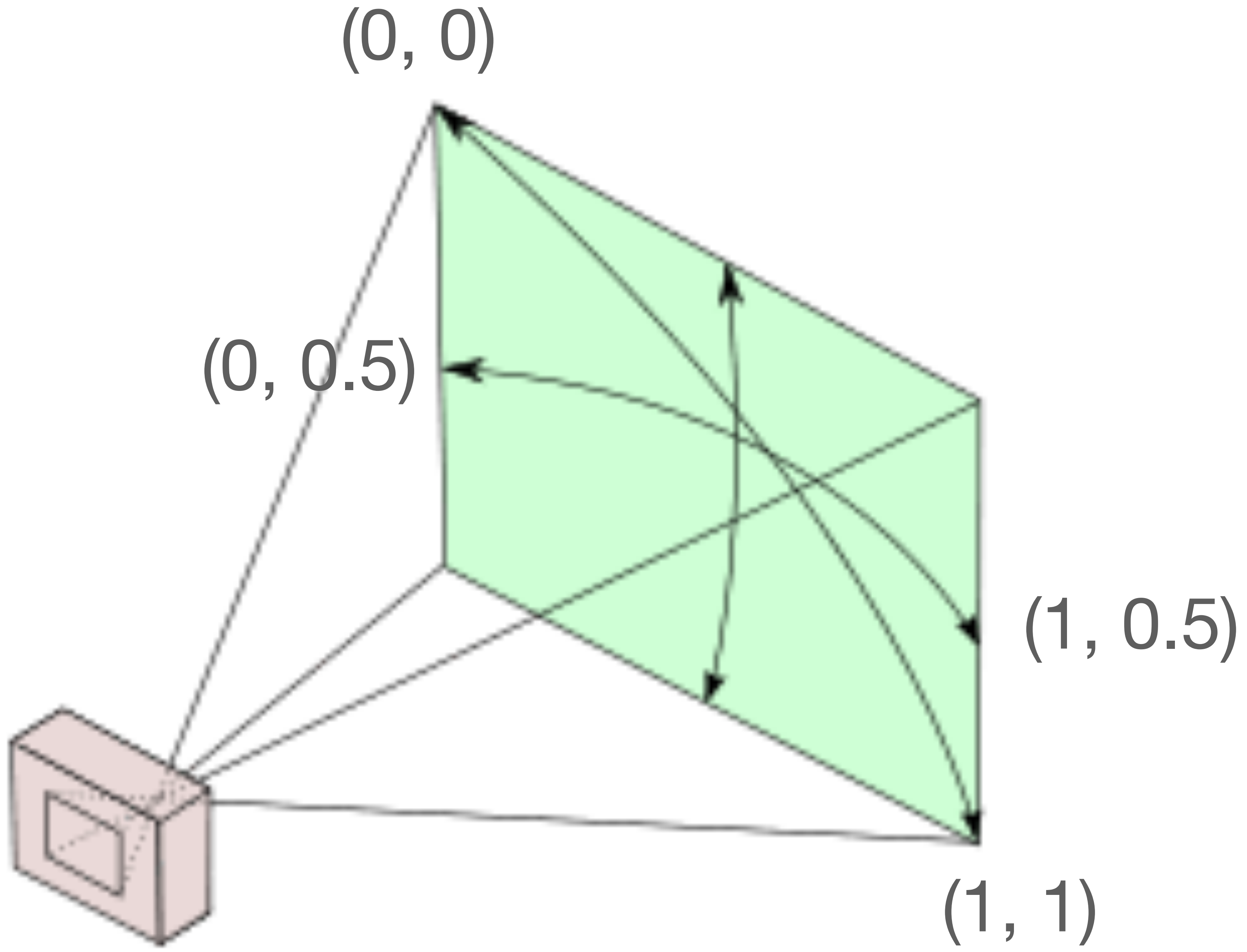
$\mathbb{P}^2$

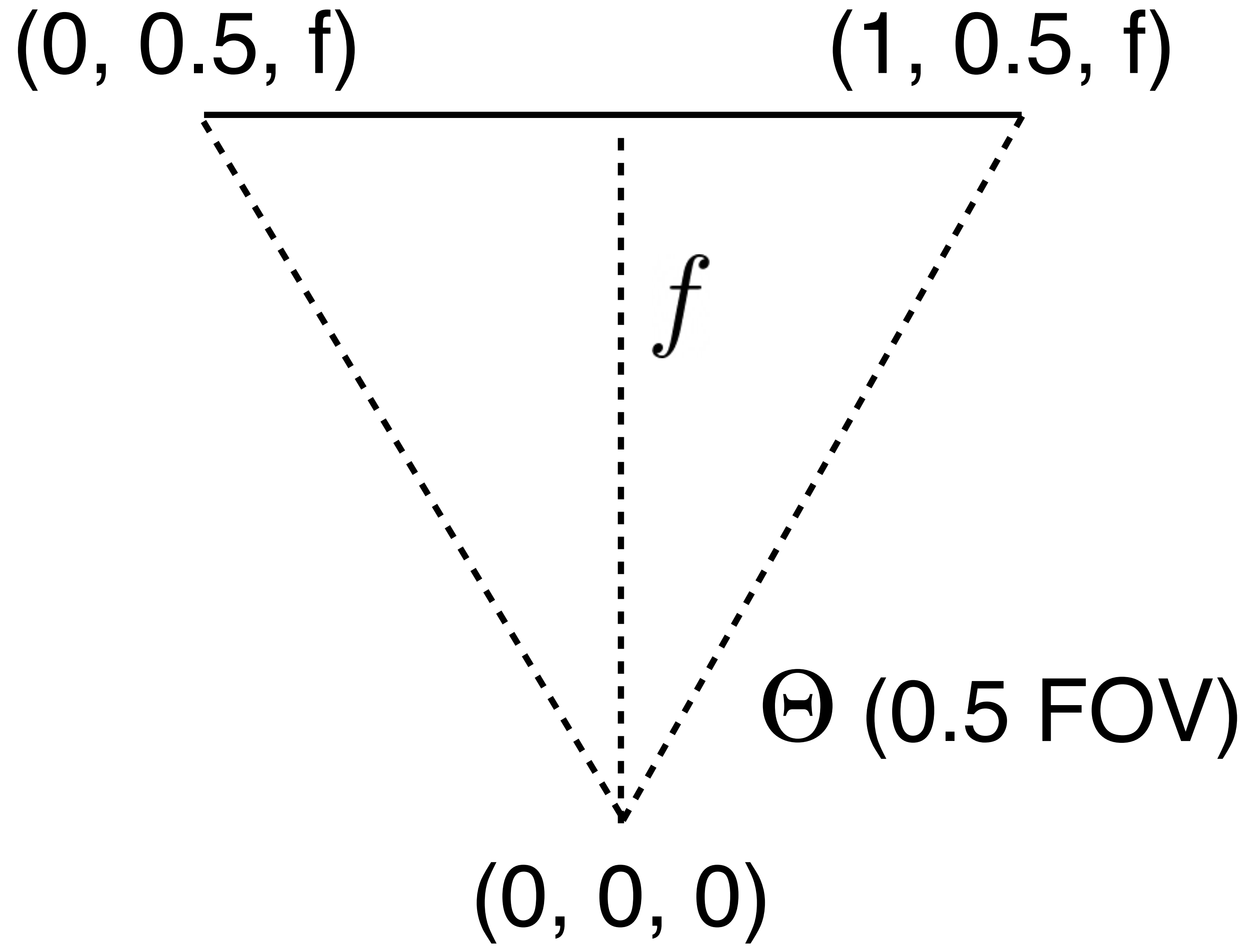
Slide credit: CMU 16-889: Learning for 3D Vision, Prof. Shubham Tulsiani

Exercise: Focal Length as Function of FOV

$$K_{cam \rightarrow img}$$

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





Exercise: Cropping an Image

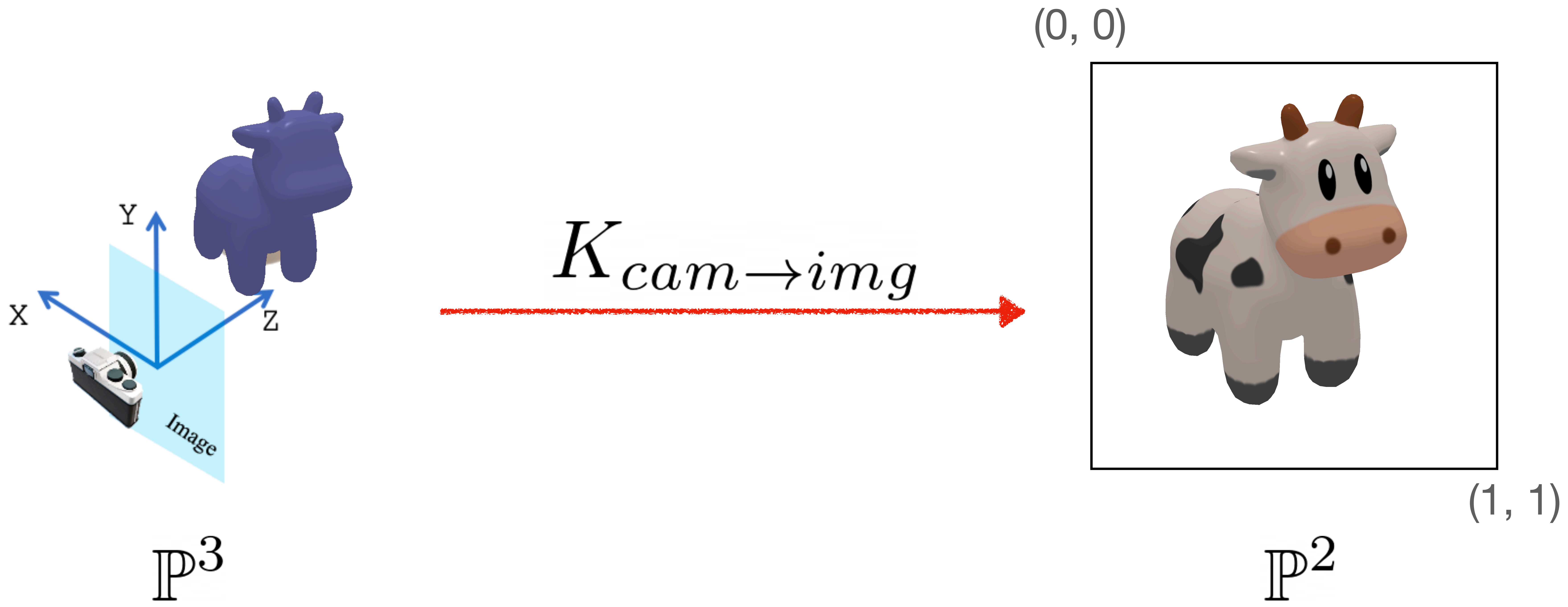
$K_{cam \rightarrow img}$

$K'_{cam \rightarrow img}$

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

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

In practice: Decoupling Projection from Image Size



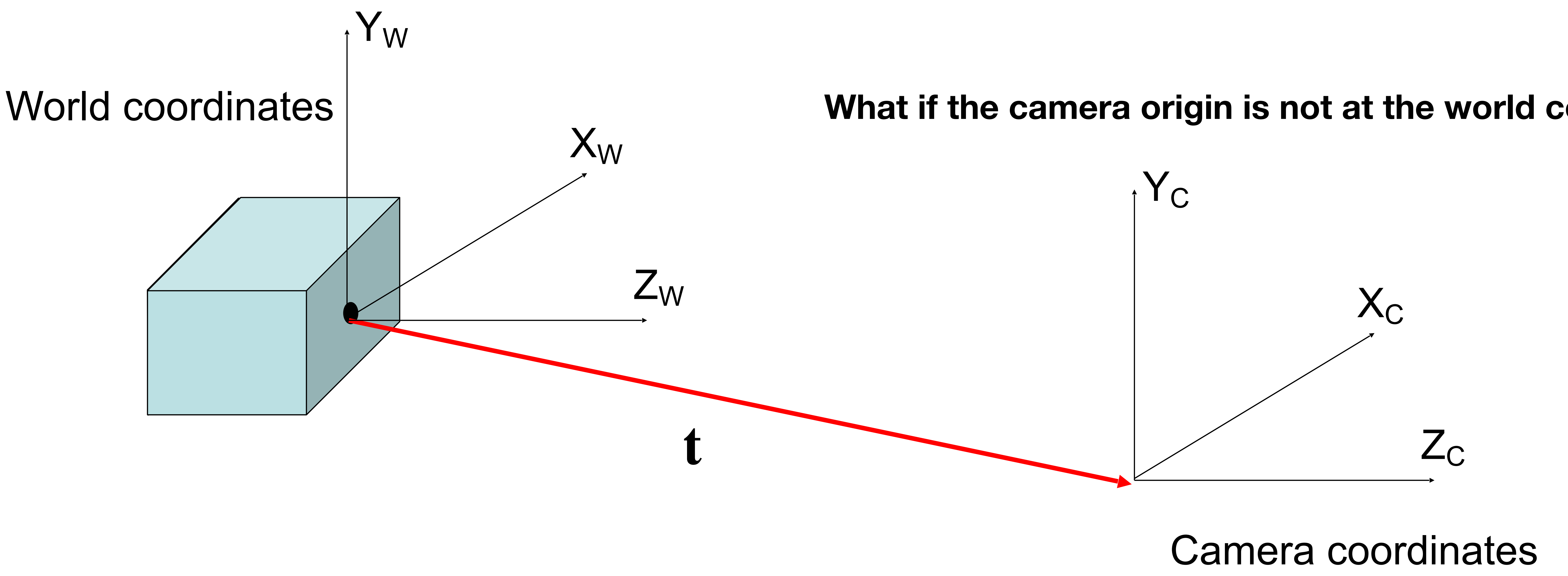
Helps (conceptually and in implementation) to reason in normalized image coordinates

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

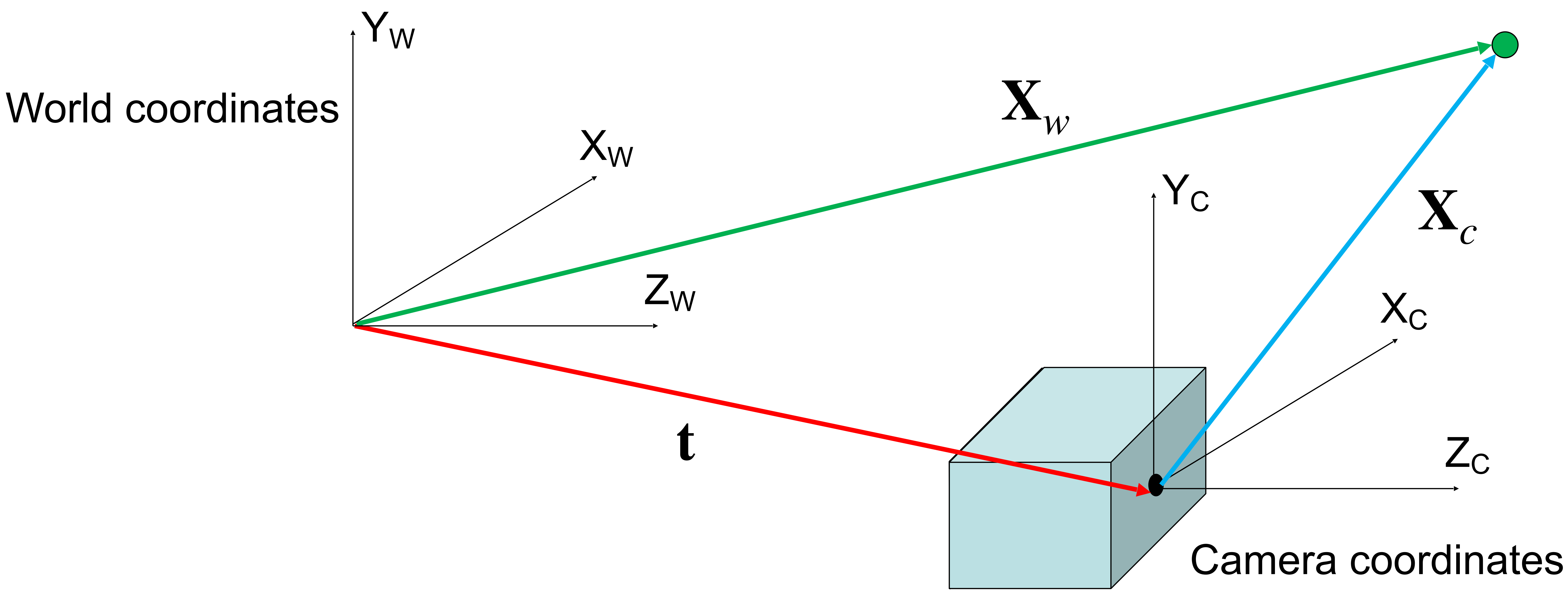
Questions?

Camera parameters

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



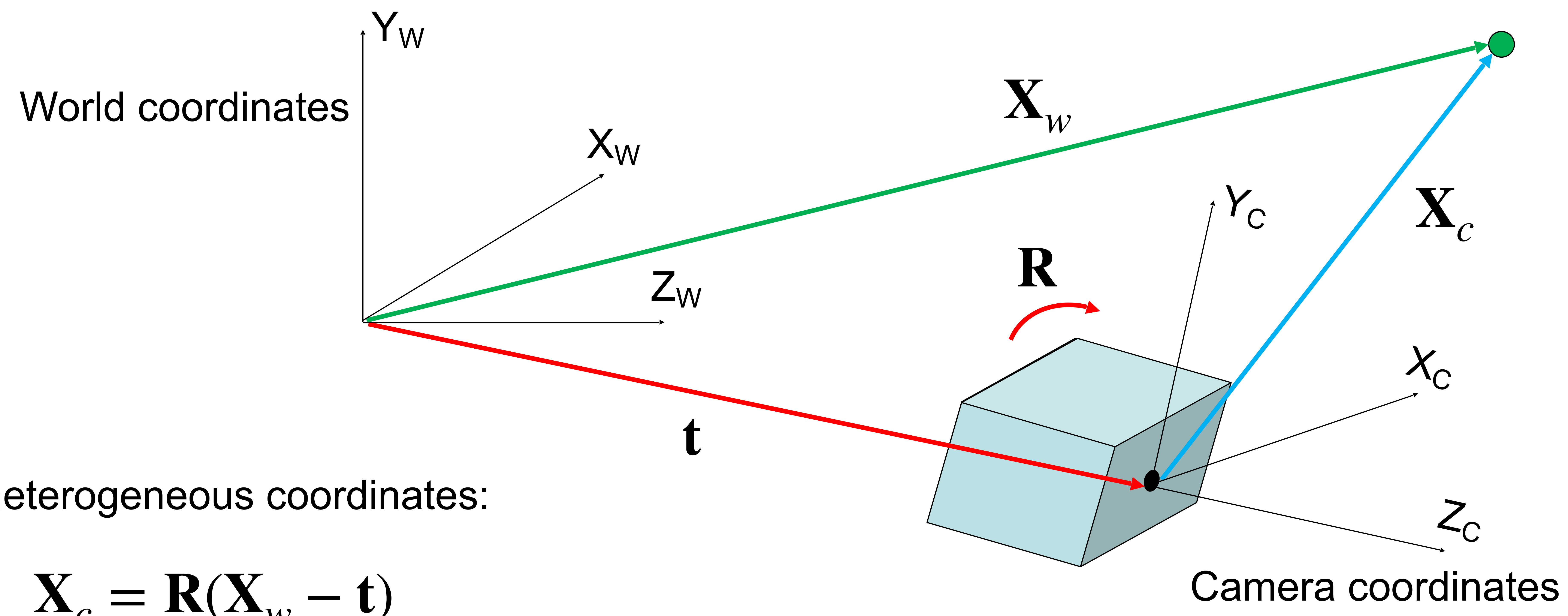
Camera parameters



In heterogeneous coordinates:

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

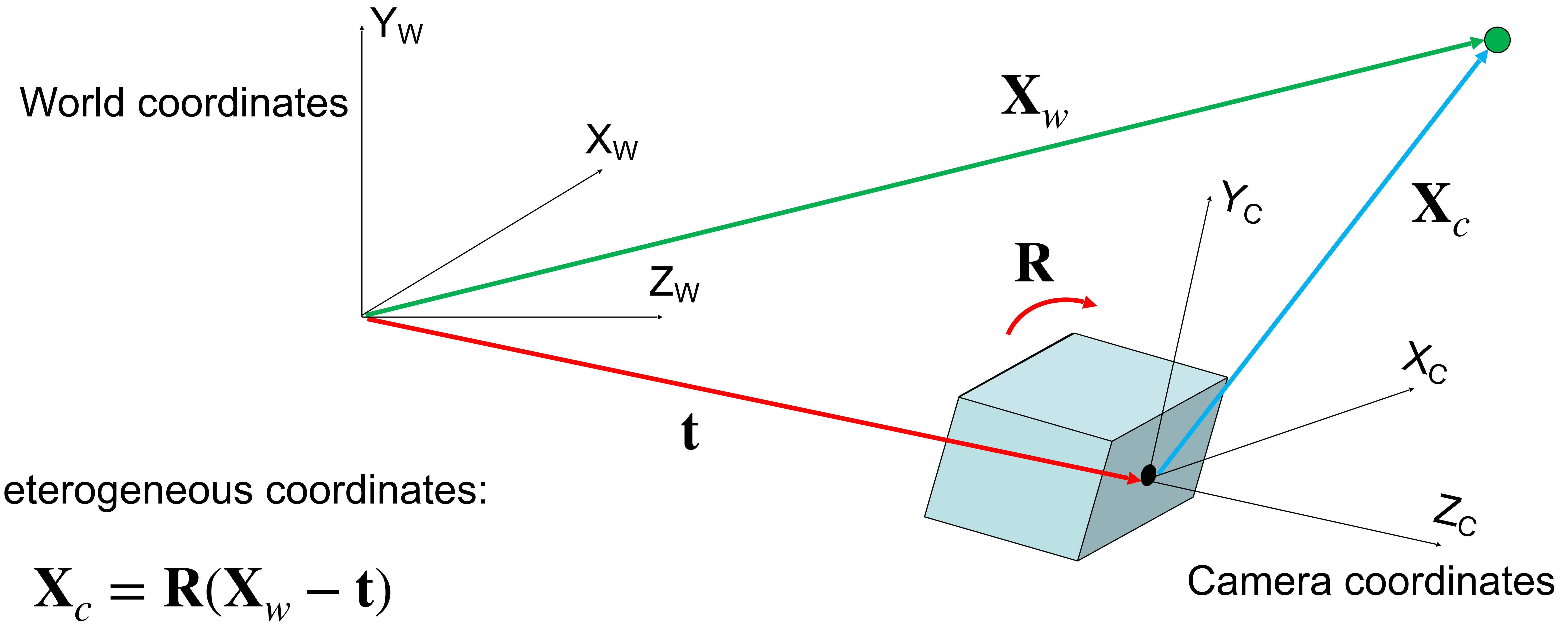
Camera parameters



In heterogeneous coordinates:

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

Camera parameters



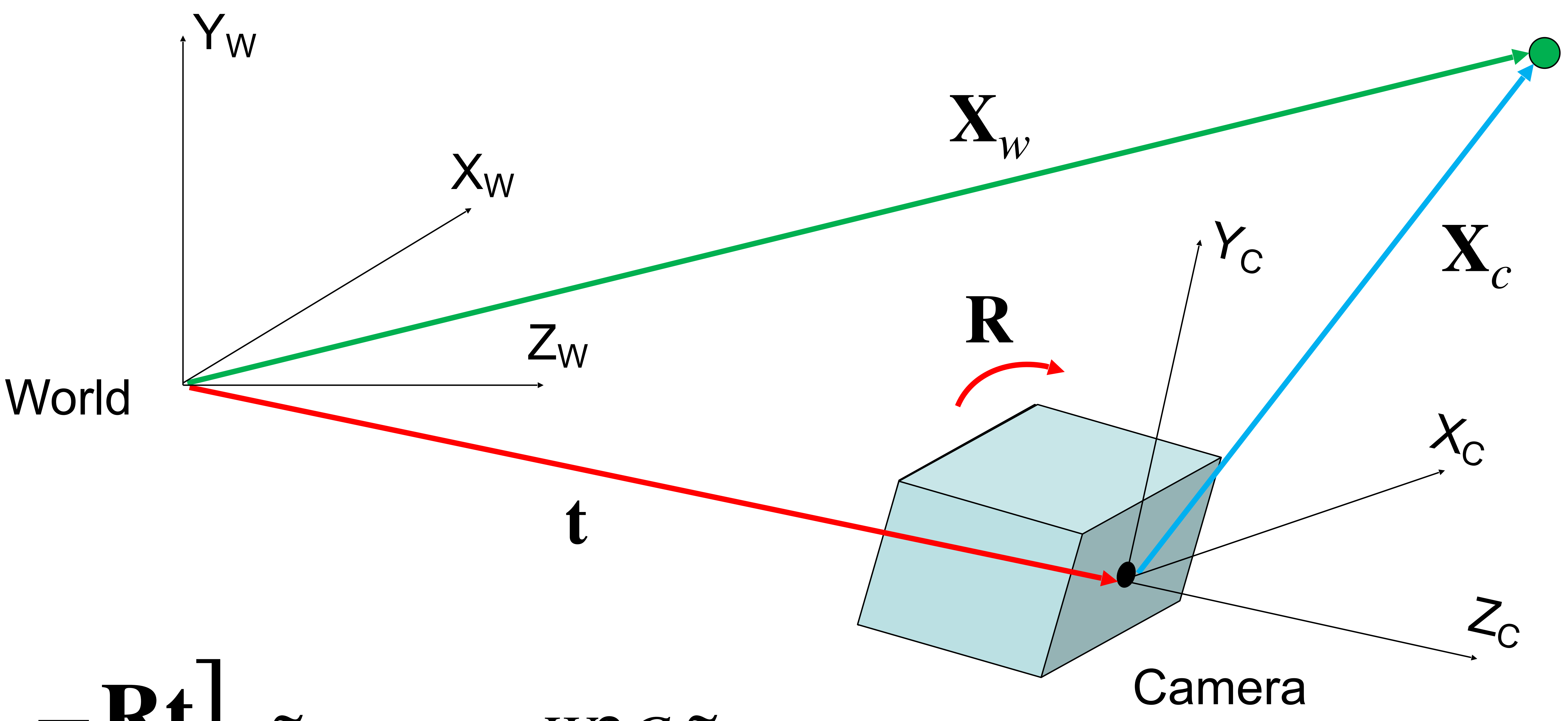
In heterogeneous coordinates:

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

In homogeneous coordinates:

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

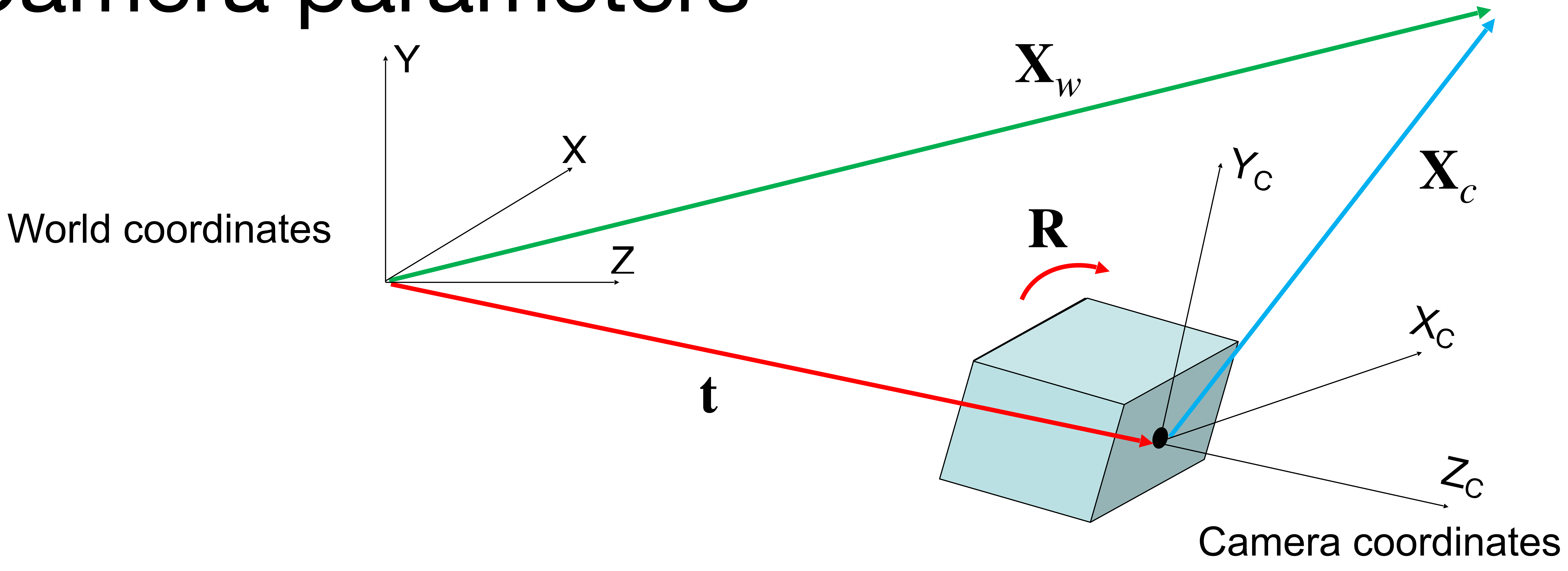
Cam2World vs. World2Cam extrinsic parameters



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

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

Camera parameters



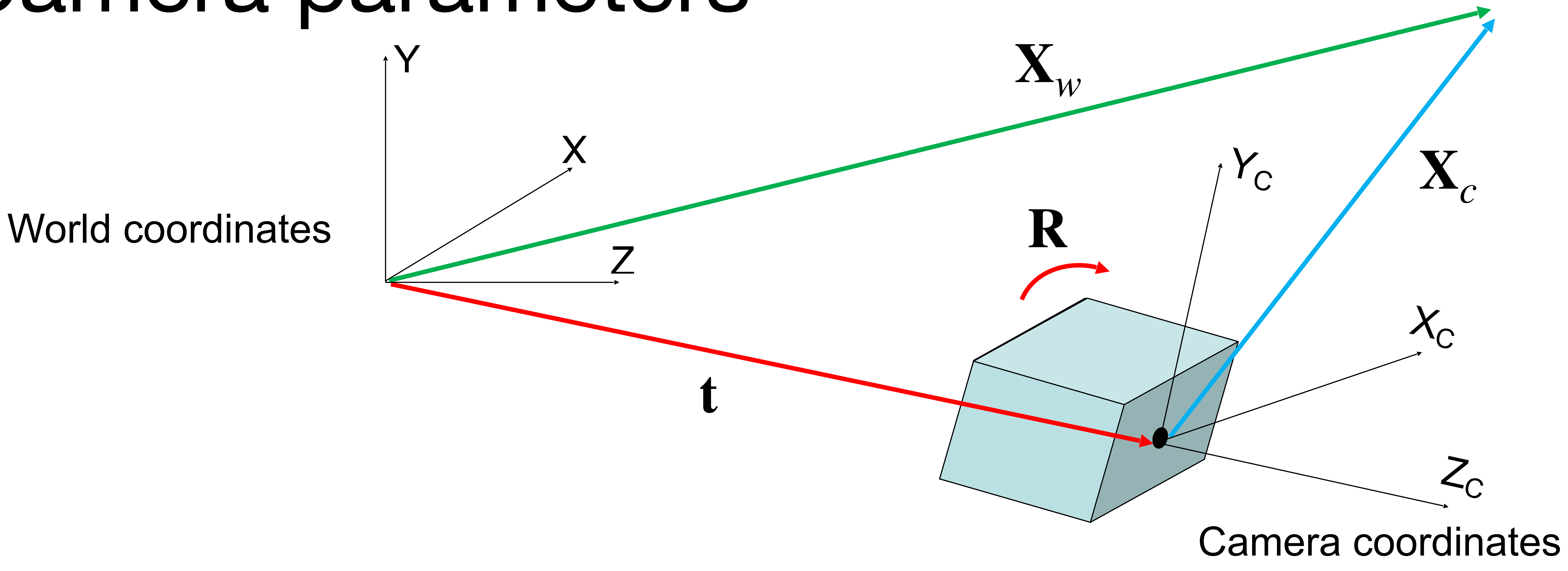
World coordinates to camera coordinates

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

Camera coordinates to image coordinates

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

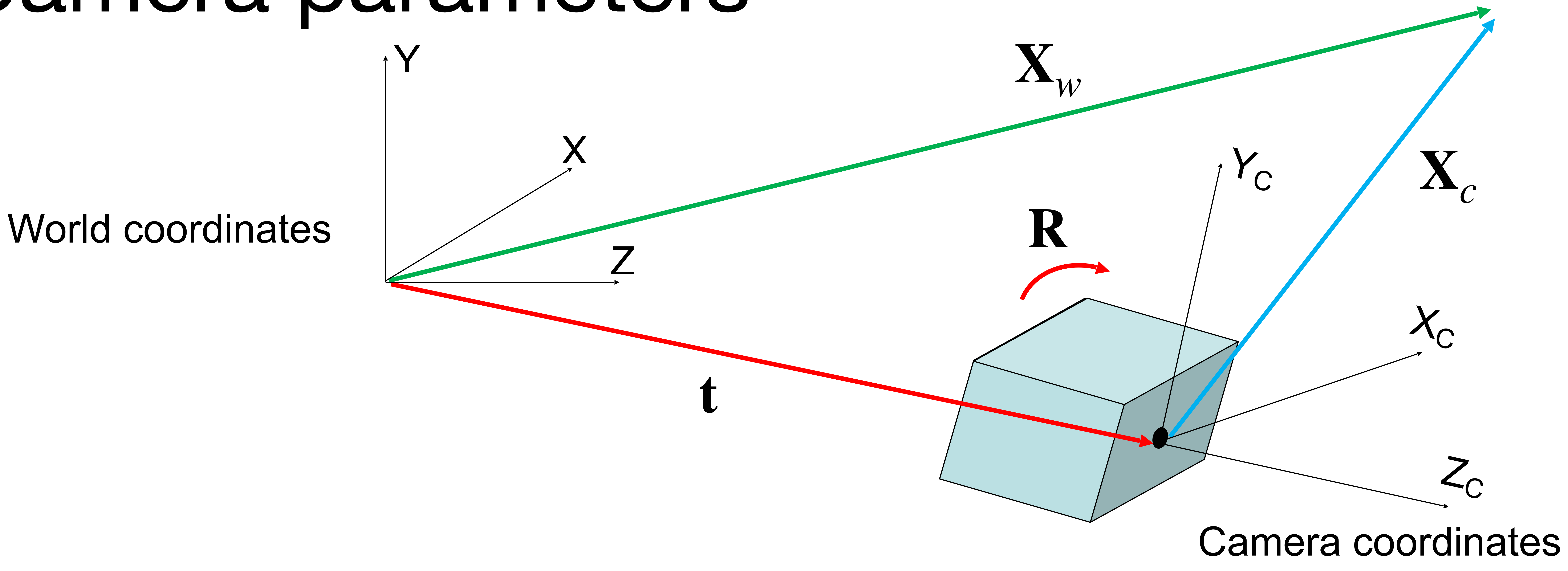
Camera parameters



World coordinates to image coordinates

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

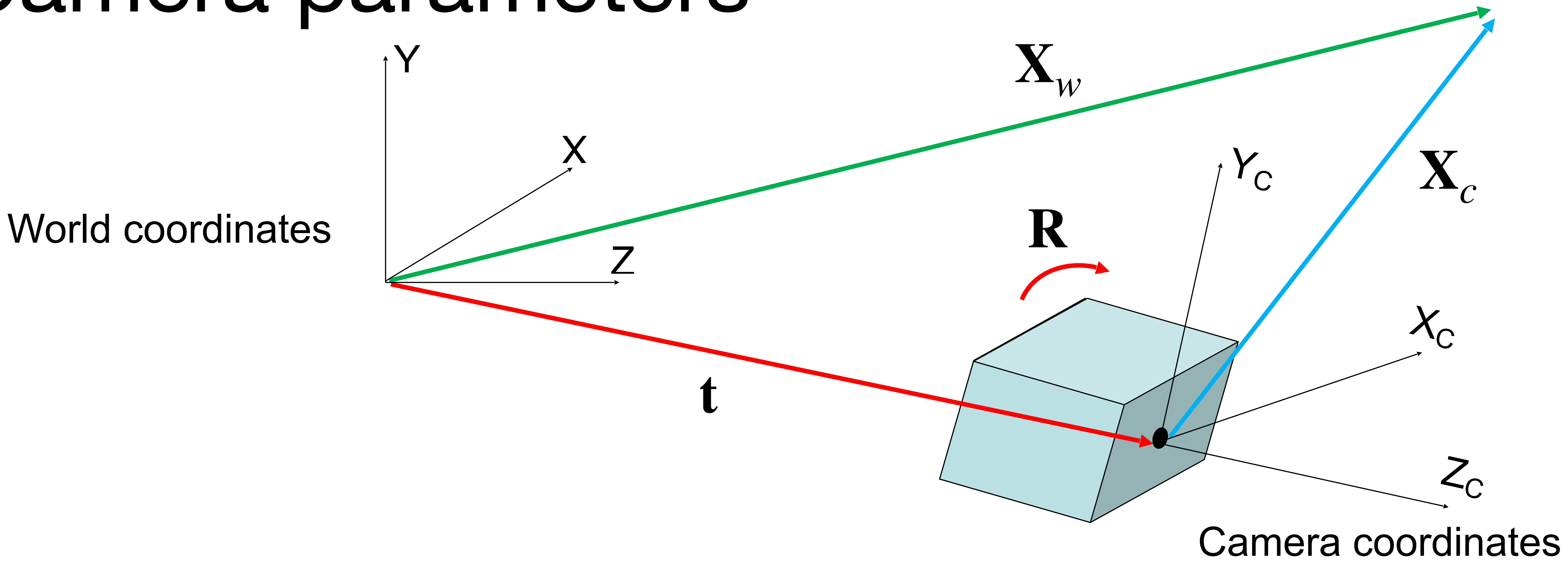
Camera parameters



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



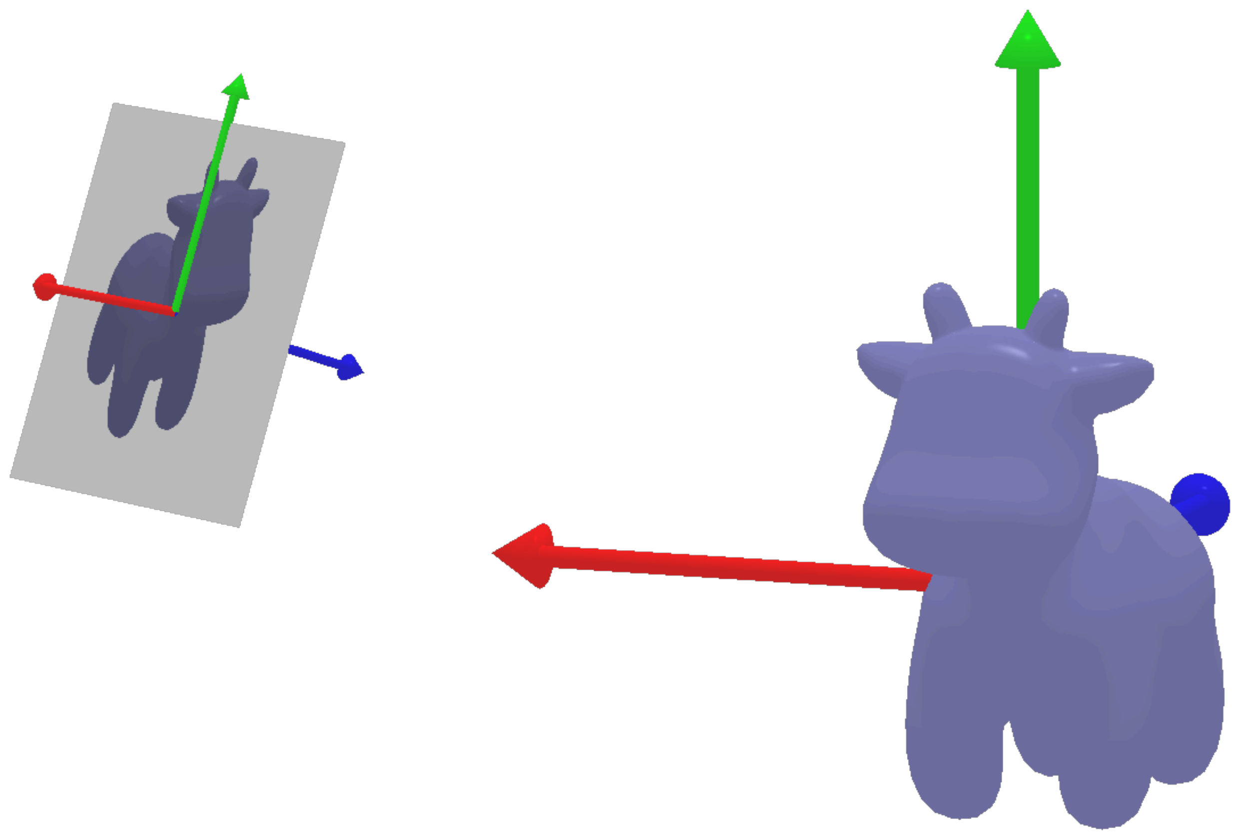
World coordinates to image coordinates

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

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)?

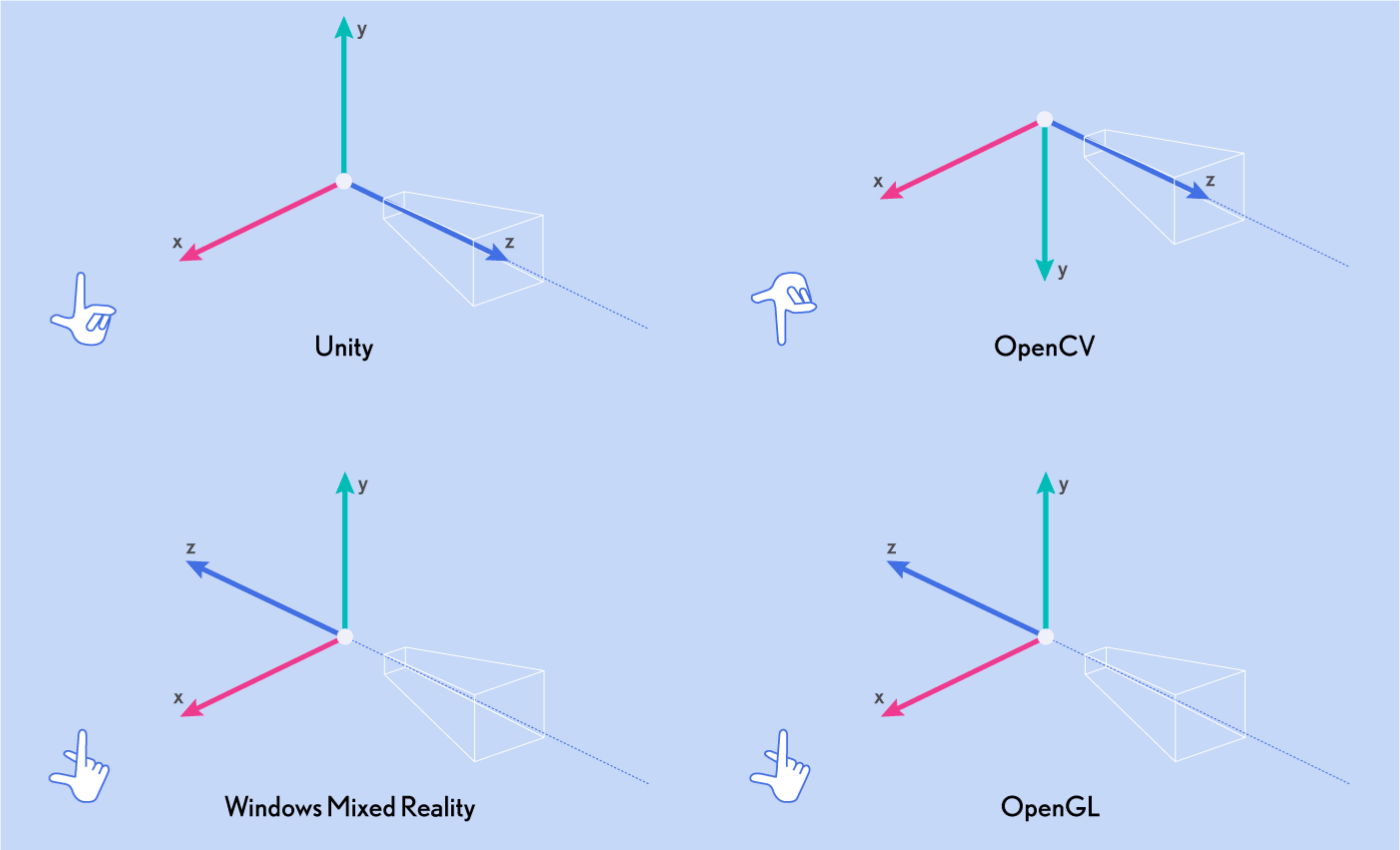
$$R = \mathbf{I}; \quad \mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

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)?

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

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

Beware of Conventions



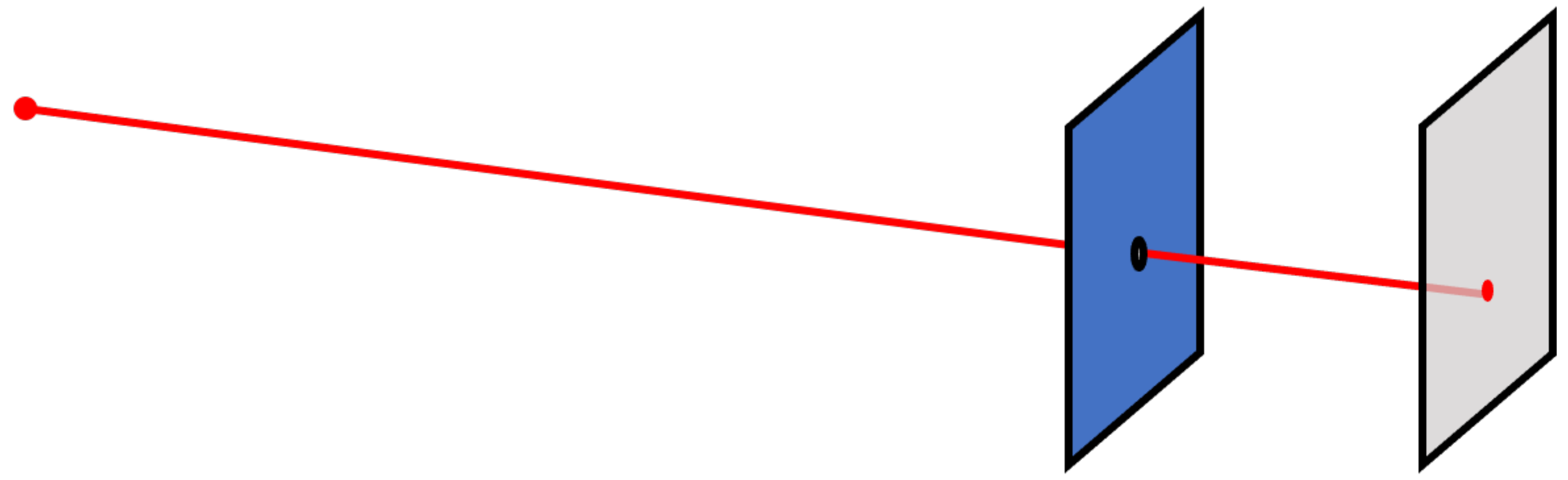
X=left or right? Y=up or down? etc..

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!)

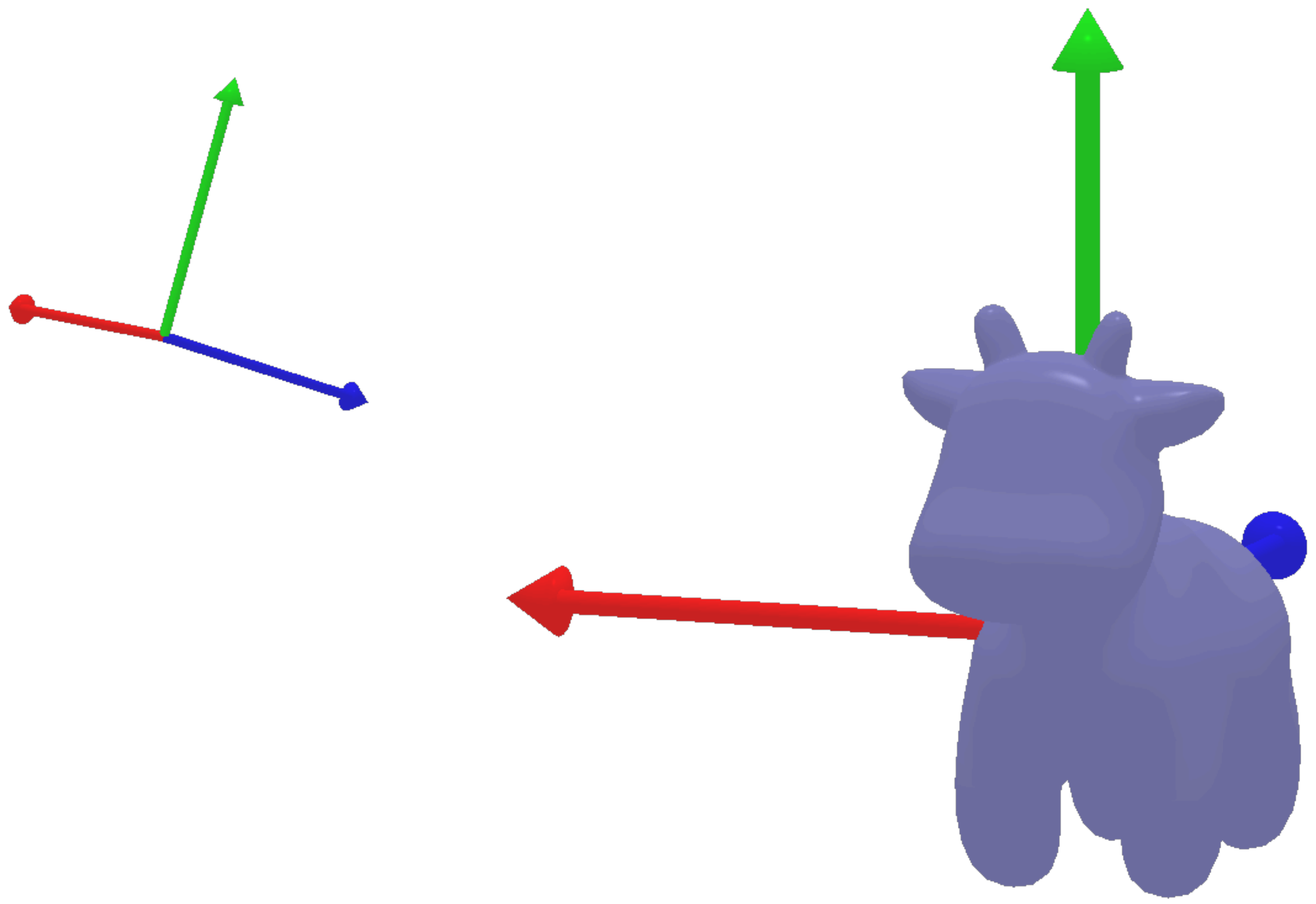
read https://pytorch3d.org/docs/cameras

Slide credit: Shubham Tulsiani. Image courtesy: Christoph Krautz

Summary



Projection — associating rays to points in a plane



Camera Transformation — from world to camera coordinates

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

Questions?

Vision systems

One camera

Two cameras

N cameras

Let’s consider two eyes

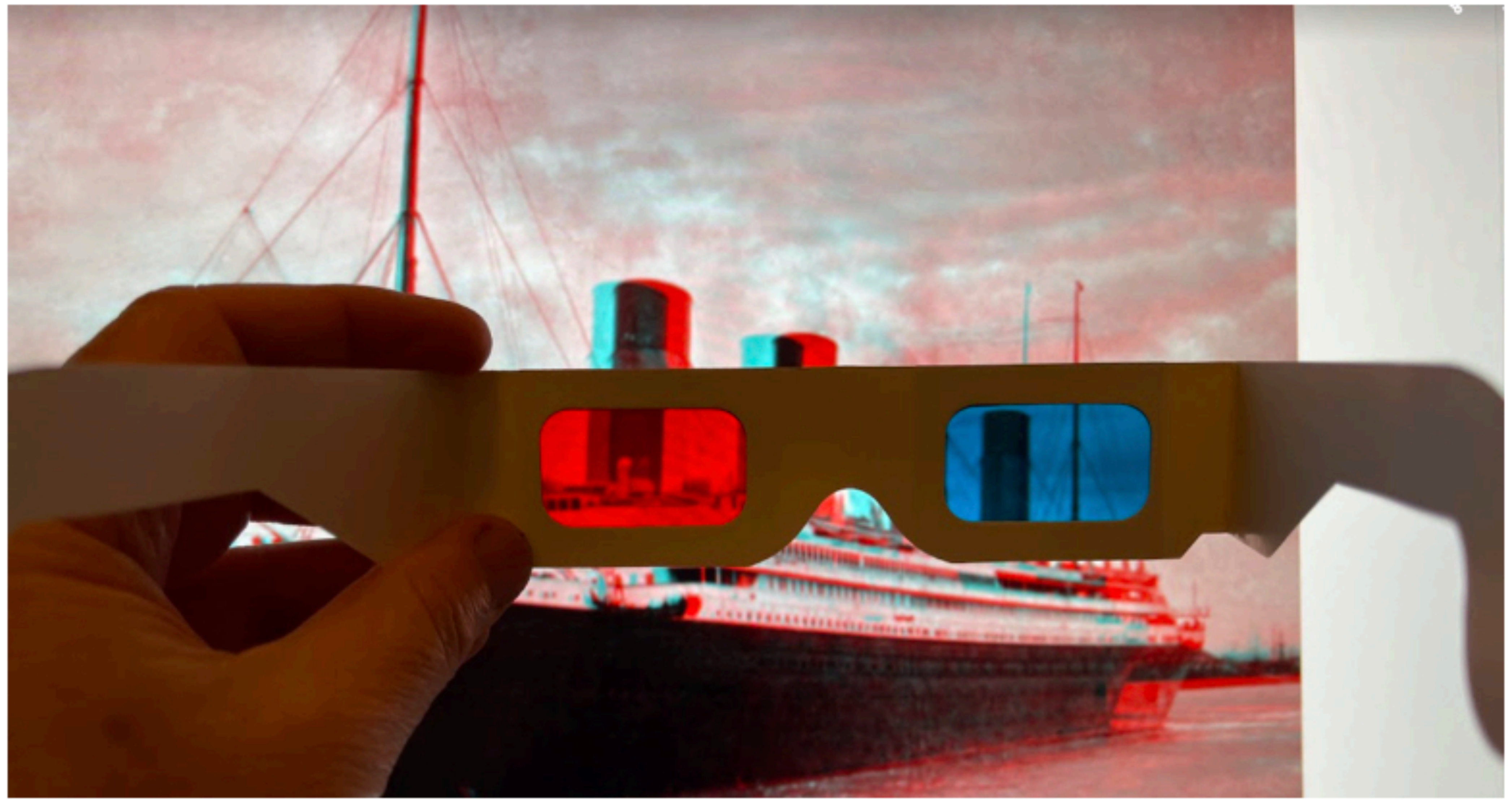
One camera

Two cameras

N cameras

Stereo images of the Titanic

(a)



(b)

Figure 1.1: (a) Stereo anaglyph of the ocean liner, the Titanic [McManus2022]. The red image shows the right eye's view, and cyan the left eye's view. When viewed through stereo red/cyan stereo glasses, as in (b), the cyan contrast appears in the left eye image and the red variations appear to the right eye, creating a the perception of 3d.

Stereoscope

View of Boston, c. 1860; an early stereoscopic card for viewing a scene from nature

More details

Soule, John P., 1827-1904 -- Photographer - This image is available from the New York Public Library's Digital Library under the digital ID G90F336_113F: digitalgallery.nypl.org → digitalcollections.nypl.org

File: Charles Street Mall, Boston Common, by Soule, John P., 1827-1904 3.jpg

Created: Coverage: 1860?-1890?. Source Imprint: 1860?-1890?. Digital item published 7-28-2005; updated 4-23-2009.

Brewster-type stereoscope, 1870

More details

Alessandro Nassiri - Museo della Scienza e della Tecnologia "Leonardo da Vinci"

Visore stereoscopico portatile di tipo Brewster, J. Fleury - Hermagis, 1870, con messa a fuoco manuale. Per la visione di lastre e stampe stereoscopiche 8,5x17cm. Museo nazionale della scienza e della tecnologia Leonardo da Vinci, Milano.

File: IGB 006055 Visore stereoscopico portatile Museo scienza e tecnologia Milano.jpg

Created: 1 July 2014

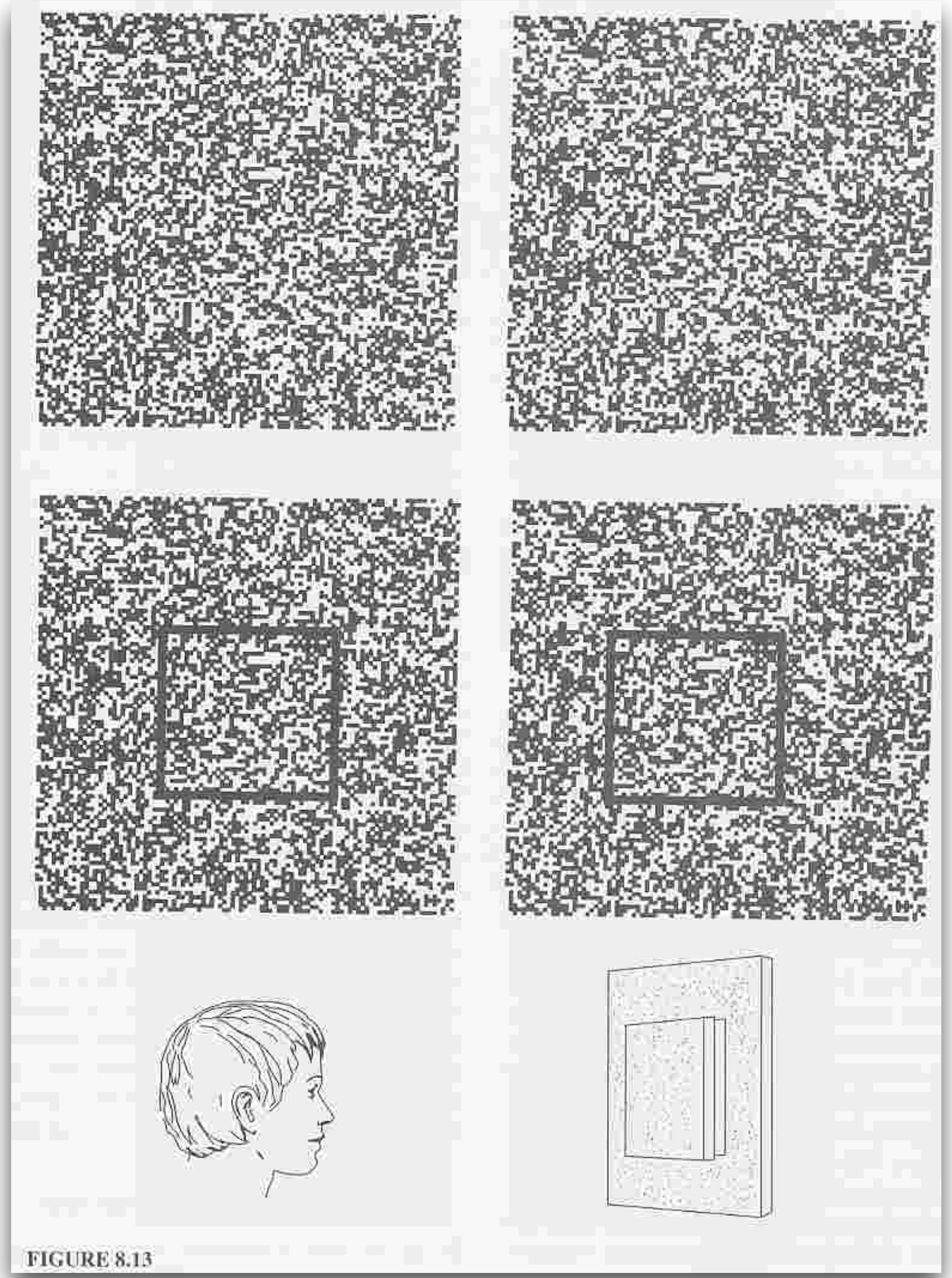
Depth without objects

Random dot stereograms (Bela Julesz)

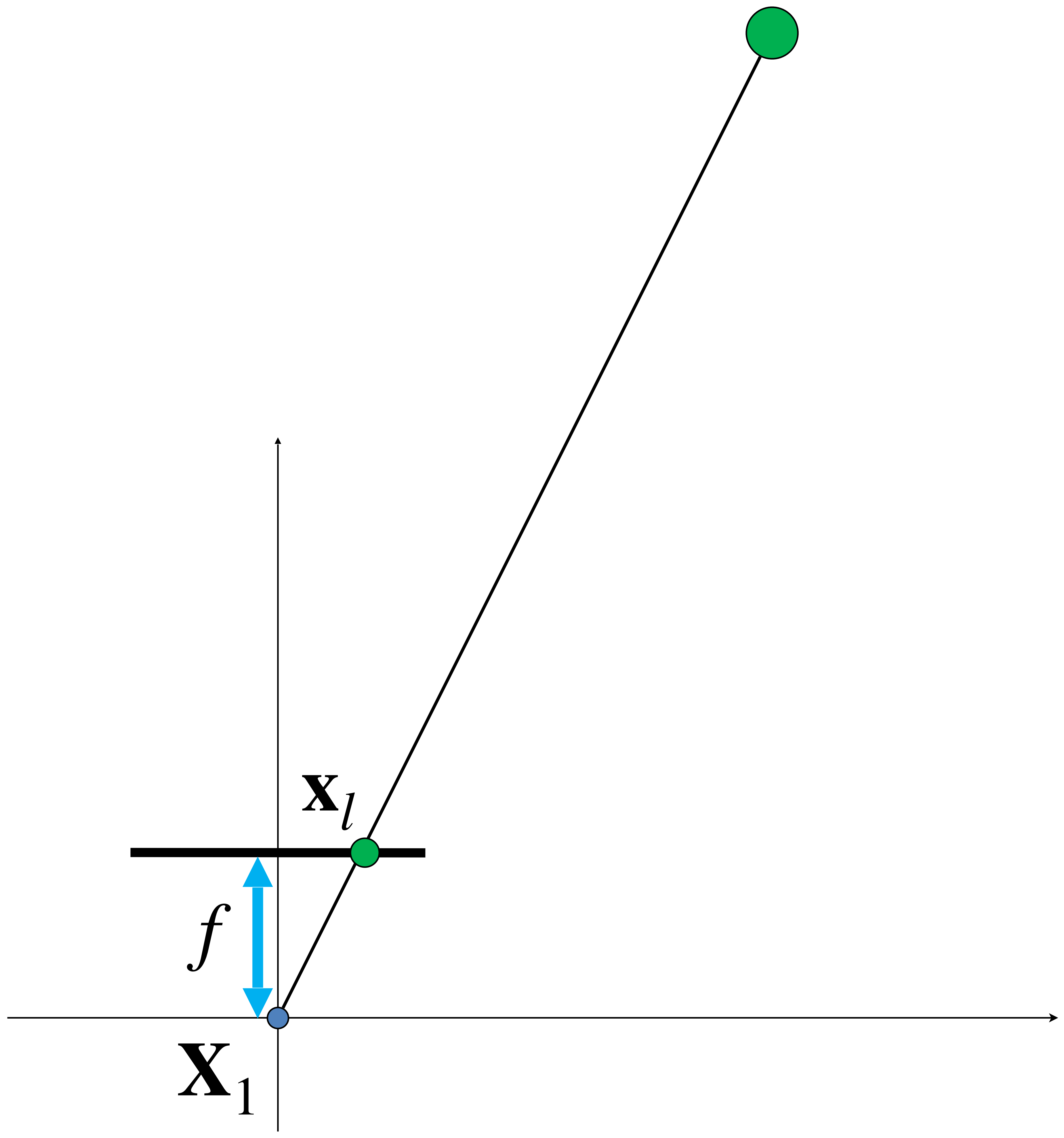
1	0	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0
0	1	0	Y	A	A	B	B	0	1
1	1	1	X	B	A	B	A	0	1
0	0	1	X	A	A	B	A	1	0
1	1	1	Y	B	B	A	B	0	1
1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0

1	0	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0
0	1	0	A	A	B	B	X	0	1
1	1	1	B	A	B	A	Y	0	1
0	0	1	A	A	B	A	Y	1	0
1	1	1	B	B	A	B	X	0	1
1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0

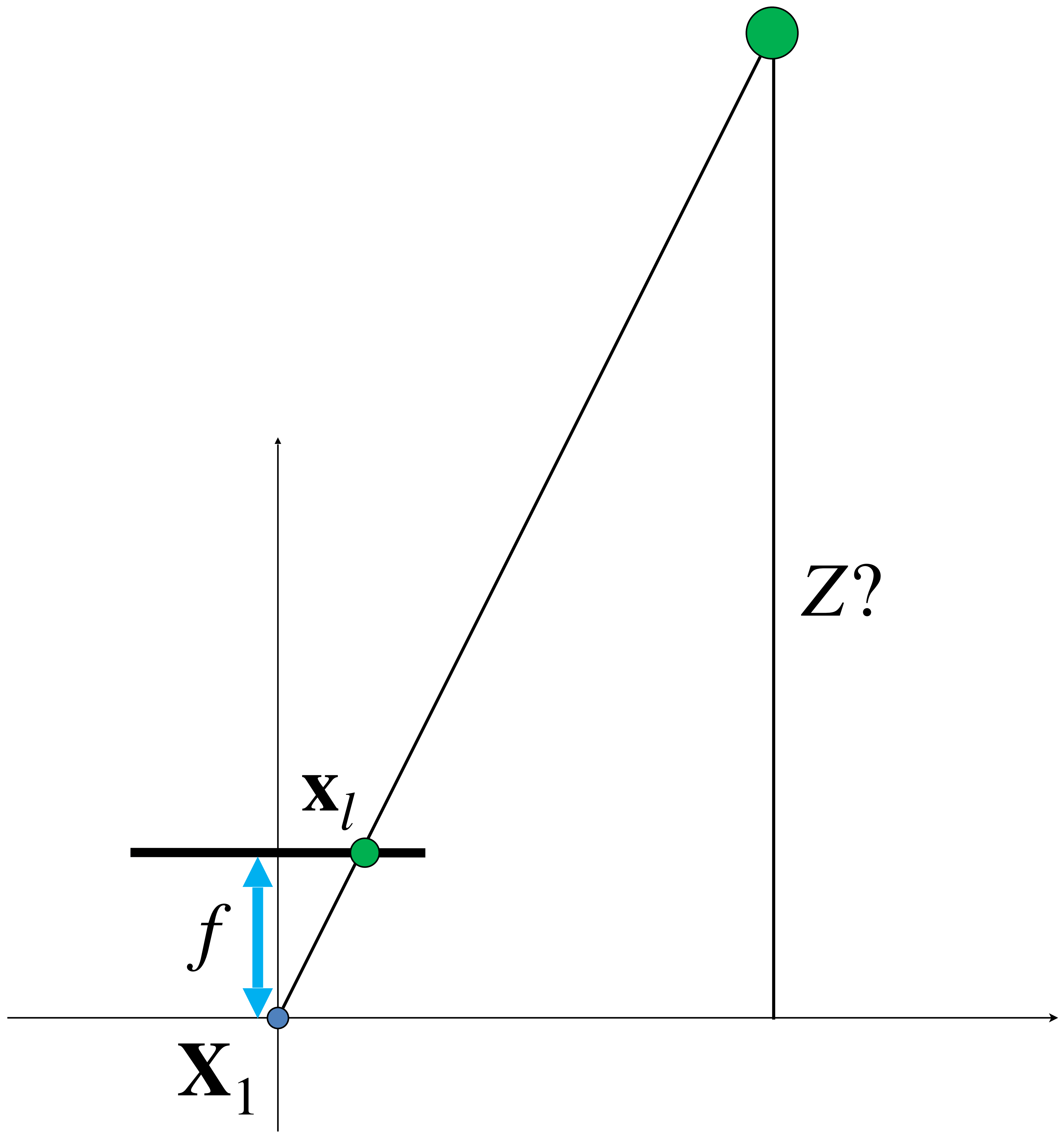
Julesz, 1971



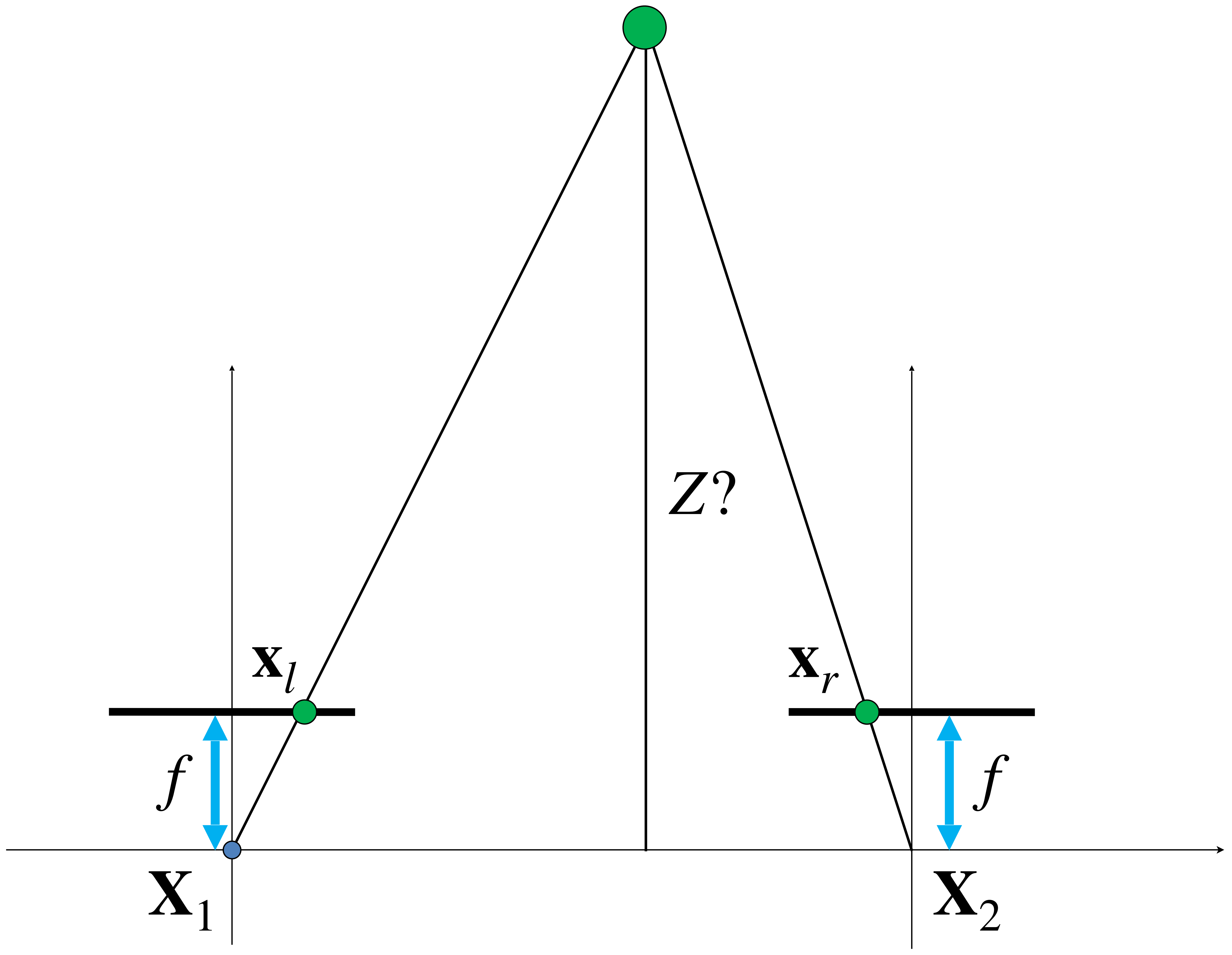
Geometry for a simple stereo system



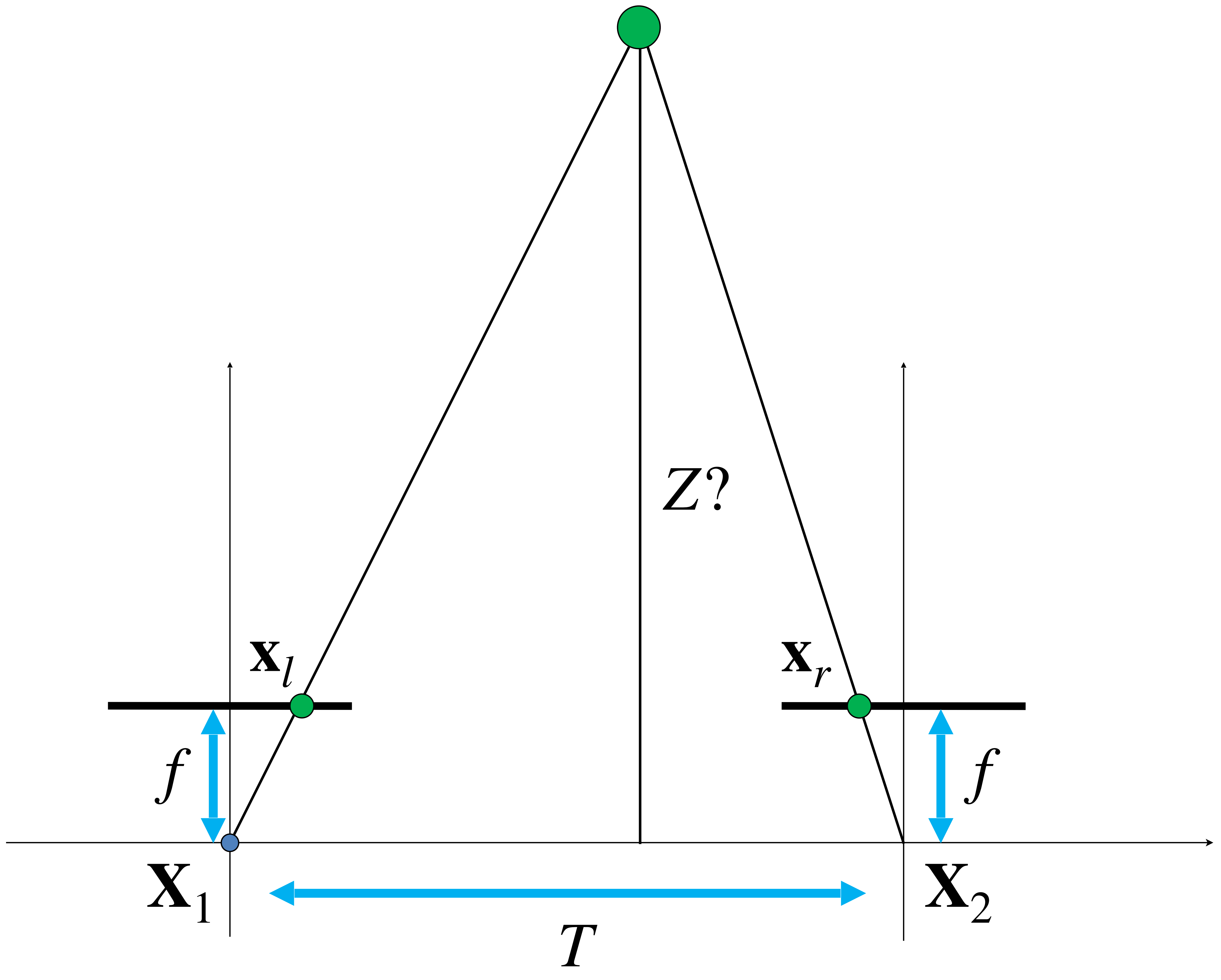
Geometry for a simple stereo system



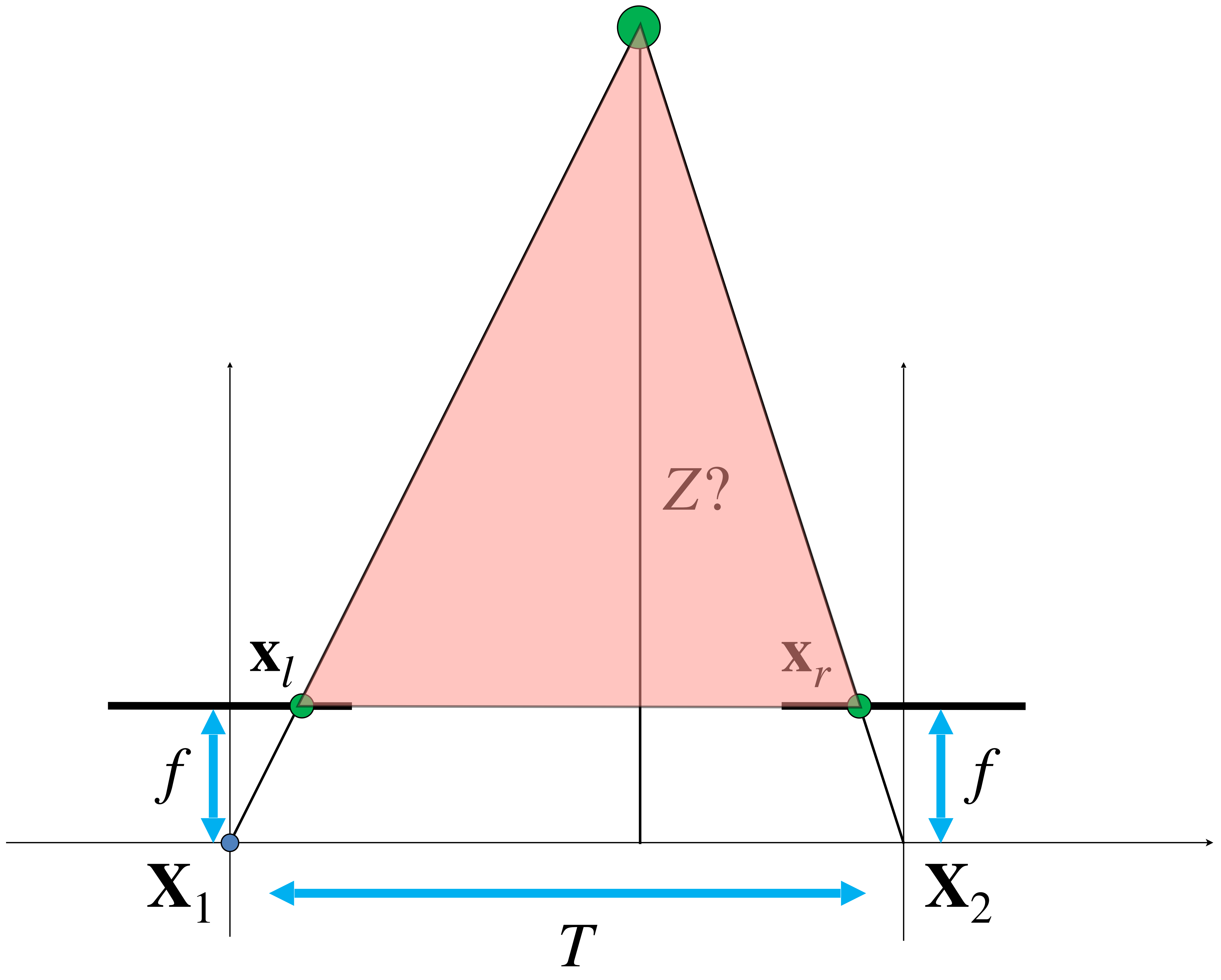
Geometry for a simple stereo system



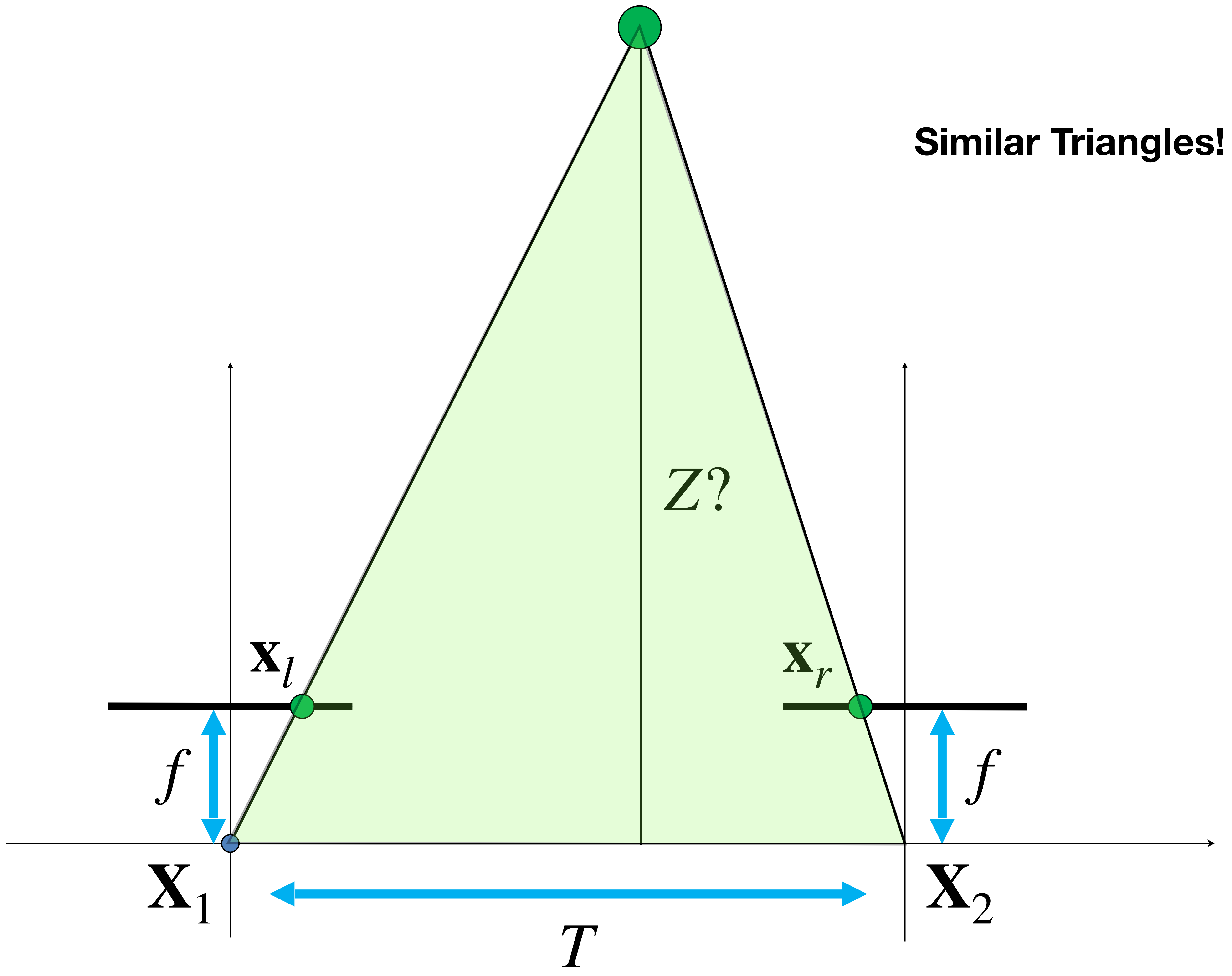
Geometry for a simple stereo system



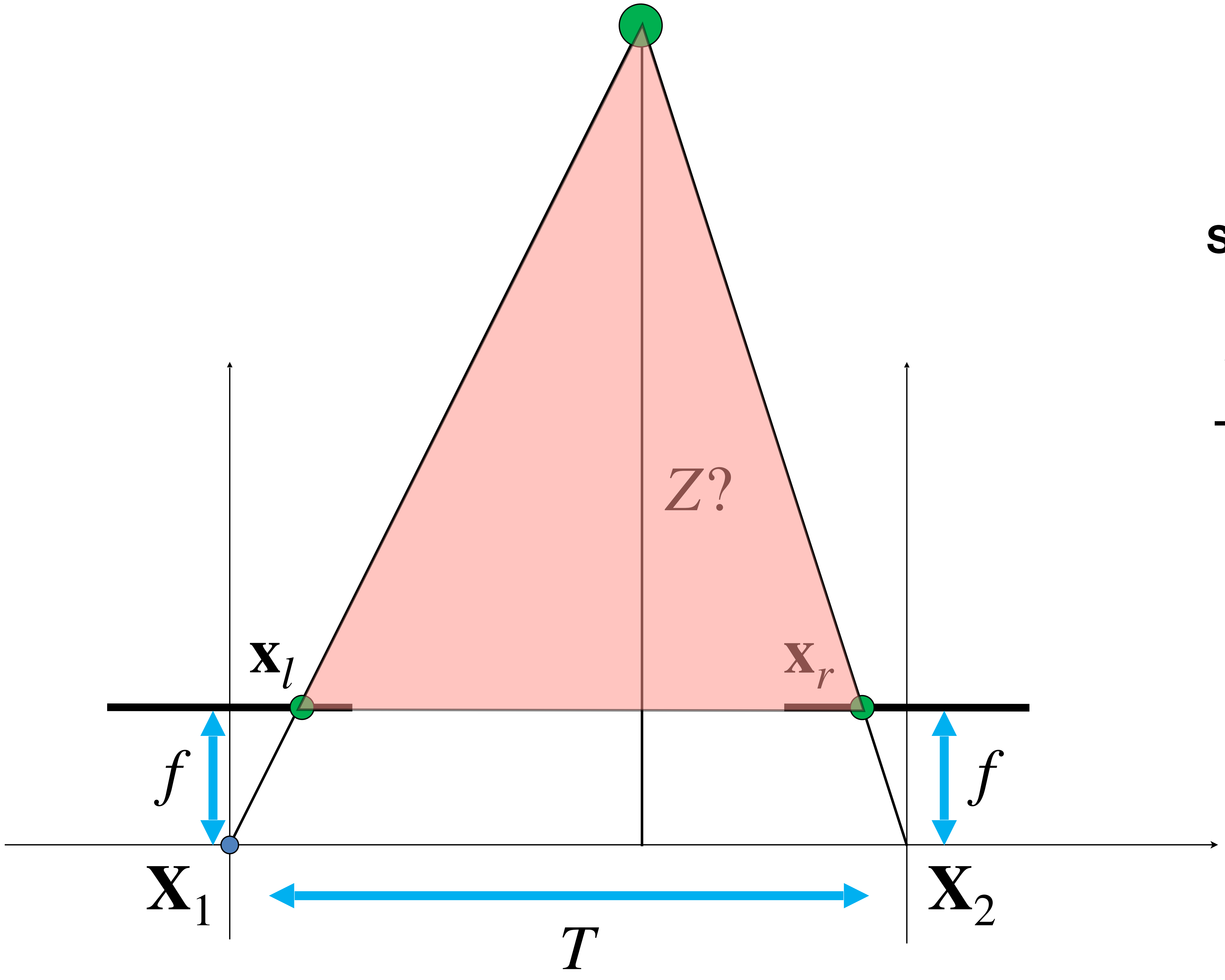
Geometry for a simple stereo system



Geometry for a simple stereo system



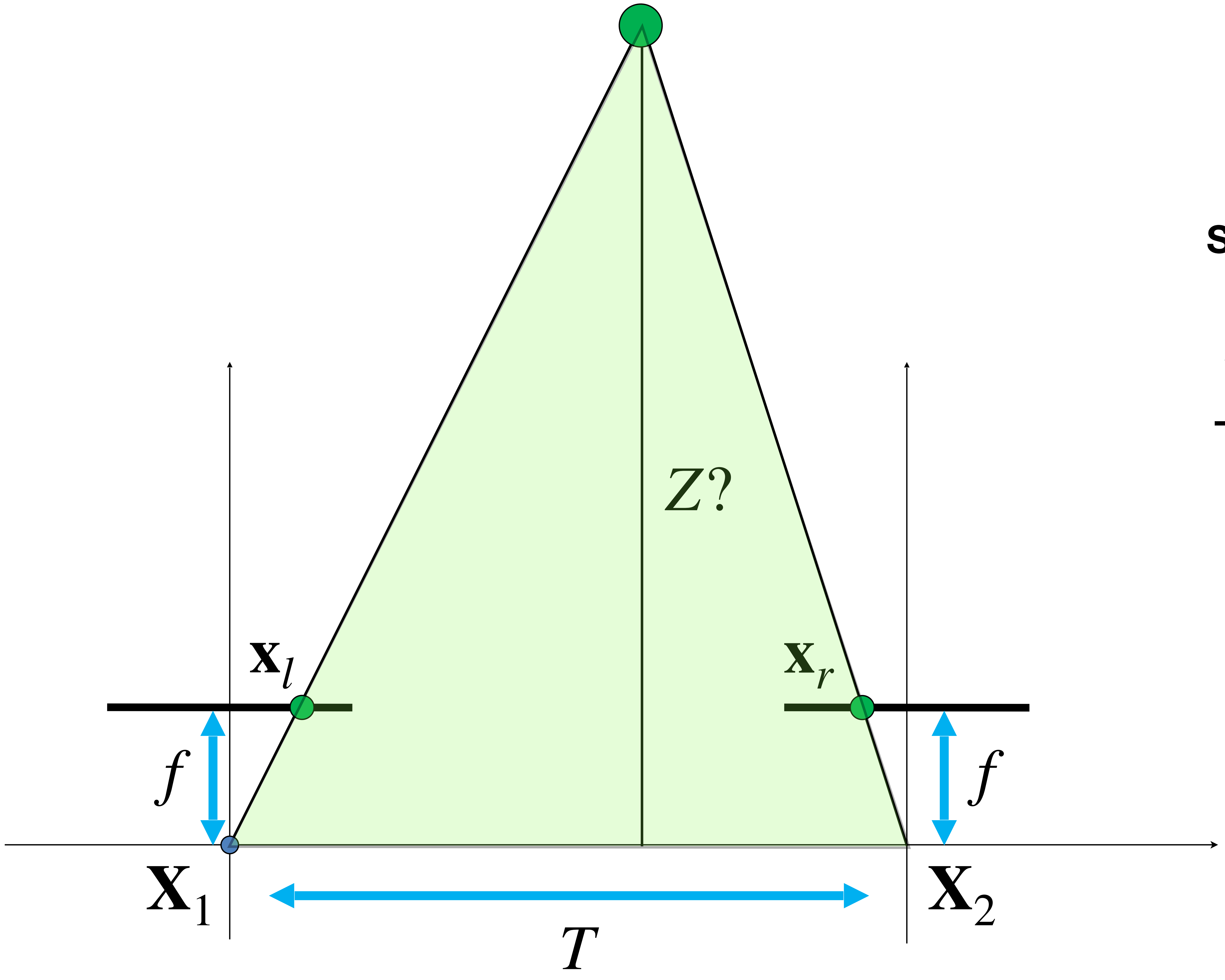
Geometry for a simple stereo system



Similar Triangles:

$$\frac{T + \mathbf{x}_r - \mathbf{x}_l}{Z - f} =$$

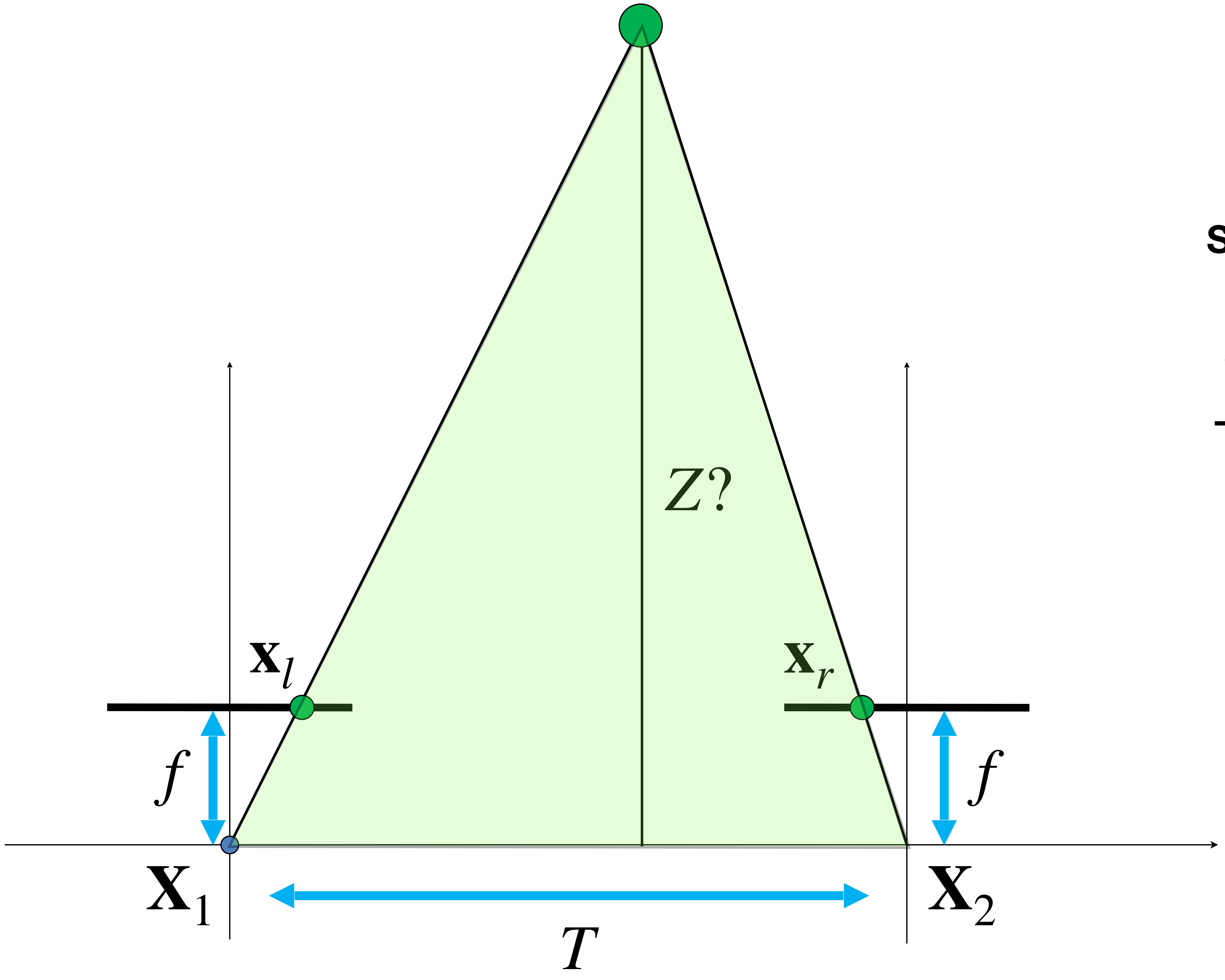
Geometry for a simple stereo system



Similar Triangles:

$$\frac{T + \mathbf{x}_r - \mathbf{x}_l}{Z - f} = \frac{T}{Z}$$

Geometry for a simple stereo system



Similar Triangles:

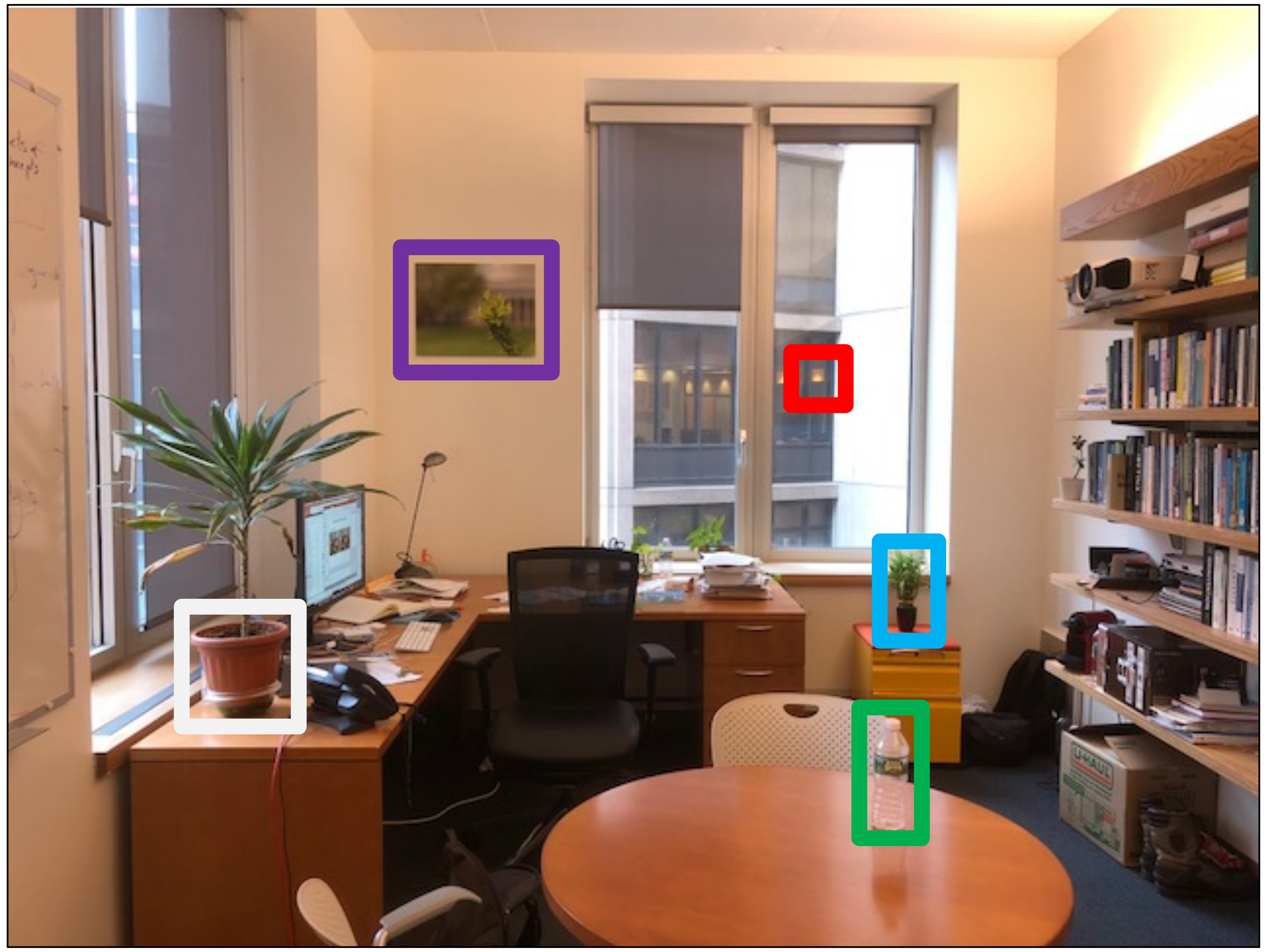
$$\frac{T + \mathbf{x}_r - \mathbf{x}_l}{Z - f} = \frac{T}{Z}$$

Solve for Z:

$$Z = f\frac{T}{\mathbf{x}_l - \mathbf{x}_r}$$

Disparity

Measuring disparity

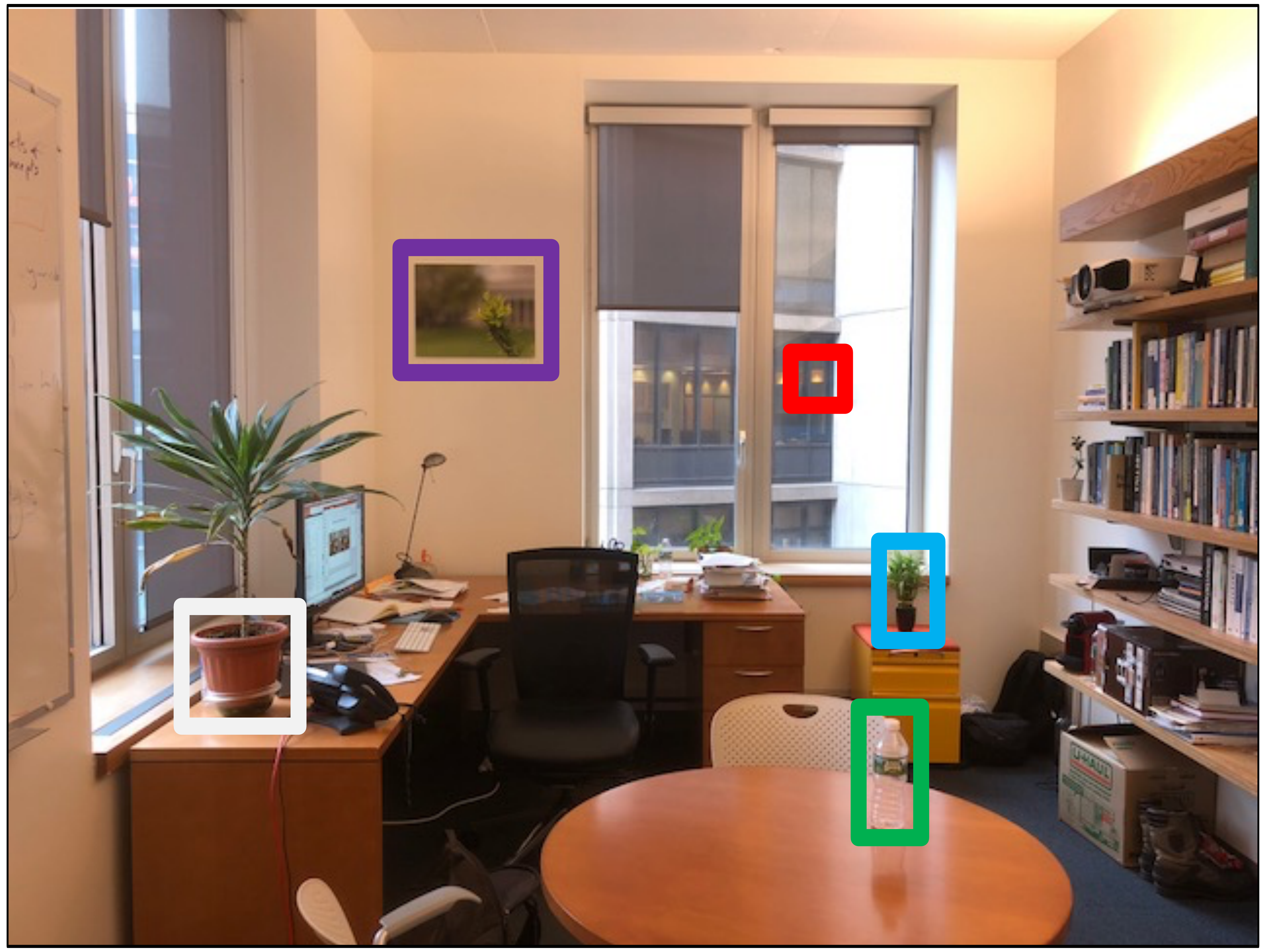


Left image

Right image

I took one picture, then I moved ~1m to the right and took a second picture.

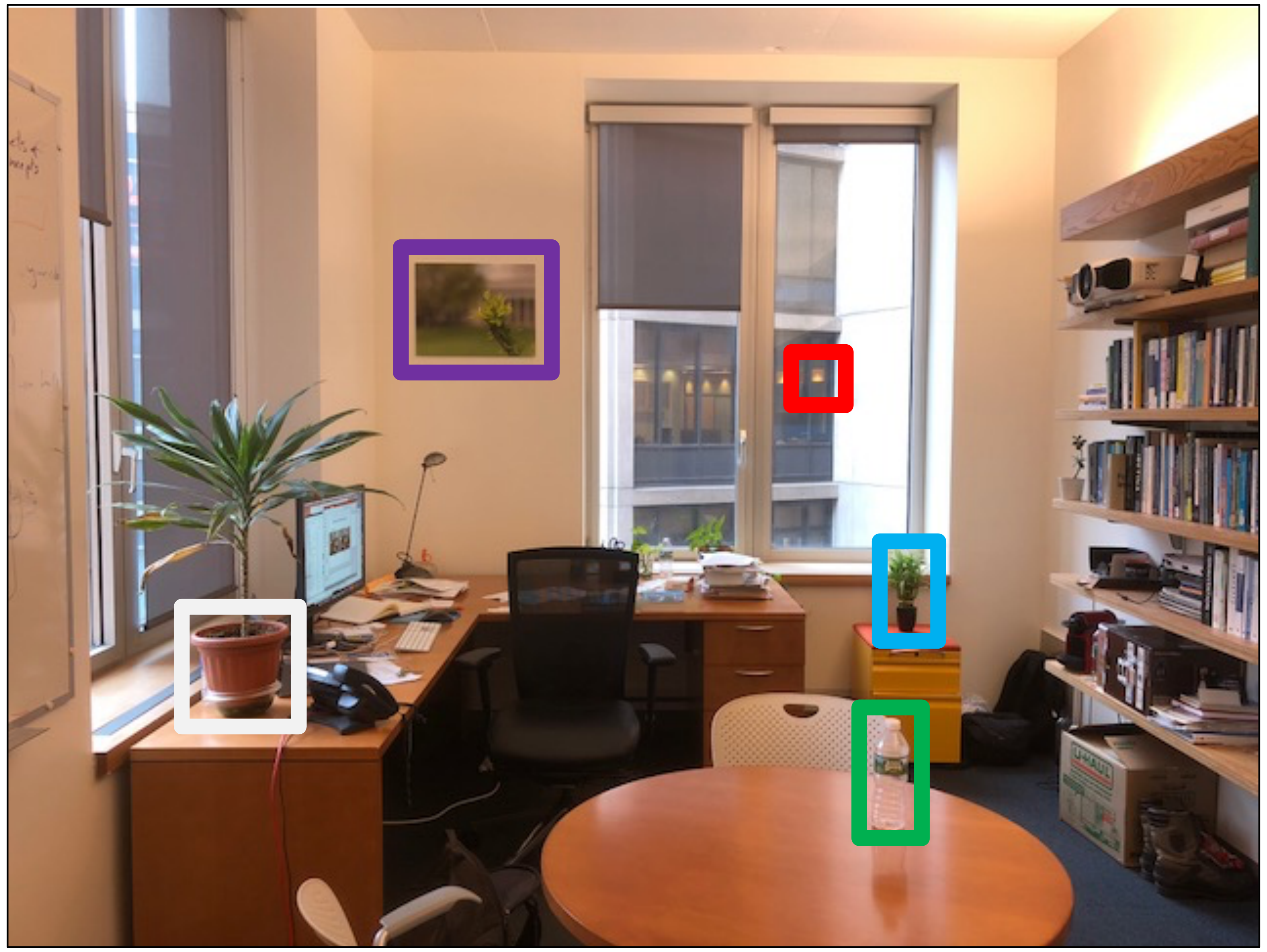
Measuring disparity



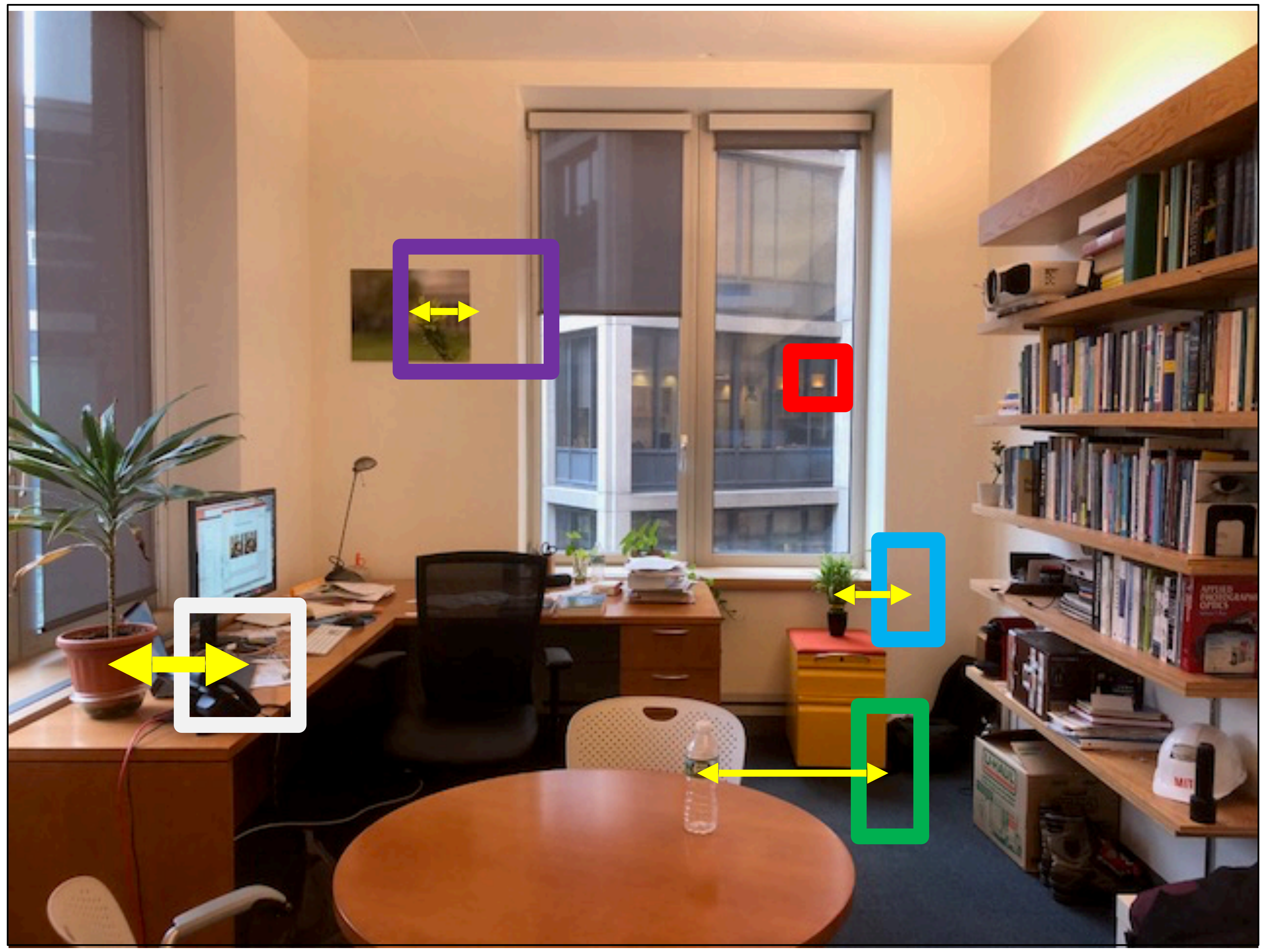
Left image

Right image

Measuring disparity



Left image



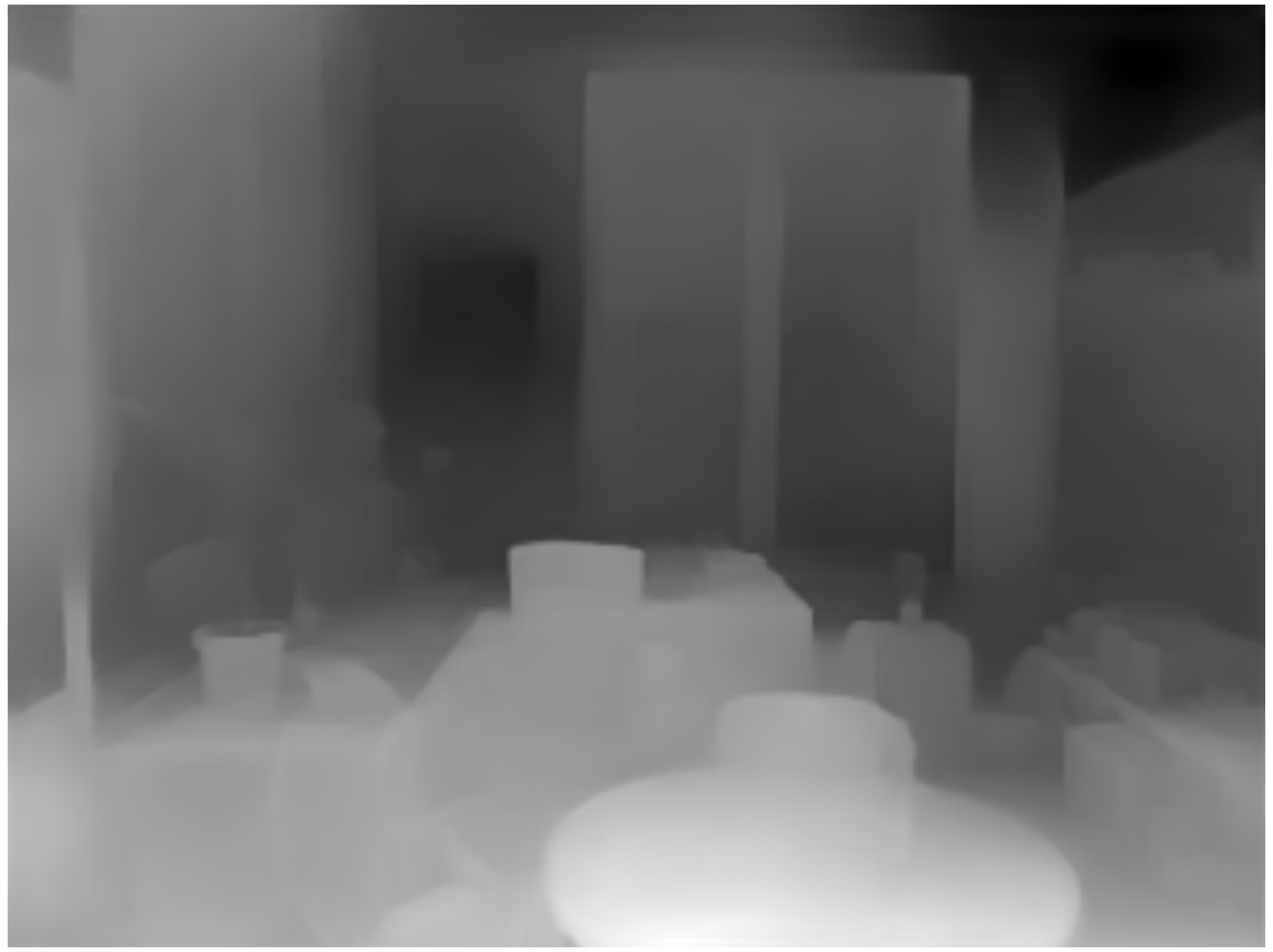
Right image

Disparity map

$I(x, y)$

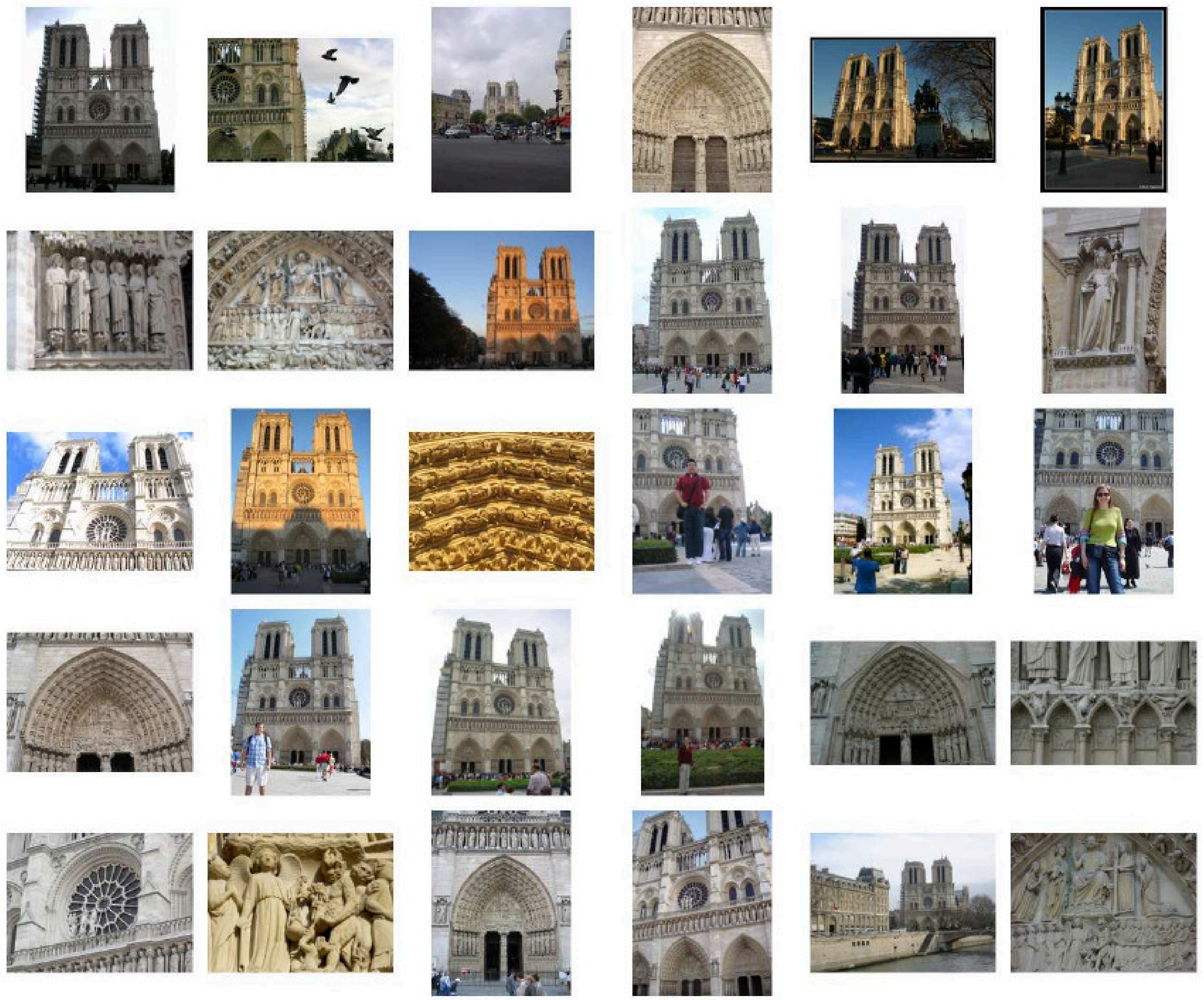
$I'(x, y) = I(x + D(x, y), y)$

$D(x, y)$



$$Z = f\frac{T}{D(x, y)}$$

Next time: 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.