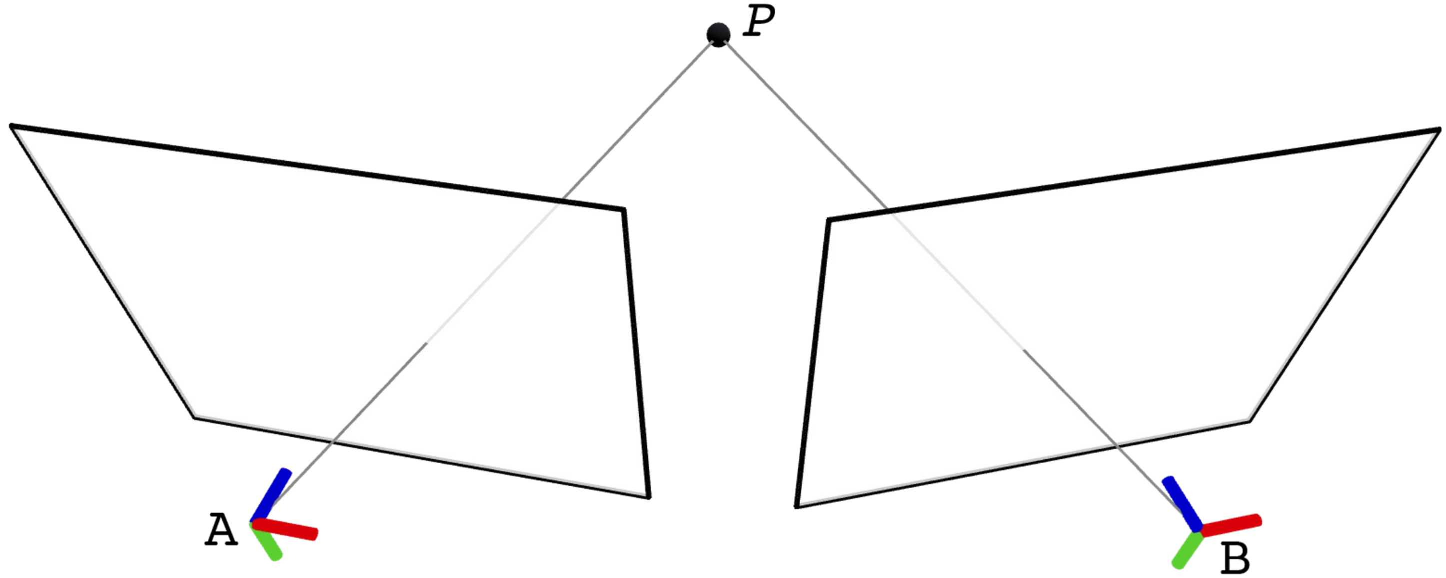
CS 180 Discussion #8

World, Cameras, Pixels, Rays



Welcome!!

Logistics

- Midterm grades this week
- Project 4 released

Agenda

3D coordinate systems

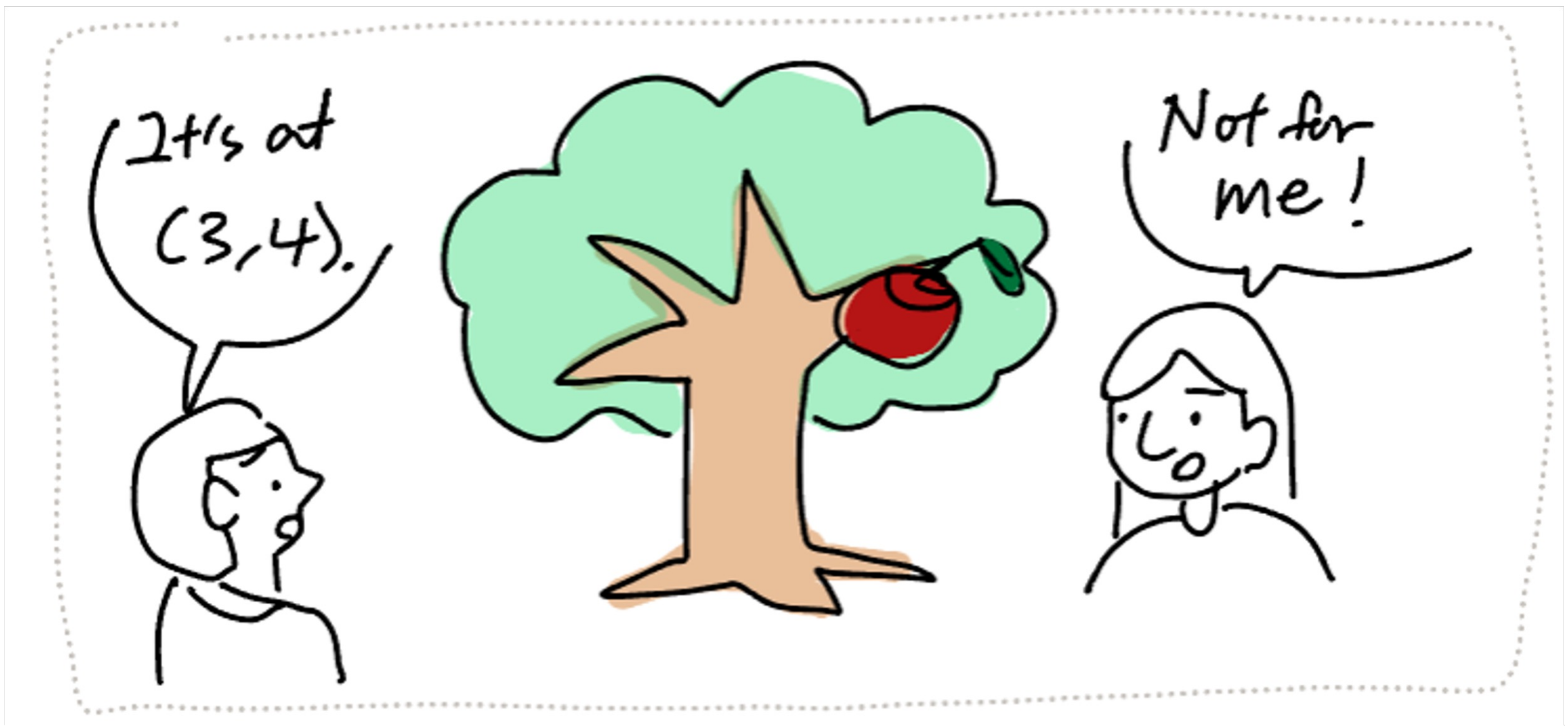
World, Camera, Pixel

Many cameras

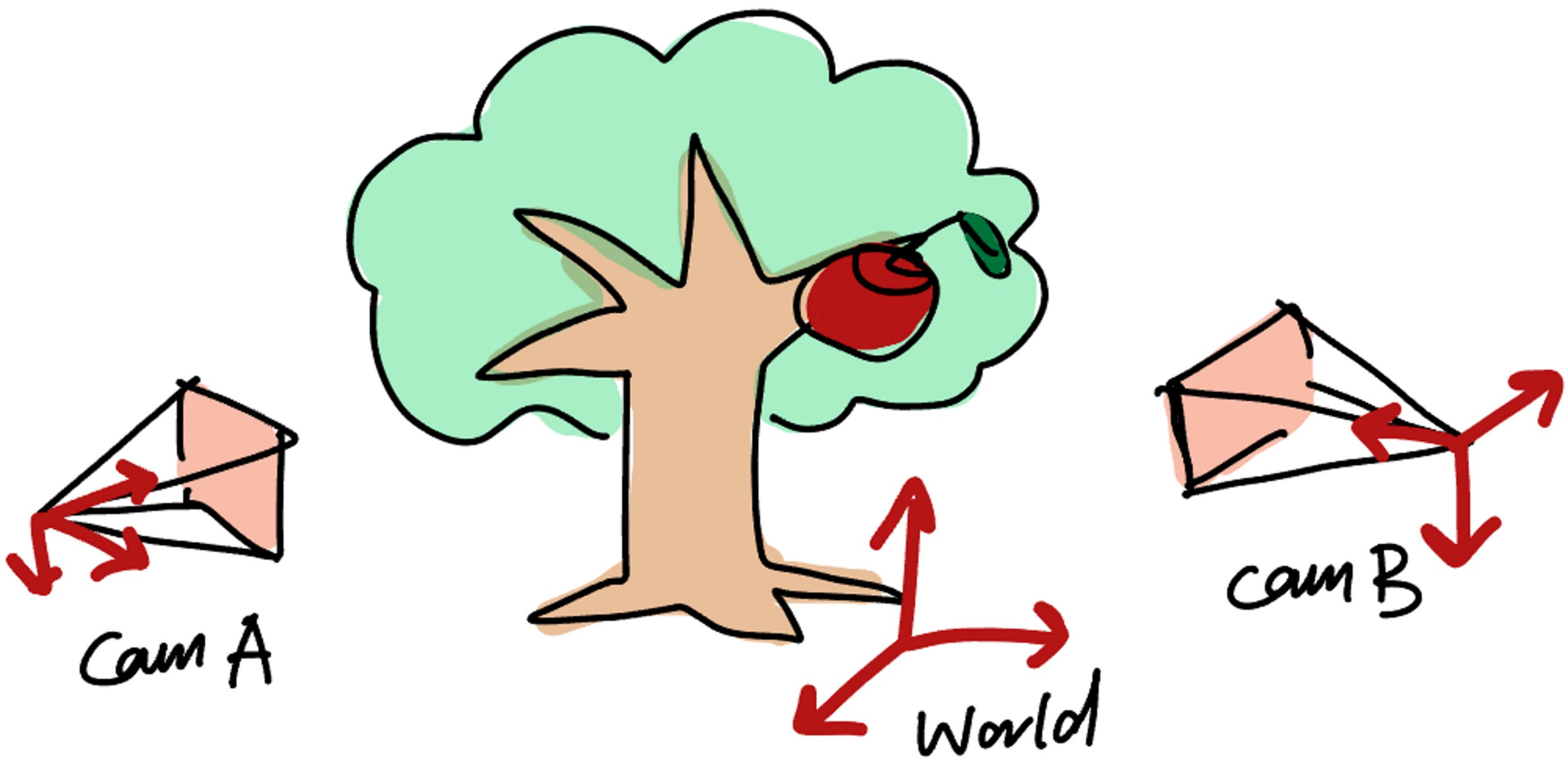
Rays

Coordinate frames matter!

How many times do you say **"your right hand side, or my side?"**

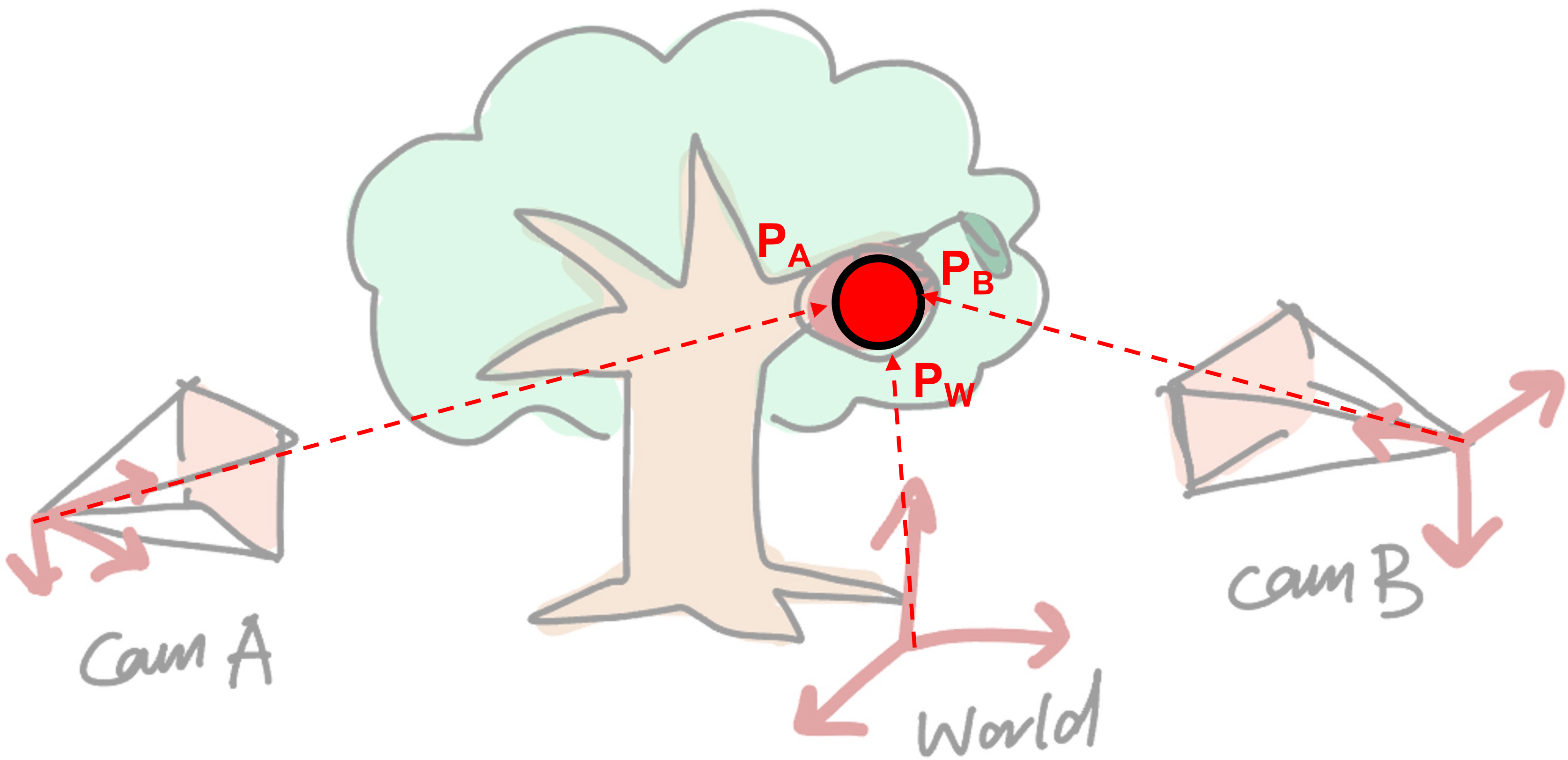


Frames (each is a frame of reference)



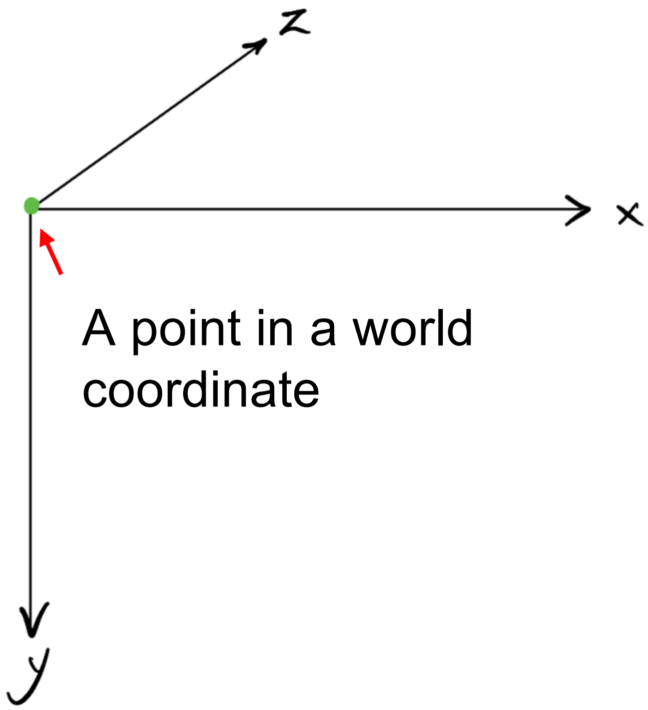
*World = another camera

Same point can be observed from different frames



Examples

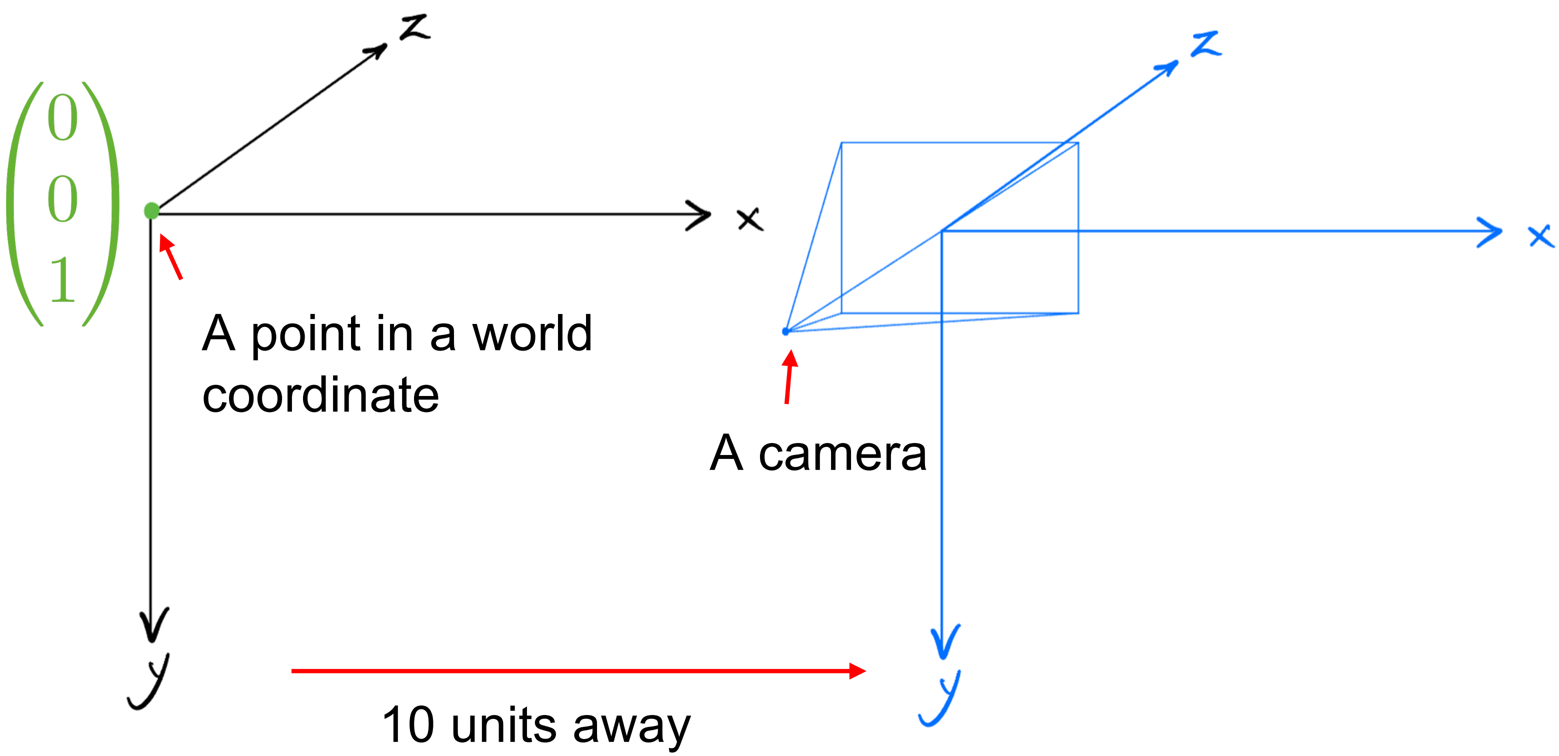
World



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

A point in a world coordinate

Camera

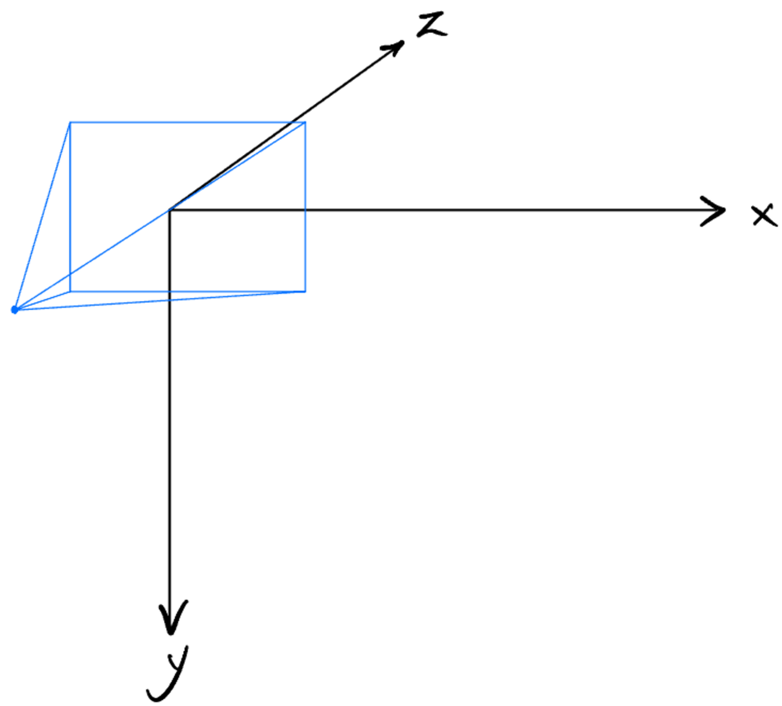


*Where is the point in “camera coordinate”?

World => Camera coordinate

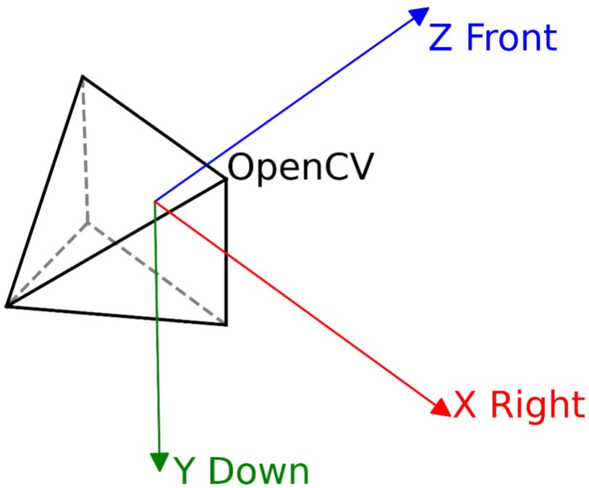
$$\begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix}$$

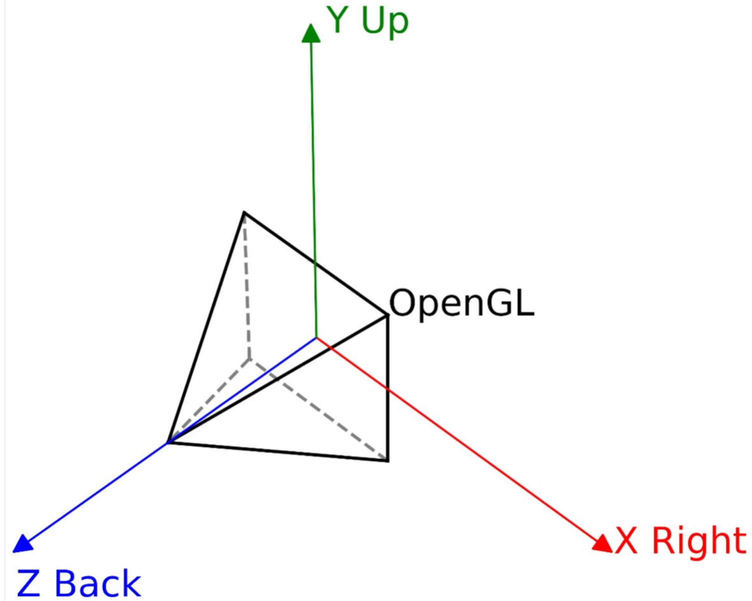
Same point in the
camera coordinate
(shifted -10 units)



*Good luck imaging rotation 🫠

Camera conventions





*We use OpenCV, but there are more…

World to camera transformation

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix}$$

World Camera

World to camera transformation

World to camera transformation

$$\underset{\text{World}}{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} \longrightarrow \underset{\text{Camera}}{\begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix}}$$

$$\underbrace{\begin{pmatrix} -10 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}_c} = \underbrace{(?)}_{T_{w \to c}} \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}_w}$$

Homogeneous

$$\begin{pmatrix} -10 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{4\text{x}4} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

World to camera transformation

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix}$$

World Camera

$$\underbrace{\begin{pmatrix} -10 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}_c} = \underbrace{(?)}_{T_{w\to c}} \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}_w}$$

Rotation Translation

$$\begin{pmatrix} -10 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{4\text{x}4} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

World to camera transformation

$$\begin{pmatrix}0\\0\\1\end{pmatrix} \longrightarrow \begin{pmatrix}-10\\0\\1\end{pmatrix}$$

World Camera

$$\underbrace{\begin{pmatrix}-10\\0\\1\\1\end{pmatrix}}_{\mathbf{x}_c} = \underbrace{(?)}_{T_{w\to c}} \underbrace{\begin{pmatrix}0\\0\\1\\1\end{pmatrix}}_{\mathbf{x}_w}$$

$$\begin{pmatrix}-10\\0\\1\\1\end{pmatrix} = \begin{pmatrix} \boxed{\quad \mathbf{R} \quad} & \boxed{\ \mathbf{t}\ } \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix}_{\text{4x4}} \begin{pmatrix}0\\0\\1\\1\end{pmatrix}$$

World to camera transformation

$$\underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}_{\text{World}} \longrightarrow \underbrace{\begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix}}_{\text{Camera}}$$

$$\underbrace{\begin{pmatrix} -10 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}_c} = \underbrace{(?)}_{T_{w \to c}} \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}_w}$$

Shorthand

$$\mathbf{x}_c = \mathbf{T}_{w \to c} \mathbf{X}_w$$

$$\begin{pmatrix} x_c \\ 1 \end{pmatrix} = \begin{pmatrix} \boxed{\mathbf{R}} & \boxed{\mathbf{t}} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix}_{4\text{x}4} \begin{pmatrix} x_w \\ 1 \end{pmatrix}$$

*Lenient usage of homogeneous coordinates

World to camera transformation

$$\mathbf{x}_c = \mathbf{T}_{w \to c}\, \mathbf{x}_w$$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix}$$

World Camera

$$\mathbf{T}_{w \to c} = \begin{pmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Camera to world

$$\mathbf{x}_w = \mathbf{T}_{w \to c}\, \mathbf{x}_c$$

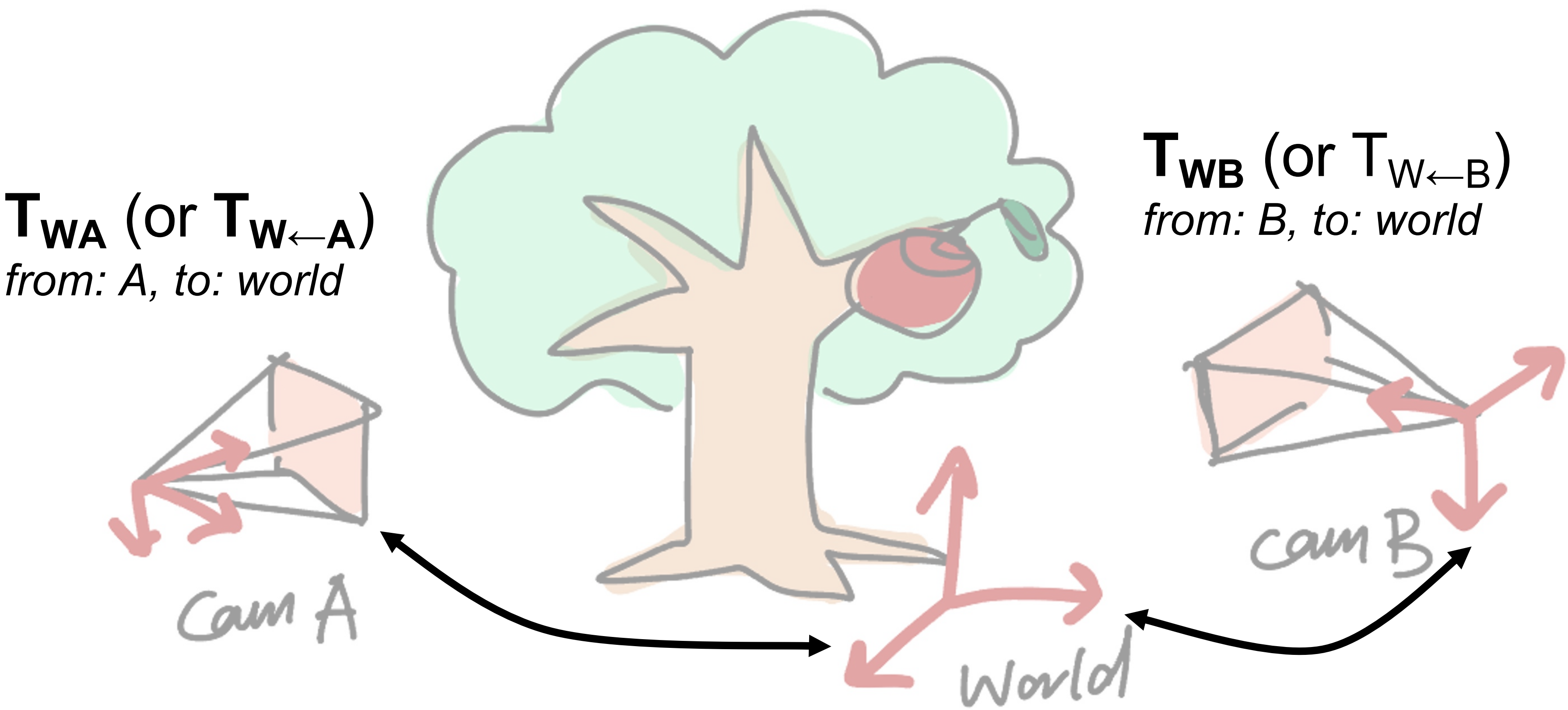
$$\begin{pmatrix} -10 \\ 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Camera World

$$\mathbf{T}_{c \to w} = \mathbf{T}_{w \to c}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

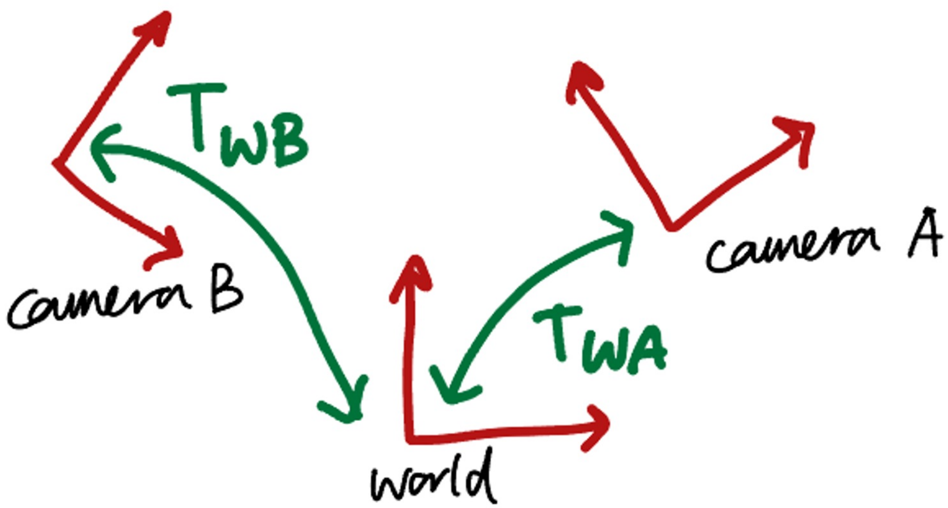
Also called “camera extrinsics”

Transformations take you from one frame to another

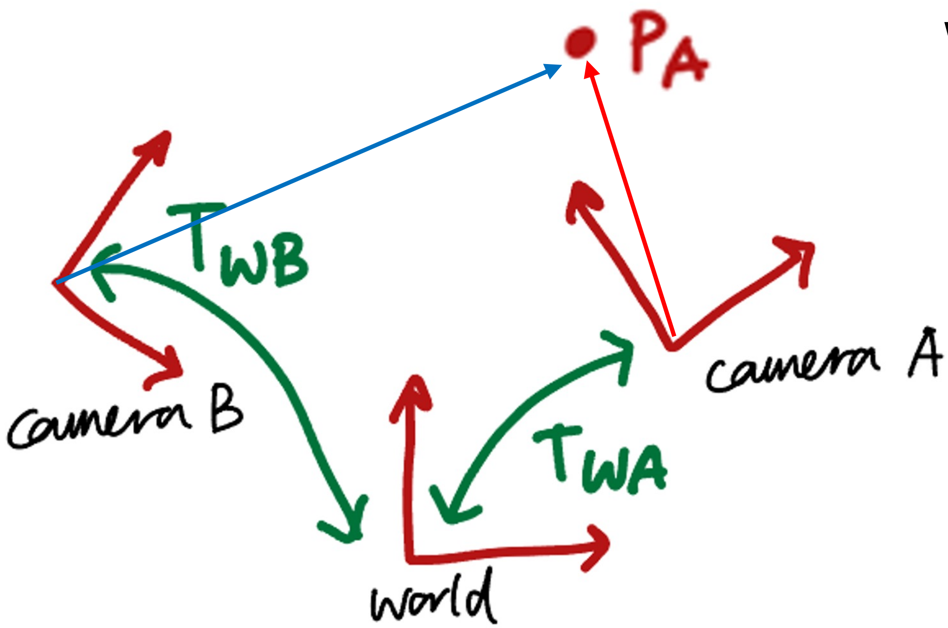


*It’s a convention, don’t blame us!

Chaining transformations



Chaining transformations



What is P_B?

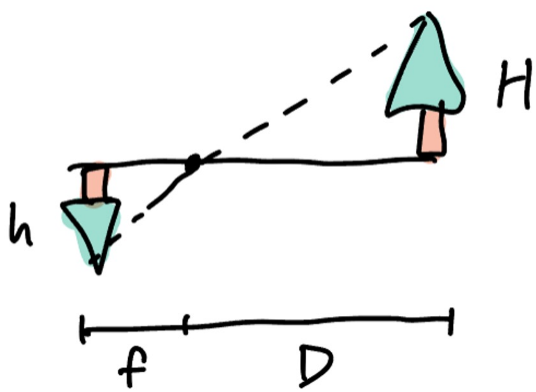
$$P_B = T_{BW} T_{WA} P_A$$
$$= \boxed{T_{WB}^{-1}} T_{WA} P_A$$

Inverse!

Do part 1a-d

Will continue on pinhole camera in 3D

Pinhole cameras + 3D



$$h = f\frac{H}{D}$$

Sensor offset

$$K = \begin{bmatrix} f_x & 0 & \boxed{c_x} \\ 0 & f_y & \boxed{c_y} \\ 0 & 0 & 1 \end{bmatrix}$$

Point in 3D

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Pixel coordinate

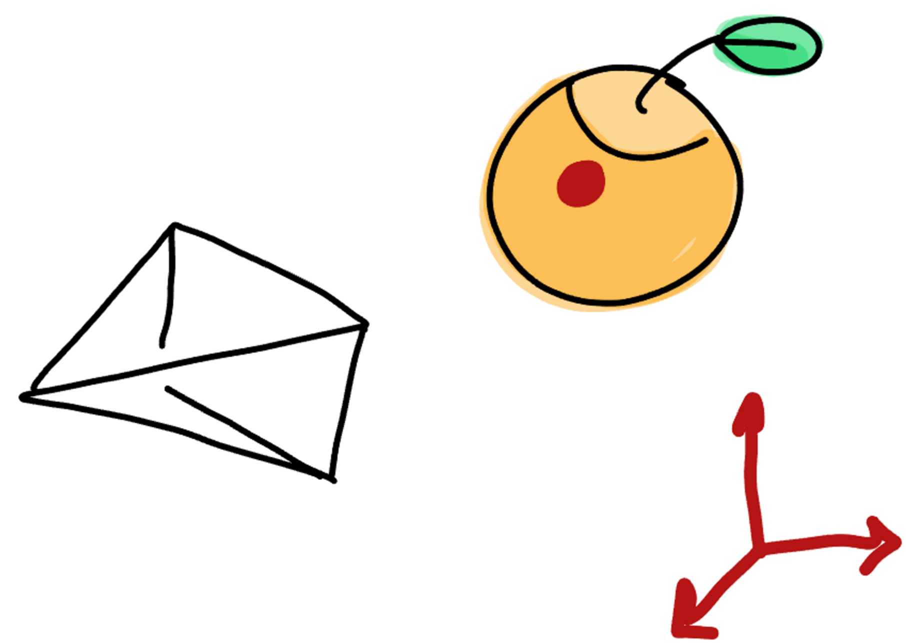
Expand

$$\Rightarrow \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f_x x \\ f_y y \\ z \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

*fx != fy sometimes because of non-square pixels

Example exercise!

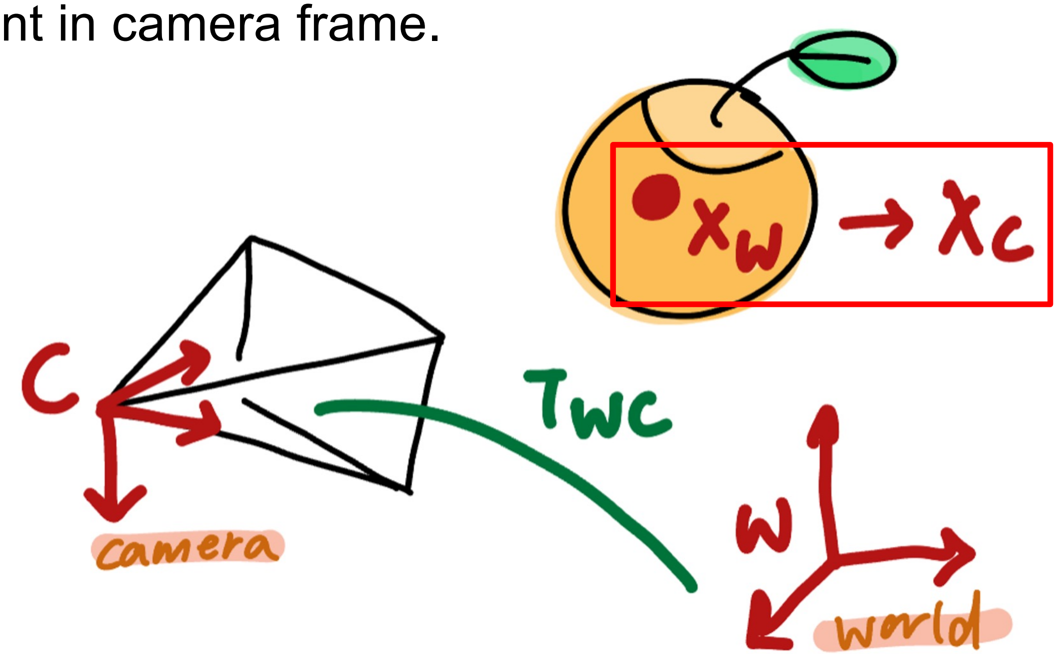
Let’s put a point on this orange onto an image.



Example exercise!

Let's put a point on this orange onto an image.

1. Put the point in camera frame.



Point in camera frame

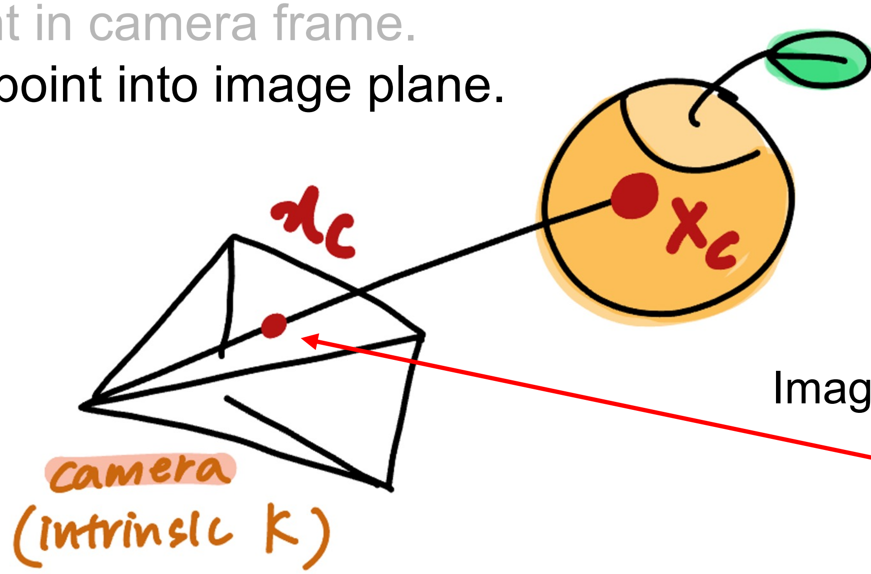
$$\mathbf{X}_c = \mathbf{T}_{cw}\,\mathbf{X}_w$$

$$= (\mathbf{T}_{wc})^{-1}\,\mathbf{X}_w$$

Example exercise!

Let’s put a point on this orange onto an image.

1. Put the point in camera frame.
2. Project the point into image plane.



Point in camera frame

$$\mathbf{X}_c = \mathbf{T}_{cw}\,\mathbf{X}_w$$

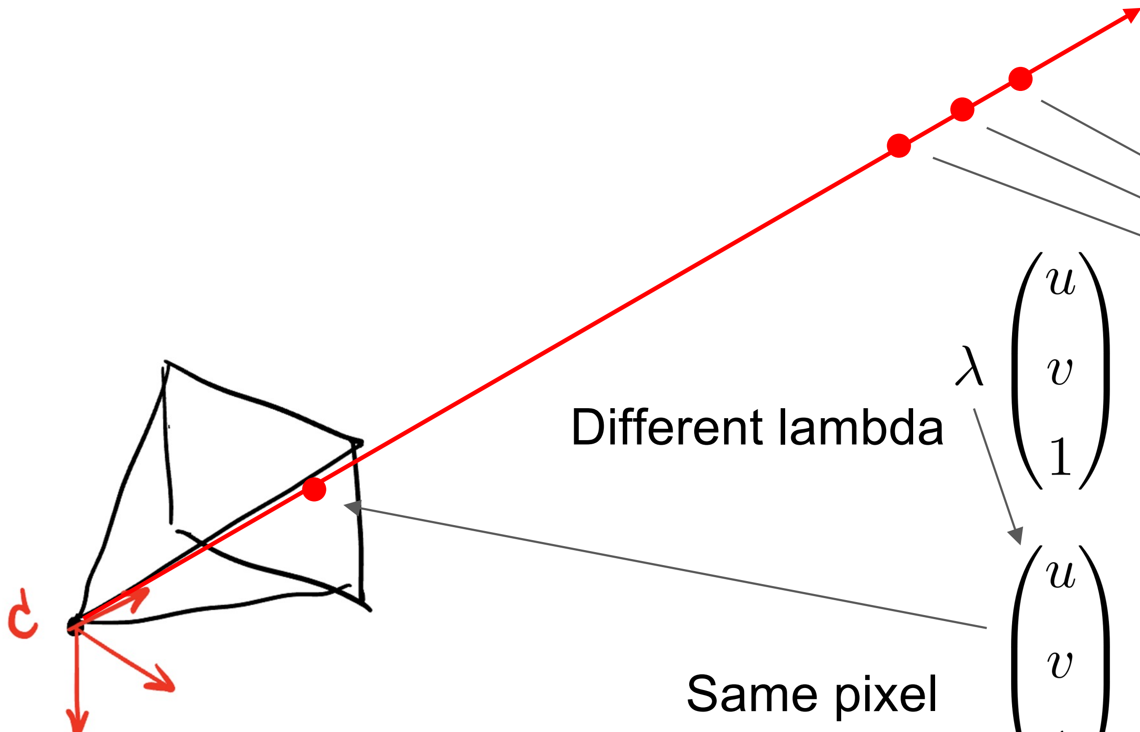
$$= (\mathbf{T}_{wc})^{-1}\,\mathbf{X}_w$$

Image coordinate

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K\mathbf{X}_c$$

Rays

Points along the ray projected on the same pixel



Different point

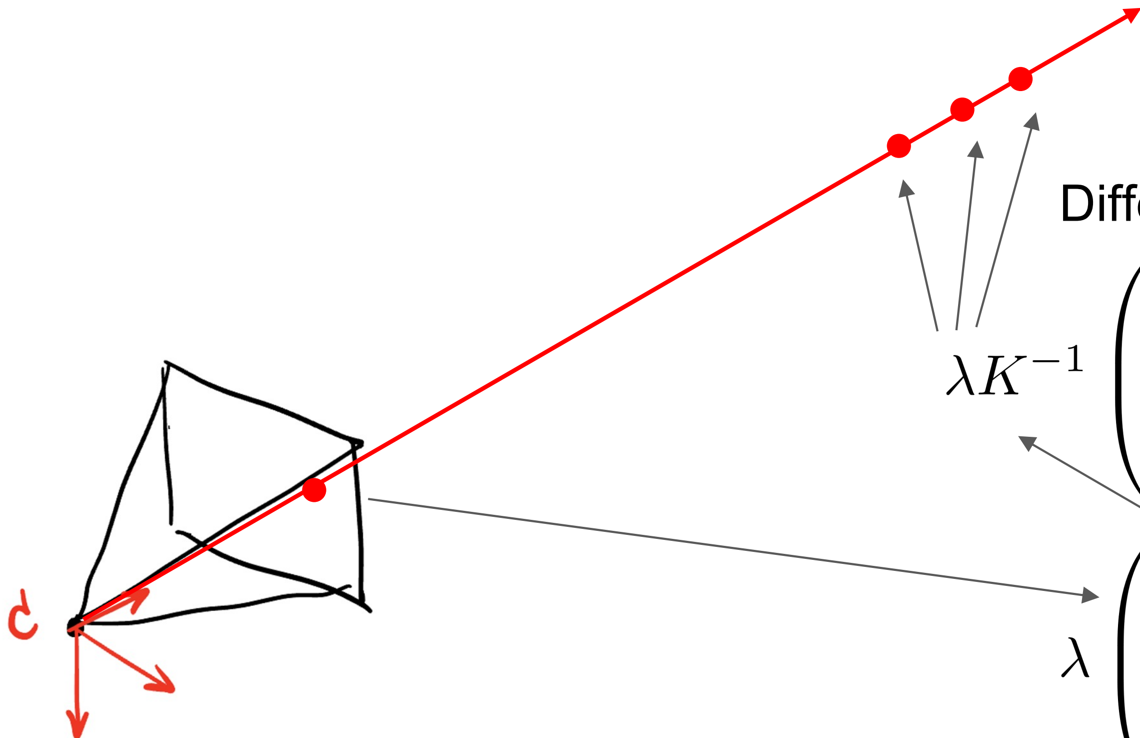
$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Different lambda

Same pixel

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

From a pixel => we can recover a ray

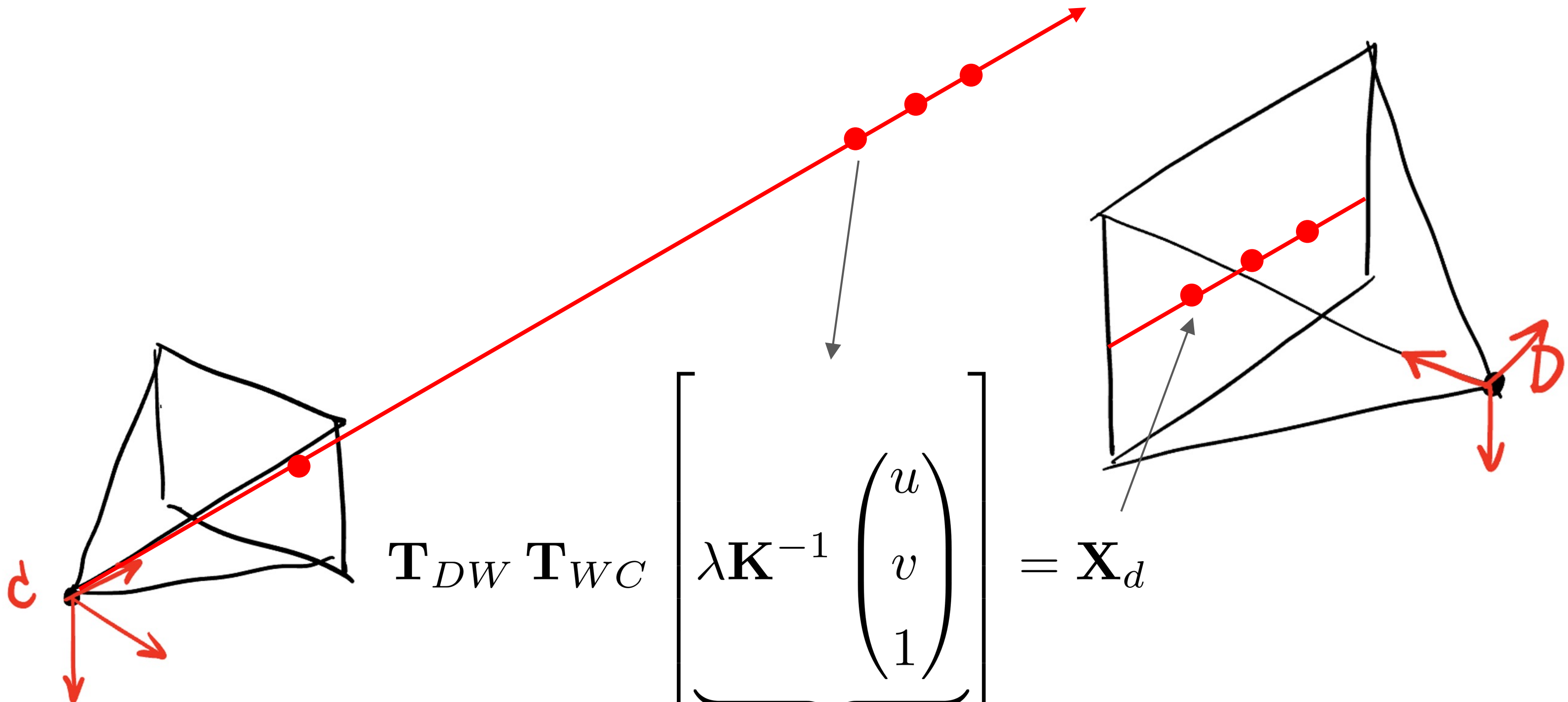


Different lambdas become line!

$$\lambda K^{-1}\begin{pmatrix}u\\v\\1\end{pmatrix}=\begin{pmatrix}x\\y\\z\end{pmatrix}$$

$$\lambda\begin{pmatrix}u\\v\\1\end{pmatrix}=K\begin{pmatrix}x\\y\\z\end{pmatrix}$$

Seeing a ray from a different camera



$$\mathbf{T}_{DW}\,\mathbf{T}_{WC}\left[\underbrace{\lambda\mathbf{K}^{-1}\begin{pmatrix}u\\v\\1\end{pmatrix}}_{\mathbf{x}_c}\right] = \mathbf{X}_d$$

Let’s do part 2