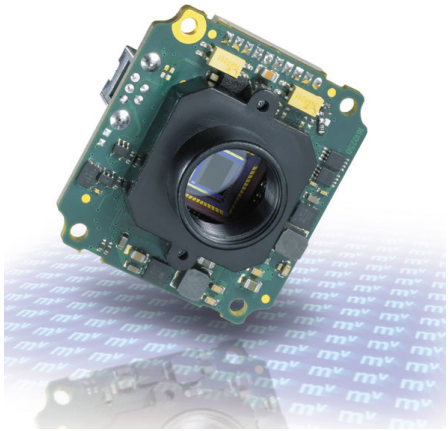
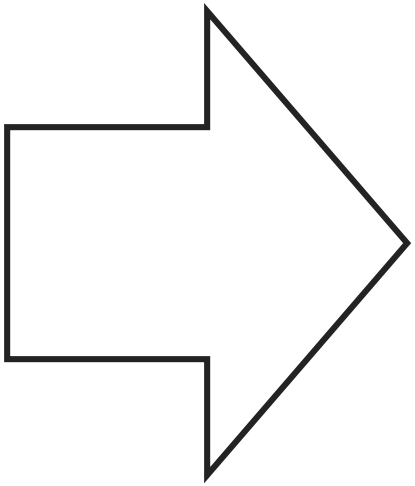
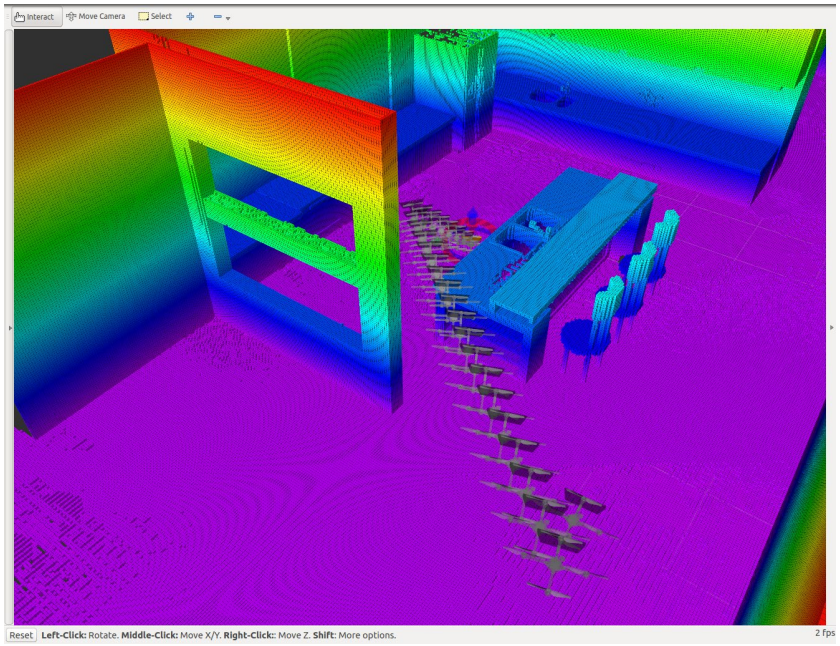
Perception:
Camera Modeling

High Level Goal

Understand how to go from images to pose and geometry

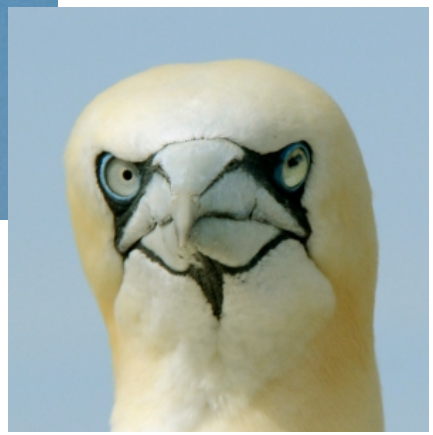






Motivating Example: Gannets

Gannets can make fast dives using only monocular vision



Each eye on the Gannet has a non-overlapping field of view

Nature Vol. 293 24 September 1981

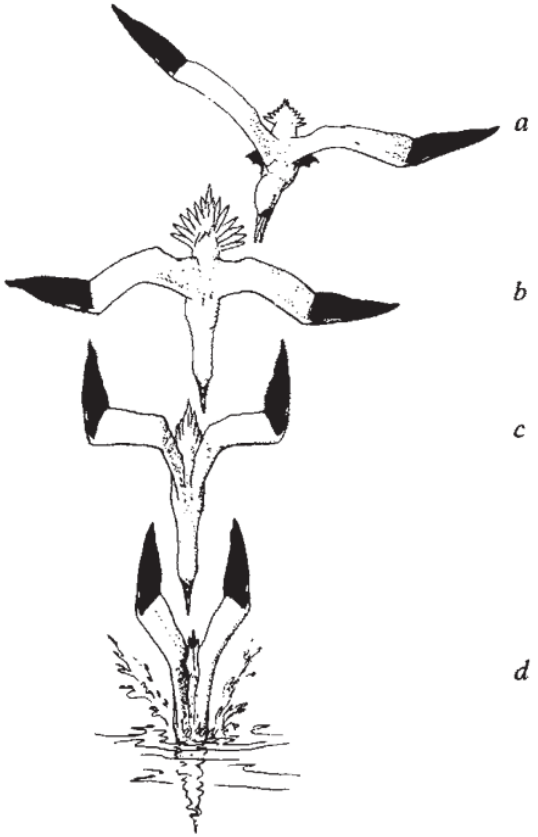
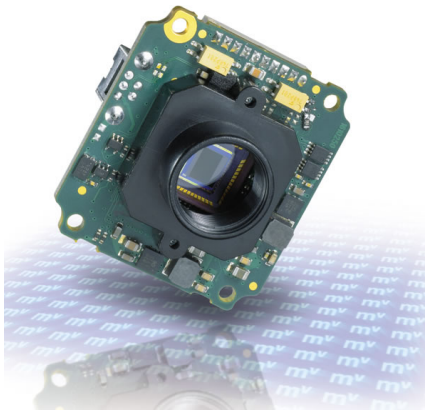


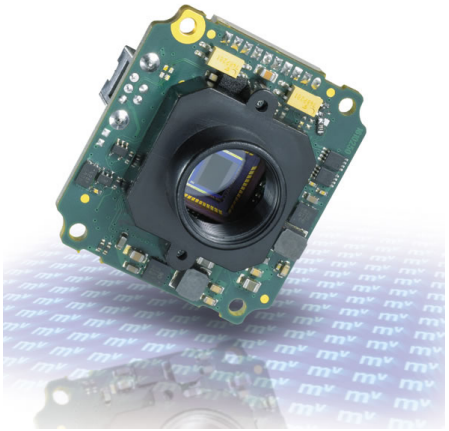
Fig. 2 Wing positions of diving gannet, *Sula bassana* (length ~0.9 m, wingspan 1.7 m). Illustration by John Busby. (Reprinted from ref. 7, courtesy of the author and publishers.)

Our ‘eyes’ – the Camera

- We use standard RGB camera
- Additional sensors will be added later
  - Focus on vision for now



Many Kinds of Cameras for Robotics



Standard RGB

Stereo

Rolling Shutter

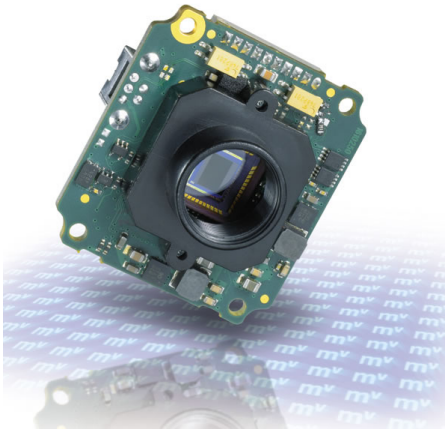
DVS Event-Based

Depth/RGBD

Multi-spectral

Multi-lens

Many Kinds of Cameras for Robotics



Standard RGB

We'll be focusing on Monocular RBG Cameras with Global Shutter

Stereo

Rolling Shutter

DAVIS Event-Based

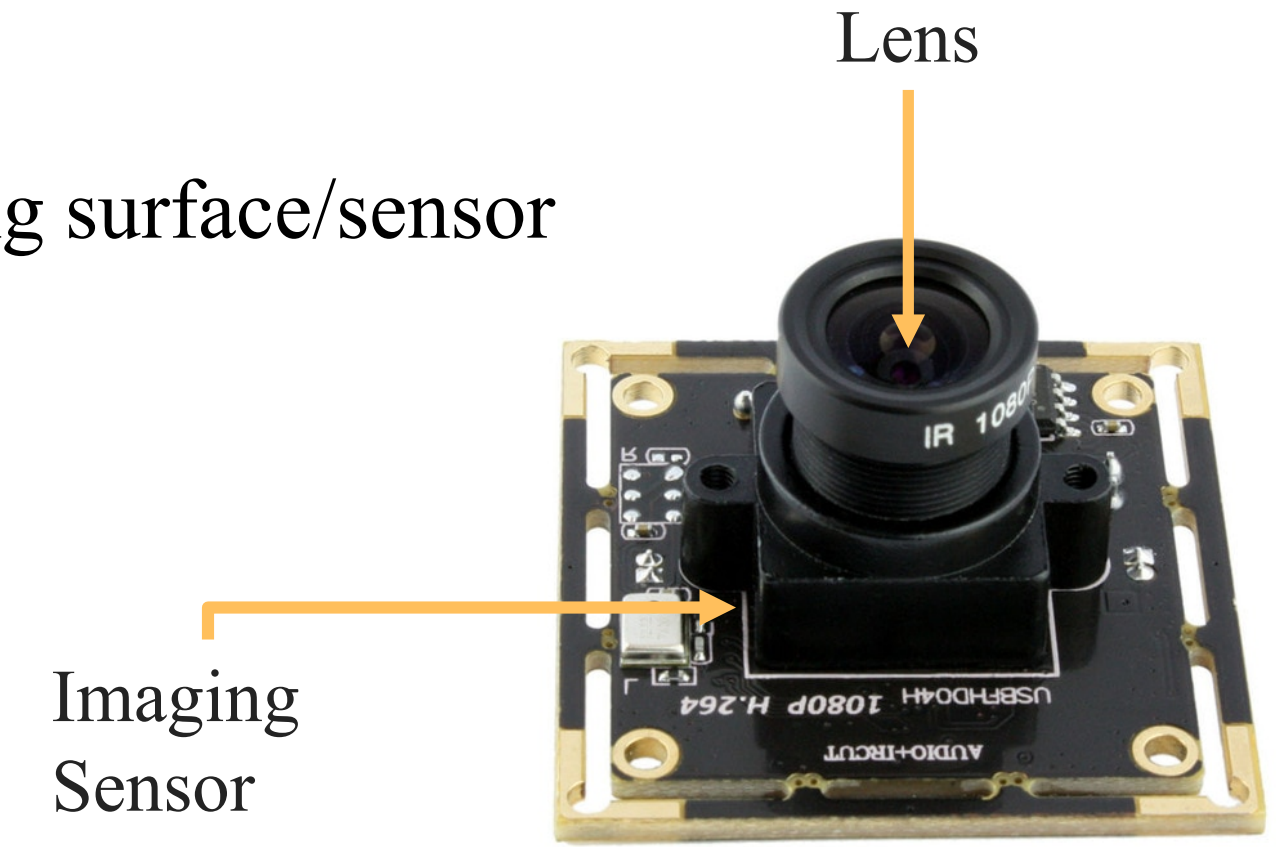
Depth/RGBD

Multi-spectral

Multi-lens

Creating a Camera Model

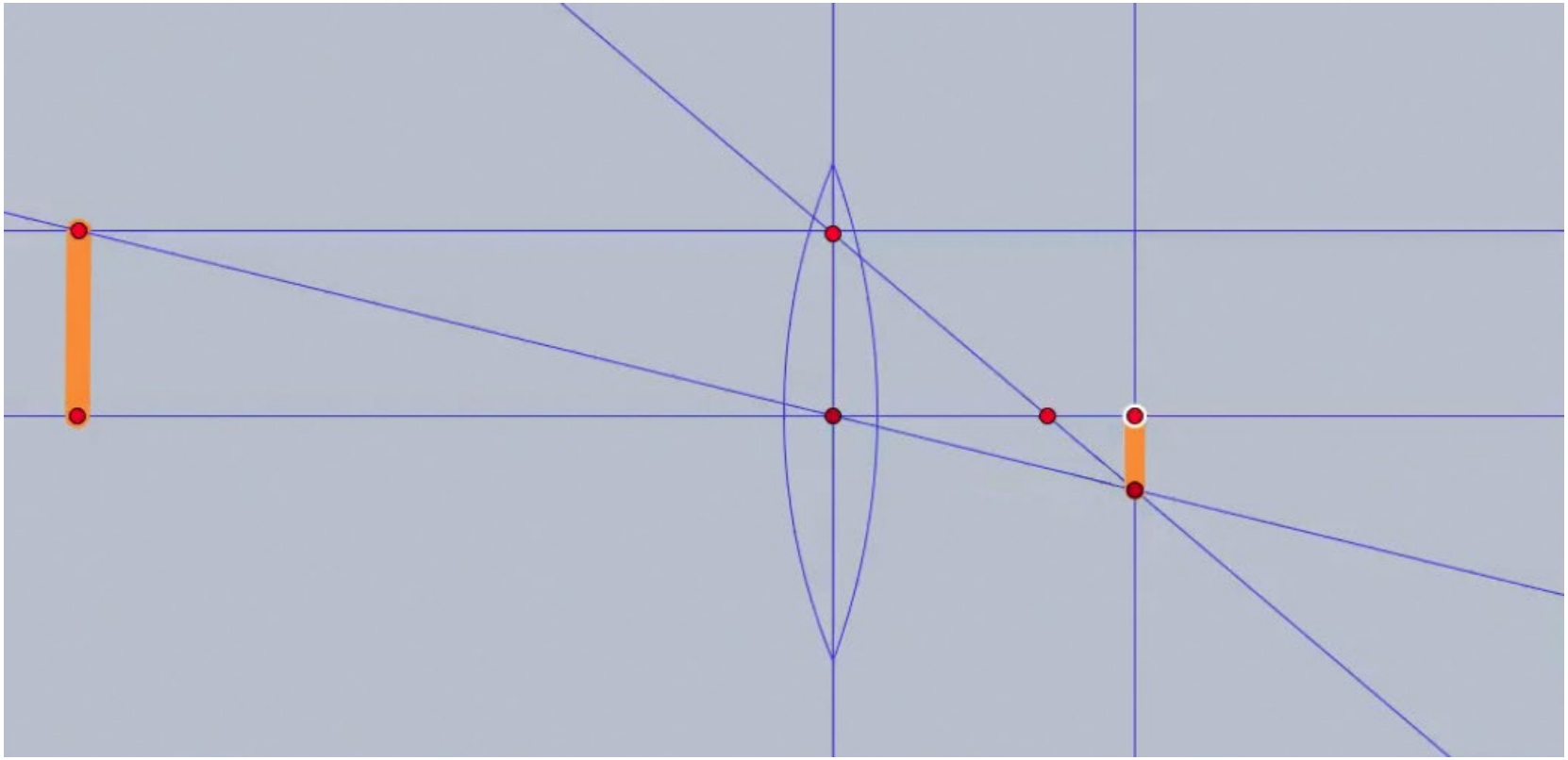
- Camera has two main components:
  - Lens
  - Imaging surface/sensor



Lens Geometry

2D Slice for ease of viewing – Thin Lens Model

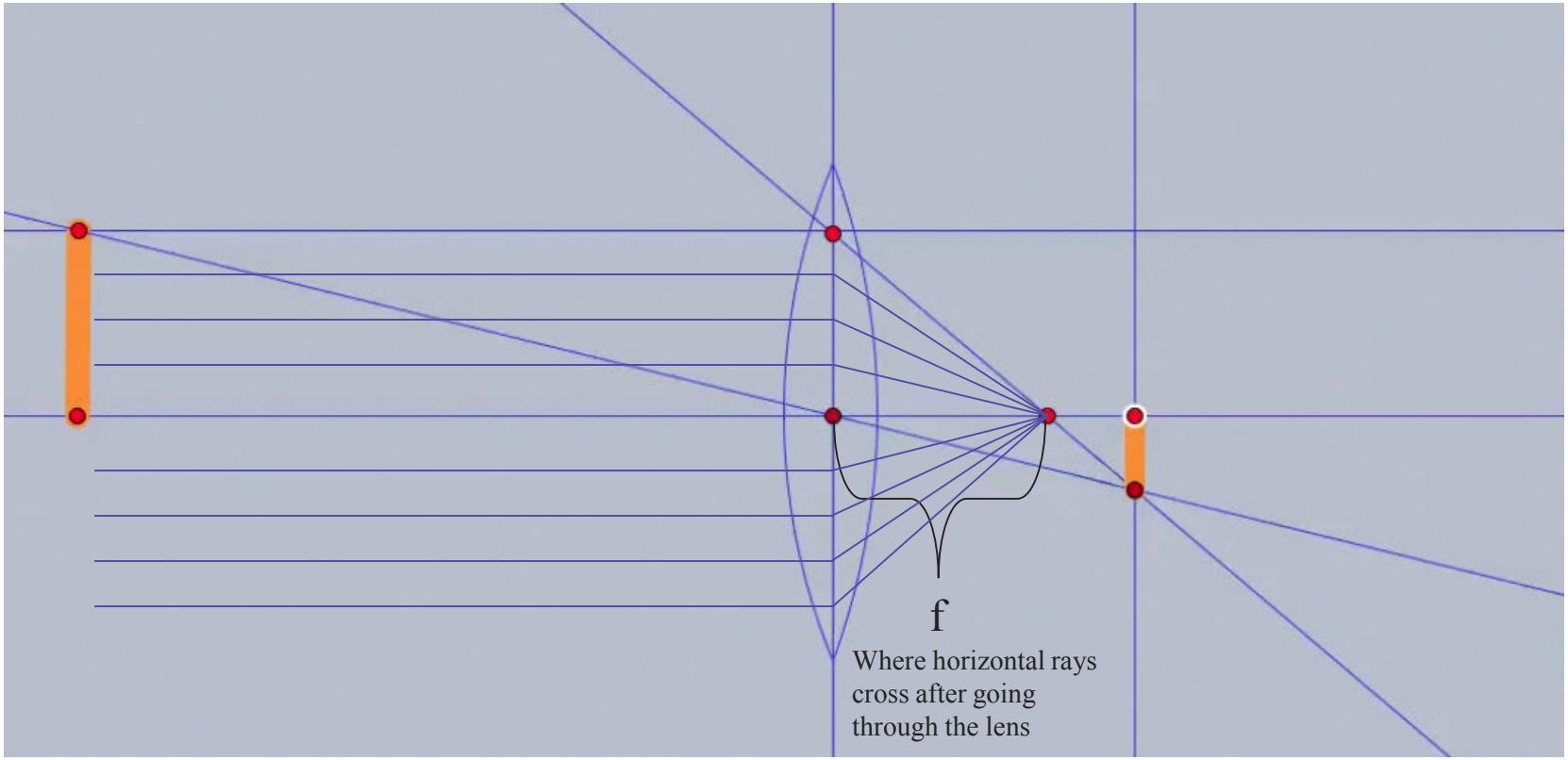
We will model from optics principles from physics



We will also not be using the standard naming conventions from physics for reasons that will be clear later

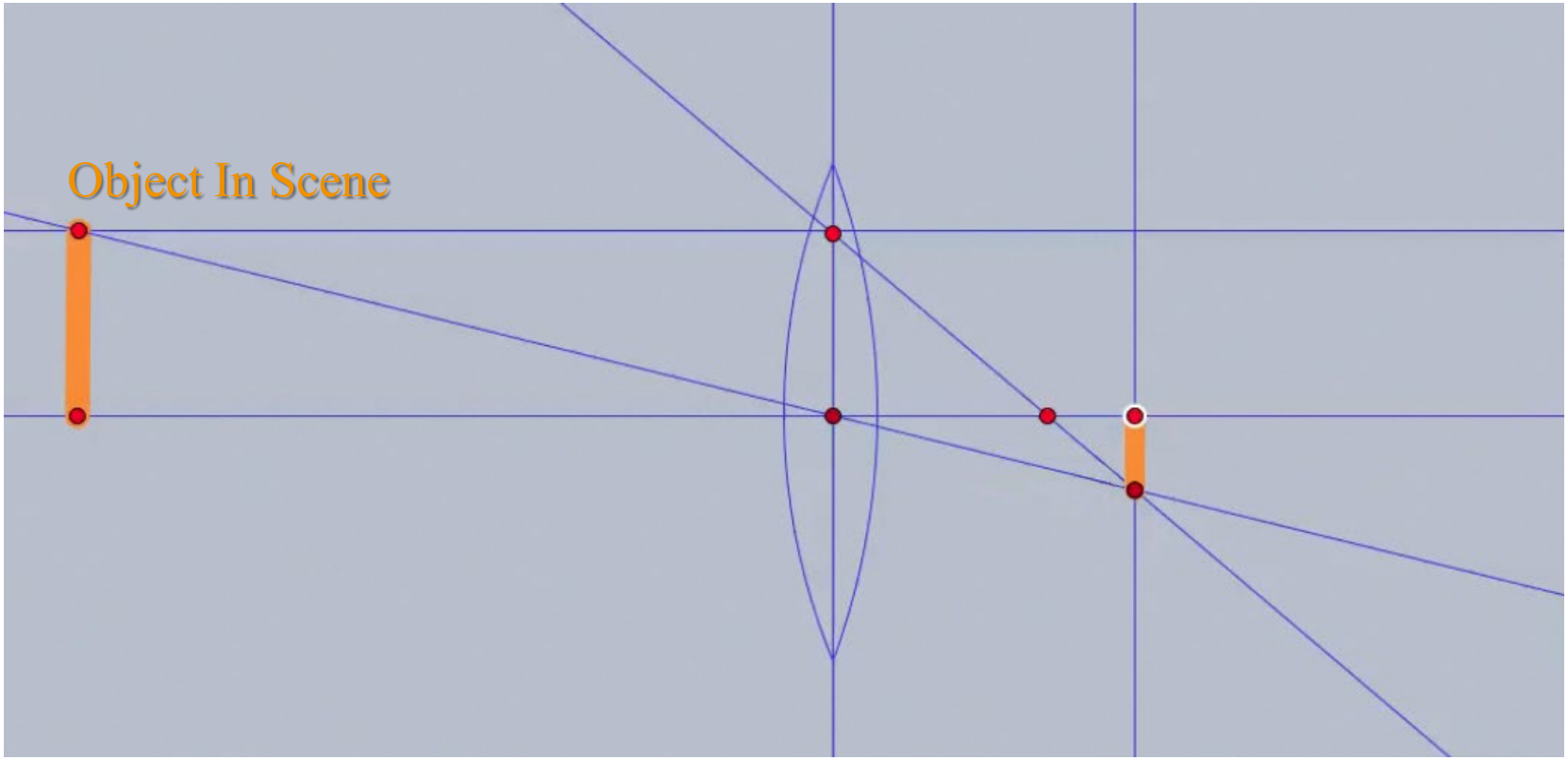
Lens Geometry

Lens properties



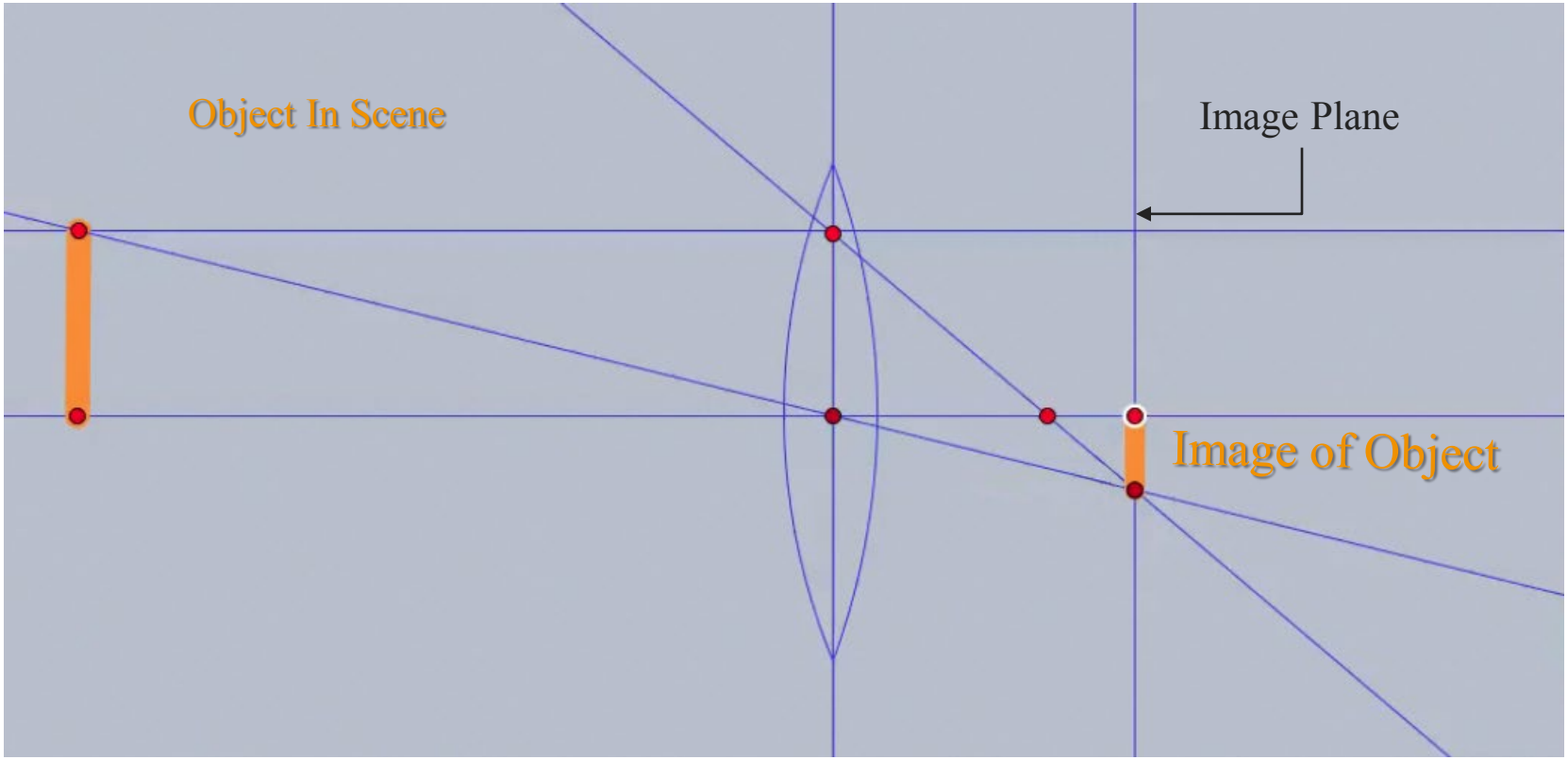
Lens Geometry

Given object in the scene



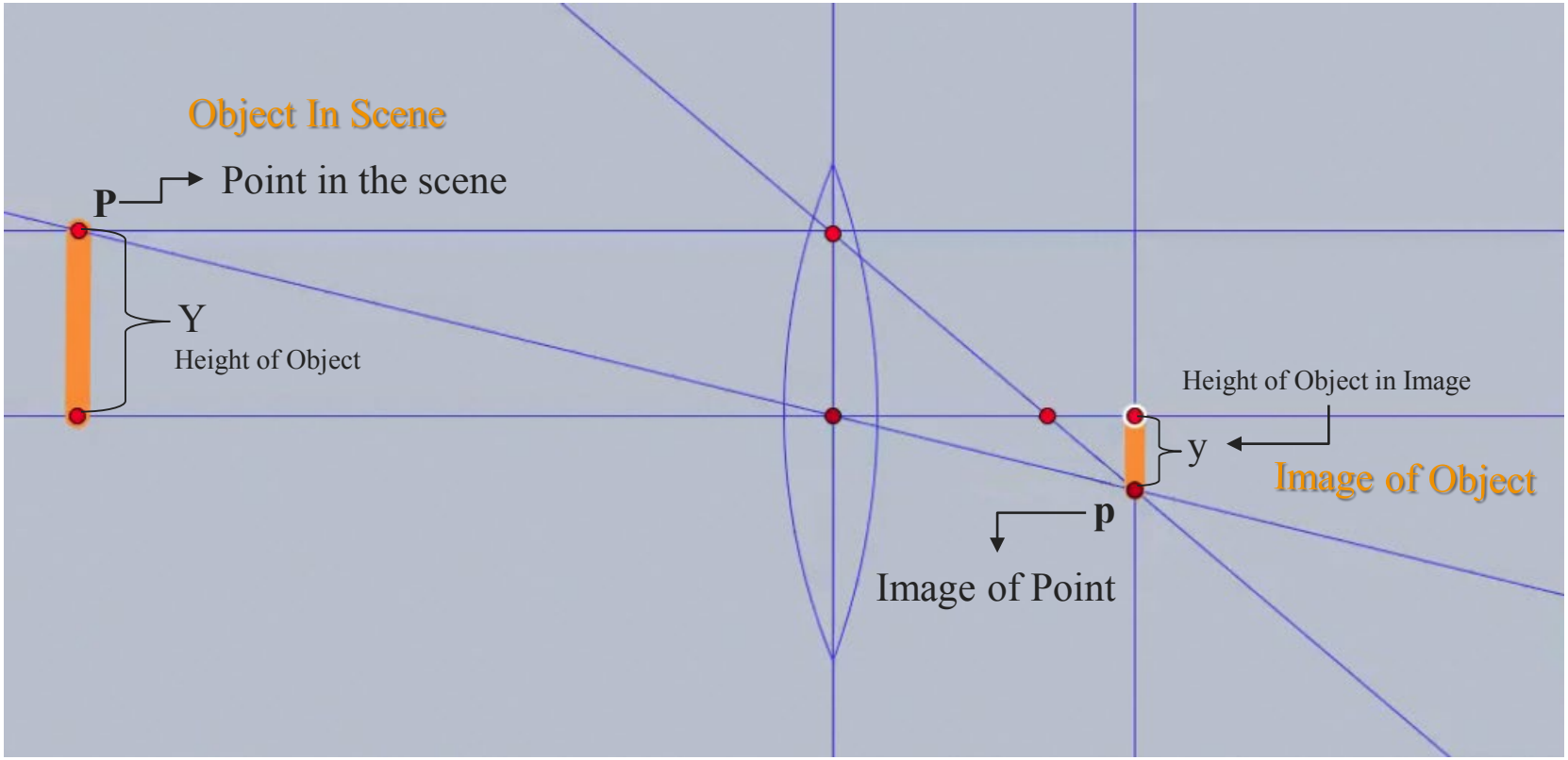
Lens Geometry

Lens and image



Lens Geometry

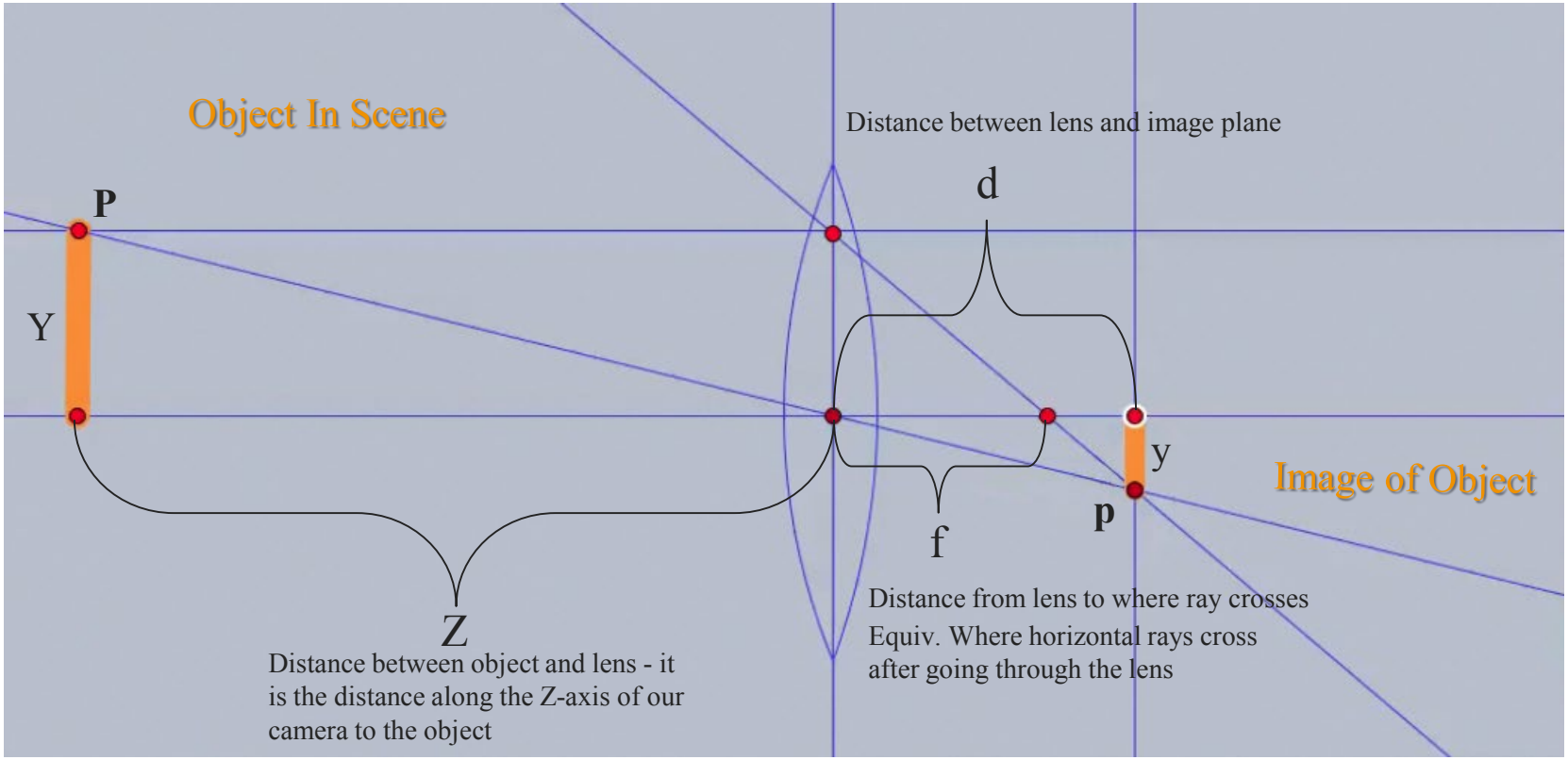
Object and Image properties



Lens Geometry

Parameters need to find image point
We get the thin lens equation

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{d}$$

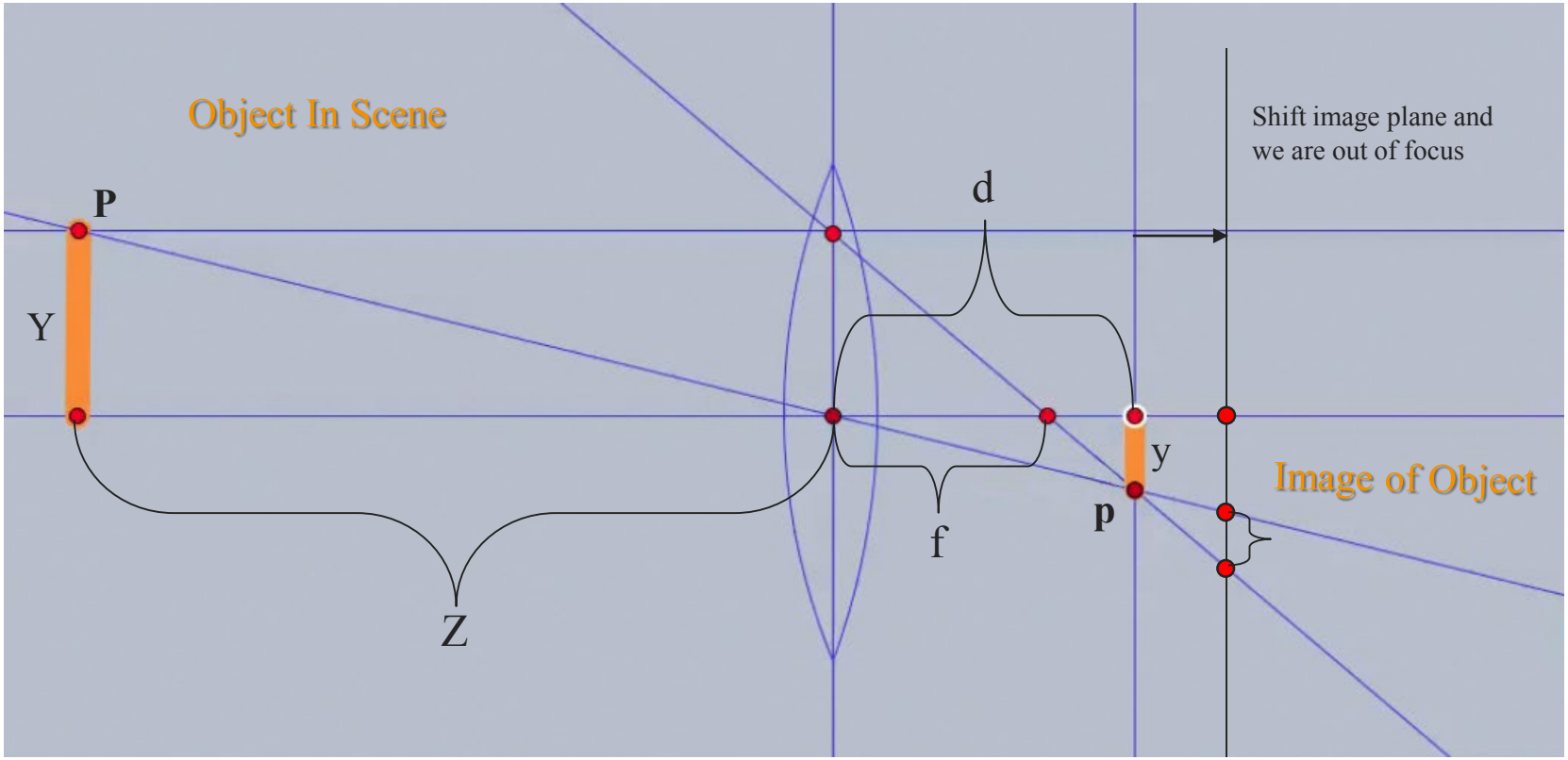


Assumes things are in focus (center and refracted rays meet)

Lens Geometry

If things are not in focus, the equation does not hold

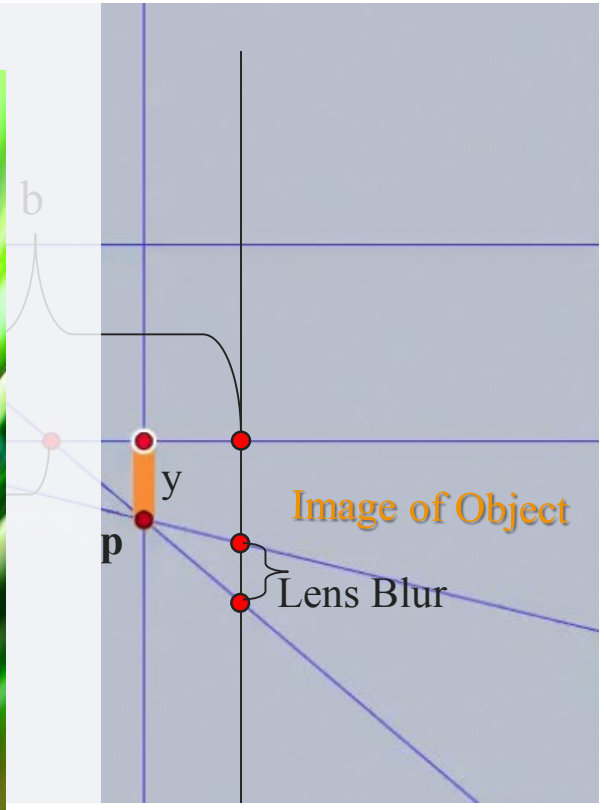
$$\frac{1}{f} \neq \frac{1}{Z} + \frac{1}{d}$$



Aside: Lens Blur

When rays do not meet, this is what causes things to be in and out of focus

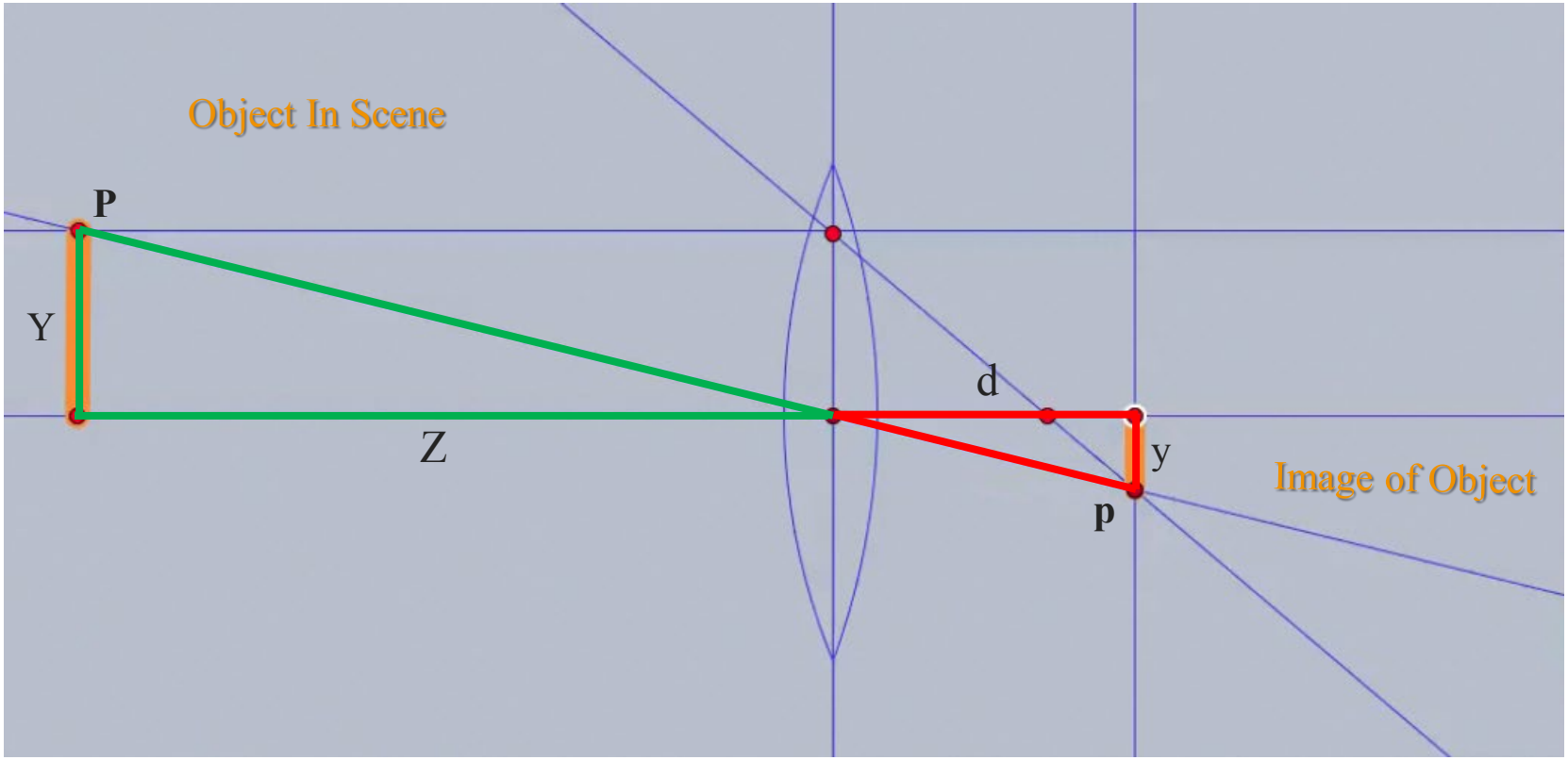
$$\frac{1}{f} \neq \frac{1}{Z} + \frac{1}{d}$$



Lens Geometry

When image is in focus, we discover similar triangles

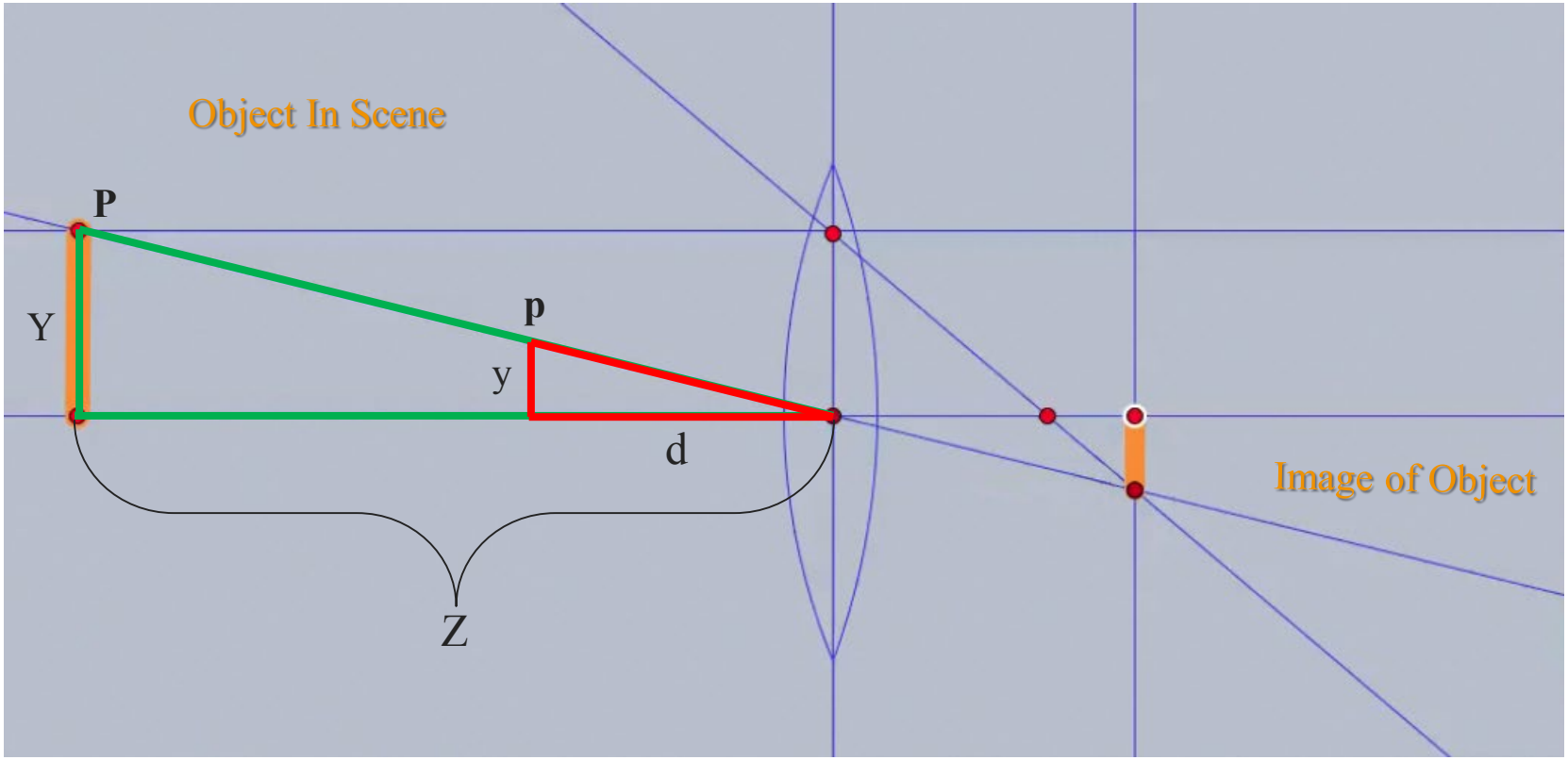
$$\frac{Y}{Z} = \frac{y}{d}$$



Lens Geometry

To not deal with the upside down image,
We put image in front of lens

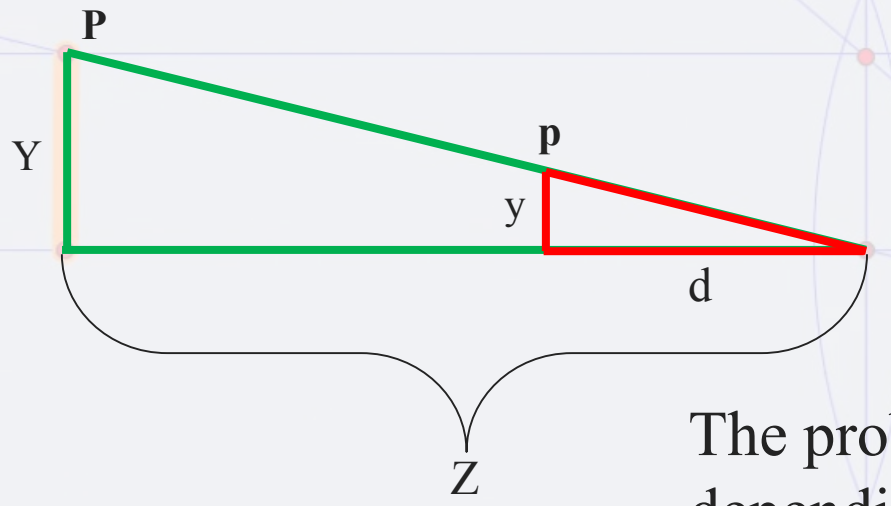
$$\frac{Y}{Z} = \frac{y}{d}$$



Lens Geometry

To not deal with the upside down image,
We put image in front of lens
Then solve for y

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



$$\Longrightarrow \quad y = d\frac{Y}{Z}$$

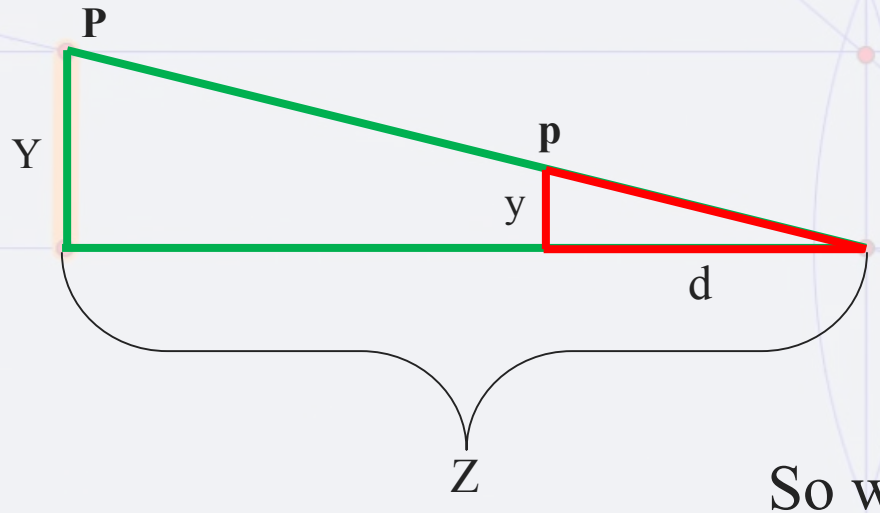
We're almost there, but this equation doesn't quite work

The problem is d – it changes depending on where the object is in the scene

Lens Geometry

To not deal with the upside down image,
We put image in front of lens
Then solve for y

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



$$\Longrightarrow \quad y = d\frac{Y}{Z}$$

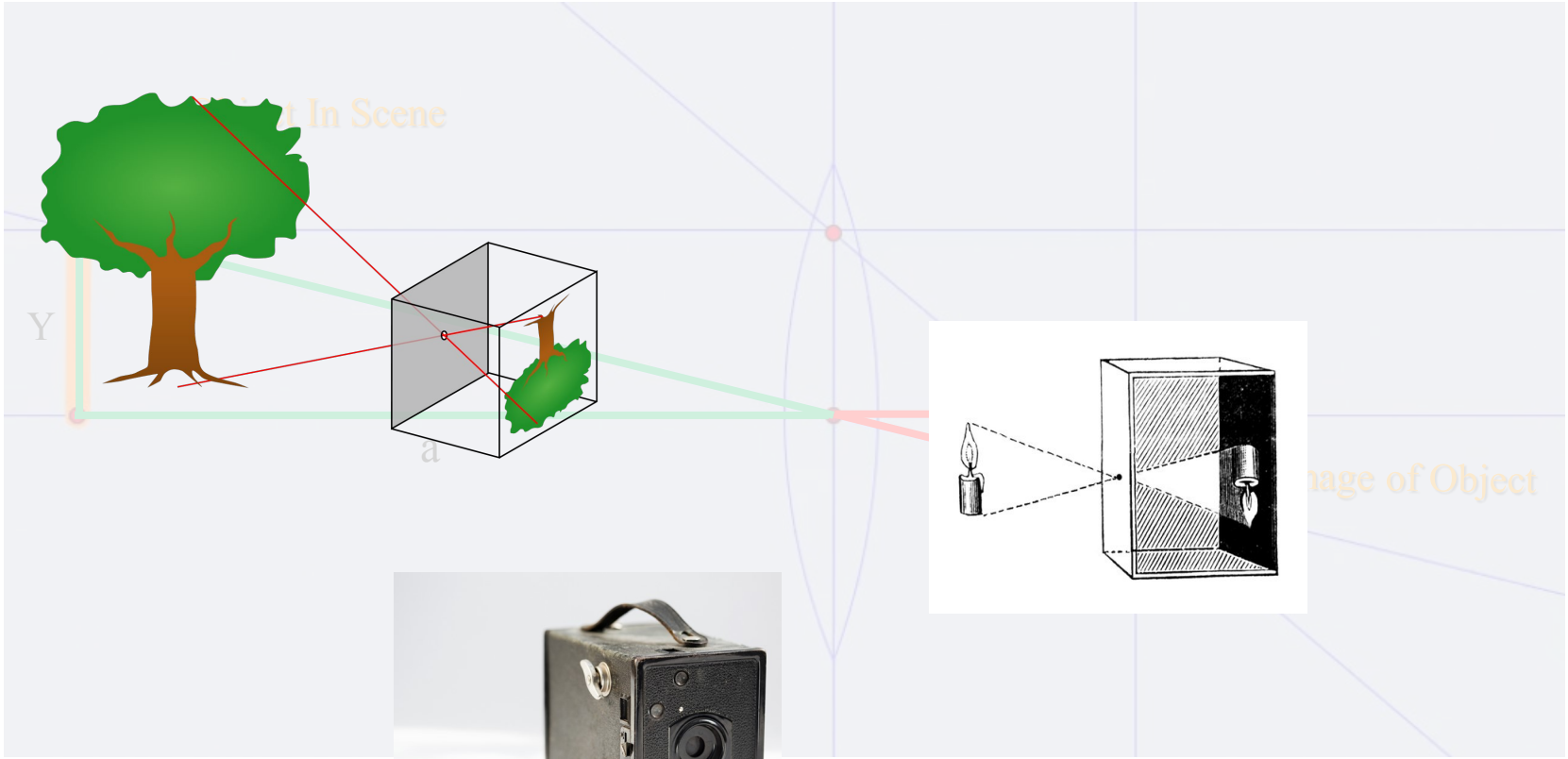
$$y \approx f\frac{Y}{Z}$$

So we replace d with f and call it day (this will be justified later)

Pinhole Model

This is known as the pinhole model,
Or perspective projection

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



Going from 2D to 3D

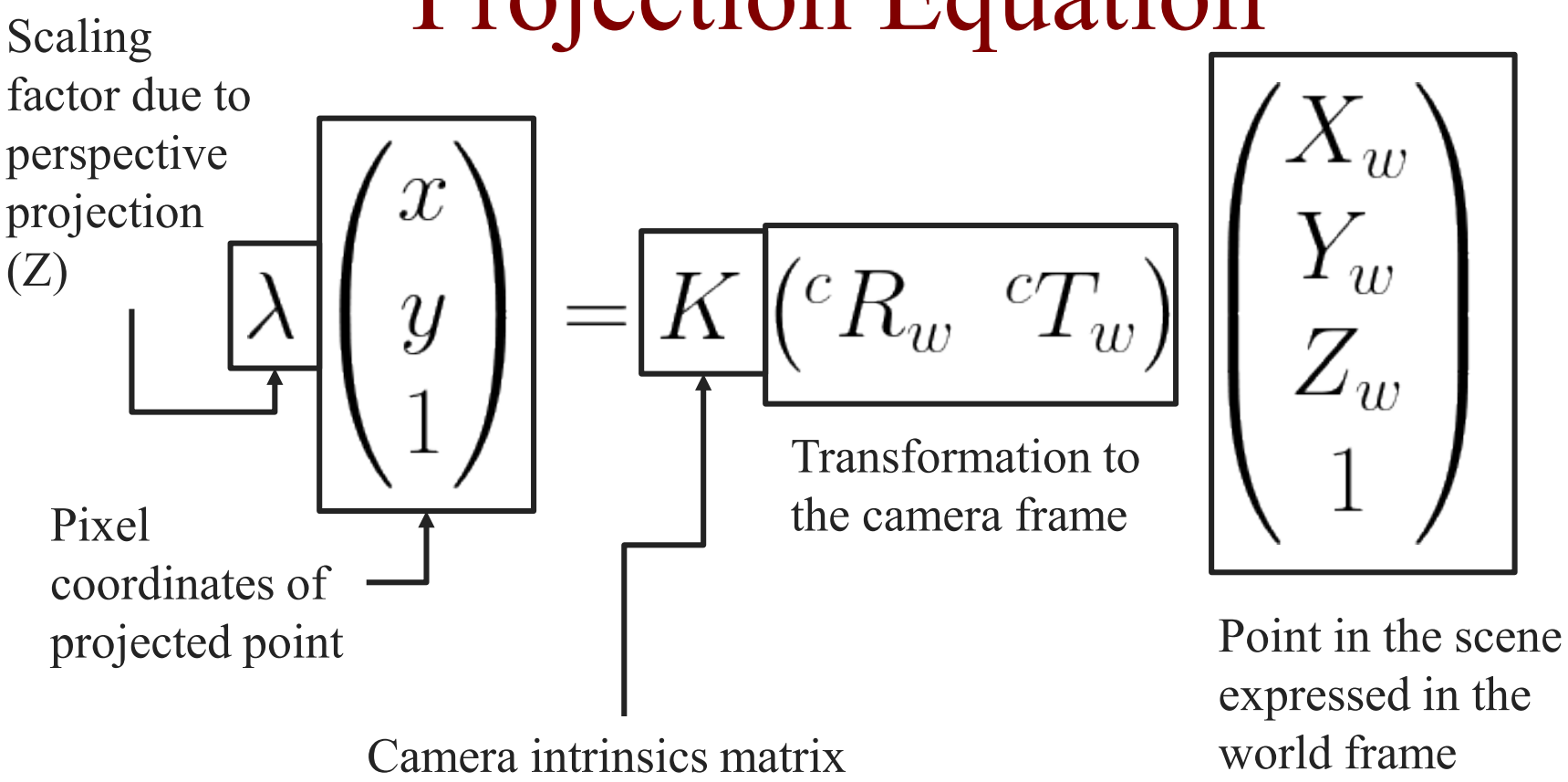
- In our applications we deal with 2D images and a 3D world, so we need to extend it:

$$x = f\frac{X_c}{Z_c} \qquad y = f\frac{Y_c}{Z_c}$$

- Normally we express this in terms of a matrix equation:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = K \begin{pmatrix} {}^cR_w & {}^cT_w \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Projection Equation

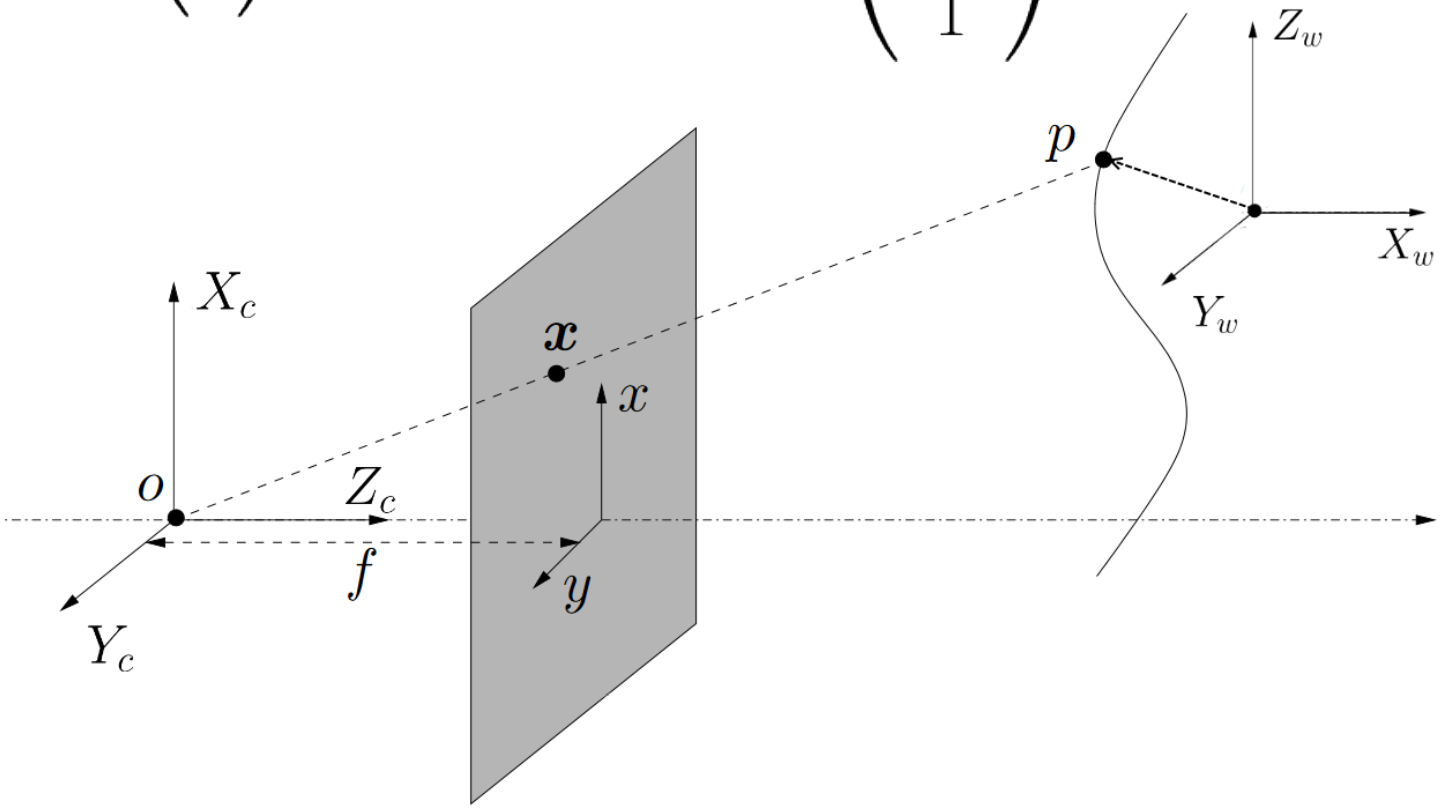


$$\begin{pmatrix} f & 0 & x_o \\ 0 & f & y_o \\ 0 & 0 & 1 \end{pmatrix}$$

3 x 3 matrix that includes the focal length and an offset in case our camera isn't centered (We'll address how we get these parameters later)

Diagram of Projection Equation

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{pmatrix} {}^{c}R_{w} & {}^{c}T_{w} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$



Problems with the Model

- We still have some questions about the model so we need to look into them to make sure we can use this in practical settings

Problem #1: Lens Blur

- Callback: Lens blur causes Depth of field effects

The fact that we have a lens and a non-zero focal length causes errors in the projections

Not a problem in true pinhole cameras

Can we ignore this?

Problem #1: Lens Blur

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

Once again problem is d – we approximated it as f it really follows the thin lens equation:

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{d}$$

Are we stuck? Or can we just ignore this?

Ignoring Lens Blur

Let's see how close f and d are:

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{d}$$

$$f = \left(\frac{Z}{Z+d}\right) d$$

$$Z \gg d \implies f \approx d$$

So if the objects are far enough away relative to the correct image plane we're fine

Ignoring Lens Blur

Let's see how close f and d are:

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{d}$$

$$f = \left(\frac{Z}{Z+d}\right) d$$

$$Z \gg d \implies f \approx d$$

So if the objects are far enough away relative to the correct image plane we're fine

Another way to look at it is through the relative error:

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{d}$$

$$\frac{d}{f} = \frac{d}{Z} + 1$$

$$\frac{d-f}{d} = \frac{f}{Z}$$

So f stays constant for a particular camera, so as Z increases the relative error decreases

Problem #1: Solved

- Just make sure what we are looking at is far enough away

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

In practice, we don't use the f given in the camera but compute the number that best satisfies this using **calibration** (more on that later)

Problem #2: Radial Distortion

- We typically want a wide field of view, but this leads to strange lens effects

This is an example image – notice the lines on the checkerboard which should be straight but due to the distortion they bend

Problem #2: Radial Distortion

We can actually model this distortion pretty well with a polynomial of the norm of the image coordinates:

$$r = \sqrt{x^2 + y^2}$$

$$x' = x(1 + k_1 r + k_2 r^2 + \ldots)$$

$$y' = y(1 + k_1 r + k_2 r^2 + \ldots)$$

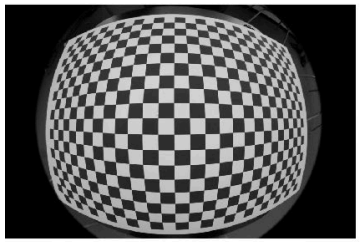
We can just invert this transform and then our coordinates then our original model will be valid again.

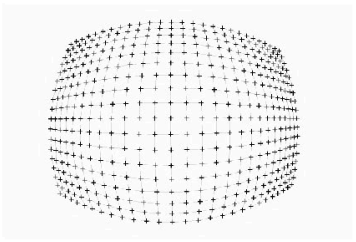
What polynomial exactly is used depends on exact method being used

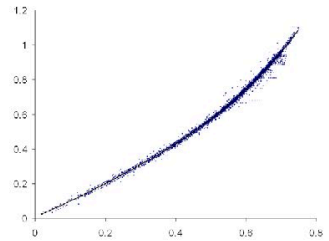
More advanced info to satiate curiosity:
https://arxiv.org/pdf/cs/0307047.pdf

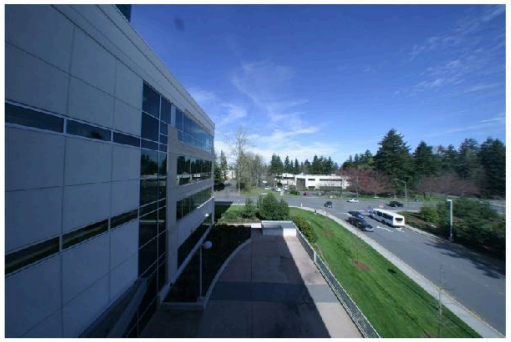
Problem #2: Radial Distortion

Undistorting this has some drastic effects especially for fish-eye lenses (very wide field of view lenses)





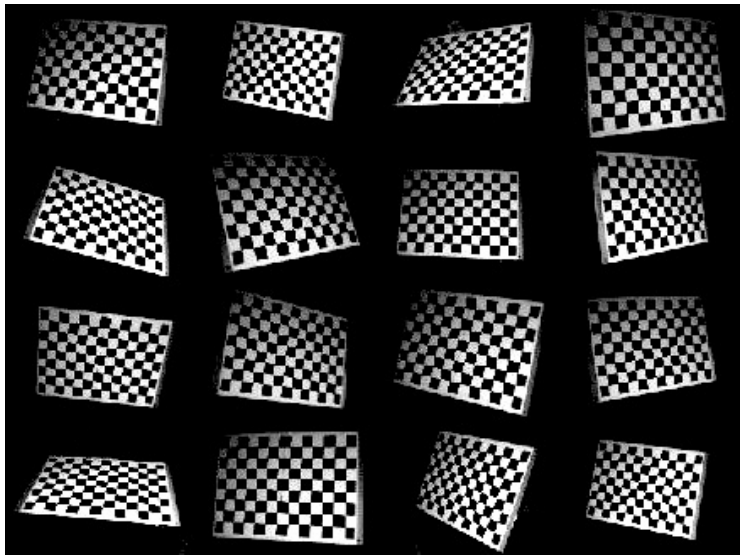




How do we compute the distortion coefficients?

Calibration

We get a checkerboard with known dimensions and use that to estimate the distortion



$$x = \frac{X_c}{Z_c}, \quad y = \frac{Y_c}{Z_c}$$

$$r = \sqrt{x^2 + y^2}$$

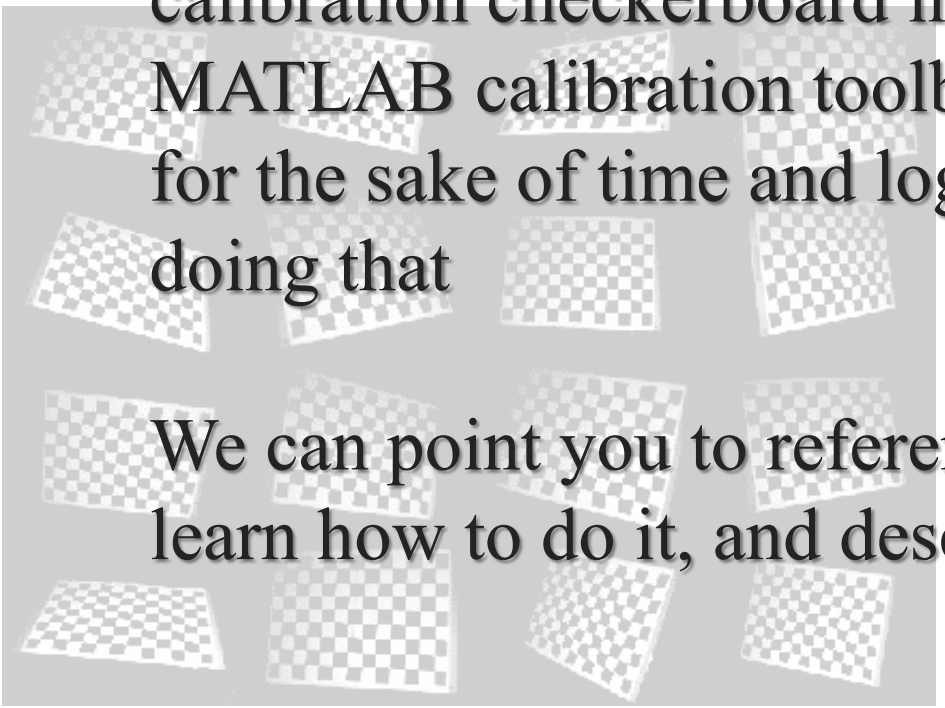
$$x' = fx(1 + k_1 r + k_2 r^2 + \ldots) + x_0$$
$$y' = fy(1 + k_1 r + k_2 r^2 + \ldots) + y_0$$

This is a well understood problem, there is a lot of code online to do this e.g. https://www.mathworks.com/help/vision/ug/camera-calibration.html

Calibration

We get a checkerboard with known dimensions and use that to estimate the distortion

We used to make you guys take pictures of a calibration checkerboard in class and run the MATLAB calibration toolbox to calibrate it but for the sake of time and logistics we are not doing that



We can point you to references if you want to learn how to do it, and describe it briefly here

$$r = \sqrt{x^2 + y^2}$$

$$x_d = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_d = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$x = f\frac{X}{Z} + x_o$$

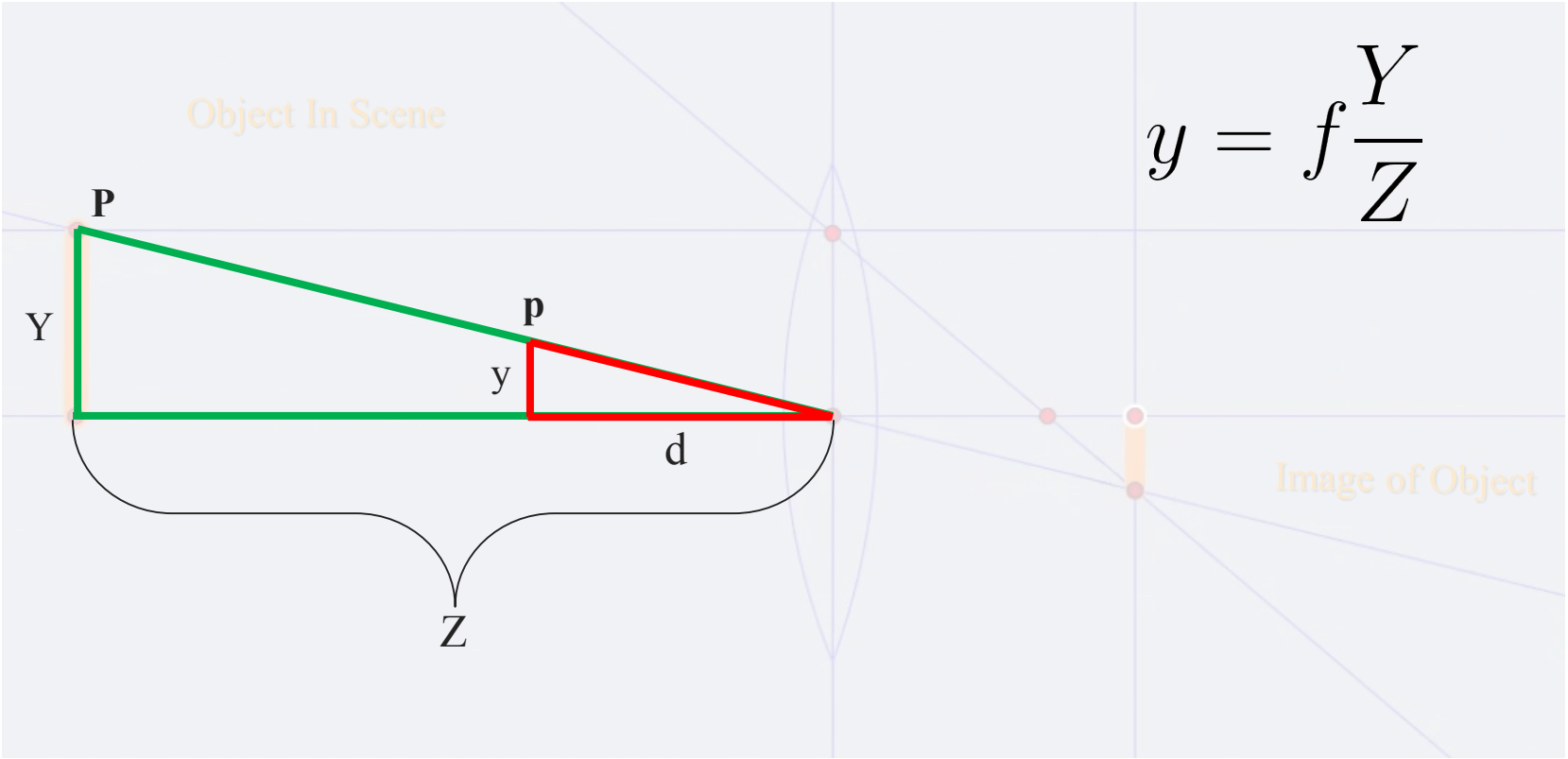
$$y = f\frac{Y}{Z} + y_o$$

Offsets in case the camera isn't perfectly aligned

This is a well understood problem, there is a lot of code online to do this e.g. https://www.mathworks.com/help/vision/ug/camera-calibration.html

Problem #3: Ambiguous Scale

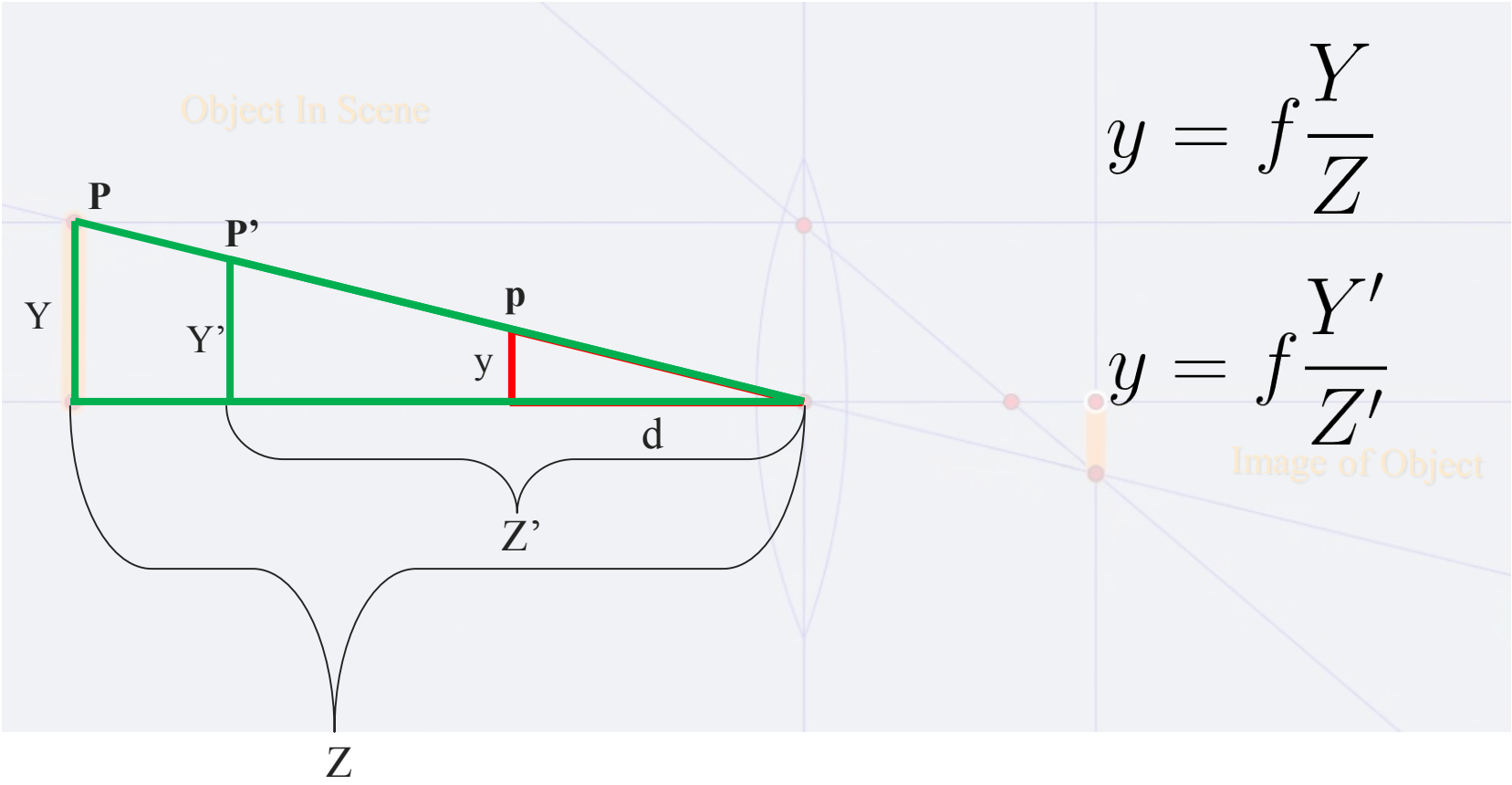
Recall when we derived our original projection equation we got a nice property using similar triangles



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

Problem #3: Ambiguous Scale

However because it only requires similar triangles, we cannot distinguish distances



Problem #3 Solution: Give up

- Unfortunately we cannot solve this one – it is an inherent ambiguity of vision due to the loss of information going from 2D to 3D

It's why images like this one can look realistic despite the vast differences in scale

What to do?

- Fortunately all hope is not lost
- The scale ambiguity leads to **Projective Geometry**
- From this we can extract geometric information about the world
- We will study this next lecture